EDB BigAnimal

Version release

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Built at 2024-05-16T22:11:04

1 EDB BigAnimal 5
2 Getting started with BigAnimal 5
2.1 Creating a cluster, connecting, and creating a database 5
2.2 Going deeper with BigAnimal 7
2.2.1 Create an EDB account to use your free credits and BigAnimal 7
2.2.2 Create a cluster 8
2.2.2.1 Creating a cluster on the command line 8
2.2.2.2 Creating a cluster using the web portal 10
2.2.3 Connect to your cluster 10
2.2.3.1 Connect to your cluster using psql 11
2.2.3.2 Connect to your cluster using pgAdmin 11
2.2.3.3 Connect to your cluster using DBeaver 12
2.2.4 Experimenting with the BigAnimal free trial 13
2.2.4.1 Back up and restore a cluster 13
2.2.4.3 Import data from external sources 15
3 Overview of service 16
3.1 Supported cluster types 16
3.1.1 Single node 16
3.1.2 Primary/standby high availability 16
3.1.3 Distributed high availability 17
3.2 Faraway replicas 21
3.3 Supported Postgres extensions and tools 24
3.4 EDB PgBouncer 25
3.5 Foreign data wrappers 25
3.6 Security 25
3.6.1 Security compliance and certifications 26
3.7 Supported regions 27
3.7.1 Country and geographical region reference 31
3.8 Shared responsibilities 36
3.9 Database version policy 37
3.10 Periodic maintenance 38
3.11 BigAnimal support options 38
3.11.1 Managing your Support cases 39
4 Choosing your deployment option 40
4.1 Deploying with AWS 41
4.2 Deploying with Google Cloud 41
4.3 Deploying with Azure 41
4.4 Deploying with BigAnimal's cloud 41
5 Choosing your cluster type 42
6 Choosing your Postgres distribution 42
5 Setting up your identity provider 42
5.1 Setting up specific identity providers 45
6 Setting up your Azure Marketplace account 45
7 Checking your cloud readiness 46
7.1 Preparing your Azure account 46
7.1.1 Understanding requirements in Azure 50
7.2 Preparing your AWS account 52
7.3 Preparing your Google Cloud account 53

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

7.3.1 Understanding requirements in Google Cloud 55
8 Connecting your cloud 56
8.1 Connecting your AWS cloud 57
8.2 Connecting your Azure cloud 58
8.3 Connecting your Google Cloud 59
9 Managing regions 60
10 Creating a cluster 61
10.1 Cluster networking architecture 65
10.2 Creating a distributed high-availability cluster 66
11 Managing a cluster 69
6 Using your cluster 70
6.1 Managing Postgres access 70
6.1.1 pg_ba_admin role 73
6.2 Managing Postgres extensions 74
6.3 Managing replicas 76
6.4 Connecting to your cluster 79
6.4.1 Connecting from Azure 79
6.4.1.1 Virtual network peering example 86
6.4.1.2 VNet-VNet example 88
6.4.2 Connecting from AWS 91
6.4.2.1 VPC peering 94
6.4.3 Connecting from Google Cloud 96
6.4.3.1 VPC peering 99
6.4.4 Connecting from a client app 99
6.5 Modifying your cluster 101
6.5.1 Modifying database configuration parameters 103
6.6 Backing up and restoring 105
6.7 Monitoring and logging 106
6.7.1 Metrics details 106
6.7.2 Viewing metrics and logs from Azure 115
6.7.3 Setting real-time alerts in Azure 117
6.7.4 Viewing metrics and logs from AWS 118
6.7.5 Monitoring using BigAnimal Observability 120
6.7.6 Other monitoring and logging solutions 124
6.7.6.1 Example metrics 125
6.7.7 Third-party monitoring integrations 129
6.7.7.1 Datadog 131
6.7.7.2 New Relic 132
6.8 Fault injection testing 133
6.9 Deleting your cluster 135
6.10 Performing a major version upgrade of Postgres on BigAnimal 136
6.11 Analyzing your data with Apache Superset 139
6.12 Demonstration of Oracle SQL compatible functions and syntax 141
6.13 BigAnimal Terraform provider 146
6.14 PGD CLI on BigAnimal 147
6.15 Postgres Distributed (PGD) presets on BigAnimal 149
6.16 Tagging BigAnimal resources 151
6.16.1 Creating and managing tags 151

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

7 Administering your account 153
7.1 Managing user access 153
7.2 Managing projects 155
7.3 Managing Superset access 158
7.4 Reviewing account activity 158
7.5 Customizing compliance rules and policies 159
7.5.1 Customizing AWS Config rules 159
7.5.2 Customizing Azure policy definitions 159
7.5.3 Customizing Google Cloud compliance policies 163
7.6 Notifications 163
8 Pricing and billing 164
9 Migrating databases to BigAnimal 167
9.1 Importing an existing Postgres database 168
9.2 Bulk loading data into PGD clusters 171
10 Using the API and CLI 178
10.1 Access key 178
10.2 Using the BigAnimal API 179
10.3 Using the BigAnimal CLI 180
10.3.1 Managing clusters using the CLI 184
10.3.2 Using BigAnimal features with the CLI 193
11 BigAnimal release notes 195
11.1 BigAnimal March 2024 release notes 195
11.2 BigAnimal February 2024 release notes 195
11.3 BigAnimal January 2024 release notes 196
11.4 BigAnimal December release notes 196
11.5 BigAnimal November release notes 196
11.6 BigAnimal October release notes 196
11.7 BigAnimal September release notes 196
11.8 BigAnimal August release notes 197
11.9 BigAnimal July release notes 197
11.10 BigAnimal June release notes 197
11.11 BigAnimal May release notes 197
11.12 BigAnimal April release notes 198
11.13 BigAnimal March release notes 198
11.14 BigAnimal February release notes 198
11.15 BigAnimal January release notes 198
12 Known issues and limitations 198
12.1 Known issues with distributed high availability/PGD 199
13 BigAnimal knowledge base and FAQs 200

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

1 EDB BigAnimal

EDB BigAnimal is a fully managed database-as-a-service that runs in your cloud account or BigAnimal's cloud account, where it's operated by one of the builders of Postgres. BigAnimal makes it easy to set up, manage, and scale your databases. You can use PostgreSQL or EDB Postgres Advanced Server with Oracle compatibility, or you can choose distributed high-availability cluster types to use geographically distributed databases.

To sample EDB BigAnimal, get started with $300 in free credits.

2 Getting started with BigAnimal

When you first sign up for BigAnimal, you'll have $300 in free credits to experience all the features and capabilities of BigAnimal.

Create an account

To get started with BigAnimal, you'll need to create an EDB account.

Registering an account on enterprisedb.com gives you access to BigAnimal, as well as EDB's training and software downloads. Accounts are free, and registration takes less than a minute.

Start using BigAnimal

Once you have set up your account, you can log in to the BigAnimal portal with full access to all its features and capabilities. See Quick start for a walkthrough of configuring a cluster and ideas for further exploration.

What happens when you use up your free credits?

If you want to keep your cluster live after your free credits run out, simply add a credit card or payment method (see Payments and billing). If you don't add a credit card or payment method, your clusters and their backup data are deleted within seven days.

2.1 Creating a cluster, connecting, and creating a database

Follow this quick walkthrough to configure a Postgres cluster on BigAnimal.

Step 1: Create an account and sign into the portal

If you haven't done so already, you'll need to create your EDB account.

Then, use your newly created account to access the BigAnimal portal.

Step 2: Create a cluster

1. On the overview page, select **Create New Cluster**.

   You should now find yourself at the Create Cluster page.

2. Select the options for your cluster. See Creating a cluster for more information.

Managing your cluster

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

After you select **Create Cluster**, you return to the Clusters page with your newly configured cluster now populating the list. From the Clusters page, you can view the details, create a replica of your cluster, and edit and delete your cluster.

Step 3: Connect to your new cluster

1. Select your cluster to get an overview of how it has been configured. Select the **Connect** tab to see more information about how to connect to your cluster.

2. Select the **Overview** tab and copy the **Quick Connect** command. Paste it into a terminal where psql is installed. It will prompt for your password and put you on a SQL command line. For example:

```
psql -W "postgres://edb_admin@p-qzwv2ns7pj.pg.biganimal.io:5432/edb_admin?sslmode=require"
Password:
__OUTPUT__
psql (15.2)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

edb_admin=>
```

> **Note**
>
> While psql is a good all-around option for working with Postgres databases, you can use the client of your choice. See Connect to a cluster for more ideas.

Things to try

Create a new database

We're going to create some sample math data, so we're going to create a database called `math`. We could use the default `edb_admin` database, but best practice is to isolate data.

1. Create a new `math` database.

```
create user math with password 'math_password';
create database math;
```

2. Grant the `math` role to edb_admin.

```
grant math to edb_admin;
```

3. Connect to the `math` database. You're prompted for the edb_admin password you provided in Step 2.

```
\connect math
```

Populate a table and query it

We're going to use temporary tables to calculate prime numbers using a Sieve of Eratosthenes.

1. Create a table called `primes` for storing prime numbers.

```
CREATE TABLE primes
(
  num
INTEGER,
  PRIMARY KEY (num)
);
```

2. Populate the table with all prime numbers up to 1000. (This code is based on code from David Fetter.)

```
-- Based on https://wiki.postgresql.org/wiki/Sieve_of_Eratosthenes

WITH RECURSIVE
t0(m) AS
(
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
    VALUES(1000)
),
t1(n) AS
(
    VALUES(2)
UNION ALL
    SELECT n+1 FROM t1 WHERE n < (SELECT m FROM
t0)
),
t2 (n, i) AS
(
    SELECT 2*n, 2
    FROM t1 WHERE 2*n <= (SELECT m FROM
t0)
UNION ALL

(
        WITH t3(k) AS
(
            SELECT max(i) OVER () + 1 FROM t2
        ),
        t4(k) AS
(
            SELECT DISTINCT k FROM
t3

)
        SELECT k*n,
k
        FROM
            t1
        CROSS JOIN
            t4
        WHERE k*k <= (SELECT m FROM
t0)

)
)
INSERT INTO primes
(
  SELECT n FROM t1
EXCEPT
  SELECT n FROM
t2
  ORDER BY 1
);
```

3. Select the largest prime number less than 1000.

```
 SELECT max(num)
FROM
primes
WHERE num <
1000;
```

Further reading

Now that you've got the basics, see what else BigAnimal offers:

- Backup and restore
- Import data
- CLI reference
- API reference

2.2 Going deeper with BigAnimal

The Quick start presents one way to get started, but there are quite a few things to try! Take a look at some ideas to get you started experimenting with the capabilities that it provides.

2.2.1 Create an EDB account to use your free credits and BigAnimal

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

You need to register with EDB to access BigAnimal and your $300 in free credits, as well as EDB's training and software downloads. EDB provides free Postgres training, found here (https://www.enterprisedb.com/training).

Registration takes less than a minute.

1. Navigate to https://biganimal.com.
2. Select **Test drive for free**.
3. Fill in the form, and agree to the terms and conditions and privacy policy.
4. Select **Submit**.
5. Check your email for a message from EDB that contains an activation link.
6. Select the activation link, and enter a password for your EDB account.

You're then redirected to the BigAnimal portal.

> **Already have an EDB account?**
>
> If you already signed up for an EDB account, navigate to (https://portal.biganimal.com) and select **EDB Account** to log in.

2.2.2 Create a cluster

We'll show two methods of creating a cluster:

- Using the portal
- Using the command line

You can also create clusters programmatically using the fully featured BigAnimal API.

2.2.2.1 Creating a cluster on the command line

We'll be using the BigAnimal command line interface, which is a convenient wrapper to the BigAnimal API. To start, download the latest binary and move it to wherever your system finds executable files (somewhere on your PATH).

> **Linux and MacOS note**
>
> If you're on a Linux or MacOS system, you'll need to mark the `biganimal` file as executable by running `chmod +x [/path/to/biganimal]` before you can use it.

Example (for Linux or MacOS):

```
curl -LO "https://cli.biganimal.com/download/$(uname -s)/$(uname -m)/latest/biganimal"
mkdir -p $HOME/bin
export PATH=$HOME/bin:$PATH
mv ./biganimal $HOME/bin/biganimal
chmod +x $HOME/bin/biganimal
```

Next, you need to create a credential on BigAnimal. You can pick any username you prefer.

```
biganimal credential create -n newuser
```

```
Visit this URL to login: https://auth.biganimal.com/activate?user_code=XXXX-XXXX
or press [Enter] to continue in the web browser
Then visit: https://auth.biganimal.com/activate
press [Enter] to continue in the web browser...
```

> **Linux dependencies**
>
> The BigAnimal CLI uses the xdg-open utility to open a browser on Linux systems. On minimal systems, you might need to install this dependency before creating a credential.

The command will direct you to open a webpage and copy the randomly generated, one-time code. You'll need to log in (or already be logged in) to activate the credentials. You can see the credentials you've verified on the command line.

```
biganimal credential show
```

```
┌─────────────────────────────────────────────────────────┐
│ Credentials                                             │
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
┌──────────┬──────────────────────────────┬──────┬──────────────────┐
| name     | address                      | port | context credential |
├──────────┼──────────────────────────────┼──────┼──────────────────┤
| new-user | portal.enterprisedb.network | 443  | x                |
└──────────┴──────────────────────────────┴──────┴──────────────────┘
```

> **Caution**
>
> If you add another credential, the newly created credential will be set as the new default context credential. You'll need to add `--credential [newuser]` to the following commands to override the default credentials. If you have only one, the option isn't needed. You can change the default credential using `biganimal config set context_credential [name]`.

Use the `biganimal region show` command to see the available active regions you can pick from for your cluster.

Edit a new file called `create_cluster.yaml`:

```
clusterName: test_cluster # or a name of your
choosing
password:
clusterArchitecture:
postgresType:
postgresVersion:
provider:
region:  # Select from the options given by the `region show`
command
instanceType:
volumeProperties:
volumeType:
networking:
highAvailability:
```

Use the config file to create a new cluster:

```
biganimal cluster create --config-file create_cluster.yaml
```

```
Are you sure you want to create cluster "test_cluster"? [y|N]:
```

Select `y`.

If successful, `cluster create` will give you the ID of your new cluster (you'll use this to manage it) as well as the command you can use to check the status of your new cluster.

```
__OUTPUT__
Create Cluster operation is started
Cluster ID is "p-xxxxxxxxxx"
To check current state, run: biganimal cluster show --id p-xxxxxxxxxx
```

Here's example output from the `cluster show` command:

```
biganimal cluster show --id p-xxxxxxxxxx
```

```
┌──────────────────────────────
```

In the meantime, you can get the connection string for your cluster. For example:

```
biganimal cluster show-connection --id p-xxxxxxxxxxx
```

```
┌─────────────┬──────────────────────────────────────────────────────────────────────────────────┐
│ Access Type │ Connection String                                                                │
├─────────────┼──────────────────────────────────────────────────────────────────────────────────┤
│ read-write  │ postgresql://edb_admin@p-xxxxxxxxxxx.pg.biganimal.io:5432/edb_admin?sslmode=require'     │
│ read-only   │ Not Supported                                                                    │
└─────────────┴──────────────────────────────────────────────────────────────────────────────────┘
```

After the cluster is created, log in by way of psql. Use the password from the `config` file for edb_admin.

```
psql 'postgres://edb_admin@p-xxxxxxxxxx.pg.biganimal.io:5432/edb_admin?sslmode=require'
```

> **Other options for connecting**
>
> Sure, psql is great, but maybe you want to use another client. See Connect to your cluster for other options.

Next steps

Connect to your cluster.

Further reading

BigAnimal CLI reference and Creating a cluster in the full version documentation.

2.2.2.2 Creating a cluster using the web portal

Navigating to the Create Cluster page

1. Navigate to the BigAnimal portal. Sign in with your account.
2. Select the **Clusters** link on the left to navigate to the Clusters page.
3. Select **Create New Cluster**, which opens the Create Cluster page.
4. Select the options you want for your cluster. See Creating a cluster in the full version documentation for details on the options.

Provisioning

Select **Create Cluster**, and you're brought back to the Clusters page with your newly configured cluster now populating the list.

Initially, the new cluster is being provisioned. This usually takes a minute or two but can take up to an hour. A progress bar is shown near the right.

Select the provisioned cluster to view the parameters you'll need to connect to it.

Next steps

Connect to your cluster.

2.2.3 Connect to your cluster

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

You've created a cluster. Now what? You probably want to connect to it!

You can connect to BigAnimal clusters as you would any other Postgres database.

You can find all of the parameters you need to connect on the portal by selecting the name of your cluster on theClusters page and then selecting the **Connect** tab.

Following are a few examples of connecting from common clients.

2.2.3.1 Connect to your cluster using psql

> **Note**
>
> You can find all of the parameters you need to connect on the portal by selecting the name of your cluster on theClusters page and then selecting the **Connect** tab.

psql is the standard PostgreSQL REPL and an all-around useful tool for working with PostgreSQL databases. In addition to letting you submit SQL and view the results, it offers quite a few built-in commands for inspecting and managing the database that make light work of even complex tasks.

You can find an example of connecting to a BigAnimal cluster by way of psql right on your cluster's **Overview** tab on the portal. For example:

```
psql -W "postgres://edb_admin@p-xxxxxxxxxxx.pg.biganimal.io:5432/edb_admin?sslmode=require"
```

The URI is all that's necessary for psql. While you *can* pass each element of that URL individually (pulling the information from the **Connect** tab on the portal, for instance), psql will parse it out for you and prompt for the password.

In case you're unfamiliar with PostgreSQL's URI format, you can find it documented on postgresql.org.

> **EDB Postgres Advanced Server enhancements**
>
> If your cluster is running EDB Postgres Advanced Server, you might want to install the version of psql that ships with it. It includes a few nice additions, such as tab completion for Oracle database-compatible syntax. However, the standard PostgreSQL client works with EDB Postgres Advanced Server as well.

Next steps

Experiment!

Further reading

Connecting to your cluster in the full version documentation.

2.2.3.2 Connect to your cluster using pgAdmin

> **Note**
>
> You can find all of the parameters you need to connect on the portal by selecting the name of your cluster on theClusters page and then selecting the **Connect** tab.

The pgAdmin project allows you to inspect, monitor, manage, and query your cluster's databases from a desktop or web UI.

From the welcome page of pgAdmin, select **Add New Server**. You're prompted to configure the connection.

Enter `BigAnimal Trial` for the name (or use the name you gave to your cluster!), and then select **Connection**.

1. In the **Host name/address** field, enter your cluster's hostname.
2. In the **Maintenance database** field, enter `edb_admin`. This is the default database that pgAdmin connects to.
3. In the **Username** field, enter `edb_admin`.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

4. In the **Password** field, enter the password you provided when configuring your cluster. You might want to save this for convenience while testing.

5. Select the **SSL** tab, and change SSL mode to **Require**.

6. Select **Save**. pgAdmin tries to establish a connection to your database. When successful, it displays the dashboard along with the list of available databases on the left.

Next steps

Experiment!

Further reading

Connecting to your cluster in the full version documentation.

2.2.3.3 Connect to your cluster using DBeaver

> **Note**
>
> You can find all of the parameters you need to connect on the portal by selecting the name of your cluster on theClusters page and then selecting the **Connect** tab.

1. Launch DBeaver.

2. To open the **Connect to a database** dialog box, on the toolbar, select **New Database Connection**.

3. Select **PostgreSQL** and select **Next**.

   You might be prompted to download the PostgreSQL JDBC driver.

4. On the **Main** tab:

   - Enter your cluster's hostname in the **Host** field.
   - Enter `edb_admin` in the **Database** field.
   - Enter `edb_admin` in the **Username** field.
   - Enter your cluster's password in the **Password** field.

5. On the **SSL** tab:

   - Select the **Use SSL** check box.
   - For the **SSL mode:** field, select **require**.

6. On the **PostgreSQL** tab, select the **Show all databases** check box.

7. To verify that DBeaver can connect to your cluster, select **Test connection**.

8. To save the connection, select **Finish**.

Next steps

Experiment!

Further reading

Connecting to your cluster in the full version documentation.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

2.2.4 Experimenting with the BigAnimal free trial

Once you've created a cluster, it's time to put it through its paces. Here are some examples to get you started:

- Backup and restore
- Import data

2.2.4.1 Back up and restore a cluster

BigAnimal automatically and continuously saves backups of your clusters. You can restore your cluster to any point in the past after the initial backup on cluster creation. That functionality is included in the free trial as well. While that means you don't have to do anything to generate backups, it's always sensible to test your backups.

If you haven't already, create a cluster on BigAnimal.

We're going to add a database called "baseball," which we'll populate with some Major League Baseball statistics.

```
create database baseball;
```

Now you can switch to your new (and empty) baseball database. You're prompted for the `edb_admin` password you provided when connecting to the cluster.

```
\c
baseball
```

Normally you use the log files to determine the moment in time that you want to restore. For the purposes of this demonstration, let's change the prompt to include a timestamp to have an accurate timing of our activities. For convenience, we'll use the (almost) ISO-8601 format that the restore command accepts.

```
\set PROMPT1 '%`date +"%Y-%m-%dT%H:%M:%S%z"` %/%R%#
'
```

For this demonstration, we're just going to import batter data from the Baseball Databank, which is in CSV form. While it's easy to import the data using PostgreSQL's COPY command, we'll need to first define a table to put that data into. Most of the columns are integers, but there are a few strings to consider as well. You can copy and paste this command into your terminal.

```
CREATE TABLE batters
(
                    id SERIAL,
                    playerid
VARCHAR(9),
                    yearid
INTEGER,
                    stint INTEGER,
                    teamid
VARCHAR(3),
                    lgid VARCHAR(2),
                    g
INTEGER,
                    ab INTEGER,
                    r
INTEGER,
                    h
INTEGER,
                    "2b" INTEGER,
                    "3b" INTEGER,
                    hr INTEGER,
                    rbi
INTEGER,
                    sb INTEGER,
                    cs INTEGER,
                    bb INTEGER,
                    so INTEGER,
                    ibb
INTEGER,
                    hbp
INTEGER,
                    sh INTEGER,
                    sf INTEGER,
                    gidp INTEGER,
                    PRIMARY KEY
(id)
);
```

Now we can populate the table from the internet using the most recent data.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
\COPY batters(playerid,yearid,stint,teamid,lgid,g,ab,r,h,"2b","3b",hr,rbi,sb,cs,bb,so,ibb,hbp,sh,sf,gidp) FROM PROGRAM 'curl
"https://raw.githubusercontent.com/EnterpriseDB/docs/main/product_docs/docs/biganimal/release/free_trial/detail/experiment/data/Battin
"' DELIMITER ',' CSV HEADER
```

Just to prove there's data loaded, let's look at the home run leaders for the 1998 season.

```
SELECT playerid, yearid, teamid,
       rank() OVER (PARTITION BY yearid ORDER BY hr desc)
hr_rank,
       hr
FROM batters
WHERE yearid =
1998
ORDER BY hr_rank LIMIT 5;
```

Suppose someone wanted to revise history a bit.

```
UPDATE batters
SET hr=0
where playerid = 'mcgwima01' AND yearid =
1998;
```

Note the time so we can restore that data later. Verify the data has been changed by rerunning the 1998 home run leader query. Now go ahead and drop the whole table.

```
DROP TABLE
batters;
```

You can verify the table is gone by looking at the list of tables.

```
\dt
```

Restore the cluster using the timestamp before the time you dropped the table.

```
biganimal cluster restore --id p-xxxxxxxxxxx --from-deleted --restore-point 2022-03-31T02:39:40+0000
```

```
✔ New Name for Restored Cluster, Leave empty to continue with source cluster data: ▉
Restored Cluster Password: *************
Region: US East 1
Instance Type: r5.large(2vCPU, 16GB RAM)
Volume Type: General Purpose SSD (GP3)
Volume Properties: gp3 (1 Gi - 16384 Gi, 3000 IOPS, 125 MiB/s tput)
✔ Specify database configuration, for example "application_name=sample_app&array_nulls=true". Leave empty to continue with source
cluster data: ▉
High Availability: Keep Existing
Networking: Keep Existing
Add IP Range to allow network traffic to your cluster from the public Internet: Keep Existing
Are you sure you want to Restore Cluster ? [y|n]: y
Restore Cluster operation is started
Cluster ID is "p-yyyyyyyyyy"
To check current state, run: biganimal show-clusters --id p-yyyyyyyyyy
```

> **Optional arguments for the CLI**
>
> We're only providing the information we have handy to the restore command: the ID of the cluster, the fact that the cluster is deleted, and the timestamp for the restore point. The CLI will prompt for additional information, such as the password or the region to restore to, and allow you to enter or select from a list as appropriate.
>
> If you know ahead of time the options you'll need, you can provide all of them using parameters, and the command will run non-interactively. The values for the other parameters you don't specify will be inherited from the source cluster.

Check the status of the new cluster from the command line.

```
biganimal cluster show
```

Once provisioning is complete, you'll want to grab the new connection information. The hostname changed!

```
biganimal cluster show-connection --id  p-yyyyyyyyyy
```

Log in to the new cluster. Be sure to use the new hostname, and use the edb_admin password you provided when restoring the cluster.

Verify that the batters table was restored:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
select playerid, yearid, teamid,
       rank() OVER (PARTITION BY yearid ORDER BY hr desc)
hr_rank,
       hr
from batters
where yearid =
1998
order by hr_rank limit 5;
```

Further reading

BigAnimal CLI reference and Backing up and restoring in the full version documentation.

2.2.4.3 Import data from external sources

PostgreSQL includes a variety of ways to import data. Here, we'll show how to import a CSV file from the internet.

For this demonstration, we're going to import batter data from the Baseball Databank, which is in CSV form. While it's easy to import the data using PostgreSQL's COPY command, we'll need to first define a table to put that data into.

We're going to add a database called "baseball," which we'll populate with some Major League Baseball statistics.

```
create database baseball;
```

Now you can switch to your new (and empty) baseball database.

```
\c
baseball
```

You can copy and paste this command into your terminal.

```
CREATE TABLE batters
(
                    id SERIAL,
                    playerid
VARCHAR(9),
                    yearid
INTEGER,
                    stint INTEGER,
                    teamid
VARCHAR(3),
                    lgid VARCHAR(2),
                    g
INTEGER,
                    ab INTEGER,
                    r
INTEGER,
                    h
INTEGER,
                    "2b" INTEGER,
                    "3b" INTEGER,
                    hr INTEGER,
                    rbi
INTEGER,
                    sb INTEGER,
                    cs INTEGER,
                    bb INTEGER,
                    so INTEGER,
                    ibb
INTEGER,
                    hbp
INTEGER,
                    sh INTEGER,
                    sf INTEGER,
                    gidp INTEGER,
                    PRIMARY KEY
(id)
);
```

Now we can populate the table from the internet using the most recent data.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
\COPY batters(playerid,yearid,stint,teamid,lgid,g,ab,r,h,"2b","3b",hr,rbi,sb,cs,bb,so,ibb,hbp,sh,sf,gidp) FROM PROGRAM 'curl
"https://raw.githubusercontent.com/EnterpriseDB/docs/main/product_docs/docs/biganimal/release/free_trial/detail/experiment/data/Battin
"' DELIMITER ',' CSV HEADER
```

Just to prove there's data loaded, let's look at the home run leaders for the 1998 season.

```
SELECT playerid, yearid, teamid,
       rank() OVER (PARTITION BY yearid ORDER BY hr desc)
hr_rank,
       hr
FROM batters
WHERE yearid =
1998
ORDER BY hr_rank LIMIT 5;
```

3 Overview of service

EDB BigAnimal is a fully managed database-as-a-service that runs in your cloud account or BigAnimal's cloud account, where it's operated by one of the builders of Postgres. BigAnimal makes it easy to set up, manage, and scale your databases. You can use PostgreSQL or EDB Postgres Advanced Server with Oracle compatibility, or you can choose distributed high-availability cluster types to use geographically distributed databases.

3.1 Supported cluster types

BigAnimal supports three cluster types:

- Single node
- Primary/standby high availability
- Distributed high availability

You choose the type of cluster you want on the Create Cluster page in the BigAnimal portal.

3.1.1 Single node

For nonproduction use cases where high availability isn't a primary concern, a cluster deployment with high availability not enabled provides one primary with no standby replicas for failover or read-only workloads.

In case of unrecoverable failure of the primary, a restore from a backup is required.

3.1.2 Primary/standby high availability

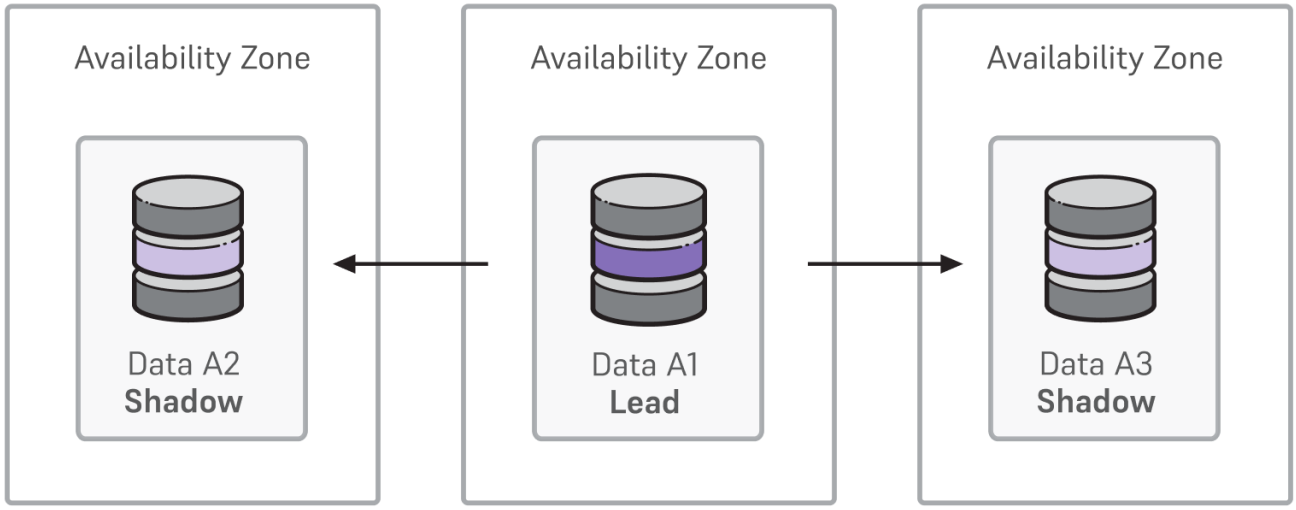
The Primary/Standby High Availability option is provided to minimize downtime in cases of failures. Primary/standby high-availability clusters—one *primary* and one or two *standby replicas*—are configured automatically. Standby replicas stay up to date through physical streaming replication.

If read-only workloads are enabled, then standby replicas serve the read-only workloads. In a two-node cluster, the single standby replica serves read-only workloads. In a three-node cluster, both standby replicas serve read-only workloads. The connections are made to the two standby replicas randomly and on a per-connection basis.

In cloud regions with availability zones, clusters are provisioned across zones to provide fault tolerance in the face of a data center failure.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

In case of temporary or permanent unavailability of the primary, a standby replica becomes the primary.



Incoming client connections are always routed to the current primary. In case of failure of the primary, a standby replica is promoted to primary, and new connections are routed to the new primary. When the old primary recovers, it rejoins the cluster as a standby replica.

Standby replicas

By default, replication is synchronous to one standby replica and asynchronous to the other. That is, one standby replica must confirm that a transaction record was written to disk before the client receives acknowledgment of a successful commit.

In a cluster with one primary and one replica (a two-node primary/standby high-availability cluster), you run the risk of the cluster being unavailable for writes because it doesn't have the same level of reliability as a three-node cluster. BigAnimal disables synchronous replication during maintenance operations of a two-node cluster to ensure write availability. You can also change from the default synchronous replication for a two-node cluster to asynchronous replication on a per-session or per-transaction basis.

In PostgreSQL terms, `synchronous_commit` is set to `on`, and `synchronous_standby_names` is set to `ANY 1 (replica-1, replica-2)`. You can modify this behavior on a per-transaction, per-session, per-user, or per-database basis using `SET` or `ALTER` commands.

To ensure write availability, BigAnimal disables synchronous replication during maintenance operations of a two-node cluster.

Since BigAnimal replicates to only one node synchronously, some standby replicas in three-node clusters might experience replication lag. Also, if you override the BigAnimal synchronous replication configuration, then the standby replicas are inconsistent.

3.1.3 Distributed high availability

Distributed high-availability clusters are powered by EDB Postgres Distributed. They use multi-master logical replication to deliver more advanced cluster management compared to a physical replication-based system. Distributed high-availability clusters let you deploy a cluster across multiple regions or a single region. For use cases where high availability across regions is a major concern, a cluster deployment with distributed high availability enabled can provide two data groups with a witness group in a third region.

This configuration provides a true active-active solution as each data group is configured to accept writes.

Distributed high-availability clusters support both EDB Postgres Advanced Server and EDB Postgres Extended Server database distributions.

Distributed high-availability clusters contain one or two data groups. Your data groups can contain either three data nodes or two data nodes and one witness node. At any given time, one of these data nodes is the leader and accepts writes, while the rest are referred to as shadow nodes. We recommend that you don't use two data nodes and one witness node in production unless you use asynchronous commit scopes.

PGD Proxy routes all application traffic to the leader node, which acts as the principal write target to reduce the potential for data conflicts. PGD Proxy leverages a distributed consensus model to determine availability of the data nodes in the cluster. On failure or unavailability of the leader, PGD Proxy elects a new leader and redirects application traffic. Together with the core capabilities of EDB Postgres Distributed, this mechanism of routing application traffic to the leader node enables fast failover and switchover.

The witness node/witness group doesn't host data but exists for management purposes. It supports operations that require a consensus, for example, in case of an availability zone failure.

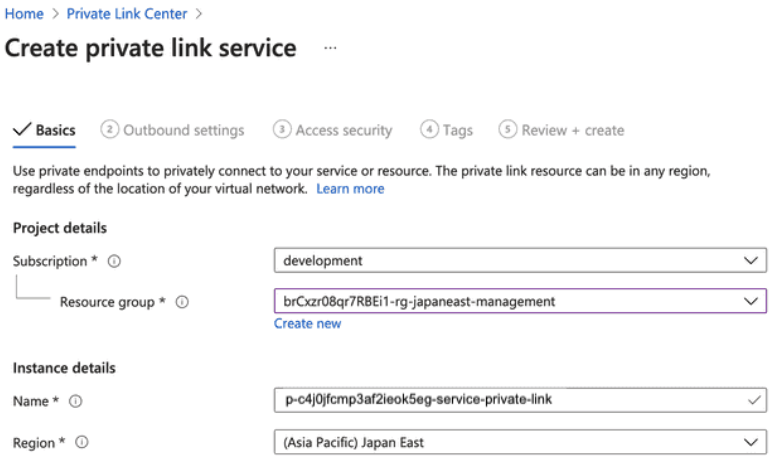
Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

> **Note**
>
> Operations against a distributed high-availability cluster leverage the EDB Postgres Distributed switchover feature, which provides subsecond interruptions during planned lifecycle operations.

Single data location

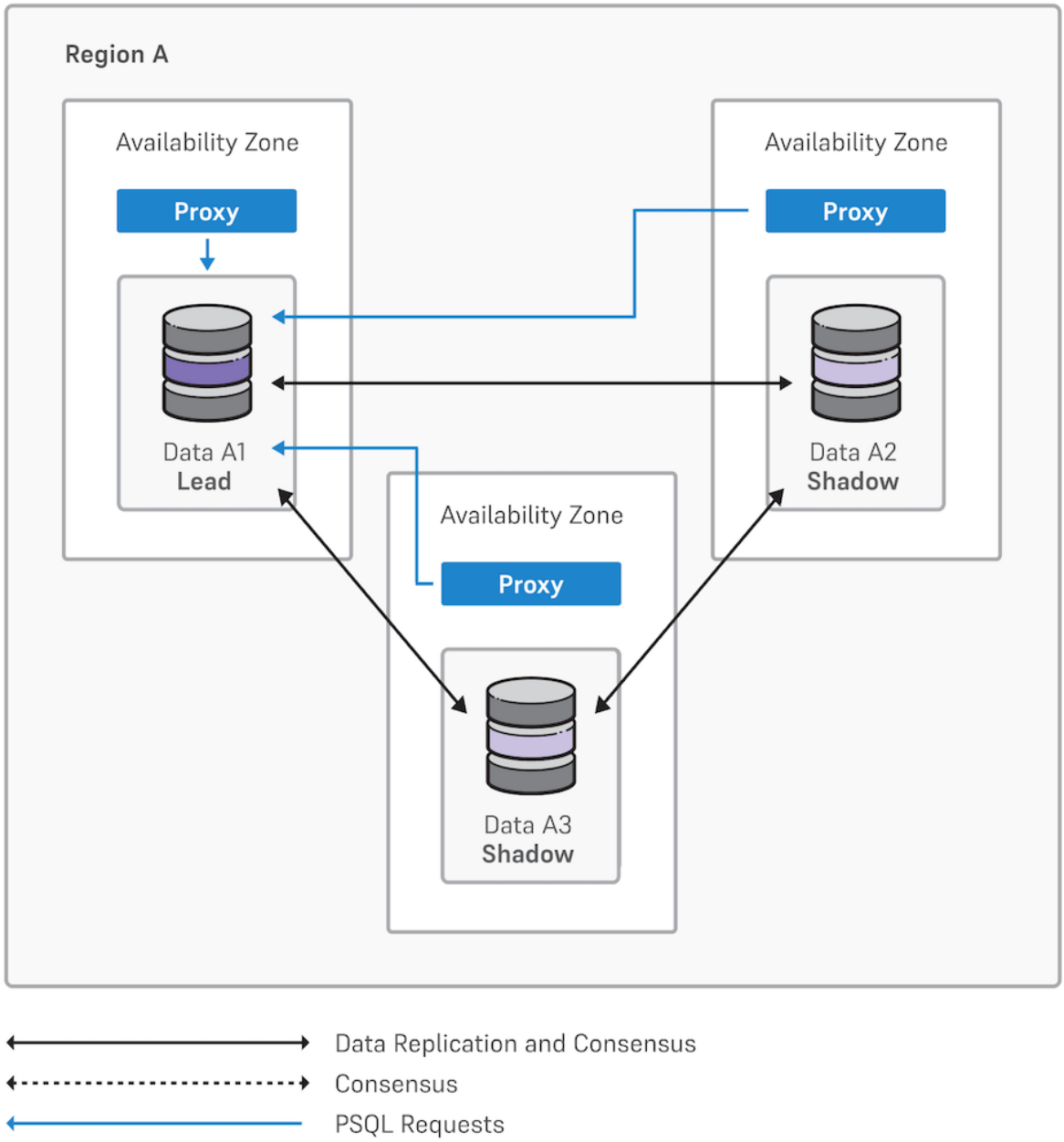
A configuration with a single data location has one data group and either:

- Two data nodes with one lead, one shadow, and a witness node, each in separate availability zones



- Three data nodes with one lead and two shadow nodes, each in separate availability zones

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

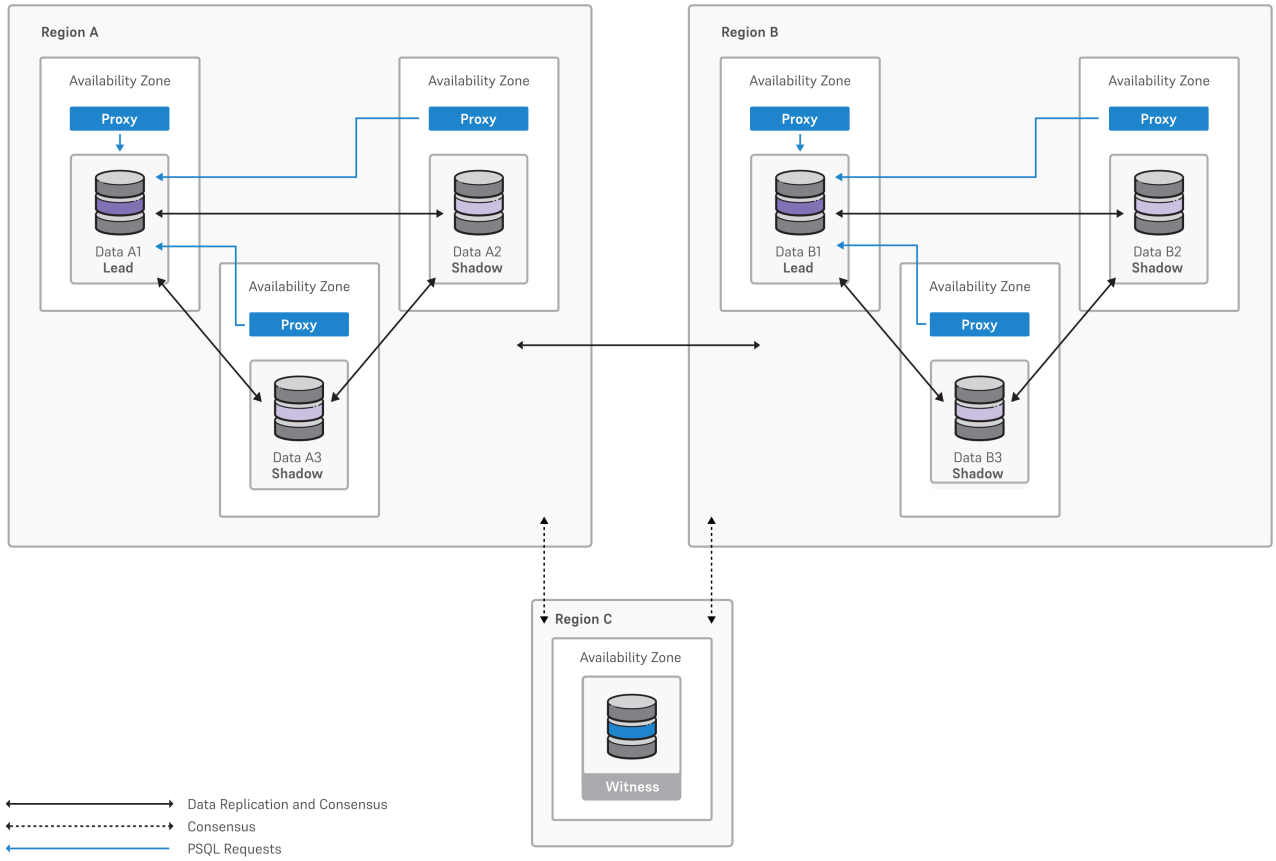


Multiple data locations and witness node

A configuration with multiple data locations has two data groups that contain either:

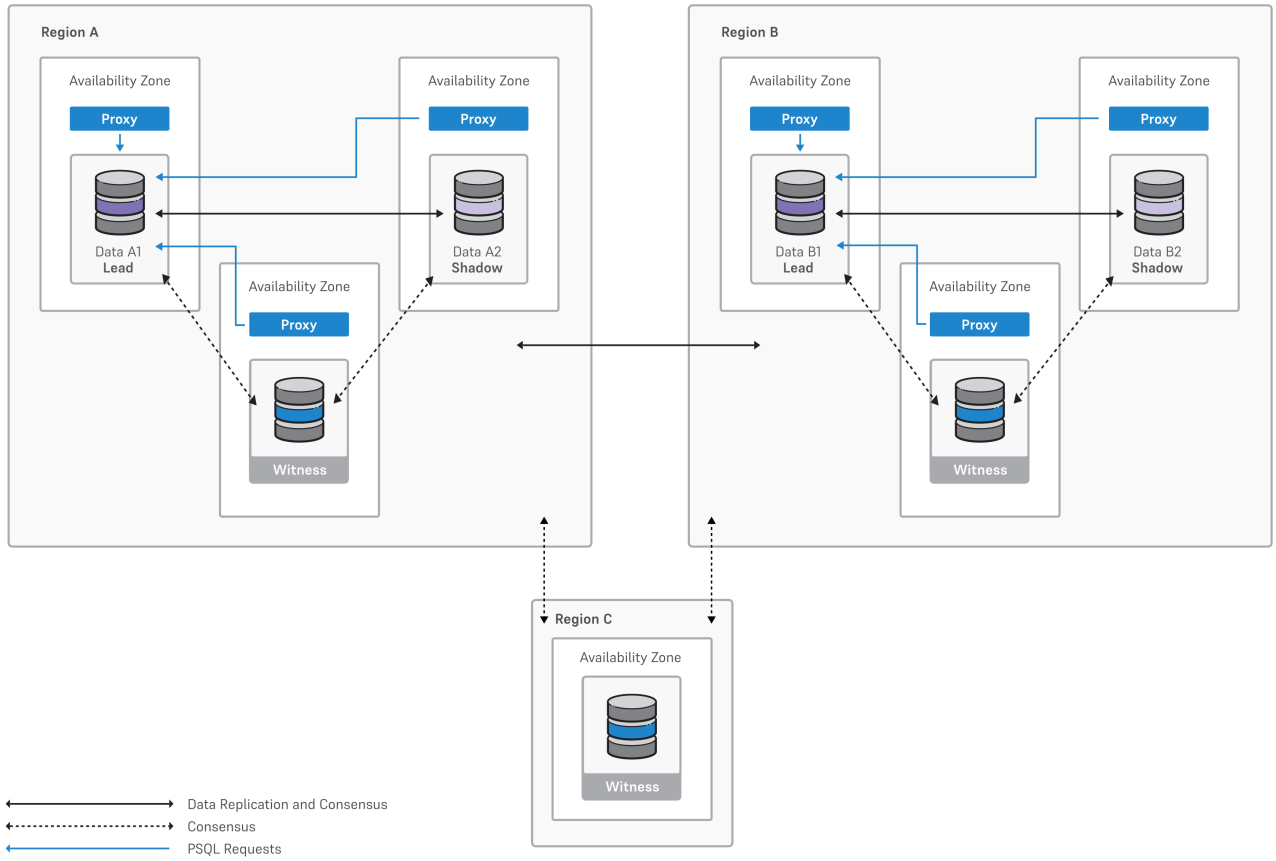
- Three data nodes:
    - A data node and two shadow nodes in one region
    - The same configuration in another region
    - A witness node in a third region

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



- Two data nodes (not recommended for production):
    - A data node, shadow node, and a witness node in one region
    - The same configuration in another region
    - A witness node in a third region

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



Cross-cloud service providers (CSP) witness node

By default, the cloud service provider selected for the data groups is preselected for the witness node.

To guard against cloud service provider failures, you can designate a witness node on a cloud service provider different from the one for data groups. This configuration can enable a three-region configuration even if a single cloud provider offers only two regions in the jurisdiction you're allowed to deploy your cluster in.

Cross-cloud service provider witness nodes are available with AWS, Azure, and Google Cloud using your own cloud account and BigAnimal's cloud account. This option is enabled by default and applies to both multi-region configurations available with PGD. For witness nodes you pay only for the infrastructure used, which is reflected in the pricing estimate.

For more information

For instructions on creating a distributed high-availability cluster using the BigAnimal portal, see Creating a distributed high-availability cluster.

For instructions on creating, retrieving information from, and managing a distributed high-availability cluster using the BigAnimal CLI, see Using the BigAnimal CLI.

3.2 Faraway replicas

Faraway replicas are read-only replicas of BigAnimal clusters that you can provision in most supported regions. You can create faraway replicas of your single-node and high-availability clusters in different regions of your cloud. Database users and applications can read from a nearby faraway replica instead of the source cluster. This ability relieves some of the workload on the source cluster and frees it up to handle write traffic.

In case of a failure, you can manually promote the faraway replica to a full-fledged BigAnimal cluster, making it capable of accepting writes. See Managing replicas.

In BigAnimal, faraway replicas use log shipping to replicate from their source clusters. This means that a replica cluster accesses the write-ahead log (WAL) of its source cluster and “replays” the changes described in it.

A WAL is a file that logs any changes made to a database. Databases write the change to the WAL before actually making the change. This way, if the database goes down before the change can be applied, the WAL has a record of the intended changes. The replica pulls in the changes from the WAL every time a new WAL file is closed, so replication is asynchronous.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Faraway replicas are priced the same as single-node clusters.

Advantages

- **Disaster recovery** — You can create faraway replicas in different regions in your cloud. Deploying faraway replicas across regions can help you build a more solid disaster recovery (DR) plan.
- **Improved read-query performance** — For an improved read-query performance, set up the faraway replica in the same region as your application. For example, applications can read the writes made to a cluster in the `us-east` region from a `us-west` replica.
- **Flexibility** — Options you might not find anywhere else:
    - Your choice of instance type and size
    - Your choice of storage volume and properties
    - No limit to the number of replicas you can create

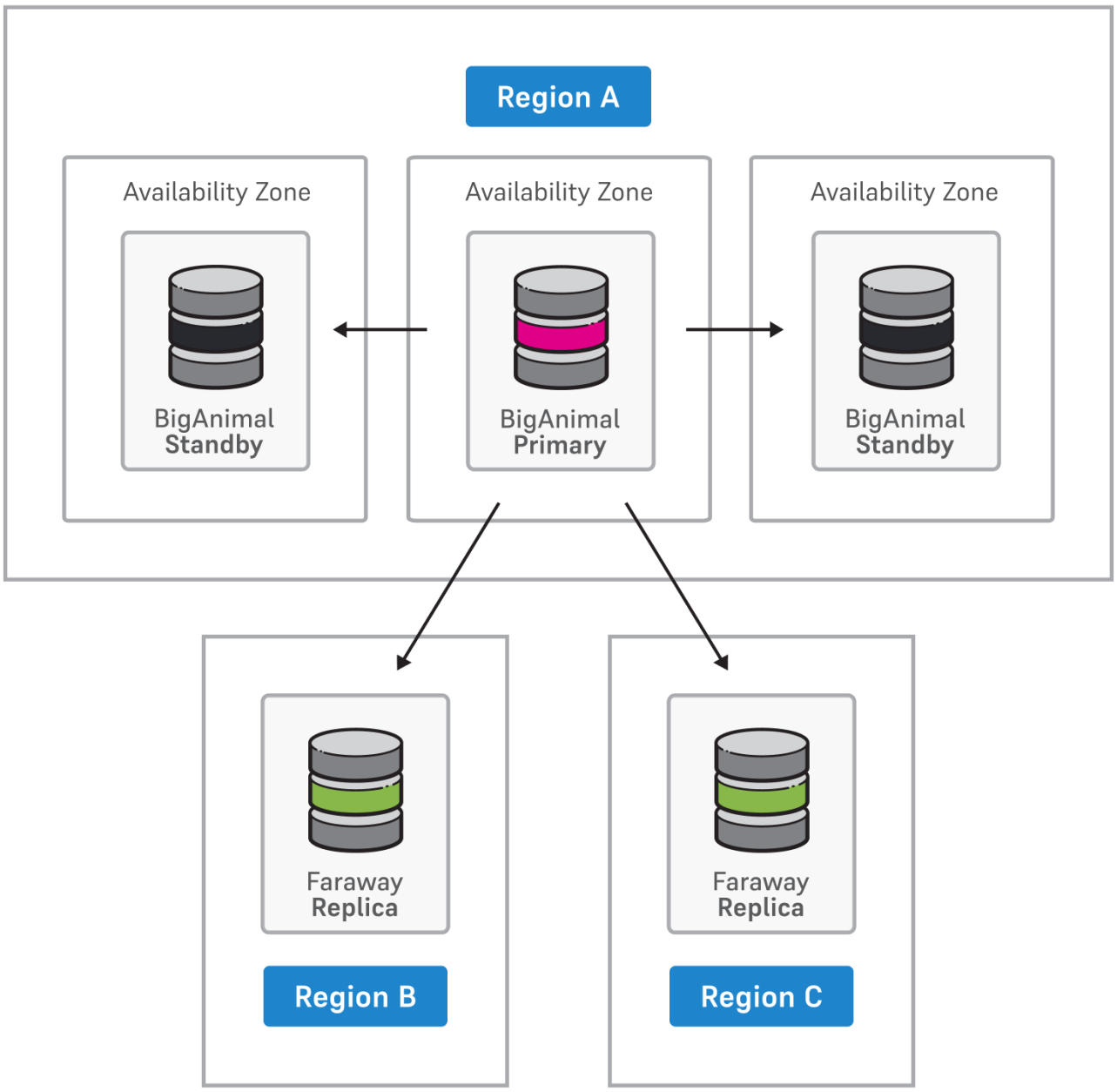
Limitations

- **Manual intervention** — Unlike standby replicas, BigAnimal doesn't automatically promote faraway replicas to the source cluster in case of a failure.
- **Replication lag** — Promoting a faraway replica to a BigAnimal cluster can result in loss of data. See Replication lag with faraway replicas.

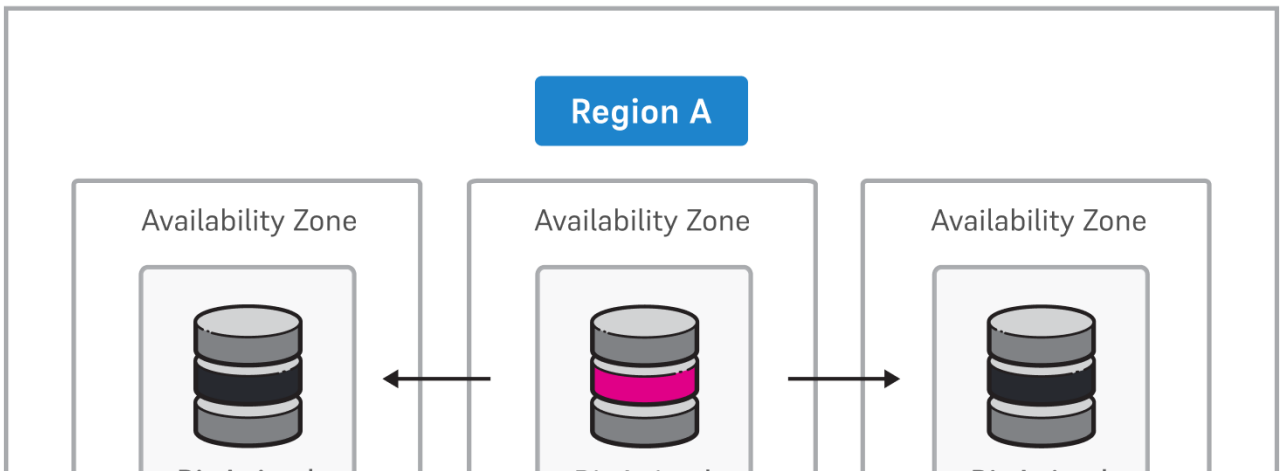
Examples

The diagram shows a three-node, high-availability cluster with two faraway replicas.

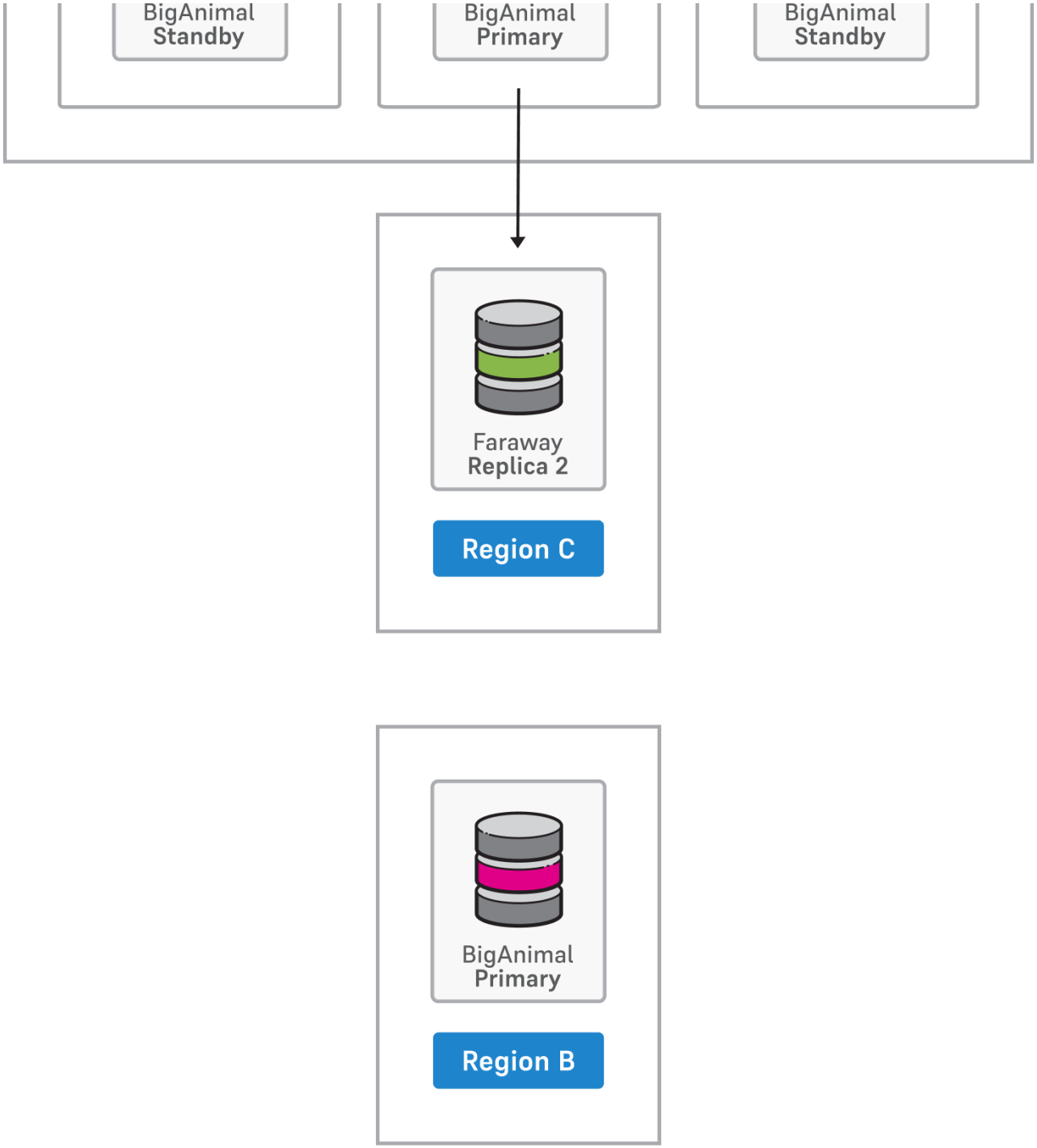
Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



The diagram shows a faraway replica in Region B that's promoted to a new cluster.



Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



Related topics

- Managing replicas
- Activating regions

3.3 Supported Postgres extensions and tools

BigAnimal supports a number of Postgres extensions and tools, which you can install on or alongside your cluster.

- See Postgres extensions available by deployment for the complete list of extensions BigAnimal supports if you're using your own cloud account.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- See Postgres extensions for the list of extensions supported when using BigAnimal's account and for more information on installing and working with extensions.

EDB Postgres extensions and tools

EDB develops and maintains several extensions and tools. These include:

- EDB Advanced Storage Pack — Provides advanced storage options for PostgreSQL databases in the form of table access method (TAM) extensions. These storage options can enhance the performance and reliability of databases without requiring application changes.

  The TAM extensions included in the Advanced Storage Pack include:

  - Autocluster — Provides faster access to clustered data by keeping track of the last inserted row for any value in a side table.
  - Refdata — Can provide performance gains of 5-10% and increased scalability.

- EDB Postgres Tuner — Provides safe recommendations that maximize the use of available resources.
- EDB Query Advisor — Provides index recommendations by keeping statistics on predicates found in WHERE statements, JOIN clauses, and workload queries.
- EDB Wait States — Probes each of the running sessions at regular intervals.
- PG Failover Slots — Is an extension released as open source software under the PostgreSQL License. If you have logical replication publications on Postgres databases that are also part of a streaming replication architecture, PG Failover Slots avoids the need for you to reseed your logical replication tables when a new standby gets promoted to primary.
- Foreign Data Wrappers — Allow you to connect your Postgres database server to external data sources.
- EDB PgBouncer — Allows you to manage your connections to your Postgres database.

3.4 EDB PgBouncer

EDB PgBouncer can manage your connections to Postgres databases and help your workloads run more efficiently. It's particularly useful if you plan to use more than a few hundred active connections. You can enable EDB PgBouncer to be entirely managed by BigAnimal, when creating your cluster. See Creating a cluster.

BigAnimal provisions up to three instances per EDB PgBouncer-enabled cluster to ensure that performance is unaffected, so each availability zone receives its own instance of EDB PgBouncer.

> **Note**
>
> Currently, you can't enable EDB PgBouncer when creating a distributed high-availability cluster.

If you want to deploy and manage PgBouncer outside of BigAnimal, see the How to configure EDB PgBouncer with BigAnimal cluster knowledge-base article.

3.5 Foreign data wrappers

BigAnimal supports EDB's MongoDB Foreign Data Wrapper and MySQL Foreign Data Wrapper. They allow you to connect your Postgres database server to external data sources. See:

- MongoDB Foreign Data Wrapper — Accesses data that resides on a MongoDB database from a Postgres database server.
- MySQL Foreign Data Wrapper — Accesses data that resides on a MySQL database from a Postgres database server.

3.6 Security

BigAnimal runs in your own cloud account or BigAnimal's cloud account. Every BigAnimal cluster is logically isolated from other BigAnimal clusters, but the security properties of the system are different in each deployment option. The key security features are:

- **Data isolation:** With both deployment options, data is fully isolated between separate clusters. No two BigAnimal clusters share a Postgres process, virtual machine, or storage volume. The implementation of this isolation depends on the deployment option.

  - **Your own cloud account:** Clusters are installed and managed on virtual machines and storage volumes deployed by BigAnimal on your behalf in your cloud environment. Complete segregation of your data is assured. Your data never leaves your cloud account, and your clusters don't share network segments with other customers' clusters.
  - **BigAnimal's cloud account:** BigAnimal deploys cloud infrastructure in accounts owned by BigAnimal. Every cluster is assigned a dedicated set of virtual machines and storage

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

volumes, and these resources are never reused by BigAnimal across multiple clusters. Two clusters can share the same network segment, but access to the system is limited to prevent communication between clusters in the BigAnimal infrastructure.

- **Granular access control:** With both deployment options, you can use single sign-on (SSO) and define your own sets of roles and role-based access control (RBAC) policies to manage your individual cloud environments. See Managing portal access for more information.

- **Data encryption:**

  - **BigAnimal's encryption** - All data in BigAnimal is encrypted in motion and at rest. Network traffic is encrypted using Transport Layer Security (TLS) v1.2 or greater. Data at rest is encrypted using AES with 256-bit keys. Data encryption keys are envelope encrypted, and the wrapped data encryption keys are securely stored in a key management system. When you use your own cloud account, encryption keys never leave your cloud environment.
  - **Your own encryption key** - Optionally enable Transparent Data Encryption (TDE) at the database level on BigAnimal's cloud account and on AWS. You can't enable TDE on existing clusters. You can enable TDE, only while creating a cluster. To enable TDE, use your own encryption keys. Connect the encryption keys to BigAnimal at the project level and select those keys to encrypt the databases. You can't disable TDE on TDE-enabled clusters.

> **Note**
>
> Enabling TDE using your own encryption key is supported on EDB Postgres Advanced Server versions 15 and later. The process of encryption and decryption adds additional overhead in terms of CPU and RAM consumption, performance, and for managing keys for faraway replicas.

- **Portal audit logging:** Activities in the portal, such as those related to user roles, organization updates, and cluster creation and deletion, are tracked and viewed in the activity log.

- **Database logging and auditing:** Functionality to track and analyze database activities is enabled automatically. For PostgreSQL, the PostgreSQL Audit Extension (pgAudit) is enabled for you when deploying a Postgres cluster. For EDB Postgres Advanced Server and EDB Postgres Extended Server, the EDB Audit extension (edb_audit) is enabled for you.

  - **pgAudit:** The classes of statements being logged for pgAudit are set globally on a cluster with `pgaudit.log = 'write,ddl'`. The following statements made on tables are logged by default when the cluster type is PostgreSQL: `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE`, AND `COPY`. All `DDL` is logged.

- **Database cluster permissions:** With both deployment options, managing database cluster permissions is your responsibility. The edb_admin user created during the cluster creation process is granted superuser-like permissions, including the CREATEDB and CREATEROLE database roles. We recommend using the edb_admin user to create a new application user and new application database for further isolation. See Managing Postgres access for more information.

See also

Security compliance and certifications

3.6.1 Security compliance and certifications

BigAnimal adheres to the following security standards and certifications:

SOC 2

Service Organization Controls (SOC) 2 is an auditing procedure that ensures service providers securely manage their customer data. Service providers can securely manage their customer data by protecting the interests of the customer's organization and the privacy of their clients. SOC 2 defines criteria for managing customer data based on up to five trust service principles: security, availability, processing integrity, confidentiality, and privacy.

SOC 2 reports are unique to each organization. In line with specific business practices, each designs its own controls to comply with one or more of the trust service principles. BigAnimal is assessed on security, availability, and confidentiality trust service principles.

A SOC 2 report arrives in two formats:

- **Type I**

  SOC 2 Type I classification describes a vendor's systems and whether their design is suitable to meet relevant trust principles. Type I focuses on the policies and procedures in place at a specific moment in time.

- **Type II**

  A SOC 2 Type II report assesses the effectiveness of security processes controls over time by observing operations for a minimum of six months. Like Type I, a Type II report is also an internal controls report capturing how a company safeguards customer data and how well those controls are operating.

GDPR

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

The General Data Protection Regulation (GDPR) is a regulation in European Union (EU) law on data protection and privacy, as well as in the European Economic Area (EEA). It also addresses the transfer of personal data outside the EU and EEA areas. The GDPR's primary aim is to enhance individuals' control and rights over their personal data and to simplify the regulatory environment for international business.

GDPR compliance implies both privacy and security mechanisms definition, enforcement, and control, including evidence collection. BigAnimal supports GDPR at service level, which means BigAnimal protects the personal data and privacy of EU citizens.

PCI DSS

The Payment Card Industry Data Security Standard (PCI DSS) is a set of security standards designed to ensure that companies that accept, process, store, or transmit credit card information maintain a secure environment. See the PCI Security Council website for more information about PCI.

BigAnimal is compliant with PCI. A PCI Qualified Security Assessor (QSA) certifies compliance. To achieve compliance with PCI DSS on a BigAnimal cluster, any information related to payments or other personally identifiable information (PII) must be encrypted, tokenized, or masked before being written to BigAnimal. You can do this from your customer application or through a third-party solution such as Satori.

Contact your EDB sales representative if you want to learn more about achieving PCI DSS compliance with your BigAnimal clusters.

3.7 Supported regions

Region support varies by whether you're using your cloud account or Biganimal's cloud account as your deployment option.

See Country and geographical region reference for information on geographical region short names and the countries that are in each geographical region.

Your cloud account

When using your cloud account, to successfully deploy a cluster, you must activate the region to prepare its cloud infrastructure. SeeActivating regions for more information.

> **Tip**
>
> You can select a cloud region for deploying your cluster even if it isn't yet active. However, your cluster creation request is added to a queue, and the cluster is created only after you activate the region.

Azure regions

When using Azure, you can create clusters in the following regions.

North America (NA)

Cloud region	Short name	Notes
Canada Central	canadacentral	
Central US	centralus	
East US	eastus	
East US 2	eastus2	
South Central US	southcentralus	Not supported for distributed high-availability clusters
West US 2	westus2	
West US 3	westus3	

Latin America (LATAM)

Cloud region	Short name	Notes
Brazil South	brazilsouth	Not supported for distributed high-availability clusters

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Europe, Middle East, and Africa (EMEA)

Cloud region	Short name	Notes
France Central	francecentral	Not supported for distributed high-availability clusters
Germany West Central	germanywestcentral	
North Europe	northeurope	
Norway East	norwayeast	Not supported for distributed high-availability clusters
UK South	uksouth	
West Europe	westeurope	Not supported for distributed high-availability clusters

Asia and Pacific (APAC)

Cloud region	Short name	Notes
Australia East	australiaeast	
Central India	centralindia	
Japan East	japaneast	
Southeast Asia	southeastasia	

AWS regions

When using AWS, you can create clusters in the following regions.

North America (NA)

Cloud region	Short name
Canada (Central)	ca-central-1
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (Oregon)	us-west-2

Europe, Middle East, and Africa (EMEA)

Cloud region	Short name
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Spain)	eu-south-2

Asia and Pacific (APAC)

Cloud region	Short name	Notes
Asia Pacific (Tokyo)	ap-northeast-1	
Asia Pacific (Seoul)	ap-northeast-2	
Asia Pacific (Mumbai)	ap-south-1	
Asia Pacific (Hyderabad)	ap-south-2	
Asia Pacific (Singapore)	ap-southeast-1	
Asia Pacific (Sydney)	ap-southeast-2	
Asia Pacific (Jakarta)	ap-southeast-3	Enable the region through the AWS console before activating it for BigAnimal.
Asia Pacific (Hong Kong)	ap-east-1	

Google Cloud regions

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

When using Google Cloud, you can create clusters in the following regions.

North America (NA)

Cloud region	Short name
Montreal	northamerica-northeast1
Toronto	northamerica-northeast2
Iowa	us-central1
South Carolina	us-east1
Northern Virginia	us-east4
Columbus	us-east5
Oregon	us-west1
Los Angeles	us-west2
Salt Lake City	us-west3
Las Vegas	us-west4
Dallas	us-south1

Latin America (LATAM)

Cloud region	Short name
Santiago	southamerica-west1
Sao Paulo	southamerica-east1

Europe, Middle East, and Africa (EMEA)

Cloud region	Short name
Warsaw	europe-central2
Finland	europe-north1
Madrid	europe-southwest1
Belgium	europe-west1
London	europe-west2
Frankfurt	europe-west3
Netherlands	europe-west4
Zurich	europe-west6
Milan	europe-west8
Paris	europe-west9
Tel Aviv	me-west1

Asia and Pacific (APAC)

Cloud region	Short name
Taiwan	asia-east1
Hong Kong	asia-east2
Tokyo	asia-northeast1
Osaka	asia-northeast2
Seoul	asia-northeast3
Mumbai	asia-south1
Delhi	asia-south2
Singapore	asia-southeast1
Jakarta	asia-southeast2
Sydney	australia-southeast1
Melbourne	australia-southeast2

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal's cloud account

Azure regions

When using Azure and BigAnimal's cloud account, you can create clusters in the following regions.

North America (NA)

Cloud region	Short name
US East (Virginia)	eastus2
Canada (Central)	canadacentral
East US 1	eastus1
West US 2	westus2

Asia and Pacific (APAC)

Cloud region	Short name
Asia Pacific (Pune)	centralindia

Europe, Middle East, and Africa (EMEA)

Cloud region	Short name
Europe (London)	uksouth

AWS regions

When using AWS and BigAnimal's cloud account, you can create clusters in the following regions.

North America (NA)

Cloud region	Short name
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (Oregon)	us-west-2

Asia and Pacific (APAC)

Cloud region	Short name
Asia Pacific (Mumbai)	ap-south-1

Europe, Middle East, and Africa (EMEA)

Cloud region	Short name
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2

Google Cloud regions

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

When using Google Cloud and BigAnimal's cloud account, you can create clusters in the following regions.

North America (NA)

Cloud region	Short name
Northern Virginia	us-east4

Europe, Middle East, and Africa (EMEA)

Cloud region	Short name
Belgium	europe-west1

Asia and Pacific (APAC)

Cloud region	Short name
Mumbai	asia-south1
Sydney	australia-southeast1

3.7.1 Country and geographical region reference

Geographical regions

Geographical region	Short name
North America	NA
Latin America	LATAM
Europe, Middle East, and Africa	EMEA
Asia and Pacific	APAC

Country by geographical region

Country	Region
Afghanistan	EMEA
Åland Islands	EMEA
Albania	EMEA
Algeria	EMEA
American Samoa	APAC
Andorra	EMEA
Angola	EMEA
Anguilla	LATAM
Antarctica	EMEA
Antigua and Barbuda	LATAM
Argentina	LATAM
Armenia	EMEA
Aruba	LATAM
Australia	APAC
Austria	EMEA
Azerbaijan	EMEA
Bahamas	LATAM
Bahrain	EMEA

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Country	Region
Bangladesh	APAC
Barbados	LATAM
Belarus	EMEA
Belgium	EMEA
Belize	LATAM
Benin	EMEA
Bermuda	LATAM
Bhutan	APAC
Bolivia, Plurinational State of	LATAM
Bosnia and Herzegovina	EMEA
Botswana	EMEA
Bouvet Island	EMEA
Brazil	LATAM
British Indian Ocean Territory	APAC
Brunei Darussalam	APAC
Bulgaria	EMEA
Burkina Faso	EMEA
Burundi	EMEA
Cabo Verde	EMEA
Cambodia	APAC
Cameroon	EMEA
Canada	NA
Cayman Islands	LATAM
Central African Republic	EMEA
Chad	EMEA
Chile	LATAM
China	APAC
Christmas Island	APAC
Cocos (Keeling) Islands	APAC
Collectivity of Saint Martin	EMEA
Colombia	LATAM
Comoros	EMEA
Congo	EMEA
Congo, the Democratic Republic of the	EMEA
Cook Islands	EMEA
Costa Rica	LATAM
Côte d'Ivoire	EMEA
Croatia	EMEA
Cuba	LATAM
Curaçao	EMEA
Cyprus	EMEA
Czechia	EMEA
Denmark	EMEA
Djibouti	EMEA
Dominica	LATAM
Dominican Republic	LATAM
Ecuador	LATAM
Egypt	EMEA
El Salvador	LATAM
Equatorial Guinea	EMEA
Eritrea	EMEA
Estonia	EMEA
Ethiopia	EMEA
Falkland Islands (Malvinas)	LATAM
Faroe Islands	EMEA
Fiji	APAC

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Country	Region
Finland	EMEA
France	EMEA
French Guiana	EMEA
French Polynesia	EMEA
French Southern Territories	EMEA
Gabon	EMEA
Gambia	EMEA
Georgia	EMEA
Germany	EMEA
Ghana	EMEA
Gibralta	EMEA
Greece	EMEA
Greenland	EMEA
Grenada	LATAM
Guadeloupe	EMEA
Guam	APAC
Guatemala	LATAM
Guernsey	EMEA
Guinea	EMEA
Guinea-Bissau	EMEA
Guyana	LATAM
Haiti	LATAM
Heard Island and McDonald Islands	EMEA
Holy See	EMEA
Honduras	LATAM
Hong Kong	APAC
Hungary	EMEA
Iceland	EMEA
India	APAC
Indonesia	APAC
Iran, Islamic Republic of	EMEA
Iraq	EMEA
Ireland	EMEA
Isle of Man	EMEA
Israel	EMEA
Italy	EMEA
Jamaica	LATAM
Japan	APAC
Jersey	EMEA
Jordan	EMEA
Kazakhstan	EMEA
Kenya	EMEA
Kiribati	APAC
Korea, Democratic People's Republic of	APAC
Korea, Republic of	APAC
Kuwait	EMEA
Kyrgyzstan	EMEA
Lao People's Democratic Republic	APAC
Latvia	EMEA
Lebanon	EMEA
Lesotho	EMEA
Liberia	EMEA
Libya	EMEA
Liechtenstein	EMEA
Lithuania	EMEA

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Country	Region
Luxembourg	EMEA
Macao	APAC
Macedonia, the former Yugoslav Republic of	EMEA
Madagascar	EMEA
Malawi	EMEA
Malaysia	APAC
Maldives	APAC
Mali	EMEA
Malta	EMEA
Marshall Islands	APAC
Martinique	EMEA
Mauritania	EMEA
Mauritius	EMEA
Mayotte	EMEA
Mexico	LATAM
Micronesia, Federated States of	APAC
Moldova, Republic of	EMEA
Monaco	EMEA
Mongolia	APAC
Montenegro	EMEA
Montserrat	LATAM
Morocco	EMEA
Mozambique	EMEA
Myanmar	APAC
Namibia	EMEA
Nauru	APAC
Nepal	APAC
Netherlands	EMEA
New Caledonia	EMEA
New Zealand	APAC
Nicaragua	LATAM
Niger	EMEA
Nigeria	EMEA
Niue	APAC
Norfolk Island	APAC
Northern Mariana Islands	APAC
Norway	EMEA
Oman	EMEA
Pakistan	APAC
Palau	APAC
Palestine, State of	EMEA
Panama	LATAM
Papua New Guinea	APAC
Paraguay	LATAM
Peru	LATAM
Philippines	APAC
Pitcairn	APAC
Poland	EMEA
Portugal	EMEA
Puerto Rico	LATAM
Qatar	EMEA
Réunion	EMEA
Romania	EMEA
Russian Federation	EMEA
Rwanda	EMEA

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Country	Region
Saint Barthélemy	EMEA
Saint Helena, Ascension and Tristan da Cunha	EMEA
Saint Kitts and Nevis	LATAM
Saint Lucia	LATAM
Saint Pierre and Miquelon	EMEA
Saint Vincent and the Grenadines	LATAM
Samoa	APAC
San Marino	EMEA
Sao Tome and Principe	EMEA
Saudi Arabia	EMEA
Senegal	EMEA
Serbia	EMEA
Seychelles	EMEA
Sierra Leone	EMEA
Singapore	APAC
Sint Maarten	EMEA
Slovakia	EMEA
Slovenia	EMEA
Solomon Islands	APAC
Somalia	EMEA
South Africa	EMEA
South Georgia and the South Sandwich Islands	LATAM
South Sudan	EMEA
Spain	EMEA
Sri Lanka	APAC
Sudan	EMEA
Suriname	LATAM
Swaziland	EMEA
Sweden	EMEA
Switzerland	EMEA
Syrian Arab Republic	EMEA
Taiwan, Province of China	APAC
Tajikistan	EMEA
Tanzania, United Republic of	EMEA
Thailand	APAC
Timor-Leste	APAC
Togo	EMEA
Tokelau	APAC
Tonga	APAC
Trinidad and Tobago	LATAM
Tunisia	EMEA
Turkey	EMEA
Turkmenistan	EMEA
Turks and Caicos Islands	LATAM
Tuvalu	APAC
Uganda	EMEA
Ukraine	EMEA
United Arab Emirates	EMEA
United Kingdom of Great Britain and Northern Island	EMEA
United States Minor Outlying Islands	APAC
United States of America	NA
Uruguay	LATAM
Uzbekistan	EMEA
Vanuatu	APAC
Venezuela, Bolivarian Republic of	LATAM

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Country	Region
Vietnam	APAC
Virgin Islands, British	LATAM
Virgin Islands, U.S.	LATAM
Wallis and Futuna	EMEA
Western Sahara	EMEA
Yemen	EMEA
Zambia	EMEA
Zimbabwe	EMEA

3.8 Shared responsibilities

Responsibility for security in BigAnimal is shared between you and EDB. EDB provides a secure platform that enables you to create and maintain secure database clusters deployed on BigAnimal. You have several responsibilities around the security of your clusters and the data they contain. These responsibilities are the same whether you use your cloud or BigAnimal's cloud account as your deployment option except where noted.

The following responsibility model describes the distribution of specific responsibilities between you and EDB.

High availability

- You are responsible for choosing whether to enable high availability.
- EDB is responsible for properly configuring and maintaining replication between database nodes.
- If you choose to use asynchronous replication (not recommended), you are responsible for managing replication lag between database nodes.
- EDB is responsible for deploying database nodes across availability zones, where available.
- You are responsible for ensuring your applications reconnect when network connectivity is interrupted.

Database performance

- EDB is responsible for deploying clusters with the infrastructure you choose and managing and monitoring these infrastructure resources.
- You are responsible for data modeling, query design, and scaling the cluster to meet your performance needs.

Deploying and scaling

- EDB is responsible for deploying, managing, and monitoring the underlying infrastructure supporting your clusters.
- You are responsible for choosing the appropriate configuration for your workload, including instance type, storage, and configuration.
- If you're using your cloud account, you are responsible for managing your cloud resource limits to ensure the underlying infrastructure can be provisioned.

Backups and restores

- EDB is responsible for taking backups and archiving transaction logs and storing them in object storage instances.
- You are responsible for the charges associated with the cloud object storage solution. If you're using BigAnimal's cloud account, these charges are passed along to you in your monthly rates.
- You are responsible for periodically restoring and verifying the restores to ensure that archives can meet your recovery time and recovery point objectives.

Encryption

- EDB is responsible for data encryption at rest for both backups and live data.
- EDB is responsible for data encryption in transit for both intra-cluster traffic and traffic between clusters and backup storage.
- You are responsible for data encryption in transit between your applications and your cluster. BigAnimal clusters support, but don't require, `verify-full` TLS connections.
- You are responsible for application-level encryption to protect particularly sensitive data from unauthorized access by your authorized users and applications.

Credential management

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- EDB is responsible for securely managing your edb_admin credential. The edb_admin credential is never stored in plaintext.
- You are responsible for managing and securing your cluster users and their passwords.

3.9 Database version policy

We support the major Postgres versions from the date they're made available until the version is retired by EDB (generally five years). See Platform Compatibility for more details on support dates for PostgreSQL, EDB Postgres Advanced Server, and EDB Postgres Extended Server. See End-of-life policy for details on our policy for retiring deprecated versions.

Supported Postgres types and versions

Postgres distribution	Versions
PostgreSQL	12-16 for (single, high-availability) clusters
EDB Postgres Extended Server	12-16 for (single, high-availability), 14-16 for (distributed high-availability) clusters
EDB Postgres Advanced Server	14-16 for all cluster types (single, high-availability, distributed high-availability)

End-of-life policy

BigAnimal deprecates support for Postgres versions following the same timeline as PostgreSQL. PostgreSQL, EDB Postgres Advanced Server, and EDB Postgres Extended Server follow the same timelines. We recommend that you take action and upgrade your Postgres databases running on the deprecated version to a later version as soon as possible.

Six months before the PostgreSQL deprecation date, BigAnimal doesn't allow you to create new instances with the deprecated database version.

Six months after the PostgreSQL deprecation date, BigAnimal forces a major version upgrade to all clusters running the deprecated database version.

The only exception is customers who purchased Extended Life Support (ELS) prior to that date. To purchase Extended Life Support, contact EDB Sales.

Key dates

While PostgreSQL officially deprecated version 11 on November 9, 2023, BigAnimal deprecated PostgreSQL 11 on November 20, 2023 in alignment with the broader EDB portfolio.

On November 20, 2023, BigAnimal deprecated support for PostgreSQL 11 and EDB Postgres Advanced Server 11 using the following schedule. We recommend that you take action and upgrade your Postgres databases running on major version 11 to a later version, such as PostgreSQL version 15.

BigAnimal automatically upgrades your Postgres 11 instances to a later version at the end of six months. Customers who purchased Extended Life Support are exempt. To purchase Extended Life Support, contact Sales. BigAnimal reserves the right to upgrade sooner in case of a security issue.

Action or recommendation	Dates - Postgres 11	Dates - Postgres 12
BigAnimal no longer supports the Postgres version.	November 20, 2023	December 9, 2024
The PostgreSQL community plans to deprecate the version and won't provide any security patches after this date.	November 9, 2023	November 14, 2024
Start upgrading your BigAnimal instances to a later major version starting on this date. You can use pg_dump to restore your cluster into a later version.	Until May 20, 2023	Until June 9, 2024
You can't create a new instance with Postgres version 11 in the BigAnimal interfaces after this date.	May 20, 2023	June 9, 2024
If you manually restore a cluster, you can't restore it with Postgres version 11 after this date. You need to restore that Postgres cluster into a newer version of Postgres.	May 20, 2023	June 9, 2024
BigAnimal will force a major version upgrade and won't allow any new clusters to be provisioned or restored to the old Postgres version. Only customers on Extended Life Support are exempt for the duration of their contract. If you don't purchase Extended Life Support on this date, your clusters will be upgraded.	April 20, 2024	May 9, 2025
Extended Life Support starts six months after PostgreSQL community deprecates the Postgres version. BigAnimal reserves the right to upgrade in case of a security issue.	April 20, 2024	May 9, 2025

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

3.10 Periodic maintenance

EDB performs periodic maintenance to ensure stability and security of your clusters. We perform minor version upgrades and patch updates as part of this periodic maintenance.

Notification of upcoming maintenance

You're notified in the BigAnimal portal before maintenance occurs. Details are available on the BigAnimal status page. You can subscribe to get these updates in a feed by selecting **Subscribe to Updates** on the status page. You can also enable the notifications to receive in-app or email notifications for upcoming, successful, and failed cluster maintenance upgrade. For more information, see managing notifications.

EDB reserves the right to upgrade customers to the latest minor version without prior notice in an extraordinary circumstance. You can't configure minor versions.

In some cases, these updates might terminate existing network connections to your clusters. If that happens, the outage is typically less than 30 seconds. Be sure your applications are configured to automatically reconnect when connections are interrupted. Most modern database libraries do this by default.

Specifying maintenance windows

If you want to control when the updates are pushed, you can specify a weekly maintenance window for each cluster or each data group in the case of a distributed high-availability cluster. BigAnimal displays a *scheduled maintenance* message on your cluster list four hours prior to the scheduled maintenance time to remind you of the upcoming maintenance window. This reminder allows you to make any necessary preparations, such as saving your work and closing any open connections. For more information on specifying maintenance windows, see Maintenance.

Maintenance for high-availability clusters

For primary/standby high-availability clusters, periodic maintenance is performed first on the standby replicas and then on the primary.

While there is no downtime during periodic maintenance, there will be a network connection reset as the primary is failing over.

Connectivity issues after an automatic upgrade

Most connectivity issues correct themselves when you reopen the connection after waiting for a minimum of five seconds. We recommend that you:

- Wait five seconds before your first attempt.
- For the next attempt, increase the wait by doubling the previous wait time. Keep trying this approach until you reach a maximum wait time of 60 seconds.

We also recommend that you set a maximum number of attempts to reopen the connection before your application reports that it can't reconnect.

When an active connection that's currently executing a command is interrupted, you might need to take extra action when reopening the connection. (For read-only transactions that were in progress, you can reopen the connection without any extra steps.) For a transaction that was writing to the database, you need to know whether the transaction was rolled back or whether it succeeded to determine whether you need to retry the transaction. If it was rolled back, you need to retry it. If it succeeded, you don't need to retry it. It's possible for a transaction to succeed without sending you the commit acknowledgment from the database server, so you'll need to add some logic to be sure.

Test your retry logic by creating an event that causes a brief downtime to see if it's handling these transactions correctly.

3.11 BigAnimal support options

If you have problems with BigAnimal, you have several options to get help from EDB's Support team. If you have a BigAnimal account, you can go directly to the Support portal or the BigAnimal portal to open a Support case, or you can leave Support a message using the Support Case widget.

If you can’t log in to your account or are having issues using the widget, send us an email atcloudsupport@enterprisedb.com.

See Managing your Support cases for more information.

Create a Support case from the Support portal (recommended)

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

1. Start a Support case using any one of these options:
    - Go to the Support portal and select **Submit a request** at the top-right of the page.
    - Log in to BigAnimal. Select the question mark (?) at the top right of the page, select **Support Portal**, and select **Submit a request** at the top right of the page.
    - Log in to BigAnimal. Select the question mark (?) at the top right of the page, and select **Create Support ticket**.
2. Enter a description in the **Subject** field.
3. (Optional) Select the cluster name from the **Cluster name** list.
4. Enter values for **Severity Level** and **Description**.
5. (Optional) Attach files to provide more details about the issue.
6. Select **Submit**.

Create a Support case from the Support widget

1. Log in to BigAnimal and select **Support** on the bottom of the left navigation pane.
2. Fill in the **Leave us a message** form.
    1. (Optional) The **Your Name** field is prefilled, but you can edit it.
    2. The **Email Address** field is prefilled, but you can edit it.
    3. (Optional) Select the cluster name from the **Cluster name** list.
    4. Enter values for **Severity Level** and **Description**.
    5. (Optional) Attach files to provide more details about the issue.
    6. Select **Submit**.

Case severity level

Level	Description
Severity 1	The cloud service is down, or there's an error with critical impact on your production environment. Covers urgent problems including database service outage, data loss, and cluster provisioning failure.
Severity 2	There's a cloud service error or an issue significantly impacting your production environment. The system is functioning but in a severely reduced capacity. Covers problems including database service interruption and backup failures.
Severity 3	There's an error or issue that doesn't have a significant impact on your production environment. Covers problems including API issues, monitoring metrics, and logging issues.
Severity 4	General questions, inquiries, or nontechnical requests.

For more information

- BigAnimal Terms
- Service Level Agreement (SLA) - BigAnimal
- SLO Support Terms - BigAnimal

3.11.1 Managing your Support cases

Viewing your Support cases

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

After you create your Support case, you can view the case and the case history and monitor the status of your case at theMy activities page in the Support portal. Select your image in the top-right corner of the Support portal, and then select **My Activities** from the menu.

If your login account has your email address, or if you submit the case by email or the Support web widget, you receive the case information by email. The case request number displays in the email subject line.

A case can have one of the following statuses:

- **Open:** A new case or reopened case.
- **Awaiting your reply:** BigAnimal Support has responded to your case and is asking for more information or letting you know that the case is resolved and asking for you to confirm.
- **Resolved:** The case is resolved.

Updating your Support cases

You can update your case at the bottom of the request page in **Add to conversation**. You can view BigAnimal Support's case updates at the request page.

If your login account has your email address, or you submit the case by email or the Support web widget, you can update the case or receive the case updates from Support by email.

Searching for Support cases

From the **My requests** list at the My activities page in the Support portal, you can search for all of your Support cases using the search bar. You can also filter the cases by status.

Reopening Support cases

You can reopen the case after it's set to **Solved** by responding with a new comment. You can reopen your Support case up to seven calendar days after case resolution.

If your login account has your email address, or you submit the case by email or Support web widget, you can reopen the case by replying to the last email of the case prior to the survey email.

You can't reopen Support cases that are in the resolved state for more than seven calendar days. However, you can create a follow-up from the original case. To create a follow-up, select the link at the bottom of the case. The follow-up creates a new case linked to the original case.

Resolving a Support case

You can resolve the case by adding a new comment and selecting **Mark as solved**. Our Support team changes the status of the case to Resolved after receiving your confirmation.

Case survey

The BigAnimal Support team asks that you provide feedback on your experience by completing the survey on the request page. The survey is available one day after the case is resolved. The survey remains open for six days.

If your login account has your email address, or you submit the case by email or Support web widget, you receive the survey email. You can select the survey link in the email to submit a survey.

Our Support team reviews the survey results to improve the support experience and appreciates your feedback.

4 Choosing your deployment option

You have two deployment options with BigAnimal. You can deploy clusters in your cloud account, or you can deploy in BigAnimal's cloud account.

To learn about the pricing of these two deployment options, see Pricing and billing.

Using your cloud account

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

When deploying in your cloud account, you need to set up your cloud service provider. You can use your AWS, Azure, or Google Cloud account. In this deployment model, you are responsible for managing your cloud resources and connecting your cloud to BigAnimal. To set up your cloud service provider, see Checking your cloud readiness.

Using BigAnimal's cloud account

BigAnimal's cloud account offers a seamless deployment option if you don't want to set up a separate cloud account for your clusters. You can deploy a cluster in BigAnimal's cloud account instantly.

> **Note**
>
> Currently, when you deploy in BigAnimal's cloud account, you can use AWS or Google Cloud as your cloud provider. If you want to use your cloud account's Identity and Access Management (IAM) service for database authentication, you must deploy clusters in your own cloud account. If you deploy using BigAnimal's cloud account, the underlying cloud provider is hidden, so you can't manage IAM access.

4.1 Deploying with AWS

To use AWS as your cloud account:

- Sign in for the first time with your EDB account, and then either use the BigAnimal portal as your identity provider orset up your own provider afterward.
- Check the readiness of your AWS account before deploying.
- Connect your AWS cloud to BigAnimal.
- Connect to BigAnimal using AWS VPC endpoint service.
- Review the supported AWS regions where you can create clusters for the AWS account, enabling you to locate your cloud resources close to your customers.
- View metrics and logs sent from BigAnimal with AWS CloudWatch.

4.2 Deploying with Google Cloud

To use Google Cloud as your cloud account:

- Sign in for the first time with your EDB account, and then either use the BigAnimal portal as your identity provider orset up your own provider afterward.
- Check the readiness of your Google Cloud account before deploying.
- Connect your Google Cloud account to BigAnimal using Google Cloud Shell.
- Connect your cluster from Google Cloud to BigAnimal using Google Cloud's Private Service Connect.
- Review the supported Google Cloud regions where you can create clusters for the Google Cloud account, enabling you to locate your cloud resources close to your customers.

4.3 Deploying with Azure

To use Azure as your cloud account:

- Sign in for the first time with your EDB account, and then either use the BigAnimal portal as your identity provider orset up your own provider afterward.
- Connect your Azure Marketplace account to BigAnimal.
- Check the readiness of your Azure subscription before deploying.
- Connect your Azure cloud to BigAnimal.
- Connect to BigAnimal from your application's virtual network in Azure.
- Review the supported Azure regions where you can create clusters for the Azure account, enabling you to locate your cloud resources close to your customers.
- View metrics and logs sent from BigAnimal in your Azure account.
- Configure Azure to trigger real-time alerts based on the metrics in BigAnimal.

4.4 Deploying with BigAnimal's cloud

To deploy clusters using BigAnimal's cloud account:

- Choose what Postgres distribution available on the BigAnimal cloud best meets your needs.
- Review the supported regions in which you can create clusters for BigAnimal's cloud account, enabling you to locate your cloud resources close to your customers, both internal or external.
- Evaluate how BigAnimal charges you for cloud infrastructure costs.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

5 Choosing your cluster type

Choose the cluster type and configuration that meets your availability requirements.

Select the cluster type names in the table heading for more information on each type and configuration.

Consideration	Single node	Primary/Standby HA	HA + standby replica	Distributed HA single region	Distributed HA multi-region
Data replication	None	Physical	Physical	Logical	Logical
Region	Single	Single	Multi	Single	Multi
VM failure tolerance					
AZ failure tolerance	TBD				
Region failure tolerance	TBD	TBD		TBD	
Recovery time objective	varies	35s-60s	varies	0	0
Recovery point objective	<5 min	0	<5 min	0	30s (configurable)
Service level agreement	99.5%	99.99%	99.99%	99.99%	99.995%

6 Choosing your Postgres distribution

Postgres distribution and version support varies by cluster and deployment type.

Postgres distribution using your cloud account

Postgres distribution	Versions	Cluster type
PostgreSQL	11–15	Single-node, primary/standby high-availability
EDB Postgres Advanced Server	12–15	Single-node, primary/standby high-availability
EDB Postgres Extended Server	14–15	Distributed high-availability
EDB Postgres Advanced Server	14–15	Distributed high-availability

Postgres distribution using the BigAnimal cloud account

Postgres distribution	Versions	Cluster type
PostgreSQL	12–15	Single-node, primary/standby high-availability
EDB Postgres Advanced Server	14–15	Single node, primary/standby high availability
EDB Postgres Extended Server	14–15	Distributed high-availability
EDB Postgres Advanced Server	14–15	Distributed high-availability

5 Setting up your identity provider

After signing in for the first time with your EDB account, you can either use the BigAnimal portal as your identity provider or set up your own.

> **Note**
>
> If you purchased through Azure Marketplace, BigAnimal authenticates users using Azure Active Directory (AD) and you don't have to complete these steps. Azure AD is linked during subscription. Also, you can still invite users that have an EDB account through the BigAnimal portal.

When using your own identity provider, you add users to BigAnimal by adding them to the designated group in your identity provider. Once you've logged into BigAnimal using your own identity provider, you can set up your cloud service provider in the BigAnimal portal to complete onboarding.

If you're using the BigAnimal portal as your identity provider, you can also invite users that have an EDB account by selecting**Invite New User** on the Users page. After providing their EDB

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

account email and their role, you can send them an invitation link. Ensure the user accepts the invitation within 48 hours, or you'll have to send a new invitation.

For more information on roles, see Managing portal access.

Setting up your own identity provider

BigAnimal supports single sign-on through SAML identity providers. The SAML application enables access to BigAnimal for groups selected in your identity provider. You configure your SAML application to send a SAML assertion response to BigAnimal for each user.

The identity provider application provides:

- Signature certificate
- Single sign-on URL (Sign In Endpoint)

The BigAnimal service provider provides:

- ACS URL (Assertion Consumer Service)
- SP Entity ID (Audience URI)

The Set Up Identity Provider page in BigAnimal provides the ACS URL and Audience URI, which you can copy to use when configuring your SAML application. The page also includes information about mandatory attributes for the configuration.

You need the following SAML assertions to map the user information in your identity provider application to BigAnimal:

```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

Configure SAML

On the Set Up Identity Provider page:

1. In the **Connection Info** section, copy the following URLs to use in your identity provider configuration:

URL	Description
Assertion Consumer Service URL	BigAnimal-specific URL to which SAML assertions from your identity provider are sent
Audience URI	The entity or audience for which the SAML assertion is intended

2. In the **Configure SAML response** section, configure your identity provider to send the SAML response. The SAML response must include the following attributes:

   - `givenname` — BigAnimal uses the value as the given name in the profile of the authenticated user.
   - `surname` — BigAnimal uses the value as the surname in the profile of the authenticated user.
   - `name` — BigAnimal uses the value as the full name (`givenname` joined with `surname`) in the profile of the authenticated user.
   - `emailaddress` — BigAnimal uses the value for the email address in the profile of the authenticated user.

   Provide a `NameID` element in the `Subject` element. Provide your email ID as the value to `NameID`. BigAnimal uses the email ID you provide as your username and primary email, so format `NameID` like an email address.

   For example:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

3. In the **SAML settings** section, enter the configuration information for your preferred SAML identity provider:

Field	Description
Single Sign-On URL	The identity provider's sign-on URL.
Identity Provider Signature Certificate	Identity provider's assertion signing certificate (`.cer` or `.cert`). Coordinate with your identity provider partner to obtain this certificate securely.
Request Binding	SAML Authentication Request Protocol binding used to send the authentication request: HTTP-Redirect, HTTP-Post, or Hybrid (SAML request is REDIRECT and response is POST).
Response Signature Algorithm (RSA) Algorithm	The signature algorithm used to sign the SAML AuthNRequest (RSA SHA-1 or RSA SHA-256).

4. Select **Test Connection**. If you connect to your identity provider successfully, your identity provider's login screen appears. If an error message appears, contactSupport.

Once your identity provider is set up, you can view your connection status, ID, login URL, audience URI, and assertion consumer service URL from the BigAnimal portal on the Identity Provider page. Select **Admin > Identity Provider** to access it.

> **Further reading**
>
> Setting up specific identity providers

Add a domain

You need a verified domain so your users can have a streamlined login experience with their email address.

1. On the **Domains** tab, enter the domain name and select **Next: Verify Domain**.

2. Copy the TXT record and follow the instructions in the on screen verify box (repeated below), to add it as a TXT record on that domain within your DNS provider's management console.

    - Log in to your domain registrar or web host account.
    - Navigate to the DNS settings for the domain you want to verify.
    - Add a TXT record.
    - In the Name field, enter @.
    - In the Value field, enter the verification string provided, eg.
    - “edb-biganimal-verification=VjpcxtIC57DujkKMtECSwo67FyfCExku”
    - Save your changes and wait for the DNS propagation period to complete. This can take up to 48 hours.

3. Select **Done**.

    Your domain and its status appear on the **Domains** tab, where you can delete or verify it. Domains can take up to 48 hours for the change of the domain record by the DNS provider to propagate before you can verify it.

4. If your domain has not verified after a day, you can debug whether your domain has the matching verification text field. Select **Verify** next to the domain at `/settings/domains` to check the exact value of the required TXT field. Query your domain directly with DNS tools, such as nslookup, to check if you have an exact match for a text = "verification" field. Domains can have many TXT fields. As long as one matches, it should verify.

```
> nslookup -type=TXT mydomain.com

;; Truncated, retrying in TCP mode.
Server:     192.168.1.1
Address:    192.168.1.1#53
Non-authoritative answer:
...
mydomain.com    text = “edb-biganimal-verification=VjpcxtIC57DujkKMtECSwo67FyfCExku”
```

To add another domain, select **Add Domain**.

When you have at least one verified domain (with Status = Verified, in green), the identity provider status becomes **Active** on the **Identity Providers** tab. When the domain is no longer verified, the status becomes **Inactive**.

> **Note**
>
> Your DNS provider can take up to 48 hours to update. Once the domain is verified, the identity provider status can take up to three minutes to update.

Domain expiry

The EDB system has a 10-day expiry set for checking whether domains are verified.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

You buy domains from DNS providers by way of a leasing system. If the lease expires, you no longer own the domain, and it disappears from the Internet. If this happens, you need to renew your domain with your DNS provider.

Whether the domain failed to verify within the 10 days or it expired months later, it appears as **Status = Expired** (in red). You can't reinstate an expired domain because expiry means you might no longer own the domain. You need to verify it again.

To delete the domain, select the bin icon. To re-create it, select **Add Domain**. Set a new verification key for the domain and update the TXT record for it in your DNS provider's management console, as described in Add a doman.

Manage roles for added users

You add users through your identity provider. A user who you add in the identity provider is automatically added to BigAnimal. BigAnimal assigns them with the default role of organization member. You manage roles and permissions from BigAnimal. See Managing portal access.

> **Note**
>
> A user is created in BigAnimal only after they log in. After they log in, you can change their BigAnimal role.

Add a tile

Once you establish the identity provider, you can create a BigAnimal tile for users to access the organization's BigAnimal application. To do so, copy the quick sign-in URL from the **Settings > Identity Provider** page of the BigAnimal portal. For details on how to add a tile, refer to your identify provider documentation for instructions on setting up SSO access to your application.

Next steps

You and other users can log in to BigAnimal using your identity provider credentials.

You can rename the default project BigAnimal creates for you. See Editing a project.

You can set up your cloud service provider so that you or other users with the correct permissions can create clusters.

You can assign roles to your default project. See Managing portal access.

5.1 Setting up specific identity providers

Step-by-step instructions for setting up specific identity providers are available in the BigAnimal knowledge base. See:

- Using Auth0 as your identity provider
- Using AWS IAM Identity Center as your identity provider
- Using Azure AD as your identity provider
- Using Google Workspace (G Suite) as your identity provider
- Using Okta as your identity provider

6 Setting up your Azure Marketplace account

Connect your cloud account with your Azure subscription.

Before starting, in Azure Active Directory, ensure your user type is Member (not Guest).

1. Select the EDB offer in the Azure portal.

1. Sign in to the Azure portal and go to Azure **Marketplace**.
2. Find an offer from **EnterpriseDB Corporation** and select it.
3. From the **Select Plan** list, select an available plan.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

4. Select **Set up + subscribe**.

2. Fill out the details for your plan.

1. In the **Project details** section, enter or create a resource group for your subscription. See What is a resource group for more information.
2. In the **SaaS details** section, enter the SaaS subscription name.
3. Select **Review + subscribe**.

3. Accept terms of use.

1. Review the terms of use provided by EDB.
2. Select **Subscribe**.

4. Configure your account.

1. To configure BigAnimal to use your Azure subscription and your Azure AD Application, select **Configure account now**.
2. Fill in the **Your BigAnimal Organization Name** parameter with the SaaS Subscription Name you assigned as your BigAnimal Organization.
3. Select **Submit**.

What's next

You can now:

- Log in to BigAnimal
- Invite new users
- Set up your cloud service provider

Log in

You can log in to your BigAnimal account using your Azure AD identity.

Invite users

You can invite new users by sharing the link to the BigAnimal portal and having them log in with their Microsoft Azure Active Directory account. New users aren't assigned any roles by default. After they log in the first time, you see them in the User Management list and can assign them a role with permissions to BigAnimal. See Users for instructions.

> **Note**
>
> Azure AD email domain is likely different from the email domain regularly used by your organization.

7 Checking your cloud readiness

When using your cloud account, each time you create a cluster, you must ensure the readiness of your cloud to work with BigAnimal. You need to perform this check only if you're not using BigAnimal's cloud account as your deployment option.

7.1 Preparing your Azure account

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal requires you to check the readiness of your Azure subscription before you deploy your clusters. (You don't need to perform this check if you're using BigAnimal's cloud account as your deployment option.) The checks that you perform ensure that your Azure subscription is prepared to meet your clusters' requirements and resource limits, such as:

- Are the necessary Azure resource providers registered for your subscription?
- Is there a restriction on SKUs for the standard Esv3 family and standard D2_v4 VM size?
- Is there a sufficient limit on the number of vCPU or public IP addresses in your region?

> **Note**
>
> Before proceeding, see Understanding requirements in Azure for details on planning for your clusters' requirements and resource limits in Azure.

Check for readiness

We recommend using the `biganimal-csp-preflight` script to check whether all requirements and resource limits are met in your subscription. However, you can also manually check the requirements using the Azure CLI or the Azure Portal.

- Method 1: Use EDB's shell script (recommended)
- Method 2: Manually check requirements

Method 1: Use EDB's shell script

EDB provides a shell script, called `biganimal-csp-preflight`, which checks whether requirements and resource limits are met in your Azure subscription based on the clusters you plan to deploy.

1. Open the Azure Cloud Shell in your browser.

2. From the Azure Cloud Shell, run the following command:

```
curl -sL https://raw.githubusercontent.com/EnterpriseDB/cloud-utilities/main/azure/biganimal-csp-preflight | bash -s <target-subscription> <region> [options]
```

The required arguments are:

Argument	Description
<target-subscription>	Azure subscription ID of your BigAnimal deployment.
<region>	Azure region where your clusters are being deployed. See Supported regions for a list of possible regions.

Possible options are:

Options	Description
`-h` or `--help`	Displays the command help.
`-i` or `--instance-type`	Azure VM instance type for the BigAnimal cluster. The `help` command provides a list of possible VM instance types. Choose the instance type that best suits your application and workload. Choose an instance type in the memory optimized ESv3 or ESv4 series for large data sets. Choose from the compute optimized FSv2 series for compute-bound applications. Choose from the general purpose DSv3 or DSv4 series if you don't require memory or compute optimization. See Sizes for virtual machines in Azure for information to help you choose the appropriate instance type.
`-a` or `--high-availability`	*DEPRECATED* - Enables high availability for the cluster. Replaced with `-x` or `--cluster-architecture` command.
`-x` or `--cluster-architecture`	Defines the cluster architecture and can be `single`, `ha`, or `eha`. See Supported cluster types for more information.
`-n` or `--networking`	Type of network endpoint for the BigAnimal cluster, either `public` or `private`. See Cluster networking architecture for more information.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Options	Description
`-r` or `--activate-region`	Specifies region activation if no clusters currently exist in the region.
`--onboard`	Checks if the user and subscription are correctly configured.

The behavior of the script defaults to `--onboard` if you provide no other options.

For example, if you want to deploy a cluster in an Azure subscription having an ID of `12412ab3d-1515-2217-96f5-0338184fcc04`, with an instance type of `e2s_v3`, in the `eastus2` region, in a `public` network, and with no existing cluster deployed, run the following command:

```
curl -sL https://raw.githubusercontent.com/EnterpriseDB/cloud-utilities/main/azure/biganimal-csp-preflight | bash -s \
12412ab3d-1515-2217-96f5-0338184fcc04 eastus2 \
--instance-type e2s_v3 \
--cluster-architecture ha \
--networking public \
--activate-region
```

The script displays the following output:

- A list of required Azure resource providers and their registration status. Ensure that you register the resource providers that are displayed as `NotRegistered` in the `RegistrationState` column. See Register Azure resource providers.

```
########################
# Provider             #
########################

Namespace                                RegistrationPolicy    RegistrationState    ProviderAuthorizationConsentState
---------------------------------------  --------------------  -----------------
Microsoft.Capacity                       RegistrationRequired  NotRegistered
Microsoft.ContainerInstance              RegistrationRequired  NotRegistered
Microsoft.Compute                        RegistrationRequired  NotRegistered
Microsoft.ContainerService               RegistrationRequired  NotRegistered
Microsoft.KeyVault                       RegistrationRequired  NotRegistered
Microsoft.ManagedIdentity                RegistrationRequired  NotRegistered
Microsoft.Network                        RegistrationRequired  NotRegistered
Microsoft.OperationalInsights            RegistrationRequired  NotRegistered
Microsoft.OperationsManagement           RegistrationRequired  NotRegistered
Microsoft.Portal                         RegistrationFree      Registered
Microsoft.Storage                        RegistrationRequired  Registered
Microsoft.AlertsManagement               RegistrationRequired  NotRegistered
```

- Whether your Azure subscription restricts vCPUs for the `Standard_D2_v4` and `Standard_E2s_v3` VM size families in your region (and availability zone, if HA is enabled). Open a support request to remove SKU restrictions for the VM families with `NotAvailableForSubscription` displayed in the `Restrictions` column. See Fix issues with SKU restrictions.

```
########################
# Virtual-Machine SKU #
########################

ResourceType      Locations              Name                  Zones    Restrictions
------------      ---------              ----                  -----    ------------
virtualMachines   eastus2                Standard_D2_v4        1,2,3    None
virtualMachines   eastus2                Standard_E2s_v3       1,2,3    NotAvailableForSubscription, type: Zone, locations:
eastus2, zones: 1,3
```

- Whether your Azure subscription has sufficient limits on vCPUs and IP addresses for your region. Open a support request to raise limits for the vCPUs and IP addresses if they exceed the available VM families with `NotAvailableForSubscription` displayed in the `Restrictions` column. See Increase Public IP addresses and Increase vCPU limits.

```
########################
# Quota Limitation #
########################

 Resource                      Limit   Used    Available    Gap   Suggestion
 Total Regional vCPUs          130     27      103          89    OK
 Standard Dv4 Family vCPUs     20      14      6            0     Need Increase
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
Standard ESv3 Family vCPUs      20      4       16          8     OK
Public IP Addresses – Standard  20      3       17          16    OK
```

Method 2: Manually check readiness

You can manually check the requirements instead of using the `biganimal-csp-preflight` script.

Check Azure resource provider registrations using Azure Cloud Shell

To check if an Azure resource provider is registered, use the following command.

```
az provider show -n Microsoft.ContainerService
```

```
Namespace                   RegistrationPolicy    RegistrationState
--------------------------  --------------------  -------------------
Microsoft.ContainerService  RegistrationRequired  Registered
```

Check for SKU restrictions for the specific VM size

You can check SKU restrictions for the VM size using the Azure Cloud Shell. For example, to check the Standard_E2s_v3 VM SKU restriction in `eastus2` location for all zones, run the following command:

```
az vm list-skus -l eastus2 --zone --size Standard_E2s_v3
```

```
ResourceType     Locations    Name              Zones    Restrictions
---------------  -----------  ----------------  -------  ------------
virtualMachines  eastus2      Standard_E2s_v3   1,2,3    NotAvailableForSubscription, type: Zone, locations: eastus2, zones: 3,2
```

Alternatively, to check for SKU restrictions using the Azure Portal, see Solution 3 - Azure portal.

Check the resource limits for vCPUs and public IP addresses for your region

To check if you have adequate Azure resources to provision new clusters:

1. In the Azure Portal, select **Subscription**.
2. Select your specific subscription.
3. Select **Usage + quotas** in the **Settings** section.
4. Search for `Total Regional vCPUs` and select the **Location** to check the regional vCPUs limits.
5. Search for `Dv4` and `Esv3` to view virtual machine limits.
6. Search for Public IP addresses to view network limits.

Configure your Azure subscription

After checking whether the requirements and resource limits are met, configure your Azure subscription.

> **Note**
>
> Before proceeding, see Understanding requirements in Azure for details on planning for your clusters' requirements and resource limits in Azure.

Register Azure resource providers

To register resource providers using the Azure Portal:

1. In the Azure Portal, select **Subscription**.
2. Select your specific subscription.
3. In the navigation panel **Settings** group, select **Resource providers**.
4. Review the status of the required providers. To register a provider, select the provider and, on the top menu, select **Register**.

To register resource providers using the Azure CLI, use the register command. For example:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
az provider register -n Microsoft.ContainerService
```

```
Registering is still on-going. You can monitor using 'az provider show -n Microsoft.ContainerService
```

Fix issues with SKU restrictions

Open a support request to remove SKU restrictions in a particular region.

Increase public IP addresses limits

Increase the limit of `Public IP Addresses - Basic` and `Public IP Addresses - Standard` for the regions where you plan to deploy your clusters with the total number of clusters you plan to use.

You can increase the number of public IP addresses for your account either by using the Azure portal or by submitting a support request. See:

- Request networking quota increase at subscription level using Usages + quotas
- Request Networking quota increase at subscription level using Help + support

Increase vCPU limits

You can increase the number of Dv4 or Esv3 family virtual machines per region by using the Azure Portal or by submitting a support request. See:

- Request a quota increase at a subscription level using Usages + quotas
- Request a quota increase by region from Help + support

7.1.1 Understanding requirements in Azure

Follow these BigAnimal requirements and recommended resource limits in Azure.

Azure resource provider registrations

BigAnimal creates and manages some of the resources using resource providers. For example, if you want to store keys, you need the Microsoft.KeyVault resource provider. This resource provider offers a resource type called *vaults* for creating the key vault. For more information, see Azure resource providers and types.

To prevent failures while creating your clusters, ensure that each of the following Azure resource providers are registered in your Azure subscription.

Provider namespace	Description
Microsoft.Compute	Runs cluster workloads on a virtual machine managed by the Azure Kubernetes service
Microsoft.ContainerInstance	Manages the Azure resource and regular maintenance job.
Microsoft.Capacity	Checks the Azure resource quota
Microsoft.AlertsManagement	Monitors failure anomalies
Microsoft.ContainerService	Manages cluster workloads run on the Azure Kubernetes Service
Microsoft.KeyVault	Encrypts and stores keys of the clusters' data volume and Azure's credential information
Microsoft.Storage	Backs up data to the Azure Service Account
Microsoft.ManagedIdentity	Manages software access to the local Azure services using Azure Managed-Identity
Microsoft.Network	Manages cluster workloads run in the Azure Kubernetes Service in the dedicated VNet
Microsoft.OperationalInsights	Manages clusters and performs workload logging (log workspace)
Microsoft.OperationsManagement	Monitors workloads and provides container insight
Microsoft.Portal	Provides a dashboard to monitor the running status of the clusters (using aggregated logs and metrics)

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Regional activation resources and access requirements

The table provides BigAnimal's regional activation resources and access requirements.

Resource type	Activated region	Network access	Description
Virtual network	Yes	N/A	
Storage account for backup	yes	Public	Public is required for faraway replication and cross region restore.
KeyVault for backup	Yes	Public	
Storage account for logs and metrics	Yes	Private	
KeyVault for AKS, logs, and metrics	Yes	Private	
Managed Identify	Yes	N/A	
NSG	Yes	N/A	
Private endpoint	Yes	N/A	
AKS	Yes	N/A	
Application insights	Yes	N/A	Not in use for custom monitoring
Log analytics workspaces	Yes	N/A	Not in use for custom monitoring

> **Note**
>
> A Storage account for TFstate is created on the first activated region per BigAnimal project. It is in private network access. Private DNS zone are required for private storage account and private key vaults.

Public IP addresses limits

Every BigAnimal cluster with public network access is assigned a single public IP address, and this IP address counts against the quota for both basic and standard IP address types in a region. BigAnimal can't create more clusters if the IP address limit is reached.

Recommended limit

The default public IP addresses limits for basic and standard type are set to 10. See Public IP address limits for more information. If you need more than 10 clusters, increase the limit to the number of clusters you plan to deploy plus current usage.

vCPU limits

Any time a new VM is deployed in Azure, the vCPUs for the VMs must not exceed the total vCPU limits for the region.

The number of cores required by the database cluster depends on the instance type and cluster type of the clusters. For exampe, if you create cluster with ESv3 instance type, you can calculate the number of ESv3 cores required for your cluster based on the following:

- A virtual machine instance of type E{N}sv3 uses {N} cores. For example, an instance of type E64sv3 uses 64 ESv3 cores.
- A cluster running on an E{N}sv3 instance without HA enabled uses exactly {N} ESv3 cores.
- A cluster running on an E{N}sv3 instance with HA enabled and 2 replicas uses 3 * {N} ESv3 cores.

BigAnimal requires an additional eight Dv4 virtual machine cores per region for management resources.

BigAnimal requires an additional six Dv4 virtual machine cores per region for periodic maintenance upgrades.

Recommended limits

BigAnimal recommends the following per region when requesting virtual machine resource limit increases:

- Total Regional vCPUs: minimum of 50 per designated region
- Standard Dv4 Family vCPUs: minimum of 14 per designated region
- Other Family vCPUs: depending on the instance type, cluster type, and number of clusters.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

7.2 Preparing your AWS account

BigAnimal requires you to check the readiness of your AWS account before you deploy your clusters. (You don't need to perform this check if you're using BigAnimal's cloud account as your deployment option.) The checks that you perform ensure that your AWS account is prepared to meet your clusters' requirements and resource limits, such as:

- Is the AWS CLI configured to access your AWS account?
- Is there a sufficient limit on the number of vCPUs and Network Load Balancers (NLBs) left in your region?

Check AWS resource limits for running BigAnimal

EDB provides a shell script, called biganimal-csp-preflight, which checks whether requirements and resource limits are met in your AWS account based on the clusters you plan to deploy.

1. Open the AWS Cloud Shell in your browser.

2. From the AWS Cloud Shell, run the following command:

```
curl -sL https://raw.githubusercontent.com/EnterpriseDB/cloud-utilities/main/aws/biganimal-csp-preflight | bash -s <AWS-account-ID> <region> [options]
```

The required arguments are:

Argument	Description
<account-id>	AWS account ID of your BigAnimal deployment.
<region>	AWS region where your clusters are being deployed. See Supported regions for a list of possible regions.

Possible options are:

Options	Description
`-h` or `--help`	Displays the command help.
`-i` or `--instance-type`	AWS instance type for the BigAnimal cluster. The help command provides a list of possible VM instance types. Choose the instance type that best suits your application and workload. Choose an instance type in the memory optimized R5, R5B, or R6I series for large data sets. Choose from the compute-optimized C5 or C6I series for compute-bound applications. Choose from the general purpose M5 or M6I series if you don't require memory or compute optimization.
`-a` or `--high-availability`	*DEPRECATED* - Enables high availability for the cluster. See [Supported cluster types(../../overview/02_high_availability) for more information.
`-x` or `--cluster-architecture`	Defines the Cluster architecture and can be `single`, `ha`, or `eha`. See Supported cluster types for more information.
`-n` or `--networking`	Type of network endpoint for the BigAnimal cluster, either `public` or `private`. See Cluster networking architecture for more information.
`-r` or `--activate-region`	Specifies region activation if no clusters currently exist in the region.
`--onboard`	Checks if the user and subscription are correctly configured.

The behavior of the script defaults to `--onboard` if you provide no other options.

For example, if you want to deploy a cluster in an AWS account having an ID of `1234-5678-9012`, with an instance type of `r5.24xlarge`, in the `us-east-1` region, in a `public` endpoint, and with no existing cluster deployed, run the following command:

```
curl -sL https://raw.githubusercontent.com/EnterpriseDB/cloud-utilities/main/aws/biganimal-csp-preflight | bash -s \
1234-5678-9012 \
us-east-1 \
--instance-type r5.24xlarge \
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
--networking public \
--activate-region \
--onboard\
```

The script displays the following output:

- Whether your AWS account restricts vCPUs, elastic IP addresses, VPCs, or NLBs in your region (and availability zone, if HA is enabled). Open an AWS support request to remove restrictions for the resources with `NotAvailable` displayed in the `Suggestion` column. See Request quota increase. For default service quota limits in AWS, seeAWS service information

```
 ############################################
Checking Service Quotas Limits on us-east-1...
############################################

Resource              Quota Name                           Limit    Used     Required    Gap      Suggestion
--------              ----------                           ----     -----    --------    ---      ----------
m5(a).large vCPUs     Running On-Demand Standard instances  512      0        6           410      OK
r5.24xlarge vCPUs     Running On-Demand Standard instances  512      0        96          410      OK
Elastic IP Addresses  EC2-VPC Elastic IPs                   5        0        3           2        OK
VPCs                  VPCs per Region                       5        1        2           2        OK
NLBs                  Network Load Balancers per Region     50       0        1           49       OK

Note: the first two Instance Types are referring to the same AWS Service Quota.
```

Configure your AWS account

Open an AWS support request to remove restrictions for the resources with `NotAvailable` displayed in the `Suggestion` column. See Request quota increase. For default service quota limits in AWS, see AWS service information.

7.3 Preparing your Google Cloud account

BigAnimal requires you to check the readiness of your Google Cloud (GCP) account before you deploy your clusters. (You don't need to perform this check if you're using BigAnimal's cloud account as your deployment option.) The checks that you perform ensure that your Google Cloud account is prepared to meet your clusters' requirements and resource limits.

> **Note**
>
> Before proceeding, see Understanding requirements in Google Cloud for details on planning for your clusters' requirements and resource limits in Google Cloud.

Required roles

Ensure you have at least the following combined roles:

- roles/iam.serviceAccountCreator
- roles/iam.serviceAccountKeyAdmin
- roles/iam.roleAdmin
- roles/resourcemanager.projectIamAdmin
- roles/compute.viewer

Alternatively, you can have an equivalent single role, such as:

- roles/owner

Required APIs and services

Ensure the following Google Cloud APIs are enabled:

- autoscaling.googleapis.com
- cloudapis.googleapis.com
- cloudresourcemanager.googleapis.com
- compute.googleapis.com
- container.googleapis.com

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- iam.googleapis.com
- iamcredentials.googleapis.com
- run.googleapis.com
- secretmanager.googleapis.com
- storage.googleapis.com
- vpcaccess.googleapis.com

Check Google Cloud resource limits for running BigAnimal

EDB provides a shell script, called biganimal-csp-preflight, which checks whether requirements and resource limits are met in your Google Cloud account based on the clusters you plan to deploy.

1. Open Google Cloud Shell in your browser.

2. From Google Cloud Shell, run the following command:

```
curl -sL https://raw.githubusercontent.com/EnterpriseDB/cloud-utilities/main/gcp/biganimal-csp-preflight | bash -s <GCP-project-ID> <region> [options]
```

The required arguments are:

Argument	Description
<project-id>	Google Cloud project ID of your BigAnimal deployment.
<region>	Google Cloud region where your clusters are being deployed. See Supported regions for a list of possible regions.

Possible options are:

Options	Description
`-h` or `--help`	Displays the command help.
`-i` or `--instance-type`	Google Cloud instance type for the BigAnimal cluster. The help command provides a list of possible VM instance types. Choose the instance type that best suits your application and workload. Choose an instance type in the memory optimized M1, M2, or M3 series for large data sets. Choose from the compute-optimized C2 series for compute-bound applications. Choose from the general purpose E2, N2, and N2D series if you don't require memory or compute optimization.
`-x` or `--cluster-architecture`	Defines the Cluster architecture and can be `single`, `ha`, or `eha`. See Supported cluster types for more information.
`-e` or `--networking`	Type of network endpoint for the BigAnimal cluster, either `public` or `private`. See Cluster networking architecture for more information.
`-r` or `--activate-region`	Specifies region activation if no clusters currently exist in the region.
`--onboard`	Checks if the user and subscription are correctly configured.

The behavior of the script defaults to `--onboard` if you provide no other options.

For example, if you want to deploy a cluster in a Google Cloud account having an ID of `1234-5678-9012`, with an instance type of `n2-standard-8`, in the `us-east1` region, with a `public` endpoint, and with no existing cluster deployed, run the following command:

```
curl -sL https://raw.githubusercontent.com/EnterpriseDB/cloud-utilities/main/gcp/biganimal-csp-preflight | bash -s \
1234-5678-9012 \
us-east1 \
--instance-type n2-standard-8 \
--networking public \
--activate-region \
--onboard\
```

The script displays the following output:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
```
##
Run GCP Preflight Checks with Google Cloud CLI 436.0.0

###################################
Checking for enabled GCP APIs...
###################################

NAME   RESULT
------------------------------------ ---------
autoscaling.googleapis.com   Enabled
cloudapis.googleapis.com   Enabled
cloudresourcemanager.googleapis.com   Enabled
compute.googleapis.com   Enabled
container.googleapis.com   Enabled
iam.googleapis.com   Enabled
iamcredentials.googleapis.com   Enabled
run.googleapis.com   Enabled
secretmanager.googleapis.com   Enabled
storage.googleapis.com   Enabled
vpcaccess.googleapis.com   Enabled

##
Checking Service Quotas Limits on us-east1...
##

Resource   Quota Name   Limit   Used   Required   Gap   Suggestion
--------   ----------   ----   -----   --------   ---   ----------
n2-standard-2 vCPUs   N2_CPUS   500   0   6   494   OK
e2-standard-8 vCPUs   E2_CPUS   2400   0   24   2376   OK
Shared-Core vCPUs   CPUS   2400   0   2   2398   OK
Static IP Addresses   STATIC_ADDRESSES   700   0   1   699   OK
VPCs   NETWORKS   50   14   1   35   OK
Cloud Routers   ROUTERS   20   6   1   13   OK

######################
Overall Suggestions
######################
```

Make sure the GCP Project ID <project_id> is the one that you want to use for BigAnimal.
Make sure the GCP account <gcp_account> has rights to create custom roles, service accounts, keys, and assign project grants.

Use the Quotas page in the Google Cloud console if you need to raise any service quota limits.
See https://cloud.google.com/docs/quota_detail/view_manage#requesting_higher_quota for more information.
```

Configure your Google Cloud account

If any APIs are listed as not enabled, see Enabling and Disabling Services in the Google Cloud documentation to enable the required APIs.

If you need to increase your quotas, see Request a higher quota limit.

7.3.1 Understanding requirements in Google Cloud

Follow these BigAnimal requirements and recommended resource limits in Google Cloud.

vCPU limits

Any time a new VM is deployed in Google Cloud, the vCPUs for the VMs must not exceed the total vCPU limits for the region.

The number of cores required by the database cluster depends on the instance type and cluster type of the clusters. For example, if you create a cluster with the standard E2 instance type, you can calculate the number of E2 cores required for your cluster based on the following:

- A virtual machine instance of type e2-standard-{N} uses {N} cores. For example, an instance of type e2-standard-32 uses 32 e2-standard cores.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- A cluster running on an e2-standard-{N} instance without HA enabled uses exactly {N} e2-standard cores.
- A cluster running on an e2-standard-{N} instance with HA enabled and 2 replicas uses 3 * {N} e2-standard cores.

BigAnimal requires an additional eight n2-standard virtual machine cores per region for management resources.

BigAnimal requires an additional four n2-standard virtual machine cores per region for periodic maintenance upgrades.

Recommended limits

BigAnimal recommends the following per region when requesting virtual machine resource limit increases:

- Total Regional vCPUs: minimum of 50 per designated region
- n2-standard vCPUs: minimum of 12 per designated region
- Other machine family vCPUs: depends on the instance type, cluster type, and number of clusters.

8 Connecting your cloud

Use these techniques to connect your cloud if you're using your own cloud account as yourdeployment option.

BigAnimal access requirements

BigAnimal needs access to your cloud to perform maintenance. BigAnimal CLI commands used for connecting your cloud to BigAnimal set some of these permissions for you. For details on these and other permissions or policies required by BigAnimal, refer to the corresponding topic for your cloud provider.

BigAnimal also requires permissions to run Kubernetes cluster services for PostgreSQL workloads and the associated storage services. It requires a set of supporting permissions:

- Services for monitoring and logging to produce service telemetry information
- To set up networking so PostgreSQL workloads are reachable by customer applications and telemetry data is collected
- To provision vaults for safe storage of data at rest encryption keys
- To create workload identities and manage their permissions
- A small set of supporting permissions to ensure access to the services above and availability of cloud account information

The scope of these permissions is limited to the associated cloud account.

Prerequisites

In your cloud provider shell, make sure that your environment is running:

- Bash shell version 4.0 or above.
- BigAnimal CLI version 2.0 or later. For details, see Installing the CLI.

For additional cloud provider-specific requirements, see Setting up specific cloud providers.

Overview of connecting your cloud

> Tip

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

> If you're using Cloud Shell, add the `./` prefix to the `biganimal` command (`./biganimal`).

1. Open your cloud provider shell in your browser.

2. Create a BigAnimal CLI credential:

   ```
   biganimal credential create --name <cred>
   ```

3. To set up your cloud provider, run the `cloud-provider setup` command :

   ```
   biganimal cloud-provider setup
   ```

   > **Important**
   >
   > Don't delete the `ba-passport.json` file created in your working directory. It contains important identity and access management information used by `cloud-provider connect` while connecting to your cloud.

4. If the cloud readiness checks pass, your cloud account is successfully set up. Connect your cloud account to BigAnimal:

   ```
   biganimal cloud-provider connect --provider  <cloud-service-provider> --project <project-name>
   ```

   Once your cloud account is successfully connected to BigAnimal, you and other users with the correct permissions can create clusters.

Setting up specific cloud providers

For step-by-step instructions for setting up specific cloud providers, see:

- Connecting your AWS cloud
- Connecting your Azure cloud
- Connecting your Google Cloud

8.1 Connecting your AWS cloud

Prerequisites

Before connecting your cloud, make sure that you're assigned the following AWS managed policies or an equivalent custom policy granting full access to resources:

- arn:aws:iam::aws:policy/IAMFullAccess
- arn:aws:iam::aws:policy/ServiceQuotasFullAccess

Connecting your cloud

> **Tip**
>
> If you're using CloudShell, add the `./` prefix to the `biganimal` command (`./biganimal`).

To connect your cloud:

1. Open AWS CloudShell in your browser.

2. Create a BigAnimal CLI credential:

   ```
   biganimal credential create --name <cred>
   ```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

3. To set up your cloud provider, run the `setup-csp` command:

   ```
   biganimal cloud-provider setup
   ```

   The command checks for cloud account readiness and displays the results.

4. If the following readiness checks aren't met for your cloud service provider, see Configure your AWS account to manually configure your cloud:

   - Is the AWS CLI configured to access your AWS account?
   - Is the limit on the number of vCPUs and network load balancers (NLBs) in your region enough for your clusters?

   If the cloud readiness checks pass, your cloud account is successfully set up.

5. Connect your cloud account to BigAnimal:

   ```
   biganimal cloud-provider connect --provider aws --project <project-name>
   ```

   After your cloud account is successfully connected to BigAnimal, you and other users with the correct permissions can create clusters.

8.2 Connecting your Azure cloud

> **Note**
>
> The BigAnimal CLI commands used for connecting your cloud to BigAnimal register an application with Azure AD and create a service principal to delegate identity and access management functions to Azure Active Directory (AD). For more information, see for Azure EDB cloud utilities in GitHub.

Prerequisites

Before connecting to your cloud, make sure that you're assigned either the Global Administrator role or the Privileged Role Administrator role in Azure AD and that you have the Owner role for your BigAnimal Azure subscription.

Connecting your cloud

> **Tip**
>
> If you're using Cloud Shell, add the `./` prefix to the `biganimal` command (`./biganimal`).

To connect your cloud:

1. Open the Azure Cloud Shell in your browser.

2. Create a BigAnimal CLI credential:

   ```
   biganimal credential create --name <cred>
   ```

3. To set up your cloud provider, run the `cloud-provider setup` command:

   ```
   biganimal cloud-provider setup
   ```

   The command checks for cloud account readiness and displays the results.

4. If the following readiness checks aren't met, see Configure your Azure subscription to manually configure your cloud:

   - Are the necessary Azure resource providers registered for your subscription?
   - Is there a restriction on SKUs for the standard Esv3 family and standard D2_v4 VM size?
   - Is the limit on the number of vCPU and public IP addresses in your region enough for your clusters?

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

If the cloud readiness checks pass, your cloud account is successfully set up.

> **Note**
>
> At this point, you can't change the Azure subscription, as you have already provided the Azure subscription ID.

5. Connect your cloud account to BigAnimal:

```
biganimal cloud-provider connect --provider azure --project <project-name>
```

After your cloud account is successfully connected to BigAnimal, you and other users with the correct permissions can create clusters.

8.3 Connecting your Google Cloud

Prerequisites

Ensure you have at least the following combined roles:

- roles/iam.serviceAccountCreator
- roles/iam.serviceAccountKeyAdmin
- roles/iam.roleAdmin
- roles/resourcemanager.projectIamAdmin
- roles/compute.viewer

Alternatively, you can have an equivalent single role, such as:

- roles/owner

Connecting your cloud

> **Tip**
>
> If you're using Cloud Shell, add the `./` prefix to the `biganimal` command (`./biganimal`).

To connect your cloud:

1. Open Google Cloud Shell in your browser.
2. Create a BigAnimal CLI credential:

```
biganimal credential create --name <cred>
```

3. To set up your cloud provider, run the `setup-csp` command:

```
 biganimal cloud-provider setup
```

   The command checks for cloud account readiness and displays the results.

4. If the following cloud readiness checks pass for your cloud service provider, your cloud account is successfully set up:
   - Is the Google Cloud CLI configured to access your Google Cloud account?
   - Is the limit on the number of vCPUs and network load balancers (NLBs) in your region enough for your clusters?

   If the readiness checks aren't met, see Configure your Google Cloud account to manually configure your cloud.

5. Connect your cloud account to BigAnimal:

```
biganimal cloud-provider connect --provider gcp --project <project-name>
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

After your cloud account is successfully connected to BigAnimal, you and other users with the correct permissions can create clusters.

9 Managing regions

> **Note**
>
> If you're using BigAnimal's cloud account, you don't need to activate regions. However, you can suspend and delete regions.

Activate a new region from the Regions page

When you activate a region, BigAnimal prepares the compute and networking resources required to deploy clusters. These added resources can increase your cloud costs.

You must activate a region before creating or restoring a cluster.

Each region you activate displays a status. The status is available on the Create cluster and Restore cluster pages and the Regions page. For more information on the different region statuses, see Region status reference.

You can activate a region ahead of time using the Regions page.

1. To activate a region before creating a cluster, go to the Regions page.
2. If you haven't set up your cloud server provider (CSP), you're prompted to do so. See Connecting your cloud.

   When your cloud server provider is set up, you see a list of regions associated with it on the Regions page.
3. Select **Activate New Region**, and select your cloud provider and the region you want to activate. You can activate multiple regions where you plan on adding clusters.
4. Select **Activate Region(s)**.

Suspend, reactivate, or delete a region

Before you suspend or delete a region, you must delete all clusters in that region.

1. On the left panel, select **Regions**.

   A list of previously activated regions appears.
2. Select the icons next to a region to:

   - Suspend or delete an active region.
   - Reactivate a suspended region.

Region status reference

Status	Description
Activating	You have activated the region, and BigAnimal is setting up the compute and networking resources to support the cluster you requested. Backups and logs aren't available yet.
Active	BigAnimal has completed setting up the compute and networking resources. The region has at least one cluster, and you can create more. The region has backups and logs.
Suspended	You have deleted any active clusters. BigAnimal has removed the compute and networking resources. The backups and logs are still available. BigAnimal continues to maintain backups and logs based on the retention period set when the cluster was last active.
Deleted	You have deleted the clusters. BigAnimal has removed the compute and networking resources. The backups and logs are no longer available. Only account owners can delete a region.
Error	An error occurred while trying to activate or suspend the region. The cause might be a quota issue. See the topic for your cloud provider underChecking your cloud readiness for information on checking whether requirements and resource limits are met and how to raise them, if needed. Contact EDB Support if you need additional help troubleshooting the error.

See also

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- Creating your cluster
- Supported regions

10 Creating a cluster

> **When using your cloud**
>
> Before creating your cluster, make sure you have enough resources. Without enough resources, your request to create a cluster fails.
>
> - If your cloud provider is Azure, see Preparing your Azure account.
> - If your cloud provider is AWS, see Preparing your AWS account.
> - If your cloud provider is Google Cloud, see Preparing your Google Cloud account.
> - Activate a region before cluster creation. See Activating regions.

> **When using BigAnimal's cloud**
>
> The following options aren't available when creating your cluster:
>
> - AWS IAM authentication
> - Superuser access
> - PgBouncer

1. Sign in to the BigAnimal portal.
2. On the Overview or Clusters page, select **Create New Cluster**.
3. On the Create Cluster page, specify the cluster settings on the following tabs:
    - **Cluster Info**
    - **Cluster Settings**
    - **DB Configuration** (optional)
    - **Additional Settings** (optional)
4. Select **Create Cluster**. It might take a few minutes to deploy.

> **Note**
>
> When you don't configure settings on optional tabs, the default values are used.

Cluster Info tab

1. Select the type of cluster to deploy.
    - Single Node creates a cluster with one primary and no standby replicas. Suited for test environments where high availability might not be required. You can create single-node clusters running PostgreSQL or EDB Postgres Advanced Server.
    - Primary/Standby High Availability creates a cluster with one primary and one or two standby replicas in different availability zones. You can create primary/standby high-availability clusters running PostgreSQL or EDB Postgres Advanced Server. Only primary/standby high-availability clusters allow you to enable read-only workloads for users. However, if you enable read-only workloads, then you might have to raise the IP address resource limits for the cluster.
    - Distributed High Availability creates a cluster, powered by EDB Postgres Distributed, with up to two data groups spread across multiple cloud regions to deliver higher performance and faster recovery. See Creating a distributed high-availability cluster for instructions.

    See Supported cluster types for more information about the different cluster types.

    > **Note**
    >
    > You can't switch from a single-node or primary/standby high-availability cluster to a distributed high-availability cluster or vice versa.

2. Select the number of standby replicas for your primary/standby high-availability cluster.
3. Select the type of cloud account for your cluster. You can choose BigAnimal's cloud account or set up your own. For more information, see Deployment options.

    > **Note**

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

> If you use BigAnimal's cloud account, your estimated cloud infrastructure costs are calculated into your cluster's estimated monthly price shown at the bottom of the page.

4. Select the cloud provider for your cluster. If you're using your account and haven't connected it to BigAnimal yet, see Connecting to your cloud.

5. Select **Next: Cluster Settings** to continue to specify the required settings for your cluster.

Cluster Settings tab

1. In the **Cluster Name** field, enter the name for your cluster.

2. In the **Password** field, enter a password for your cluster. This is the password for the user edb_admin.

3. Under **Tags**, select **+**.

4. To assign an existing tag, in the search bar under **Tags**, enter a tag name. To add a new tag, instead select **+ Add Tag**.

5. In the **Database Type** section:

    1. In the **Postgres Type** field, select the type of Postgres you want to use:

        - **EDB Postgres Advanced Server** is EDB's Oracle-compatible database offering. View a quick demonstration of Oracle compatibility on BigAnimal. EDB Postgres Advanced Server is compatible with all three cluster types.
        - **EDB Postgres Extended Server** is EDB's PostgreSQL-compatible database offering that uses advanced logical replication.
        - **PostgreSQL** is the open-source, object-relational database management system. PostgreSQL is compatible with single-node and primary/standby high-availability cluster types.

    2. In the **Postgres Version** list, select the version of Postgres that you want to use. See Database version policy for more information.

6. In the **Region** section, select the region where you want to deploy your cluster.

    > **Tip**
    >
    > For the best performance, we strongly recommend that this region be the same as your other resources that communicate with your cluster. For a list of available regions, see Supported regions. If you're interested in deploying a cluster to a region that isn't currently available, contact Support.

7. In the **Instance Type** section:

    1. Select the category that works best for your applications and workload:

        - Memory optimized for large data sets
        - Compute optimized for compute bound applications
        - General purpose if you don't require memory or compute optimization

    2. Select the instance series and size. See Sizes for virtual machines in Azure, Amazon EC2 Instance Types, or the Google Cloud Machine families resource and comparison guide for information to help you choose the appropriate instance type.

        > **Note**
        >
        > When provisioning a cluster, some CPU and memory resources are reserved for use by BigAnimal and your cloud provider. For example, when using Kubernetes, provisioning a server with 8GB of memory yields only about 6GB of memory after accounting for the requirements of Kubernetes and BigAnimal.

        > **Tip**
        >
        > To maximize your disk size for AWS, select R5b as your instance and then io2 Block Express as your storage to get a maximum disk size of 64 TB and 256,000 IOPS.

8. In the **Storage** section:

    By default, the **Database Storage** volume stores the Postgres data and the write-ahead logs (WAL) together. If you want to improve write performance for WAL files, you can allocate separate storage volume for the WAL files. To allocate separate storage volume for WAL files, select **Use a separate storage volume for Write-Ahead Logs**. Then select the volume type, size, IOPS, and disk throughput separately for **Database Storage** and **Write-Ahead Logs Storage**. If you allocate separate storage volume for the WAL files, you have to pay cloud infrastructure costs for the second volume. Once separate storage volume is allocated for WAL files, you can't remove it from the cluster settings later on.

    From the **Volume Type** list, select your volume type.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- For Azure, in **Volume Type**, select **Premium SSD** or **Ultra Disk**. Compared to Premium SSD volumes, ultra disks offer lower-latency, high-performance options and direct control over your disk's input/output operations per second (IOPS). For BigAnimal, we recommend using ultra disks for workloads that require the most demanding performance. See Using Azure ultra disks for more information.

  - For Premium SSD, in **Volume Properties**, select the type and amount of storage needed for your cluster. See Azure Premium SSD storage types for more information.

  - For ultra disk, in **Volume Properties**, select the disk size and IOPS for your cluster. BigAnimal calculates disk throughput based on your IOPS settings, but you have the option of updating the value.

    > **Important**
    >
    > While setting the required IOPS for the disk that you selected, consider the VM limits that are tied to the VM size that you selected. See Ultra disk IOPS for more information.

- For AWS, in **Volume Type**, select **General Purpose SSD (GP3)**, **io2**, or **io2 Block Express**.

  > **Note**
  >
  > io2 Block Express is available for selected instance types, such as R5b. However, you can't switch between io2 and io2 Block Express after creating your cluster.

  In **Volume Properties**, select the disk size for your cluster, and configure the IOPS.

- For Google Cloud, in **Volume Type**, select **SSD Persistent Disk**. In **Volume Properties**, select the disk size for your cluster.

> **Note**
>
> When provisioning database storage, not all of the storage space you specify is available for holding your data. Some space is reserved for other purposes. For a full explanation of the structure of a Postgres data directory, see Database File Layout. You can make more storage space available for data if you specify separate storage for write ahead logs (WAL).

9. Network, Logs, & Telemetry section

   In **Connectivity Type**, specify whether to use private or public networking. Networking is set to **Public** by default. Public means that any client can connect to your cluster's public IP address over the internet. Optionally, you can limit traffic to your public cluster by specifying an IP allowlist, which allows access only to certain blocks of IP addresses. To limit access, add one or more classless inter-domain routing (CIDR) blocks in the **IP Allowlists** section. CIDR is a method for allocating IP addresses and IP routing to a whole network or subnet. If you have any CIDR block entries, access is limited to those IP addresses. If none are specified, all network traffic is allowed.

   Private networking allows only IP addresses in your private network to connect to your cluster.

   If you're using private networking and you're using BigAnimal's cloud account, you must enter your cloud service provider's account information. BigAnimal uses it to auto-approve your private endpoints and also to provide access for logging and telemetry. To enable you to create your private endpoint, BigAnimal provides you with the corresponding service name, which you can use to connect to your cluster privately.

   See Cluster networking architecture for more information.

10. To optionally make updates to your database configuration parameters, select **Next: DB Configuration**.

DB Configuration tab

In the **Parameters** section, you can update the value of the database configuration parameters as needed.

To update the parameter values, see Modifying your database configuration parameters.

For other optional settings, select **Next: Additional Settings**.

Additional Settings tab

Backups

Change the default database backup retention period of 30 days using the **Retention Time** controls in the **Backups** section. You can configure the retention period to a number of days, weeks, or months. The retention period must be between 1-180 days, 1-25 weeks, or 1-6 months.

BigAnimal deletes backups older than the retention period.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Maintenance

Enable the **Custom Maintenance Window** option and use the controls to set a weekly 60-minute maintenance window in which maintenance upgrades occur for the cluster. If you don't set a window, the updates are applied at EDB's discretion with prior notification.

> **Note**
>
> Typically, maintenance updates take only a few minutes to complete.

For more information, see Periodic maintenance.

Connections

Read-only workloads

> **Note**
>
> The **Read-only Workloads** option is available only for primary/standby high-availability cluster types.

Enable **Read-only Workloads**. This feature directs read-only operations exclusively toward replicas. If this option is enabled, you might have to raise the IP address resource limits for the cluster:

- For Azure, the IP address quota is Standard Public IP Address.
- For AWS, the IP address quota is Elastic IP. You might also have to increase the **Network Load Balancers per Region** value.

When enabling read-only workloads, keep in mind the following:

- Read-only workloads are routed to Postgres physical standbys. Commands run on read-only workloads aren't filtered by BigAnimal. The connection is read-only because it runs on a standby replica where Postgres doesn't permit changes to the contents of database tables. A privileged connection to a standby replica can still execute other sensitive commands permitted by Postgres on standby replicas. For example, it can modify replication slots or Postgres configuration settings, terminate backends, see activity from other users, and more. We recommend that you use a Postgres role with minimal privileges for your application, even for read-only workloads.
- Advisory locks aren't replicated between Postgres nodes, so advisory locks taken on a standby replica don't conflict with advisory locks taken on the primary or another standby replica. We recommend that applications that rely on advisory locking avoid using read-only workloads for those transactions.

For information on replication lag while using read-only workloads, see Standby replicas.

PgBouncer

> **Note**
>
> Enabling PgBouncer incurs additional costs. For more information, see PgBouncer costs.

Enable **PgBouncer** to have it manage your connections to Postgres databases and help your workloads run more efficiently — all entirely managed by BigAnimal. Learn more about EDB PgBouncer.

Use the **PgBouncer Configuration Settings** menu to set PgBouncer-specific settings. Select the **Read-Write** and **Read-Only** tabs according to the type of connection you want to configure. The **Read-Only** tab is available if you're creating a primary/standby high-availability cluster and have enabled read-only workloads.

Access

Identity and Access Management (IAM) Authentication

Enable **Identity and Access Management (IAM) Authentication** to turn on the ability to log in to Postgres using your AWS IAM credentials. For this feature to take effect, after you create the cluster, you must add each user to a role that uses AWS IAM authentication in Postgres. For details, see IAM authentication for Postgres.

Superuser Access

Enable **Superuser Access** to grant superuser privileges to the edb_admin role. This option is available for single-node and primary/standby high-availability clusters. See Notes on the edb_admin role.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Security

Enable **Transparent Data Encryption (TDE)** to use your own encryption key. This option is available for EDB Postgres Advanced Server and EDB Postgres Extended Server for version 15 and later on AWS. Select an encryption key from your project and region to encrypt the cluster with TDE.

> **Important**
>
> - To enable and use TDE for cluster, the encryption key must be enabled and added at the project level. For more information, see Adding a TDE key at project level.
> - If the TDE enabled cluster creation is in progress and the process is waiting providing **Encryption Key Error** next to the cluster name on the clusters page:
>   - Select the cluster name and go to the cluster's home page.
>   - See the **Action required: grant key permissions to activate the cluster** on the cluster's overview tab.
>   - Copy the **Principal** and add it to your key policy to provide **kms:Encrypt** and **kms:Decrypt** permission.

What's next

After you create your cluster, use these resources to learn about cluster use and management:

- Using your cluster
- Managing Postgres access

Related CLI commands

For information on related CLI commands, see:

- Managing clusters using the CLI
- Maintenance windows CLI command

10.1 Cluster networking architecture

BigAnimal clusters can be exposed to client applications in two ways:

- Public — The cluster is available on the internet.
- Private — The IP address or DNS name is private to the VNet or VPC hosting your BigAnimal services. By default, it isn't routable from other networks. See Connecting to your cluster from your application for details and instructions on how to properly configure routing for private clusters.

Basic architecture

BigAnimal deploys a dedicated virtual network (VNet) in Azure, an Amazon Virtual Private Cloud (VPC) in AWS, or a Google VPC in Google Cloud to host clusters and their supporting management services. This VNet or VPC is named after the region where the cluster is deployed. In Azure, for example, if the cluster is deployed in the East US region, the name is `vnet-eastus`.

Load balancing

BigAnimal uses the following resources for making routing decisions and distributing requests:

Standard SKU load balancer in Azure

When a cluster is created with public network access, a load balancer is created and configured with a public IP address. Once assigned, this IP address doesn't change unless you change the networking configuration for your cluster. The load balancer always routes to the leader of your cluster.

Only one load balancer is typically deployed in an Azure region. BigAnimal adds more IP addresses to the existing load balancer for subsequent clusters in the Azure region.

Every BigAnimal cluster, regardless of public or private networking status, is assigned a single DNS zone that maps to its exposed IP address, either public or private. When toggling between public and private, wait up to 120 seconds for DNS caches to flush.

Clusters can change from public to private and vice versa at any time. When this happens, the IP address previously assigned to the cluster is deallocated, a new one is assigned, and DNS is

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

updated accordingly.

Amazon network load balancer in AWS

BigAnimal creates a new load balancer for each cluster and tags it with the cluster ID in the following format:

```
service.k8s.aws/stack: default/<cluster_ID>
```

An example is `service.k8s.aws/stack: default/p-c4j0jfcmp3af2ieok5eg`.

Because the load balancer IP address used in AWS is dynamic, make sure that your application uses the correct DNS name to access the network load balancer of a particular cluster. In your application's AWS account:

1. Select the **Load Balancers** service.
2. Search for the load balancer with the cluster ID you want to access.
3. Use the DNS name provided in the details section to access your cluster.

Google load balancer in Google Cloud

BigAnimal creates a new load balancer using the Premium Network Service Tier for each cluster and tags it using a unique identifier. The corresponding front-end forwarding rule uses the same unique identifier and includes the cluster ID in the following format:

```
{"kubernetes.io/service-name":"default/<cluster_ID>-<service_type>"}
```

An example is `{"kubernetes.io/service-name":"default/p-8jz4kedbiy-rw-external-lb"}`.

Because the load balancer IP address used in Google Cloud is dynamic, make sure that your application uses the correct DNS name to access the network load balancer of a particular cluster. To be able to access it, make sure you're using the FQDN that BigAnimal provides in the Cluster Overview or Connect page.

10.2 Creating a distributed high-availability cluster

> **When using your cloud**
>
> Before creating your cluster, make sure you have enough resources. Without enough resources, your request to create a cluster fails.
>
> - If your cloud provider is Azure, see Preparing your Azure account.
> - If your cloud provider is AWS, see Preparing your AWS account.
> - If your cloud provider is Google Cloud, see Preparing your Google Cloud account.
> - Activate a region before cluster creation. See Activating regions.

> **When using BigAnimal's cloud**
>
> The following options aren't available when creating your cluster:
>
> - AWS IAM authentication
> - Superuser access
> - PgBouncer

1. Sign in to the BigAnimal portal.

2. On the Overview or Clusters page, select **Create New Cluster**.

3. On the Create Cluster page, start with the **Cluster Info** tab.

Cluster Info tab

1. Select the type of cluster to deploy.

    - Distributed High Availability creates a cluster, powered by EDB Postgres Distributed, with up to two data groups spread across multiple cloud regions to deliver higher performance and faster recovery.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

If you want to create a single-node or primary/standby high-availability cluster, follow the Single Node or Primary/Standby High Availability page.

See Supported cluster types for more information about the different cluster types.

> **Note**
>
> You can't switch from a single-node or primary/standby high-availability cluster to a distributed high-availability cluster or vice versa.

2. Select the type of cloud account for your cluster. You can choose BigAnimal's cloud account or set up your own. For more information, see Deployment options.

> **Note**
>
> If you use BigAnimal's cloud account, your estimated cloud infrastructure costs are calculated into your cluster's estimated monthly price shown at the bottom of the page.

3. Select **Next: Cluster Settings**

Cluster Settings tab

1. In the **Cluster Name** field, enter the name for your cluster.
2. In the **Password** field, enter a password for your cluster. This is the password for the user edb_admin.
3. Under **Tags**, select **+**.
4. To assign an existing tag, in the search bar under **Tags**, enter a tag name. To add a new tag, instead select **+ Add Tag**.
5. Select **Next: Data Groups**.

Data Groups Tab

1. Go to the **Nodes Settings** tab for the first Data Group. In the **Nodes** section, select **Two Data Nodes** or **Three Data Nodes**.

   For more information on node architecture, see Distributed high availability.

2. In the **Provider & Region** section, select the provider and region where you want to deploy your cluster.

> **Tip**
>
> For the best performance, we strongly recommend that this region be the same as your other resources that communicate with your cluster. For a list of available regions, see Supported regions. If you're interested in deploying a cluster to a region that isn't currently available, contact Support.

3. In the **Database Type** section:

   1. Select the type of Postgres you want to use in the **Postgres Type** field:

      - **EDB Postgres Advanced Server** is EDB's Oracle-compatible database offering. View a quick demonstration of Oracle compatibility on BigAnimal.
      - **EDB Postgres Extended Server** is EDB's advanced logical replication, PostgreSQL-compatible database offering.

4. In the **Postgres Version** list, select 14, 15, or 16 as the version of Postgres that you want to use.
5. In the **Instance Type** section:

   1. Select the category that works best for your applications and workload:

      - Memory optimized for large data sets
      - Compute optimized for compute bound applications
      - General purpose if you don't require memory or compute optimization

   2. Select the instance series and size. See Sizes for virtual machines in Azure, Amazon EC2 Instance Types, or the Google Cloud Machine families resource and comparison guide for information to help you choose the appropriate instance type.

      > **Note**

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

> When provisioning a cluster, some CPU and memory resources are reserved for use by BigAnimal and your cloud provider. For example, when using Kubernetes, provisioning a server with 8GB of memory yields only about 6GB of memory after accounting for the requirements of Kubernetes and BigAnimal.

> **Tip**
>
> To maximize your disk size for AWS, select R5b as your instance and then io2 Block Express as your storage to get a maximum disk size of 64 TB and 256,000 IOPS.

6. In the **Storage** section:

> **Tip**
>
> When choosing your storage options, for most workloads, consider using at least 20GB of storage.

By default, the **Database Storage** volume stores the Postgres data and the write-ahead logs (WAL) together. If you want to improve write performance for WAL files, you can allocate separate storage volume for the WAL files. To allocate separate storage volume for WAL files, select **Use a separate storage volume for Write-Ahead Logs**. Then select the volume type, size, IOPS, and disk throughput separately for **Database Storage** and **Write-Ahead Logs Storage**. If you allocate separate storage volume for the WAL files, you have to pay cloud infrastructure costs for the second volume. Once separate storage volume is allocated for WAL files, you can't remove it from the cluster settings later on.

From the **Volume Type** list, select your volume type.

- For Azure, in **Volume Type**, select **Premium SSD** or **Ultra Disk**. Compared to Premium SSD volumes, ultra disks offer lower-latency, high-performance options and direct control over your disk's input/output operations per second (IOPS). For BigAnimal, we recommend using ultra disks for workloads that require the most demanding performance. See Using Azure ultra disks for more information.
  - For Premium SSD, in **Volume Properties**, select the type and amount of storage needed for your cluster. See Azure Premium SSD storage types for more information.
  - For ultra disk, in **Volume Properties**, select the disk size and IOPS for your cluster. BigAnimal calculates disk throughput based on your IOPS settings, but you have the option of updating the value.

    > **Important**
    >
    > While setting the required IOPS for the disk that you selected, consider the VM limits that are tied to the VM size that you selected. See Ultra disk IOPS for more information.

- For AWS, in **Volume Type**, select **General Purpose SSD (GP3)**, **io2**, or **io2 Block Express**.

  > **Note**
  >
  > io2 Block Express is available for selected instance types, such as R5b. However, you can't switch between io2 and io2 Block Express after creating your cluster.

  In **Volume Properties**, select the disk size for your cluster, and configure the IOPS.

- For Google Cloud, in **Volume Type**, select **SSD Persistent Disk**.

  In **Volume Properties**, select the disk size for your cluster.

  > **Note**
  >
  > When provisioning database storage, not all of the storage space you specify is available for holding your data. Some space is reserved for other purposes. For a full explanation of the structure of a Postgres data directory, see Database File Layout. You can make more storage space available for data if you specify separate storage for WAL.

7. In the **Network, Logs, & Telemetry** section:

In **Connectivity Type**, specify whether to use private or public networking. Networking is set to **Public** by default. Public means that any client can connect to your cluster's public IP address over the internet. Optionally, you can limit traffic to your public cluster by specifying an IP allowlist, which allows access only to certain blocks of IP addresses. To limit access, add one or more classless inter-domain routing (CIDR) blocks in the **IP Allowlists** section. CIDR is a method for allocating IP addresses and IP routing to a whole network or subnet. If you have any CIDR block entries, access is limited to those IP addresses. If none are specified, all network traffic is allowed.

Private networking allows only IP addresses in your private network to connect to your cluster.

If you're using private networking and you're using BigAnimal's cloud account, you must enter your cloud service provider's account information. BigAnimal uses it to auto-approve your private endpoints and also to provide access for logging and telemetry. To enable you to create your private endpoint, BigAnimal provides you with the corresponding service name, which you can use to connect to your cluster privately.

See Cluster networking architecture for more information.

8. In the **Backups** section:

Change the default database backup retention period of 30 days using the **Retention Time** controls. You can configure the retention period to a number of days, weeks, or months. The retention period must be between 1-180 days, 1-25 weeks, or 1-6 months.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal deletes backups older than the retention period.

9. In the **Maintenance** section:

   The **Custom Maintenance Window** is enabled and set for a weekly 60-minute maintenance window in which maintenance upgrades occur for the cluster. You can change the maintenance window to a time that works best for you.

   > **Note**
   >
   > Typically, maintenance updates take only a few minutes to complete.

   For more information, see Periodic maintenance.

10. Select **Next: DB Configuration**

11. In the **Parameters** section on the **DB Configuration** tab, you can update the value of the database configuration parameters for the data group as needed.

    To update the parameter values, see Modifying your database configuration parameters.

12. Select **Create: Data Group**. The data group preview appears.

13. To finish creating your cluster, select **Create Cluster**. If you want to create a second data group, select **Add a Data Group**.

Creating a second data group

After creating the first data group, you can create a second data group for your distributed high-availability cluster by selecting **Add a Data Group** before you create the cluster.

By default, the settings for your first data group populate the second data group's settings. However, you can change certain settings if you want to. Just know that your changes can change the settings for the entire cluster. That being said, the database type and cloud provider must be consistent across both data groups.

The data groups and the witness group must all be in different regions. Otherwise, you can choose the second data group's settings as needed.

When choosing the number of data nodes for the second data group, see Distributed high availability for information on node architecture.

> **Note**
>
> To maintain high availability, BigAnimal doesn't allow the maintenance windows of data groups to overlap.

What's next

After you create your cluster, use these resources to learn about cluster use and management:

- Using your cluster
- Managing Postgres access

Related CLI commands

For information on related CLI commands, see:

- Managing clusters using the CLI
- PGD CLI on BigAnimal
- Maintenance windows CLI command

11 Managing a cluster

Pausing and resuming clusters

Pausing a cluster allows you to save on compute costs without losing data or cluster configuration settings. There's no added cost for paused clusters, although you must still pay for any disk

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

and object storage that the paused cluster requires.

Pausing a cluster drops all existing connections, blocks future actions such as editing the cluster, and stops monitoring. While the cluster is paused, you can't make or maintain connections or read, write, or edit cluster settings. Automatic backups are paused. However, backups taken while the cluster was running remain until their retention period expires.

> **Note**
>
> If you defined the backup retention period for a cluster to be less that the duration of the pause, it's possible for write-ahead logs (WALs) to be deleted from the backup object store, and resumption might fail.

While paused, clusters aren't upgraded or patched, but upgrades are applied when the cluster resumes. Pausing a high availability or Postgres Distributed cluster shuts down all cluster nodes.

After seven days, single-node and high-availability clusters automatically resume. Resuming a cluster applies any pending maintenance upgrades. Monitoring begins again.

With CLI 3.7.0 and later, you can pause and resume cluster using CLI.

You can enable in-app inbox or email notifications to get alerted when the paused cluster is or will be reactivated. For more information, see managing notifications.

Pausing a cluster

1. Go to the Clusters page in the BigAnimal portal.
2. Do one of the following:
   - In the row for the cluster, at right, from the ellipsis menu, select **Pause Cluster**.
   - Select the cluster you want to pause. From **Quick Actions** on the cluster details page, select **Pause Cluster**.
3. Confirm that you want to pause the cluster. When the process finishes, the cluster status appears as Paused. It can take up to 15 minutes for the compute instance in your CSP to scale down.

Resuming a cluster

1. Go to the Clusters page in the BigAnimal portal.
2. Do one of the following:
   - In the row for the cluster, at right, from the ellipsis menu, select **Resume Cluster**.
   - Select the cluster you want to resume. From **Quick Actions** on the cluster details page, select **Resume Cluster**.
3. Confirm that you want to resume the cluster. The process might take a few minutes. When it finishes, the cluster status appears as Healthy.

> **Note**
>
> A TDE enabled cluster, resumes only if the TDE key status is ready or available. Clusters are automatically paused if there is any issue with the TDE key. You need to resolve/give permissions to the key in your respective cloud region. Resume the cluster manually after resolving the issues.

6 Using your cluster

Account owners and contributors can connect, edit, scale, monitor, back up, and restore clusters through the BigAnimal portal.

If your organization coordinated with BigAnimal Support to enable the Apache Superset feature, see Analyzing your data with Apache Superset for information on using Apache Superset to analyze, explore, and visualize data stored in your Postgres clusters.

6.1 Managing Postgres access

You control access to your Postgres database using database authentication implemented by creating databases with specific roles and privileges. Database authentication differs from portal authentication, which controls access to the BigAnimal portal.

For information on portal authentication, see:

- Setting up your identity provider if you purchased BigAnimal directly from EDB
- Setting up your Azure Marketplace account if you purchased BigAnimal through Azure Marketplace

Setting up your database authentication

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Don't use the edb_admin database role and edb_admin database created when creating your cluster in your application. Instead, create a new database role and a new database, which provides a high level of isolation in Postgres. If multiple applications are using the same cluster, each database can also contain multiple schemas, essentially a namespace in the database. If you need strict isolation, use a dedicated cluster or dedicated database. If you don't need that strict isolation level, you can deploy a single database with multiple schemas. See Privileges in the PostgreSQL documentation to further customize ownership and roles to your requirements.

To create a new role and database, first connect using `psql`:

```
psql -W "postgres://edb_admin@xxxxxxxxx.xxxxx.biganimal.io:5432/edb_admin?sslmode=require"
```

> **Note**
>
> Avoid storing data in the postgres system database.

Admin roles

pg_ba_admin

So that we can effectively manage the cloud resources and ensure users are protected against security threats, BigAnimal provides a special administrative role, pg_ba_admin. The edb_admin user is a member of the pg_ba_admin role. The pg_ba_admin role has privileges similar to a Postgres superuser. Like the edb_admin user, the pg_ba_admin role shouldn't be used for day-to-day application operations and access to the role must be controlled carefully. See pg_ba_admin role for details.

superuser

Superuser access in BigAnimal is available only where the users are in control of their infrastructure. When using your own cloud account, you can grant the edb_admin role superuser privileges for a cluster. See Superuser access.

When granting superuser privileges, to avoid degrading service or compromising availability, ensure you limit the number of connections used by superusers. BigAnimal reserves and requires a few superuser connections for the readiness probe for these reasons:

- To check if the database is up and able to accept connections
- For creating specific roles required in PostgreSQL instances and some extensions

> **Note**
>
> Superuser privileges allow you to make Postgres configuration changes using `ALTER SYSTEM` queries. We recommend that you don't do this because it might lead to an unpredictable or unrecoverable state of the cluster. In addition, `ALTER SYSTEM` changes aren't replicated across the cluster.

For BigAnimal hosted and distributed high-availability clusters, there's no superuser access option. Use the edb_admin role for most superuser level activities. Unsafe activities aren't available to the edb_admin role.

Distributed high-availability clusters also have a bdr_superuser role. This isn't a general superuser but a specific user/role that has privileges and access to all the bdr schemas and functions. For more information, see bdr_superuser.

See the PostgreSQL documentation on superusers for best practices.

Notes on the edb_admin role

- Changes to system configuration (GUCs) made by edb_admin or other Postgres users don't persist through a reboot or maintenance. Use the BigAnimal portal to modify system configuration.
- You have to remember your edb_admin password, as EDB doesn't have access to it. If you forget it, you can set a new one in the BigAnimal portal on the Edit Cluster page.
- Don't use the edb_admin user or the edb_admin database in your applications. Instead, use `CREATE USER; GRANT; CREATE DATABASE.`
- BigAnimal stores all database-level authentication securely and directly in PostgreSQL. The `edb_admin` user password is `SCRAM-SHA-256` hashed prior to storage. This hash, even if compromised, can't be replayed by an attacker to gain access to the system.

One database with one application

For one database hosting a single application, replace `app1` with your preferred user name:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

1. Create a new database user. For example,

```
edb_admin=# create user app1 with password 'app1_pwd';
```

2. Assign the new role to your edb_admin user. Assigning this role allows you to assign ownership to the new user in the next step. For example:

```
edb_admin=# grant edb_admin to app1;
```

3. Create a new database to store application data. For example:

```
edb_admin=# create database app1 with owner app1;
```

Using this example, the username and database in your connection string is app1.

One database with multiple schemas

If you use a single database to host multiple schemas, create a database owner and then roles and schemas for each application. This example shows creating two database roles and two schemas. The default `search_path` for database roles in BigAnimal is `"$user",public`. If the role name and schema match, then objects in that schema match first, and no `search_path` changes or fully qualifying of objects are needed. The PostgreSQL documentation covers the schema search path in detail.

1. Create a database owner and new database. For example:

```
edb_admin=# create user prod_admin with password 'prod_pwd';
edb_admin=# create database prod1 with owner prod_admin;
```

2. Connect to the new database. For example:

```
edb_admin=# \c prod1
```

3. Create new application roles. For example:

```
prod1=# create user app1 with password 'app1_pwd';
prod1=# grant edb_admin to app1;

prod1=# create user app2 with password 'app2_pwd';
prod1=# grant edb_admin to app2;
```

4. Create a new schema for each application with the `AUTHORIZATION` clause for the application owner. For example:

```
prod1=# create schema app1 authorization app1;
prod1=# create schema app2 authorization app2;
```

IAM authentication for Postgres

Any user with a supported cloud account connected to a BigAnimal subscription who has the Postgres IAM role iam_aws, iam_azure, or iam_gcp can authenticate to the database using their IAM credentials.

Configuring IAM for Postgres

Provision your cluster before configuring IAM for Postgres.

1. In BigAnimal, turn on the IAM authentication feature when creating or modifying the cluster:

    1. On the **Additional Settings** tab, under **Authentication**, select **Identity and Access Management (IAM) Authentication**.
    2. Select **Create Cluster** or **Save**.

    > **Note**
    >
    > To turn on IAM authentication using the CLI, see Using IAM authentication on AWS.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

2. From your cloud provider, get the user name of each IAM user requiring database access. In the cloud account connected to BigAnimal, use Identity and Access Management (IAM) to perform user management.

3. In Postgres, if the IAM role doesn't exist yet, use the `CREATE ROLE` command. For example, for AWS, use:

   ```
   CREATE ROLE "iam_aws";
   ```

4. For each IAM user, run the `CREATE USER` Postgres command. For example, for AWS, use:

   ```
   CREATE USER "<ARN>" IN ROLE iam_aws;
   ```

   Where <ARN> is the Amazon resource name. (For Azure, use the user principal name. For GCP, use the email address.)

Logging in to Postgres using IAM credentials

If IAM integration is configured for your cluster, you can log in to Postgres using your cloud credentials. Alternatively, you can use your token instead of your password. Logging in either way allows you to connect to your Postgres database using your cloud account's IAM standard credentials.

For either method, you must first authenticate to your cloud service provider IAM to get your password or token.

> **Note**
>
> You can continue to log in using your Postgres username and password. However, doing so doesn't provide IAM authentication even if this feature is configured.

1. Get your credentials for your IAM-managed cloud account.
   - For AWS, your password is your access key (in the form <access key id>:<secret access key>). To get your access key, seeget-access-key-info To get your authorization token, see get-authorization-token.
   - For GCP, to get your access token, see Create a short-lived access token.
   - For Azure, to get your access token, see the get-access-token command.

2. Connect to Postgres using your IAM credentials.

Using IAM authentication CLI commands

For information on integrating with IAM on AWS using the CLI, seeIAM authentication CLI commands.

6.1.1 pg_ba_admin role

Learn about the operations the pg_ba_admin role can perform and its Postgres built-in roles.

What pg_ba_admin can do

A user with the pg_ba_admin role can perform many operations typically reserved for the Postgres superuser. For example, they can:

- Create roles.
- Create databases.
- Create, modify, or delete any non-superuser role, including setting passwords.
- SET ROLE to any user except a superuser without requiring a password.
- Grant pg_ba_admin membership and the corresponding powers to other roles.
- Read contents of all Postgres system catalogs in pg_catalog.
- Read and write to all user objects, that is, tables, views, and so on.
- Perform all monitoring functions on the Postgres instance. This role has the pg_monitor role.
- Install supported EDB extensions. See Supported Postgres extensions and tools.
- Bypass row-level security policies.
- Create foreign servers for foreign data wrappers. Use might be confined by network access control rules.
- Grant the pg_checkpoint role to itself and to other users

What pg_ba_admin can't do

A user with the pg_ba_admin role can't perform the following operations that a Postgres superuser can:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- Execute programs on the server. pg_ba_admin doesn't have the pg_execute_server_program role.
- Create tablespaces, casts, operator classes, access methods, and text search templates.
- Read and write files on the server. pg_ba_admin doesn't have the permissions pg_read_server_files and pg_write_server_files.
- Define LEAKPROOF functions.
- Set parameters related to pg_audit (PostgreSQL only) or edb_audit (EDB Postgres Advanced Server only).
- Create functions in non-trusted languages such as Python and C.
- Install unsupported extensions. See Supported Postgres extensions and tools.
- Use SQL to set any superuser-only configuration parameters not explicitly listed as allowed. Administrators can modify some parameters using the BigAnimal portal. SeeModifying database configuration parameters.
- Execute ALTER SYSTEM ... SET to modify the PostgreSQL configuration. Administrators can use the BigAnimal portal instead. SeeModifying database configuration parameters.
- Execute CREATE LANGUAGE with custom handlers. Administrators can use CREATE EXTENSION for supported languages.
- Execute CREATE FOREIGN DATA WRAPPER with custom handlers. Use CREATE EXTENSION for supported foreign data wrappers.
- Execute CREATE TYPE for basic types. Composite types and domains are allowed.
- Execute REINDEX system catalogs.
- Execute index maintenance operations of catalog tables.
- Execute SET SESSION AUTHORIZATION

A user with the pg_ba_admin role can't perform the following additional operations using an EDB Postgres Advanced Server distribution:

- Use portions of the DBMS package that require writing the network or the file system
- Execute CREATE USER IDENTIFIED BY
- Create resource groups
- Access the network or make use of the EDB Postgres Advanced Server http packages that access the network
- Change the encoding using DBMS_SESSION
- Make use of the EDB Postgres Advanced Server UTIL_FILE packages that access the file system
- Change qreplace_function settings
- Use the DBMS_AQ package
- Use the UTL_HTTP package
- Use the UTL_TCP package
- Use the UTL_SMTP package

Predefined roles assigned to pg_ba_admin

A user with the pg_ba_admin role has the following roles that are predefined in Postgres.

Role	Description
pg_read_all_data	Read all data (tables, views, sequences) as if having SELECT rights on those objects and USAGE rights on all schemas, even without having it explicitly. This role doesn't have the role attribute BYPASSRLS set. If RLS is being used, an administrator might want to set BYPASSRLS on roles that this role is granted to.
pg_write_all_data	Write all data (tables, views, sequences) as if having INSERT, UPDATE, and DELETE rights on those objects and USAGE rights on all schemas, even without having it explicitly. This role doesn't have the role attribute BYPASSRLS set. If RLS is being used, an administrator might want to set BYPASSRLS on roles that this role is granted to.
pg_read_all_settings	Read all configuration variables, even those normally visible only to superusers.
pg_read_all_stats	Read all pg*stat** views and use various statistics-related extensions, even those normally visible only to superusers.
pg_stat_scan_tables	Execute monitoring functions that might take ACCESS SHARE locks on tables, potentially for a long time.
pg_monitor	Read/execute various monitoring views and functions. This role is a member of pg_read_all_settings, pg_read_all_stats, and pg_stat_scan_tables.

Predefined roles not assigned to pg_ba_admin

A user with the pg_ba_admin role doesn't have, and can't be granted, the following roles that are predefined in Postgres.

Postgres predefined role	Description
pg_read_server_files	Allow reading files from any location the database can access on the server with COPY and other file-access functions.
pg_write_server_files	Allow writing to files in any location the database can access on the server with COPY and other file-access functions.
pg_execute_server_program	Allow executing programs on the database server as the user the database runs as with COPY and other functions that allow executing a server-side program.

6.2 Managing Postgres extensions

BigAnimal supports many Postgres extensions. See Postgres extensions available by deployment for the complete list.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Extensions available when using your cloud account

Installing many Postgres extensions requires superuser privileges. The table in Postgres extensions available by deployment indicates whether an extension requires superuser privileges. If you're using your cloud account, you can grant superuser privileges to edb_admin so that you can install these extensions on your cluster (see superuser).

Extensions available when using BigAnimal's cloud account

If you're using BigAnimal's cloud account, you can install and use the following extensions.

PostgreSQL contrib extensions/modules:

- auth_delay
- bloom
- btree_gin (PostgreSQL v12 only)
- btree_gist (PostgreSQL v12 only)
- citext (PostgreSQL v12 only)
- cube (PostgreSQL v12 only)
- earthdistance
- fuzzystrmatch
- hstore (PostgreSQL v12 only)
- hstore_plperl
- intagg
- intarray (PostgreSQL v12 only)
- isn (PostgreSQL v12 only)
- jsonb_plperl (PostgreSQL v12 only)
- ltree (PostgreSQL v12 only)
- old_snapshot (PostgreSQL and EDB Postgres Advanced Server v14 and v15 only)
- pageinspect
- pg_buffercache
- pg_freespacemap
- pg_prewarm
- pg_stat_statements
- pg_surgery (PostgreSQL and EDB Postgres Advanced Server v14 and v15 only)
- pg_trgm (PostgreSQL v12 and EDB Postgres Advanced Server v15 only)
- pg_visibility
- pgaudit (PostgreSQL only)
- pgcrypto (PostgreSQL v12 only)
- pglogical3 (PostgreSQL v12 and v13 only)
- pgrowlocks
- pgstattupple
- postgis
- postgres_fdw
- seg (PostgreSQL v12 only)
- sslinfo
- tablefunc (PostgreSQL v12 only)
- tcn (PostgreSQL v12 only)
- tsm_system_rows (PostgreSQL v12 only)
- tsm_system_time (PostgreSQL v12 only)
- unaccent (PostgreSQL v12 only)
- uuid-ossp (PostgreSQL v12 only)
- xml2
- spi - autoinc
- spi - refint
- spi - insert_username
- spi - moddatetime

EDB extensions:

- edb_dbo
- sql-profiler
- query_advisor
- refdata
- autocluster
- edb_pg_tuner

Installing extensions

Use the `CREATE EXTENSION` command to install most extensions. You must enable certain extensions, including the EDB Postgres Tuner (pg_tuner) extension and PostGIS on the **DB Configuration** tab of the Create or Edit Cluster page of the BigAnimal portal.

Example: Installing multiple extensions

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

This example shows one way of installing multiple extensions simultaneously.

1. Create a text file containing the `CREATE EXTENSION` command for each of the extensions you want to install. In this example, the file is named `create_extensions.sql`.

   ```
   CREATE EXTENSION <extension_name_1> SCHEMA <schema_name>;
   CREATE EXTENSION <extension_name_2> SCHEMA <schema_name>;
   CREATE EXTENSION <extension_name_3> SCHEMA <schema_name>;
   CREATE EXTENSION <extension_name_4> SCHEMA <schema_name>;
   ```

2. Use your Postgres client of choice to load the extensions. For example, using psql:

   ```
   psql <biganimal_connection_string> -f create_extensions.sql
   ```

   For more information about connecting to your cluster using a client, see Connecting to your cluster.

Working with extensions

Use the DROP EXTENSION command to remove extensions.

Use the pg_available_extensions view to see a list of all PostreSQL extensions.

The catalog_pg_extension catalog stores information about the installed extensions.

6.3 Managing replicas

Create a faraway replica

You can create faraway replicas in any active regions in your cloud. There's no limit on the number of faraway replicas each cluster can have. However, the added resources and data transfers can increase your cloud costs.

1. Go to the Clusters page. A list of previously created clusters appears.
2. Select the **Create Replica** icon next to a cluster to create its faraway replica. The Create Faraway Replica page appears.
3. You can't edit the **Cloud Provider** field. The cloud provider is inherited from the source cluster. Move to the **Cluster Settings** tab.
4. On the **Cluster Settings** tab, enter a name for your replica in the **Cluster Name** field.
5. Skip to the **Region** section and select an active region where you want to deploy your replica. You can also choose to deploy your replica in the same region as the source cluster. The password and database type values are inherited from the source cluster.
6. Select the instance type in the **Instance Type** section. See Creating a cluster for details on instance type settings.
7. Select the storage settings in the **Storage** section. See Creating a cluster for details on the storage settings.
8. In the **Networking** section, specify whether to use private or public networking. See Creating a cluster for details on the **Networking** settings.
9. On the **DB Configuration** tab, to avoid replication issues when running a replica (standby server), keep the inherited value or increase it for the following database configuration parameters:
   - `max_connections`
   - `max_locks_per_transaction`
   - `max_prepared_transactions`
   - `max_wal_senders`
   - `max_worker_processes`
10. On the **Additional Settings** tab,

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Under the **Backups** section, change the default replica backup retention period of 30 days using the **Retention Time** controls. You can configure the retention period as follows:

- 1–180 days
- 1–25 weeks
- 1–6 months

Enable the **Custom Maintenance Window** option and use the controls to set a weekly 60-minute maintenance window in which maintenance upgrades occur for the cluster. If you don't set a window, the updates are applied at EDB's discretion with prior notification.

> **Note**
>
> Typically, maintenance updates take only a few minutes to complete.

For more information, see Periodic maintenance.

Under the **Extensions** section,

- enable the **pgvector** extension to add support for vector storage and vector similarity search to Postgres.
- enable the **PostGIS** extension to add support for storing, indexing and querying geographic data.

Under the **Security** section, the **Transparent Data Encryption (TDE)** option is enabled by default only when your primary cluster is TDE-enabled. It automatically enables TDE and allows you to select the encryption key from the available List.

> **Note**
>
> TDE key material for faraway replicas must be same as the primary cluster encryption key. In case you use different key material, the cluster provisioning fails.
>
> We recommend, not to enable TDE for faraway replica cluster creation, if the source cluster is a non-TDE cluster.

11. To turn on the ability to log in to Postgres using your AWS IAM credentials, enable Identity and Access Management (IAM) Authentication. See Access.
12. Select **Create Replica**.

Modify a faraway replica

1. Sign in to the BigAnimal portal.
2. Go to the Clusters page. A list of previously created clusters appears.
3. Select the cluster with the replica you want to modify. In the **Overview** tab, you can see the cluster's replicas under **Faraway Replicas**.
4. Select the **Edit Cluster** icon next to the replica you want to modify. The Edit Faraway Replica page appears.
5. In the **Cluster Settings** tab, in the **Cluster Name** field, enter the name for your replica.
6. In the **Password** field, enter a password for your replica.

   In the **Database Type** section, you can see the Postgres type and Postgres version inherited from the source cluster. You can't edit the Postgres type and Postgres version.
7. Skip the **Region** section. The region is inherited from the source cluster and you can't modify it.
8. Select the instance type in the **Instance Type** section. See Creating a cluster for details on instance type settings.

   > **Note**
   >
   > To avoid lag during replication, ensure that your replica instance type is at least as large as the source cluster's instance type.
9. Select the storage settings in the **Storage** section. See Creating a cluster for details on the storage settings.

   > **Note**
   >
   > To avoid lag during replication, ensure that your replica storage type is at least as large as the source cluster's storage type.
10. In the **Networking** section, you specify whether to use private or public networking. See Creating a cluster for details on the **Networking** settings.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

11. On the **DB Configuration** tab, to avoid replication issues when running a replica (standby server), set the following database configuration parameters to the same or higher value than on the source cluster (primary server). Otherwise, queries aren't allowed in the replica. If you plan to increase these GUC values, we highly recommend that you set the same value for all faraway replicas first and then set the value on the source cluster. Conversely, if you plan to decrease these GUC values, we recommend setting the source cluster's value first and then the value for all faraway replicas.

    - `max_connections`
    - `max_locks_per_transaction`
    - `max_prepared_transactions`
    - `max_wal_senders`
    - `max_worker_processes`

12. Select **Save**.

Promote a faraway replica

You can promote a faraway replica to a full-fledged cluster, which makes it capable of accepting writes. The new cluster created is different from the source cluster, and it doesn't replace the source cluster.

1. Go to the Clusters page. A list of previously created clusters appears.
2. Select the cluster with the replica you want to promote. In the **Overview** tab, you can see the cluster's replicas under **Faraway Replicas**.
3. Select the **Promote Faraway Replica** icon next to the replica you want to promote. The Promote Faraway Replica page appears.
4. The cluster settings are populated with values inherited from the source cluster. You can edit the cluster settings while creating your cluster.

    > **Notes**
    >
    > - You can promote a faraway replica to a single node or high-availability cluster but not to a distributed high-availability cluster.
    > - While promoting a replica to a cluster, you can't modify the **Provider** and **Region** fields and the **Database Type** section or enable read-only workloads.

5. Select **Promote Replica**.

Delete a faraway replica

1. Sign in to the BigAnimal portal.
2. Go to the Clusters page. A list of previously created clusters appears.
3. Select the cluster with the replica you want to delete. In the **Overview** tab, you can see the cluster's replicas under **Faraway Replicas**.
4. Select the replica you want to delete.
5. Select the **Delete** icon next to the replica.
6. Confirm that you want to permanently delete the replica. The replica is deleted.

Restore a replica backup

You can restore a replica backup to a standalone cluster. A new standalone cluster is created and initialized with data from the replica backup.

1. Select the cluster with the replica you want to restore on the Clusters page in the BigAnimal portal.
2. In the **Overview** tab, you can see the cluster's replicas under **Faraway Replicas**. Select the replica.
3. From **Quick Actions**, select **Restore Replica**.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

4. On the Restore Replica page:

    1. Fill in the required fields.
    2. To restore to the last possible recovery point, in the **Source** section, in the **Point in Time Restore** field, select **Now** on the calendar. Or, to restore further back in time, choose a timestamp.
    3. Review your selections and select **Restore** to begin the restore process.

5. The new standalone cluster is now available on the Clusters page.

Using the CLI to manage faraway replicas

See Faraway replicas CLI commands for information on CLI commands for managing faraway replicas.

6.4 Connecting to your cluster

You can connect to your cluster from your cloud applications, from your client apps, and from integration points such as EDB's migration tools.

From your applications

The private networking option offers a higher level of isolation and security by moving your cluster out of the public internet. Clusters with private networking enabled by default aren't accessible from outside of your cluster's resource network. You need to perform additional configuration steps to connect your applications in other parts of your cloud infrastructure to your clusters by way of private network links.

> **Note**
>
> We strongly discourage provisioning additional resources in the cluster's resource virtual network.

See:

- Connecting from Azure
- Connecting from AWS
- Connecting from Google Cloud

From a client app

For details on connecting from psql or edb-psql, other common database drivers, and pgAdmin, see Connecting from a client app.

- psql or edb-psql
- other common database drivers
- pgAdmin

There you'll also find details on connecting read-only workloads and configuring SSL settings.

From an EDB's migration tools integration point

To connect your cluster to EDB's migration tools, see Migrating databases to BigAnimal.

See also

- Get cluster connection information using the CLI

6.4.1 Connecting from Azure

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Three different methods enable you to connect to your cluster from your application's virtual network in Azure. Each method offers different levels of accessibility and security. We recommend the Azure private endpoint method. It's the most commonly used.

Azure private endpoint (recommended)

Azure private endpoint is a network interface that securely connects a private IP address from your Azure virtual network (VNet) to an external service. You grant access only to a single cluster instead of the entire BigAnimal resource virtual network, thus ensuring maximum network isolation. Other advantages include:

- You need to configure the Private Link only once. Then you can use multiple private endpoints to connect applications from many different VNets.
- There's no risk of IP address conflicts.

Private endpoints are the same mechanism used by first-party Azure services such as CosmosDB for private VNet connectivity. For more information, seeWhat is a private endpoint?. Private Links (required by private endpoints) aren't free, however. See Azure Private Link pricing for information on the costs associated with Private Links (required by private endpoints).

> **Note**
>
> If you set up a private endpoint and want to change to a public network, you must remove the private endpoint resources before making the change.

Using BigAnimal's cloud account

When using BigAnimal's cloud account, when creating a cluster, you provide BigAnimal with your Azure subscription ID (see Networking). BigAnimal, in turn, provides you with a private link alias, which you can use to connect to your cluster privately.

1. When creating your cluster, on the **Cluster Settings** tab, in the **Network** section:
    1. Select **Private**.
    2. Enter your application's Azure subscription ID.
2. After the cluster is created, go to the cluster details to see the corresponding endpoint service name. You need the service name while creating a private endpoint.
3. Create a private endpoint in the client's VNet. The steps for creating a private endpoint in the client's VNet are the same whether you're using BigAnimal's cloud or your own. SeeStep 1: Create an Azure private endpoint and Step 2: Create an Azure Private DNS Zone for the private endpoint.
4. In your application's Azure account, select **Private Link Center**, and then select **Private endpoints**. Select the endpoint you created previously, and use the service name provided in the details section in BigAnimal to access your cluster.

Using your Azure account

Example

This example shows how to connect your cluster using Azure private endpoint.

Assume that your cluster is on a subscription called `development` and is being accessed from a Linux client VM on another subscription called `test`. It has the following properties:

- Cluster:
    - Subscription: `development`
    - Cluster ID: `p-mckwlbakq5`
    - Account ID: `brcxzr08qr7rbei1`
    - Project ID: `brcxzr08qr7rbei1`
    - Region: `Japan East`
    - Organization's domain name: `biganimal.io`
- Linux client VM called `vm-client`:
    - Subscription: `test`
    - Resource group: `rg-client`
    - Virtual network: `vnet-client`
    - Virtual network subnet: `snet-client`

Prerequisites

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

To walk through an example in your own environment, you need:

- Your cluster URL. You can find the URL in the **Connect** tab of your cluster instance in the BigAnimal portal.
- The IP address of your cluster. You can find the IP address of your cluster using the following command:

```
dig +short p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io
```

```
10.240.1.218
```

- A Postgres client, such as psql, installed on your client VM.

> **Note**
>
> BigAnimal automatically provisions an Azure Private Link service for every private Postgres cluster. You can easily find this managed Private Link service by looking for the one that has the Cluster ID in its name, like `p-mckwlbakq5-rw-internal-lb`.

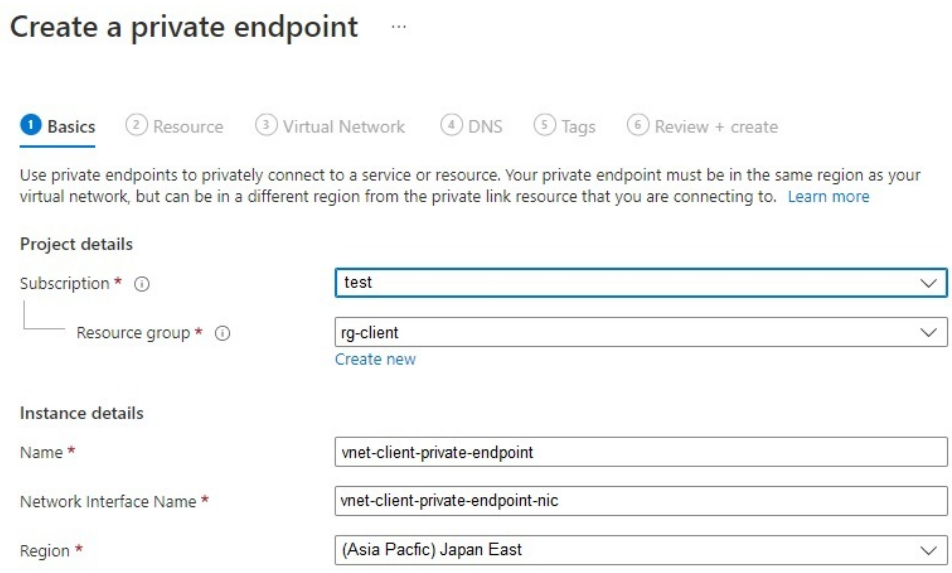
In this example, you create an Azure private endpoint in your client VM's virtual network. After you create the private endpoint, you can use its private IP address to access the Postgres cluster. You must perform this procedure for every virtual network you want to connect from.

Step 1: Create an Azure private endpoint

Create an Azure private endpoint in each client virtual network that needs to connect to your BigAnimal cluster. You can create the private endpoint using either theAzure portal or the Azure CLI.

Using the Azure portal

1. If you prefer to create the private endpoint using the Azure portal, on the upper-left side of the screen, select**Create a resource > Networking > Private Link**. Alternatively. in the search box enter `Private Link`.
2. Select **Create**.
3. In Private Link Center, select **Private endpoints** in the menu on the left.
4. In Private endpoints, select **Add**.
5. Enter the details for the private endpoint in the **Basics** tab:



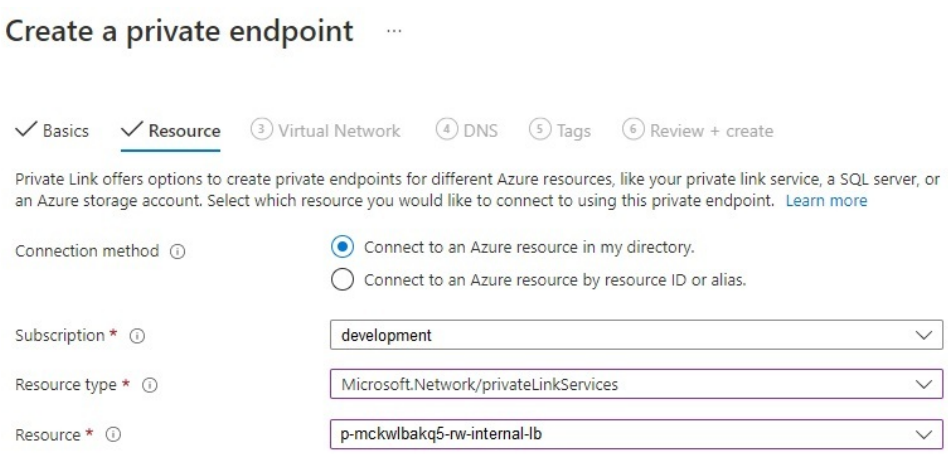
   - Subscription — Select the subscription where your vm-client resides. In this case, it's `test`.
   - Resource group — Select a resource group in the same region where your vm-client resides. This exanple uses `rg-client`.
   - Name — Use a unique name for the private endpoint. For example, enter `vnet-client-private-endpoint`, where `vnet-client` is the client VNet ID.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- Network Interface Name — This takes the name of the private endpoint and appends it with `-nic`.
- Region — The private endpoint must be in the same region as your VNet. In this case, it's `(Asia Pacific) Japan East`.

> **Note**
>
> In a later step, you need the private endpoint's name to get its private IP address.

6. On the **Resource** tab, connect the private endpoint to the Private Link service that you created by entering the following details:

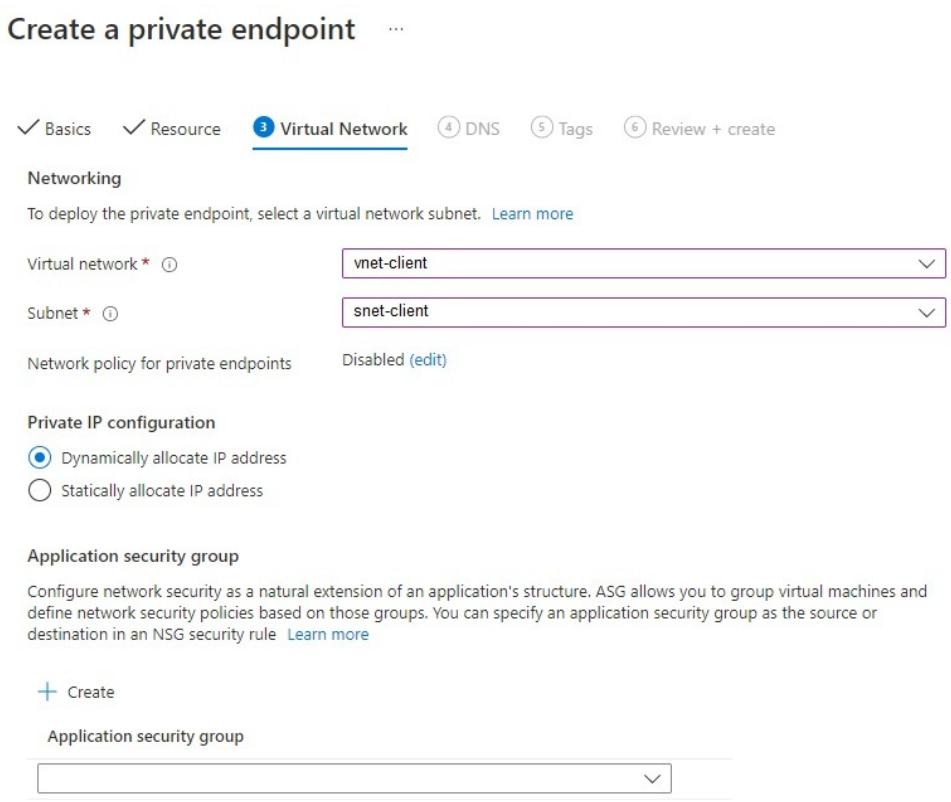


- Connection Method — Select **Connect to an Azure resource in my directory.**
- Subscription — Select the subscription in which the target BigAnimal Postgres cluster resides. In this example, it's `development`.
- Resource type — Select **Microsoft.Network/privateLinkServices**. This is the type of resource you want to connect to using this private endpoint.
- Resource — Select the Private Link service resource whose name starts with the cluster ID. In this case, it's**p-mckwlbakq5-rw-internal-lb**.

> **Note**
>
> BigAnimal creates the Private Link service in a resource group managed by Azure Kubernetes Service in the corresponding project/region. Its name follows this pattern: `MC_dp-PROJECT_ID-REGION-counter_REGION`. In this example, it's `MC_dp-brcxzr08qr7rbei1-japaneast-1_japaneast`.

7. On the **Virtual Network** tab, enter the client VM's Virtual Network details:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



- Virtual Network — Enter the VM client's virtual network. In this case, it's `vnet-client`.
- Subnet — To deploy the private endpoint, you must select a virtual network subnet to receive the private IP address assignment. In this example, the snet client subnet was already defined and will be assigned the private IP address. However, if a subnet isn't yet defined, you can select the default subnet, and a private IP address will be assigned.
- Private IP Configuration — This option defaults to **Dynamically allocate IP address**. This example uses the default.
- Application security group — You can leave this blank, or you can create or assign an Application Security Group. In this example, it's blank.

8. You can either skip or configure both **DNS** and **Tags** as you need and then go to **Review + Create**.
9. Select **Create**.
10. Proceed to Accessing the cluster.

Using the Azure CLI

If you prefer to create the private endpoint using the Azure CLI, either use your local terminal with an Azure CLI profile already configured or open a new Azure Cloud Shell using the Azure portal.

Use the following Azure CLI command to create the private endpoint by setting these parameters:

```
az network private-endpoint create \
    --connection-name p-mckwlbakq5-rw-internal-lb \
    --name vnet-client-private-endpoint \
    --private-connection-resource-id "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/mc_dp-brcxzr08qr7rbei1-
japaneast-1_japaneast/providers/Microsoft.Network/privateLinkServices/p-mckwlbakq5-rw-internal-lb" \
    --resource-group rg-client \
    --subnet snet-client \
    --vnet-name vnet-client \
    --subscription test
```

- `connection-name` needs to be the Private Link service name, like `p-mckwlbakq5-rw-internal-lb`.
- `name` is the private endpoint name, like `vnet-client-private-endpoint`.
- `private-connection-resource-id` is the Azure Resource Manager path of the Private Link Service.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- `resource-group` is the resource group in which to create the private endpoint.
- `subnet` is the Azure VNet subnet in which to create the private endpoint.
- `vnet-name` is the Azure VNet name in which to create the private endpoint.
- `subscription` is the Azure subscription in which to create the private endpoint.

Accessing the cluster

You have successfully built a tunnel between your client VM's virtual network and the cluster. You can now access the cluster from the private endpoint in your client VM. The private endpoint's private IP address is associated with an independent virtual network NIC. Get the private endpoint's private IP address using the following commands:

```
NICID=$(az network private-endpoint show -n vnet-client-private-endpoint -g rg-client --query "networkInterfaces[0].id" -o tsv)
az network nic show -n ${NICID##*/} -g rg-client --query "ipConfigurations[0].privateIPAddress" -o tsv
```

```
100.64.111.5
```

From the client VM `vm-client`, access the cluster by using the private IP address:

```
psql -h 100.64.111.5 -U edb_admin
```

```
Password for user edb_admin :

psql (13.4 (Ubuntu 13.4-1.pgdg20.04+1), server 13.4.8 (Debian 13.4.8-1+deb10))
WARNING : psql major version 13, server major version 13. Some psql features might not work.
SSL connection (protocol : TLSV1.3, cipher : TLS_AES_256_GCM_SHA384, bits : 256, compression : off) Type "help" for help.

edb_admin=>
```

Step 2: Create an Azure Private DNS Zone for the private endpoint

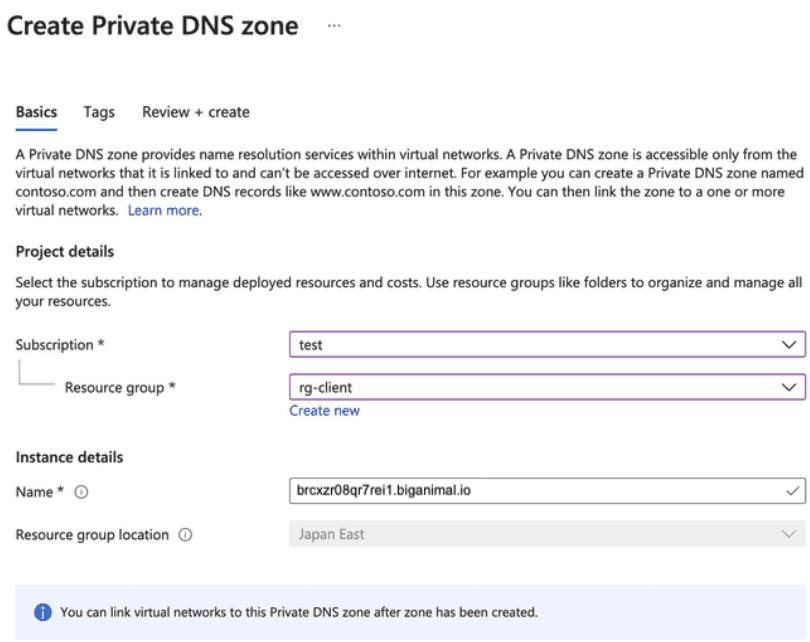
EDB strongly recommends using a private Azure DNS zone with the private endpoint to establish a connection with a cluster. You can't validate TLS certificates using `verify-full` when connecting to an IP address.

With a private DNS zone, you configure a DNS entry for your cluster's public hostname. Azure DNS ensures that all requests to that domain name from your VNet resolve to the private endpoint's IP address instead of the cluster's IP address.

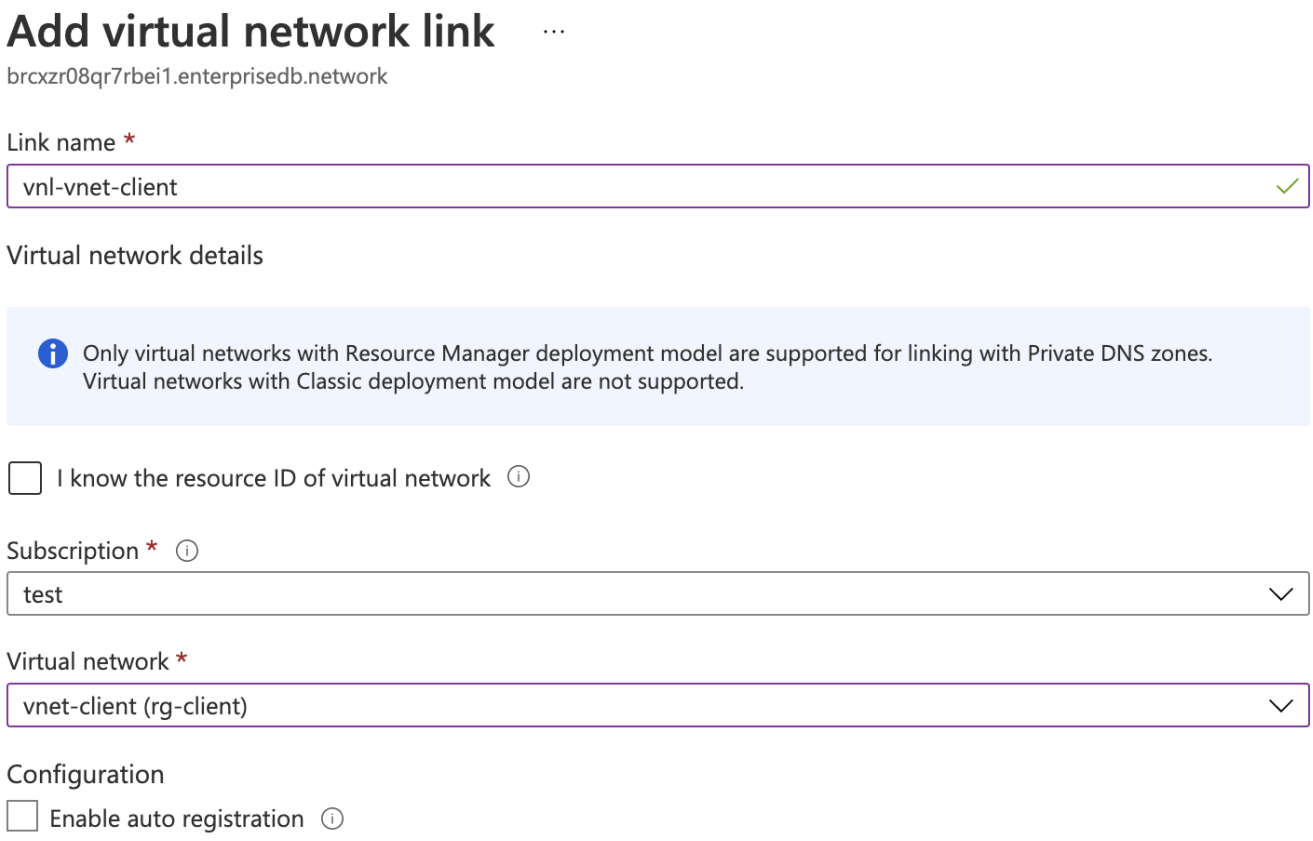
> **Note**
>
> You need to create a single private Azure DNS zone for each VNet, even if you're connecting to multiple clusters. If you already created a DNS zone for this VNet, you can skip to step 6.

1. In the Azure portal, search for `Private DNS Zones`.
2. Select **Private DNS zone**.
3. Select **Create private DNS zone**.
4. Create a private DNS zone using your organization's domain name as an apex domain. The organization's domain name must be unique to your BigAnimal organization. For example, use `biganimal.io`.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



5. Select the **Virtual network** link on the Private DNS Zone page of `brcxzr08qr7rbei1.biganimal.io`, and link the private DNS Zone to the client VM's virtual network `vnet-client`.



6. Add a record for the private endpoint. The address is a private IP address, the one created with the private endpoint in the previous step.

7. You can now access your cluster with this private domain name.

```
dig +short p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io
psql -h p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io -U edb_admin
```

```
10.240.1.123
Password for user edb_admin:

psql (13.4 (Ubuntu 13.4-1.pgdg28.84+1), server 13.4.8 (Debian 13.4.8-1+deb10))
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
WARNING : psql major version 13, server major version 13. Some psql features might not work.
SSL connection (protocol : TLSV1.3cipherTLS_AES_256_GCM_SHA384, bits : 256, compression : off) Type "help" for help.

edb_admin=>
```

> **Tip**
>
> You might need to flush your local DNS cache to resolve your domain name to the new private IP address after adding the private endpoint.

Other methods

- Virtual network peering
- Azure VNet-VNet connection

6.4.1.1 Virtual network peering example

Virtual network peering connects two Azure virtual networks, allowing traffic to be freely routed between the two. Once peered, the two virtual networks act as one with respect to connectivity. Network security group rules are still observed. Virtual network peering is simple and easy to set up, but the limitation is that the IP ranges of two peered virtual networks can't overlap. See pricing for virtual network peering for the associated cost.

Example

This example shows how to connect using virtual network peering. The IP ranges of two peered virtual networks can't overlap. BigAnimal VNets use the 10.240.0.0/14 address space and can't be peered with VNets using this same space.

> **Note**
>
> If your cluster was deployed before April 19, 2022, your BigAnimal VNet might be using the 10.0.0.0/8 address space.

Assume that your cluster is on a subscription called `development` and is being accessed from a Linux client VM on another subscription called `test`. It has the following properties:

- Cluster:
  - Subscription: `development`
  - Cluster ID: `p-mckwlbakq5`
  - Account ID: `brcxzr08qr7rbei1`
  - Organization's domain name: `biganimal.io`
- Linux client VM called `vm-client`:
  - Subscription: `test`
  - Resource group: `rg-client`
  - Virtual network: `vnet-client`
  - Virtual network subnet: `snet-client`

Prerequisites

To walk through an example in your own environment, you need:

- Your cluster URL. You can find the URL in the **Connect** tab of your cluster instance in the BigAnimal portal.
- The IP address of your cluster. You can find the IP address of your cluster using the following command:

```
dig +short p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io
```

```
10.240.1.218
```

- A PostgreSQL client, such as psql, installed on your client VM.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Step 1: Create a virtual network peering link

You need to add two peering links, one from the client VM's VNet `vnet-client` and the other from your cluster's VNet `vnet-japaneast`.

> **Note**
>
> In this example, you create virtual network peering for virtual networks that belong to subscriptions in the same Azure Active Directory tenants. For steps to create virtual network peering for virtual networks that belong to subscriptions in different Azure Active Directory tenants, see peering virtual networks from different Azure Active Directory tenants.

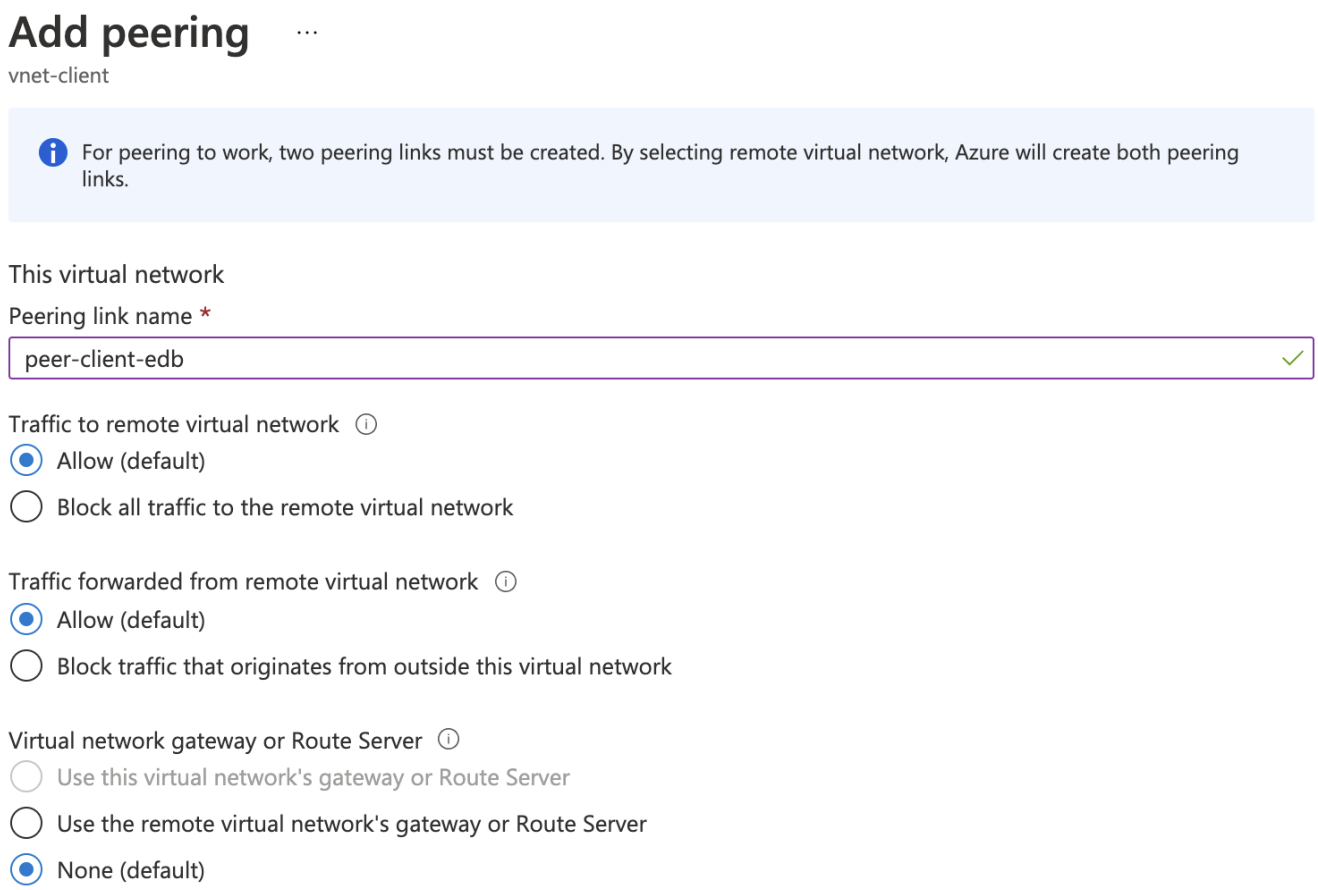
1. In the Azure portal, search for `Virtual networks`. When **Virtual networks** appears in the search results, select it. Don't select **Virtual networks (classic)**, as you can't create a peering from a virtual network deployed through the classic deployment model.

2. Select the client VM's virtual network `vnet-client` from the list that you want to create a peering for.

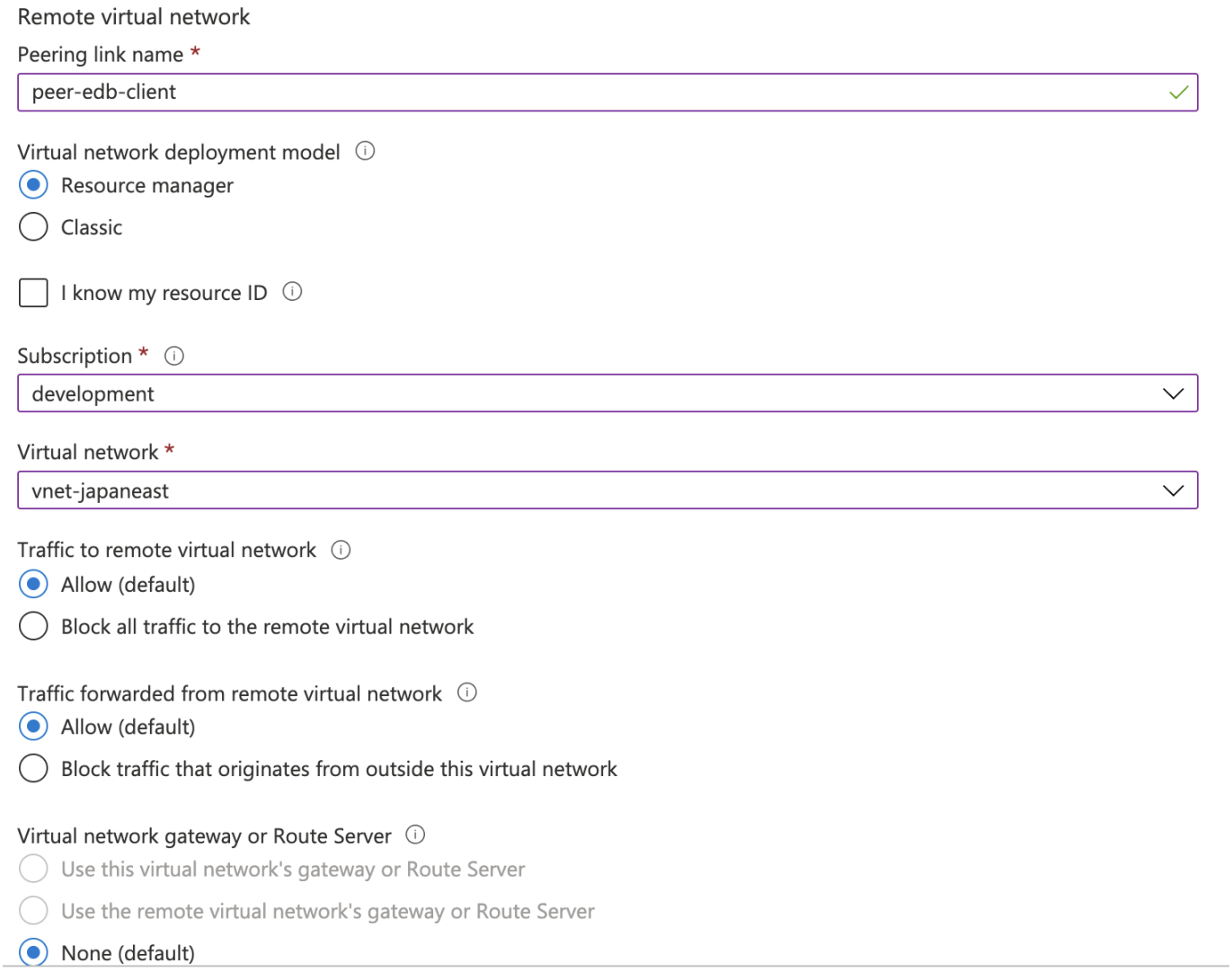
3. Under **Settings**, select **Peerings** and then select **+ Add**.

4. From the Peerings page of the client VM's virtual network `vnet-client`, add two peering links called `peer-client-edb` and `peer-edb-client` to join the address space of two virtual networks together.

   To simplify the process, Azure creates both peering links for you when you add peering from either side.



Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



Step 2: Access the cluster

Access the cluster with its domain name from your cluster's connection string. It's accessible from `vnet-client` after peering.

```
dig +short p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io
psql -h p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io -U edb_admin
```

```
10.240.1.123
Password for user edb_admin:

psql (13.4 (Ubuntu 13.4-1.pgdg28.84+1), server 13.4.8 (Debian 13.4.8-1+deb10))
WARNING : psql major version 13, server major version 13. Some psql features might not work.
SSL connection (protocol : TLSV1.3cipherTLS_AES_256_GCM_SHA384, bits : 256, compression : off) Type "help" for help.

edb_admin=>
```

6.4.1.2 VNet-VNet example

VNet-VNet connections use VPN gateways to send encrypted traffic between Azure virtual networks. Advantages include:

- Cluster domain name is directly accessible without a NAT.
- VNets from different subscriptions don't need to be associated with the same Active Directory tenant.

Cons include:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- Bandwidth is limited. See the virtual network gateway planning table.
- Configuration is complicated.
- There's an associated cost. See the virtual network gateway planning table.

Example

This example shows how to connect using VNet-VNet connections.

To use this method, you need to create Azure VPN gateways in each connected virtual network.

> **Note**
>
> VPN gateway creation can take up to 45 minutes.

Assume that your cluster is on a subscription called `development` and is being accessed from a Linux client VM on another subscription called `test`. It has the following properties:

- Cluster:
  - Subscription: `development`
  - Cluster ID: `p-mckwlbakq5`
  - Account ID: `brcxzr08qr7rbei1`
  - Organization's domain name: `biganimal.io`
- Linux client VM called `vm-client`:
  - Subscription: `test`
  - Resource group: `rg-client`
  - Virtual network: `vnet-client`
  - Virtual network subnet: `snet-client`

Prerequisites

To walk through an example in your own environment, you need:

- Your cluster URL. You can find the URL in the **Connect** tab of your cluster instance in the BigAnimal portal.
- The IP address of your cluster. You can find the IP address of your cluster using the following command:

```
dig +short p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io
```

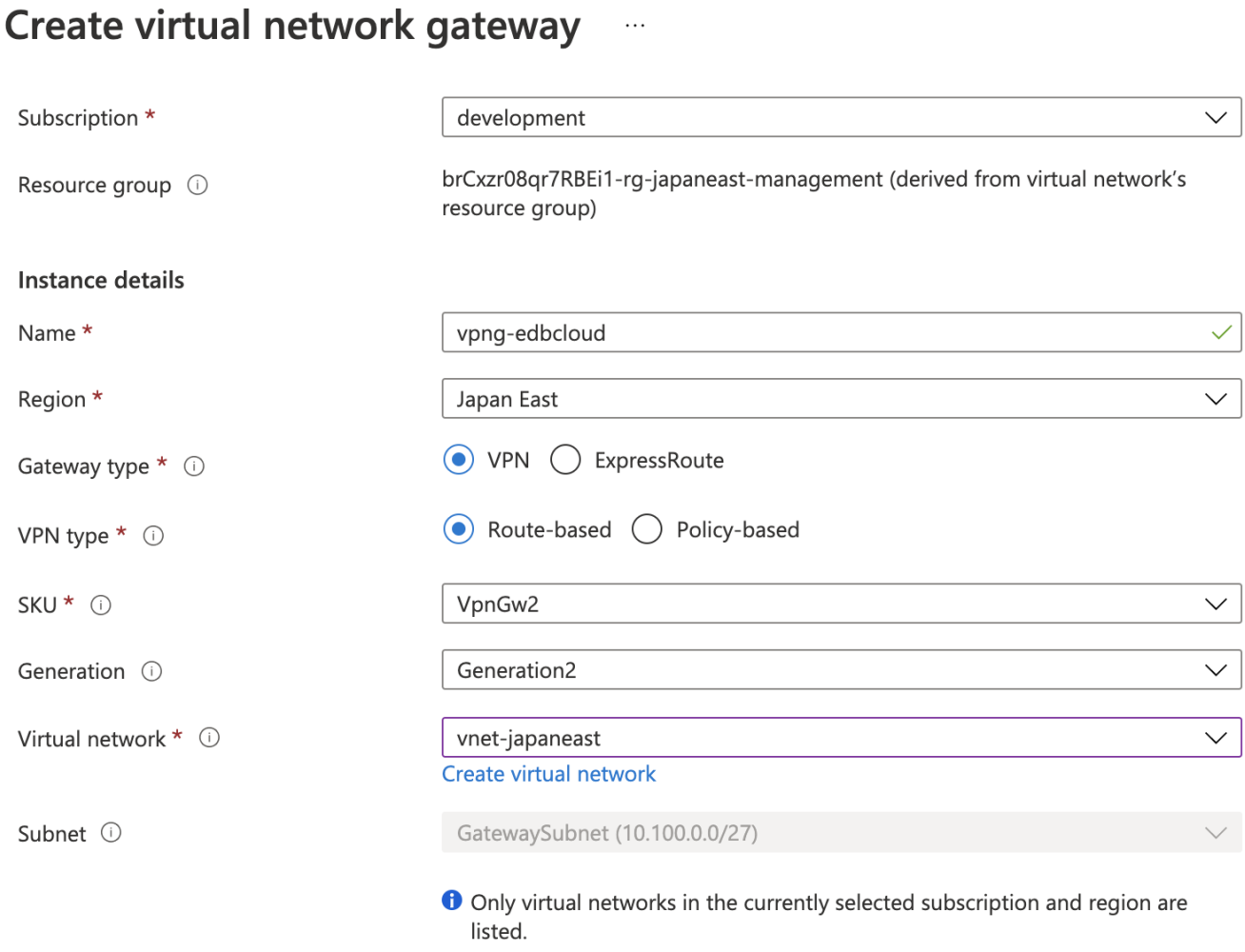
```
10.240.1.218
```

- A Postgresql client, such as psql, installed on your client VM.

Step 1: Create a VPN gateway for the cluster's virtual network

1. In the Azure portal, search for `Virtual network gateways`. Locate **Virtual network gateways** in the search results and select it.
2. On the Virtual network gateways page, select **+ Create**.
3. On the Create virtual network gateway page, create the VPN gateway for the cluster's resource virtual network `vnet-japaneast`. Name the VPN gateway `vpng-biganimal`.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



> **Note**
>
> The VPN gateway creates a dedicated subnet to accommodate its gateway VMs. Ensure that your cluster's virtual network address space has enough IP range for the subnet to prevent errors in the virtual network. For more information, see Add a subnet.

Step 2: Create a VPN gateway for the client VM virtual network

In the same way, create the gateway for your client VM virtual network `vnet-client`. Name the client VPN gateway `vpng-client`.

Step 3: Add a gateway connection between the two VPN gateways

Use the Azure CLI or PowerShell to add a VPN gateway connection from `vpng-biganimal`:

> **Note**
>
> The Azure portal allows you to create VPN gateway connections only between virtual networks belonging to the same subscription.

1. Get the VPN gateway ID of `vpng-client`.

   From the BigAnimal subscription:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
az network vnet-gateway show -n vpng-biganimal -g brCxzr08qr7RBEi1-rg-japaneast-management --query "[id]" -otsv
```

```
subscriptions/.../vpng-biganimal
```

From the client VM's subscription:

```
az network vnet-gateway show -n vpng-client -g rg-client --query "[id]" -o tsv
```

```
/subscriptions/.../vpng-client
```

2. From the BigAnimal subscription, create a connection from `vpng-biganimal` to `vpng-client`:

```
az network vpn-connection create \
    -n vpnc-biganimal-client \
    -g brCxzr08qr7RBEi1-rg-japaneast-management \
    --vnet-gateway1 /subscriptions/.../vpng-biganimal \
    -l japaneast
    --shared-key "a_very_long_and_complex_psk" \
    --vnet-gateway2 /subscriptions/.../vpng-client
```

Note the value for `--shared-key`. It is a PSK for pairing authentication from both sides needed in the next step.

3. From the client VM's subscription, create another connection from `vpng-client` to `vpng-ebdcloud`:

```
az network vpn-connection create \
    -n vpnc-client-biganimal \
    -g rg-client \
    --vnet-gateway1 /subscriptions/.../vpng-client \
    -l japaneast \
    --shared-key "a_very_long_and_complex_psk!" \
    --vnet-gateway2 /subscriptions/.../vpng-biganimal
```

Step 4: Verify the connection

1. After a few minutes, verify the gateway connection status from either virtual networks with the following command:

```
az network vpn-connection show --name vpnc-client-biganimal -g rg-client \
    --query "[connectionStatus]" -o tsv
```

```
Connected
```

2. Verify the connectivity to the cluster:

```
dig +short p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io
psql -h p-mckwlbakq5.brcxzr08qr7rbei1.biganimal.io -U edb_admin
```

```
10.240.1.123
Password for user edb_admin:

psql (13.4 (Ubuntu 13.4-1.pgdg28.84+1), server 13.4.8 (Debian 13.4.8-1+deb10))
WARNING : psql major version 13, server major version 13. Some psql features might not work.
SSL connection (protocol : TLSV1.3cipherTLS_AES_256_GCM_SHA384, bits : 256, compression : off) Type "help" for help.

edb_admin=>
```

6.4.2 Connecting from AWS

AWS VPC endpoint (AWS Private Link) service is a network interface that securely connects a private IP address from your AWS VPC to an external service. You grant access only to a single cluster instead of the entire BigAnimal resource VPC, thus ensuring maximum network isolation.

For more information, see VPC endpoint services (AWS PrivateLink).

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

The way you create a private endpoint differs when you're using your AWS account versus using BigAnimal's cloud account.

Using BigAnimal's cloud account

When using BigAnimal's cloud account, you provide BigAnimal with your AWS account ID when creating a cluster (see Networking). BigAnimal, in turn, provides you with an AWS service name, which you can use to connect to your cluster privately.

1. When creating your cluster, on the **Cluster Settings** tab, in the **Network** section:
    1. Select **Private**.
    2. Enter your application's AWS account ID using just numbers (no hyphens), for example, `123456789012`.
2. After the cluster is created, go to the cluster details to see the corresponding endpoint service name. You need the service name while creating a VPC endpoint.
3. Create a VPC endpoint in the client's VPC. The steps for creating a VPC endpoint in the client's VPC are the same whether you're using BigAnimal's cloud or your own. The steps are available here.
4. In your application's AWS account, select **VPC**, and then select **Endpoints**. Select the endpoint you created previously, and use the service name provided in the details section in BigAnimal to access your cluster.

Using your AWS account

Two different methods enable you to connect to your private cluster from your application's VPC in AWS. Each method offers different levels of accessibility and security. The VPC endpoint method is recommended and is most commonly used. However, you can also use the VPC peering connection method if required by your organization.

AWS VPC endpoint (recommended)

AWS VPC endpoint (AWS Private Link) service is a network interface that securely connects a private IP address from your AWS VPC to an external service. You grant access only to a single cluster instead of the entire BigAnimal resource VPC, thus ensuring maximum network isolation. Other advantages include:

- You need to configure the PrivateLink only once. Then you can use multiple VPC endpoints to connect applications from different VPCs.
- There's no risk of IP address conflicts.

There's an associated cost of resources, however.

For more information, see VPC endpoint services (AWS PrivateLink).

Example

This example shows how to connect your cluster using VPC endpoints.

Assume that your cluster is on an account called `development` and is being accessed from a client on another account called `test`. It has the following properties:

- BigAnimal cluster:
    - AWS account: `development`
    - Amazon resource name (ARN): `arn:aws:iam::123456789123:root`
    - Cluster ID: `p-mckwlbakq5`
    - Account ID: `brcxzr08qr7rbei1`
    - Organization's domain name: `biganimal.io`
- Client:
    - AWS account: `test`
    - Resource group: `rg-client`
    - VPC: `vpc-client`
    - VPC subnet: `snet-client`

Prerequisites

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

To walk through an example in your own environment, you need:

- Your cluster URL. You can find the URL in the **Connect** tab of your cluster instance in the BigAnimal portal.

Step 1: Create an endpoint service for your cluster

In the AWS account connected to BigAnimal, create an endpoint service to provide access to your clusters from other VPCs in other AWS accounts. Perform this procedure for each cluster to which you want to provide access.

1. Open the Amazon EC2 console. Ensure that the region where your cluster is deployed is selected in the upper-right corner of the console.
2. In the navigation pane, under **Load Balancing**, select **Load Balancers**.
3. Identify the load balancer that's tagged with the ID of the cluster to which you want to connect (`<cluster-id>-rw-internal-lb`), for example, `p-96fh28m3cb-rw-internal-lb` . Note the name of that network load balancer.
4. Open the Amazon VPC console.
5. From the navigation pane on the left, under **Virtual Private Cloud**, select **Endpoint Services**, and then select **Create endpoint service**.
6. Enter a suitable name for the endpoint service.
7. Select **Network** for the load balancer type.
8. Under **Available load balancers**, select the network load balancer of the cluster to which you want to connect.
9. Leave all the other fields with their default values, and select **Create**.
10. Under **Details**, note the **Service name** of the created endpoint service (for example, `com.amazonaws.vpce.us-east-1.vpce-svc-0e123abc123198abc`). You need the service name while creating a VPC endpoint.
11. In the navigation pane, select **Endpoint Services**.
12. Select your endpoint service from the **Actions** list, and select **Allow principals**.
13. Add the AWS account with which you want to connect to the endpoint service by specifying the ARN for the principal. The ARN must be in this format:

    arn:aws:iam::<AWS ACCOUNT ID>:root

Step 2: Create a VPC endpoint in the client's VPC

Now that your endpoint service is created, you can connect it to the cluster VPC using a VPC endpoint. Perform this procedure in your application's AWS account.

> **Note**
>
> In your application's AWS account, ensure that you allow your application's security group to connect to your cluster.

1. Open the Amazon VPC console.
2. Ensure that the region where your cluster is deployed is selected in the upper-right corner of the console.
3. From the navigation pane on the left, under **Virtual Private Cloud**, select **Endpoints**, and then select **Create endpoint**.
4. Enter a suitable name for the endpoint service.
5. Under **Service category**, select **Other endpoint services**.
6. Under **Service Name**, enter the name of the endpoint service that you created earlier:

   - If following the example using your AWS account: `com.amazonaws.vpce.us-east-1.vpce-svc-0e123abc123198abc`
   - If using BigAnimal's cloud: the service name provided in your BigAnimal cluster's details

   To verify whether you successfully allowed access to the endpoint, select **Verify service**.
7. Under VPC, select the client's VPC in which to create the endpoint.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

8. Under **Subnets**, select the subnets (availability zones) in which to create the endpoint network interfaces. Enable the endpoint in all availability zones used by your application.

9. Select **Create endpoint**.

Step 3: Accept and test the connection

1. In your AWS account connected to BigAnimal, select **VPCs**, and then select **Endpoint services**.

2. Select the endpoint service instance you created previously, and accept the endpoint connection request under **Endpoint connections**.

3. You can now successfully connect to your cluster.

   In your application's AWS account, select **VPC** and then select **Endpoints**. Select the endpoint you created previously and use the DNS name provided in the details section to access your cluster.

Other method when using your account

VPC peering

6.4.2.1 VPC peering

VPC peering allows traffic to be freely routed between two VPCs. Once peered, the two VPCs act as one with respect to connectivity. However, network security group rules are still observed. VPC peering is simple and easy to set up, but the limitation is that IP ranges of two peered VPCs can't overlap.

Example

This example shows how to connect using VPC peering.

> **Note**
>
> BigAnimal uses the 10.0.0.0/16 address space and can't be peered with VPCs using this same space. If they are, the status of the VPC peering connection immediately goes to failed.

Assume that your cluster is on an account called `development` and is being accessed from a Linux client on another account called `test`. It has the following properties:

- Cluster:
  - AWS account name: `development`
  - Cluster ID: `p-mckwlbakq5`
  - Account ID: `brcxzr08qr7rbei1`
  - Organization's domain name: `biganimal.io`
  - VPC: `vpc-cluster`
- Linux client:
  - Subscription: `test`
  - VPC: `vpc-client`
  - VPC subnet: `snet-client`

Prerequisites

To walk through an example in your own environment, you need:

- Your cluster URL. You can find the URL in the **Connect** tab of your cluster instance in the BigAnimal portal.
- A Postgresql client, such as psql, installed on your client VM.

You need to add two peering links: one from the client VPC `vpc-client` and the other from your cluster's VPC `vnet-japaneast`.

VPC peering connection with a VPC in another AWS account

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

You can create a VPC peering connection with a VPC in the same region or a different region.

Request a VPC peering connection with a VPC in another account

1. Log in to the AWS account associated with your BigAnimal account.
2. Open the Amazon VPC console.
3. In the navigation pane, select **Peering Connections**, and then select **Create Peering Connection**.
4. Enter a suitable name for the peering connection.
5. For **VPC (Requester)**, select the cluster's VPC in your account.
6. Select **Another account**.
7. Enter the AWS account ID of the owner of the accepter VPC.
8. (Optional) Select **Another region**, and then select the region in which the accepter VPC resides.
9. For **VPC (Accepter)**, enter the ID of the client VPC.
10. Select **Create Peering Connection**.
11. In the confirmation, select **OK**.
12. The VPC peering connection that you created isn't active. To activate it, the owner of the accepter VPC must accept the VPC peering connection request. To enable traffic to be directed to the peer VPC, update your VPC route table. Three route tables are created at BigAnimal VPC. You need to update all of them.

    For more information, see Update your route tables for a VPC peering connection and this FAQ article.

VPC peering connection with another VPC in your account

You can create a VPC peering connection with a VPC in the same region or a different region.

Create a VPC peering connection with a VPC in the same region

1. Open the Amazon VPC console.
2. In the navigation pane, select **Peering Connections**, and then select **Create Peering Connection**.
3. For **VPC (Requester)**, select the cluster VPC in your account.
4. Ensure **My account** is selected.
5. Select **Another VPC to peer with**. Then select **Add tag** and enter the key-value pair of the VPC you want to connect with.
6. Select **Create Peering Connection**.
7. In the confirmation, select **OK**.
8. Select the VPC peering connection that you created, select **Actions**, and then select **Accept Request**.
9. In the confirmation, select **Yes, Accept**. A second confirmation appears. Select **Modify my route tables now** to go directly to the route tables page, or select **Close** to do this later.
10. Now that your VPC peering connection is active, you must add an entry to your VPC route tables to enable traffic to be directed between the peered VPCs. For more information, see Update your route tables for a VPC peering connection. Three route tables are created at BigAnimal VPC. You need to update all of them. For more information, see this FAQ article.
11. Access the cluster with its domain name from your cluster's connection string. It's accessible from `vpc-client` after peering.

```
psql -h vpce-XXXXXXXXXXXXXXXXXXXX.eu-west-1.vpce.amazonaws.com -U edb_admin
```

```
Password for user edb_admin:
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
psql (13.4 (Ubuntu 13.4-1.pgdg28.84+1), server 13.4.8 (Debian 13.4.8-1+deb10))
WARNING : psql major version 13, server major version 13. Some psql features might not work.
SSL connection (protocol : TLSV1.3cipherTLS_AES_256_GCM_SHA384, bits : 256, compression : off) Type "help" for help.

edb_admin=>
```

Create a VPC peering connection with a VPC in a different region

1. Open the Amazon VPC console.
2. In the navigation pane, select **Peering Connections > Create Peering Connection**.
3. You can optionally name your VPC peering connection. Doing so creates a tag with a key of the name and a value that you specify.
4. Select the requester VPC in your account with which to request the VPC peering connection.
5. Ensure **My account** is selected.
6. Select **Another region**, and then select the region in which the accepter VPC resides.
7. Enter the ID of the client VPC.
8. Select **Create Peering Connection**.
9. In the confirmation, select **OK**.
10. Select the region of the accepter VPC in the upper-right corner of the AWS console.
11. In the navigation pane, select **Peering Connections**. Select the VPC peering connection that you created, select **Actions**, and then select **Accept Request**.
12. In the confirmation, select **Yes, Accept**. A second confirmation appears. Select **Modify my route tables now** to go directly to the route tables page, or select **Close** to do this later.
13. Now that your VPC peering connection is active, you must add an entry to your VPC route tables to enable traffic to be directed between the peered VPCs. For more information, see Update your route tables for a VPC peering connection. Three route tables are created at BigAnimal VPC. You must update all of them. For more information, see this FAQ article.
14. Access the cluster with its domain name from your cluster's connection string. It's accessible from `vpc-client` after peering.

```
psql -h vpce-XXXXXXXXXXXXXXXXXXXX.eu-west-1.vpce.amazonaws.com -U edb_admin
```

```
Password for user edb_admin:

psql (13.4 (Ubuntu 13.4-1.pgdg28.84+1), server 13.4.8 (Debian 13.4.8-1+deb10))
WARNING : psql major version 13, server major version 13. Some psql features might not work.
SSL connection (protocol : TLSV1.3cipherTLS_AES_256_GCM_SHA384, bits : 256, compression : off) Type "help" for help.

edb_admin=>
```

6.4.3 Connecting from Google Cloud

The way you create a private Google Cloud endpoint differs when you're using your Google Cloud account versus using BigAnimal's cloud account.

Using BigAnimal's cloud account

When using BigAnimal's cloud account, when creating a cluster, you provide BigAnimal with your Google Cloud project ID (see Networking). BigAnimal, in turn, provides you with a Google Cloud service attachment, which you can use to connect to your cluster privately.

1. When creating your cluster, on the **Cluster Settings** tab, in the **Network** section:
    1. Select **Private**.
    2. Enter your application's Google Cloud project ID.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

2. After the cluster is created, go to the cluster details to see the corresponding service attachment. You need the service attachment while creating a PSC-connected endpoint.

3. Create a connected endpoint in the client's VPC. The steps for creating a connected endpoint in the client's VPC are the same whether you're using BigAnimal's cloud or your cloud. See Step 2: Create a connected endpoint for the VM client/application.

4. In your application's Google Cloud, select **Private Service Connect**, and then select **Connected Endpoints**. Select the endpoint you created previously, and use the service attachment provided in the details section in BigAnimal to access your cluster.

Using your Google Cloud account

Two different methods enable you to connect to your private cluster from your application's VPC in Google Cloud. Each method offers different levels of accessibility and security.

- You can use Google Cloud Private Service Connect (PSC) to publish services using internal IP addresses in your VPC network. PSC is a network interface that securely connects a private IP address from your Google Cloud VPC to an external service. You grant access only to a single cluster instead of the entire BigAnimal resource VPC, thus ensuring maximum network isolation. We refer to this process of connecting as using PSC-connected endpoints.

- We recommend the PSC-connected endpoint method, which is most commonly used and is used in the example. However, if required by your organization, you can also use the VPC peering connection method.

PSC-connected endpoint example

This example shows how to connect your cluster using PSC-connected endpoints.

Assume that your cluster is in a project called `development` and is being accessed from a client in another project called `test`. It has the following properties:

- BigAnimal cluster:
  - Google Cloud Project Project: `development`
  - Google Cloud Project ID: `development-001`
  - BigAnimal Cluster ID: `p-mckwlbakq5`
  - Region where BigAnimal cluster is deployed: `us-central1`
  - BigAnimal Organization ID: `brcxzr08qr7rbei1`
  - Organization's domain name: `biganimal.io`
  - Host Name: `p-mckwlbakq5.private.brcxzr08qr7rbei1.biganimal.io`

- VM Client:
  - Google Cloud Project Name: `test`
  - Google Cloud Project ID: `test-001`
  - VM Client/App: `test-app-1`
  - VM Client's VPC: `client-app-vpc`
  - VM Client's Subnet: `client-app-subnet`

Prerequisites

To walk through an example in your own environment, you need a:

- BigAnimal Postgres cluster deployed with private connectivity.
- VM with a client/application installed in your Google Cloud project.
- Subnet in the VM's VPC in the same region as the BigAnimal cluster.

Step 1: Publish a service from BigAnimal

> **Note**
>
> Publish a service from BigAnimal in the Google Cloud project connected to your BigAnimal subscription.

In the Google Cloud project connected to BigAnimal, to provide access to your cluster from other VPCs in other Google Cloud projects, create a PSC published service. Publish a service from BigAnimal for each Postgres cluster to which you want to provide access.

1. Get the hostname of your Postgres cluster from the **Connect** tab of the Cluster page on the BigAnimal portal (`P-mckwlbakq5.private.brcxzr08qr7rbei1.biganimal.io`).

2. Using Cloudshell, the command prompt, or some other terminal, get the internal IP address of the host by performing a ping, nslookup, or dig +short <host> against the hostname (`10.247.200.9`).

3. In the Google Cloud portal, go to **Network Services > Load balancing**.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

4. In the Filter area, under **Load Balancers**, select **Addresses** and filter for the host IP (`10.247.200.9`). Note the load balancer name (`a58262cd80b234a3aa917b719e69843f`).
5. Go to **Private Service Connect > Published Services**.
6. Select **+ Publish Service**.
    1. Under **Load Balancer Type**:
        1. Select **Internal passthrough Network Load Balancer**.
        2. In the **Internal load balancer** field, paste the load balancer name (`a58262cd80b234a3aa917b719e69843f`).
    2. For **Service Name**, enter the published service a name (`p-mckwlbakq5`).
    3. For **Subnets**, select **Reserve New Subnet**.
7. In the Reserve subnet for Private Service Connect window, enter the following details, and then select **Add**.
    1. For **Name**, use the name of the Postgres cluster (`p-mckwlbakq5`).
    2. For **IPv4 range**, assign the CIDR for the field IPv4 range, for example, `10.247.214.0/29` .

        > **Recommendations for IP range**
        >
        > - Allocate at least 8 IP addresses to the CIDR. The subnet mask must not be greater than 29.
        > - Avoid overlap with other reserved IP ranges by not allocating too many IP addresses at one time.
        > - If you encounter the error "This IPv4 address range overlaps with a subnet you already added. Enter an address range that doesn't overlap.", use another CIDR block until no error is returned.
8. (Optional) To accept connections automatically, add the consumer (where the client app resides) Google Cloud project ID (`test-001`).
9. Select **Add Service** and get the name of the service attachment. You might need to select the newly created published service to find the name of the service attachment. (`projects/development-001/regions/us-central1/serviceAttachments/p-mckwlbakq5`).

Step 2: Create a connected endpoint for the VM client/application

> **Note**
>
> Create a connected endpoint for the VM client/application in the Google Cloud project where your VM client/application resides.

1. From the Google Cloud console, switch over to the project where your VM client/application resides (`test`).
2. To get the VPC of your VM (`client-app-vpc`), go to **Compute Engine > VM Instances**. Under **Network Interface**, note the network information.
3. To create an endpoint with the VPC, go to **Network Services > Private Service Connect**. Under **Connected Endpoints**, select **+ Connect Endpoint**.
    1. For the target, select **Published service**, and use the service attachment captured earlier (`projects/development-001/regions/us-central1/serviceAttachments/p-mckwlbakq5`).
    2. For the endpoint name, use the name of your VM client/application (`test-app-1`).
    3. For the network (VPC), use the name of your VM client's VPC (`client-app-vpc`).
    4. For the subnetwork, use your VM client's subnet (`client-app-subnet`).

        > **Note**
        >
        > If no subnet is available, create a subnet in the VPC for the region where your Postgres cluster was created as shown in this knowledge base article.
    5. For the IP address, create an IP address, or choose an existing IP that isn't used by the other endpoints.
    6. Enable **Global Access**.

        > **Note**
        >
        > If your VM is running in a different region from BigAnimal, then always enable **Global Access**.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

4. Select **Add Endpoint**.

5. Check to see if the endpoint status is Accepted, and obtain the IP address.

> **Note**
>
> If the endpoint status is Pending, see this knowledge base article.

6. Connect to your BigAnimal cluster from your client application using the endpoint IP address (for example, `psql "postgres://edb_admin@<endpoint IP>:5432/edb_admin?sslmode=require"`).

Step 3: (Optional) Set up a private DNS zone

Setting up a private DNS zone in your Google Cloud project allows you to connect BigAnimal with the host. For instructions on setting up a private DNS zone, seethis knowledge base article.

6.4.3.1 VPC peering

VPC peering allows traffic to be freely routed between two VPCs. Once peered, the two VPCs act as one with respect to connectivity. However, network security group rules are still observed. VPC peering is simple and easy to set up, but the limitation is that IP ranges of two peered VPCs can't overlap.

For an example of VPC peering see Connect to BigAnimal private cluster using GCP VPC peering for step-by-step instructions in the knowledgebase.

6.4.4 Connecting from a client app

You can connect to your cluster using the client of your choice including:

- `psql` — Terminal-based client for Postgres
- Other common database drivers.
- pgAdmin — Desktop or web UI client to inspect, monitor, manage, and query your cluster's databases.

Recommended settings for SSL mode

Different clients can have different default TLS/SSL modes (sslmode). For example, `psql` defaults to `prefer`, which means the client attempts to establish a TLS connection but falls back to non-TLS if the server doesn't support it. In the `psql` example provided by EDB in the **Quick Connect** field, `sslmode` is explicitly set to `require`, which means the client attempts a TLS connection and fails if the connection to the server can't be encrypted.

For public connections and in most environments, EDB recommends setting `sslmode` to `verify-full`. This setting ensures that you connect to the server you specified and that the connection is encrypted.

BigAnimal generates certificates with LetsEncrypt, a widely trusted certificate authority. Your client machine might already have a bundled CA certificate for LetsEncrypt, for example, at `/etc/ssl/certs/ca-certificates.crt` or `/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem`. If it doesn't, your client machine needs aCA certificate for Let's Encrypt. Once the CA certificate is in place on your client machine, configure the `sslrootcert` parameter to its location, and set the `sslmode` parameter to `verify-full` to verify the certificate to fully validate the connection.

To view the encryption protocol being used for communication, connect to the cluster using `psql` and use the `conninfo` meta-command. In the case of BigAnimal, TLS (v1.2+) is supported:

```
edb_admin=> \conninfo
You are connected to database "edb_admin" as user "edb_admin" on host "xxxxxxxxx.xxxxx.biganimal.io" at port "5432".
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
```

Connect to your cluster using `psql` or `edb-psql`

> **Note**
>
> When connecting to an EDB Postgres Advanced Server database, use `edb-psql`. `edb-psql` is a symbolic link to a binary called `psql`, a modified version of the PostgreSQL community `psql` with added support for EDB Postgres Advanced Server features.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

1. Sign in to the BigAnimal portal.
2. Go to the Clusters page.
3. Select the name of your cluster.
4. On the **Overview** tab, select the copy icon to the right of the**Quick Connect** field to copy the command for connecting to your cluster using `psql` to your clipboard. `psql` prompts for the edb_admin user password you selected when you created the cluster.
5. Paste the command in your terminal.

Connect to your cluster using common database drivers

1. Sign in to the BigAnimal portal.
2. Go to the Clusters page.
3. Select the name of your cluster.
4. Select the **Connect** tab. You can review and copy all the relevant information you need from this screen except for the edb_admin user password.

   Connection string examples for common database drivers using the recommended setting for SSL mode (consult the client driver documentation for more information):

   - **libpq (psql or edb-psql)**

     ```
     psql -W "postgres://edb_admin@xxxxxxxxx.xxxxx.biganimal.io:5432/edb_admin?sslmode=verify-full&sslrootcert=/usr/share/ca-certificates/ca-cert_name.pem"
     ```

   - **JDBC for Java**

     ```
     jdbc:postgresql://xxxxxxxxx.xxxxx.biganimal.io:5432/edb_admin?&sslrootcert=/usr/share/ca-certificates/ca-cert_name.pem&sslmode=verify-full?user=edb_admin&password=$PWD
     ```

   - **Npgsql for DotNet Core**

     ```
     Host=xxxxxxxxx.xxxxx.biganimal.io;Port=5432;Username=edb_admin;Password=$PWD;Database=edb_admin;SslRootCert=/usr/share/ca-certificates/ca-cert_name.pem;SslMode=verify-full
     ```

   - **ODBC for Windows**

     ```
     Driver={PostgreSQL};Server=xxxxxxxxx.xxxxx.biganimal.io;Port=5432;Database=myDataBase;Uid=edb_admin;Pwd=$PWD;sslrootcert=C:\\ssl\\ca-certificate.pem;sslmode=verify-full;
     ```

Connect to your cluster using pgAdmin

To connect to your BigAnimal cluster from pgAdmin, you need to enter your cluster values into pgAdmin. Keep BigAnimal and pgAdmin open to copy and paste the values.

Navigate to the location of the values:

1. Sign in to the BigAnimal portal.
2. Go to the Clusters page.
3. Select the cluster you want to connect to.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

4. Select the **Connect** tab.

Enter the values in pgAdmin:

1. Open pgAdmin.
2. In the **Quick Links** panel, select **Add New Server**.
3. In the Create-Server dialog box, in the **General** tab, enter a server name in the **Name** field.
4. Select the **Connection** tab.
5. Copy the corresponding values from BigAnimal and paste them into pgAdmin.

BigAnimal field	pgAdmin field	Example value
Host	Host name/address	p-n85scw2ies.fcrziuxgkqazmhkl.s.edbcloud.io
Port	Port	5432
Dbname	Maintenance database	edb_admin

6. In the **Username** and **Password** fields, enter the cluster's administrator credentials. These are the same credentials you set when configuring the cluster. If you didn't set the username, then copy and paste the default administrator username from the **User** field in BigAnimal.
7. In the **SSL** tab, set the **SSL mode** field to **Require**.
8. Select **Save**. pgAdmin attempts to connect to the BigAnimal cluster.

Connect a read-only workload

Clusters that are read-only enabled display separate connection strings for **Read-only** and **Read/write** access.

To connect to your cluster with the read-only workload:

1. Sign in to the BigAnimal portal.
2. Go to Clusters.
3. Select the cluster you want to connect to.
4. Select the **Connect** tab.
5. Copy the **Read-only Service URI**.
6. Use the **Read-only Service URI** to connect to the cluster according to your database driver.

We recommend that your applications run `SET default_transaction_read_only = true` for all read-only activity. For a libpq driver, add `options='-c default_transaction_read_only=true'` to the database connection string. Other drivers use similiar connection parameters. Also, applications can run `BEGIN TRANSACTION READ ONLY` to mark individual transactions as read-only.

> **Note**
>
> On read-only workloads, BigAnimal might cancel long-running queries and display an error message like the following:
>
> ```
> ERROR: canceling statement due to conflict with recovery
> ```

For more information on querying standby replicas, see the PostgreSQL Hot Standby documentation.

6.5 Modifying your cluster

You can modify your cluster by modifying its configuration settings.

You can also modify your cluster by installing Postgres extensions. See Postgres extensions for more information.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Modify your cluster's configuration settings

1. Sign in to the BigAnimal portal.
2. From the Clusters page, select the name of the cluster you want to edit.
3. From the top-right corner of the **Cluster Info** panel, select **Edit Cluster**.
4. You can modify the following settings on the corresponding tab of the Edit Cluster page.

> **Note**
>
> Any changes made to the cluster's instance type, volume type, or volume properties aren't automatically applied to replica settings. To avoid lag during replication, ensure that replica instance and storage types are at least as large as the source cluster's instance and storage types. See Modify a faraway replica.

Settings	Tab	Notes
Cluster type	**Cluster Info**	You can't switch from a single-node cluster or a high-availability cluster to a distributed high-availability cluster or vice versa.
Number of replicas (for a high-availability cluster)	**Cluster Info**	—
Cluster name and password	**Cluster Settings**	—
Instance type	**Cluster Settings**	Changing the instance type can incur higher cloud infrastructure charges.
Volume type	**Cluster Settings**	You can't switch between the io2 and io2 Block Express volume types in an AWS cluster.
Volume properties	**Cluster Settings**	It can take up to six hours to tune IOPS or resize the disks of your cluster because AWS requires a cooldown period after volume modifications, as explained in Limitations. The volume properties are disabled and can't be modified while this is in progress.
Networking type (public or private)	**Cluster Settings**	If you're using Azure and previously set up a private link and want to change to a public network, you must remove the private link resources before making the change.
Nodes (for a distributed high-availability cluster)	**Data Groups**	After you create your cluster, you can't change the number of data nodes.
Database configuration parameters	**DB Configuration**	If you're using faraway replicas, only a small subset of parameters are editable. These parameters need to be modified in the replica when increased in the replica's source cluster. See Modify a faraway replica for details.
Retention period for backups	**Additional Settings**	—
Custom maintenance window	**Additional Settings**	Set or modify a maintenance window in which maintenance upgrades occur for the cluster. See Maintenance.
Read-only workloads	**Additional Settings**	Enabling read-only workloads can incur higher cloud infrastructure charges.
PgBouncer	**Additional Settings**	Enabling PgBouncer incurs additional infrastructure costs that depend on your cloud provider. See PgBouncer costs.
Identity and Access Management (IAM) Authentication	**Additional Settings**	Turn on the ability to log in to Postgres using AWS IAM credentials. You must then run a command to add each user's credentials to a role that uses IAM authentication in Postgres. See IAM authentication for Postgres.
Superuser access	**Additional Settings**	Disabling the option removes superuser access for edb_admin, but any other superusers existing in the database retain their superuser privileges.

5. Save your changes.

> **Note**
>
> Saving changes might require restarting the database.

Modify a data group

You can modify the data groups in your distributed high-availability cluster by editing the configuration settings.

1. Sign in to the BigAnimal portal.
2. On the Clusters page, select the data group you want to edit. Data groups appear under the cluster they reside in.
3. Select **Edit** next to the data group.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

4. Edit the cluster settings in the **Data Groups** tab. See the table in Modify your cluster configuration settings.

5. Select **Save**.

6. In the popup window, confirm your changes.

6.5.1 Modifying database configuration parameters

The database parameters listed on the **DB Configuration** tab are also referred to as Grand Unified Configuration (GUC) variables. See What Is a GUC Variable? for more information.

The list of parameters is populated based on the type of database you selected on the **Operational Settings** tab when you created your cluster. For more information about the parameters for your database type:

- For PostgreSQL parameters, see Setting Parameters and Server Configuration in the Postgres documentation.
- For EDB Postgres Advanced Server, see Summary of configuration parameters and Configuration parameters.
- For more information on the types of values parameters accept, see Parameter Names and Values. The types apply to both PostgreSQL and EDB Postgres Advanced Server parameters.

> **Note**
>
> Not all database configuration parameters are supported by BigAnimal. Some parameters, such as `wal_level` and `restore_command`, are reserved for EDB to provide the managed database features of BigAnimal.

Using formulas for parameter values

In addition to entering specific values for parameters, for some parameters you can specify formulas to calculate a value. You can use formulas for parameters of type integer and real in ternary formulas, such as the shared buffer example, using the following operators: `+ - / * > >= < <= == != && || ! ? : ( )`. Use `?` and `:`. Use `( )` to specify order of operations, if needed. GUCs used in formulas must also be of type integer or real. All arithmetic is done on 64-bit floating point values rounded to an integer result if the target GUC is of type integer and not real.

BigAnimal has what we refer to as *psuedo GUCs* to help with creating equations. These read-only GUCs are:

- `cpu_cores` — Number of cores provided by the VM running the cluster nodes.
- `disk_iops` — Number of IOPS allocated to the disks backing the primary data volume.
- `disk_size` — Amount of memory backing the primary data volume, in KB.
- `ram` — Amount of memory provided by the VM running the cluster nodes, in KB.
- `replica_count` — Total number of nodes in the cluster.

Formulas can calculate a value relative to the specifications you selected for the cluster or relative to other non-pseudo GUCs. If you resize your cluster, BigAnimal recalculates formulas using the new values for the pseudo GUCs.

Order of operations

`*` and `/` are at the same operator precedence. `+` and `-` are at the same operator precedence. `*` and `/` are evaluated before `+` and `-`. Use parentheses to specify additional order of operations.

Units

GUCs can either be a bare number, such as 8, or have a unit, such as GB. Units are categorized into kinds of units, such as size or duration.

The kind of unit of the target GUC must match the kind of unit used in the formula. For example, you can't use a formula that tries to multiply two seconds with three kilobytes (`2s * 3KB`), but you can use a formula that multiplies two sizes even if they have different scales (`2KB * 8GB`).

The unit of the target GUC must match the unit used in the formula. For example, you can multiply two seconds by four days but you can't assign the result to a GUC of unit KB.

If a target GUC has a unit, a bare number in an equation is assumed to be in the default unit for the target GUC.

Recursion

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

We recommend no more than approximately 100 levels of recursion depth.

Examples

Instead of entering a specific value for the `shared_buffers` parameter, with GUC equations you can specify that the value for `shared_buffers` is a certain percentage of RAM but no higher than a certain size. For example:

```
((ram * 0.25) > 8 GB) ? 8 GB : (ram * 0.25)
```

What if you want `wal_buffers` to be 3% of `shared_buffers`, up to a maximum of 16MB? The formula is:

```
((shared_buffers * 0.03) > 16MB) ? 16MB : (shared_buffers * 0.03)
```

The formula refers to the `shared_buffers`, which, in turn, has its own formula in this example. This example shows how GUCs can refer to other GUCs.

Common errors and error handling

Malformed formulas result in an error. Here are some examples:

- Following a number with two units, for example:

  ```
  work_mem = 10 GB kB
  ```

  This syntax gives the error:

  ```
  unexpected token "kB" at end of input (line 1, col 7)
  ```

- Creating an infinite evaluation loop by having a GUC referring to itself, for example:

  ```
  effective_cache_size = effective_cache_size
  ```

  This syntax gives the error:

  ```
  cycle in equation; GUC "effective_cache_size" already seen
  ```

- Mixing unit types, for example:

  ```
  work_mem = 10kB * 8s
  ```

  This syntax gives the error:

  ```
  both size and time units used in calculation; cannot proceed
  ```

Modify a parameter

1. Go to the parameter you want to modify using these methods:

   - To search for a specific parameter, use the search field.
   - To filter the parameters that show, select one of the following in the**Show only** list:
     - **Custom Values** — Shows only parameters that changed from the default values, either in the current session or modified in a previous session and applied to the cluster.
     - **Currently Edited Values** — Shows only parameters that changed during the current editing session.

2. Enter the new value in the parameter value field.

   > **Danger**
   >
   > Parameters identified with a yellow exclamation point icon trigger a restart when you save your changes. The restart terminates all open connections and takes a few minutes to complete. It takes a few more seconds for the new connection to establish. During this process, it isn't possible to connect to the database.

3. Save your changes.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

6.6 Backing up and restoring

Backups

BigAnimal backs up the data in your PostgreSQL clusters. You can change the retention period on theBackups tab when you create or edit your cluster. Depending on the cloud your cluster is deployed on, BigAnimal uses either of the following cloud object storage solutions.

Cloud	Object storage solution
AWS	Amazon S3 (standard tier)
Azure	Azure Blob Storage (in the "hot" access tier with geo-zone-redundant storage (GZRS))
Google Cloud	Cloud Storage

Your organization is responsible for the charges associated with the cloud object storage solution.

PostgreSQL clusters in BigAnimal are continuously backed up through a combination of base backups and transaction log (WAL) archiving. When a new cluster is created, an initial "base" backup is taken. After that, every time a WAL file is closed, which is, by default, up to every 5 minutes, it's uploaded to the cloud object storage solution. If your cluster has faraway replicas, BigAnimal copies the WAL files from your cloud object storage solution and asynchronously transfers them to the faraway replicas. Your organization is responsible for the charges associated with the cloud object storage solution.

You can change the retention period on theAdditional Settings tab when you create or edit your cluster.

Replication lag with faraway replicas

With faraway replicas, the primary server writes to the archive, which is moved to the object store. The replica reads from the object store as files arrive. Prior to the archive replicating across to the replica, the log file must write at least 16MB (the WAL segment size) of data to the WAL to cut a new archive file. If the time interval for closing a WAL file is too large, it can introduce a delay until the replica receives the latest record. The amount of time that it takes to fill up the 16MB log file and copy it to the archive is the replication lag. The replication lag is the period of time when data can be lost. It can be minutes or hours, depending on the amount of activity in your database. You need to measure the replication lag to determine if it's acceptable from a possible data-loss perspective. If it isn't acceptable, consider using a distributed high-availability cluster.

To determine the replication lag, you can compare the last log sequence number (LSN) in the 16MB log file on the primary to the LSN in the log file in the replica.

Restores

If a restore is necessary—for example, in case of an accidental `DROP TABLE` statement—you can restore clusters to any point in the backup retention period.

Cluster restores aren't performed in place on an existing cluster. Instead, a new cluster is created and initialized with data from the backup archive. Restores must replay the transaction logs between the most recent full database backup and the target restore point. Thus restore times (that is, RTO) depend on the write activity in the source cluster.

BigAnimal retains backups for 30 days by default. You can restore backups into a new cluster in any region supported by BigAnimal.

Perform a cluster restore

The restore operation is available for any cluster that has at least one available backup. For newly created clusters, the initial backup becomes available a few minutes after the new cluster is fully operational.

Restore a single-node or primary/standby high-availability cluster

1. Select the cluster you want to restore on the Clusters page in the BigAnimal portal.
2. From **Quick Actions**, select **Restore**.
3. On the Restore Cluster page:
    1. Fill in the required fields.
    2. To restore to the last possible recovery point, in the **Source** section, in the **Point in Time Restore** field, select **Now** on the calendar. Or, to restore further back in time, choose a timestamp.
    3. Review your selections in **Cluster Summary** and select **Restore Cluster** to begin the restore process.
4. The new cluster is now available on the Clusters page.

Restore a distributed high-availability cluster

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

1. On the Clusters page in the BigAnimal portal, select the cluster you want to restore.
2. Select **Quick Actions > Restore**.
3. In the **Cluster Settings** tab, enter a cluster name and password for your restored cluster.
4. Select **Next: Data Group**.
5. Select the **Node Settings** tab.
6. In the **Source** section, select **Fully Restore** or **Point in Time Restore**. A point-in-time restore restores the data group as it was at the specified date and time.
7. In the **Nodes** section, select **Two Data Nodes** or **Three Data Nodes**. For more information on node architecture, see Distributed high availability.
8. Follow Steps 3-5 in Creating a distributed high-availability cluster.
9. Select **Restore**.

6.7 Monitoring and logging

With BigAnimal, you have a few options for monitoring and logging solutions:

- If you're using your own cloud, the default observability solution is viewing metrics and logs from your cloud service provider. For more information about viewing metrics and logs from your cloud provider, see:
    - Viewing metrics and logs from Azure
    - Viewing metrics and logs from AWS

  For details on the types and format of metrics available from BigAnimal in your cloud provider, see Metrics details.
- BigAnimal provides a Prometheus-compatible endpoint you can use to connect to your own metrics infrastructure, such as your AWS Managed Grafana. It also provides the option to view logs from your cloud provider's blob storage solution. For more information, see Other monitoring and logging solutions. This ability is an optional solution when you're using your own cloud account but is the only option when using BigAnimal's cloud account.
- BigAnimal provides BigAnimal Observability as another metric endpoint. It consolidates and exposes metrics in each data plane. It is an integrated monitoring and alerting solution in BigAnimal. It's designed to monitor performance of the Postgres clusters. See BigAnimal Observability for more information.
- Existing Postgres Enterprise Manager (PEM) users who want to monitor BigAnimal clusters alongside self-managed Postgres clusters can use the remote monitoring capability of PEM. See Remote monitoring.

  With remote monitoring, you have access to many PEM features, including the ability to profile the workloads on your BigAnimal clusters. See Profiling workloads for more information.

6.7.1 Metrics details

BigAnimal collects a wide set of metrics about Postgres instances and makes them available in your cloud provider. Most of these metrics are acquired directly from Postgres system tables, views, and functions. The Postgres documentation is the main reference for these metrics.

Some data from Postgres monitoring system views, tables, and functions is transformed to be easier to consume in Prometheus metrics format. For example, timestamp fields are generally converted to Unix epoch time and can be accompanied by a relative time-interval metric. Other metrics are aggregated into categories by label dimensions to limit the number of very specific and narrowly scoped individual metrics emitted. It isn't useful to report the inactivity period of every single backend, for example, so backend statistics are aggregated by database, user, `application_name`, and backend state.

The number of tables in your database affects the number of metrics in your cloud logging platform, thus affecting your cloud provider costs for storing these metrics. To ensure stability of the metrics pipeline, metrics might be dropped when the number of tables in your database exceeds 2500.

Prometheus labels are included in the exposed metrics. These will be in the `$.Message.labels` JSON object when consuming a metrics stream, or in a cloud-provider-specific format for metrics ingested into cloud provider monitoring platforms. Dimensions vary depending on the individual metric and are documented separately for each group of related metrics.

The available set of metrics is subject to change. Metrics might be added, removed, or renamed. Where possible, we change the metric name when changing the meaning or type of existing metrics.

`cnp_backends`

Backend counts from `pg_stat_activity` aggregated by the listed label dimensions. Useful for identifying busy applications, excessive idle backends, and so on.

Derived from the `pg_stat_activity` view.

Metric	Usage	Description
`cnp_backends_total`	GAUGE	Number of backends in this group

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Metric	Usage	Description
cnp_backends_max_tx_duration_seconds	GAUGE	Maximum duration of a transaction in seconds in this group
cnp_backends_max_backend_xmin_age	GAUGE	Maximum duration of a transaction in seconds in this group

The metrics in this group can have these labels:

Label	Description
datname	Name of the database for this group of backends
usename	Name of the user in this group of backends
application_name	Name of the application for this group of backends
state	State of the group of backends (pg_stat_activity.state)

cnp_backends_waiting

Postgres instance-level aggregate information on backends that are blocked waiting for locks. Doesn't count I/O waits or other reasons backends might wait or be blocked.

Derived from the pg_locks view.

Metric	Usage	Description
cnp_backends_waiting_total	GAUGE	Total number of backends that are currently waiting on other queries

cnp_pg_database

Per-database metrics for each database in the Postgres instance. Includes per-database vacuum progress information.

Derived from the pg_database catalog.

See also cnp_pg_stat_database.

Metric	Usage	Description
cnp_pg_database_size_bytes	GAUGE	Disk space used by the database
cnp_pg_database_xid_age	GAUGE	Number of transactions from the frozen XID to the current one
cnp_pg_database_mxid_age	GAUGE	Number of multiple transactions (Multixact) from the frozen XID to the current one

The metrics in this group can have these labels:

Label	Description
datname	Name of the database

cnp_pg_postmaster

Data on the Postgres instance's managing "postmaster" process.

Derived from the pg_postmaster_start_time() function.

Metric	Usage	Description
cnp_pg_postmaster_start_time	GAUGE	Time at which postgres started (based on epoch)

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

cnp_pg_replication

Physical replication details for a standby replica postgres instance as captured from the standby replica.

Derived from the `pg_last_xact_replay_timestamp()` function.

Relevant only on standby replicas.

See also cnp_pg_stat_replication, cnp_pg_replication_slots.

Metric	Usage	Description
`cnp_pg_replication_lag`	GAUGE	Replication lag behind primary in seconds
`cnp_pg_replication_in_recovery`	GAUGE	Whether the instance is in recovery

cnp_pg_replication_slots

Details about replication slots on a Postgres instance. In most configurations, only the primary server has active replication clients, but other nodes can still have replication slots.

Logical replication slots are specific to a database, whereas physical replication slots have an empty "database" label as they apply to the Postgres instance as a whole.

Derived from the `pg_replication_slots` view.

See also cnp_pg_stat_replication, cnp_pg_replication.

Metric	Usage	Description
`cnp_pg_replication_slots_active`	GAUGE	Flag indicating if the slot is active
`cnp_pg_replication_slots_pg_wal_lsn_diff`	GAUGE	Replication lag in bytes

The metrics in this group can have these labels:

Label	Description
`slot_name`	Name of the replication slot
`database`	Name of the database

cnp_pg_stat_archiver

Progress information about WAL archiving. Only the currently active primary server generally performs WAL archiving.

WAL archiving is important for backup and restore. If WAL archiving is delayed or failing for too long, the point-in-time recovery backups for a Postgres cluster won't be up to date. This condition has disaster recovery implications and can potentially also affect failover.

Occasional WAL archiving failures are normal, but pay attention to a growing delay in the time since the last successful WAL archiving operation.

The following metrics are reset when a Postgres stats reset is issued on the db server.

Derived from the `pg_stat_archiver` view.

Metric	Usage	Description
`cnp_pg_stat_archiver_archived_count`	COUNTER	Number of WAL files that have been successfully archived
`cnp_pg_stat_archiver_failed_count`	COUNTER	Number of failed attempts for archiving WAL files
`cnp_pg_stat_archiver_seconds_since_last_archival`	GAUGE	Seconds since the last successful archival operation
`cnp_pg_stat_archiver_seconds_since_last_failure`	GAUGE	Seconds since the last failed archival operation
`cnp_pg_stat_archiver_last_archived_time`	GAUGE	Epoch of the last time WAL archiving succeeded
`cnp_pg_stat_archiver_last_failed_time`	GAUGE	Epoch of the last time WAL archiving failed

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Metric	Usage	Description
cnp_pg_stat_archiver_last_archived_wal_start_lsn	GAUGE	Archived WAL start LSN
cnp_pg_stat_archiver_last_failed_wal_start_lsn	GAUGE	Last failed WAL LSN
cnp_pg_stat_archiver_stats_reset_time	GAUGE	Time at which these statistics were last reset

cnp_pg_stat_bgwriter

Stats for the Postgres background writer and checkpointer processes, which are instance-wide and shared across all databases in a Postgres instance.

Very long delays between checkpoints on a busy system increase the time taken for it to return to read/write availability if crash recovery is required. Excessively frequent checkpoints can increase I/O load and the size of the WAL stream for backup and replication.

The Postgres documentation discusses checkpoints, dirty writeback, and checkpoint tuning in detail.

These metrics are reset when a Postgres stats reset is issued on the db server.

Derived from the `pg_stat_bgwriter` catalog.

Metric	Usage	Description
cnp_pg_stat_bgwriter_checkpoints_timed	COUNTER	Number of scheduled checkpoints that have been performed
cnp_pg_stat_bgwriter_checkpoints_req	COUNTER	Number of requested checkpoints that have been performed
cnp_pg_stat_bgwriter_checkpoint_write_time	COUNTER	Total amount of time that has been spent in the portion of checkpoint processing where files are written to disk, in milliseconds
cnp_pg_stat_bgwriter_checkpoint_sync_time	COUNTER	Total amount of time that has been spent in the portion of checkpoint processing where files are synchronized to disk, in milliseconds
cnp_pg_stat_bgwriter_buffers_checkpoint	COUNTER	Number of buffers written during checkpoints
cnp_pg_stat_bgwriter_buffers_clean	COUNTER	Number of buffers written by the background writer
cnp_pg_stat_bgwriter_maxwritten_clean	COUNTER	Number of times the background writer stopped a cleaning scan because it had written too many buffers
cnp_pg_stat_bgwriter_buffers_backend	COUNTER	Number of buffers written directly by a backend
cnp_pg_stat_bgwriter_buffers_backend_fsync	COUNTER	Number of times a backend had to execute its own fsync call (normally the background writer handles those even when the backend does its own write)
cnp_pg_stat_bgwriter_buffers_alloc	COUNTER	Number of buffers allocated

cnp_pg_stat_database

This metrics group directly exposes the summary data Postgres collects in its own `pg_stat_database` view. It contains statistical counters maintained by Postgres for database activity.

These metrics are reset when a Postgres stats reset is issued on the db server.

Derived from the `pg_stat_database` catalog.

See also cnp_pg_database.

Metric	Usage	Description
cnp_pg_stat_database_xact_commit	COUNTER	Number of transactions in this database that have been committed
cnp_pg_stat_database_xact_rollback	COUNTER	Number of transactions in this database that have been rolled back
cnp_pg_stat_database_blks_read	COUNTER	Number of disk blocks read in this database
cnp_pg_stat_database_blks_hit	COUNTER	Number of times disk blocks were found already in the buffer cache, so that a read was not necessary (this only includes hits in the PostgreSQL buffer cache, not the operating system's file system cache)

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Metric	Usage	Description
`cnp_pg_stat_database_tup_returned`	COUNTER	Number of rows returned by queries in this database
`cnp_pg_stat_database_tup_fetched`	COUNTER	Number of rows fetched by queries in this database
`cnp_pg_stat_database_tup_inserted`	COUNTER	Number of rows inserted by queries in this database
`cnp_pg_stat_database_tup_updated`	COUNTER	Number of rows updated by queries in this database
`cnp_pg_stat_database_tup_deleted`	COUNTER	Number of rows deleted by queries in this database
`cnp_pg_stat_database_conflicts`	COUNTER	Number of queries canceled due to conflicts with recovery in this database
`cnp_pg_stat_database_temp_files`	COUNTER	Number of temporary files created by queries in this database
`cnp_pg_stat_database_temp_bytes`	COUNTER	Total amount of data written to temporary files by queries in this database
`cnp_pg_stat_database_deadlocks`	COUNTER	Number of deadlocks detected in this database
`cnp_pg_stat_database_blk_read_time`	COUNTER	Time spent reading data file blocks by backends in this database, in milliseconds
`cnp_pg_stat_database_blk_write_time`	COUNTER	Time spent writing data file blocks by backends in this database, in milliseconds

The metrics in this group can have these labels:

Label	Description
`datname`	Name of this database

`cnp_pg_stat_database_conflicts`

These metrics provide information on conflicts between queries on a standby replica and the standby replica's replay of the change-stream from the primary. These are called *recovery conflicts*.

These metrics are unrelated to "INSERT ... ON CONFLICT" conflicts or multi-master replication row conflicts. They are relevant only on standby replicas.

These metrics are reset when a Postgres stats reset is issued on the db server.

Defined only on standby replicas.

Derived from the `pg_stat_database_conflicts` view.

Metric	Usage	Description
`cnp_pg_stat_database_conflicts_confl_tablespace`	COUNTER	Number of queries in this database that have been canceled due to dropped tablespaces
`cnp_pg_stat_database_conflicts_confl_lock`	COUNTER	Number of queries in this database that have been canceled due to lock timeouts
`cnp_pg_stat_database_conflicts_confl_snapshot`	COUNTER	Number of queries in this database that have been canceled due to old snapshots
`cnp_pg_stat_database_conflicts_confl_bufferpin`	COUNTER	Number of queries in this database that have been canceled due to pinned buffers
`cnp_pg_stat_database_conflicts_confl_deadlock`	COUNTER	Number of queries in this database that have been canceled due to deadlocks

The metrics in this group can have these labels:

Label	Description
`datname`	Name of the database

`cnp_pg_stat_user_tables`

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Access and usage statistics maintained by Postgres on nonsystem tables.

These metrics are reset when a Postgres stats reset is issued on the db server.

Derived from the `pg_stat_user_tables` view.

See also cnp_pg_statio_user_tables.

Metric	Usage	Description
`cnp_pg_stat_user_tables_seq_scan`	COUNTER	Number of sequential scans initiated on this table
`cnp_pg_stat_user_tables_seq_tup_read`	COUNTER	Number of live rows fetched by sequential scans
`cnp_pg_stat_user_tables_idx_scan`	COUNTER	Number of index scans initiated on this table
`cnp_pg_stat_user_tables_idx_tup_fetch`	COUNTER	Number of live rows fetched by index scans
`cnp_pg_stat_user_tables_n_tup_ins`	COUNTER	Number of rows inserted
`cnp_pg_stat_user_tables_n_tup_upd`	COUNTER	Number of rows updated
`cnp_pg_stat_user_tables_n_tup_del`	COUNTER	Number of rows deleted
`cnp_pg_stat_user_tables_n_tup_hot_upd`	COUNTER	Number of rows HOT updated (i.e., with no separate index update required)
`cnp_pg_stat_user_tables_n_live_tup`	GAUGE	Estimated number of live rows
`cnp_pg_stat_user_tables_n_dead_tup`	GAUGE	Estimated number of dead rows
`cnp_pg_stat_user_tables_n_mod_since_analyze`	GAUGE	Estimated number of rows changed since last analyze
`cnp_pg_stat_user_tables_last_vacuum`	GAUGE	Last time at which this table was manually vacuumed (not counting VACUUM FULL)
`cnp_pg_stat_user_tables_last_autovacuum`	GAUGE	Last time at which this table was vacuumed by the autovacuum daemon
`cnp_pg_stat_user_tables_last_analyze`	GAUGE	Last time at which this table was manually analyzed
`cnp_pg_stat_user_tables_last_autoanalyze`	GAUGE	Last time at which this table was analyzed by the autovacuum daemon
`cnp_pg_stat_user_tables_vacuum_count`	COUNTER	Number of times this table has been manually vacuumed (not counting VACUUM FULL)
`cnp_pg_stat_user_tables_autovacuum_count`	COUNTER	Number of times this table has been vacuumed by the autovacuum daemon
`cnp_pg_stat_user_tables_analyze_count`	COUNTER	Number of times this table has been manually analyzed
`cnp_pg_stat_user_tables_autoanalyze_count`	COUNTER	Number of times this table has been analyzed by the autovacuum daemon

The metrics in this group can have these labels:

Label	Description
`datname`	Name of current database
`schemaname`	Name of the schema that this table is in
`relname`	Name of this table

`cnp_pg_stat_replication`

Realtime information about replication connections to this Postgres instance, their progress, and their activity.

These metrics aren't reset when a Postgres stats reset is issued on the db server. The "stat" in the name is a historic artifact from Postgres development.

Derived from the `pg_stat_replication` view.

See also cnp_pg_replication_slots, cnp_pg_replication.

Metric	Usage	Description
`cnp_pg_stat_replication_backend_start_age`	GAUGE	How long ago in seconds this process was started
`cnp_pg_stat_replication_backend_xmin_age`	COUNTER	The age of this standby's xmin horizon
`cnp_pg_stat_replication_sent_diff_bytes`	GAUGE	Difference in bytes from the last write-ahead log location sent on this connection

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Metric	Usage	Description
`cnp_pg_stat_replication_write_diff_bytes`	GAUGE	Difference in bytes from the last write-ahead log location written to disk by this standby server
`cnp_pg_stat_replication_flush_diff_bytes`	GAUGE	Difference in bytes from the last write-ahead log location flushed to disk by this standby server
`cnp_pg_stat_replication_replay_diff_bytes`	GAUGE	Difference in bytes from the last write-ahead log location replayed into the database on this standby server
`cnp_pg_stat_replication_write_lag_seconds`	GAUGE	Time elapsed between flushing recent WAL locally and receiving notification that this standby server has written it
`cnp_pg_stat_replication_flush_lag_seconds`	GAUGE	Time elapsed between flushing recent WAL locally and receiving notification that this standby server has written and flushed it
`cnp_pg_stat_replication_replay_lag_seconds`	GAUGE	Time elapsed between flushing recent WAL locally and receiving notification that this standby server has written, flushed and applied it

The metrics in this group can have these labels:

Label	Description
`usename`	Name of the replication user
`application_name`	Name of the application

`cnp_pg_statio_user_tables`

I/O activity statistics maintained by Postgres on nonsystem tables.

These metrics are reset when a Postgres stats reset is issued on the db server.

Derived from the `pg_statio_user_tables` view.

See also cnp_pg_stat_user_tables.

Metric	Usage	Description
`cnp_pg_statio_user_tables_heap_blks_read`	COUNTER	Number of disk blocks read from this table
`cnp_pg_statio_user_tables_heap_blks_hit`	COUNTER	Number of buffer hits in this table
`cnp_pg_statio_user_tables_idx_blks_read`	COUNTER	Number of disk blocks read from all indexes on this table
`cnp_pg_statio_user_tables_idx_blks_hit`	COUNTER	Number of buffer hits in all indexes on this table
`cnp_pg_statio_user_tables_toast_blks_read`	COUNTER	Number of disk blocks read from this table's TOAST table (if any)
`cnp_pg_statio_user_tables_toast_blks_hit`	COUNTER	Number of buffer hits in this table's TOAST table (if any)
`cnp_pg_statio_user_tables_tidx_blks_read`	COUNTER	Number of disk blocks read from this table's TOAST table indexes (if any)
`cnp_pg_statio_user_tables_tidx_blks_hit`	COUNTER	Number of buffer hits in this table's TOAST table indexes (if any)

The metrics in this group can have these labels:

Label	Description
`datname`	Name of current database
`schemaname`	Name of the schema that this table is in
`relname`	Name of this table

`cnp_pg_settings`

Expose the subset of Postgres server settings that can be represented as Prometheus compatible metrics—any integer, Boolean, or real number. Text-format settings, list-valued settings, and enumeration-typed settings aren't captured or reported.

This set of metrics doesn't expose per-database settings assigned with `ALTER DATABASE ... SET ...`, per-user settings assigned with `ALTER USER ... SET ...`, or per-session values. It shows only the database systemwide global values. You can explore other settings interactively using Postgres system views.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Derived from the `pg_settings` view.

Metric	Usage	Description
`cnp_pg_settings_setting`	GAUGE	Setting value. Note that settings are only reported when they were changed via Cloud Native PostgreSQL.

The metrics in this group can have these labels:

Label	Description
`name`	Name of the setting

`cnp_xlog_insert`

Reports the postgres instance's transaction log insert position in bytes. Useful to compare one postgres instance's WAL insert position with other instances' replication replay positions in monitoring.

Metric	Usage	Description
`cnp_xlog_insert_lsn`	GAUGE	Node xlog insert position (lsn)

`cnp_bdr_rep_slot_stats`

Metrics from pg_catalog.pg_stat_replication_slots for each BDR replication slot. These metrics can be used to monitor logical decoding activity and performance the sending (upstream) side of a logical replication connection. See `pg_stat_replication_slots` for details.

Metric	Usage	Description
`cnp_bdr_rep_slot_stats_spill_txns`	COUNTER	spill_txns
`cnp_bdr_rep_slot_stats_spill_count`	COUNTER	spill_count
`cnp_bdr_rep_slot_stats_spill_bytes`	COUNTER	spill_bytes
`cnp_bdr_rep_slot_stats_stream_txns`	COUNTER	stream_txns
`cnp_bdr_rep_slot_stats_stream_count`	COUNTER	stream_count
`cnp_bdr_rep_slot_stats_stream_bytes`	COUNTER	stream_bytes
`cnp_bdr_rep_slot_stats_total_txns`	COUNTER	total_txns
`cnp_bdr_rep_slot_stats_total_bytes`	COUNTER	total_bytes

The metrics in this group can have these labels:

Label	Description
`peer_name`	peer_name
`slot_name`	slot_name

`cnp_bdr_rep_lag`

Metrics based on the bdr.node_replication_rates monitoring catalog for monitoring BDR replication performance and replication lag. See Monitoring Outgoing Replication and `bdr.node_replication_rates`

Metric	Usage	Description
`cnp_bdr_rep_lag_replay_lag_s`	GAUGE	replay_lag_s
`cnp_bdr_rep_lag_replay_lag_bytes`	GAUGE	replay_lag_bytes
`cnp_bdr_rep_lag_apply_rate`	GAUGE	apply_rate

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Metric	Usage	Description
cnp_bdr_rep_lag_catchup_interval_s	GAUGE	catchup_interval_s

The metrics in this group can have these labels:

Label	Description
peer_name	peer_name

cnp_bdr_node_slots

Metrics derived from the bdr.node_slots view. These metrics provide lower level insight into the progress of outbound BDR replication, including transaction ID limits and WAL retention and the connection status of replication sessions.

Metric	Usage	Description
cnp_bdr_node_slots_active_pid	GAUGE	active_pid
cnp_bdr_node_slots_xmin_age	GAUGE	xmin age
cnp_bdr_node_slots_catalog_xmin_age	GAUGE	catalog_xmin age
cnp_bdr_node_slots_restart_lsn_age	GAUGE	restart_lsn age
cnp_bdr_node_slots_confirmed_flush_lsn_age	GAUGE	confirmed_flush_lsn age
cnp_bdr_node_slots_flush_lag_bytes	GAUGE	flush_lag in bytes
cnp_bdr_node_slots_replay_lag_bytes	GAUGE	replay_lag in bytes
cnp_bdr_node_slots_slot_state	GAUGE	slot_state enumeration. disconnected = 0, streaming = 1, catchup = 2, unknown/unrecognised - 1

The metrics in this group can have these labels:

Label	Description
peer_name	peer_name
slot_name	slot_name

cnp_bdr_global_locking

metrics for bdr global lock acquire and hold durations for both DDL and DML lock types. Useful for detection of long global lock waits or frequent global locks that may impact performance. These metrics are not fine grained and do not expose information about individual tables, etc. Details are available in the bdr.global_locks view.

Metric	Usage	Description
cnp_bdr_global_locking_since_locally_requested_s	GAUGE	since_locally_requested_s
cnp_bdr_global_locking_since_local_granted_s	GAUGE	since_local_granted_s

The metrics in this group can have these labels:

Label	Description
lock_type	lock_type

Disabled: cnp_bdr_raft_mon

This metric has been disabled for performance and reliability reasons. It will no longer be generated after 2023-08-16. It was used to report on the health of the Raft distributed consensus layer on PGD nodes. Please see the EDB Postgres Distributed monitoring documentation for other methods to monitor Raft health on PGD clusters.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Metric	Usage	Description
~~cnp_bdr_raft_mon_raftstatus~~	GAUGE	Raft health status; 0 for unhealthy, 1 for healthy

Other metrics streams

In addition to Postgres metrics from the Cloud Native PostgreSQL operator that manages databases in BigAnimal, additional metrics on Kubernetes workload health etc are available from the BigAnimal metrics endpoints. Specific metrics exposed may vary depending on Kubernetes version, BigAnimal deployment model and more. Any such metrics are generally well-known metrics from widely used tools, documented by the upstream vendor of the component. The BigAnimal platform makes no guarantees about the availability or stability of these metrics unless explicitly documented otherwise.

See also Kubernetes cluster metrics.

6.7.2 Viewing metrics and logs from Azure

BigAnimal sends all metrics and logs from Postgres clusters to Azure. You can view metrics and logs directly from Azure.

Azure log analytics

When BigAnimal deploys workloads on Azure, the logs from the Postgres clusters are forwarded to the Azure Log Workspace. To query BigAnimal logs, you must use Azure log analytics and Kusto query language.

Query Postgres cluster logs

All logs from your Postgres clusters are stored in the Customer Log Analytics workspace. To find your Customer Log Analytics workspace:

1. Sign in to the Azure portal.
2. Select **Resource Groups**.
3. Select the resource group corresponding to the region where you choose to deploy your BigAnimal cluster. You see resources included in that resource group.
4. Select the resource of type **Log Analytics workspace** with the prefix `log-workspace-`.
5. Select the logs in the menu on the left in the **General** section.
6. Close the dashboard with prebuilt queries, which brings you to the KQL editor.

The following tables containing BigAnimal logs are available in the Customer Log Analytic workspace.

Table name	Description	Logger
PostgresLogs_CL	Logs of the customer clusters databases (all Postgres-related logs)	`logger = postgres`
PostgresAuditLogs_CL	Audit logs of the customer clusters databases	`logger = pgaudit or edb_audit`

Logs are split into structured fields matching those of the Postgres csvlog format with a `record_` prefix and a type suffix. For example, the `application_name` is in the `record_application_name_s` log field. The `pg_cluster_id_s` field identifies the specific Postgres cluster that originated the log message.

You can use the KQL Query editor to compose your queries over these tables. For example:

```
PostgresLogs_CL
| project record_log_time_s, record_error_severity_s, record_detail_s
| sort by record_log_time_s desc

PostgresAuditLogs_CL
| where logger_s == "pgaudit"
| project record_log_time_s, record_error_severity_s, record_audit_statement_s
| sort by record_log_time_s desc
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
PostgresAuditLogs_CL
| where logger_s == "edb_audit"
| project record_log_time_s, record_error_severity_s, record_message_s
| sort by record_log_time_s desc
```

Using BigAnimal's cloud account

To access your Postgres cluster logs, when using BigAnimal's cloud account, generate a SAS token from BigAnimal and use it to download the logs.

1. In the BigAnimal portal, select **Clusters**, select your cluster, and select the **Monitoring & Logging** tab.

2. Select **Generate Token** and copy the SAS token. The SAS token is a sensitive value, so don't make it publicly available. The following is a sample SAS token:

   ```
   https://blobsamples.blob.core.windows.net/?sv=2022-11-02&ss=b&srt=sco&sp=rwlc&se=2023-05-24T09:51:36Z&st=2023-05-24T01:51:36Z&spr=https&sig=<signature>
   ```

3. Enter the `azcopy` command to download the Postgres logs from BigAnimal. For example:

   ```
   azcopy copy '$TOKEN' . --recursive
   INFO: Scanning...
   INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support

   Job aa4b74a0-bc92-be4e-551c-47aec1c1cfc3 has started
   Log file is located at: /Users/sample_user/.azcopy/aa4b74a0-bc92-be4e-551c-47aec1c1cfc3.log

   100.0 %, 5 Done, 0 Failed, 0 Pending, 0 Skipped, 5 Total, 2-sec Throughput (Mb/s): 0.5375


   Job aa4b74a0-bc92-be4e-551c-47aec1c1cfc3 summary
   Elapsed Time (Minutes): 0.0333
   Number of File Transfers: 5
   Number of Folder Property Transfers: 0
   Number of Symlink Transfers: 0
   Total Number of Transfers: 5
   Number of File Transfers Completed: 5
   Number of Folder Transfers Completed: 0
   Number of File Transfers Failed: 0
   Number of Folder Transfers Failed: 0
   Number of File Transfers Skipped: 0
   Number of Folder Transfers Skipped: 0
   TotalBytesTransferred: 134416
   Final Job Status: Completed
   $ tail p-a1b2c3d4d5/kubernetes-logs/p-a1b2c3d4d5/2023/09/26/13/19/azure_customer_postgresql_cluster.var.log.containers.p-a1b2c3d4d5-1_p-a1b2c3d4d5_postgres-c798aa19ea0481c8d9575f025405b3ad9212816ca7e928f997473055499a692c.log
   {"@timestamp":"2023-09-26T13:19:19.572442Z","level":"info","ts":"2023-09-26T13:19:19Z","logger":"wal-archive","msg":"Archived WAL file","logging_pod":"p-a1b2c3d4d5-1","walName":"pg_wal/000000010000000000000006","startTime":"2023-09-26T13:19:18Z","endTime":"2023-09-26T13:19:19Z","elapsedWalTime":1.060413255,"stream":"stdout","logtag":"F","message":"
   {\"level\":\"info\",\"ts\":\"2023-09-26T13:19:19Z\",\"logger\":\"wal-archive\",\"msg\":\"Archived WAL
   ```

Metrics

BigAnimal collects a wide set of metrics about Postgres instances into the `DpMetrics_CL` log analytics table. Most of these metrics are acquired directly from Postgres system tables, views, and functions. The Postgres documentation serves as the main reference for these metrics.

Available metrics

The following tables in the Customer Log Analytic workspace contain metrics specific to BigAnimal.

Table name	Description
DpMetrics_CL	Metrics streams from BigAnimal Prometheus

You can use the KQL Query editor in the Log Workspace view to compose queries over these tables.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

The forwarded Prometheus metrics use structured JSON fields, particularly the `Message` and `Message.labels` fields. To query these fields, you need to use the KQL function `todynamic()` in your queries.

Currently, all metrics forwarded from Prometheus are in the `DpMetrics_CL` table. This might change in a future release.

The available set of metrics is subject to change. Metrics might be added, removed or renamed. Where feasible, we will change the metric name when the meaning or type of an existing metric changes.

Metrics labels

All Postgres metrics share a common labeling scheme. Entries generally have at least the following labels.

Name	Description
instance	IP address of the host the metric originated from
postgresql	BigAnimal postgres cluster identifier, e.g., `p-abcdef012345`
role	Postgres instance role: primary or standby replica
datname	Postgres database name (where applicable)
pod	k8s pod name
aks_cluster_name	AKS cluster name

When querying for labels, for best performance, apply filters that don't require inspecting labels (for example, filters by metric name) before label-based filters.

The `labels` field of a metrics entry is a nested field under the JSON-typed `Message` field. To query the field for individual values, you JSON parse the `Message` field and dot reference (`m.labels`) the labels property. Some uses of values might require explicit casts to another type, for example, `tostring(...)`.

Example usage:

```
DpMetrics_CL
| where Message has "cnp_"
| extend m = todynamic(Message)
| where m.labels.role == "primary"
| project postgres_cluster_id = tostring(m.labels.postgresql), metric_name = m.name, metric_value = m.value
```

For a detailed list of available metrics, see metrics.

Workbooks

You can view logs from your Postgres clusters using Azure Workbooks. See Azure Monitor Workbooks.

To access Azure Workbooks:

1. Sign in to the Azure portal.
2. Search for `workbooks`.
3. Filter the search results as needed. For example, you might need to filter by subscription if your account has more than one and the location that corresponds with the region where you deployed your cluster. You can either use the filter controls or search to narrow the search results.
4. Select the workbook link in the **Name** column to open the workbook details. Names have the format `${GUID} (xxx-rg-${region}-management-xxx-customer)`, for example, `62b6d449-e0e5-480a-9809-cae2ff6510e9 (123-rg-australiaeast-management-abcd-customer)`.
5. Select **Open Workbook** in the workbook details to launch an interactive workbook with various metrics for your deployed clusters.

6.7.3 Setting real-time alerts in Azure

BigAnimal provides a wide range of metrics for your clusters. For details, see Metrics details. You can use a Log Analytics query to trigger a real-time alerts based on these metrics.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

This example sets up an alert that's triggered when one of your cluster groups goes down. Assume that your cluster group has an ID = `p-gjlmcos8n7` and is on a subscription called `development`.

1. In the Azure portal, search for and select **Alerts**. On the Alerts page, you can see the metrics that BigAnimal collects from your environment.
2. Select **+ Create** and select **Alert rule** to create new alert rule.
3. Select **+ Select scope**. Under **Select a resource**:
    1. Select your Azure subscription (**development**) in the **Filter by subscription** field.
    2. Select **Log Analytics workspaces** in the **Filter by resource type** field.
    3. Select the Azure region of your cluster in the **Filter by location** field.
4. BigAnimal creates a Log Analytics workspace for each region in your Azure subscription. Select the Log Analytics workspace for your region, and select **Done**. The right column shows the regions.
5. In the **Condition** tab:
    1. To set when the alert rule is triggered, Select **Add condition**.
    2. To monitor the Log Analytics service, select **Custom log search**.
6. To trigger an alert when the database cluster isn't detected, enter the following query:

    ```
    DpMetrics_CL
    | where Message has "cnp_collector_up"
    | extend m = todynamic(Message)
    | where m.labels.role == "primary" and m.labels.postgresql == "p-gjlmcos8n7"
    ```

    The query programs the alert to check for the `cnp_collector_up` metric for the p-gjlmcos8n7 database cluster.
7. After running the query, select **Continue editing alert**.
8. Design the alert logic based on the number of results. Under **Alert logic**:
    1. Set **Operator** to **Equals to**.
    2. In the **Threshold value** field, enter **0**.

        The alert is triggered when the database cluster is unavailable and there are no metrics of `cnp_collector_up`. The alert can indicate a monitoring problem or a critical error.
9. In the **Details** tab, create the alert rule name.
10. Select **Review + Create**.
11. Select **Create**.

6.7.4 Viewing metrics and logs from AWS

BigAnimal sends all metrics and logs from Postgres clusters to AWS. You can view metrics and logs directly from AWS using AWS CloudWatch.

Logs

AWS CloudWatch

When BigAnimal deploys workloads on AWS, the logs from the Postgres clusters are forwarded to AWS CloudWatch log groups as log streams. To query BigAnimal logs, you can use filter and patterns syntax or the CloudWatch Logs Insights section.

Query Postgres cluster logs

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

All logs from your Postgres clusters are presented as log streams in CloudWatch log groups. To find the CloudWatch log groups:

1. Sign in to the AWS console.
2. From Services, select **CloudWatch**.
3. Select the region corresponding where you choose to deploy your BigAnimal cluster.
4. Under **Logs**, select **Log groups** in the menu on the left.

The following log groups and log streams are available in the Customer CloudWatch space.

Log group name	Log stream name	Description	Logger
/biganimal/PostgresLogs	from-pg_cluster_logs_for_customer	Logs of the Customer clusters databases (all Postgres-related logs)	`logger = postgres`
/biganimal/PostgresAuditLogs	from-pg_cluster_audit_logs_for_customer	Audit Logs of the Customer clusters databases	`logger = pgaudit or edb_audit`

You can use the CloudWatch Logs Insights query syntax to compose your queries over these log streams.

Logs are split into structured fields matching those of the Postgres csvlog format with a `record.` prefix. For example, the `sql_state_code` is in the `record.sql_state_code` log field. The `pg_cluster_id` field identifies the specific Postgres cluster that originated the log message.

For example, with the `/biganimal/PostgresLogs` log group selected in CloudWatch Logs Insights, you can view all log entries with the following CloudWatch Logs Insights query:

```
fields record.log_time, record.error_severity, record.message
| sort record.log_time desc
```

With the `/biganimal/PostgresAuditLogsog` log group selected, you can view the pgaudit and edb_audit logs with:

```
fields record.log_time, record.error_severity, record.audit.statement
| filter logger == "pgaudit"
| sort record.log_time desc

fields record.log_time, record.error_severity, record.message
| filter logger == "edb_audit"
| sort record.log_time desc
```

Metrics

The BigAnimal instances collect a variety of metrics and make them available to your AWS subscription for dashboarding, alerting, querying, and other analytics. Most of these metrics are acquired directly from Postgres system tables, views, and functions. The Postgres documentation serves as the main reference for these metrics.

Available metrics

The following log group in AWS CloudWatch contains metrics specific to BigAnimal.

Log group name	Log stream name	Description
/biganimal/Telemetry	from-http.0	Metrics streams from BigAnimal Prometheus.

You can use the CloudWatch Logs Insights query language to compose queries that analyze time-series metrics over these log groups and report the latest samples of metrics. Query the `from-http.0` log stream in the `/biganimal/Telemetry` log group.

The available set of metrics is subject to change. Metrics might be added, removed, or renamed. Where feasible, we will change the metric name when the meaning or type of an existing metric changes.

Currently, all metrics forwarded from Prometheus are in the `/biganimal/Telemetry` log group. This might change in a future release.

Metrics labels

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

All Postgres metrics share a common labeling scheme. Entries generally have at least the following labels.

Name	Description
instance	IP address of the host the metric originated from
postgresql	BigAnimal postgres cluster identifier, e.g., `p-abcdef012345`
role	Postgres instance role: primary or standby replica
datname	Postgres database name (where applicable)
pod	k8s pod name
aks_cluster_name	AKS cluster name

The `labels` field of a metrics entry is a nested field under the JSON-typed `Message` field. To query the field for individual values, you dot reference (`Message.labels`) the labels object.

Example usage:

With `/biganimal/Telemetry` log group selected:

```
fields Message.labels.postgresql,Message.name, Message.value
| filter strcontains (Message.name, 'cnp') and Message.labels.role == 'primary'
| sort @timestamp desc
```

See Metrics for a detailed list of available metrics.

Dashboards

You can view logs and metrics from your Postgres clusters using AWS CloudWatch dashboards.

To access AWS CloudWatch dashboards:

1. Sign in to the AWS console.
2. From Services, select **CloudWatch**.
3. Select the region where you choose to deploy your BigAnimal cluster.
4. In the **CloudWatch** menu on the left, select **Dashboards**.
5. Select **Telemetry Dashboard** to launch a CloudWatch dashboard for your deployed clusters.

6.7.5 Monitoring using BigAnimal Observability

BigAnimal Observability is an integrated monitoring and alerting solution in BigAnimal. It's designed to monitor the performance of the Postgres clusters. It actively monitors various metrics of the Postgres clusters and triggers an alert as the defined thresholds are exceeded. This solution smoothly integrates into the BigAnimal ecosystem, using the vast amount of metric data that's gathered from Postgres clusters and their underlying infrastructure across all the regions.

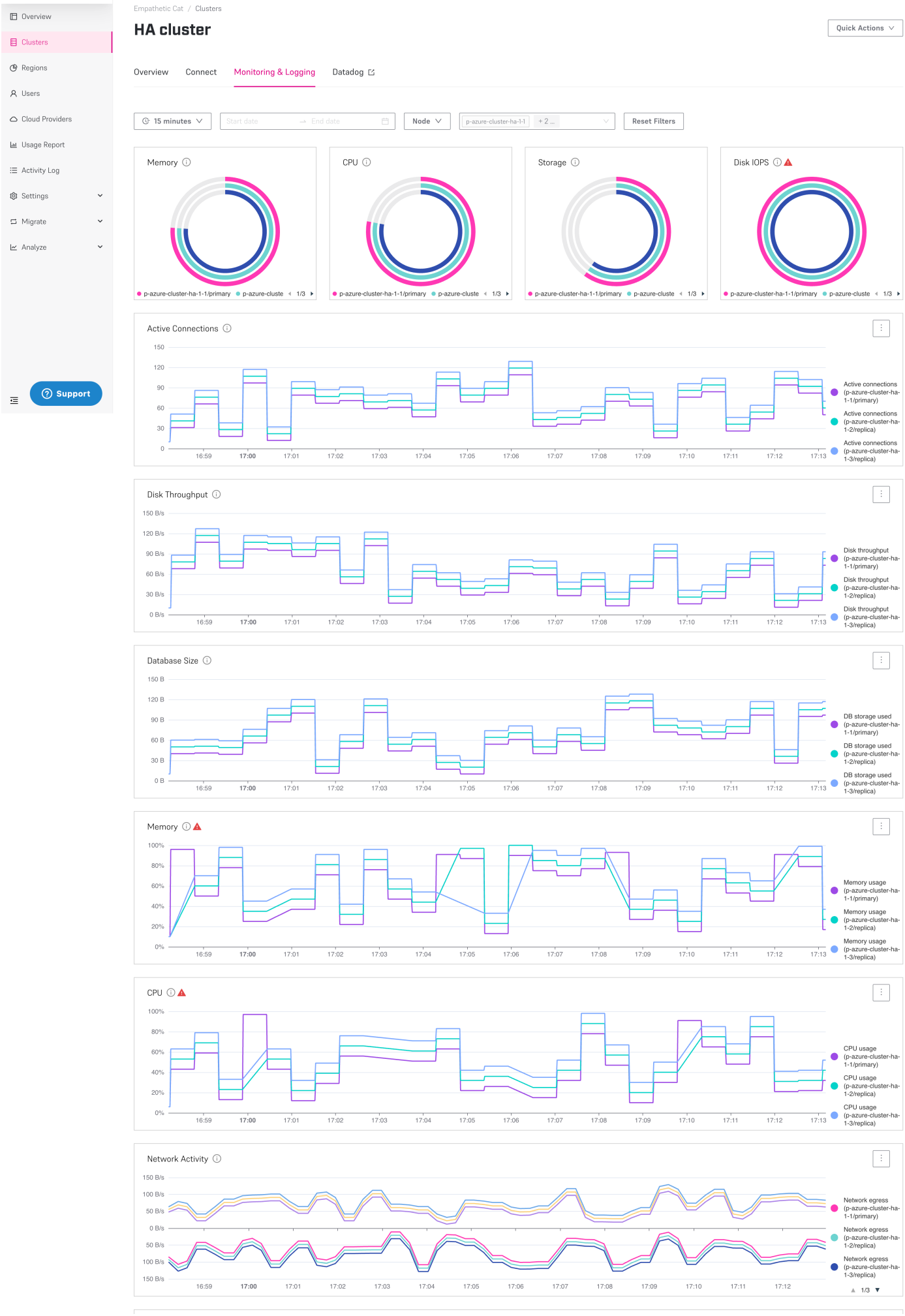
BigAnimal Observability renders insightful charts that empower you to actively monitor each metric's behavior. It provides a streamlined path for you to take prompt and informed actions based on the generated alerts. This cohesive monitoring and alert system ensures comprehensive oversight and prompt responses in the BigAnimal environment.

When you log in, BigAnimal Observability monitoring widgets are displayed on the overview page on the project summary page. These widgets are available only for the deployed Postgres clusters and aren't available for the clusters in the provisioning process. They provide high-level, key information on:

- Memory
- CPU
- Storage usage percentages
- Transactions per second
- Database size

To see more in-depth metrics specific to the Postgres cluster, select any widget on the overview page. Selecting the widget opens the **Monitoring and Logging** tab of the cluster page.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

© 2024 Copyright EnterpriseDB Corporation - All Rights Reserved Privacy Policy Terms of Use About API CLI Terraform Docs Status

To view the **Monitoring and Logging** tab from the BigAnimal portal:

1. In the left navigation of BigAnimal portal, go to **Clusters**.
2. Select any ready cluster.
3. On the cluster detail page, select the **Monitoring and Logging** tab.

The **Monitoring and Logging** tab displays the detailed monitoring metrics at a user-specified level of aggregation in the form of charts:

- Single-value charts
- Historical charts

Levels of aggregation

To view the monitoring metrics at different levels, select any of these levels:

- **Cluster** — This is the default level. It displays the aggregated metrics for the selected cluster to view a bigger picture of the system status.
- **Node** — Select the node from the list at the top to display metrics for each node that composes a cluster. You can select one or more nodes using the button next to the node level. Selecting multiple nodes displays the metrics for each selected node on the chart using a different color for each node. Also, it displays whether a node is primary or replica on the charts.

Single-value charts

These charts display a specific single value based on the last value in the selected time interval. For example, if you choose **Last N**, the chart displays the current value. For the specific time interval, it displays the value at the end of that time interval. These charts display key metrics in text, gauges, or pie and donut form. They provide a concise snapshot of information such as:

- **Memory** (gauge chart) — The percentage of memory used by the Postgres cluster in the hosting node.
- **CPU** (gauge chart) — The percentage of CPU used by the Postgres cluster in the hosting node.
- **Storage** (gauge chart) — The percentage of the storage volume used by the Postgres cluster in the hosting node.
- **Disk IOPS** (gauge chart) — The disk I/O operations ratio of the total disk capacity per second for the Postgres cluster.

Historical charts

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

By default, these charts displays the historical data of the last 15 minutes. To view the historical data of a particular time range, customize the time range using a time-range picker. These charts display key metrics in single-line or multi-line form. They provide a concise snapshot of the information such as:

- **Memory** (line chart) — The historical trend of memory usage percentage over a time period.
- **CPU** (line chart) — The historical trend of CPU usage percentage over a time period.
- **Network activity** (line chart) — The historical data transfer to and from the network card per second, over a time period.
- **Disk IOPS** (line chart) — The historical trends in the number of reads, writes, and total operations on the disk per second, over a time period.
- **Transaction per second** (line chart) — The historical trends in the number of transactions per second, over a time period.
- **Active connections** (line chart) — The current number of connections between the client applications and the Postgres cluster.
- **Disk throughput** (line chart) — The amount of data transferred to and from the disk per second for the Postgres cluster.
- **Database size** (line chart) — The amount of storage volume used by the Postgres cluster.

Features for both types of charts

All these charts have tools and features that help you to get more information about the metrics or the chart. Thetime-range picker helps with viewing the data on these charts for a specific time-range interval. The different level helps to view the data on these charts at cluster and node level. The information tooltip helps you to view the information for a particular chart. The charts error state helps you to find the error and provides the option to edit the configurations and fix the error.

Time-range picker

To view the data of a particular time range configure the time range, on the**Monitoring and Logging** tab, use:

- The **Last X** list
- The date-time picker

The **Last X** list provides several time-range options. Each option in the list is enabled only after the specific time duration has elapsed since the Postgres cluster was created. The default time range is 15 minutes, and the maximum is **Last 30 days**.

> **Note**
>
> When you select the time range from the**Last X** list, the data on the **Monitoring and Logging** tab refreshes every 30 seconds.

Information tooltip

Each chart has an information tooltip that provides a detailed description of the chart. To view the information, hover over the tooltip icon on the right side of the chart name.

Charts error state

A red warning icon is displayed next to the tooltip if there's any error for that chart. If any of the metric exceeds its threshold, an error indicator appears. Selecting the red icon displays a window with a description of the error and an **Edit Cluster** button. Select **Edit Cluster** to go to the Edit Cluster page. Make the configuration changes based on the specific metric that brings the cluster to a healthy state.

The table shows a list of errors and the corresponding solutions.

Error	Solution
High CPU	On the Edit Cluster page, select the preferred category, instance series, and the instance size to increase the CPU.
High Memory	On the Edit Cluster page, select the preferred category, instance series, and the instance size to increase the memory size.
High Storage	On the Edit Cluster page, go to the cluster settings and increase the volume size to increase the storage.
High Disk IOPS	On the Edit Cluster page, go to the cluster settings, volume properties and increase the IOPS within the allowed range.

Features for historical charts

You can zoom the historical charts and also download the data of the historical charts.

Zooming charts

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

To zoom in an area on the historical chart, drag and select that specific area. To reset, select **Reset zoom** from the ellipsis menu at the top-right corner of the chart.

Download CSV

To download the metrics data used to produce the chart in CSV format, from the ellipsis menu in the top-right corner of the chart, select **Download CSV**. The download includes only the data currently visible on the chart. To download the different data, configure the time-range picker before selecting **Download CSV**.

6.7.6 Other monitoring and logging solutions

BigAnimal provides a Prometheus-compatible endpoint you can use to connect to your own monitoring infrastructure as well as Postgres logs by way of blob storage.

If you're using your own cloud account, contact BigAnimal Support to enable this feature on your clusters.

Metrics

You can access metrics in a Prometheus format if you request this feature from BigAnimal Support. You can retrieve the hostname and port for your clusters by using the Prometheus URL available on the **Monitoring and logging** tab on each cluster's detail page in the BigAnimal portal.

These example metrics can help you get started.

Patterns for accessing metrics

A common pattern for metric shipping is to have the vendor-supplied agent scrape the metrics endpoint and send the metrics to the desired platform.

For more information on some common monitoring services, see:

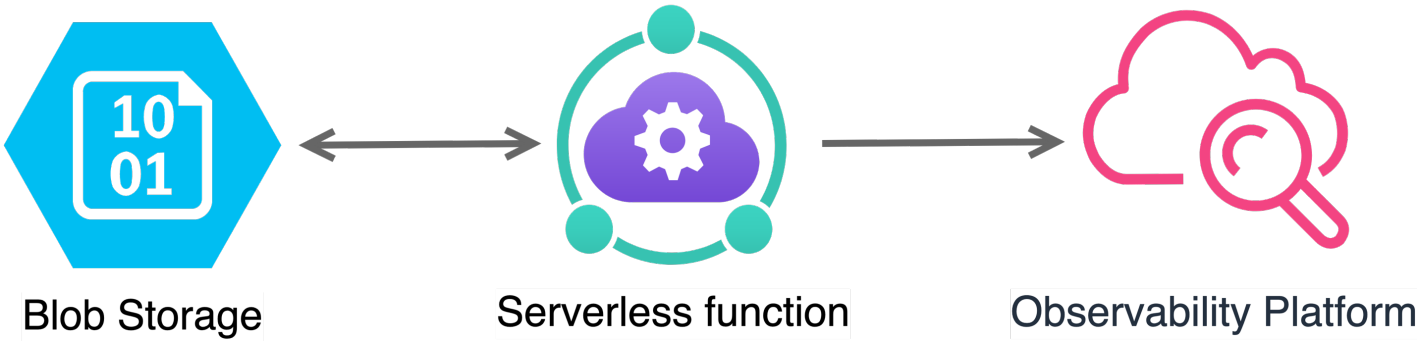
- Dynatrace Prometheus extension
- Self-managed Grafana (not Grafana Cloud): Grafana Prometheus datasource

Logs

You can view your logs in your cloud provider's blob storage solution if you request this feature from BigAnimal Support. You can retrieve the location of your object storage on the **Monitoring and logging** tab on your cluster's detail page in the BigAnimal portal.

The general pattern for getting logs from blob storage into the cloud provider's solution is to write a custom serverless function that watches the blob storage and uploads to the desired solution.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



Watching for logs on AWS

You can leverage some Python code to read the S3 bucket and then use the AWS APIs to upload to your custom monitoring solution. For example, for CloudWatch:aws-load-balancer-logs-to-cloudwatch.

Watching for logs on Azure

You can leverage Azure functions to read the Azure Blob Storage object. Then use the API to upload the data to your monitoring solution.

Uploading logs to common third-party providers

After your function observes new log data, you can use your monitoring provider's API to push log data to their platform.

Some platform providers have limitations regarding the ingestion of logs. Read the vendor documentation carefully.

- Datadog
    - AWS: Collecting logs from S3 buckets
    - Azure: Log collection from Blob Storage
- Dynatrace
    - AWS: S3 log forwarder
    - Azure: Azure log forwarder

    > **Warning**
    >
    > Currently, the Dynatrace integration works only if you can stream the logs from Azure Storage to Azure Event Hub.
- New Relic
    - AWS: Lambda for sending logs from S3
    - Azure: Send logs from Azure Blob storage

CLI command

You can also get the metrics and logs URLs using the `cluster show-monitoring-urls` CLI command. See Logging and metrics CLI command for more information.

6.7.6.1 Example metrics

Here are some helpful queries for monitoring the health of your BigAnimal database. This isn't a complete list. It's intended only as a starting point.

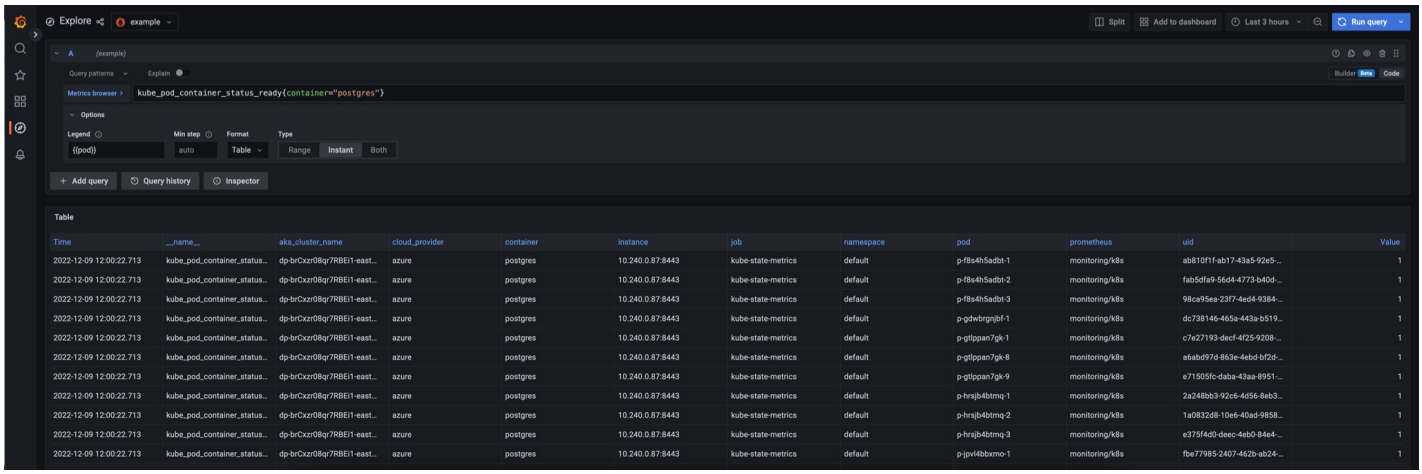
Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

The examples are in PromQL with variables suitable for use in a Grafana dashboard.

Status

This query shows if the pods are up as provided by Kube State metrics:

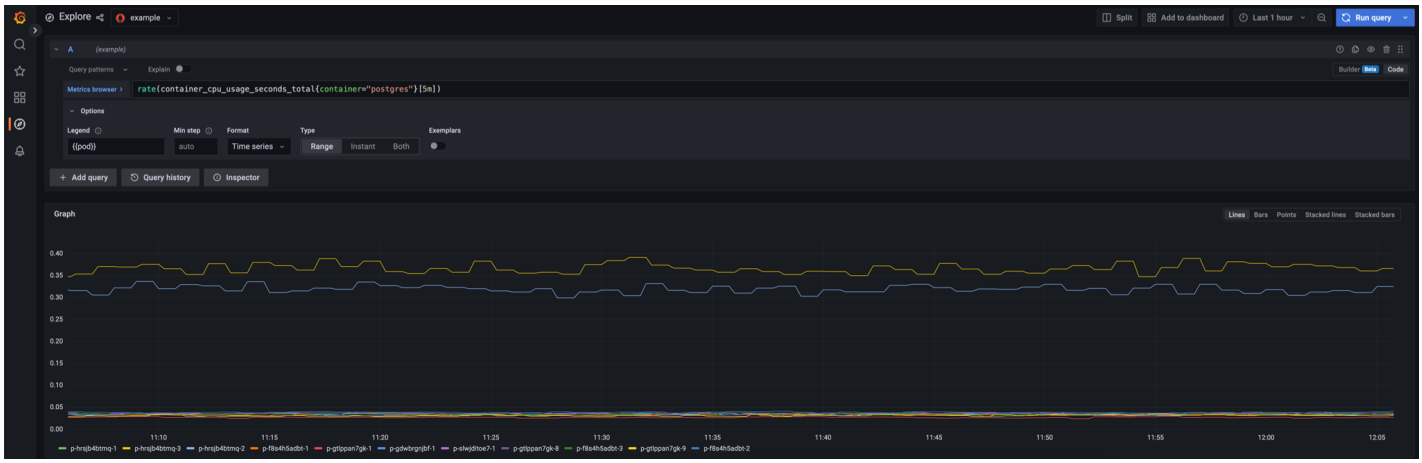
```
kube_pod_container_status_ready{container="postgres"}
```



CPU

This query shows CPU usage as provided by Kubernetes metrics:

```
rate(container_cpu_usage_seconds_total{container="postgres"}[5m])
```



Memory

This query shows memory usage in megabytes as provided by Kubernetes metrics:

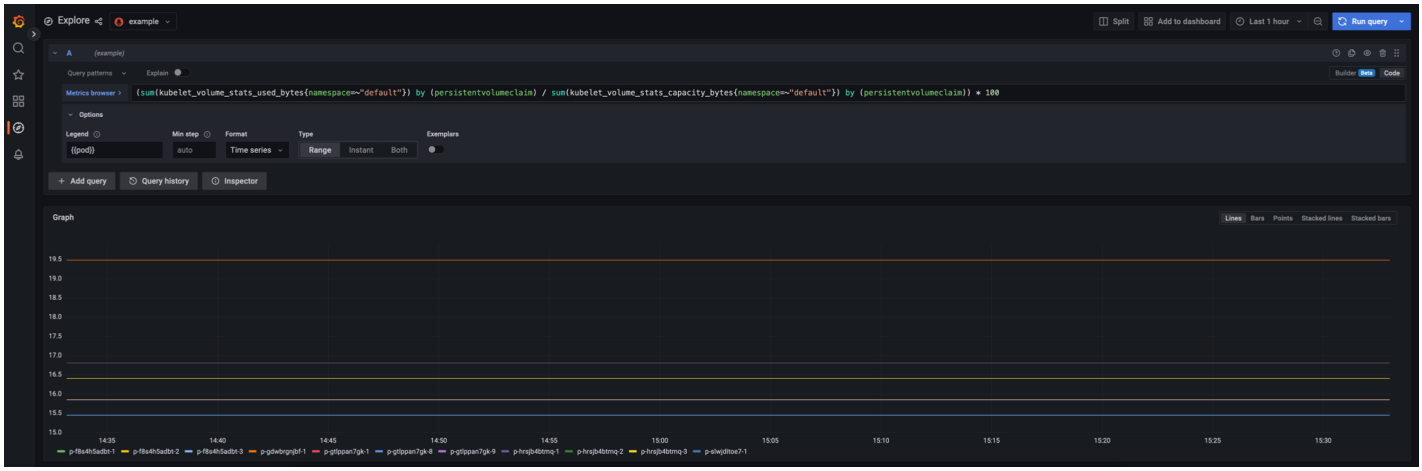
```
container_memory_working_set_bytes{container="postgres"}/1000000
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Storage

This query shows the percentage of available storage used as provided by Kubernetes metrics:

```
(sum(kubelet_volume_stats_used_bytes{namespace=~"default"}) by (persistentvolumeclaim) /
sum(kubelet_volume_stats_capacity_bytes{namespace=~"default"}) by (persistentvolumeclaim)) * 100
```



Replication lag

This query shows the replication lag as provided by the CNP replication lag metric filtered by the `Postgres Instance ($instance)` pod:

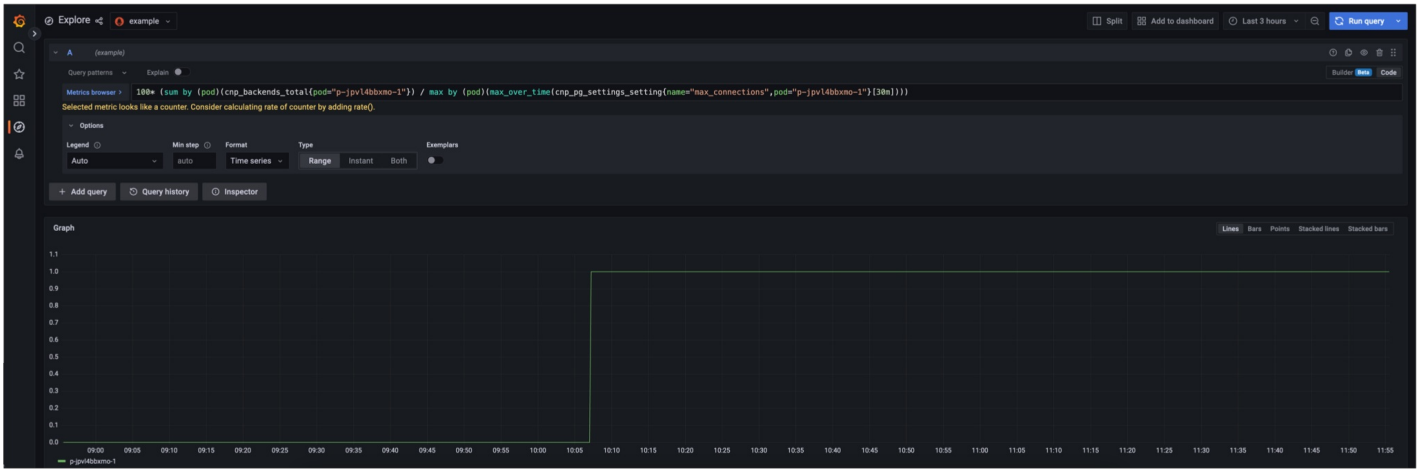
```
cnp_pg_replication_lag{container="postgres", pod="$instance"}
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Connections used

To calculate the connections used as a percentage, you can leverage the cnp_backends_total and cnp_pg_settings_setting metrics. This query calculates the percentage of connections used filtered by the `Postgres Instance ($instance)` pod:

```
100* (sum by (pod)(cnp_backends_total{pod="$instance"}) / max by (pod)
(max_over_time(cnp_pg_settings_setting{name="max_connections",pod="$instance"}[30m])))
```



Transaction ID age

This query shows the database transaction ID age as provided by the CNP database xid metric filtered by the `Postgres Instance ($instance)` pod:

```
(max by (pod) (cnp_pg_database_xid_age{container="postgres", pod="$instance"}))
```

The age is relative to the workloads being run on your database and the following settings. See the Postgres documentation for details:

- autovacuum_freeze_max_age. The default is 200000000.
- vacuum_multixact_failsafe_age. The default is 1600000000.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

6.7.7 Third-party monitoring integrations

BigAnimal provides support for third-party monitoring integrations for using both your own account and BigAnimal's cloud account.

Monitoring integrations are configured at the project level in BigAnimal. You can't turn these integrations on or off for individual clusters. An admin or a project owner can set up an integration. You can set up only one integration per project.

By default, all the integrations are disabled. After creating the project, enable an integration using the**Integrations** tab.

All the metrics collected from all the clusters in the project are sent to the integrated tool and displayed in the BigAnimal**Monitoring and logging** tab using BigAnimal's Observability. The collected logs are exported to the object storage by default.

You can enable in-app inbox or email notifications to get notified incase third-party monitoring integration fails. For more information, see managing notifications.

The third-party integrations available in BigAnimal are:

- Datadog
- New Relic

Metric naming

When metrics from BigAnimal are exported to third-party monitoring services, they're renamed according to the naming conventions of the target platform.

The following table provides a mapping between BigAnimal metric names and the name that metric will be assigned when exported to a third-party service.

> **Kubernetes metrics**
>
> In addition to these metrics, which pertain to the Postgres instances, BigAnimal also exports metrics from the underlying Kubernetes infrastructure. These are prefixed with `k8s.` .

BigAnimal metric name	Metric name for third-party integrations
cnp_pg_backends_waiting_total	postgres.raw.backends_waiting_total
cnp_pg_database_size_bytes	postgres.raw.pg_database_size_bytes
cnp_pg_database_xid_age	postgres.raw.pg_database_xid_age
cnp_pg_database_mxid_age	postgres.raw.pg_database_mxid_age
cnp_pg_postmaster_start_time	postgres.raw.pg_postmaster_start_time
cnp_pg_replication_lag	postgres.raw.pg_replication_lag
cnp_pg_replication_in_recovery	postgres.raw.pg_replication_in_recovery
cnp_pg_replication_slots_active	postgres.raw.pg_replication_slots_active
cnp_pg_replication_slots_pg_wal_lsn_diff	postgres.raw.pg_replication_slots_pg_wal_lsn_diff
cnp_pg_stat_archiver_archived_count	postgres.raw.pg_stat_archiver_archived_count
cnp_pg_stat_archiver_failed_count	postgres.raw.pg_stat_archiver_failed_count
cnp_pg_stat_archiver_seconds_since_last_archival	postgres.raw.pg_stat_archiver_seconds_since_last_archival
cnp_pg_stat_archiver_seconds_since_last_failure	postgres.raw.pg_stat_archiver_seconds_since_last_failure
cnp_pg_stat_archiver_last_archived_time	postgres.raw.pg_stat_archiver_last_archived_time
cnp_pg_stat_archiver_last_failed_time	postgres.raw.pg_stat_archiver_last_failed_time
cnp_pg_stat_archiver_last_archived_wal_start_lsn	postgres.raw.pg_stat_archiver_last_archived_wal_start_lsn
cnp_pg_stat_archiver_last_failed_wal_start_lsn	postgres.raw.pg_stat_archiver_last_failed_wal_start_lsn
cnp_pg_stat_archiver_stats_reset_time	postgres.raw.pg_stat_archiver_stats_reset_time
cnp_pg_stat_bgwriter_checkpoints_timed	postgres.raw.pg_stat_bgwriter_checkpoints_timed
cnp_pg_stat_bgwriter_checkpoints_req	postgres.raw.pg_stat_bgwriter_checkpoints_req
cnp_pg_stat_bgwriter_checkpoint_write_time	postgres.raw.pg_stat_bgwriter_checkpoint_write_time
cnp_pg_stat_bgwriter_checkpoint_sync_time	postgres.raw.pg_stat_bgwriter_checkpoint_sync_time
cnp_pg_stat_bgwriter_buffers_checkpoint	postgres.raw.pg_stat_bgwriter_buffers_checkpoint
cnp_pg_stat_bgwriter_buffers_clean	postgres.raw.pg_stat_bgwriter_buffers_clean
cnp_pg_stat_bgwriter_maxwritten_clean	postgres.raw.pg_stat_bgwriter_maxwritten_clean
cnp_pg_stat_bgwriter_buffers_backend	postgres.raw.pg_stat_bgwriter_buffers_backend
cnp_pg_stat_bgwriter_buffers_backend_fsync	postgres.raw.pg_stat_bgwriter_buffers_backend_fsync
cnp_pg_stat_bgwriter_buffers_alloc	postgres.raw.pg_stat_bgwriter_buffers_alloc

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal metric name	Metric name for third-party integrations
cnp_pg_stat_database_xact_commit	postgres.raw.pg_stat_database_xact_commit
cnp_pg_stat_database_xact_rollback	postgres.raw.pg_stat_database_xact_rollback
cnp_pg_stat_database_blks_read	postgres.raw.pg_stat_database_blks_read
cnp_pg_stat_database_blks_hit	postgres.raw.pg_stat_database_blks_hit
cnp_pg_stat_database_tup_returned	postgres.raw.pg_stat_database_tup_returned
cnp_pg_stat_database_tup_fetched	postgres.raw.pg_stat_database_tup_fetched
cnp_pg_stat_database_tup_inserted	postgres.raw.pg_stat_database_tup_inserted
cnp_pg_stat_database_tup_updated	postgres.raw.pg_stat_database_tup_updated
cnp_pg_stat_database_tup_deleted	postgres.raw.pg_stat_database_tup_deleted
cnp_pg_stat_database_conflicts	postgres.raw.pg_stat_database_conflicts
cnp_pg_stat_database_temp_files	postgres.raw.pg_stat_database_temp_files
cnp_pg_stat_database_temp_bytes	postgres.raw.pg_stat_database_temp_bytes
cnp_pg_stat_database_deadlocks	postgres.raw.pg_stat_database_deadlocks
cnp_pg_stat_database_blk_read_time	postgres.raw.pg_stat_database_blk_read_time
cnp_pg_stat_database_blk_write_time	postgres.raw.pg_stat_database_blk_write_time
cnp_pg_stat_database_conflicts_confl_tablespace	postgres.raw.pg_stat_database_conflicts_confl_tablespace
cnp_pg_stat_database_conflicts_confl_lock	postgres.raw.pg_stat_database_conflicts_confl_lock
cnp_pg_stat_database_conflicts_confl_snapshot	postgres.raw.pg_stat_database_conflicts_confl_snapshot
cnp_pg_stat_database_conflicts_confl_bufferpin	postgres.raw.pg_stat_database_conflicts_confl_bufferpin
cnp_pg_stat_database_conflicts_confl_deadlock	postgres.raw.pg_stat_database_conflicts_confl_deadlock
cnp_pg_stat_user_tables_seq_scan	postgres.raw.pg_stat_user_tables_seq_scan
cnp_pg_stat_user_tables_seq_tup_read	postgres.raw.pg_stat_user_tables_seq_tup_read
cnp_pg_stat_user_tables_idx_scan	postgres.raw.pg_stat_user_tables_idx_scan
cnp_pg_stat_user_tables_idx_tup_fetch	postgres.raw.pg_stat_user_tables_idx_tup_fetch
cnp_pg_stat_user_tables_n_tup_ins	postgres.raw.pg_stat_user_tables_n_tup_ins
cnp_pg_stat_user_tables_n_tup_upd	postgres.raw.pg_stat_user_tables_n_tup_upd
cnp_pg_stat_user_tables_n_tup_del	postgres.raw.pg_stat_user_tables_n_tup_del
cnp_pg_stat_user_tables_n_tup_hot_upd	postgres.raw.pg_stat_user_tables_n_tup_hot_upd
cnp_pg_stat_user_tables_n_live_tup	postgres.raw.pg_stat_user_tables_n_live_tup
cnp_pg_stat_user_tables_n_dead_tup	postgres.raw.pg_stat_user_tables_n_dead_tup
cnp_pg_stat_user_tables_n_mod_since_analyze	postgres.raw.pg_stat_user_tables_n_mod_since_analyze
cnp_pg_stat_user_tables_last_vacuum	postgres.raw.pg_stat_user_tables_last_vacuum
cnp_pg_stat_user_tables_last_autovacuum	postgres.raw.pg_stat_user_tables_last_autovacuum
cnp_pg_stat_user_tables_last_analyze	postgres.raw.pg_stat_user_tables_last_analyze
cnp_pg_stat_user_tables_last_autoanalyze	postgres.raw.pg_stat_user_tables_last_autoanalyze
cnp_pg_stat_user_tables_vacuum_count	postgres.raw.pg_stat_user_tables_vacuum_count
cnp_pg_stat_user_tables_autovacuum_count	postgres.raw.pg_stat_user_tables_autovacuum_count
cnp_pg_stat_user_tables_analyze_count	postgres.raw.pg_stat_user_tables_analyze_count
cnp_pg_stat_user_tables_autoanalyze_count	postgres.raw.pg_stat_user_tables_autoanalyze_count
cnp_pg_stat_replication_backend_start_age	postgres.raw.pg_stat_replication_backend_start_age
cnp_pg_stat_replication_backend_xmin_age	postgres.raw.pg_stat_replication_backend_xmin_age
cnp_pg_stat_replication_sent_diff_bytes	postgres.raw.pg_stat_replication_sent_diff_bytes
cnp_pg_stat_replication_write_diff_bytes	postgres.raw.pg_stat_replication_write_diff_bytes
cnp_pg_stat_replication_flush_diff_bytes	postgres.raw.pg_stat_replication_flush_diff_bytes
cnp_pg_stat_replication_replay_diff_bytes	postgres.raw.pg_stat_replication_replay_diff_bytes
cnp_pg_stat_replication_write_lag_seconds	postgres.raw.pg_stat_replication_write_lag_seconds
cnp_pg_stat_replication_flush_lag_seconds	postgres.raw.pg_stat_replication_flush_lag_seconds
cnp_pg_stat_replication_replay_lag_seconds	postgres.raw.pg_stat_replication_replay_lag_seconds
cnp_pg_statio_user_tables_heap_blks_read	postgres.raw.pg_statio_user_tables_heap_blks_read
cnp_pg_statio_user_tables_heap_blks_hit	postgres.raw.pg_statio_user_tables_heap_blks_hit
cnp_pg_statio_user_tables_idx_blks_read	postgres.raw.pg_statio_user_tables_idx_blks_read
cnp_pg_statio_user_tables_idx_blks_hit	postgres.raw.pg_statio_user_tables_idx_blks_hit
cnp_pg_statio_user_tables_toast_blks_read	postgres.raw.pg_statio_user_tables_toast_blks_read
cnp_pg_statio_user_tables_toast_blks_hit	postgres.raw.pg_statio_user_tables_toast_blks_hit
cnp_pg_statio_user_tables_tidx_blks_read	postgres.raw.pg_statio_user_tables_tidx_blks_read
cnp_pg_statio_user_tables_tidx_blks_hit	postgres.raw.pg_statio_user_tables_tidx_blks_hit

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal metric name	Metric name for third-party integrations
cnp_pg_settings_setting	postgres.raw.pg_settings_setting
cnp_xlog_insert_lsn	postgres.raw.xlog_insert_lsn

6.7.7.1 Datadog

Datadog integration is based on the OpenTelemetry Collector and the Datadog plugin bundled with the OpenTelemetry collector. Once the integration is enabled, BigAnimal configures data planes to send the telemetry to your Datadog account by way of Datadog's cloud receive endpoint.

> **Note**
>
> Datadog integration doesn't install the Datadog agent. It sends telemetry to the Datadog API.

Prerequisites

You need a Datadog account before you can use this integration.

Enabling Datadog

To enable the Datadog integrations:

1. From the Projects page, select an existing project.
2. Go to **Settings** on the left-side navigation.
3. From the **Settings** list, select **Integrations**.
4. Select **Datadog**. In the window, provide the details:
    - **Datadog API Key** — For details, see Datadog API and Application Keys. The API key is sensitive information. You can't retrieve it after you save it.
    - **Datadog Site Name** — Select a site name from the list. The telemetry goes to the Datadog site you choose.
    - **Datadog Site URL** — The Datadog site URL appears based on the Datadog site name you selected.
    - **Datadog API Key ID** — Assign a name for the API key ID. You can use the API key ID to check the configuration details.
5. Select **Save**.

Enabling this integration sends BigAnimal telemetry to your Datadog account.

> **Important**
>
> Generate a new API key in Datadog to enable this integration in BigAnimal. This API key must be specific to this integration. Don't share the key. Whenever you disable this integration, revoke the API key using the Datadog interface. Revoking it disables the Datadog ingestion and billable usage from the BigAnimal integration. Revoking the API key doesn't impact the other services that use the Datadog API.

Metrics

BigAnimal sends a subset of the OpenTelemetry Collector's metrics from the hostmetrics and kubeletstats receivers. It also sends BigAnimal custom metrics for the monitored Postgres instance.

You can see a list of metrics in the Datadog interface along with the tags for each metric.

To see the list of metrics, select **Metrics > Summary**. Then select a metric to see the tags for that metric.

The set of metrics delivered to Datadog is subject to change. Metrics with names that begin with `postgres.preview.` or `biganimal.preview.` are likely to be renamed or removed in a future release. Other metrics may also be renamed, added, or removed to better integrate into the DataDog platform.

Cost

Telemetry sent to Datadog will incur charges on your Datadog plan according to the Datadog price list. We recommend that you continually monitor your billable Datadog usage as the cost could change. Datadog provides usage metrics for this purpose, such as `datadog.estimated_usage.hosts`, `datadog.estimated_usage.containers`, and `datadog.estimated_usage.metrics.custom`.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Be aware of the following cost considerations:

- Datadog bills for each monitored Kubernetes node as an "Infrastructure Host".
- Datadog may also bill for each Postgres container and monitoring agent container at the "Container Monitoring" rate.
- Datadog counts some of the metrics sent by BigAnimal as custom metrics. Custom metrics dimensions above the free limit are billable at a rate set in the Datadog price list.
- The Datadog metrics without limits feature can limit cardinality-based billing for custom metrics. However, it enables ingestion-based billing instead, so the overall price might actually be greater.

Disabling Datadog

To disable the Datadog integration for BigAnimal and ensure no further costs are incurred on your Datadog account, you must revoke the API key provided to BigAnimal. Disabling the integration on the BigAnimal portal doesn't prevent costs from being incurred.

Further reading

For more information, see:

- Datadog pricing list
- Custom metrics billing
- Usage metrics billing

6.7.7.2 New Relic

The New Relic integration is based on the OpenTelemetry Collector, shipping the metrics over the standard OLTP protocol to the New Relic cloud endpoint.

> **Note**
>
> New Relic integration doesn't install the New Relic agent. It sends telemetry to the New Relic API.

Prerequisites

You need a New Relic account before you can use this integration.

Enable New Relic

To enable the **New Relic** integration:

1. From the **Projects** page, select an existing project.
2. Go to **Settings** on the left-side navigation.
3. From the **Settings** list, select **Integrations** .
4. Select **New Relic**. In the window, provide the details:
    - **New Relic API Key** — For details, see New Relic API Key. The API key is sensitive information. You can't retrieve it after you save it.
    - **New Relic API Key Name** — Assign a name for the API key. You can use the API Key ID name to check the configuration details.
    - **New Relic Account ID** — Provide the New Relic account ID.
5. Select **Save**.

Enabling this integration sends BigAnimal telemetry to your New Relic account.

> **Important**
>
> Generate a new API license key in New Relic to enable this integration in BigAnimal. This API key must be specific to this integration. Don't share the key. Whenever you disable this integration, revoke the API key using the New Relic interface. Revoking the key disables the New Relic ingestion and billable usage from the BigAnimal integration. Revoking the key doesn't impact the other services that use the New Relic API.

Metrics

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal sends a subset of the OpenTelemetry Collector's metrics from the hostmetrics and kubeletstats receivers. It also sends BigAnimal custom metrics for the monitored Postgres instance.

You can see a list of metrics in the New Relic interface along with dimensions for each metric.

To see a list of metrics, select **Metrics & Events**. Then select a metric to see the dimensions sent for that metric.

The set of metrics delivered to New Relic is subject to change. Metrics with names that begin with `postgres.preview.` or `biganimal.preview.` might be renamed or removed in a future release. Other metrics might also be renamed, added, or removed to better integrate into the New Relic platform.

Cost

After enabling the BigAnimal telemetry integration, check your billable New Relic usage and continue to monitor it over time.

Be aware of the following important cost considerations:

- *You are responsible for all costs* charged to your New Relic account by telemetry sent by the BigAnimal New Relic integration. Charges are based on usage, but they can also result from BigAnimal errors or oversights. Enable the New Relic integration only if you accept this responsibility.

- New Relic bills usage by bytes ingested and for data retention. You must monitor and configure alerts on your New Relic data ingestion to reduce the risk of unexpectedly large usage bills. Also check your retention settings.

New Relic has features to limit usage and ingestion. Review your limits before enabling the BigAnimal integration. In New Relic, select**Administration > Data management > Limits**. Also see the following related topics in the New Relic documentation:

- Understand and manage data ingest
- Data ingest: Billing and rules
- Understand New Relic data limits
- Get more detail about your data limits

To disable the New Relic integration for BigAnimal and ensure no further costs are incurred on your New Relic account, you must revoke the API key provided to BigAnimal. Disabling the integration on the BigAnimal portal doesn't prevent costs from being incurred.

6.8 Fault injection testing

You can test the fault tolerance of your cluster by deleting a VM in order to inject a fault. Once a VM is deleted, you can monitor the availability and recovery of the cluster.

Requirements

Ensure you meet the following requirements before using fault injection testing:

- You have connected your BigAnimal cloud account with your Azure subscription. See Setting up your Azure Marketplace account for more information.
- You have permissions in your Azure subscription to view and delete VMs and also the ability to view Kubernetes pods via Azure Kubernetes Service RBAC Reader.
- You have PGD CLI installed. See Installing PGD CLI for more information.
- You have created a `pgd-cli-config.yml` file in your home directory. See Configuring PGD CLI for more information.

Fault injection testing steps

Fault injection testing consists of the following steps:

1. Verifying cluster health
2. Determining the write leader node for your cluster
3. Deleting a write leader node from your cluster
4. Monitoring cluster health

Verifying Cluster Health

Use the following commands to monitor your cluster health, node info, raft, replication lag, and write leads.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
pgd check-health -f pgd-cli-config.yml
pgd verify-cluster -f pgd-cli-config.yml
pgd show-nodes -f pgd-cli-config.yml
pgd show-raft  -f pgd-cli-config.yml
pgd show-replslots -verbose -f pgd-cli-config.yml
pgd show-subscriptions -f pgd-cli-config.yml
pgd show-groups -f pgd-cli-config.yml
```

You can use `pgd help` for more information on these commands.

To list the supported commands, enter:

```
pgd help
```

For help with a specific command and its parameters, enter `pgd help <command_name>`. For example:

```
pgd help show-nodes
```

Determining the write leader node for your cluster

```
pgd show-groups -f pgd-cli-config.yml
```

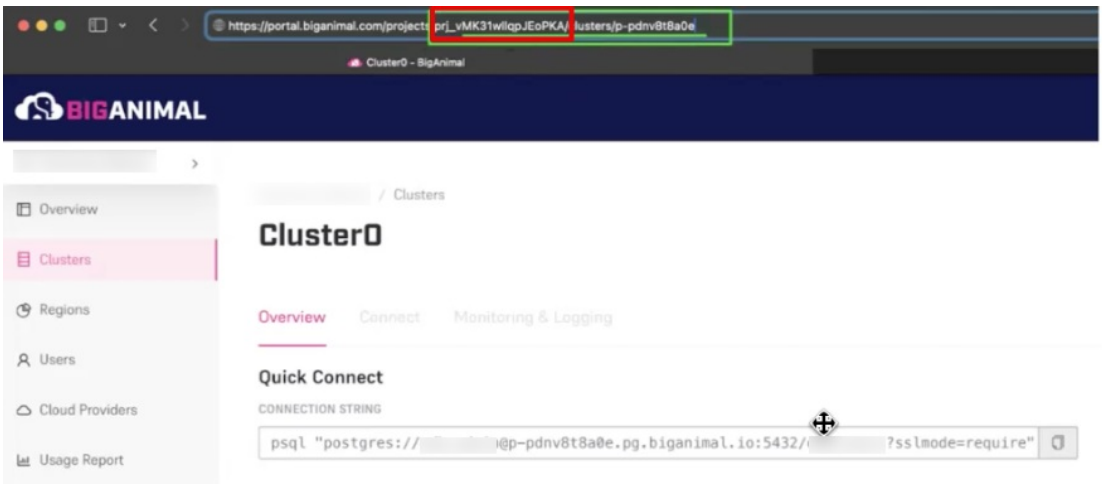
```
Group              Group ID         Type     Write Leader
--------        ------------------   ----     ------------
world           3239291720  global           p-x67kjp3fsq-d-1
p-x67kjp3fsq-a  2456382099  data     world   p-x67kjp3fsq-a-1
p-x67kjp3fsq-c  4147262499  data     world
p-x67kjp3fsq-d  3176957154  data     world   p-x67kjp3fsq-d-1
```

In this example, the write leader node is **p-x67kjp3fsq-a-1**.

Deleting a write leader node from your cluster

To delete a write lead node from the cluster:

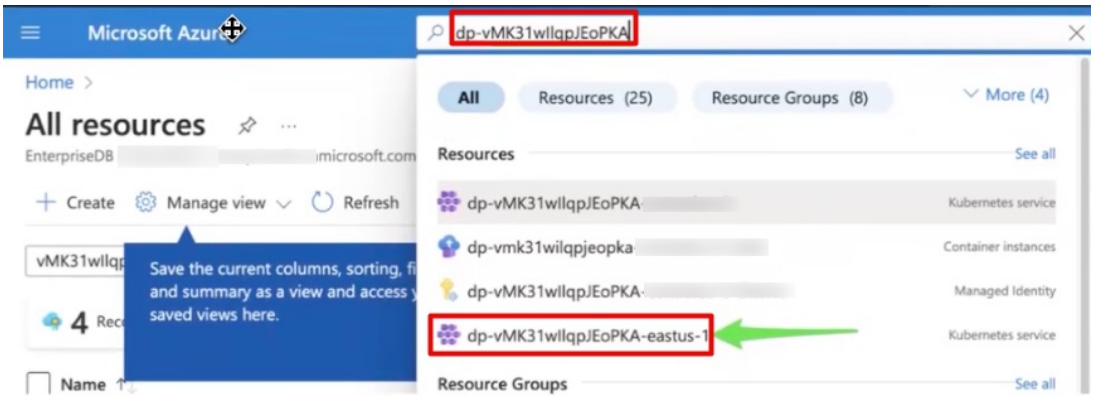
1. Log into BigAnimal.
2. In a separate browser window, log into your Microsoft Azure subscription.
3. In the left navigation of BigAnimal portal, choose **Clusters**.
4. Choose the cluster to test fault injection with and copy the string value from the URL. The string value is located after the underscore.



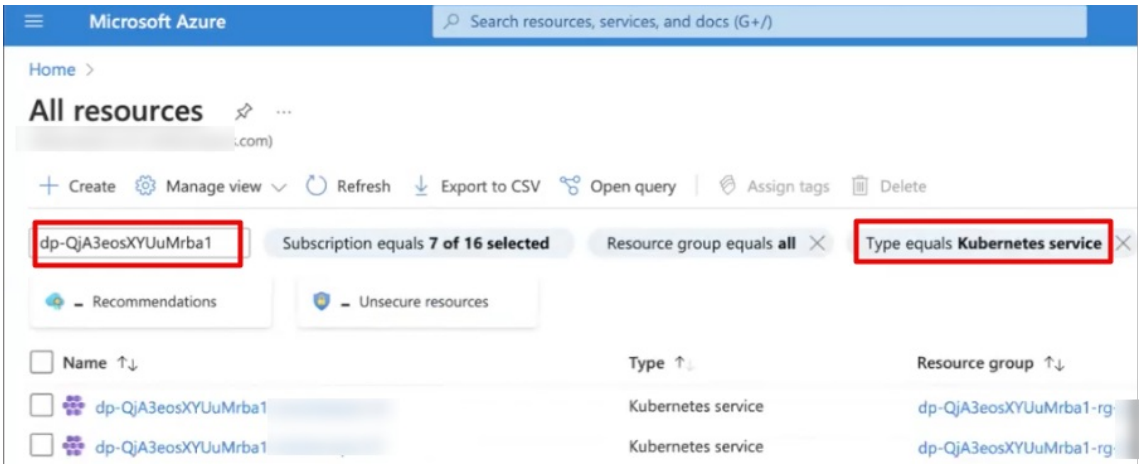
5. In your Azure subscription, paste the string into the search and prefix it with **dp-** to search for the data plane.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- From the results, choose the Kubernetes service from the Azure Region that your cluster is deployed in.



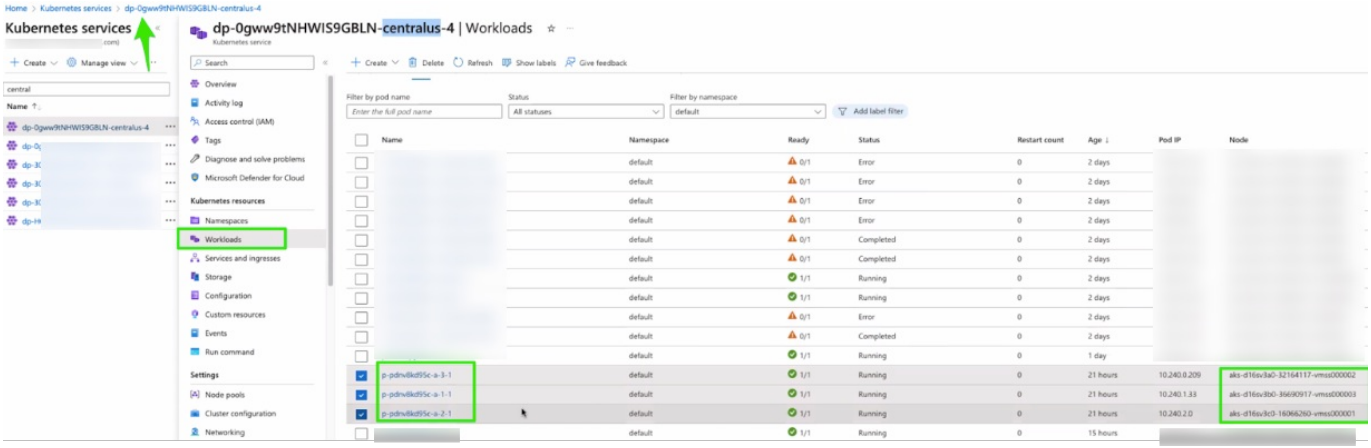
6. Identify the Kubernetes service for your cluster.



Note

Don't delete the Azure Kubernetes VMSS here or sub resources directly.

7. Browse to the Data Plane, choose Workloads, and locate the Kubernetes resources for your cluster to delete a chosen node.



Monitoring cluster health

After deleting a cluster node, you can monitor the health of the cluster using the same PGD CLI commands that you used to verify cluster health.

6.9 Deleting your cluster

If you no longer need a particular cluster, you can delete it from the BigAnimal portal. You can restore deleted clusters after you delete the cluster for as long as the backup is available. The

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

backup remains available for the retention period set for each cluster. After the retention period, the backup is deleted.

To stop the management costs on the cluster, contact EDB's BigAnimal Support team, Otherwise, costs might continue to accumulate.

Delete your cluster

> **Important**
>
> You can delete a cluster only if you have either promoted or deleted each of its faraway replicas. See Faraway replicas.

1. Go to the Clusters page in the BigAnimal portal.
2. To delete the cluster, do one of the following:
   - Select the **Delete cluster** icon in the row for the cluster.
   - Select the cluster you want to delete. From **Quick Actions** on the cluster details page, select **Delete**.
3. Confirm that you want to delete the cluster. When the process finishes, the cluster is listed on the **Deleted** tab of the Clusters page.
4. To stop the associated management costs, contact Support. For more information on management costs, see Pricing and billing.

Restore your cluster

You can restore your deleted cluster for as long as the backup is available.

1. Select the **Delete** tab on the Clusters page in the BigAnimal portal.
2. Select the **Restore** icon for the cluster you want to restore.
3. On the Restore Cluster page, review your selections in the **Cluster Summary**. Select **Restore Cluster** to begin the restore process.

   When the process completes, the restored cluster is available on the Clusters page.

> **Note**
>
> To restore a TDE enabled cluster, the TDE key material must match with source cluster encryption key material. In case a different key material is used the restore operation fails.
>
> We recommend, not to enable TDE while restoring a cluster, if the source cluster is a non-TDE cluster.

6.10 Performing a major version upgrade of Postgres on BigAnimal

Using logical replication

> **Note**
>
> This procedure does not work with distributed high-availability BigAnimal instances.

Logical replication is a common method for upgrading the Postgres major version on BigAnimal instances, enabling a transition with minimal downtime.

By replicating changes in real-time from an older version (source instance) to a newer one (target instance), this method provides a reliable upgrade path while maintaining database availability.

> **Important**
>
> Depending on where your older and newer versioned BigAnimal instances are located, this procedure may accrue ingress and egress costs from your cloud service provider (CSP) for the migrated data. Please consult your CSP's pricing documentation to see how ingress and egress fees are calculated to determine any extra costs.

Overview of upgrading

To perform a major version upgrade, use the following steps, explained in further detail below:

1. Create a BigAnimal instance

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

2. Gather instance information
3. Confirm the Postgres versions before migration
4. Migrate the database schema
5. Create a publication
6. Create a logical replication slot
7. Create a subscription
8. Validate the migration

Create a BigAnimal instance

To perform a major version upgrade, create a BigAnimal instance with your desired version of Postgres. This will be your target instance.

Ensure your target instance is provisioned with a storage size equal to or greater than your source instance.

For detailed steps on creating a BigAnimal instance, see this guide.

Gather instance information

Use the BigAnimal console to obtain the following information for your source and target instance:

- Read/write URI
- Database name
- Username
- Read/write host

Using the BigAnimal console:

1. Select the **Clusters** tab.
2. Select your source instance.
3. From the Connect tab, obtain the information from **Connection Info**.

Confirm the Postgres versions before migration

Confirm the Postgres version of your source and target BigAnimal instances:

```
psql "<biganimal_read/write_uri>" -c "select version();"
```

Output using Postgres 16:

```
                                                                 version
-------------------------------------------------------------------------------------------------------------------------------------
 PostgreSQL 16.2 (Debian 16.2.0-3.buster) (BigAnimal Edition) on x86_64-pc-linux-gnu, compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit
(1 row)
```

Migrate the database schema

On your source instance, use the `dt` command to view the details of the schema to be migrated:

```
/dt+;
```

Here is a sample database schema for this example:

```
                                          List of relations
 Schema |       Name       | Type  |   Owner   | Persistence | Access method |    Size    | Description
--------+------------------+-------+-----------+-------------+---------------+------------+-------------
 public | pgbench_accounts | table | edb_admin | permanent   | heap          | 1572 MB    |
 public | pgbench_branches | table | edb_admin | permanent   | heap          | 8192 bytes |
 public | pgbench_history  | table | edb_admin | permanent   | heap          | 0 bytes    |
 public | pgbench_tellers  | table | edb_admin | permanent   | heap          | 120 kB     |
```

Use pg_dump with the `--schema-only` flag to copy the schema from your source to your target instance. For more information on using `pg_dump`, see the Postgres documentation.

```
pg_dump --schema-only  -h <source_biganimal_host> -U <source_biganimal_username> -d <source_biganimal_databasename> | psql -h
<target_biganimal_host> -U <target_biganimal_username> -d <target_biganimal_databasename>
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

On the target instance, confirm the schema was migrated:

```
                                  List of relations
 Schema |       Name       | Type  |   Owner   | Persistence | Access method |  Size   | Description
--------+------------------+-------+-----------+-------------+---------------+---------+-------------
 public | pgbench_accounts | table | edb_admin | permanent   | heap          | 0 bytes |
 public | pgbench_branches | table | edb_admin | permanent   | heap          | 0 bytes |
 public | pgbench_history  | table | edb_admin | permanent   | heap          | 0 bytes |
 public | pgbench_tellers  | table | edb_admin | permanent   | heap          | 0 bytes |
```

> **Note**
>
> A successful schema-only copy shows the tables with zero bytes.

Create a publication

Use the `CREATE PUBLICATION` command to create a publication on your source instance. For more information on using `CREATE PUBLICATION`, see the Posgres documentation.

```
CREATE PUBLICATION
<pub_name>;
```

In this example:

```
CREATE PUBLICATION
v12_pub;
```

The expected output is: `CREATE PUBLICATION`.

Add tables that you want to replicate to your target instance:

```
ALTER PUBLICATION <pub_name> ADD TABLE
<table_name>;
```

```
ALTER PUBLICATION v12_pub ADD TABLE
pgbench_accounts;
ALTER PUBLICATION v12_pub ADD TABLE
pgbench_branches;
ALTER PUBLICATION v12_pub ADD TABLE
pgbench_history;
ALTER PUBLICATION v12_pub ADD TABLE
pgbench_tellers;
```

The expected output is: `ALTER PUBLICATION`.

Create the logical replication slot

Then, on the source instance, create a replication slot using the `pgoutput` plugin:

```
SELECT pg_create_logical_replication_slot('<slot_name>','pgoutput');
```

In the current example:

```
SELECT pg_create_logical_replication_slot('v12_pub','pgoutput');
```

The expected output returns the `slot_name` and `lsn`.

```
 pg_create_logical_replication_slot
------------------------------------
 (v12_pub,0/AC003330)
```

The replication slot tracks changes to the published tables from the source instance and replicates changes to the subscriber on the target instance.

Create a subscription

Use the `CREATE SUBSCRIPTION` command to create a subscription on your target instance. For more information on using `CREATE SUBSCRIPTION`, see the Postgres documentation.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
CREATE SUBSCRIPTION <name_of_subscription> CONNECTION 'user=<source_instance_username> host=<source_instance_read/write_host>
sslmode=require port=<source_instance_port> dbname=<source_instance_dbname> password=<source_instance_password>' PUBLICATION
<publication_name> WITH (enabled=true, copy_data = true, create_slot = false, slot_name=<slot_name>);
```

Creating a subscription on a Postgres 16 instance to a publication on a Postgres 12 instance:

```
CREATE SUBSCRIPTION v16_sub CONNECTION 'user=edb_admin host=p-x67kjhacc4.pg.biganimal.io sslmode=require port=5432 dbname=edb_admin
password=XXX' PUBLICATION v12_pub WITH (enabled=true, copy_data = true, create_slot = false, slot_name=v12_pub);
```

The expected output is: `CREATE SUBSCRIPTION`.

In this example, the subscription uses a connection string to specify the source database and includes options to copy existing data and to follow the publication identified by 'v12_pub'.

The subscriber pulls schema changes (with some exceptions, as noted in the PostgreSQLdocumentation on Limitations of Logical Replication) and data from the source to the target database, effectively replicating the data.

Validate the migration

To validate the progress of the data migration, use `dt+` from the source and target BigAnimal instances to compare the size of each table.

```
                                    List of relations
 Schema |       Name       | Type  |   Owner   | Persistence | Access method |    Size    | Description
--------+------------------+-------+-----------+-------------+---------------+------------+-------------
 public | pgbench_accounts | table | edb_admin | permanent   | heap          | 1572 MB    |
 public | pgbench_branches | table | edb_admin | permanent   | heap          | 8192 bytes |
 public | pgbench_history  | table | edb_admin | permanent   | heap          | 0 bytes    |
 public | pgbench_tellers  | table | edb_admin | permanent   | heap          | 120 kB     |
```

```
                                  List of relations
 Schema |       Name       | Type  |   Owner   | Persistence | Access method |  Size   | Description
--------+------------------+-------+-----------+-------------+---------------+---------+-------------
 public | pgbench_accounts | table | edb_admin | permanent   | heap          | 0 bytes |
 public | pgbench_branches | table | edb_admin | permanent   | heap          | 0 bytes |
 public | pgbench_history  | table | edb_admin | permanent   | heap          | 0 bytes |
 public | pgbench_tellers  | table | edb_admin | permanent   | heap          | 0 bytes |
```

If logical replication is running correctly, each time you run `\dt+;` you see that more data has been migrated:

```
                                  List of relations
 Schema |       Name       | Type  |   Owner   | Persistence | Access method |  Size   | Description
--------+------------------+-------+-----------+-------------+---------------+---------+-------------
 public | pgbench_accounts | table | edb_admin | permanent   | heap          | 344 MB  |
 public | pgbench_branches | table | edb_admin | permanent   | heap          | 0 bytes |
 public | pgbench_history  | table | edb_admin | permanent   | heap          | 0 bytes |
 public | pgbench_tellers  | table | edb_admin | permanent   | heap          | 0 bytes |
```

> **Note**
>
> You can optionally use LiveCompare to generate a comparison report of the source and target databases to validate that all database objects and data are consistent.

6.11 Analyzing your data with Apache Superset

You can use Apache Superset to analyze, explore, and visualize data stored in your Postgres clusters when using your own cloud account. (Apache Superset isn't supported on BigAnimal's cloud account.) You can open Superset from the BigAnimal portal using the **Analyze** links.

> **Note**
>
> Contact BigAnimal Support to enable the Apache Superset feature.

Default permissions include the ability to view data and create dashboards. To add or modify data sources, you need additional permissions. For more information on permissions to access Superset, see Managing Superset access.

Connecting Superset to your cluster

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

To connect to Superset, in addition to your password, you need your user, host, port, and database name. To find this information for your cluster:

1. Sign in to the BigAnimal portal.
2. Go to the Clusters page.
3. Select the name of your cluster.
4. Select the **Connect** tab.

> **Note**
>
> To connect with Superset in a private network database cluster, both the cluster and Superset must be on the same network.

To connect to a BigAnimal cluster:

1. Sign in to the BigAnimal portal.
2. Select **Analyze > Connections**.
3. Select **+ Database**.
4. In the Add Database dialog box, enter a value for **Database Name**.
5. To connect to the database, you need a database user with a password. Enter the connection string for your cluster in the **SQLALCHEMY URI** field, using the following format:

   ```
   postgresql://{<username>}:{<password>}@{<host>}:{<port>}/{<dbname>}?sslmode=verify-full
   ```

   > **Note**
   >
   > Your password is always encrypted before storage and never leaves your cloud environment. It's used only by the Superset software running in your BigAnimal infrastructure. As a defense-in-depth mechanism, we recommend using a Postgres user dedicated to Superset with a minimal set of privileges to just the database you're connecting. Never use your edb_admin superuser or equivalent user with Superset.

6. Check the connection by selecting **Test Connection**. Select **Add** if the connection was successful.

For more information on connecting to Superset, see the Superset documentation:

- Connecting Superset to a new database. Connecting Superset to a BigAnimal cluster is similar to connecting to any new database.
- Superset documentation on connecting to Postgres.

Upon successful connection, you can add datasets, charts, and dashboards.

Using Superset dashboards

You can use Superset dashboards to analyze data stored in your cluster. For a tutorial on creating a simple dashboard, see Creating Your First Dashboard.

To view all available Superset dashboards, select **Analyze > Dashboards**.

To create a dashboard for monitoring EDB Postgres Distributed, you can use the template we provide. See Configuring an EDB Postgres Distributed dashboard for more information.

Configuring an EDB Postgres Distributed dashboard

We provide a template for an EDB Postgres Distributed dashboard in JSON format (`utils/superset/pgd_monitoring_template.json`) in the cloud-utilities repository. The JSON file includes the schema of the dashboard and the individual charts.

To add the dashboard:

1. Clone the cloud-utilities repository on your local system.
2. Using Python 3.4 or later, create an output JSON file:
   1. Change your working directory:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
cd cloud-utilities/utils/superset
```

2. Change the permissions on the script to make it executable:

```
chmod +x db_name_change.py
```

3. Run the script:

```
./db_name_change.py <database_name> -i <input_file> -o <output_file>
```

For example:

```
./db_name_change.py edb -i utils/superset/pgd_monitoring_template.json  -o
utils/superset/upload.json
```

To get more information on the `db_name_change` script, run:

```
./db_name_change.py -h
```

3. In Superset, import your output file by selecting **Analyze > Dashboards > Import dashboard**.

Using Superset charts

You can use Superset charts to visualize data stored in your cluster. See the Superset documentation for instructions and examples of creating Superset charts.

To view all available Superset charts, select **Analyze > Charts**.

Using Superset SQL Lab

You can use Superset SQL Lab to write queries to access and modify data stored in your cluster. To access SQL Lab, select **Analyze > SQL Editor**.

6.12 Demonstration of Oracle SQL compatible functions and syntax

BigAnimal lets you run Oracle SQL queries in the cloud using EDB Postgres Advanced Server. This demonstration shows two Oracle SQL-syntax queries running unmodified on a BigAnimal test cluster, populated with the Chinook sample database.

Watch the video, or load up psql and follow along.

Connecting to the demo cluster with psql

You can use any recent version of psql to connect to EDB Postgres Advanced Server. The version that ships with Advanced Server has a few nice SQL*Plus compatibility features (with more availability in EDB*Plus). The queries and commands used here work the same in either version of psql. For convenience, these examples use the version of psql available in the EDB Postgres Advanced Server container image used by Cloud Native PostgreSQL and internally by BigAnimal. You can follow along by installing Docker and running:

```
docker pull quay.io/enterprisedb/edb-postgres-advanced
docker run --rm -it quay.io/enterprisedb/edb-postgres-advanced /bin/bash
```

```
[postgres@0fe3c244563c /]$
```

> **Note**

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

> If you prefer a graphical tool to execute Oracle-syntax-compatible queries or run Oracle PL/SQL-compatible code, we recommend pgAdmin.

The connection string for this demo's EDB Postgres Advanced Server cluster looks like this:

```
postgres://demo:password@p-vmpc7c40fm.pg.biganimal.io:5432/chinook?sslmode=require
```

In case you're unfamiliar with PostgreSQL connection URIs:

- `demo` is the user role you're connecting as. This is a user set up with select privileges on the database.
- `password` is the password for this user.

  > **Passwords in connection strings.**
  >
  > This example illustrates a complete connection URL, including the password. This is fine for a demonstration and might also be acceptable for application configuration if access to the configuration is limited. Avoid this practice for admin, superuser, or other roles used interactively. psql prompts for a password if none is supplied.

- `p-vmpc7c40fm.pg.biganimal.io` is the host name for the EDB Postgres Advanced Server cluster on BigAnimal that you're connecting to.
- `5432` is the usual PostgreSQL port number.
- `chinook` is the name of the database.
- `sslmode=require` ensures that you establish a secure connection.

With that in hand, launch psql:

```
psql postgres://demo:password@p-vmpc7c40fm.pg.biganimal.io:5432/chinook?sslmode=require
```

```
psql (14.2.1, server 13.6.10 (Debian 13.6.10-1+deb10))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

chinook=>
```

Here's the schema:

```
\dt
```

```
             List of relations
 Schema |     Name      | Type  |   Owner
--------+---------------+-------+-----------
 public | album         | table | edb_admin
 public | artist        | table | edb_admin
 public | customer      | table | edb_admin
 public | employee      | table | edb_admin
 public | genre         | table | edb_admin
 public | invoice       | table | edb_admin
 public | invoiceline   | table | edb_admin
 public | mediatype     | table | edb_admin
 public | playlist      | table | edb_admin
 public | playlisttrack | table | edb_admin
 public | track         | table | edb_admin
(11 rows)
```

An employee table is defined as follows:

```
\d+
employee
```

```
                                          Table "public.employee"
   Column   |            Type             | Collation | Nullable | Default | Storage  | Stats target | Description
------------+-----------------------------+-----------+----------+---------+----------+--------------+-------------
 employeeid | numeric                     |           | not null |         | main     |              |
 lastname   | character varying(20)       |           | not null |         | extended |              |
 firstname  | character varying(20)       |           | not null |         | extended |              |
 title      | character varying(30)       |           |          |         | extended |              |
 reportsto  | numeric                     |           |          |         | main     |              |
 birthdate  | timestamp without time zone |           |          |         | plain    |              |
 hiredate   | timestamp without time zone |           |          |         | plain    |              |
 address    | character varying(70)       |           |          |         | extended |              |
 city       | character varying(40)       |           |          |         | extended |              |
 state      | character varying(40)       |           |          |         | extended |              |
 country    | character varying(40)       |           |          |         | extended |              |
 postalcode | character varying(10)       |           |          |         | extended |              |
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
 phone      | character varying(24)       |           |          |         | extended |              |
 fax        | character varying(24)       |           |          |         | extended |              |
 email      | character varying(60)       |           |          |         | extended |              |
Indexes:
    "pk_employee" PRIMARY KEY, btree (employeeid)
Foreign-key constraints:
    "fk_employeereportsto" FOREIGN KEY (reportsto) REFERENCES employee(employeeid)
Referenced by:
    TABLE "customer" CONSTRAINT "fk_customersupportrepid" FOREIGN KEY (supportrepid) REFERENCES employee(employeeid)
    TABLE "employee" CONSTRAINT "fk_employeereportsto" FOREIGN KEY (reportsto) REFERENCES employee(employeeid)
Access method: heap
```

This table has a `reportsto` field. That means this is a hierarchical reporting structure, with some employees reporting to other employees who might in turn report to still other employees.

Demo #1: Exposing an organization hierarchy with `CONNECT BY`

Construct a hierarchical query to expose this chain of command.

Modern SQL can use a recursive CTE for this, as those are widely supported. But Oracle has, for decades, supported an alternative mechanism for querying hierarchy in the form of `CONNECT BY`, as shown in the following:

```
SELECT firstname, lastname,
(
    SELECT LISTAGG(lastname, ',
')
    FROM employee
rt
    START WITH rt.employeeid=e.reportsto
    CONNECT BY employeeid = PRIOR reportsto
    ) AS "chain of
command"
FROM employee
e;
```

```
 firstname | lastname | chain of command
-----------+----------+------------------
 Andrew    | Adams    |
 Nancy     | Edwards  | Adams
 Jane      | Peacock  | Edwards, Adams
 Margaret  | Park     | Edwards, Adams
 Steve     | Johnson  | Edwards, Adams
 Michael   | Mitchell | Adams
 Robert    | King     | Mitchell, Adams
 Laura     | Callahan | Mitchell, Adams
(8 rows)
```

Here, the `CONNECT BY` and the `LISTAGG` functions are used in a subquery to generate the chain of command for each employee: who they report to, who that person reports to, and so on.

The `LISTAGG()` function was introduced in Oracle 11g Release 2. Very few database systems support it. PostgreSQL *does* support `string_agg()`. In the previous example, that could be used as a drop-in replacement:

```
SELECT firstname, lastname,
(
    SELECT string_agg(lastname, ',
')
    FROM employee
rt
    START WITH rt.employeeid=e.reportsto
    CONNECT BY employeeid = PRIOR reportsto
    ) AS "chain of
command"
FROM employee
e;
```

```
 firstname | lastname | chain of command
-----------+----------+------------------
 Andrew    | Adams    |
 Nancy     | Edwards  | Adams
 Jane      | Peacock  | Edwards, Adams
 Margaret  | Park     | Edwards, Adams
 Steve     | Johnson  | Edwards, Adams
 Michael   | Mitchell | Adams
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
 Robert    | King     | Mitchell, Adams
 Laura     | Callahan | Mitchell, Adams
(8 rows)
```

But the semantics of the two functions are different for even slightly less-trivial uses, specifically when using the grouping construct, as shown in the next demonstration.

Demo #2: Group concatenation with `LISTAGG`

As shown in the first demonstration, this database has album and track tables containing metadata on digital recordings. You can use some analytic functions, including `LISTAGG`, to put together a report on average track storage requirements for albums with "baby" in the title.

```
SELECT UNIQUE
title,
       ROUND(AVG(bytes) OVER (PARTITION BY mediatypeid)/1048576 )
media_avg_mb,
       LISTAGG(t.name || ' (' || ROUND(bytes/1048576) || ' mb)',
chr(10))
         WITHIN GROUP (ORDER BY
trackid)
         OVER (PARTITION BY title)
track_list
FROM track
t
JOIN album USING (albumid)
JOIN mediatype USING (mediatypeid)
WHERE lower(title) LIKE '%baby%'
ORDER BY
title;
```

```
    title     | media_avg_mb |                    track_list
--------------+--------------+-------------------------------------------------
 Achtung Baby |            9 | Zoo Station (9 mb)                             +
              |              | Even Better Than The Real Thing (7 mb)         +
              |              | One (9 mb)                                     +
              |              | Until The End Of The World (9 mb)              +
              |              | Who's Gonna Ride Your Wild Horses (10 mb)      +
              |              | So Cruel (11 mb)                               +
              |              | The Fly (8 mb)                                 +
              |              | Mysterious Ways (8 mb)                         +
              |              | Tryin' To Throw Your Arms Around The World (7 mb)+
              |              | Ultraviolet (Light My Way) (10 mb)             +
              |              | Acrobat (8 mb)                                 +
              |              | Love Is Blindness (8 mb)
(1 row)
```

Trying to eplace `LISTAGG` with `string_agg` in this example fails. The expression syntax for `string_agg` is different.

```
SELECT UNIQUE
title,
       ROUND(AVG(bytes) OVER (PARTITION BY mediatypeid)/1048576 )
media_avg_mb,
       string_agg(t.name || ' (' || ROUND(bytes/1048576) || ' mb)', chr(10))
         WITHIN GROUP (ORDER BY
trackid)
         OVER (PARTITION BY title)
track_list
FROM track
t
JOIN album USING (albumid)
JOIN mediatype USING (mediatypeid)
WHERE lower(title) LIKE '%baby%'
ORDER BY
title;
```

```
ERROR:  function string_agg(text, text, numeric) does not exist
LINE 3:        string_agg(t.name || ' (' || ROUND(bytes/1048576) || ...
               ^
HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
```

This isn't difficult to correct, but it requires restructuring the query to replace the grouping construct. Such work can quickly accumulate errors. Fortunately, EDB Postgres Advanced Server supports `LISTAGG` in addition to `string_agg`, so this query doesn't need to change when migrating from Oracle.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Compatibility preserves the value of your existing work

In both of the examples shown here, you probably would not use the functions and syntax demonstrated for new work. There are better, more familiar or at least more widely available equivalents provided natively by PostgreSQL and many other databases. But by supporting them, EDB Advanced Server lets you reuse existing logic with minimal modification, allowing you to focus your time and expertise on solving new problems.

Try it on your own cluster: export and import

If you want to try these examples on your own BigAnimal cluster, follow these instructions to import the example database.

1. Connect to your EDB Postgres Advanced Server cluster as edb_admin using psql. You can get the command for this from your cluster's **Overview** tab. It looks like this, where `<YOUR CLUSTER HOSTNAME>` is specific to your cluster:

   ```
   psql -W "postgres://edb_admin@<YOUR CLUSTER HOSTNAME>:5432/edb_admin?sslmode=require"
   ```

   You're prompted for the password you specified when creating your cluster.

2. Create a new database and connect to it. You're prompted again for your cluster password:

   ```
   CREATE DATABASE
   chinook;
   \c chinook
   ```

3. Create a limited-privilege user for connecting to this database:

   ```
   CREATE USER demo WITH PASSWORD 'password';
   ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO demo;
   ```

   See Details on managing access in BigAnimal databases for more information.

4. Quit psql (`\q`).

5. To export and import, you need compatible versions of pg_dump and pg_restore. If you aren't already running EDB Postgres Advanced Server, you can use the container image used by Big Animal to ensure compatibility.

   ```
   docker pull quay.io/enterprisedb/edb-postgres-advanced
   docker run --rm -it quay.io/enterprisedb/edb-postgres-advanced /bin/bash
   ```

6. Export the Chinook sample database from EDB's cluster using pg_dump:

   ```
   pg_dump --format custom \
       "postgres://demo:password@p-vmpc7c40fm.pg.biganimal.io:5432/chinook?sslmode=require" \
       > /tmp/chinook.dump
   ```

7. Get the host name of your cluster from the **Connect** tab on the BigAnimal portal. Use it in place of `<YOUR CLUSTER HOSTNAME>` to invoke pg_restore:

   ```
   pg_restore --no-owner \
       -d "postgres://edb_admin@<YOUR CLUSTER HOSTNAME>:5432/chinook?sslmode=require" \
       /tmp/chinook.dump
   ```

   > **Note**
   >
   > You might see an error about pg_stat_statements. You can safely ignore this error. pg_stat_statements is a very useful extension and is installed by default on BigAnimal clusters. However, since you're connecting as the admin user and not a superuser, you can't modify it. The rest of the schema and data is restored, however.

8. Finally, connect to this database:

   ```
   psql "postgres://demo:password@<YOUR CLUSTER HOSTNAME>:5432/chinook?sslmode=require"
   ```

Now you can try some queries on your own cluster.

Next steps

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- Read more on Oracle compatibility features in the EDB Advanced Server documentation.
- Learn about migrating existing databases to BigAnimal.

6.13 BigAnimal Terraform provider

BigAnimal's Terraform provider is an infrastructure-as-code service that allows you to provision cloud resources with the Terraform CLI and incorporate those resources into your existing BigAnimal cloud infrastructure workflows.

The current version of the Terraform provider offers resources and data sources for creating, reading, updating, and deleting clusters and regions.

The current version of the Terraform provider offers resources and data sources for:

- Creating, updating, and deleting clusters.
- Creating, updating, and deleting faraway replicas. Promoting faraway replica isn't supported in the current version.
- Activating and deactivating regions.
- Creating projects. Connecting the cloud service provider isn't supported in the current version.

The Terraform provider is licensed under the MPL v2.

> **Note**
>
> We provide support for the BigAnimal Terraform provider and not for the underlying environment. To report suspected defects or to submit feature requests, open a GitHub issue using the guidance found here.

Prerequisites

To use Terraform with BigAnimal, you need:

- A BigAnimal account with an organization set up. If you don't already have a BigAnimal account, see Getting started with BigAnimal.
- Terraform (version 0.13*x* or later) downloaded and installed.
- A BigAnimal API token for use within the Terraform application. See Getting an API Token.

Example usage

```
# Configure the BigAnimal Provider
provider "biganimal" {
  ba_bearer_token = "<redacted>"
  // ba_access_key: if set, this will be used instead of the ba_bearer_token above.
  // This can also be set as an environment variable. If it is set both here and
  // in an environment variable then the access key set in the environment variable
  // will take priority and be used
  ba_access_key = "<redacted>"
}
# Manage the resources
```

Environment Variables

Credentials can be provided by using the `BA_BEARER_TOKEN` or `BA_ACCESS_KEY` environment variables.

Schema

Optional

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- `ba_access_key` (String) BigAnimal Access Key
- `ba_bearer_token` (String) BigAnimal Bearer Token

Getting an API token

To use the BigAnimal API, fetch an API bearer token and export it into your environment. (For additional information about using the BigAnimal API, seehere.) Optionally, credentials can be provided by using the `BA_API_URI` environment variable.

1. Access the script located here.
2. Open the script in `Raw` format.
3. Copy the script and save it locally with the name `get-token.sh`.
4. Modify permissions for the script in your local shell.
5. Run the script locally using a command like the following:

   ```
   sh <local path>/get-token.sh
   ```

   The resulting output instructs you to log in to a URL with an 8-digit user code. For example:

   ```
   Please login to https://auth.biganimal.com/activate?user_code=JWPL-RCXL with your BigAnimal account
   ```

6. In a browser, access the URL, confirm, and reauthenticate if necessary. A notice lets you know that the code was verified.
7. In your local shell, a prompt asks:

   ```
   Have you finished the login successfully. (y/n)
   ```

8. Enter `y`. The shell responds with output that provides the access token, refresh token, scope, expiration period, and token type.
9. Export the access token into your environment as follows, replacing `<REDACTED>` with the access token:

   ```
   export BA_BEARER_TOKEN=<REDACTED>
   ```

   Rather than export the token, you can alternatively use the token to set the value of the `ba_bearer_token` when configuring the BigAnimal provider, as shown inExample usage.
10. Now you can follow along with the examples in the Terraform repository.

6.14 PGD CLI on BigAnimal

When running a distributed high-availability cluster on BigAnimal, you can use thePGD CLI to manage cluster operations. Examples of these operations include switching over write leaders, performing cluster health checks, and viewing various details about nodes, groups, or other aspects of the cluster.

Installing the PGD CLI

To install the PGD CLI, for Debian and Ubuntu machines, replace `<your-token>` with your EDB subscription token in the following command:

```
curl -1sLf 'https://downloads.enterprisedb.com/&lt;your-token&gt;/postgres_distributed/setup.deb.sh' | sudo -E bash
sudo apt-get install edb-pgd5-cli
```

For RHEL, Rocky, AlmaLinux, or Oracle Linux machines, make the replacement in this command:

```
curl -1sLf 'https://downloads.enterprisedb.com/&lt;your-token&gt;/postgres_distributed/setup.rpm.sh' | sudo -E bash
sudo yum install edb-pgd5-
cli
```

Connecting to your BigAnimal cluster

Discovering your database connection string

To connect to your distributed high-availability BigAnimal cluster using the PGD CLI, you need todiscover the database connection string. From your BigAnimal console:

1. Log in to the BigAnimal clusters view.
2. To show only clusters that work with PGD CLI, in the filter, set **Cluster Type** to **Distributed High Availability**.
3. Select your cluster.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

4. In the view of your cluster, select the **Connect** tab.
5. Copy the read/write URI from the connection info. This is your connection string.

Using the PGD CLI with your database connection string

> **Important**
>
> PGD doesn't prompt for interactive passwords. Accordingly, you need a `.pgpass` file properly configured to allow access to the cluster. Your BigAnimal cluster's connection information page has all the information needed for the file.
>
> Without a properly configured `.pgpass`, you receive a database connection error when using a PGD CLI command, even when using the correct database connection string with the `--dsn` flag.

To use the PGD CLI with your database connection string, use the `--dsn` flag with your PGD CLI command:

```
pgd show-nodes --dsn "<your_connection_string>"
```

PGD commands in BigAnimal

> **Note**
>
> Three EDB Postgres Distributed CLI commands don't work with distributed high-availability BigAnimal clusters: `create-proxy`, `delete-proxy`, and `alter-proxy-option`. These commands are managed by BigAnimal, as BigAnimal runs on Kubernetes. It's a technical best practice to have the Kubernetes operator handle these functions.

The examples that follow show the most common PGD CLI commands with a BigAnimal cluster.

`pgd check-health`

`pgd check-health` provides statuses with relevant messaging regarding the clock skew of node pairs, node accessibility, the current raft leader, replication slot health, and versioning consistency:

```
$ pgd check-health --dsn "postgres://edb_admin@p-mbx2p83u9n-a.pg.biganimal.io:5432/bdrdb?sslmode=require"
```

```
Check      Status  Message
-----      ------  -------
ClockSkew  Ok      All BDR node pairs have clockskew within permissible limit
Connection Ok      All BDR nodes are accessible
Raft       Warning There is no RAFT_LEADER, an election might be in progress
Replslots  Ok      All BDR replication slots are working correctly
Version    Ok      All nodes are running same BDR versions
```

`pgd show-nodes`

`pgd show-nodes` returns all the nodes in the distributed high-availability cluster and their summaries, including name, node id, group, and current/target state:

```
$ pgd show-nodes --dsn "postgres://edb_admin@p-mbx2p83u9n-a.pg.biganimal.io:5432/bdrdb?sslmode=require"
```

```
Node             Node ID    Group          Type Current State Target State Status Seq ID
----             -------    -----          ---- ------------- ------------ ------ ------
p-mbx2p83u9n-a-1 3537039754 dc1            data ACTIVE        ACTIVE       Up     1
p-mbx2p83u9n-a-2 3155790619 p-mbx2p83u9n-a data ACTIVE        ACTIVE       Up     2
p-mbx2p83u9n-a-3 2604177211 p-mbx2p83u9n-a data ACTIVE        ACTIVE       Up     3
```

`pgd show-groups`

`pgd show-groups` returns all groups in your distributed high-availability BigAnimal cluster. It also notes the node that's the current write leader of each group:

```
$ pgd show-groups --dsn "postgres://edb_admin@p-mbx2p83u9n-a.pg.biganimal.io:5432/bdrdb?sslmode=require"
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
 Group          Group ID   Type   Parent Group   Location Raft  Routing Write Leader
 -----          --------   ----   ------------   -------- ----  ------- ------------
world           3239291720 global                         true  true    p-mbx2p83u9n-a-1
dc1             4269540889 data   p-mbx2p83u9n-a          false false
p-mbx2p83u9n-a  2800873689 data   world                    true  true    p-mbx2p83u9n-a-3
```

`pgd switchover`

`pgd switchover` manually changes the write leader of the group and can be used to simulate afailover.

```
$ pgd switchover --group-name world --node-name p-mbx2p83u9n-a-2 --dsn "postgres://edb_admin@p-mbx2p83u9n-
a.pg.biganimal.io:5432/bdrdb?sslmode=require"
```

```
switchover is complete
```

See the PGD CLI command reference for the full range of PGD CLI commands and their descriptions.

6.15 Postgres Distributed (PGD) presets on BigAnimal

Commit scope

Commit scopes in PGD are a set of rules that describe the behavior of the system as transactions are committed. Because they define how transactions are replicated across a distributed database, they have an effect on consistency, durability, and performance.

The actual behavior depends on the kind of commit scope a commit scope's rule uses:Group Commit, Commit At Most Once, Lag Control, Synchronous Commit, or a combination of these.

This flexibility means that selecting a balanced combination of rules can take time. To speed up deployment, BigAnimal's PGD has a preset selection of commit scopes for typical user requirements. These presets don't prevent you from creating and applying your own commit scopes as needed.

BigAnimal's commit scope preset options

The presets include a rule for high consistency in a subgroup and a separate setting for lag control both inside and outside the subgroup. You can use combinations of these rules for high consistency in a subgroup. You can also use lag control for other subgroups and high consistency within the subgroup and with one or more nodes of another subgroup.

Preset ID	Local group	External groups	Use case
ba001	MAJORITY, SYNCHRONOUS_COMMIT	N/A	High consistency for critical data
ba002	ALL, LAG_CONTROL	ALL, LAG_CONTROL	Manages replication lag locally and externally
ba003	MAJORITY, SYNCHRONOUS_COMMIT	ANY 1, LAG_CONTROL	Combines high consistency locally with lag control for external
ba004	MAJORITY, SYNCHRONOUS_COMMIT	ANY 1, SYNCHRONOUS COMMIT	Higher consistency via external validation

> **Note**
>
> ba001 is also the only BigAnimal preset that works with PGD clusters that have only one data group. All of the presets, including ba001, work with PGD clusters that have two or more data groups.

ba001

ba001 offers a default setup that optimizes data consistency and reliability across distributed environments. This example shows a definition for transactions originating from the group of nodes `location-a`. The full name of the actual commit scope for this specific group is `ba001_location-a`.

```
SELECT
bdr.add_commit_scope(
    commit_scope_name := 'ba001_location-a',
    origin_node_group := 'location-a',
    rule := 'MAJORITY (location-a)
SYNCHRONOUS_COMMIT',
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
    wait_for_ready :=
true
);
```

This rule says that for transactions originating from a node in `location-a`, the majority of the nodes in `location-a` must acknowledge the transaction before it's committed.

ba001 uses `MAJORITY` instead of `ALL` so that in the case of 3 data nodes, the third can be updated asynchronously. This ability allows for a node failure without interrupting a single region service.

ba001 is the baseline commit scope provided by BigAnimal and the default commit scope for all new subgroups.

ba002

ba002 allows for asynchronous commits but with lag control for tighter consistency (that is, tighter than the system's default asynchronous behavior) across the distributed system. The following is an example definition of ba002 for nodes in `location-a`:

```
SELECT
bdr.add_commit_scope(
    commit_scope_name := 'ba002_location-a',
    origin_node_group := 'location-a',
    rule := 'ALL (location-a) LAG_CONTROL (max_commit_delay=1s, max_lag_size=100MB) AND ALL NOT (location-a) LAG_CONTROL
(max_lag_size=1000MB, max_commit_delay=4s, max_lag_time=30s)',
    wait_for_ready :=
true
);
```

The first part of the rule says that all nodes of `location_a`, which is the `origin_node_group` in this case, are monitored according to `max_lag_size=100MB` and `max_commit_delay=1s`. Specifically, as lag size approaches the set max of 100MB, the system begins injecting commit delays on the origin node of up to 1 second so that replication to the other nodes in the group can catch up.

The second part of the rule (after `AND`) says that for all nodes not in `location_a`, those nodes are monitored according to `max_lag_size=1000MB`, `max_lag_time=30s`, and `max_commit_delay=4s`. Specifically, as lag size or time approaches either of their respective limits (1000MB or 30s), the system begins injecting commit delays on the origin node of up to 4 seconds. This capability allows replication to those nodes outside the origin node group to catch up.

ba003

ba003 is a combination of Synchronous Commit for the local group with Lag Control parameters being met by at least one external node.

This example definition of a ba003 commit scope uses the `bdr.add_commit_scope` command, again applying to nodes in `location-a`:

```
SELECT
bdr.add_commit_scope(
    commit_scope_name := 'ba003_location-a',
    origin_node_group := 'location-a',
    rule := 'MAJORITY (location_a) SYNCHRONOUS_COMMIT AND ANY 1 NOT (location_a) LAG_CONTROL (max_lag_size=1000MB,
max_commit_delay=4s, max_lag_time=30s)',
    wait_for_ready :=
true
);
```

The first part of the rule states that for any transaction originating on a node in the `origin_node_group` (in this case `location-a`), a majority of the nodes in `location_a` must confirm the transaction before it's committed.

The second part of the rule states that at least one node not in `location_a` must keep lag within the `LAG_CONTROL` parameters (`max_lag_size=1000MB, max_lag_time=30s`) with respect to its replicating transactions originating from a node in the `origin_node_group` (here location_a). Otherwise, a commit delay of up to 4 seconds will be used to slow down processing on the originating node to allow for the lagging node to catch up.

ba004

```
SELECT
bdr.add_commit_scope(
    commit_scope_name := 'ba004_location-a',
    origin_node_group := 'location-a',
    rule := 'MAJORITY (location_a) SYNCHRONOUS_COMMIT AND ANY 1 NOT (location_a)
SYNCHRONOUS_COMMIT',
    wait_for_ready :=
true
);
```

ba004 says that for all transactions originating from a node in the `origin_node_group` (`location-a` in the example), a majority of the nodes of `location-a`, and at least one node

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

not in `location-a`, must confirm the transaction before it's committed.

Just as in ba001 and ba003, ba004 uses `SYNCHRONOUS_COMMIT` to keep potential data loss at near zero, as the data is replicated before commit success is signaled to the application. Again, `MAJORITY` is used so that in a three-node scenario in a region, one node can experience failure without compromising the cluster. However, to confirm the transaction, in this case the commit scope also requires confirmation from one node outside of the group.

Setting default_commit_scope

As mentioned, the default BigAnimal PGD commit scope preset is an instantiation of ba001.

To change the default commit scope, you need to know the names of your commit scope presets and the name of the origin node group for your cluster. To see the BigAnimal presets as defined for your cluster's groups, connect to the cluster using psql, and enter:

```
SELECT * FROM bdr.commit_scopes
```

```
 commit_scope_id |  commit_scope_name   | commit_scope_origin_node_group |
commit_scope_rule
-----------------+----------------------+--------------------------------+-------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
      1131842441 | ba001_p-mbx2p83u9n-a |                     2800873689 | MAJORITY (p-mbx2p83u9n-a) SYNCHRONOUS_COMMIT
      2997404763 | ba002_p-mbx2p83u9n-a |                     2800873689 | ALL (p-mbx2p83u9n-a) LAG CONTROL (max_commit_delay=1s,
max_lag_size=100MB) AND ALL NOT (p-mbx2p83u9n-a) LAG CONTROL (max_lag_size=1000MB, max_commit_delay=4s, max_lag_time=30s)
       671768582 | ba003_p-mbx2p83u9n-a |                     2800873689 | MAJORITY (p-mbx2p83u9n-a) SYNCHRONOUS_COMMIT AND ANY 1 NOT
(p-mbx2p83u9n-a) LAG CONTROL (max_lag_size=1000MB, max_commit_delay=4s, max_lag_time=30s)
      1568192748 | ba004_p-mbx2p83u9n-a |                     2800873689 | MAJORITY (p-mbx2p83u9n-a) SYNCHRONOUS_COMMIT AND ANY 1 NOT
(p-mbx2p83u9n-a) SYNCHRONOUS_COMMIT
(4 rows)
```

Note the `commit_scope_name` of the preset you want to set as your default. The origin node group follows `ba00<#>_`. In this case, it's `p-mbx2p83u9n-a`.

To make a commit_scope the `default_commit_scope` for a node group, use the command `bdr.alter_node_group_option`. Replace `<origin_node_group>` with the origin node group (in this case `p-mbx2p83u9n-a`). Replace `<commit_scope_name>` with the `commit_scope_name` of the desired preset.

```
SELECT bdr.alter_node_group_option(
  node_group_name := '<origin_node_group>',
  config_key := 'default_commit_scope',
  config_value := '<commit_scope_name>'
);
```

> **Note**
>
> Commit scopes can be applied per transaction. In that case, they override a BigAnimal preset.

For more information, see Commit scopes and Commit scope rules in the PGD documentation.

6.16 Tagging BigAnimal resources

BigAnimal provides a shared tags system that allows you to assign and manage tags for resources across different resource types.

The tags are assigned to the following resource types:

- Project
- Cluster

Key features of shared tags system are:

- **Shared tags** — You can create shared tags that are accessible and applicable to all the supported resource types. Shared tags provides a standard way to categorize and organize resources. Tags are scoped to the organizations and shared among all users in the organization. The access to tags is controlled by existing permission system.
- **Tag assignment** — You can assign one or more shared tags to individual resource items. This capability enables you to categorize resources effectively and according to the needs.
- **Tag permissions** — The tag management includes permission settings to control who can create, edit, or delete shared tags. This ensures security and control over the tagging system.

6.16.1 Creating and managing tags

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal supports tagging of the following resources:

- Project
- Cluster

Create and assign a tag

You can create a tag using either of these methods:

- Create a tag and assign it to a resource
- Create and assign a tag while creating a resource

Create a tag and assign it to a resource

To create a tag:

1. Log in to the BigAnimal portal.
2. From the menu under your name in the top right of the panel, select **Tags**.
3. From the Create Tag page, select **Create Tag**.
4. Enter the tag name.
5. Select the color for the tag.
6. View the tag in **Preview**.
7. Select **Save**.

The generated tag is available for the user.

To assign a tag to an existing cluster:

1. Go to cluster's home page.
2. In the clusters list, select the edit icon next to the cluster.
3. On the Edit Cluster page, go to the **Cluster Settings** tab.
4. Under **Tags**, select **+**.
5. In the search bar, enter the name of the tag and select the tag.
6. To assign the tag, select **Save**.

To assign a tag to an existing project:

1. Go to the project's home page.
2. In the projects list, select the edit icon next to the project.
3. On the Edit Project page, under **Tags**, select **+**.
4. In the search bar, enter the tag name and select the tag.
5. To assign the tag, select **Save**.

Create a tag while creating a resource

Create and assign a tag while creating a project and creating a cluster.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Edit a tag

1. Log in to the BigAnimal portal.
2. From the menu under your name in the top right of the panel, select **Tags**.
3. On the Tags page, select the edit button next to the tag name.
4. Edit the tag name and color.
5. Select **Save**.

Delete a tag

1. Log in to the BigAnimal portal.
2. From the menu under your name in the top right of the panel, select **Tags**.
3. On the Tags page, select the delete icon next to the tag name.

   You're prompted to type **delete tag** in the field.
4. To delete the tag, enter the text as instructed, and select **Yes, Delete tag**.

7 Administering your account

Administrative activities for the BigAnimal account include portal and database user access management as well as account activity reviews and customizing cloud provider policy definitions.

If you coordinated with BigAnimal Support to enable the Apache Superset feature, see Managing Superset access for information on setting up roles and permission for access to the Superset data sources.

7.1 Managing user access

BigAnimal uses role-based access controls to grant users access to different parts of the application using organization-level and project-level roles. Roles are sets of permissions. BigAnimal uses roles to manage permissions assigned to users.

Each customer has a unique BigAnimal organization. Each organization has at least one project by default. An organization's database clusters are deployed and managed in the customer's cloud account within a project. You can create multiple projects within a single organization.

With multiple projects within an organization you can:

- Separate workflows to provide secure and isolated environments.
- Assign different users to different projects or give different roles to users in different projects to ensure they have the correct level of permissions.

To access a BigAnimal organization, first each user needs to be added to the organization's identity provider. For more information, see Setting up your identity provider. Your identity provider establishes the identity of users that can log in to BigAnimal.

> **Note**
>
> You can invite people to your organization by copying a link from the Users page in the BigAnimal portal and sending it to their email. They'll need to create an EDB account if they don't have one already.

Once a user has logged in to BigAnimal, you can assign them a role.

Organization-level roles

The following roles grant privileges within an organization.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- **Organization owner** — This role has management privileges to the organization and can perform the following actions:
    - Create and view projects within their organization
    - Update and delete their own projects
    - View and assign organization-level and project-level roles
    - View an activity log for the whole organization and each project
    - View and download a usage report for the whole organization and each project
    - View the identity provider details

    > **Notes**
    >
    > - The first user in a BigAnimal organization is an organization owner and project owner of the initial project, by default.
    > - At least one user must be an organization owner.

- **Organization admin** — This role has read-only permissions to the organization. They can:
    - View a list of projects within the organization
    - View and download a usage report for the whole organization
    - View other users with organization-level roles
    - View the identity provider details of the BigAnimal subscription

Project-level roles

The following roles grant privileges within a project:

- **Project owner** — This role has management privileges to the project and can perform the following actions within the project:
    - Connect the cloud service provider accounts to BigAnimal
    - View, edit, and delete the project
    - Create, view, edit, and delete clusters
    - Activate, suspend, and deactivate regions
    - View and assign project-level roles
    - View an activity log
    - View and download a usage report

    > **Note**
    >
    > At least one user must be a project owner.

- **Project editor** — This role has edit privileges to the project and can perform the following actions within the project:
    - View the cloud service provider accounts connected to BigAnimal
    - Create, view, edit, and delete clusters
    - Activate regions
    - View users with project-level roles
    - View an activity log
    - View and download a usage report

- **Project viewer** — This role has read-only permissions to the project. They can:
    - View clusters
    - View users with project-level roles

Users

Organization owners can assign users organization-level roles to complete certain tasks:

1. From the menu next to your organization name in the top right of the portal, select**User Management**.
2. Select the edit icon for the user.
3. Select **Assign Roles**.
4. Select the roles for the user.
5. Select **Submit**.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

See Adding a user to a project for information on adding users to projects.

Machine users

A machine user is used to drive access to BigAnimal through an approach other than using the UI. This access can occur by way of user-customized automation scripts, BigAnimal CLI commands, Terraform manifest, or BigAnimal's first-party binaries that run from your environment.

- Any user with the organization's owner role can create, update, and delete a machine user.
- The organization owner can be a user authenticated by way of IDP or a machine user.
- An organization owner can manage the access key for the machine user but not for the normal user.
- The organization owner and project owner can assign and unassign the roles to any machine user in the same organization.
- BigAnimal's authorization system performs permission checks on machine users as well as normal users.
- The machine user is associated with only one organization and can't switch to another organization.
- The machine user can't be invited, and the only way to authenticate and authorize a machine user is with an access key.

> **Note**
>
> There is quota limit for the number of machine users if you hit it and need more, please contact our support team.

Add machine user

Only the organization owner can add a machine user.

To add a machine user:

1. From the menu next to your organization name in the top right of the portal, select**User Management**.
2. Select **Add New User**.
3. Select **Machine User** as the user type.
4. Provide the **Username**.
5. Optionally, provide the **Email ID**.
6. Optionally, select the **Create Access Key** check box.
    - If you select the check box, an access key is created for this user.
    - If you don't select the check box, no access key is created while adding this user. You can create the access key for this user from the user's home page later.
7. Provide the **Access Key Name**.
8. Select the **Expires in (Day/s)** value for the access key.
9. To save the settings and provide the generated access key for the user, select**Add User**.

Copy and this access key and save it in a secure location. The access key is available only when you create it. If you lose your access key, you must delete it and create a new one. For more information, see Access key.

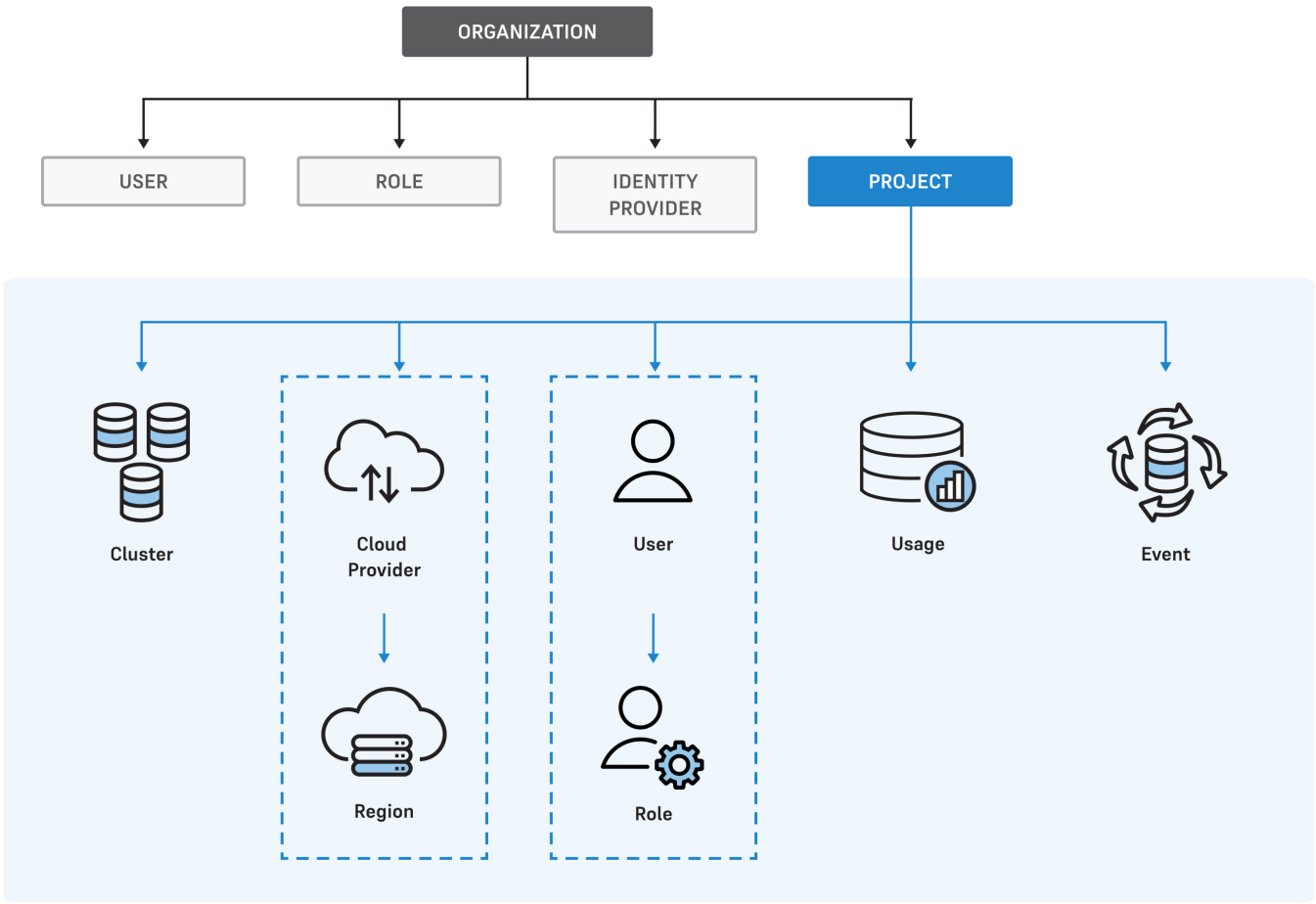
Assign some organization role or project role to this newly created machine user. For more information, see users.

> **Note**
>
> The user management on BigAnimal's UI at project level is used to assign the project role to the machine user, and not for managing the machine users and their access key.

7.2 Managing projects

Projects are a group of database clusters within a BigAnimal organization. Access to projects is controlled by a user's role. With projects, you can separate workflows to provide secure and isolated environments and assign different users to different projects. See Managing portal access for more information.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.



Projects actions are available from the left navigation bar. The actions apply to the project in focus. You can switch between projects using the **Projects** menu.

Adding a user to a project

Before adding a user to a project:

- The user must be added to the organization through the organization's identity provider.
- The user must log in to the BigAnimal portal at least once.
- You must be an organization owner or the project owner.

To add a user:

1. From the **Projects** menu, select the project you want to add a user to.
2. From the left navigation bar, select **Users**.
3. Select the edit icon for the user.
4. Select **Assign Roles**.
5. Depending on the level of access you want for the user, select the appropriate role.
6. Select **Submit**.

You can enable in-app inbox or email notifications to get alerted when a user is invited to a project. For more information, see managing notifications.

Creating a project

Before creating a project:

- You must have an organization owner role.
- You must create a new Azure subscription or AWS account. For Google Cloud, you don't need to do so.
    - To create a new Azure subscription, see Create a Microsoft Customer Agreement subscription.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- To create a new AWS account, see Creating an AWS account in your organization.

> **Notes**
>
> - You can use a single Azure subscription or AWS account for only one project at a time. For example, if you have two projects and only need to deploy your clusters in AWS, you need to connect a different AWS account for each project.
> - To reuse an Azure subscription or AWS account, you must first delete the project that was previously connected with that account.

- For Google Cloud, you must create a new Google Cloud project for each BigAnimal project. To create a new Google Cloud project, see Creating and managing projects.

> **Note**
>
> Google Cloud assigns a quota to the number of Google Cloud projects based on your account type. If your account has reached its limit, you can complete a Request Project Quota Increase form or schedule for some projects to be deleted.

To create a new project:

1. Select **Projects > Create New Project**.
2. In the **Project Name** field, enter a unique name.
3. Optionally, under **Tags*, select **+**.
4. To assign an existing tag, in the search bar under **Tags**, enter a tag name. To add a new tag, instead select **+ Add Tag**.
5. Select **Create New Project**.
6. Select **Project > See All Projects**.
7. Select your new project.
8. Set up the cloud provider for the project. See Connecting your cloud.
9. Activate the region for the project. See Activating regions.
10. Select **Create Project**.

Editing a project

To edit a project:

1. On the Projects page, in the projects list, select the edit icon next to the project.
2. Enter a new name.
3. Select **Update**.

Deleting a project

> **Note**
>
> Contact BigAnimal Support if you want to delete the initial default project created by BigAnimal.

To delete a project that you created:

1. On the Projects page, in the projects list, select the edit icon next to the project.
2. Delete any remaining clusters in the project.
3. Delete the regions for the project.
4. Select **Yes, Delete Project**.

Adding a TDE key

1. From the Projects page, select an existing project.
2. Go to **Settings** on the left-side navigation.
3. From the **Settings** list, select **Security**.
4. Select **Add a key**.
5. On the **Add a key** page, select:
   - Select the **Provider & Region**
   - Select **Next**
   - Enter AWS Key Management System ARN
   - Enter a friendly name for your key
   - Select **Finish**

Now, use this TDE key to create a cluster. For more information, see Creating a cluster.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Deleting a TDE key

1. From the Projects pageXOffset, select an existing project.
2. Go to **Settings** on the left-side navigation.
3. From the **Settings** list, select **Security**.
4. Select a key and select the **delete** icon for the key you want to delete.

   You're prompted to type **delete key name** in the field.
5. To delete the key, enter the text as instructed and select **Yes, Delete key**.

> **Note**
>
> A TDE key is deleted only when it isn't used and not associated with any cluster.

7.3 Managing Superset access

> **Note**
>
> If you're using BigAnimal's cloud account, you don't have access to Superset.

You control access to Superset data sources by using Superset roles and permissions.

Superset has three roles mapped to BigAnimal roles:

- Gamma
- Alpha
- Admin

Learn more about Superset roles, users, and permissions management in Superset Security.

The Superset roles map to BigAnimal permissions.

Superset role	Description	BigAnimal permissions mapped to Superset role
Gamma	View data that user has been granted access to, create dashboards	Project viewer
Alpha	All Superset Gamma privileges, plus the ability to add or modify data sources	Project editor
Admin	All Superset Alpha privileges, plus access to SQL Lab and the ability to grant or revoke access to data to other users	Project owner

> **Notes**
>
> - Access to Superset is currently limited to the initial default project set up by BigAnimal. The user needs to have a project role for the initial project to access Superset.
> - While Admin users have access to all databases by default, both Alpha and Gamma users need to be given access by way of the Superset sql_lab role on a per-database basis. The sql_lab role grants access to SQL Lab.

To assign BigAnimal user roles, see Users.

7.4 Reviewing account activity

The activity logs collect BigAnimal events based on user activity in the portal. There's an organization-level log of available users with organization roles and a project-level log. You can use the log to audit activities performed by users from your organizations or research activities that might have affected your account at either the organization level or the project level.

Events

Events describe actions performed by users. The available actions are:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- Create
- Read
- Update
- Delete

Events are related to the following resource types:

- Cluster
- Data plane
- User
- User roles
- Role permissions
- Organization
- Project

> **Note**
>
> Database events don't log activity on the Postgres server. They log the use of the portal to create or modify database clusters.

View and search the activity log

To view events, navigate to the Activity Log page on the BigAnimal portal.

- If you're an organization owner or admin, the organization-level activity log is available from the menu next to your organization name in the top right of the portal.
- The project-level activity log is available from the **Project** menu on the left of the portal.

To search events, use the filters at the top of the page.

The following fields are in the activity logs.

Field	Description
Projects	Projects in the organization (available only from the organization log)
Activity Name	Name of an event in the format *Action Resource-Type, Resource-name*
User	User responsible for the event
Date	Date when the action was performed
Resource	Resource type of the resource

7.5 Customizing compliance rules and policies

Your cloud provider has rules and policies to help you to monitor, identify, and remediate noncompliant resources. You can customize the default policies and rules to match BigAnimal's resource configurations.

7.5.1 Customizing AWS Config rules

AWS Config rules represent desired configuration settings for AWS resources and help you to monitor, identify, and remediate noncompliant ones. AWS Security Hub leverages AWS Config by introducing dedicated sets of AWS Config security rules associated with several security standards. It aggregates findings from rule violations and other AWS or third-party services.

For more information, see:

- What Is AWS Config?
- What is AWS Security Hub?

BigAnimal doesn't customize your AWS Config rules to prevent conflicts with external workloads.

7.5.2 Customizing Azure policy definitions

Azure policies help you to monitor, identify, and remediate noncompliant resources. Azure assigns a default set of policies to each subscription. If needed, you can customize these default

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Azure policies to match BigAnimal's resource configurations.

> **Note**
>
> BigAnimal doesn't customize your Azure policies to prevent conflicts with external workloads.

Customize default policy definitions in Azure

Customize the policy definitions in each of your BigAnimal-enabled Azure subscriptions.

> **Note**
>
> You need `Microsoft.Authorizations/PolicyAssignments/write` permissions to update policy initiatives (sets of policies) in Azure.

1. In the Azure portal, enter `Policy` in the search box at the top, and open the Policy service.
2. On the left side of the Policy page, Select**Compliance**.
3. On the Compliance page, set the scope by selecting the ellipsis and then selecting all subscriptions. At the bottom of the Scope page, select**Select** to add your selection.

   You can see a list of all the policy initiatives (sets of policies) assigned by Azure's onboarding process. The policy initiative for each subscription is labeled **ASC Default (subscription: <Subscription_ID>)**.
4. From the list, select a policy initiative, and select**Edit assignment**.
5. On the Edit Initiative Assignment page, select the**Parameters** tab.
6. Clear the **Only show parameters that need input or review** check box.
7. Configure your default ASC policy parameters to allow only BigAnimal's specific configurations. Use the parameter values specified inCustomizable policy definition parameters to update the parameters.
8. Select the **Review + create** tab at the top of the wizard.
9. Review your selections. At the bottom of the page, select**Create**.

You're now ready to monitor, identify, and remediate noncompliant resources to improve the compliance state of the resources in your subscription.

Customizable policy definition parameters

The following are the recommended parameters and values that are based on BigAnimal's resource configurations.

Use the values below each parameter while configuring the default ASC policy of a subscription.

> **Note**
>
> JSON values are provided where applicable.

Allowed service ports list in Kubernetes cluster

BigAnimal runs services on several ports in Kubernetes clusters in your cloud account to provide the BigAnimal services. You must allow the following ports:

```
["5432",  "9402",  "443", "8080",
 "9090",  "3000",  "8443",  "9443",  "9100", "9201", "8088"]
```

Allowed AppArmor profiles

BigAnimal requires the `runtime/default` AppArmor security profile:

```
["runtime/default"]
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Allowed capabilities

Restrict the capabilities to reduce the attack surface of containers in a Kubernetes cluster. BigAnimal generally runs containers with limited capability to limit the attack surface of Kubernetes clusters but requires some capabilities to function:

```
["FOWNER"]
```

Allowed host paths for pod in Kubernetes cluster

BigAnimal requires the following `HostPath` mounts:

```
{
 "paths":
[

{
     "pathPrefix": "/var/log",
     "readOnly": false

},

{
     "pathPrefix": "/var/lib/docker/containers",
     "readOnly": true

},

{
     "pathPrefix": "/",
     "readOnly": true

},

{
     "pathPrefix": "/sys",
     "readOnly": true

},

{
     "pathPrefix": "/proc",
     "readOnly": true

},

{
     "pathPrefix": "/var/run/docker.sock",
     "readOnly": false

},

{
     "pathPrefix": "/run/containerd/containerd.sock",
     "readOnly": false

},

{
     "pathPrefix": "/dev",
     "readOnly": false

},

{
     "pathPrefix": "/boot",
     "readOnly": true

},

{
     "pathPrefix": "/lib/modules",
     "readOnly": false

},
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
{
     "pathPrefix": "/usr",
     "readOnly": true

},

{
     "pathPrefix": "/etc",
     "readOnly": true

}

]
}
```

Other recommendations from Microsoft Defender for Cloud

Microsoft Defender for Cloud (which includes Azure Secure Center and Azure Defender) analyzes the configurations of your Azure resources to identify potential vulnerabilities.

You might see recommendations from Microsoft Defender for Cloud even after customizing your policies and remediating noncompliant resources. Microsoft offers these recommendations for the reasons that follow.

Restrict unauthorized network access

- Restrict use of host networking and ports.

  BigAnimal runs containers that use the node network namespace to monitor network traffic statistics of Kubernetes cluster worker nodes. To prevent traffic sniffing and configuration changes to the worker node system, BigAnimal has removed all security capabilities for those containers.

- Protect virtual networks with Azure Firewall.

  BigAnimal doesn't enable the Azure Firewall. Instead, BigAnimal uses Azure Network Security Group allowlists to specify allowed inbound and outbound traffic.

  If your organization requires an Azure Firewall for compliance purposes, contact Big Animal support.

Manage access and permissions

- Avoid privileged containers.

  Avoid running containers as root user. However, to achieve some management functionality, like securing and monitoring the application, BigAnimal needs to run some containers in privileged mode.

- Enforce immutable (read-only) root filesystem for containers.

  Avoid running containers with a read-only root filesystem. However, for BigAnimal to achieve some control plane functionality, BigAnimal needs to run some containers with a read-only root filesystem. For example, for BigAnimal to use system calls to secure and monitor the BigAnimal application, it needs to run containers with a read-only root filesystem.

- Avoid running containers as root user.

  BigAnimal must run some containers as the root user to provide some aspects of control plane functionality, such as logging. BigAnimal tightly restricts the use of the root user, and no containers running as root expose network connectivity.

- Avoid containers sharing sensitive host namespaces.

  BigAnimal must run some containers that can share the host process ID namespace to monitor network traffic statistics for cluster worker nodes. To prevent traffic sniffing and configuration changes to the worker node system, BigAnimal has removed all security capabilities for those containers.

- Avoid containers with privilege escalation.

  To enable some monitoring capabilities for Kubernetes, BigAnimal must run some containers that might allow privilege escalation.

Implement security best practices

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

- Disable auto mounting API credentials for Kubernetes clusters.

  Microsoft recommends disabling auto mounting API credentials to prevent a potentially compromised pod from running API commands against a Kubernetes cluster.

  BigAnimal creates service accounts and roles with the least privileges for Kubernetes operators and operands to prevent this scenario.

Enable auditing and logging

Microsoft recommends enabling diagnostic logs in Kubernetes services, Key Vault, and Virtual Machine Scale Sets.

BigAnimal doesn't enable diagnostic logs for Kubernetes services and Key Vault, but it does enable diagnostic logs for Virtual Machine Scale Sets. Resources managed by BigAnimal are logged in Virtual Machine Scale Sets logs. If you must enable other logs for compliance purposes, contact BigAnimal support.

Enable enhanced security features

Microsoft Defender for Cloud includes the capabilities of Microsoft Defender for open-source relational databases.

BigAnimal doesn't enable any of the following capabilities:

- Microsoft Defender for Servers
- Microsoft Defender for Storage
- Microsoft Defender for Key Vault
- Microsoft Defender for Containers
- Microsoft Defender for Kubernetes Service clusters
- Microsoft Defender for Resources Manager
- Microsoft Defender for DNS

If you have questions about enabling any of those capabilities for BigAnimal, contact BigAnimal support.

7.5.3 Customizing Google Cloud compliance policies

Google Cloud uses Assured Workloads and the Organization Policy Service to ensure compliance rules are respected. Assured Workloads monitoring scans your environment in real time and provides alerts whenever organization policy changes violate the defined compliance posture. The monitoring dashboard shows which policy is being violated and provides instructions for resolving the finding.

For more information, see:

- Overview of Assured Workloads
- Introduction to the Organization Policy Service

BigAnimal doesn't customize your Google Cloud compliance rules to prevent conflicts with external workloads.

7.6 Notifications

With BigAnimal, you can opt to get specific types of notifications and receive both in-app and email notifications.

Different types of events are sent as notifications. Users with different roles can configure the preferences to receive these notifications in the in-app inbox, by email, or both. The notifications are categorized into following preference sections:

- Account
- Organizations
- Projects

The notifications under **Account** preference section:

- New project role assigned to you
- Project role unassigned from you
- Personal access key is about to expire

The notifications under **Organizations** preference section:

- Payment method added (specific to organizations that opted credit card payment option)
- Machine user access key is expiring

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

> **Note**
>
> This section is visible only to the organization owner. If the current user is owner of more than one organization, then this section lists the preferences for all the organizations.

The notifications under **Projects** preference section:

- Upcoming cluster maintenance upgrade
- Successful cluster maintenance upgrade
- Failed cluster maintenance upgrade
- User invited to project
- Paused cluster will reactivate
- Paused cluster reactivated
- Cluster paused due to encryption key error
- Encryption key permission required to provision the cluster
- Failed third-party monitoring integration

Configuring notifications

The project owners/editors and organization owners/admins can configure the notifications for the events visible to them. They can choose if they want to receive notifications in the in-app inbox, by email, or both. They can also configure email notifications for their teams in their organization.

Project-level notifications are configured in the project.

Notification settings made by a user apply only to that user. If an email notification is enabled, the email is sent to the email address associated with the user's login.

Viewing notifications

Users in the following roles can view the notifications:

- Organization owners can view the organization notifications.
- Project owners/editors can view the project notifications.
- User can view their own account notifications.

Each notification indicates the level and project it belongs to for the user having multiple roles in BigAnimal.

Select the bell at the top of your BigAnimal portal to view the in-app notifications. By selecting the bell, you can read the notification, mark it as unread, and archive it.

To view the email notifications, check the inbox of your configured email addresses.

Manage notifications

To manage the notifications:

1. Log in to the BigAnimal portal.
2. From the menu under your name in the top-right panel, select **My Account**.
3. Select the **Notifications** tab. Notifications are grouped by account, organizations, and projects available to you.
4. Select account, organization, or project to manage the notifications.
    - Enable/disable the notification for a particular event using the toggle.
    - Select **Email** and **Inbox** next to an event to enable/disable the email and in-app notifications for the event.

8 Pricing and billing

The costs include database pricing for BigAnimal and the associated costs from other providers. You can also view usage and metering information.

BigAnimal calculates the estimated monthly price for your cluster and displays it at the bottom of the page while you're creating a cluster. This estimate includes database costs. If you're using BigAnimal's cloud account, it also includes infrastructure costs.

Database pricing

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Pricing is based on the number of virtual central processing units (vCPUs) provisioned for the database software offering. Consumption of vCPUs is metered hourly.

Single-node and primary/standby high-availability pricing

When primary/standby high-availability configurations are enabled, to calculate the full price for all resources used, multiply the number of vCPUs per instance by the number of replicas configured.

This table shows the cost breakdown.

Database type	Hourly price	Monthly price*
PostgreSQL	$0.0856 / vCPU	$62.49 / vCPU
EDB Postgres Extended Server	$0.1655 / vCPU	$120.82 / vCPU
EDB Postgres Advanced Server	$0.2568 / vCPU	$187.46 / vCPU

* The monthly cost is approximate and assumes 730 hours in a month.

Distributed high-availability pricing

When distributed high-availability configurations are enabled, to calculate the full price for all resources used, multiply the number of vCPUs per instance by the number of data nodes configured. You aren't charged for the database price for witness nodes or groups in distributed high-availability configurations, just the infrastructure resources, such as compute. Distributed high-availability clusters are powered by EDB Postgres Distributed.

This table shows the cost breakdown.

Database type	Hourly price	Monthly price*
EDB Postgres Extended Server	$0.2511 / vCPU	$183.30 / vCPU
EDB Postgres Advanced Server	$0.3424 / vCPU	$249.95 / vCPU

* The monthly cost is approximate and assumes 730 hours in a month.

Cloud infrastructure costs

BigAnimal's cloud account

If you're connecting directly to BigAnimal through BigAnimal's cloud account, EDB bills you for cloud infrastructure. You can view monthly estimates for BigAnimal's cloud infrastructure costs while you're creating a cluster.

A BigAnimal infrastructure unit (BIU) is a unit of measure for cloud infrastructure resources, such as compute, storage, and data transfer. The nominal value of one BIU is $1.00. For example, an AWS compute instance priced at $0.192/instance-hour is multiplied by 1 BIU, so that the equivalent BIU cost is $0.192/BIU-hour.

Using your cloud account

If you're connecting directly to BigAnimal through your Microsoft Azure AWS, or Google Cloud account, EDB doesn't bill you for cloud infrastructure, such as compute, storage, data transfer, monitoring, and logging.

Your cloud provider bills you directly for the cloud infrastructure provisioned according to the terms of your account agreement.

- Microsoft Azure provides invoice and usage information on the Microsoft Azure Portal billing page. Learn more.
- AWS provides invoice information on the AWS Billing and Cost Management console. Learn more.
- Google Cloud provides invoice information on your Cloud Billing account. Learn more.

Management costs

To give you full control over your data, BigAnimal deploys infrastructure in each region to manage the clusters in that region. BigAnimal charges you for this infrastructure only if you're using

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal's cloud. Your cloud provider (Microsoft Azure, AWS, or Google Cloud) bills directly for the infrastructure according to the terms of your account agreement.

If you deploy a large number of clusters in a single region, for example, more than 30, BigAnimal might deploy additional management infrastructure to support your workloads. In addition, BigAnimal provisions a number of other free resources in your account to help with management.

If you want to remove management resources provisioned from your cloud provider, contact Support.

Microsoft Azure management costs

The table shows a breakdown of management costs for Microsoft Azure:

Type	Details
Azure Kubernetes Service (AKS)	BigAnimal uses AKS to orchestrate and manage the database service Managed Kubernetes Virtual Machines provisioned on Windows and Linux.
Azure Monitor	BigAnimal feeds all logs and metrics to Azure Monitor.
Azure blob storage	BigAnimal uses blob storage to store metadata about your account.
Key vault	BigAnimal uses key vault to securely store credentials for managing your infrastructure.

At list price, estimated overall monthly management costs are $400–$700 for a single region. Check with your Microsoft Azure account manager for specifics that apply to your account.

To get a better sense of your Microsoft Azure costs, check out the Microsoft Azure pricing calculator and reach out to BigAnimal Support.

AWS management costs

The table shows a breakdown of management costs for AWS:

Type	Details
Elastic Kubernetes Service (EKS)	BigAnimal uses EKS to orchestrate and manage the database service Managed Kubernetes Virtual Machines provisioned on Windows and Linux.
AWS CloudWatch	BigAnimal feeds all logs and metrics to AWS CloudWatch.
Simple Storage Service (S3)	BigAnimal uses S3 to store metadata about your account.
Key Management Service	BigAnimal uses the key management service to securely store credentials for managing your infrastructure.

At list price, estimated overall monthly management costs are $400–$600 for a single region. Check with your AWS account manager for specifics that apply to your account.

To get a better sense of your AWS costs, check out the AWS pricing calculator and reach out to BigAnimal Support.

Google Cloud management costs

The table shows a breakdown of management costs for Google Cloud:

Type	Details
Google Kubernetes Engine (GKE)	BigAnimal uses GKE to orchestrate and manage the database service Managed Kubernetes Virtual Machines provisioned on Windows and Linux.
Cloud Storage	BigAnimal uses Google Cloud Storage to store metadata about your account.
Cloud Key Management	BigAnimal uses the key management service to securely store credentials for managing your infrastructure.

At list price, estimated overall monthly management costs are $600–$800 for a single region. Check with your Google Cloud account manager for specifics that apply to your account.

Apache Superset costs

Enabling Apache Superset to analyze your data has an added cost. In most cases the costs are approximately $150 per month, based on your cloud provider, instance and storage type selections, and other factors.

PgBouncer costs

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Enabling PgBouncer to pool your connections incurs additional costs that depend on your cloud provider. In addition to the cloud provider costs, PgBouncer connects to your primary server and requires an IP address. BigAnimal provisions up to three instances per PgBouncer-enabled cluster to ensure that performance is unaffected, so each availability zone receives its own instance of PgBouncer. The extra VM costs are the 2vcpu SKU times the number of PgBouncer instances. For AWS, the instance type is c5.large. For Azure, the instance type is F2s_v2.

Payments and billing

Your payment and billing options include:

- Digital self-service using a credit card
- Direct purchase using the Sales Order form
- Azure Marketplace

Digital self-service

BigAnimal charges the credit card for your EDB account month to month and sends invoices to your email. This invoice includes database costs. If you're using BigAnimal's cloud account, it also includes infrastructure costs.

> **Note**
>
> If you want to take advantage of discounts, contactSales.

Direct purchase

If you're using BigAnimal's cloud account, you're invoiced monthly, or you can arrange for a longer term.

If you're using your Microsoft Azure or AWS account, usage details are included in your invoice. Account owners can download a usage report in CSV format from the BigAnimal Usage page. You can set the time frame, database type, and cloud provider prior to downloading the report. The information on the page refreshes hourly.

Azure Marketplace

If you're using Microsoft Azure Marketplace, all billing is handled directly by Microsoft Azure. This includes infrastructure costs as well as database costs. For database costs, Microsoft bills on actual hours in a given month. You can view invoices and usage on the Microsoft Azure Portal billing page. Learn more.

9 Migrating databases to BigAnimal

EDB provides migration tools to bring data from Oracle, PostgresSQL, and EDB Postgres Advanced Server databases into BigAnimal. These tools include Migration Portal and Migration Toolkit for Oracle migrations. More sophisticated migration processes can use tools such as Replication Server for ongoing migrations and LiveCompare for data comparisons.

Migrating from Oracle

For helpful considerations and information when migrating from Oracle, review the EDB Migration Handbook.

EDB also provides a tool, Migration Portal, which provides the details for executing the migration steps:

1. Schema extraction
2. Schema assessment
3. Schema migration
4. Data migration

You can also use the Migration Toolkit for the data migration step. This toolkit is a good option for smaller databases.

Accessing remote Oracle servers from BigAnimal

BigAnimal supports the edb_dblink_oci extension and a set of dblink_ora functions, which you can use to create Oracle syntax-compatible database links and include calls to functions that are only available in Oracle, respectively.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

See the following BigAnimal knowlege base articles for step-by-step instructions for creating links to remote Oracle servers from EDB Postgres Advanced Server clusters:

- Using edb_dblink_oci
- Using dblink_ora_connect()

Migrating from Postgres

Several options are available for migrating EDB Postgres Advanced Server and PostgreSQL databases to BigAnimal. One option is to use the Migration Toolkit. Another simple option for many use cases is to import an existing PostgreSQL or EDB Postgres Advanced Server database to BigAnimal. See Importing an existing Postgres database.

Migrating to distributed high availability clusters

When migrating to a PGD-powered distributed high availability (DHA) cluster, we recommend that you follow the instructions inDHA/PGD bulk migration. This content provides a step-by-step process for migrating your data to a DHA cluster while minimizing the impact of subsequent replication on the process.

9.1 Importing an existing Postgres database

The simplest way to import a database into BigAnimal is using logical backups taken with `pg_dump` and loaded using `pg_restore`. This approach provides a way to export and import a database across different versions of Postgres, including exporting from PostgreSQL and EDB Postgres Advanced Server versions prior to 10.

The high-level steps are:

1. Export existing roles.
2. Import existing roles.
3. For each database, you are migrating:
    1. logical export using `pg_dump`
    2. logical import with `pg_restore`

In case your source PostgreSQL instance hosts multiple databases, you can segment them in multiple BigAnimal clusters for easier management, better performance, increased predictability, and finer control of resources. For example, if your host has 10 databases, you can import one database and related users on a different BigAnimal cluster, one at a time.

Downtime considerations

This approach requires suspending write operations to the database application for the duration of the export/import process. You can then resume the write operations on the new system. This is because `pg_dump` takes an online snapshot of the source database. As a result, the changes after the backup starts aren't included in the output.

The required downtime depends on many factors, including:

- Size of the database
- Speed of the network between the two systems
- Your team's familiarity with the migration procedure

To minimize the downtime, you can test the process as many times as needed before the actual migration. You can perform the export with `pg_dump` online, and the process is repeatable and measurable.

Before you begin

Make sure that you:

- Understand the terminology conventions.
- Have the required Postgres client binaries and libraries.
- Can access the source and target databases.

Terminology conventions

Term	Alias	Description

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Term	Alias	Description
source database	`pg-source`	Postgres instance from which you want to import your data.
target database	`pg-target`	Postgres cluster in BigAnimal where you want to import your data.
migration host	`pg-migration`	Temporary Linux machine in your trusted network from which to execute the export of the database and the subsequent import into BigAnimal. The migration host needs access to both the source and target databases. Or, if your source and target databases are on the same version of Postgres, the source host can serve as your migration host.

Postgres client libraries

The following client binaries must be on the migration host:

- `pg_dumpall`
- `pg_dump`
- `pg_restore`
- `psql`

They must be the same version as the Postgres version of the target database. For example, if you want to import a PostgreSQL 10 database from your private network into a PostgreSQL 14 database in BigAnimal, use the client libraries and binaries from version 14.

Access to the source and target database

Access requirements:

- PostgreSQL superuser access to the source database. This can be the postgres user or another user with superuser privileges.
- Access to the target database in BigAnimal as the edb_admin user.

Verify your access

1. Connect to the source database using `psql`. For example:

   ```
   psql -d "host=<pg-source> user=postgres dbname=postgres"
   ```

   Replace `<pg-source>` with the actual hostname or IP address of the source database and the `user` and `dbname` values as appropriate. If the connection doesn't work, contact your system and database administrators to make sure that you can access the source database. This might require changes to your `pg_hba.conf` and network settings. If `pg_hba.conf` changes, reload the configuration with either `SELECT pg_reload_conf();` using a psql connection or `pg_ctl reload` in a shell connection to the database host.

2. Connect to the target database using the edb_admin user. For example:

   ```
   psql -d "host=<pg-target> user=edb_admin dbname=edb_admin"
   ```

   Replace `<pg-target>` with the actual hostname of your BigAnimal cluster.

Export existing roles

Export the existing roles from your source Postgres instance by running the following command on the migration host:

```
pg_dumpall -r -d "host=pg-source user=postgres dbname=postgres" > roles.sql
```

The generated SQL file looks like this:

```
--
-- PostgreSQL database cluster
dump
--
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
SET default_transaction_read_only =
off;

SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;

--
-- Roles
--

--- … Your roles are here
...

--
-- PostgreSQL database cluster dump
complete
--
```

Import the roles

1. Your BigAnimal cluster already contains the `edb_admin` user, as well as the following-system required roles:

- `postgres` — The superuser, needed by BigAnimal to manage the cluster.
- `streaming_replica` — Required to manage streaming replication.

   As a result, you need to modify the `roles.sql` file to:

1. Remove the lines involving the postgres user. For example, remove lines like these:

   ```
   CREATE ROLE postgres;
   ALTER ROLE postgres WITH SUPERUSER
   ….;
   ```

2. Remove any role with superuser or replication privileges. For example, remove lines like these:

   ```
   CREATE ROLE admin;
   ALTER ROLE admin WITH SUPERUSER
   ….;
   ```

3. For every role that's created, grant the new role to the edb_admin user immediately after creating the user. For example:

   ```
   CREATE ROLE
   my_role;
   GRANT my_role TO edb_admin;
   ```

4. Remove the `NOSUPERUSER`, `NOCREATEROLE`, `NOCREATEDB`, `NOREPLICATION`, `NOBYPASSRLS` permission attributes on the other users.

   The role section in the modified file, then, looks similar to:

   ```
   CREATE ROLE
   my_role;
   GRANT my_role TO edb_admin;
   ALTER ROLE my_role WITH INHERIT LOGIN PASSWORD 'SCRAM-SHA-256$4096:my-Scrambled-Password';
   ```

5. From the migration host, execute:

   ```
   psql -1 -f roles.sql -d "postgres://edb_admin@<pg-target>:5432/edb_admin?sslmode=verify-full"
   ```

   Replace `<pg-target>` with the fully qualified domain name (FQDN) of your BigAnimal cluster.

   This command tries to create the roles in a single transaction. In case of errors, the transaction is rolled back, leaving the database cluster in the same state as before the import attempt. Enforce this behavior using the `-1` option of `psql`.

Export a database

From the migration host, use the `pg_dump` command to export the source database into the target database in BigAnimal. For example:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
pg_dump -Fc -d "host=pg-source user=postgres dbname=app" -f app.dump
```

> **Note**
>
> You can use the `--verbose` option to monitor the progress of the operation.

The command generates a custom `.dump` archive (`app.dump` in this example), which contains the compressed dump of the source database. How long it takes the command to execute varies depending on several variables, including size of the database, network speed, disk speed, and CPU of both the source instance and the migration host. You can inspect the table of contents of the dump with `pg_restore -l <db_name>.dump`.

As with any other custom format dump produced with `pg_dump`, you can take advantage of the features that `pg_restore` provides you with, including:

- Selecting a subset of the import tasks by editing the table of contents and passing it to the `-L` option.
- Running the command in parallel using the `-j` option with the directory format.

For more information, see the `pg_restore` documentation.

Import a database

Use the `pg_restore` command and the `.dump` file you created when exporting the source database to import the database into BigAnimal. For example:

```
pg_restore -C -d  "postgres://edb_admin@pg-target:5432/edb_admin?sslmode=verify-full" app.dump
```

This process might take some time depending on the size of the database and the speed of the network.

In case of error, repeat the restore operation after you delete the database using the following command:

```
psql -d "postgres://edb_admin@pg-target:5432/edb_admin?sslmode=verify-full" \
    -c 'DROP DATABASE app'
```

9.2 Bulk loading data into PGD clusters

Bulk loading data into PGD clusters

This guidance is specifically for environments where there's no direct access to the PGD nodes, only PGD Proxy endpoints, such as BigAnimal's distributed high availability deployments of PGD.

Without using care, bulk loading data into a PGD cluster can cause a lot of replication load on a cluster. With that in mind, this content describes a process to mitigate that replication load.

Provision or prepare a PGD cluster

You must provision a PGD cluster, either manually, using TPA, or on BigAnimal. This will be the target database for the migration. Ensure that you provision it with sufficient storage capacity to hold the migrated data.

We recommend that, when provisioning or, if needed, after provisioning, you set the following Postgres GUC variables.

GUC variable	Setting
maintenance_work_mem	1GB
wal_sender_timeout	60min
wal_receiver_timeout	60min
max_wal_size	Set to either: • A multiple (2 or 3) of your largest table or • More than one third of the capacity of your dedicated WAL disk (if configured)

Make note of the target's proxy hostname (target-proxy) and port (target-port). You also need a user (target-user) and password (target-password) for the target cluster.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

The following instructions give examples for a cluster named `ab-cluster` with an `ab-group` subgroup and three nodes: `ab-node-1`, `ab-node-2`, and `ab-node3`. The cluster is accessed through a host named `ab-proxy` (the target-proxy).

On BigAnimal, a cluster is configured, by default, with an `edb_admin` user (the target-user) that can be used for the bulk upload. The target-password for the target-user will be available from the BigAnimal dashboard for the cluster. A database named `bdrdb` (the target-dbname) will also have been created.

Identify your data source

You need the source hostname (source-host), port (source-port), database name (source-dbname), user , and password for your source database.

Also, you currently need a list of tables in the database that you want to migrate to the target database.

Prepare a bastion server

Create a virtual machine with your preferred operating system in the cloud to orchestrate your bulk loading.

- Use your EDB account.
  - Obtain your EDB repository token from the EDB Repos 2.0 page.
- Set environment variables.
  - Set the `EDB_SUBSCRIPTION_TOKEN` environment variable to the repository token.
- Configure the repositories.
  - Run the automated installer to install the repositories.
- Install the required software.
  - Install and configure:
    - psql
    - PGD CLI
    - Migration Toolkit
    - LiveCompare

Use your EDB account

Go to the EDB Repos 2.0 page and log in with your EDB account. Make a note of the repository token that you will use to configure the repositories on the bastion server.

Set environment variables

Set the `EDB_SUBSCRIPTION_TOKEN` environment variable to the repository token you obtained from the EDB Repos 2.0 page.

```
export EDB_SUBSCRIPTION_TOKEN=your-repository-token
```

Configure repositories

The required software is available from the EDB repositories. You need to install the EDB repositories on your bastion server.

- Red Hat

```
curl -1sLf "https://downloads.enterprisedb.com/$EDB_SUBSCRIPTION_TOKEN/postgres_distributed/setup.rpm.sh" | sudo -E bash
curl -1sLf "https://downloads.enterprisedb.com/$EDB_SUBSCRIPTION_TOKEN/enterprise/setup.rpm.sh" | sudo -E bash
```

- Ubuntu/Debian

```
curl -1sLf "https://downloads.enterprisedb.com/$EDB_SUBSCRIPTION_TOKEN/postgres_distributed/setup.deb.sh" | sudo -E bash
curl -1sLf "https://downloads.enterprisedb.com/$EDB_SUBSCRIPTION_TOKEN/enterprise/setup.deb.sh" | sudo -E bash
```

Install the required software

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Once the repositories are configured, you can install the required software.

Installing `psql` and `pg_dump` / `pg_restore` / `pg_dumpall`

The psql command is the interactive terminal for working with PostgreSQL. It's a client application and can be installed on any operating system. Packaged with psql are pg_dump and pg_restore, command-line utilities for dumping and restoring PostgreSQL databases.

- Ubuntu

```
sudo apt install postgresql-client-16
```

- Red Hat

```
sudo dnf install postgresql-client-16
```

To simplify logging in to the databases, create a.pgpass file for both your source and target servers:

```
source-host:source-port:source-dbname:source-user:source-password
target-proxy:target-port:target-dbname:target-user:target-password
```

Create the file in your home directory and change its permissions to read/write only for the owner. Ensure that your passwords are appropriately escaped in the .pgpass file. If an entry needs to contain : or \, escape this character with \.

```
chmod 0600 $HOME/.pgpass
```

Installing PGD CLI

PGD CLI is a command-line interface for managing and monitoring PGD clusters. It's a Go application and can be installed on any operating system.

- Ubuntu

```
sudo apt-get install edb-pgd5-cli
```

- Red Hat

```
sudo dnf install edb-pgd5-cli
```

Create a configuration file for the PGD CLI:

```
cluster:
  name: target-cluster-name
  endpoints:
    - host=target-cluster-hostname dbname=target-cluster-dbname port=target-cluster-port user=target-cluster-user-
name
```

For the example `ab-cluster`:

```
cluster:
  name: ab-
cluster
  endpoints:
    - host=ab-proxy dbname=bdrbd port=5432
user=edb_admin
```

Save it as `pgd-cli-config.yml`.

See also Installing PGD CLI.

Installing Migration Toolkit

EDB's Migration Toolkit (MTK) is a command-line tool that can be used to migrate data from a source database to a target database. It's a Java application and requires a Java runtime environment to be installed.

- Ubuntu

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
sudo apt-get -y install edb-migrationtoolkit
sudo wget https://jdbc.postgresql.org/download/postgresql-42.7.2.jar -P /usr/edb/migrationtoolkit/lib
```

- Red Hat

```
sudo apt-get -y install edb-migrationtoolkit
sudo wget https://jdbc.postgresql.org/download/postgresql-42.7.2.jar -P /usr/edb/migrationtoolkit/lib
```

See also Installing Migration Toolkit

Installing LiveCompare

EDB LiveCompare is an application that can be used to compare two databases and generate a report of the differences. It will be used later on in this process to verify the data migration.

- Ubuntu

```
sudo apt-get -y install edb-livecompare
```

- Red Hat

```
sudo dnf -y install edb-livecompare
```

See also LiveCompare requirements.

Set up and tune the target cluster

On the target cluster and within the regional group required, select one node to be the destination for the data.

If you have a group `ab-group` with `ab-node-1`, `ab-node-2`, and `ab-node-3`, you can select `ab-node-1` as the destination node.

Set up a fence

Fence off all other nodes except for the destination node.

Connect to any node on the destination group using the psql command. Use `bdr.alter_node_option` and turn the `route_fence` option to `true` for each node in the group apart from the destination node:

```
select bdr.alter_node_option('ab-node-2','route_fence','t');
select bdr.alter_node_option('ab-node-3','route_fence','t');
```

The next time you connect with psql, you're directed to the write leader, which should be the destination node. To ensure that it is, you need to send two more commands.

Make the destination node both write and raft leader

To minimize the possibility of disconnections, move the raft and write leader roles to the destination node.

Make the destination node the raft leader using `bdr.raft_leadership_transfer`. You need to specify the node and the group name that the node is a member of.:

```
bdr.raft_leadership_transfer('ab-node-1',true,'ab-group');
```

Because you fenced off the other nodes in the group, this command triggers a write leader election in the `ab-group` that elects the `ab-node-1` as write leader.

Record then clear default commit scopes

You need to make a record of the default commit scopes in the cluster. The next step overwrites the settings. (At the end of this process, you need to restore them.) Run:

```
select node_group_name,default_commit_scope from bdr.node_group_summary
;
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

This command produces an output similar to::

```
 node_group_name | default_commit_scope
-----------------+----------------------
 world           |
 ab-group        | ba001_ab-group-a
```

Record these values. You can now overwrite the settings:

```
select bdr.alter_node_group_option('ab-group','default_commit_scope', 'local');
```

Prepare to monitor the data migration

Check that the target cluster is healthy.

- To check the overall health of the cluster, run `pgd -f pgd-cli-config.yml check-health`:

```
 Check      Status Message
-----      ------ -------
ClockSkew  Ok     All BDR node pairs have clockskew within permissible limit
Connection Ok     All BDR nodes are accessible
Raft       Ok     Raft Consensus is working correctly
Replslots  Ok     All BDR replication slots are working correctly
Version    Ok     All nodes are running same BDR versions
```

(When the cluster is healthy, all checks pass.)

- To verify the configuration of the cluster, run `pgd -f pgd-cli-config.yml verify-cluster`:

```
 Check                                                                        Status Groups
-----                                                                        ------ ------
There is always at least 1 Global Group and 1 Data Group                      Ok
There are at least 2 data nodes in a Data Group (except for the witness-only group) Ok
There is at most 1 witness node in a Data Group                               Ok
Witness-only group does not have any child groups                             Ok
There is at max 1 witness-only group iff there is even number of local Data Groups  Ok
There are at least 2 proxies configured per Data Group if routing is enabled  Ok
```

(When the cluster is verified, all checks.)

- To check the status of the nodes, run `pgd -f pgd-cli-config.yml show-nodes`:

```
 Node      Node ID    Group    Type Current State Target State Status Seq ID
----      -------    -----    ---- ------------- ------------ ------ ------
ab-node-1 807899305  ab-group data ACTIVE        ACTIVE       Up     1
ab-node-2 2587806295 ab-group data ACTIVE        ACTIVE       Up     2
ab-node-3 199017004  ab-group data ACTIVE        ACTIVE       Up     3
```

- To confirm the raft leader, run `pgd -f pgd-cli-config.yml show-raft`.
- To confirm the replication slots, run `pgd -f pgd-cli-config.yml show-replslots`.
- To confirm the subscriptions, run `pgd -f pgd-cli-config.yml show-subscriptions`.
- To confirm the groups, run `pgd -f pgd-cli-config.yml show-groups`.

These commands provide a snapshot of the state of the cluster before the migration begins.

Migrating the data

Currently, you must migrate the data in four phases:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

1. Transferring the “pre-data” using pg_dump and pg_restore, which exports and imports all the data definitions.
2. Transfer the role definitions using pg_dumpall and psql.
3. Using MTK with the `--dataonly` option to transfer only the data from each table, repeating as necessary for each table.
4. Transferring the “post-data” using pg_dump and pg_restore, which completes the data transfer.

Transferring the pre-data

Use the `pg_dump` utility against the source database to dump the pre-data section in directory format:

```
pg_dump -Fd -f predata --section=pre-data -h <source-hostname> -p <source-port> -U <source-user> <source-dbname> >> predatadump.log
```

Consult `predatadump.log` to ensure that the dump was successful. If it fails, you can repeat the dump after resolving the issue.

Once the pre-data is dumped into the predata directory, you can load it into the target cluster using `pg_restore`:

```
pg_restore -Fd --section=pre-data -d "host=ab-node-1-host dbname=<target-dbname> user=<target-user> options='-cbdr.ddl_locking=off -cbdr.commit_scope=local'"  predata >> predatarestore.log
```

The `options=` section in the connection string to the server is important. The options disable DDL locking and set the commit scope to `local`, overriding any default commit scopes. Using `--section=pre-data` limits the restore to the configuration that precedes the data in the dump.

Consult `predatarestore.log` to ensure that the restore was successful. If it fails, you can repeat the restore after resolving the issue.

Transferring role definitions

Use the `pg_dumpall` utility to dump the role definitions from the source database:

```
pg_dumpall -r -h <source-hostname> -p <source-port> -U <source-user> > roles.sql >> rolesdump.log
```

Consult `rolesdump.log` to ensure that the dump was successful. If it fails, you can repeat the dump after resolving the issue.

Then load the role definitions into the target cluster:

```
psql -h <target-proxy> -p <target-port> -U <target-user> -d bdrdb -f roles.sql >> rolesrestore.log
```

Consult `rolesrestore.log` to ensure that the restore was successful. If it fails, you can repeat the restore after resolving the issue.

Transferring the data

In this step, Migration Toolkit is used to transfer the table data between the source and target.

Edit `/usr/edb/migrationtoolkit/etc/toolkit.properties`. You need to use sudo to raise your privilege to do this, that is, `sudo vi /usr/edb/migrationtoolkit/etc/toolkit.properties`.

```
SRC_DB_URL=jdbc:postgresql://<source-host>:<source-port>/<source-dbname>
SRC_DB_USER=<source-user>
SRC_DB_PASSWORD=<source-password>

TARGET_DB_URL=jdbc:postgresql://<target-host>:<target-port>/<target-dbname>
TARGET_DB_USER=<target-user>
TARGET_DB_PASSWORD=<target-password>
```

Edit the relevant values in the settings.

Ensure that the configuration file is owned by the user you intend to run the data transfer as and read-write only for its owner.

Now, select sets of tables in the source database that must be transferred together, ideally grouping them for redundancy in case of failure:

```
nohup /usr/edb/migrationtoolkit/bin/runMTK.sh -sourcedbtype postgres -targetdbtype postgres  -loaderCount 1  -tableLoaderLimit 1  -fetchSize 4000  -parallelLoadRowLimit 1000  -truncLoad -dataOnly -tables <tablename1>,<tablename2>,... <schemaname> > mtk.log >>mtkerr.log
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

This command uses the `-truncLoad` option and drops indexes and constraints before the data is loaded. It then recreates them after the loading has completed.

You can run multiple instances of this command in parallel. To do so, add an `&` to the end of the command. Ensure that you write the output from each to different files (for example, `mtk_1.log`, `mtk_2.log`).

For example:

```
nohup /usr/edb/migrationtoolkit/bin/runMTK.sh -sourcedbtype postgres -targetdbtype postgres  -loaderCount 1  -tableLoaderLimit 1  -
fetchSize 4000  -parallelLoadRowLimit 1000  -truncLoad -dataOnly -tables warehouse,district,item,new_order,orders,history public
>mtk_1.log >>mtkerr_1.log &

nohup /usr/edb/migrationtoolkit/bin/runMTK.sh -sourcedbtype postgres -targetdbtype postgres -loaderCount 1  -tableLoaderLimit 1  -
fetchSize 4000  -parallelLoadRowLimit 1000  -truncLoad -dataOnly -tables customer public >mtk_2.log >>mtkerr_2.log&

nohup /usr/edb/migrationtoolkit/bin/runMTK.sh -sourcedbtype postgres -targetdbtype postgres  -loaderCount 1  -tableLoaderLimit 1  -
fetchSize 4000  -parallelLoadRowLimit 1000  -truncLoad -dataOnly -tables order_line public >mtk_3.log >>mtkerr_3.log &

nohup /usr/edb/migrationtoolkit/bin/runMTK.sh -sourcedbtype postgres -targetdbtype postgres  -loaderCount 1  -tableLoaderLimit 1  -
fetchSize 4000  -parallelLoadRowLimit 1000  -truncLoad -dataOnly -tables stock public >mtk_4.log >>mtkerr_4.log &
```

This sets up four processes, each transferring a particular table or sets of tables as a background process.

While this is running, monitor the lag. Log into the destination node with psql, and monitor lag with:

```
SELECT NOW(); SELECT pg_size_pretty( pg_database_size('bdrdb') ); SELECT * FROM
bdr.node_replication_rates;
```

Once the lag is consumed, return to the shell. You can now use `tail` to monitor the progress of the data transfer by following the log files of each process:

```
tail -f mtk_1.log mtk_2.log mtk_3.log mtk_4.log
```

You can also consult the error logs (`mtkerr_1.log`, `mtkerr_2.log`, `mtkerr_3.log`, `mtkerr_4.log`) to troubleshoot any issues that arise.

Transferring the post-data

Make sure there's no replication lag across the entire cluster before proceeding with post-data.

Now dump the post-data section of the source database:

```
pg_dump -Fd -f postdata --section=post-data -h <source-hostname> -p <source-port> -U <source-user> <source-dbname>
```

Then load the post-data section into the target database:

```
pg_restore -Fd -d "host=ab-node-1-host dbname=<target-dbname> user=<target-user> options='-cbdr.ddl_locking=off -
cbdr.commit_scope=local'"  --section=post-data postdata
```

If this step fails due to a disconnection, return to monitoring lag (as described previously). Then, when no synchronization lag is present, repeat the restore.

Resume the cluster

Remove the routing fences you set up earlier on the other nodes

Connect directly to the destination node using psql. Use `bdr.alter_node_option` and turn off the `route_fence` option for each node in the group except for the destination node, which is already off:

```
select bdr.alter_node_option('ab-node-2','route_fence','f');
select bdr.alter_node_option('ab-node-3','route_fence','f');
```

Proxies can now route to all the nodes in the group.

Reset commit scopes

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

You can now restore the default commit scopes to the cluster to allow PGD to manage the replication load. Set `default_commit_scope` for the groups to the value for the groups that you recorded in an earlier step.

```
select bdr.alter_node_group_option('ab-group','default_commit_scope', 'ba001_ab-group-a');
```

The cluster is now loaded and ready for production. For more assurance, you can run the `pgd -f pgd-cli-config.yml check-health` command to check the overall health of the cluster and the other PGD commands from when you checked the cluster earlier.

Verify the data migration

Use LiveCompare to compare the source and target databases. Create a configuration file for LiveCompare:

```
[General Settings]
logical_replication_mode = off
difference_tie_breakers = first

[First
Connection]
dsn = host=<source-host> port=<source-port> dbname=<source-dbname> user=<source-username>

[Second
Connection]
dsn = host=<target-proxy> port=<target-port> dbname=<target-dbname> user=<target-username>

[Output
Connection]
dsn = host=<target-proxy> port=<target-port> dbname=<target-dbname> user=<target-username>
```

This configuration file should be saved as `migrationcheck.ini`. The `[First Connection]` and `[Second Connection]` sections should be updated with the appropriate values, with the `[First Connection]` section pointing to the source database and the `[Second Connection]` section pointing to the target database. The `[Output Connection]` section defines a database where a `livecompare` schema will be created to store the comparison results.

Run LiveCompare using the configuration file you created:

```
livecompare migrationcheck.ini --compare
```

LiveCompare will compare the source and target databases and generate a report of the differences. Review the report to ensure that the data migration was successful.

Refer to the LiveCompare documentation for more information on using LiveCompare.

10 Using the API and CLI

System administrators and developers can use the BigAnimal API and BigAnimal command line interface (CLI) to integrate directly with BigAnimal for management activities. Management activities include, for example, cluster provisioning, deprovisioning, and scaling as well as automating administrative operations.

10.1 Access key

An access key provides an authentication process for BigAnimal users to access the BigAnimal API directly, without the `OAuth2` authorization flow. The access key links to only one user. Each access key link is immutable since its creation. Each access key has an expiration time you specify, ranging from 1 to 365 days.

An access key belongs to only one organization. An access key can be created for a machine user or for a normal user. A normal user can't use their access key across the organizations. The key is managed by an organization owner for the machine user, whereas the normal user manages their own access key. Once the access key expires, you must create a new one. Also, if you lose the access key, you have to delete it and create a new one.

User type	Quota - Maximum keys	Keys created and managed
Machine user	2	Two access keys, created while the organization owner is creating a machine user. Optionally, can be individually added from **Access keys** tab on machine user's details page using **Create New Key** option.
Normal user	1	One access key, created and managed by user from their home page.

An access key can be revoked from a user. Revoking the key doesn't affect the role or any other authentication process of the linked user.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Create your personal access key

To create an access key:

1. Log in to the BigAnimal portal.
2. From the menu next to your organization name in the top right of the portal, select **My Account**.
3. From the My Account page, select the **Access Keys** tab.
4. From the **Access Keys** tab, select **Create New Key**.
5. Provide the **Access Key Name**.
6. Provide the **Expiry days** in the range of 1 to 365.

   The generated access key is provided for the user.

Copy this access key and save it in a secure location. The access key is available only when you create it. If you lose your access key, you must delete it and create a new one.

You can enable in-app inbox or email notifications to get alerted when your personal key is about to expire. For more information, see manage notifications.

Manage access key

You can manage the access keys for a machine user from the **Users** home page.

To view details about a user's access keys:

1. From the menu next to your organization name in the top right of the portal, select **Users Management** .
2. Select the **Username** from the list.
3. View the **Access Keys** tab on the user's home page. Each row provides the access key details:
   - **Name** — Access key name.
   - **Access Key** — Access key in half encrypted format.
   - **Expiry at** — The expiry date for the access key.
   - **Created at** — The creation date for the access key.

To delete a specific key, select **Delete** next to the key. You're prompted to confirm the deletion.

To create a key, select **Create New Key** at the top-right corner of the **Access Keys** tab.

Using the access key

The BigAnimal API detects if the caller is using an access key in the request. If the caller uses the access key, the API checks:

- The validity of the key
- The organization of the user requesting access
- The user permissions

It then authorizes the access. For more information, see Using the BigAnimal API.

10.2 Using the BigAnimal API

> **Deprecation notice**
>
> EDB released v3 of BigAnimal API in January 2023. v2 of BigAnimal API goes out of support on June 30, 2023. To see the changes in v3, see the Change log. Update your applications accordingly.

Use the BigAnimal API to integrate directly with BigAnimal for management activities such as cluster provisioning, deprovisioning, and scaling.

The API reference documentation is available from the BigAnimal portal. To access the documentation directly, go to API docs.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Prerequisite

Before calling API, create an access key. For more information, see Access key.

Call the API

To call the BigAnimal API, your application must pass the retrieved access key as a bearer token in the `x-access-key` header of your HTTP request. For example:

```
curl --request GET \
  --url "https://portal.biganimal.com/api/v3/projects" \
  --header "x-access-key: $ACCESS_KEY"
```

Where the `ACCESS_KEY` variable is the access key's text copied from the BigAnimal UI.

Example response:

```
{
    "projectId": "prj_abcd1234",
    "projectName": "Test
project",
    "clusterCount": 4,
    "userCount": 10,
    "cloudProviders": [
    {
       "cloudProviderId": "aws",
       "cloudProviderName": "AWS"
    },
    {
       "cloudProviderId": "azure",
       "cloudProviderName": "Azure"
    }
  ]
}
```

10.3 Using the BigAnimal CLI

Use the command line interface (CLI) for BigAnimal management activities, such as cluster provisioning and getting cluster status from your terminal. The CLI is an efficient way to integrate with BigAnimal and enables system administrators and developers to script and automate the BigAnimal administrative operations.

Installing the CLI

The CLI is available for Linux, MacOS, and Windows operating systems.

Download the binary executable

- For Linux operating systems, use the following command to get the latest version of the binary executable:

  ```
  curl -LO "https://cli.biganimal.com/download/$(uname -s)/$(uname -m)/latest/biganimal"
  ```

- For all other operating systems, download the executable binary here. After downloading, move the binary executable under a directory on your executable search path.

(Optional) Validate the download

- For Linux users:

  1. Copy the SHA256 checksum code for Linux distribution from the BigAnimal CLI page and store it as a local file, such as `biganimal_linux_amd64.sha256`. Alternatively, click the SHA256 code to download it as a file directly and verify the content of the downloaded file is identical to the checksum code showed on the page.
  2. From your local shell, validate the binary executable file against the checksum file:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
echo "$(<biganimal_linux_amd64.sha256) biganimal" | sha256sum --check
```

- For Windows users:

  1. Download the SHA256 checksum code for Windows distribution from the BigAnimal CLI page and store it as a local file, such as `biganimal_windows_amd64.sha256`. Alternatively, click the SHA256 code to download it as a file directly and verify the content of the downloaded file is identical to the checksum code showed on the page.
  2. Validate the binary executable file against the checksum file using `CertUtil`:

     ```
     CertUtil -hashfile biganimal.exe SHA256 type biganiml_windows_amd64.sha256
     ```

- For MacOS users:

  1. Download the SHA256 checksum code for MacOS distribution from the BigAnimal CLI page and store it as a local file, such as `biganimal_darwin_amd64.sha256`. Alternatively, click the SHA256 code to download it as a file directly and verify the content of the downloaded file is identical to the checksum code showed on the page.
  2. From MacOS terminal, validate the binary executable file against the checksum file:

     ```
     echo "$(<biganimal_darwin_amd64.sha256)  biganimal" | shasum -a256 -c
     ```

Make the CLI command executable within Cloud Shell

Change the permissions of the CLI to make it executable in Cloud Shell:

```
chmod +x biganimal
```

Authenticating user using access key

You can create and manage login credentials using an access key that allows users to access BigAnimal resources. We recommend that you use the access key authentication method.

BigAnimal CLI provides the following access key operations.

Importing access key

This command supports two modes: flag mode and interactive mode.

This example shows how to import an access key in flag mode:

```
biganimal credential import-access-key --name ba-user-1 --access-key <your-key>
```

```
Access key "ba-user-1" is imported, operation succeeded
Switched the context credential to "ba-user-1".
```

> **Note**
>
> Avoid adding an access key by way of the `import-access-key` command as plain text in flag mode. Instead you can use:
>
> ```
> biganimal credential import-access-key --name <name> --access-key $(< file_contains_the_key.txt)
> ```
>
> Or, to avoid leaking the key into your shell command history, you can use interactive mode.

This example shows how to import an access key in interactive mode:

```
biganimal credential import-access-key
? Credential Name: ba-user-2
? Access Key: ****************************************************
```

```
Access key "ba-user-2" is imported, operation succeeded
Switched the context credential to "ba-user-2".
```

Setting access key environment variable

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

You can configure an environment variable `BA_ACCESS_KEY`. This setting overrides the default context credential. If you want to use the default context credential, then remove this environment variable, or set it to `""`.

Switching access keys

You can import multiple access keys and then set one key as the default context credential for all commands. Use `credential show` to list all available credentials, and use `config set context_credential` to set a default credential.

Authenticating as a valid user

Before using the CLI to manage BigAnimal, you need to authenticate as a valid BigAnimal user. Use the `credential create` command to authenticate through the BigAnimal website and assign a refresh token and an access token to a local credential. For example:

```
biganimal credential create\
  --name "ba-user1"
```

```
Querying Authentication Endpoint for 'portal.biganimal.com'
First, copy your one-time code:
        CWWG-SMXC
Then visit: https://auth.biganimal.com/activate
press [Enter] to continue in the web browser...

Credential "ba-user1" created!
```

Refresh tokens expire after 30 days. To continue using the credential to access the CLI, use the `credential reset` command to authenticate through the BigAnimal website and receive a new refresh token:

```
biganimal credential reset ba-user1
```

```
Visit this URL to login: https://auth.biganimal.com/activate?user_code=****-****
or press [Enter] to continue in the web browser

Credential "ba-user1" reset operation succeeded
```

You can create multiple credentials for different BigAnimal accounts and then set one as context of your current management session. Use `credential show` to list all available credentials, and use `config set context_credential` to set a default credential for the current context. For example:

```
biganimal credential show
```

```
┌──────────────────────────────────────────────────────────────────────────────────────────┐
│ Credentials                                                                              │
├──────────────┬──────────────┬──────────────┬──────────────────┬──────────────────┬──────┬──────────┤
│ name         │ type         │ key          │ organization     │ address          │ port │ context  │
├──────────────┼──────────────┼──────────────┼──────────────────┼──────────────────┼──────┼──────────┤
│ ba-user1     │ AccessKey    │ baak_1e2vAJO*** │                  │ portal.biganimal.com │ 443  │ x        │
│ ba-user2     │ ClassicToken │              │ My_organization  │ portal.biganimal.com │ 443  │          │
└──────────────┴──────────────┴──────────────┴──────────────────┴──────────────────┴──────┴──────────┘
```

```
biganimal config set context_credential ba-user1
```

Creating credentials for an alternative organization

The BigAnimal CLI supports the capability to create credentials for an alternative organization to which you have been invited or are already a member. You can invite users that have an EDB account to join an organization by selecting the **Invite New User** option on the Users page in the BigAnimal portal.

To create credentials for an alternative organization, you can use either interactive mode or specify the settings with flags on the command line:

- Select the organization from the list in interactive mode:

  ```
  biganimal credential create
  ```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
? Credential Name: ba-user2
? Organization:  [Use arrows to move, type to filter]
  My_Organization
> My_Organization_2

? Organization: My_Organization_2
Credential "ba-user2" is created operation succeeded
Switched the context credential to "ba-user2".
```

- Specify the organization name in the `credential create` command:

```
biganimal credential create -name ba-user2 --organization "My_Organization_2"
```

You can verify the operation using `credential show`:

```
biganimal credential show
```

```
┌──────────────────────────────────────────────────────────────────────────────────────────────────┐
│ Credentials                                                                                      │
├──────────────┬──────────────┬──────────────┬──────────────────┬──────────────────┬──────┬─────────┤
│ name         │ type         │ key          │ organization     │ address          │ port │ context │
├──────────────┼──────────────┼──────────────┼──────────────────┼──────────────────┼──────┼─────────┤
│ ba-user2     │ ClassicToken │              │ org_aNR9SNuNcUN6vGSn │ portal.biganimal.com │ 443  │ x       │
└──────────────┴──────────────┴──────────────┴──────────────────┴──────────────────┴──────┴─────────┘
```

Configuring

The initial running of the CLI creates a hidden configuration folder in your user root directory. For example, for Linux it's `${HOME}/.edb-cli`. The CLI persists the configuration file in this directory as well as the credentials.

Don't edit files in this directory directly. Instead, use the `config` subcommand to list and update the configuration settings of the CLI. Use the following command to get detailed usage and available configurations information:

```
biganimal config
```

Available configuration settings

Setting	Description
`context_credential`	The default credential used in the following commands.
`context_project`	The default project used to run the following commands.
`output_mode`	The command line output format: table, json, xml, or yaml.
`confirm_mode`	If enabled, create/update/delete commands require user confirmation.
`interactive_mode`	If enabled, CLI prompts for missing flags and available options. See Interactive mode for more information.
`check_update_mode`	If enabled, CLI detects new updates and prompts for download.
`warning_mode`	If enabled, CLI displays warning messages.

Usability features

Online command reference and help

Use the `-h` or `--help` flags for more information on the CLI commands. You can use these flags on the `biganimal` command to get a listing of all the available subcommands (`biganimal -h`) or on a subcommand to get information on that particular command (for example, `biganimal create cluster -h`).

Interactive mode

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

In interactive mode, the CLI prompts you for any missing mandatory flags and lists any available options for your current context. To enable interactive mode:

```
biganimal config set interactive_mode on
```

Sample use cases

See:

- Managing clusters with the CLI
- Accessing BigAnimal features with the CLI

10.3.1 Managing clusters using the CLI

---Add pause/resume cluster and PGD

These examples show Azure as the cloud provider unless indicated otherwise.

Although the functionality is the same when using AWS or Google Cloud, there may be additional input flags based on the cloud provider type. Use the `-h` or `--help` flags for more information on the CLI commands.

Managing single-node and primary/standby high-availability clusters

Use the `cluster` commands to create, retrieve information on, and manage single-node and primary/standby high-availability clusters.

Create a cluster in interactive mode

The default mode for the `cluster create` and `pgd create` commands is an interactive mode that guides you through the required cluster configuration by providing you with the valid values.

> **Tip**
>
> You can turn off prompting using the `biganimal config set interactive_mode off` command. With prompting disabled, if any required flags are missing, the CLI exits with an error.

For example, to create a primary/standby high-availability cluster:

```
biganimal cluster create
```

```
Cluster architecture: Primary/Standby High Availability
Number of standby replicas: 2 Replicas
Enable read-only workloads: No
Provider: Azure
Cloud Provider Subscription ID: "111,222"
Service Account IDs, (leave empty to stop adding): "id1@iam.gcp"
Cluster Name: my-biganimal-cluster
Password: ****************
PostgreSQL type: EDB Postgres Advanced Server
PostgreSQL version: 14
Region: East US
Instance type: E2s v3(2vCPU, 16GB RAM)
Volume type: Azure Premium Storage
Volume properties: P1 (4 Gi, 120 Provisioned IOPS, 25 Provisioned MB/s)
Networking: Public
By default your cluster allows all inbound communications, add IP allowed list to restrict the access: Yes
Add CIDR blocks "192.168.1.1/16=Sample Description" leave empty to stop adding:
Add database config in the format "application_name=sample_app&array_nulls=true", Leave empty for default configuration:
Backup Retention Period, note backups will incur storage charges from the cloud provider directly. e.g. "7d", "2w" or "3m": 30d
```

You're prompted to confirm that you want to create the cluster. After the cluster creation process is complete, it generates a cluster ID.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
biganimal cluster create
```

```
........
Are you sure you want to Create Cluster ? [y|N]: y
Create Cluster operation is started
Cluster ID is "p-gxhkfww1fe"
To check current state, run: biganimal cluster show --id p-gxhkfww1fe
```

Check your cluster was created successfully using the `cluster show` command shown in the return message:

```
biganimal cluster show --id p-gxhkfww1fe
```

```
┌──────────────
```

```
volumeType: azurepremiumstorage                                    # <string:  volume type
id>
volumeProperties: P1                                               # <string:  Applicable to Azure Premium Storage only, volume
properties id>
volumePropertySize: 4Gi                                            # <string:  Not Applicable to Azure Premium Storage, volume size
in Gibibytes or Tebibytes, you may append unit suffix Ti,Gi(the default unit).>
volumePropertyIOPS: 1000                                           # <number>: Not Applicable to Azure Premium Storage and GCP:[pd-
ssd], volume Input/Output Operations Per Second>
networking: public                                                 # <string:  input "private" or "public"
network>
allowIpRangeMap:                                                   # <list: IP Range to allow network traffic to your cluster from
the public Internet>
  - cidr: 9.9.9.9/28                                               # <string: CIDR of allowed source IP
range>
    description: Allow traffic from App A                          # <string: The description of this allowed ip
range>
  - cidr: 10.10.10.10/27                                           # <string: CIDR of allowed source IP
range>
    description: Allow traffic from App B                          # <string: The description of this allowed ip
range>
readOnlyWorkloads: true                                            # <bool: Set True to enable read-only connection and route all
read-only queries to standby replicas and reduce the workload on primary>
pgConfigMap:                                                       # <Object: Postgres
configuration>
  application_name: test_app                                       # <string: set the database "application_name" property to
"test_app">
  array_nulls: true                                                # <bool: set the database "array_nulls" property to
True>
backupRetentionPeriod: 30d                                         # <string: Retention period must be between 1-180 days or 1-25
weeks or 1-6 months. Using strings like "7d" or "2w" or "3m" to specify days, weeks and months respectively.>
superuserAccess: true                                              # <bool:    Superuser Access, Grants superuser access to the
edb_admin role, this will bypass all the permission checks and should be used very carefully.>
pgvector: true                                                    # <bool: Set True to enable pgvector extension. Adds support for
vector storage and vector similarity search to Postgres.>
postgis: true                                                     # <bool: Set True to enable postgis extension. PostGIS extends the
capabilities of the PostgreSQL relational database by adding support storing, indexing and querying geographic data.>
```

> **Note**
>
> For backward compatibility, `allowIpRangeMap` and `pgConfigMap` properties also support embedded JSON format.
>
> ```
> allowIpRangeMap: [["9.9.9.9/28", "Allow traffic from App A"],["10.10.10.10/27", "Allow traffic from App
> B"]]
> pgConfigMap: [["application_name","test_app"],["array_nulls","true"]]
> ```

To create the cluster using the sample configuration file `config_file.yaml`:

```
biganimal cluster create --config-file "./config_file.yaml"
```

To enable you to view valid values to use in the configuration file for BigAnimal and cloud service provider-related properties, the CLI provides a series of cluster subcommands. For example, you can use `cluster show-architectures` to list all BigAnimal database architectures available in your cloud service provider account:

```
biganimal cluster show-architectures
```

```
┌──────────────────────────────────────────────────────────┐
│ Architecture                                             │
├─────────┬──────────────────────────────────────┬─────────┤
│ ID      │ Name                                 │ Status  │
├─────────┼──────────────────────────────────────┼─────────┤
│ ha      │ Primary/Standby High Availability    │ enabled │
│ pgd     │ Extreme High Availability            │ disabled│
│ single  │ Single Node                          │ enabled │
└─────────┴──────────────────────────────────────┴─────────┘
```

> **Tip**
>
> You can turn off the confirmation step with the `biganimal disable-confirm` command.

Get cluster connection information

To use your BigAnimal cluster, you first need to get your cluster's connection information. To get your cluster's connection information, use the `cluster show-connection` command:

```
biganimal cluster show-connection \
  --name "my-biganimal-cluster" \
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
  --provider "azure" \
  --region "eastus"
```

```
┌──────────────────────────┬──────────────────────────────────────────────────────────────────────────────────────────┐
| Name                     | Details                                                                                  |
├──────────────────────────┼──────────────────────────────────────────────────────────────────────────────────────────┤
| read-write-connection    | postgresql://edb_admin@p-gxhkfww1fe.30glixgayvwhtmn3.enterprisedb.network:5432/edb_admin |
| rw-service-name          | postgresql://edb_admin@p-gxhkfww1fe.30glixgayvwhtmn3.enterprisedb.network:5432/edb_admin |
| read-only-connection     | Disabled                                                                                 |
| ro-service-name          | Disabled                                                                                 |
└──────────────────────────┴──────────────────────────────────────────────────────────────────────────────────────────┘
```

> **Tip**
>
> You can query the complete connection information with other output formats, like JSON or YAML. For example:
>
> ```
> biganimal cluster show-connection \
>   --name "my-biganimal-cluster" \
>   --provider "azure" \
>   --region "eastus" \
>   --output "json"
> ```

Update cluster

After the cluster is created, you can update attributes of the cluster, including both the cluster's profile and its deployment architecture. You can update the following attributes:

- Cluster name
- Password of administrator account
- Cluster architecture
- Number of standby replicas
- Instance type of cluster
- Instance volume properties
- Networking
- Allowed IP list
- Postgres database configuration
- Volume properties, size, IOPS
- Retention period
- Read-only workloads
- IAM authentication
- Cloud service provider subscription IDs
- Service account IDs

For example, to set the public allowed IP range list, use the `--cidr-blocks` flag:

```
./biganimal cluster update --name "my-biganimal-cluster" --provider "azure" \
  --region "eastus" \
  --cidr-blocks "9.9.9.9/28=Traffic from App A"
```

To check whether the setting took effect, use the `cluster show` command, and view the detailed cluster information output in JSON format. For example:

```
biganimal cluster show --name "my-biganimal-cluster" --provider "azure" \
  --region "eastus" \
  --output "json" \
| jq '.[0].allowIpRangeMap'
```

```
[
  [
    "9.9.9.9/28",
    "Traffic from App A"
  ]
]
```

Update the Postgres configuration of a cluster

To update the Postgres configuration of a BigAnimal cluster directly from the CLI:

```
biganimal cluster update --id "p-gxhkfww1fe" \
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
  --pg-config "application_name=ba_test_app,array_nulls=false"
```

```
Update Cluster operation is started
Cluster ID is "p-gxhkfww1fe"
```

To specify multiple configurations, you can use multiple `--pg-config` flags or include multiple configuration settings as a key-value array string separated by commas in one `--pg-config` flag. If a Postgres setting contains a comma, you need to specify it with a separate `--pg-config` flag.

> **Note**
>
> You can update the cluster architecture with the `--cluster-architecture` flag. The only supported scenario is to update a single-node cluster to a primary/standby high-availability cluster.

Delete a cluster

To delete a cluster you no longer need, use the `cluster delete` command. For example:

```
biganimal cluster delete \
  --name "my-biganimal-cluster" \
  --provider "azure" \
  --region "eastus"
```

You can list all deleted clusters using the `show-deleted-clusters` command and restore them from their history backups as needed.

Restore a cluster

BigAnimal continuously backs up your PostgreSQL clusters. Using the CLI, you can restore a cluster from its backup to any point in time as long as the backups are retained in the backup storage. The restored cluster can be in another region and have different configurations. You can specify new configurations in the `cluster restore` command. For example:

```
biganimal cluster restore\
  --name "my-biganimal-cluster" \
  --provider "azure" \
  --region "eastus" \
  --password "mypassword@123" \
  --new-name "my-biganimal-cluster-restored" \
  --new-region="eastus2" \
  --cluster-architecture "single" \
  --instance-type "azure:Standard_E2s_v3" \
  --volume-type "azurepremiumstorage" \
  --volume-property "P1" \
  --networking "public" \
  --cidr-blocks="10.10.10.10/27=Traffic from App B" \
  --restore-point "2022-01-26T15:04:05+0800" \
  --backup-retention-period "2w" \
  --postgis=true
  --pgvector=true
  --read-only-workloads: "true"
  --csp-subscription-ids "123123123,456456456"
  --service-account-ids "service-account-1234567b@development-data-123456.iam.gserviceaccount.com,
service-account-1234567b@development-data-123456.iam.gserviceaccount.com"
  --credential "my_credential"
```

The password for the restored cluster is mandatory. The other parameters, if not specified, inherit the source database's settings.

To restore a deleted cluster, use the `--from-deleted` flag in the command.

> **Note**
>
> You can restore a cluster in a single cluster to a primary/standby high-availability cluster and vice versa. You can restore a distributed high-availability cluster only to a cluster using the same architecture.

Pausing a cluster

To pause a cluster, use the `cluster pause` command. The `cluster pause` command supports `flag` or `interactive` mode. The syntax for the command is:

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

```
biganimal cluster pause {--id | --provider --region --name}
```

Where `id` is a valid cluster ID. The `id` is mandatory. `provider` is a cloud provider of the cluster. `region` is the region of the cluster. `name` is the name of the cluster.

If `id` of the cluster isn't known then use `--provider --region --name` to identify the cluster.

Examples:

Pausing a cluster using ID:

```
biganimal cluster pause --id p-c5fh47nf
```

Pausing a cluster using name, provider, and region:

```
biganimal cluster pause
--name my-cluster
--provider azure
--region eastus2
```

Pausing a cluster in interactive mode:

```
./biganimal cluster pause
? Provider: AWS (Your Cloud Account)
? Region: AP South 1
? Cluster Name: user1-test-tags-IA
```

```
Pause Cluster operation succeeded, "p-94pjd2w0ty"
```

Resuming a cluster

To resume a cluster, use the `cluster resume` command. The `cluster resume` command supports `flag` and `interactive` mode. The syntax for the command is:

```
biganimal cluster resume {--id | --provider --region --name}
```

Where `id` is a valid cluster ID. The `id` is mandatory. `provider` is a cloud provider of the cluster. `region` is the region of the cluster. `name` is the name of the cluster.

If `id` of the cluster isn't known then use `--provider --region --name` to identify the cluster.

Examples:

Resuming a cluster using ID:

```
biganimal cluster resume --id p-c5fh47nf
```

Resuming a cluster using name, provider, and region:

```
biganimal cluster resume
--name my-cluster
--provider azure
--region eastus2
```

Resuming a cluster using interactive mode:

```
./biganimal cluster resume
? Provider: AWS (Your Cloud Account)
? Region: AP South 1
? Cluster Name: user1-test-tags-IA
```

```
Resume Cluster operation succeeded, "p-94pjd2w0ty"
```

Managing distributed high-availability clusters

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Use the BigAnimal `pgd` commands to create, retrieve information on, and manage distributed high-availability clusters.

> **Note**
>
> In addition to the BigAnimal `pgd` commands, you can switch over and use commands available in the EDB Postgres Distributed CLI to perform PGD-specific operations. The only EDB Postgres Distributed CLI commands that don't apply to BigAnimal are `create-proxy` and `delete-proxy`.

Create a distributed high-availability cluster

Create a distributed high-availability cluster using a YAML configuration file.

The syntax of the command is:

```
biganimal pgd create --config-file <config_file>
```

Where `<config_file>` is a valid path to a YAML configuration file. For example:

```
clusterName: pgd-cli-name
password: Meredith Palmer Memorial
postgresType: epas
postgresVersion: "14"
provider: azure
dataNodes: 2
dataGroups:
  - iamAuthentication: false
    region: westus2
    instanceType: azure:Standard_E2s_v3
    volumeType: azurepremiumstorage
    volumeProperties: P2
    customMaintenanceWindow:
      maintenanceStartTime: 18:00
      maintenanceStartDay: wednesday
    networking: public
    allowIpRangeMap:
      - cidr: 9.9.9.9/28
        description: Allow traffic from App A
      - cidr: 10.10.10.10/27
        description: Allow traffic from App B
    pgConfigMap:
      application_name: test
      array_nulls: true
    backupRetentionPeriod: 30d
  - iamAuthentication: false
    region: canadacentral
    instanceType: azure:Standard_E2s_v3
    volumeType: azurepremiumstorage
    volumeProperties: P2
    customMaintenanceWindow:
      maintenanceStartTime: 18:00
      maintenanceStartDay: tuesday
    networking: public
    allowIpRangeMap:
      - cidr: 9.9.9.9/28
        description: Allow traffic from App A
      - cidr: 10.10.10.10/27
        description: Allow traffic from App B
    pgConfigMap:
      application_name: test1
      array_nulls: true
    backupRetentionPeriod: 30d
witnessGroups:
  - provider: azure
    region: uksouth
    customMaintenanceWindow:
      maintenanceStartTime: 18:00
      maintenanceStartDay: monday
```

Add a data group

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Add a data group using a YAML configuration file.

The syntax of the command is:

```
biganimal pgd add-group --config-file <config_file>
```

Where `<config_file>` is a valid path to a YAML configuration file. For example:

```
clusterId: clusterID
password: Meredith Palmer Memorial
dataGroups:
  - iamAuthentication: false
    region: centralus
    instanceType: azure:Standard_E2s_v3
    volumeType: azurepremiumstorage
    volumeProperties: P2
    customMaintenanceWindow:
      maintenanceStartTime: 18:00
      maintenanceStartDay: monday
    networking: public
    allowIpRangeMap:
      - cidr: 9.9.9.9/28
        description: Allow traffic from App A
      - cidr: 10.10.10.10/27
        description: Allow traffic from App B
    pgConfigMap:
      application_name: test1
      array_nulls: true
    backupRetentionPeriod: 30d
witnessGroups:
  - provider: aws
    region: ap-south-1
    customMaintenanceWindow:
      maintenanceStartTime: 15:00
      maintenanceStartDay: wednesday
```

Update a distributed high-availability cluster

Update a distributed high-availability cluster and its data groups using a YAML configuration file.

The syntax of the command is:

```
pgd update [--config-file]
```

Where `<config_file>` is a valid path to a YAML configuration file with the same format as a configuration file for creating a distributed high-availability cluster. See Create a distributed high-availability cluster.

Show distributed high-availability clusters

Show all active clusters or a specific cluster. You can also optionally show deleted clusters.

The syntax of the command is:

```
biganimal pgd show [--id] [--deleted]
```

Restore a distributed high-availability cluster

Restore a distributed high-availability cluster or a deleted distributed high-availability cluster to a new cluster on the same cloud provider. You can restore an active cluster or a deleted cluster within its retention period. You can restore only one data group. By default, the new cluster inherits all settings of the source cluster. You can change the cluster setting and database configurations by specifying new values in the configuration file.

The syntax of the command is:

```
pgd restore [--config-file]
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Where `<config_file>` is a valid path to a YAML configuration file. For example:

```
clusterName: pgd-restore-name
password: Meredith Palmer Memorial
dataNodes: 2
clusterId: p-9fdkl5ju29
dataGroups:
  - iamAuthentication: false
    region: uksouth
    instanceType: azure:Standard_E2s_v3
    volumeType: azurepremiumstorage
    volumeProperties: P2
    allowIpRangeMap:
      - cidr: 9.9.9.9/28
        description: Allow traffic from App A
      - cidr: 10.10.10.10/27
        description: Allow traffic from App B
    pgConfigMap:
      application_name: test
      array_nulls: true
    backupRetentionPeriod: 30d
    sourceGroupId: p-9fdkl5ju29-a
```

Get distributed high-availability cluster connection information

To connect to and use your BigAnimal distributed high-availability cluster, you first need to get your cluster group's connection information.

The syntax of the command is:

```
biganimal pgd show-group-connection {--id --group-id} [--read-only] \
                                    [--read-write]
```

Delete a distributed high-availability cluster

Delete a specific BigAnimal distributed high-availability cluster.

The syntax of the command is:

```
biganimal pgd delete-group {--id --group-id}
```

The `--id` and `--group-id` flags are mandatory. For example:

```
biganimal pgd delete-group --id clusterID --group-id clusterDataGroupID
```

Pausing a distributed high-availability cluster

To pause a distributed high-availability cluster, use `pgd pause` command. The `pgd pause` command supports the `flag` mode only. The syntax for the command is:

```
biganimal pgd pause {--id}
```

Where `id` is a valid cluster ID. The `id` flag is mandatory.

Example:

```
biganimal pgd pause --id p-c5fh47nf
```

Resume a distributed high-availability cluster

To resume a distributed high-availability cluster, use the `pgd resume` command. The `pgd resume` command supports `flag` mode only. The syntax for the command is:

```
biganimal pgd resume {--id}
```

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Where, `id` is a valid cluster ID. The `id` flag is mandatory.

Example:

```
biganimal pgd resume --id p-c5fh47nf
```

10.3.2 Using BigAnimal features with the CLI

Faraway replicas CLI commands

You can use the faraway-replica-specific CLI commands to create, promote, and get information on faraway replicas.

To update, delete, and restore faraway replicas, use the `faraway-replica update`, `faraway-replica delete`, and `faraway-replica restore` commands. See Managing clusters with the CLI for more information.

Create a faraway replica

You use the `faraway-replica create` command to create a replica. You can use either interactive mode or a config file. This example shows interactive mode:

```
biganimal faraway-replica create
```

```
? Source Cluster Provider ID: Azure
? Source Cluster Region ID: Canada Central
? Source Cluster Name: abcd
? Faraway Replica Name: abcd-replica-1
? Faraway Replica Region: France Central
? Instance type: D2s v4(2vCPU, 8GB RAM)
? Volume type: Azure Premium Storage
? Volume properties: P1 (4 Gi, 120 Provisioned IOPS, 25 Provisioned MB/s)
? Networking: Public
? Cloud Provider Subscription ID: "111,222"
? Service Account IDs, (leave empty to stop adding): "id1@iam.gcp"
? By default your cluster allows all inbound communications, add IP allowed list to restrict the access: No
? Add database config in the format "application_name=sample_app&array_nulls=true", Leave empty for default configuration:
? Backup Retention Period, note backups will incur storage charges from the cloud provider directly. e.g. "7d", "2w" or "3m": 3m
```

You're prompted to confirm that you want to create the faraway replica. After the faraway replica creation process is complete, it generates a replica cluster ID.

Get information on faraway replicas

You use the `faraway-replica show-connected` command to get information on faraway clusters for a specified source cluster. You can use either interactive mode or specify the settings with flags on the command line. This example shows interactive mode:

```
biganimal faraway-replica show-connected
```

```
? Provider: Azure
? Region: Norway East
? Cluster Name: abcd
┌──────────────
```

```
└──────────────┴──────────────────┴──────────────────────────────────────┴──────────┴──────────┴──────────────┴──────────────────────────────
──┴──────────┘
```

Promote a faraway replica

You use the `faraway-replica promote` command to promote an existing replica to a standalone single-node or primary/standby high-availability cluster. You can use either interactive mode or specify the settings with flags on the command line. This example shows interactive mode:

```
biganimal faraway-replica promote
```

```
./biganimal faraway-replica promote
? Source Replica Provider ID: Azure
? Source Replica Region ID: France Central
? Source Replica Name: abcd-replica-1
? Promoted cluster name: abcd-2
? Promoted cluster architecture: High Availability
? Number of standby replicas: 2 Replicas
? Enable read-only workloads: No
? Promoted cluster password: ************
? Instance type: D2s v4(2vCPU, 8GB RAM)
? Volume type: Azure Premium Storage
? Volume properties: P1 (4 Gi, 120 Provisioned IOPS, 25 Provisioned MB/s)
? Networking: Public
? By default your cluster allows all inbound communications, add IP allowed list to restrict the access: No
? Cloud Provider Subscription ID: "111,222"
? Service Account IDs, (leave empty to stop adding): "id1@iam.gcp"
? Add database config in format "application_name=sample_app&array_nulls=true":
autovacuum_max_workers=5&autovacuum_vacuum_cost_limit=3000&checkpoint_completion_target=0.9&checkpoint_timeout=15min&cpu_tuple_cost=0.0
ffective_cache_size=0.75 * ram&maintenance_work_mem=(0.15 * (ram - shared_buffers) / autovacuum_max_workers) > 1GB ? 1GB : (0.15 *
(ram - shared_buffers) / autovacuum_max_workers)&random_page_cost=1.1&shared_buffers=((0.25 * ram) > 80GB) ? 80GB : (0.25 *
ram)&tcp_keepalives_idle=120&tcp_keepalives_interval=30&wal_buffers=64MB&wal_compression=on
? Backup Retention Period, use strings like '7d' or '2w' or '3m' to specify days, weeks and months respectively.: 3m
```

You're prompted to confirm that you want to promote the faraway replica. After the faraway replica promotion process is complete, it generates a cluster ID.

IAM authentication CLI commands

To create a cluster that's enabled for IAM authentication, set the `--iam-authentication` flag on the `cluster create` command to `Yes` or in the configuration file to `true`.

To change the IAM authentication setting after creating a cluster, use the `--iam-authentication` flag on the `cluster update` command.

To change the IAM authentication setting when restoring a cluster, use the `--iam-authentication` flag on the `cluster restore` command.

For more information on IAM authentication for AWS, see IAM authentication for Postgres.

Logging and metrics CLI command

You can get the URLs to access Prometheus metrics and logs in your cloud provider's blob storage solution using the `cluster show-monitoring-urls` CLI command.

For single-node and primary/standby high-availability clusters, the syntax of the command is:

```
biganimal cluster show-monitoring-urls {--id | --provider --region \
                                       --name} [--metrics] [--logs]
```

For distributed high-availability clusters, the syntax of the command is:

```
biganimal pgd show-group-monitoring-urls {--id --group-id} [--metrics] \
                                         [--logs]
```

If you don't use the optional flags to specify the output type, the output includes both the metrics URL and the logs URL.

See Other monitoring and logging solutions for more information about using the URLs to access metrics and logs.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Maintenance windows CLI command

You can set and view maintenance windows using the `cluster set-maintenance-window` and `cluster get-maintenance-window` CLI commands.

The syntax of the command is:

```
biganimal cluster [set|get]-maintenance-window
                                    {--id | --provider --region --name}
                                    {--enable} [--start-day --start-time]
```

For more information on maintenance windows, see Periodic maintenance.

11 BigAnimal release notes

The BigAnimal documentation describes the latest version of BigAnimal, including minor releases and patches. These release notes provide information on what was new in each release. For new functionality introduced in a minor or patch release, the content also indicates the release that introduced the feature.

Month
March 2024
February 2024
January 2024
December 2023
November 2023
October 2023
September 2023
August 2023
July 2023
June 2023
May 2023
April 2023
March 2023
February 2023
January 2023

11.1 BigAnimal March 2024 release notes

BigAnimal's March 2024 includes the following enhancements and bugfixes:

Type	Description
Enhancement	EDB Postgres Extended Server is now available in BigAnimal for single-node, high-availability, and Distributed High Availability clusters.
Enhancement	You can now use Transparent Data Encryption (TDE) for clusters running on EDB Postgres Advanced Server or EDB Postgres Extended Server versions 15 and later in BigAnimal's AWS account. With TDE, you can connect your keys from AWS's Key Management Service to encrypt your clusters at the database level in addition to the default volume-level encryption.
Enhancement	BigAnimal Terraform provider v0.8.1 is now available. Learn more about what's new here and download the provider here.
Enhancement	BigAnimal CLI v3.6.0 is now available. Learn more about what's new here.

11.2 BigAnimal February 2024 release notes

BigAnimal's February 2024 includes the following enhancements and bugfixes:

Type	Description
Enhancement	Added **US West 2** and **East US 1** regions for Azure on BigAnimal hosted accounts.
Enhancement	Added **Asia Pacific (Hong Kong)** region for AWS on your cloud accounts.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Type	Description
Enhancement	Added access key authentication method and machine user to support access key authentication method.
Enhancement	BigAnimal CLI v3.6.0 is now available. Learn more about what's new here.

11.3 BigAnimal January 2024 release notes

BigAnimal's January 2024 includes the following enhancements and bugfixes:

Type	Description
Enhancement	BigAnimal has added new integrations with third-party monitoring services. You can now set up monitoring integrations with DataDog and New Relic on your BigAnimal Projects.
Enhancement	BigAnimal now supports adding a storage volume to your cluster for the Write-Ahead Log (WAL). Dedicated WAL storage volumes significantly improve write performance for WAL files, boosting the IO of the overall cluster.
Enhancement	BigAnimal Terraform provider v0.7.0 is now available. Learn more about what's new here and download the provider here.
Enhancement	BigAnimal CLI v3.5.0 is now available. Learn more about what's new here.
Enhancement	BigAnimal now supports pausing and resuming clusters on demand. You can now pause clusters when you aren't using them without losing your data or configurations, giving you more control over your cluster operations as well as helping you save on compute costs.

11.4 BigAnimal December release notes

BigAnimal's December 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Built-in monitoring is now available for BigAnimal clusters. You can view critical database and infrastructure level metrics for your clusters.

11.5 BigAnimal November release notes

BigAnimal's November 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Added support for new extension PostGIS to allow storage and processing of geospatial data types in Postgres on BigAnimal.
Enhancement	BigAnimal CLI version 3.3.0 is now available.
Enhancement	Added support for EDB Postgres Advanced Server and EDB Postgres Extended Server version 16.
Enhancement	Added support for AWS Europe (Spain) region.
Enhancement	BigAnimal is now available on AWS marketplace.

11.6 BigAnimal October release notes

BigAnimal's October 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Added support for PostgreSQL version 16.
Enhancement	BigAnimal CLI version 3.2.0 is now available.
Enhancement	BigAnimal Terraform version 0.6.1 is now available.
Enhancement	Added support for new extension pg_cron, a cron-based job scheduler for Postgres.

11.7 BigAnimal September release notes

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

BigAnimal's September 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Added support to deploy fully managed Postgres database-as-a-service in BigAnimal's Azure account.
Enhancement	Added support for database-level authentication on bring-your-own-account deployments using your cloud account's Identity and Access Management (IAM) credentials. Admins can now manage database users centrally within your cloud provider's IAM service, and database users can authenticate to Postgres using their existing Azure Active Directory, AWS IAM, or Google Cloud IAM credentials.

11.8 BigAnimal August release notes

BigAnimal's August 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	EDB Postgres Distributed (PGD) on BigAnimal is now generally available.
Enhancement	Added support for BigAnimal hosted provisioning with CLI version 3.0.0.

11.9 BigAnimal July release notes

BigAnimal's July 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Added support Google Cloud, in your cloud account or ours. With the addition of Google Cloud, you can now run BigAnimal on the three largest cloud service providers.
Enhancement	Added support for GCP with CLI version 2.1.0.
Enhancement	BigAnimal Terraform provider version 0.5.1 is now available.
Enhancement	Added support for several new extensions, including EDB Wait States, EDB Query Advisor, and Postgres Failover Slots.

11.10 BigAnimal June release notes

BigAnimal's June 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Added new deployment and payment options. You can now choose to deploy fully managed Postgres database-as-a-service in BigAnimal's cloud account, with simplified credit card billing options to get your workloads up and running seamlessly.
Enhancement	Added support for a self-service pay-as-you-go purchase experience for BigAnimal.
Enhancement	You can now configure Postgres superuser privileges for your BigAnimal clusters directly from the portal. Enable superuser access when creating or editing a single-node or high availability cluster if you need to bypass all permission checks on your database.
Enhancement	Added support for the AWS Asia Pacific South 2 (Hyderabad) region.
Enhancement	Added support to invite other users by their organizations.

11.11 BigAnimal May release notes

BigAnimal's May 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	PgBouncer is now available for single node and high availability clusters with BigAnimal. You can now toggle on connection pooling with the integration available in BigAnimal.
Enhancement	Added support for additional AWS region: Asia Pacific Southeast 3 (Jakarta).
Enhancement	Added support to the CLI for provisioning PGD with v2.0.0.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

11.12 BigAnimal April release notes

BigAnimal's April 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	BigAnimal Terraform provider version 0.4.0 is now available.
Enhancement	Multi-region EDB Postgres Distributed is now available in preview.

11.13 BigAnimal March release notes

BigAnimal's March 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Added support for custom maintenance windows. With custom maintenance windows, you can now have more control over the database restarts during the maintenance upgrade events.
Enhancement	Added Autocluster, Refdata, and EDB Postgres Tuner extensions. - Autocluster — A table access method extension from the Advanced Storage Pack that supports faster access for clustered data. - Refdata — A table access method extension from the Advanced Storage Pack that increases performance and scalability of foreign key lookups on normalized data models. - EDB Postgres Tuner — Available in BigAnimal to provide intelligent parameter tuning recommendations based on your clusters' settings.
Enhancement	BigAnimal CLI v1.14.0 is now available.
Security fix	Added PCI DSS Compliance Certification.

11.14 BigAnimal February release notes

BigAnimal's February 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Added support for EDB Postgres Advanced Server version 15.
Enhancement	BigAnimal Terraform provider version 0.3.0 is now available.

11.15 BigAnimal January release notes

BigAnimal's January 2023 release includes the following enhancements and bug fixes:

Type	Description
Enhancement	Added support for an additional AWS region: AWS Asia Pacific Southeast 2 (Sydney).
Enhancement	BigAnimal CLI v1.12.0 & v1.13.0 now allows users to provision faraway replicas and get monitoring info for their clusters.

12 Known issues and limitations

These known issues and/or limitations are in the current release of BigAnimal and the Postgres deployments it supports:

- Known issues with distributed high availability

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

12.1 Known issues with distributed high availability/PGD

These are currently known issues in EDB Postgres Distributed (PGD) on BigAnimal as deployed in distributed high availability clusters. These known issues are tracked in our ticketing system and are expected to be resolved in a future release.

For general PGD known issues, refer to the Known Issues and Limitations in the PGD documentation.

Management/administration

Deleting a PGD data group may not fully reconcile

When deleting a PGD data group, the target group resources is physically deleted, but in some cases we have observed that the PGD nodes may not be completely partitioned from the remaining PGD Groups. We recommend avoiding use of this feature until this is fixed and removed from the known issues list.

Adjusting PGD cluster architecture may not fully reconcile

In rare cases, we have observed that changing the node architecture of an existing PGD cluster may not complete. If a change hasn't taken effect in 1 hour, reach out to Support.

PGD cluster may fail to create due to Azure SKU issue

In some cases, although a regional quota check may have passed initially when the PGD cluster is created, it may fail if an SKU critical for the witness nodes is unavailable across three availability zones. To check for this issue at the time of a region quota check, run:

```
biganimal-csp-preflight --onboard -i d2s_v3 -x eha <azure-sub-id> <azure-region>
```

If you have already encountered this issue, reach out to Azure support:

```
We're going to be provisioning a number of instances of <SKU TYPE> in <TARGET REGION> and need to be able to provision these
instances in all AZs. Can you please ensure that subscription <AA BYOA AZURE SUBSCRIPTION> is able to provision this VM type in all
AZs of <TARGET REGION>. Thank you!
```

Changing the default database name is not possible

Currently, the default database for a replicated PGD cluster is `bdrdb`. This cannot be changed, either at initialization or after the cluster is created.

Replication

Replication speed is slow during a large data migration

During a large data migration, when migrating to a PGD cluster, you may experience a replication rate of 20 MBps.

PGD leadership change on healthy cluster

PGD clusters that are in a healthy state may experience a change in PGD node leadership, potentially resulting in failover. The client applications will need to reconnect when a leadership change occurs.

Extensions which require alternate roles are not supported

Where an extension requires a role other than the default role (`bdr_application`) used for replication, it will fail when attempting to replicate. This is because PGD runs replication writer operations as a `SECURITY_RESTRICTED_OPERATION` to mitigate the risk of privilege escalation. Attempts to install such extensions may cause the cluster to fail to operate.

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.

Migration

Connection interruption disrupts migration via Migration Toolkit

When using Migration Toolkit (MTK), if the session is interrupted, the migration errors out. To resolve, you need to restart the migration from the beginning. The recommended path to avoid this is to migrate on a per-table basis when using MTK so that if this issue does occur, you retry the migration with a table rather than the whole database.

Ensure loaderCount is less than 1 in Migration ToolKit

When using Migration Toolkit to migrate a PGD cluster, if you adjusted the loaderCount to be greater than 1 to speed up migration, you may see an error in the MTK CLI that says “pgsql_tmp/': No such file or directory.” If you see this, reduce your loaderCount to 1 in MTK.

Tools

Verify-settings command via PGD CLI provides false negative for PGD on BigAnimal clusters

When used with PGD on BigAnimal clusters, the command verify-settings in the PGD CLI displays that a “node is unreachable.”

13 BigAnimal knowledge base and FAQs

The BigAnimal Support team provides a knowledge base with articles addressing frequently asked questions about using BigAnimal. The types of articles range from troubleshooting to step-by-step instructions for specific use cases. The topics range from onboarding to how to tune your database.

Follow the knowledge base to be notified of new articles.

Setting up specific identity providers

See the following articles for step-by-step instructions for setting up specific identity providers:

- Using Auth0 as your identity provider
- Using AWS IAM Identity Center as your identity provider
- Using Azure AD as your identity provider
- Using Google Workspace (G Suite) as your identity provider
- Using Okta as your identity provider

Creating links to Oracle from EDB Postgres Advanced Server clusters

See the following articles for step-by-step instructions for creating links to remote Oracle servers from EDB Postgres Advanced Server clusters:

- Using edb_dblink_oci
- Using dblink_ora_connect()

Deploying PgBouncer outside of BigAnimal

See the following article for step-by-step instructions for setting up deploying PgBouncer outside of BigAnimal:

- How to configure PgBouncer with a BigAnimal cluster

Copyright © 2009 - 2024 EnterpriseDB Corporation. All rights reserved.