## Contents

1 What’s New ........................................................................... 2
  1.1 Limitations ................................................................. 2
2 EDB Ark - Overview .......................................................... 4
  2.1 Architecture Overview .................................................. 6
  2.2 Using Ark as a Template Only User ............................... 8
  2.3 Using a Federated Console ............................................. 9
3 Accessing the Ark Console .................................................. 11
  3.1 Using Self-Registration on an Amazon Hosted Console ..... 13
4 Using the Ark Console ....................................................... 22
  4.1 The Dashboard Tab ..................................................... 23
    4.1.1 Using the Console Switcher Feature ......................... 24
  4.2 The Clusters Tab ......................................................... 25
    4.2.1 The Details Panel .................................................. 28
    4.2.2 The Events Panel .................................................. 33
  4.3 The Backups Tab ......................................................... 34
  4.4 The User Tab .............................................................. 35
    4.4.1 Updating a Password on Amazon AWS ................. 36
5 Creating a Server Cluster .................................................. 37
  5.1 Manually Creating a Cluster .......................................... 39
    5.1.1 Using a Template to Create a Cluster .................... 44
  5.2 Cloning a Server Cluster ............................................. 47
    5.2.1 Using a Template to Clone a Cluster .................... 50
  5.3 Modifying a Cluster’s Administrative Settings ................ 52
6 Connecting to an EDB Ark Cluster ..................................... 54
  6.1 Using ssh to Access a Server ........................................ 56
  6.2 Connecting to EDB Ark with the psql Client ................. 57
  6.3 Using iptables Rules ................................................... 60
7 Managing Backups and Recovery ....................................... 61
  7.1 Performing a Base Backup for Point-In-Time Recovery .... 62
  7.2 Reviewing Stored Backups .......................................... 63
  7.3 Restoring a Cluster from Backup ................................... 64
EDB PostgresArk™

EDB Ark automatically provisions PostgreSQL or EDB Postgres Advanced Server databases in single instances, high-availability clusters, or application development sandboxes. EDB Ark frees DBAs and application developers from the rigors of setting up and administering robust database environments. In minutes, EDB Ark configures a cluster of database machines with:

- Streaming replication
- Connection pooling
- Load balancing
- Automatic failover (transaction or recovery time preferred)
- Secure data encryption
- Rotating user-scheduled backups
- Point-in-time recovery
- Elastic storage
- Elastic scale out

EDB Ark’s automatic scaling of storage resources and scale out of read replicas when a database cluster reaches user-defined thresholds provides unattended, around-the-clock responsiveness to unpredictable load demands on your database infrastructure.

This document will demonstrate how to use the EDB Ark console successfully in your cloud-based database management activities.
The following features have been added to the EDB Ark user console for this release.

- Ark now supports EDB Postgres Advanced Server and PostgreSQL version 12 database clusters.
- EnterpriseDB has introduced a new RPM repository structure that combines the contents of the previous three repositories into a single repository. The new structure is supported by Ark. For information about connecting to the repository, visit https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%2004-09-2019.pdf.
- Ark now provides extended support for multi-region features (including manual promotion and API support).
- You can use the Promote option (located on a cluster’s context menu in the Details panel) to replace the master node with a standby node. For more information, see the EDB Postgres Ark Getting Started Guide.
- The /templates and /templates/id resources now support the region property.
- Support for Amazon r4.* and r5.8x instance types has been reinstated.

1.1 Limitations

Cloning, Recovering, or Scaling Encrypted Clusters from Previous Versions

Encrypted clusters created with Ark 3.3 or prior may not be used to clone, recover, or scale to a machine type that is not supported by that earlier version. If you will be moving encrypted clusters created with Ark version 3.3 or prior to a new machine type (as supported by Ark 3.4 or 3.5), you will need to:

1. Clone the encrypted cluster to a new, unencrypted cluster.
2. Upgrade the Ark console on which the cluster resides.
3. Clone the unencrypted cluster to a new encrypted cluster of the new machine type.

For detailed information about cloning a cluster, see the EDB Postgres Ark Getting Started Guide.

Cloning with a Template from a Foreign Region

You cannot use a template when cloning from a foreign region.
Monitoring a Federated Console with PEM

Ark does not support federated consoles configured in local PEM server mode.

Federating Consoles that host clusters with the Same Name

If you are upgrading to version 3.4 or 3.5, and plan to federate existing Ark version 3.3 consoles that host clusters with the same name, you must first create a clone of one of the clusters specifying an alternate name; federated consoles cannot be used to host clusters with the same name. For example, if you plan to federate two consoles that both contain a cluster named acctg, you should clone one of the clusters specifying an alternate name for the cluster (i.e. acctg-west). Then, you can federate the consoles; both consoles will have access to acctg and acctg-west.
EDB Ark simplifies the process of provisioning robust Postgres deployments, while taking advantage of the benefits of cloud computing. When used with Advanced Server, EDB Ark also provides a platform with compatibility with the Oracle database, offering dramatic cost savings and competitive advantages.

A cloud is a collection of virtual machines; each virtual machine runs a separate copy of an operating system and an installation of Postgres.

Fig. 2.1: Using EDB Ark in a Private Cloud.

You can specify different combinations of CPU speed, RAM, and disk space to suit your needs when provisioning an
EDB Ark cluster. EDB Ark makes it easy to scale up to a more capable cluster, or scale down as your requirements change.

EDB Ark solves common challenges faced by businesses that need more agility, velocity, and thrift in deploying and using relational, ACID-compliant databases:

- **Develop / Test / Deploy.** Quickly create and delete Postgres databases with standard configurations to support software development and testing activities, then deploy applications to the database or cluster – all at a pace dramatically quicker than physical provisioning.

- **Workload Portability.** The same Postgres database trusted in the datacenter also runs in a cloud cluster with scalability and high-availability.

- **Enterprise-class power.** Postgres was designed to solve critical business challenges requiring reliable, high-performance, ACID-compliant database processing. As the only open source database meeting those requirements, it offers an extremely attractive alternative to more expensive options.

EDB Ark includes the following functionality:

- **Scale computing resources up and out.** EDB Ark automatically scales up storage capacity, and provides a simple button to scale your server class up when data processing loads and usage characteristics require a change in the underlying virtual machine resources.

- **Automatic Connection Pooling and Load Balancing.** EDB Ark maintains a cluster of database nodes, automatically scaling out replicas based on increasing user demand. The integrated connection pooling manager increases database read performance by distributing requests across all cluster members.

- **Self-Healing Failover.** EDB Ark automatically replaces downed database nodes, preserving the continuity and performance of the cluster. Users can choose to replace the master with a new master (preserving all committed transactions) or with a promoted replica (for faster recovery time).

- **Automatic Online backup.** EDB Ark uses user-directed rotating backups to protect your data from loss due to mishaps.

- **Supports data encryption.** EDB Ark offers SSL data encryption that protects data at rest, and is transparent to connecting clients.

- **Cost-saving Compatibility with the Oracle Database.** Using a database that is compatible with Oracle is a reliable, fast and cost-effective way to support Oracle applications in public and private clouds.

- **Web-based interface.** EDB Ark provides easy to use point-and-click cluster lifecycle management from start to finish from your favorite web browser.

- **Database Cloning.** EDB Ark allows you to quickly and easily create developer 'sandboxes' based on real production data, saving System Administrators setup, configuration and data load time.

- **Professional Postgres Support.** EnterpriseDB provides support from Postgres experts who work with top Postgres open source developers.

- **JSON Compatible API Support.** EDB Ark supports a JSON-compatible API.
2.1 Architecture Overview

The Ark console is designed to help you easily create and manage high-availability database clusters from a web browser.

Note: Traditionally, the expression *cluster* refers to a single instance of Postgres managing multiple databases; an EDB Ark database server cluster is a collection of high-availability Postgres server instances that reside in a cloud or on a traditional network.

When you create a new cluster (a group of replicated database servers), EDB Ark initializes one or more Postgres instances (virtual machines) according to your specifications. EDB Ark uses Postgres streaming replication to synchronize replicas in the cluster, and pgpool-II to implement load balancing and connection pooling among all active instances. The following figure provides a general overview of the EDB Ark architecture.

The master node of the cluster contains a host operating system with a running instance of Postgres, along with the load balancer. Database modifications are automatically routed to the master node; any modifications to the master node are subsequently propagated to each replica using Postgres streaming replication.

EDB Ark installs Postgres on each replica node in a read-only hot-standby role that automatically duplicates all data found on the master node, and all changes made to that data. In hot-standby mode, the data is available to service user queries providing read scalability to the cluster. In addition, any schema changes made to the master are also replicated to the replica nodes, making development and deployment of application changes easy and seamless without interruption to normal operations.
Replicas provide balanced user support as needed - if any instance in the cluster goes offline, the cluster’s load is re-balanced among the remaining servers while the instance is automatically replaced.

When used in the default healing configuration, in the event of a failure of the master node, an existing replica is used to replace the failed master node. While the replica nodes are standing by, they are read-only resources, load balancing client queries without a risk of compromising data integrity.

EDB Ark automatically archives data at regular intervals; you can specify a convenient backup window and how many backups to retain when creating a database cluster. EDB Ark also offers backup on demand - simply click the Backup icon to save a copy of the instance. Automatic backups are retained according to your specifications; on-demand backups are retained until you delete them. Each backup is a complete copy of the cluster; you can use a backup to restore a cluster.

EDB Ark makes it easy to scale a database cluster:

- To increase read performance, you can add read replicas to the cluster (manually or automatically).

- To handle expanding data requirements you can increase the amount of storage available (manually or automatically).

- To increase the RAM or CPU processing power of the cluster’s underlying virtual machine, you can manually scale a cluster into a more appropriate server class.
2.2 Using Ark as a Template Only User

Some features of the Ark console are not available to a Template Only user. A Template Only user:

- must specify a template when deploying, scaling, or restoring a cluster.
- is restricted to the scaling policies defined in the template.
- cannot modify a manually-defined cluster created by another user.
- can only scale clusters to a server class that exists in a template that is available to the current tenant.
- may only delete backups of template created clusters.
- may not delete last backup of a template created cluster if the cluster had been deleted.

If you are a Template Only user, the Ark console displays a note in the upper-left header when you log in.

![Template Only User](image)

Fig. 2.4: Template Only User.

For Template Only users, the Ark dialogs used to create a cluster, clone a cluster, or to restore a backup offer a subset of the fields presented on the dialogs viewed by a user that is not a Template Only user.

For detailed information about using a template to:

- create a cluster, see Using a Template to Create a Cluster.
- restore from backup, see Using a Template to Restore from Backup.
- clone a cluster, see Using a Template to Clone a Cluster.

Please note: a user that is not restricted to template usage may override template policy when modifying a cluster created with a template.
2.3 Using a Federated Console

An Administrative user can create a federation of consoles; once federated, a user can create or clone clusters or add resources in any region in which one of the federated consoles resides. Standby nodes that reside in other regions will reflect the state of the master node.

Standby nodes that reside in a different region than the master console will be used for load balancing. Failover to nodes that reside on a different region than the master node is not supported. If a master node fails and you do not have a standby node in the same region as the master node, Ark will create a replacement node in the same region as the original master.

If a console is a member of a federation, any other member of the federation will be able to create resources in the region in which the console resides. All of the resources that reside on federated consoles will be visible on the Clusters tab of all of the consoles within the federation.

When connected to a federated console, you can:

• Create a cluster with nodes that reside in multiple regions; the console from which you create the cluster will be responsible for managing the master node.
• Initiate a cluster backup on any cluster within the federation. If the cluster master is managed by the console where the backup was initiated, then the backup will run locally; if the cluster is not managed by the console where the backup was requested, then a backup request will be made to the managing console and the backup will run remotely.
• Delete a node or cluster regardless of the region in which the node or cluster resides.
• Initiate a clone operation regardless of the region in which the node resides; the clone operation will be able to create one or more nodes in any region of the federation.
• Initiate a yum update on any node or cluster of the federation (remote or local).
• Add resources and replicas in any region of a federated console during a manual scaling operation.
• Perform multi-region machine scaling. If the cluster contains nodes running in different regions, then the console that manages the master node will coordinate machine scaling in the foreign regions. A new master node is created first on the new machine type, the old master is removed, and then each replica node is replaced.

On a federated console, dialogs that allow you to create and manage clusters or nodes (create, scale, clone, or delete) will make the regions, VPCs, availability zones, and subnets of other federation members available for selection.

![Fig. 2.5: Selecting a region.](image-url)
For example, you might have two federated consoles; one that resides in `us-east-1` and one that resides in `us-east-2`. After federating the consoles (sharing the URL and token of each console within the federation with the other console), you will be allowed to select the Region in which replica nodes are created or cloned.
Accessing the Ark Console

If your Ark console resides on an Amazon host, the console configuration will determine the user registration process. An administrative user can enable or disable self-registration. If you are an administrative user and need information about enabling or disabling self-registration, please refer to the EDB Ark Administrative User’s Guide. If you are a non-administrative user connecting to the Ark console on an Amazon host with self-registration enabled, see Using Self-Registration on an Amazon Hosted Console.

When you navigate to the URL of an installed Ark console, the console will display a login dialog.

![Login dialog](image)

Fig. 3.1: The Login dialog.
Enter your user name in the **Email** field, and the associated password in the **Password** field, and click **Log In** to connect to the Ark console.

![The Ark console](image)

Fig. 3.2: *The Ark console.*
3.1 Using Self-Registration on an Amazon Hosted Console

If self-registration is enabled, on your first visit to the Ark console, you should create an Amazon role and register an Ark console user.

As part of the registration process for the Ark console, you must create an Amazon IAM role and perform a *handshake* between the Ark console and the Amazon management console. The handshake associates the external ID provided by the Ark console with the Amazon role, and the Role Arn provided by the Amazon console with the Ark user.

Please note that each time you refresh the Ark New User dialog, the external ID displayed on the registration dialog will change; you must have access to both the Ark console and the Amazon management console while registering an Ark user.

To start the registration process, connect to the Amazon management console, and navigate to the Identity and Access Management dashboard.

![The Amazon IAM Dashboard](image)

*Fig. 3.3: The Amazon IAM Dashboard.*

Navigate to the Roles dashboard, and click the Create New Role button.
Fig. 3.4: Provide a role name.

When the Set Role Name dialog opens, specify a name for the new role and click Next Step to specify a role type.

Fig. 3.5: Specify that the role allows EC2 instances to call AWS services.

On the Select Role Type dialog, select the AWS Service Roles radio button, and then the Select button to the right of Amazon EC2 to continue to the Attach Policy dialog.
When the **Attach Policy** dialog opens, do not specify a policy; instead, click **Next Step** to continue to the **Review** dialog.

**Fig. 3.6: The Attach Policy dialog.**

When the **Review** dialog opens, review the information displayed, and then click **Create Role** to instruct the AWS management console to create the described role.

**Fig. 3.7: Review the role information.**

3.1. Using Self-Registration on an Amazon Hosted Console
The role will be displayed in the role list on the Amazon IAM Roles page. The Summary tab will display a Role ARN, but the ARN will not be enabled until the security policy and trust policy are updated.

After completing the Create Role wizard, you must modify the inline policy and trust relationship (defined by the security policy) to allow Ark to use the role. Highlight the role name; then navigate to the Permissions tab and open the Inline Policies menu. Select Click here to add a new policy.

When the Set Permissions dialog opens, select the Custom Policy radio button, and then click the Select button.
Use the fields on the Set Permissions dialog to define the security policy:

- Provide a name for the security policy in the Policy Name field.
- Copy the security policy text into the Policy Document field. The security policy required by Ark is available in AWS Resources.

After providing security policy information, click Apply Policy to return to the Role information page. Then, select the Edit Trust Relationship button (located in the Trust Relationships section) to display the Policy Document.
Replace the displayed content of the policy document with the content of the file available in *AWS Resources*.

**EDB-PPCD-CONSOLE** is a placeholder within the trust policy. You must replace the placeholder with the *External ID* provided on the *Step 2* tab of the *Ark* console *New User Registration* dialog.

To retrieve the *External ID*, open another browser window and navigate to the Log In page of your *Ark* console. Click the Register button to open the New User Registration dialog.
Enter user information in the User Details box located on the Step 1 tab:

- Enter your first and last names in the First Name and Last Name fields.
- Enter a password that will be associated with the user account, and confirm the password in the Password and Verify Password fields.
- Provide an email address in the Email field; please note that the email address is used as the Login identity for the user.
- Use the drop-down listbox in the Cloud Provider field to select the host on which the cloud will reside.
- Enter the name of the company with which you are associated in the Company Name field.

When you’ve completed Step 1, click Next to access the Step 2 tab.
The Step 2 tab of the New User Registration dialog will display a random External ID number. Copy the External ID from the Step 2 dialog into the trust policy, replacing EDB-PPCD-CONSOLE. Please note that you must enclose the External ID in double-quotes (""). Click the Update Trust Policy button to save your edits and exit the dialog.

Your Amazon IAM role ARN is displayed on the IAM Roles detail panel of the Amazon management console. Highlight a role name to display the assigned value on the Summary page.

Enter your Amazon IAM role ARN in the Role Arn field on the Step 2 dialog, and click Finish to complete the registration. Select Cancel to exit without completing the registration.

After completing the registration, you can use the Login/Register dialog to access the Ark console.

3.1. Using Self-Registration on an Amazon Hosted Console
Enter the registered email address in the Username field, and the associated password in the Password field, and click Log In to connect to the Ark console.
CHAPTER 4

Using the Ark Console

The Ark console has four browser tabs that help you to manage clusters that reside in an AWS or Azure cloud:

- The Dashboard tab provides an overview of the current resources in use, a quick-start Launch DB Cluster button, and (optionally), links to documentation and tutorials.
- The Clusters tab provides a management and information resources for your clusters.
- The Backups tab provides a list of the existing snapshots of your Ark clusters and buttons that allow you to create or delete a snapshot.
- The User tab provides a graphical management interface that you can use to review and manage your user account information.

Please note that some features of the Ark console are not available to a Template Only user.
4.1 The Dashboard Tab

The **Dashboard** tab provides an overview of the EDB Ark service status, resources, useful information links and a quick-start **Launch DB Cluster** button.

**Fig. 4.1: The Dashboard tab.**

To launch a cluster from the **Dashboard** tab, use the **Tenant** drop-down listbox to select the tenant in which the cluster will be created. Then, use the **Launch DB Cluster** button located in the **Getting Started** panel to open the **Create New Cluster** dialog and define the cluster attributes. For more information about defining a cluster, see **Creating a Server Cluster**.

The **Resources** panel contains a table that displays the resource usage (instances, snapshots, and volumes) for each role.

The **Hot Topics** panel provides a link to the EDB Ark website.

Use the links in the **EDB Ark Tutorials and Documentation** section to access EDB Ark and Postgres documentation.
4.1.1 Using the Console Switcher Feature

The console switcher provides convenient access to a list of user-defined console names and their associated addresses. When you select a name from the **Consoles** drop-down listbox, the Ark console opens a browser tab and navigates to the address associated with the shortcut name.

![Consoles drop-down](image)

*Fig. 4.2: The Consoles drop-down.*

An Ark administrative user can use management features located on the Admin tab of the administrative console to add consoles to the list, or remove consoles from the list. For more information about populating the console switcher, please see the *EDB Ark Administrative User’s Guide.*
4.2 The Clusters Tab

Use the **Clusters** tab to create, monitor and manage active clusters that reside in the cluster.

![The Clusters tab.

Indicators in the columns to the right of a cluster name display the current health of the cluster. Click on a column name to sort the contents of the column; click a second time to reverse the sort-order.

- The **VM** column displays the state of the virtual machine on which the cluster resides.
- The **HA** column displays the state of the high-availability cluster.
- The **DB** column displays the state of the database server.
- The **UP** column displays the current status of the packages installed on the cluster. Periodically, the cluster manager performs a check to see if the packages are up to date.

Status indicators provide quick visual feedback about each feature:

- A green checkmark indicates that an object is healthy.
- A yellow alert symbol calls attention to an object that requires attention.
- A red error symbol signifies that an object is not available.
- A busy-indicator signals that the cluster is processing a request.
- A question mark indicates that the state of the resource is unknown.
Management options are on the node’s context menu; right-click the node name to access the menu. The menu is context-sensitive; when applicable, select from:

- **Download DB Logs** to download database server logs.
- **Download pgPool Logs** to download pgPool (load balancing) logs.
- **Promote** to promote the selected standby node to master.

If you choose to download a log file archive, the archive name is in the following form:

```
<address>-<log_type>-<date>.tar.gz
```

Where:

- `address` is the address of the selected node.
- `log_type` is the type of log file (server type or pgPool).
- `date` is the date that the archive was generated.
Use the icons along the left side of the **Clusters** tab to create new clusters or manage existing clusters:

Status indicators provide quick visual feedback about each feature:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Cluster</td>
<td>Use the Add Cluster icon to create a new Postgres cluster.</td>
</tr>
<tr>
<td>Scale Up</td>
<td>Select the Scale Up icon to manually add one or more replicas to the current cluster, or add additional storage to the current cluster servers.</td>
</tr>
<tr>
<td>Scale Down</td>
<td>Use the Scale Down icon to remove one or more specified replicas from the cluster.</td>
</tr>
<tr>
<td>Backup</td>
<td>Select the Backup icon to take a backup of the highlighted cluster (a single backup of the cluster data, and a backup of the cluster configuration files).</td>
</tr>
<tr>
<td>Clone</td>
<td>Select the Clone icon to create an exact duplicate of the master node of the selected database. When you clone a database, only the master node is recreated in the new cluster.</td>
</tr>
<tr>
<td>Upgrade</td>
<td>Use the Upgrade icon to perform a yum update (keeping the same server version) or perform an upgrade to a later server version. After the update, the cluster nodes will be rebooted (initiating any kernel updates required). Please note that a software update can take some time to complete.</td>
</tr>
<tr>
<td>Scale Machine Type</td>
<td>Use the Scale Machine Type icon to change the size of the virtual machine for the selected cluster. EDB Ark will copy the cluster into a new cluster of a different server class (RAM and CPU), and optionally re-assign the IP address of the existing cluster to the new cluster.</td>
</tr>
<tr>
<td>Download SSH Key</td>
<td>Use the Download SSH Key icon to download the SSH key associated with the selected cluster. Each cluster has a unique key that you can use to access nodes in that cluster. Before downloading the SSH key, you should disable pop-up blocker software from restricting pop-ups from the URL/s used by the Ark console or Ark clusters.</td>
</tr>
<tr>
<td>Restart/Reload DB Servers</td>
<td>Use the Restart/Reload DB Servers icon to reload one or more nodes of a selected cluster. Select the instance you wish to restart or reload on the Restart/Reload DB Servers dialog and specify if you would like to Restart or Reload the server before clicking Confirm to initiate the action.</td>
</tr>
<tr>
<td>Cluster Administrative Settings</td>
<td>Use the Cluster Administrative Settings icon to access a popup dialog that allows you to view or modify the ownership and notification email address of a cluster.</td>
</tr>
<tr>
<td>Delete Cluster</td>
<td>Use the Delete Cluster icon to delete the currently selected cluster. A popup dialog will ask you to confirm you wish to terminate the cluster; once terminated, a cluster may only be restored from a backup. By default, the box next to Release elastic IP address is checked. Deselect this option if you wish to retain the IP address for re-use with other clusters. If you release the IP address, it will be made available for use by other clusters. When you terminate an active cluster, backups are not deleted. Backups (including user data) are retained until they are selected and deleted from the Backups tab.</td>
</tr>
</tbody>
</table>

- Select the **Details bar** to view information about the state of the selected cluster.
- Select the **Monitoring bar** to view usage statistics for the selected cluster.
- Select the **Events bar** to review event logs describing activities on the selected cluster.
4.2.1 The Details Panel

Click the Details navigation bar to open the Details panel.

The left pane of the Details panel displays information about the currently selected cluster:

- The name of the cluster
- The date and time that the cluster was created
- The name of the database superuser for the cluster
- The name of the cluster owner
- The email address to which notifications about the cluster will be sent
- The elastic IP address of the master node of the cluster
- The cluster size
- If the cluster is encrypted
- If applicable, the IOPS value for the cluster
- The region in which the cluster resides
- The virtual network or VPC ID in which the cluster resides
- The cluster’s hardware type or Server Class
- The engine type and version that resides on the server
• If a template was used, the template name
• If the cluster is configured to update when provisioned

You can use controls on the Details panel to specify:
• If load balancer health should be monitored
• If database health should be monitored
• Failover preferences for the cluster
• Auto-scaling thresholds for the cluster
• Backup preferences for the cluster
• If continuous archiving should be enabled for the cluster

When you modify the settings on the Details panel, EDB Ark displays a New value saved notice, confirming that the change has been saved.

Please note: If a template was used to specify the configuration details for the cluster, the template may prohibit access to auto-scaling or manual scaling functionality.

Monitoring Load Balancer Health

By default, EDB Ark monitors the health of the load balancer to ensure that service is not interrupted. If the load balancer (pgpool) should fail while monitoring is enabled, PgPool will be automatically restarted. If the load balancer cannot be automatically restarted, EDB Ark will display a warning sign in the DB column next to the cluster name, and send a notification email to the cluster user.

Deselect the Monitor Load Balancer Health checkbox to indicate that you do not wish for load balancer health to be monitored and automatically restarted if an interruption in service is detected.

Monitoring Database Health

By default, the Monitor Database Health checkbox is checked, indicating that EDB Ark is monitoring the health of the database to ensure that service is not interrupted. If the state of the database server changes to any state other than running while Monitor Database Health is checked, Ark will attempt to restart the database.

If the database restart fails, Ark will restore the configuration files to their original settings and attempt a restart. If the server fails to restart after restoring the configuration, Ark will failover to a new instance.

Uncheck the Monitor Database Health checkbox to instruct Ark to not automatically restart the database if the database stops.

Selecting a Cluster Healing Mode

Use the Cluster healing mode radio buttons to specify the type of failover that should be employed:

• Select the Replace failed master with a new master radio button to specify that the cluster manager should create a new master to replace a failed master node.
  
  When replacing a failed master node with a new master node, the data volumes from the failed instance are attached to the new master node, preserving data integrity, while the replicas continue serving client queries.

• Select the Replace failed master with existing replica radio button to specify that the cluster manager should promote a replica node to be the new master node for the cluster.
  
  When replacing a failed master node with an existing replica, a replica node is marked for promotion to master node, while the other replica nodes are re-configured to replicate data from the new master node. Since replica nodes use asynchronous replication, any data that was committed to the old master node, but not pushed to the replica prior to the node failure will be lost.
Please note that replacing a failed master node with a new master node can take a bit longer than promoting a replica node to the role of master, but it does have the advantage of guaranteeing that no committed data will be lost. If recovery time for your cluster is more important than preserving any non-replicated transactions, then select Replace failed master with existing replica as the healing mode.

Adjusting Auto-Scaling Thresholds

Use the Auto-Scaling Thresholds controls on the Details panel to adjust the threshold at which EDB Ark automatically scales up cluster resources. For more information about using the controls, see Adjusting the Automatic Scaling Thresholds.

Modifying Backup Settings

Use the fields in the Backup Settings box to change your backup preferences for the selected cluster:

- Use the Backup Window drop-down listbox to select an optimal time to process cluster backups; specify a time when the number of clients accessing the database is minimal.
- Use the Backup Retention field to specify the number of backups that should be stored for the selected cluster.
- Select the checkbox next to Continuous Archiving (Point-in-Time Recovery) to enable point-in-time recovery for a cluster. When enabled, a base backup is automatically performed that can be used to restore to a specific point in time. All subsequent automatic scheduled backups will also support point-in-time recovery. Note that if you deselect this option, the cluster (and subsequent automatic backups) will be re-configured to not include support for point-in-time recovery.

When point-in-time recovery is enabled, the value specified in the Backup Retention field determines the duration of the point-in-time recovery backup window. For example, if you specify a value of 7, the backup window will be 7 calendar days long. When the backup retention threshold is reached, the oldest base backup is removed, as well as any WAL files required to perform a recovery with that backup.

Reviewing Cluster Connection and Status Information

The DNSNAME table (located on the right side of the Details panel) contains a status overview and connection information for the selected cluster. If you have created replicas, the secondary server nodes are listed below the master node in the tree control; expand the tree control to view the status of the replication nodes.

Status indicators provide quick visual feedback about each feature:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Checkmark]</td>
<td>A green checkmark indicates that an object is healthy.</td>
</tr>
<tr>
<td>![Alert]</td>
<td>A yellow alert symbol calls attention to an object that requires attention.</td>
</tr>
<tr>
<td>![Error]</td>
<td>A red error symbol signifies that an object is not available.</td>
</tr>
<tr>
<td>![Busy]</td>
<td>A busy-indicator signals that the cluster is processing a request.</td>
</tr>
<tr>
<td>![Question Mark]</td>
<td>A question mark indicates that the state of the resource is unknown.</td>
</tr>
</tbody>
</table>

Use the drop-down tab in the upper-right corner of the DNSNAME pane to select the columns that will be displayed in the panel:

- The **AZ** column displays the Availability Zone in which the cluster resides.
- The **VPC ID** column displays the identifier of the VPC on which the cluster resides.
- The **LBPORT** column displays the port number to which a client application should connect to utilize load balancing.
- The **DBPORT** column displays the default listener port for the Advanced Server or PostgreSQL server.
• The `CXN` column displays the current number of connections to the node.

• The `VM` column displays the state of the virtual machine on which the cluster resides.

• The `HA` column displays the state of the high-availability cluster.

• The `DB` column displays the state of the database server.

• The `UP` column displays the current status of the packages installed on the cluster. Periodically, the cluster manager performs a check to see if the packages are up to date. If an update becomes available, the `UP` column will display a yellow alert symbol if the update is non-critical, or a red error symbol if the update is a critical (security) alert.

  If alerted to an out-of-date package, you can use the `Upgrade` icon to invoke a yum update to update the package on all of the nodes on your cluster.
The Monitoring Panel

The Monitoring panel displays graphs that allow you to review statistical usage information about the amount of storage and the CPU load for the selected cluster.

Fig. 4.5: The Monitoring panel displays usage information.

Use the Time Range drop-down listbox to modify the time period that the charted information on the Monitoring panel spans.

The graphs on the Monitoring panel display resource usage information:

- The Data Space chart displays the amount of allocated data space used by the selected cluster. The red line denotes the threshold specified by the Data Space Threshold slider on the Details panel (the threshold at which the cluster will be scaled-up). The blue line indicates the amount of the data space that is currently in use.

- The Connections chart displays a graph of the number of connections to the cluster during the selected time range. The red line denotes the threshold specified by the Connections slider on the Details panel.

- The Load chart displays the processing load placed on the CPU by connecting clients. The value displayed is the actual load average as read from the program, /proc/loadavg. The chart shows the number of jobs in the run queue or waiting for disk I/O, averaged over 15 minute periods.

- The Replication Lag chart displays the replication lag (in seconds) for the cluster. Each replica node is displayed as a separate colored line on the chart; the key at the bottom of the chart identifies the IP address of the node.
4.2.2 The Events Panel

The Events panel displays an event log containing a history of selected events for the connected user.

<table>
<thead>
<tr>
<th>ID</th>
<th>TIME</th>
<th>MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>5721</td>
<td>Tue Sep 17 2010 08:53:23 GMT</td>
<td>2GB expansion of cluster ecg_payables completed.</td>
</tr>
<tr>
<td>5720</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>2GB expansion of cluster ecg_payables started.</td>
</tr>
<tr>
<td>5719</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Data-size threshold reached, scaling up data size for cluster ecg_payables</td>
</tr>
<tr>
<td>5718</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Deleting backup completed for snap 06bdaf9e8f5e5bdc45f5e6</td>
</tr>
<tr>
<td>5717</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Deleting snapshots for backup snap-05be69358f355f30a55a</td>
</tr>
<tr>
<td>5716</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Deleting backup started for snap-05be69358f355f30a55a</td>
</tr>
<tr>
<td>5715</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Finished copying all snapshots of backup snap-06bdaf9e8f5e5bdc45f5e6 to region us-east1</td>
</tr>
<tr>
<td>5714</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Copying backup snapshot-05be69358f355f30a55a to region us-east1</td>
</tr>
<tr>
<td>5713</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Backup of cluster ecg_payables completed.</td>
</tr>
<tr>
<td>5712</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Saving snapshot snap-06bdaf9e8f5e5bdc45f5e6 volume vol-05be69358f355f30a55a (15%)</td>
</tr>
<tr>
<td>5711</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Saving snapshot snap-06bdaf9e8f5e5bdc45f5e6 volume vol-05be69358f355f30a55a (15%)</td>
</tr>
<tr>
<td>5710</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Saving snapshots of attached volumes</td>
</tr>
<tr>
<td>5709</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Backup in progress</td>
</tr>
<tr>
<td>5708</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Backup of cluster ecg_payables started.</td>
</tr>
<tr>
<td>5707</td>
<td>Tue Sep 17 2010 08:53:12 GMT</td>
<td>Deleting replicas...</td>
</tr>
<tr>
<td>5668</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Backup of cluster ecg_payables completed.</td>
</tr>
<tr>
<td>5667</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Saving snapshot snap-0e8874e8eaf30776e volume vol-0e8874e8eaf30776e (100%)</td>
</tr>
<tr>
<td>5666</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Saving snapshot snap-0e8874e8eaf30776e volume vol-0e8874e8eaf30776e (100%)</td>
</tr>
<tr>
<td>5665</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Saving snapshots of attached volumes</td>
</tr>
<tr>
<td>5664</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Backup in progress</td>
</tr>
<tr>
<td>5663</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Backup of cluster ecg_payables started.</td>
</tr>
<tr>
<td>5662</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Backup of cluster ecg_payables completed.</td>
</tr>
<tr>
<td>5661</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Saving snapshot snap-06519a9fa8f5732cd volume vol-06519a9fa8f5732cd (100%)</td>
</tr>
<tr>
<td>5660</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Saving snapshot snap-06519a9fa8f5732cd volume vol-06519a9fa8f5732cd (100%)</td>
</tr>
<tr>
<td>5659</td>
<td>Mon Sep 16 2010 08:45:43 GMT</td>
<td>Saving snapshots of attached volumes</td>
</tr>
</tbody>
</table>

Fig. 4.6: The Events panel displays server activity.

Highlight a cluster name to display only events for that cluster; if you do not select a cluster, the Events panel will display the collected events for the connected user.

- Click a column heading to sort the logged activity by the selected column; click again to reverse the sort order.
- Use a mouse to select multiple rows from the event log for copy and paste operations.
4.3 The Backups Tab

Use the Backups tab to manage cluster backups; the tab displays a list of the available backups.

![Fig. 4.7: The Backups tab of the Ark console.](image)

A backup captures and stores the status and condition of a cluster at a specific point-in-time. Click a column heading to sort the column contents; click again to reverse the sort order.

If the comment in the NOTES column for a specific cluster includes PITR, point in time recovery is enabled on the cluster. When point in time recovery is enabled, the backup can be used to restore your cluster to a state at any given time since the backup was taken.

Use the icons on the left side of the Backups tab to restore or delete backups:

- **Highlight a backup in the list, and click the Recover Backup icon to open a dialog that allows you to restore a cluster from the selected backup. Specify a name for the cluster, and click the Recover button to continue. A popup confirms that the cluster is being restored; navigate to the Clusters tab to monitor the restoration process.**

- **Highlight one or more backups in the list and click the Delete Backup icon to delete the selected backups. A popup will ask you to confirm that you wish to delete the selected backups before the backups are actually deleted.**
4.4 The User Tab

Fields on the User tab allow you to view or modify information about the current user.

![Image of the User tab of the Ark console.](image)

Fig. 4.8: *The User tab of the Ark console.*

To change the First Name, Last Name, or Company Name of the registered user, modify the corresponding fields and click the Apply Changes button. A popup will confirm that the changes have been applied.

The Notification Email field on the User tab displays the default email address that will be used for cluster notifications unless an alternate address is provided. You can optionally:

- provide an alternate email address when a cluster is created (on the Create a new Server Cluster dialog).
- modify a cluster's notification email address on the Administrative Settings dialog.

To change the default notification email address, enter a new address in the Notification Email field, and click the Apply Changes button. A popup dialog will open, prompting you to enter your password to confirm the change of address. Enter your password, and click Confirm to modify the address, or click Cancel to exit the popup without applying the change.

If you elect to change the notification email address, EDB Ark will send a confirmation email to both the old notification address and the new notification address.
4.4.1 Updating a Password on Amazon AWS

If your Ark console is deployed on Amazon AWS, the User tab displays the Amazon Role ARN associated with your Ark user account, and provides an option that allows you to modify your password. To modify your password, click the Change Password button.

![Change Password](image)

Fig. 4.9: The Change Password dialog.

To modify your password:

- Provide your current password in the Current Password field.
- Enter the new password in the New Password field.
- Confirm the new password in the Confirm New Password field.

Click the Confirm button to change the password to the new value; click Cancel to exit the dialog without modifying the password.

When you change your password, a popup will confirm that the password has been changed and Ark will send an email to the notification email address associated with the account.
Creating a Server Cluster

To create a server cluster, you can:

- Use the Launch DB Cluster button (located on the Dashboard tab) to open the Create a new Server Cluster dialog and define a new cluster.

- Click the New Server button (located on the Clusters tab) to open the Create a new Server Cluster dialog and define a new cluster.

If you are not a Template Only user (a user required to use a template when defining a cluster), the Create a New Server Cluster dialog will prompt you to select a deployment option.

If you select Manually Define A Cluster, a dialog opens that allows you to specify detailed information about the cluster configuration. For information about manually defining a cluster, see Manually Creating a Cluster.
You can also use an existing cluster or a backup as a starting point for a new cluster:

- For information about cloning a new server cluster from an existing cluster, see *Cloning a Server Cluster*.
- For information about restoring a backup to create a new cluster, see *Restoring a Cluster from Backup*.
5.1 Manually Creating a Cluster

Before you can connect to Postgres from a client application, you must create a server cluster. Use the Launch DB Instance button (located in the upper left panel of the Dashboard tab) or click the Add Server button on the Clusters tab to open the Create a New Server Cluster dialog.

Please note: not all fields and tabs documented in the following sections are applicable for all console host types.

Fig. 5.2: Specify information about the new cluster on the Step 1 tab.

Use fields on the Create a New Server Cluster dialog to specify information about the new cluster:

- Specify a name for the new server cluster in the Cluster Name field. Ark uses the name specified in the Cluster Name field to identify the cluster when performing management functions.

- Use the drop-down listbox in the Engine Version field to select the version of the Postgres engine that you wish to use.

- Use the drop-down listbox in the Server Class field to specify the size of each cluster node. The server class determines the size and type (compute power and RAM) of each node within the cluster. When you select a server class, the attributes of the selected class are displayed below the server class field.

You can adjust the amount of storage used by the cluster, or number of replicas in the cluster as your resource
demands change. For example, you can start with a m1.small instance, and later, easily upgrade to a more capable c1.medium instance as your performance requirements dictate.

- If your cluster resides on an Amazon host, you can check the box to the left of Use Private IP addresses to display addresses that are in your private network in the VPC field.

- If your cluster resides on an Amazon host, use the drop-down list box in the VPC field to specify the identity of the network in which the cluster should reside.

- Use the drop-down list box in the Number of nodes field to specify the number of server nodes that you wish to create. The name specified in the Cluster Name field will apply to the master node; each additional node will act as a replication server for the master node.

- Use the Storage GB field to specify the initial size of the data space (in Gigabytes).

- Check the box next to Encrypted to indicate that the cluster should be encrypted. EDB Ark uses the aes-xts-plain (512-bit) cipher suite to provide an encryption environment that is both secure and transparent to connecting clients. When encryption is enabled, everything residing on the cluster is encrypted except for the root filesystem.

- If your cluster resides on an AWS host, check the box next to EBS Optimized to specify that your cluster should use an Amazon EBS-optimized instance and provisioned IOPS to guarantee a level of I/O performance.

  The IOPS field is enabled for those clusters that will reside on an EBS-optimized instance. If applicable, use the field to specify the level of I/O performance that will be maintained for the cluster by automatic scaling. The maximum value is 30 times the size of your cluster; for example, if you have a 4 Gigabyte cluster, you can specify a maximum value of 120.

- Check the box next to Perform OS and Software update? to instruct EDB Ark to perform a yum update whenever the cluster is provisioned; this option is disabled for clusters that use statically provisioned servers. The yum update command updates all of the outdated packages that reside on the cluster. The update will occur when a cluster is scaled up, scaled down, or during a failover.

  When you check the box next to Perform OS and Software update?, EDB Ark will warn you that enabling this functionality can significantly slow down cluster operations. Updating packages may slow down cluster maintenance operations; an update can easily take 10 minutes or more, and will initiate a reboot of the node. This setting is persistent; if you enable software updates for a cluster, you cannot directly disable software updates for that cluster at a later time.

- Enter the name of the database superuser in the Master User field.

- Enter the password associated with the database superuser in the Master Password field.

- Use the Notification Email field to provide the email address that will receive notices about changes to the cluster status.

If applicable on your host system, click the Next button to continue to the Step 2 tab.
Use the fields on the Step 2 tab to specify additional database information for each node of the cluster; the Type column identifies if a node is a Master or Replica node:

- Use the Region drop-down listbox to the right of each node to select the region in which the node will reside.
- Use the VPC drop-down listbox to select the VPC in which the node will be created.
- Use the AZ drop-down listbox to select the Availability Zone in which the node will be created.
- Use the Subnet drop-down listbox to the right of each node to select the subnet that the node will use. Please note that if you manually specify a subnet, you must select a subnet that resides on your private network.

Click the Next button to continue to the Step 3 tab.

Use the fields on the Step 3 tab to define security rules that allow access to the cluster. By default, the load balancer port is open to any IP address for client connections; you may choose to delete the rule, and specify a more restrictive IP range.

To delete a rule from the list, highlight the entry and click the Delete Rule button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.

Click the Add Rule button to open the Add Rule dialog and provide access to a port.
On the Add Rule dialog:

- Use the Port drop-down listbox to select the port that can be accessed from the specified CIDR. A non-administrative user can allow access to ports:
  - 9999 - for client connections and load balancing.
  - 5432 or 5444 - the cluster specific database listener port.

An administrative user can use the Add Rule dialog to add a rule that allows SSH access to Port 22.

- Use the CIDR field to specify the address (or address range) that will be allowed access to the server through the selected port.

When you’re finished, click Apply to create the security rule.

Then, click Next to continue to the Step 4 tab.

Use the fields on the Step 4 tab to specify additional database information:

- Use the # of automatic backups to retain field to specify the number of server backups stored. When the specified number of server backups is reached, EDB Ark will delete the oldest backup to make room for a new backup.

  When point-in-time recovery (PITR) is enabled, the value specified in the # of automatic backups to retain setting determines the duration of the PITR backup window. For example, if you specify a value of 7, the PITR backup window will be 7 calendar days long.

- Use the Backup Window field to specify a time that it is convenient to backup the server (you may wish to schedule backups to occur when the CPU load is the lightest).
• Check the box next to Continuous Archiving (Point-in-Time Recovery) to enable point-in-time recovery for the cluster. When enabled, a base backup is automatically performed that can be used to restore to a specific point in time. All subsequent automatic scheduled backups will also support point-in-time recovery. Note that if you deselect this option, the cluster (and subsequent automatic backups) will be re-configured to not include support for point-in-time recovery.

Please Note:

If your cluster resides on an Amazon host that is running CentOS 6.x, point-in-time recovery support is limited to the following regions:

- ap-northeast-1
- ap-southeast-1
- ap-southeast-2
- eu-west-1
- sa-east-1
- us-standard (us-east-1)
- us-west-1
- us-west-2

Use the Previous button or select a tab to return to a tab to review or update information; when you have completed the Create a New Server dialog, click Launch to create the database cluster.

A popup dialog confirms that EDB Ark is creating a new cluster; click the X in the upper-right corner of the popup to close the popup.

![A new database cluster is now being created.](image)

Fig. 5.7: A popup confirms that the new cluster is being created.

Navigate to the Clusters tab of the Ark console to monitor the creation of the cluster.
5.1.1 Using a Template to Create a Cluster

If you select the Launch From Template option when deploying a cluster or are a Template only user, a dialog that offers limited options will open when you deploy a cluster.

![Create a New Server Cluster](image)

Fig. 5.8: Creating a cluster from a template.

Use fields on the Create a New Server Cluster dialog to define your cluster:

- Use the Template Filter panel to filter the list of templates that will be made available in the Template Name drop-down listbox. Select:
  - The database or databases from which you would like to choose.
  - The version or versions from which you would like to choose.
- Use the Template Name drop-down listbox to select the template that you wish to use to configure your cluster. To review template configuration details, select the Full Template Details link; the Template Details dialog opens.
After selecting the template that you wish to use, use the fields on the dialog to finish defining your cluster:

- **Use the Cluster Name field** to specify a name for the new cluster.
- **Use the DB Master User field** to specify the name of the database superuser.
- **Use the DB Master Password field** to specify the password associated with the database superuser.
- **Use the Notification Email field** to provide the email address that will receive notices about changes to the cluster status.
- **Use fields** in the **Availability Zone and Subnet Details** section to specify the regions in which the cluster nodes will be deployed:
  - Use the **Region** drop-down listbox to select the region in which the node will be deployed.
  - Use the **VPC** drop-down listbox to select the vpc on which the node will be deployed.
  - Use the **AZ** drop-down listbox to select the availability zone in which the node will reside.
  - Use the **Subnet** drop-down listbox to select the subnet that the node will use.
- **If applicable, highlight a rule, and use the Add Rule or Delete Rule button** to define security rules that allow access to the cluster. By default, the load balancer port is open to any IP address for client connections; you may choose to delete the rule, and specify a more restrictive IP range.

Fig. 5.9: **Reviewing Template details**

5.1. Manually Creating a Cluster
– To delete a rule from the list, highlight the entry and click the Delete Rule button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.

– Click the Add Rule button to open the Add Rule dialog and provide access to a port. On the Add Rule dialog:
  - Use the Port drop-down listbox to select the port that can be accessed from the specified CIDR.
  - Use the CIDR field to specify the address (or address range) that will be allowed access to the server through the selected port.

After completing the Launch From Template dialog, click the Launch button to provision a cluster that conforms to the cluster configuration.
5.2 Cloning a Server Cluster

With a few simple steps, you can create a developer sandbox that contains a duplicate of the original master node. To clone a cluster,

navigate to the Clusters tab and highlight the name of the cluster you wish to clone; then, click the Clone icon from the left margin. If you are not a Template Only user, a dialog will open that allows you to select your deployment method.

![Choose a clone option](image)

Fig. 5.10: Selecting a Cloning option.

If you are a Template Only user or select the Clone From Template option on the deployment method selection dialog, a dialog opens that allows you to use a pre-defined template for the cluster configuration; for detailed information about using a template to clone a cluster, see Using a Template to Create a Cluster.

If you select Manually Clone A Backup, the dialog shown below opens.
When the Create clone... dialog opens, provide values in the requested fields:

- Provide a name for the new cluster in the Cluster Name field.
- The clone will be created in the Clone Region listed below the cluster name.
- Check the box next to Encryption if you would like the clone to be created in an encrypted cluster.
- Check the box next to Perform OS and Software update? if you would like the server to perform a software update each time the clone is provisioned. Please note: this option is disabled if the database engine is statically provisioned.
• If applicable, check the box next to **Use Private IP addresses** to create the clone your private network.
• If your cluster resides on an Amazon host, use the **VPC** drop-down list box to specify a network name.
• Use the **Availability Zone** drop-down listbox to specify the availability zone in which the new cluster will be created.
• Use the **Subnet** drop-down listbox to select a subnet that will be used by the new cluster.
• Use the **Server Class** drop-down listbox to specify the initial size of the new cluster. The attributes of the selected server class will be listed below the **Server Class** field.
• If your cluster resides on an AWS host, check the box next to **EBS Optimized** to specify that your cluster should use an Amazon EBS-optimized instance and provisioned IOPS to guarantee a level of I/O performance.
• The **IOPS** field is enabled for those clusters that will reside on an EBS-optimized instance. If applicable, use the field to specify the level of I/O performance that will be maintained for the cluster by automatic scaling.
• Check the box next to **Continuous Archiving (Point-in-Time Recovery)** to enable point-in-time recovery on the clone.
• Use the buttons in the **Configure Network Security Rules** section to define security rules for the clone.
  – To delete a rule from the list, highlight the entry and click the **Delete Rule** button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.
  – Click the **Add Rule** button to open the **Add Rule** dialog and provide access to a port. On the **Add Rule** dialog:
    * Use the **Port** drop-down listbox to select the port that can be accessed from the specified CIDR.
    * Use the **CIDR** field to specify the address (or address range) that will be allowed access to the server through the selected port.

When you’ve completed the dialog, click the **Clone** button to create the sandbox.

When you clone a database, only the master node is recreated in the new cluster; for information about manually adding replica servers to the new cluster, see **Manual Scaling**.
5.2.1 Using a Template to Clone a Cluster

A clone deployed with a template will be an exact duplicate of the original master node, but will adhere to the cluster deployment rules described in the template by the system administrator. If you are a Template Only user, you will be required to use a template when cloning a cluster. A non-Template Only user may find it easier (especially when cloning a number of clusters) to use a template to define the properties that are common to multiple deployments.

![Creating a clone of a database.](image)

When the Create clone... dialog opens, provide values in the requested fields:

- Use the Template Name drop-down listbox to select a template that will be used for the new cluster; to review the cluster details associated with the template, click the Full Template Details link.
- Provide a name for the clone in the Cluster Name field.
- If applicable, use the Availability Zone drop-down listbox to specify the availability zone in which the new cluster will be created.
- If applicable, use the Subnet drop-down listbox to select a subnet that will be used by the new cluster.

Use the fields in the Configure Network Security Rules section to define security rules that allow access to the cluster. By default, the load balancer port is open to any IP address for client connections; you may choose to delete the rule, and specify a more restrictive IP range.

To delete a rule from the list, highlight the rule and click the Delete Rule button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.

Click the Add Rule button to open the Add Rule dialog and provide access to a port.
On the Add Rule dialog:

- Use the Port drop-down listbox to select the port that can be accessed from the specified CIDR. A non-administrative user can allow access to ports:

  9999 - for client connections and load balancing.
  5432 or 5444 - the cluster specific database listener port.

  An administrative user can use the Add Rule dialog to add a rule that allows SSH access to Port 22.

- Use the CIDR field to specify the address (or address range) that will be allowed access to the server through the selected port.

When you’re finished, click Apply to create the security rule.

After providing the details for the cluster, click the Clone button to create the clone of the cluster; select Cancel to exit the dialog without creating a cluster.
5.3 Modifying a Cluster’s Administrative Settings

Fields on the Administrative Settings dialog display the current owner and the email address to which notification emails about the state of the cluster are sent. To modify the owner of a cluster or the email address associated with a cluster, highlight the name of a cluster on the Clusters tab, and click the Administrative Settings icon. The dialog shown below opens.

![Administrative Settings dialog](image)

Fig. 5.14: The Administrative Settings dialog.

Use the fields on the dialog to modify the administrative settings for the cluster:

- Use the drop-down listbox in the Owner field to select a new cluster owner. Please note that only those users with permissions to access the tenant on which the cluster resides are included in the list.
- Use the Notification Email field to specify the address to which you wish notices about the state of the cluster to be sent.
- Check the box next to Allow non-SSL DB connections to specify that the database can accept connections from a source that is not authenticated via SSL; this will adjust the settings in the pg_hba.conf file.

Use the fields in the Configure Network Security Rules section to define security rules that allow access to the cluster. By default, the load balancer port is open to any IP address for client connections; you may choose to delete the rule, and specify a more restrictive IP range.

To delete a rule from the list, highlight the rule and click the Delete Rule button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.

Click the Add Rule button to open the Add Rule dialog and provide access to a port.
On the Add Rule dialog:

- Use the Port drop-down listbox to select the port that can be accessed from the specified CIDR. A non-administrative user can allow access to ports:
  - 9999 - for client connections and load balancing.
  - 5432 or 5444 - the cluster specific database listener port.
  - An administrative user can use the Add Rule dialog to add a rule that allows SSH access to Port 22.

- Use the CIDR field to specify the address (or address range) that will be allowed access to the server through the selected port.

When you're finished, click Apply to create the security rule.

After modifying the configuration details for the cluster, click the Confirm button; a dialog will open, prompting you to provide the password associated with the connected session.

Provide a password in the Password field and click Confirm to save your changes and exit, or Cancel to exit without saving the changes.
To connect to an Ark cluster, provide the IP address and port of the server, and the credentials associated with the role defined when the server cluster was created.

![Fig. 6.1: The Details panel on the Clusters tab.](image)

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>AZ</th>
<th>VPC ID</th>
<th>LB PORT</th>
<th>DB PORT</th>
<th>CIN</th>
<th>VM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ec2-13-58-119-207.us-east-2.compute.amazonaws.com</td>
<td>us-east-2c</td>
<td>vpc-0532ef6d4747d0e18</td>
<td>9999</td>
<td>5444</td>
<td>3</td>
<td>✔</td>
</tr>
<tr>
<td>ec2-13-59-245-170.us-east-2.compute.amazonaws.com</td>
<td>us-east-2b</td>
<td>vpc-0532ef6d4747d0e18</td>
<td>9999</td>
<td>5444</td>
<td>1</td>
<td>✔</td>
</tr>
<tr>
<td>ec2-19-135-192-246.us-east-2.compute.amazonaws.com</td>
<td>us-east-2a</td>
<td>vpc-0532ef6d4747d0e18</td>
<td>9999</td>
<td>5444</td>
<td>1</td>
<td>✔</td>
</tr>
<tr>
<td>ec2-18-217-149-52.us-east-2.compute.amazonaws.com</td>
<td>us-east-2b</td>
<td>vpc-0532ef6d4747d0e18</td>
<td>9999</td>
<td>5444</td>
<td>1</td>
<td>✔</td>
</tr>
<tr>
<td>ec2-18-220-43-9.us-east-2.compute.amazonaws.com</td>
<td>us-east-2a</td>
<td>vpc-0532ef6d4747d0e18</td>
<td>9999</td>
<td>5444</td>
<td>1</td>
<td>✔</td>
</tr>
</tbody>
</table>
If you have defined a cluster with two or more servers, client applications should always connect to the load balancing port of the master server (the first DNS name listed on the Details panel). This will ensure that read requests are distributed efficiently across the cluster replicas to maximize performance, while write requests are directed only to the cluster master. Replica server nodes are listed below the master node in the tree view.

- The **DNSNAME** column displays the address of the node; a connecting client should use this address when connecting to a specific server.

- The **LB PORT** column displays the port number to which a client application should connect to utilize load balancing.

  Since only the master node of a multi-server cluster operates in read/write mode, all write queries will be directed to the master node, while any read-only queries may be directed to a replica node.

- The **DB PORT** column displays the default listener port for the Advanced Server or PostgreSQL server. To connect directly to the database listener port, modify the cluster's security group to allow connections from your client.

Use the authentication information (Master User and Master Password) provided on the Create a New Server Cluster dialog to establish the initial connection to the cluster as the database superuser. Please note that connecting with this identity grants you superuser privileges on the server; you should not share this connection information with un-trusted users.

After connecting as the database superuser, you should create lesser-privileged user roles with which non-administrative users will connect.
6.1 Using ssh to Access a Server

Before you can connect to a cluster node with ssh, an administrator must modify the security group of the cluster node to which you are connecting to permit ssh connections. If the console is federated (and the node resides in a remote region), you must modify the security group on the remote console to allow ssh connections. You can modify the security group via the Administrative Settings dialog on the Ark console or at the AWS console.

EDB Ark creates an ssh key when you create a new cluster; each cluster has a unique key. Before connecting to an Ark instance, you must download the ssh key, and adjust the privileges on the key file.

To download your private key, navigate to the Clusters tab, and click the Download SSH Key icon. The Accessing Your Cluster Instance popup opens as shown below.

![Accessing Your Cluster Instance](image)

The popup displays the tenant name, the cluster name, the name that you should use when connecting to the cluster, and the IP address to which you should connect.

Before using the private key, you must modify the permissions on the keyfile. Use the following command to restrict file permissions:

```
chmod 0600 <ssh_key_file.pem>
```

Where `ssh_key_file.pem` specifies the complete path and name of the EDB Ark ssh private key file.

After modifying the key file permissions, you can use ssh to connect to the cluster. Include the complete path to the key file when invoking the command provided on the Accessing Your Cluster Instance popup.

After connecting via ssh, you can:
- Stop, start, or restart the Postgres server.
- Download and install Postgres extensions.
- Use the PostgreSQL Client Applications.
- Invoke PostgreSQL Server Applications.

Please note: Postgres Server applications must be invoked by the Postgres cluster owner (identified when creating an EDB Ark cluster as the Master User). If you are using a PostgreSQL server, the default user name is `postgres`; if you are using Advanced Server, the default user name is `enterprisedb`. To change your identity after connecting via ssh, use the `su` command:

```
# sudo su <database_user_name>
```
6.2 Connecting to EDB Ark with the psql Client

After connecting to a server hosted on EDB Ark with the psql client, you can invoke SQL commands or use meta-
commands to:

- Execute queries
- Insert, update, and delete data
- Create and manage database objects (tables, indexes, views, etc.)
- Create user roles and manage privileges
- Review object and role attributes
- Invoke scripts containing complex (or simple) commands

By default, an EDB Ark cluster is only open to connections via port 9999 on the master node. Port 9999 is a good
choice if you are connecting for the purpose of querying the database, but if you are modifying database objects, or
performing administrative functions, you should connect directly to the server’s listener port.

Please note: some administrative functions, if executed over port 9999, may be directed to the incorrect node of a
multi-node cluster where they may not have the intended effect, or may return an invalid value.

The listener port number is displayed in the DBPORT column of the Details panel of the Clusters tab.

Before connecting to the server’s listener port, an Ark administrator must modify the security group to allow connec-
tions from the host of your client application.

Connecting with psql From a Local Workstation

After installing Advanced Server or PostgreSQL on a local workstation, you can use psql to perform administrative
tasks on an EDB Ark cluster.

To open the psql client on an Advanced Server Windows workstation, navigate through the operating system menu to
the EDB Postgres menu and select EDB-PSQL.

To open a psql client on a PostgreSQL workstation, navigate through the operating system menu to the PostgreSQL
menu, and select SQL Shell (psql).

If you have used a package to install Advanced Server or PostgreSQL, you will find the executable in the bin directory
under your installation. You can invoke the executable at the command line; for example, to start the client on an
Advanced Server host, use the command:

```
/usr/edb/as11/bin/psql -d edb -U enterprisedb
```
Provide connection information for the server to which you are connecting:

- When prompted for a Server, enter the IP address or DNS name of the EDB Ark server. The IP address is displayed in the \texttt{DNSNAME} column on the Details panel of the Clusters tab of the Ark console.

- When prompted for a Database, enter the name of the database to which you wish to connect. By default, an Advanced Server cluster is created with a database named edb; a PostgreSQL cluster is created with a database named postgres.

- When prompted for a Port, enter the port on which the server is listening. For database queries, you can use port 9999; if you are modifying database objects or performing administrative functions, you should use the server’s listener port (5444 for an Advanced Server cluster, 5432 for a PostgreSQL cluster).

- When prompted for a Username, enter the role you wish to use when connecting to the server. The name of the database superuser is specified in the Master User field when defining an EDB Ark server cluster. By default, the Advanced Server database superuser is \texttt{enterprisedb}. The default superuser of a PostgreSQL database is \texttt{postgres}.

- When prompted for a Password, enter the password associated with that role. The database superuser’s password is specified in the Master Password field when defining an EDB Ark server cluster.

After connecting, the prompt will display the name of the database to which you are connected.

**Invoking psql on an EDB Ark Server**

To use a copy of the psql client that resides on the EDB Ark host, first connect to the cluster using ssh:

```
ssh -i path <ssh_key> <root@host_name>
```

After connecting to the host, assume the identity of the database superuser (or a user with sufficient privileges to invoke the client). On an Advanced Server host, use the command:

```
sudo su enterprisedb
```

On a PostgreSQL host, use the command:

```
sudo su postgres
```

Then, invoke the psql client. On an Advanced Server host, use the command:

```
/usr/bin/edb-psql -d edb
```

On a PostgreSQL host, use the command:
/usr/bin/psql -d postgres

Include the -d option to specify the name of the database to which you wish to connect. The session opens as shown below.

![Figure 6.4: A psql session on the EDB Ark host.](image)

To exit the psql client, enter \q.

For information about using psql and the psql meta-commands, please see the Postgres documentation at:
6.3 Using iptables Rules

EDB Ark uses iptables rules to manage security on the console and cluster nodes. Please note that you must not modify the iptables rules provided by EDB Ark.

If you are using iptables on the host of the Ark console, do not modify the following rules:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080</code></td>
<td>Redirect http traffic on port 80 to default port 8080.</td>
</tr>
<tr>
<td><code>iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 443 -j REDIRECT --to-port 8181</code></td>
<td>Redirect https traffic on port 443 to default port 8181.</td>
</tr>
<tr>
<td><code>iptables -I INPUT 1 -p tcp --dport 8181 -j ACCEPT</code></td>
<td>Allow inbound traffic to the default administration port.</td>
</tr>
<tr>
<td><code>iptables -I INPUT 1 -p tcp --dport 8080 -j ACCEPT</code></td>
<td>Allow inbound traffic on default ports.</td>
</tr>
</tbody>
</table>

These rules:

- redirect http and https traffic on ports 80 and 443 to the default ports (8080 and 8181).
- allow inbound traffic to the default administration port.
- allow inbound traffic on 8080 and 8181.
- save the configuration (to preserve the behaviors when the system reboots).

If you are using iptables on an Advanced Server cluster, do not modify the following rules:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>iptables -I INPUT 1 -p tcp --dport 7800:7802 -j ACCEPT</code></td>
<td>Allow inbound traffic from the Ark console on ports 7800 and 7802.</td>
</tr>
<tr>
<td><code>iptables -I INPUT 1 -p tcp --dport 5432 -j ACCEPT</code></td>
<td>Allow inbound traffic on database listener ports.</td>
</tr>
<tr>
<td><code>iptables -I INPUT 1 -p tcp --dport 9999 -j ACCEPT</code></td>
<td>Save the configuration (to preserve the behaviors when the system reboots).</td>
</tr>
</tbody>
</table>

If you are using iptables on a PostgreSQL cluster, do not modify the following rules:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>iptables -I INPUT 1 -p tcp --dport 7800:7802 -j ACCEPT</code></td>
<td>Allow inbound traffic from the Ark console on ports 7800 and 7802.</td>
</tr>
<tr>
<td><code>iptables -I INPUT 1 -p tcp --dport 5432 -j ACCEPT</code></td>
<td>Allow inbound traffic on database listener ports.</td>
</tr>
<tr>
<td><code>iptables -I INPUT 1 -p tcp --dport 9999 -j ACCEPT</code></td>
<td>Save the configuration (to preserve the behaviors when the system reboots).</td>
</tr>
</tbody>
</table>

The rules:

- allow inbound traffic from the Ark console on ports 7800 and 7802.
- allow inbound traffic on database listener ports.
- save the configuration (to preserve the behaviors when the system reboots).
When you use the Ark console to take a backup, EDB Ark makes a copy of the contents of the PostgreSQL `PGDATA` directory. The `PGDATA` directory contains the data and the meta-data required to construct an exact copy of the Postgres data cluster (the data and the database objects that reside within that Postgres instance).

To capture a backup of a cluster, navigate to the Clusters tab, highlight a name in the cluster list, and click the Backup icon. The Backup Data? dialog opens as shown below:

![Backup Data? dialog](image)

Fig. 7.1: *The Backup Data? dialog.*

You can include a reference note about the backup that can be viewed on the Backups tab by adding a message to the Optional notes field on the Backup Data? dialog before clicking the Backup button.

When you click the Backup button, EDB Ark will perform the backup. While EDB Ark performs the backup, the PENDING column of the selected cluster on the Clusters tab will display the message: Backup in progress.
7.1 Performing a Base Backup for Point-In-Time Recovery

When point-in-time recovery is enabled, a base backup is automatically performed that can be used to restore to a specific point in time. All subsequent automatic scheduled backups will also support point-in-time recovery. Note that if you deselect this option, the cluster (and subsequent automatic backups) will be re-configured to not include support for point-in-time recovery.

When point-in-time recovery is enabled, the value specified in the Backup Retention field of the Create cluster dialog determines the duration of the point-in-time recovery backup window. For example, if you specify a value of 7, the backup window will be 7 calendar days long. When the backup retention threshold is reached, the oldest base backup is removed, as well as any WAL files required to perform a recovery with that backup.

Please note that you cannot perform a base backup on a cluster while the database is in recovery and not accepting connections. If you attempt to perform a base backup during recovery, the backup will fail (the failure will be noted on the Events panel of the Clusters tab). You should instead wait until the database recovery is complete to enable point-in-time recovery for the cluster.

Point-in-time recovery is enabled on the Details panel of the Clusters tab. If a base backup fails, you can trigger EDB Ark to perform a base backup by disabling point-in-time recovery, and then (after waiting a few minutes) re-enable point-in-time recovery.
7.2 Reviewing Stored Backups

Navigate to the Backups tab to review a list of stored cluster backups.

![The Backups tab of the Ark console.]

A backup captures and stores the status and condition of a cluster at a specific point-in-time.

- The ID column contains a unique backup identifier.
- The CLUSTER column displays the name of the cluster that was the target of the backup.
- The NOTES column displays an informational note (provided by either the user or the system at the time of backup). Those messages that include (PITR) can be restored to a specific point-in-time within the backup window.
- The ENGINE_VERSION column contains a description of the Postgres version that the saved cluster is using.
- The CAPACITY column contains the storage capacity of the cluster at the time that the backup was taken.
- The STARTED column displays the date and time that the backup was initiated.
- The ENDED column displays the date and time that the backup completed.

You can use the icons on the left side of the Backups tab to restore or delete the selected backup:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>📁</td>
<td>Highlight a backup in the list, and click the Recover Backup icon to open a dialog that allows you to restore a cluster from the selected backup.</td>
</tr>
<tr>
<td>🔴</td>
<td>Highlight one or more backups in the list and click the Delete Backup icon to delete the selected backups. A popup will ask you to confirm that you wish to delete the selected backups before the backups are actually deleted.</td>
</tr>
</tbody>
</table>

Status indicators provide quick visual feedback about each feature:
7.3 Restoring a Cluster from Backup

You can use a template or manually provide cluster properties when restoring a cluster from a backup. To restore a backup into a new cluster, navigate to the Backups tab and highlight the name of a backup. Then, click the Recover Backup icon (located in the left margin).

If you are not a Template Only user, a dialog will open that allows you to select the method by which you will specify cluster preferences.

If you are a Template Only user or select the Restore From Template option on the recovery method dialog, a dialog opens that allows you to use a pre-defined template for the cluster configuration; for detailed information about using a template to clone a cluster, see Using a Template to Restore from Backup.

If you select Manually Restore A Backup, the following dialog opens:
When the Recover Data from a Backup dialog opens:

- If applicable, use the calendar selector in the Recovery Point field to specify the recovery target (the date and time that the database was in the state in which you wish the new cluster to start). The Recovery Point field is only displayed for backups that were taken with point-in-time recovery implemented; you cannot perform a point-in-time recovery with a backup unless point-in-time recovery is enabled for the cluster when the backup was taken.
- Specify a name for the new cluster in the Cluster Name field.
- Check the box next to Encryption to specify that the new cluster should reside in an encrypted cluster. Please
note that you can restore a non-encrypted backup into an encrypted cluster.

- Check the box next to Perform OS and Software update to instruct EDB Ark to perform a yum update whenever the cluster is provisioned. Please note: this option is disabled if the database engine is statically provisioned.

- If applicable, check the box to the left of Use Private IP addresses to restore the backup into a private IP address.

- If your cluster resides on an Amazon host, use the VPC drop-down listbox to select a VPC on which the cluster will reside.

- Use the Availability Zone drop-down listbox to the right of each node to select the availability zone in which the node will reside.

- Use the Subnet drop-down listbox to the right of each node to select the subnet that the node will use.

- Use the Server Class drop-down listbox to specify the server class of the new cluster.

- If your cluster resides on an AWS host, check the box next to EBS Optimized to specify that your cluster should use an Amazon EBS-optimized instance and provisioned IOPS to guarantee a level of I/O performance; The IOPS field is enabled for those clusters that will reside on an EBS-optimized instance. If applicable, use the field to specify the level of I/O performance that will be maintained for the cluster by automatic scaling. The maximum value is 30 times the size of your cluster; for example, if you have a 4 Gigabyte cluster, you can specify a maximum value of 120.

Note that you can increase the IOPS value of your cluster by recovering the cluster from a snapshot into a cluster with a higher value or cloning your database into a cluster with a higher IOPS value.

- Check the box next to Continuous Archiving (Point-In-Time Recovery) to indicate that the new cluster should implement point-in-time recovery. Please note that to restore into a cluster with point-in-time recovery enabled, the backup from which you are restoring must have had point-in-time recovery implemented when the backup was taken. The checkbox will not be available if point-in-time recovery was not implemented when the backup was taken.

- Use the Add Rule button to open a dialog that allows you to open a port for connections from a specified CIDR formatted address. On the Add Rule dialog:

  Use the Port drop-down listbox to select the port that can be accessed from the specified CIDR. A non-administrative user can allow access to ports:

  9999 - for client connections and load balancing.

  5432 or 5444 - the cluster specific database listener port.

  An administrative user can use the Add Rule dialog to add a rule that allows SSH access to Port 22.

  Use the CIDR field to specify the address (or address range) that will be allowed access to the server through the selected port.

  When you’re finished, click Apply to create the security rule and continue.

- Highlight a rule and click the Delete Rule button to remove a security rule.

Click the Recover button to continue, or the Cancel button to exit without starting the recovery process. A popup confirms that the cluster is being restored; close the popup and navigate to the Clusters tab to monitor the restoration process.
Please note: when you restore a backup, the server configuration will match the original configuration, but the server addresses will change.

Please note: when restoring a cluster from backup, you may need to modify parameters in the postgresql.conf file on the restored cluster to reflect the available memory of the new instance if the server class has changed from the original setting (the default value in the Server Class field). After modifying the server configuration, restart the server for the changes to take effect.
7.3.1 Using a Template to Restore from Backup

If you are a Template Only user, you will be required to use a template when restoring a backup into a new cluster. A non-Template Only user may find it easier (especially when restoring a number of clusters) to use a template to define the properties that are common to multiple deployments.

![Fig. 7.6: Using a template to restore from a backup.](image)

If you are using a template when restoring a cluster from backup, use the dialog shown to provide the non-template details:

- Use the Template Name drop-down listbox to select a template that will be used for the new cluster; to review the cluster details associated with the template, click the Full Template Details link.
- If applicable, use the calendar selector to specify a point-in-time to which to recover.
- Specify a name for the new cluster in the Cluster Name field.
- Use the AZ drop-down listbox to the right of each node to select the availability zone in which the node will reside.
- Use the Subnet drop-down listbox to the right of each node to select the subnet that the node will use.
- Use the Add Rule button to open a dialog that allows you to open a port for connections from a specified CIDR formatted address.
- Highlight a rule and click the Delete Rule button to remove a security rule.
The EDB Ark cluster manager constantly monitors the state of each cluster. Each cluster is composed of a single master Postgres instance that operates in read-write mode (performing all writes to the database) and one or more replica Postgres instances. Replica nodes are read-only, automatically duplicating all data found on the master node, and all changes made to that data.

If a replica fails, EDB Ark automatically spins up a new replica instance and attaches it to the master database. The cluster continues operating during the replacement process, with the master servicing writes and reads, and the remaining replicas servicing reads. Overall read performance may degrade for a short period of time until the cluster is returned to full strength.

If a master failover occurs, the server will enforce one of two behaviors, specified by the Cluster healing mode radio buttons, located on the Details panel of the Clusters tab:

- Select the **Replace failed master with a new master** radio button to specify that the cluster manager should create a new master to replace a failed master node. When replacing a failed master node with a new master node, the data volumes from the failed instance are attached to the new master node, preserving all transactions that were committed on the master.

- Select the **Replace failed master with existing replica** radio button to specify that the cluster manager should promote a replica node to be the new master node for the cluster. Choose this option when speed of recovery is important, and your application can tolerate the loss of some transactions. This is the default behavior. When replacing a failed master node with an existing replica, a replica node is marked for promotion to master node, while the other replica nodes are re-configured to replicate data from the new master node. Since replica nodes use asynchronous replication, any data that was committed to the old master node, but not yet pushed to the replica prior to the node failure will be lost.

If you opt to promote a replica to replace the master node, a replacement replica will also be added to the cluster during the failover process, returning the cluster to full strength. This self-healing property is at the heart of providing high availability to cluster users.

Please note that replacing a failed master node with a new master node can take a bit longer than promoting a replica node to the role of master, but it does have the advantage of guaranteeing that no committed data will be lost.
Triggering a Failover

By design, EDB Ark does not perform a failover when the Postgres server is stopped, because the server stop or restart may be intentional:

- A user may intentionally restart the server when performing maintenance or tuning. For example, a server restart is required when updating server configuration parameters; this restart will not invoke failover.
- If a user intentionally kills the postmaster process, the server will not failover; the postmaster process is responsible for restarting the server.
- The Postgres server may intentionally perform a server restart. For example, when a backend server process crashes (or is intentionally killed by a user), the Postgres server automatically invokes a restart.

When a failover is complete, EDB Ark does not delete the original master instance of the database; you can use the preserved master instance to perform any post-mortem activities that may be required. If you do not wish to utilize the preserved instance, you should use the management console to delete the instance.
8.1 Manually Promoting a Standby Node

You can use the Promote option on a cluster’s context menu to manually promote a standby node to master node. When you promote the standby, the master node is deleted. You must initiate the promotion from a console that resides in the same region as the standby that is being promoted.

To promote a node to master, right-click on the cluster name in the Details panel, and select Promote from the context menu.

![Fig. 8.1: The Promote option on a context menu](image1)

When you select the Promote option, the console will ask you to confirm that you wish to promote the standby to master and delete the original master node.

![Fig. 8.2: Confirming that you wish to promote a standby](image2)

Click Continue to instruct Ark to perform the promotion. During the promotion, status messages will inform you of the state of the cluster.

![Fig. 8.3: Status messages display the state of the cluster](image3)
The Ark console makes it simple to add replicas and storage to an existing cluster, or to upgrade to a larger server class (i.e. vertical scaling).

- Adding additional replicas to your database cluster increases the CPU power available to handle additional client requests or applications, increasing the number of client connections that can be serviced. When the scale up is complete, each additional replica automatically assumes a share of the read-only workload from incoming queries.

- Adding additional storage to the cluster increases the amount of data that can be stored by the database servers. When you add additional storage to the cluster, each member of the cluster gets the additional storage amount.

- Vertically scaling to a larger server class increases the processing capabilities of your cluster, allowing the server to process customer requests with greater speed.

You can also downsize a cluster by selectively removing a replica. You can machine scale to a smaller machine type to reduce resource usage (cpu/memory) and/or cost.

Please note: if you are a Template Only user, access to automatic scaling behaviors is determined by the configuration specified on the template used to deploy your cluster.
9.1 Manually Adding Replicas and Storage

EDB Ark’s Scale Up dialog makes it simple to manually add additional replicas to a cluster if you find that server resources are strained. The dialog also allows you to increase the amount of storage available to a cluster.

If you specify that EDB Ark should add both storage and replicas, EDB Ark will process the request for additional storage before adding replicas to the cluster. All of the nodes on the cluster will be of the newly specified storage size.

To add a replica or storage space to a cluster, navigate to the Clusters tab, highlight a cluster name, and select the Scale Up icon. The Scale Up dialog opens as shown below:

![Fig. 9.1: The Scale Up dialog.](image)

Use the drop-down listboxes on the Step 1 tab to specify:

- The number of replicas to add to the cluster.
- The region in which each node of the cluster will be provisioned.
- The VPC in which the node will be provisioned.
- The AZ (availability zone) in which the node will be provisioned.
- The subnet that will be used by each node of the cluster.
- The amount of storage memory (in Gigabytes) that will be added to each server in the cluster.

When you’ve completed the dialog, click Next to continue to the Step 2 tab.
Use the Previous button to return to the Step 1 tab to modify specified values. Use the Scale Up button to confirm that you wish to add the specified number of replication servers or the specified amount of memory to the cluster. Use the Cancel button, or simply close the dialog to exit without modifying the cluster.

When scaling begins, EDB Ark will confirm that replicas or memory are being added to the cluster.

Fig. 9.2: The Step 2 tab.
9.2 Manually Removing a Replica

EDB Ark’s Scale Down dialog makes it simple to manually remove one or more unneeded replicas from a cluster. To delete a replica, navigate to the Clusters tab, and click the Scale Down icon. The Scale Down dialog opens as shown below:

![The Scale Down dialog.](image)

Check the box to the left of the name of a replica, and click Next to proceed to the Step 2 tab of the dialog.

![The Step 2 tab.](image)

Click Scale Down to confirm that you wish to remove the replica, or Previous to return to the Step 1 tab. Use the Cancel button, or simply close the dialog to exit without modifying the cluster.
9.3 Manually Changing the Server Class

When your RAM processing needs, CPU power, or other circumstances warrant a larger virtual machine for your application, you can vertically scale to a larger server class. You can either:

- **Use the Scale Machine Type dialog to copy the cluster into a larger server class.**
  
  When you use the Scale Machine Type dialog to move your cluster into a larger server class, you must provide a new name for the upgraded cluster. You can also use the dialog to specify that EDB Ark should re-assign the IP address of the cluster, so the upgrade will be transparent to connecting clients.

  Please note: you may wish to postpone the IP address reassignment to perform configuration tasks or test the new server size.

- **Use the `pg_dump` and `pg_restore` utilities to move the cluster into a larger server class.**
  
  To move to a larger server class, use the `pg_dump` utility to make a backup of the cluster. After backing up the cluster, create a new instance with the larger server class, and use `pg_restore` to restore the cluster on the new instance. For information about using `pg_dump` and `pg_restore`, see *Moving an Existing Database into a New Cluster*.

You can also downsize a cluster by selectively removing a replica. You can machine scale to a smaller machine type to reduce resource usage (cpu/memory) and/or cost.

When you vertically scale your cluster with the Scale Machine Type dialog, EDB Ark will copy the existing cluster into a new cluster of a different server class, and optionally re-assign the IP address of the existing cluster to the new cluster. To open the Scale Machine Type dialog, navigate to the **Clusters tab** and select the Scale Machine Type icon.

![Fig. 9.5: The Scale Machine Type dialog.](image)

Use the fields on the Scale Machine Type dialog to specify details about the new cluster:

- Check the box next to Perform OS and Software update to instruct EDB Ark to perform a yum update whenever the cluster is provisioned. Please note: this option is disabled if the database engine used to provision the cluster is statically provisioned.

- Use the Server Class drop-down listbox to specify the size of the new cluster.

  Please note that if you are a Template Only user, the types listed in the Server Class drop-down listbox will be limited to those types that are included in template definitions for the current tenant.

  When you click the Scale button to start scaling the cluster, EDB Ark will confirm that the scaling is in progress.
Before creating the new cluster, EDB Ark will perform a backup of the original cluster. During the process, status indicators in the PENDING column of the Clusters tab will keep you informed as EDB Ark backs up the original cluster, and initializes the new cluster.
Adding additional replicas to your database cluster increases the number of client connections and queries that each cluster can handle, while maintaining a high-level of overall performance. Each additional replica automatically assumes a share of the read-only workload from incoming queries.

When auto-scaling is enabled, EDB Ark monitors the server storage and connection resources in use, and automatically adds additional resources when usage exceeds a user specified percent.

- **When the % of Storage Size used is reached**, EDB Ark will automatically increase your data space by 50%.
- **When the # of Server Connections is reached**, EDB Ark adds replica nodes.

Please note: if you are a Template Only user, access to automatic scaling behaviors is determined by the configuration specified on the template used to deploy your cluster.
10.1 Adjusting the Automatic Scaling Thresholds

Use the Auto Scale Options controls to adjust the threshold at which EDB Ark automatically scales up cluster resources. The Auto Scale Options controls are located on the Details panel; to access the Details panel, navigate to the Clusters tab, and highlight the name of a cluster.

![Auto Scale Options controls](image)

Fig. 10.1: The Auto Scale Options controls.

Adjust the Auto Scale Options sliders to increase or decrease the thresholds at which automatic scaling is invoked. When you modify the values, EDB Ark will display a New Value Saved notice, alerting you that your changes have been saved.
EDB Ark uses pgPool functionality to implement automatic load balancing. Load balancing increases system performance by distributing client queries to replica nodes, while routing database modifications to the master node. Any modifications to the master node are subsequently propagated to each replica using Postgres streaming replication.

**Utilizing Load Balancing**

By default, load balancing is enabled on an EDB Ark cluster. To utilize load balancing, you should direct client applications to connect to the load balancing port (by default, 9999). A cluster’s load balancing port number is displayed in the LBPORT column on the Details pane of the Clusters tab of the Ark console.

pgPool may direct the following statement types to *either* a primary or a standby node:

- SELECT statements (not listed below)
- COPY TO
- DECLARE
- FETCH
- CLOSE
- SHOW
- SET
- DISCARD
- DEALLOCATE ALL

When deciding which node a query should be routed to, pgPool checks the transaction log number; if the transaction log number on the standby server is lower than the log number on the master, pgPool routes the statement to the master node. This helps to ensure that the data returned by the query is the most recent available.

In some cases, specific clauses within a query statement will signal pgPool to direct a statement to the master node. In other cases, the transaction type, or order of commands within a transaction can direct a statement to the master node. By default, the following transaction types will always be executed on the master node:

- SELECT INTO, SELECT FOR UPDATE or SELECT FOR SHARE statements
• SELECT statements within SERIALIZABLE transactions
• SELECT statements that follow an INSERT statement
• SET SESSION CHARACTERISTICS AS TRANSACTION... READ WRITE statements
• SET transaction_read_only = off statements
• EXPLAIN and EXPLAIN ANALYZE SELECT statements
• START TRANSACTION... READ WRITE statements
• LOCK commands that are stricter than ROW EXCLUSIVE MODE
• Transactions that start with a BEGIN statement
• The nextval() and setval() sequence functions
• Large objects creation commands

Please Note: If your application uses JDBC, and the autocommit option is set to false, the JDBC driver will include a BEGIN and COMMIT statement with each SELECT statement. To enable load balancing when using the JDBC driver, your application must include a call to setAutoCommit(true).

pgPool directs the following non-query statement types to the master node only:

• INSERT
• UPDATE
• DELETE
• COPY FROM
• TRUNCATE
• CREATE
• DROP
• ALTER
• COMMENT
• PREPARE TRANSACTION
• COMMIT PREPARED
• ROLLBACK PREPARED
• LISTEN
• UNLISTEN
• NOTIFY
• VACUUM

Selectively Enforcing Load Balancing

pgPool does not enforce load balancing for SELECT statements with a leading white space or leading comment. For example, the following statement would be directed to the master node:

/*Ignore load balancing*/ SELECT * FROM emp;

To enforce load balancing of SELECT statements with leading white space or comments, modify the pgpool.conf file, and set the ignore_leading_white_space parameter to true.

You can also use the black_list and white_list parameters (located in the pgpool.conf file) to instruct pgPool to direct specific statements or functions to the master node. This is useful for cases where a SELECT statement
(normally directed to a replica) calls a function that in turn might modify the database, and so should be directed to
the master.

**Monitoring Load Balancer Health**

By default, EDB Ark monitors the health of the load balancer to ensure that service is not interrupted. If the load
balancer (pgpool) should fail while monitoring is enabled, PgPool will be automatically restarted. If the load balancer
cannot be automatically restarted, EDB Ark will display a warning sign next to the cluster name on the Details panel
and send a notification email to the cluster user.

Deselect the Monitor Load Balancer Health checkbox (located on the Details panel of the Clusters tab) to indicate that
you do not wish for load balancer health to be monitored and automatically restarted if an interruption in service is
detected.
CHAPTER 12

Customizing Your Cluster

EDB Ark creates fully-functioning, high-availability database clusters of various sizes complete with replication, load balancing, connection pooling, backup and failover capabilities. An EDB Ark cluster can be defined in minutes without any special database knowledge or skills. This characteristic is greatly appreciated by application developers who want to create robust, data-intensive applications quickly, and who may not have the time, inclination, or skills to otherwise achieve the same results. This type of black box setup was designed to dramatically increase the productivity of developers, DBAs, and system administrators alike.

However, there are many users who, while enjoying the black box benefits described above, prefer to take a more hands-on approach to managing their databases. EDB Ark was also designed with these users in mind.

You can also use supporting components to extend the functionality of your EDB Ark cluster; the following sections provide an overview of how to add an extension to a new or existing cluster.

Note: The EDB Ark Administrator’s console provides an easy way to install and maintain the latest server-related packages. Talk to your system administrator about automatically including supporting components for your cluster when provisioning the database engine.
12.1 Adding an Extension to a New Cluster

Supporting components and utilities can extend the functionality of your Postgres cluster. For example, you may want to consider adding EDB Postgres Enterprise Manager for management, monitoring, and statistical analysis functionality, or PostGIS, to provide support for spatial data types and functions.

An administrative user can use the Optional Node Packages field on the Add Engine or Edit Engine dialog to modify a database engine definition, providing the names of optional rpm packages that will be installed (from the specified repository) during provisioning. All engines created with that definition will contain the new component; the component will be provisioned on each replica as well as on the master node. As each rpm is installed, yum will satisfy the dependencies for the new component.

Packages added via the Optional Node Packages field on the master node of the cluster will be provisioned on any standby nodes that are subsequently created. If the package requires manual configuration steps, you will be required to repeat those steps on each node of the cluster; package configurations will not be propagated to standby nodes. If you add a node through cluster operations (such as failover, scaling, or restoring a node from backup), any packages on the new node will also require manual configuration.

For information about modifying a database engine to add a supporting component, see the EDB Ark Administrative User’s Guide.
This section details some of the tasks that are performed outside of the Ark console’s graphical interface:

- Moving an existing database into an EDB Ark cluster
- Manually modifying configuration parameters on a cluster
- Stopping and starting the server
13.1 Moving an Existing Database into a New Cluster

You can use the Postgres `pg_dump` utility to migrate an existing Postgres database (schema, data, and associated database objects) into an EDB Ark cluster.

`pg_dump` creates an archive that contains the commands needed to re-create and populate your existing database. After moving the archive to the master node of the Ark cluster, use `pg_restore` to uncompress and play the SQL commands contained in the archive. The following section will walk you through the process of moving a database to EDB Ark using `pg_dump`.

You can also use the `pg_dumpall` utility to move an entire Postgres cluster (data, schema information, and roles) to EDB Ark; for detailed information about using `pg_dumpall`, please see the Postgres documentation at:

http://www.postgresql.org/docs/11/static/app-pg-dumpall.html

The following example assumes that you have provisioned an EDB Ark cluster and opened a port for SSH connections.

**Step One – Navigate into the directory that contains pg_dump**

Open a terminal window on the system that contains your Postgres source database, assume the identity of the Postgres superuser, and navigate into the `bin` directory that resides under your Postgres installation directory.

On Advanced Server the path to the `bin` directory is:

```
/usr/ppas-9.x/bin
```

On PostgreSQL, the path to the directory is:

```
/usr/pgsql-9.x/bin
```

**Step Two - Create the pg_dump Archive**

Use the `pg_dump` utility to create an archive that contains the commands required to recreate a database. When invoking `pg_dump`, include the `-Ft` flag to instruct `pg_dump` to format the output as a tar file, and the `-U` flag to specify the name of the database superuser:

```
pg_dump -Ft -U <db_superuser> <db_name> > <archive_name.tar>
```

Where:

- `db_superuser` is the name of a Postgres database superuser.
- `db_name` is the name of the database that you wish to move to EDB Ark.
- `archive_name.tar` is the complete path and name of the archive. Please note that you must have permission to write a file to the location specified.

If prompted, enter the password associated with the database superuser.

**Step Three - Move the Archive to EDB Ark**

Use the `scp` command to copy the archive to the master server in the EDB Ark cluster; include the `-i` option to specify the location of your ssh key:

```
scp -i <ssh_key_file> <file_name> <user_name@host_name:target>
```

Where:

- `ssh_key_file` specifies the pathname of the EDB Ark ssh private key file.
- `file_name` specifies the archive name.
- `user_name` specifies the name used to connect to the master node of the EDB Ark cluster.
- `host_name` specifies the host name of the master node of the EDB Ark cluster; the host name is located on the Details panel of the Clusters tab in the Ark console.
**Step Four - Connect to EDB Ark with ssh**

Use ssh to connect to your EDB Ark cluster master node. Provide the user identity of the operating system superuser, and the location of the ssh key (on your local host) in the command:

```
ssh -i/<path>/<ssh_key.pem root@host_name>
```

Where:

- `path` specifies the location of your EDB Ark ssh certificate on the system from which you are connecting.
- `ssh_key.pem` specifies the name of the EDB Ark ssh private key file.
- `host_name` specifies the host name of the master node of the EDB Ark cluster; the host name is located on the Details panel of the Clusters tab in the Ark console.

**Step Five – Navigate into the bin directory on the Ark Host**

After connecting, assume the identity of the database superuser and navigate into the directory on the Ark host that contains the pg_restore utility. On an Advanced Server host:

```
cd /opt/PostgresPlus/CloudDB/bin
```

On a PostgreSQL host:

```
cd /opt/PostgreSQL/CloudDB/bin
```

**Step Six - Invoke pg_restore on the master server in the EDB Ark cluster**

Before invoking the `pg_restore` utility, you must create the target database in the master server; you can use the `createdb` client utility at the command line to create the target:

```
createdb -U <db_superuser> <database_name>
```

Where:

- `db_superuser` specifies the name of the database superuser. On an Advanced Server cluster, the default is `enterprisedb`; on a PostgreSQL cluster, the default is `postgres`.
- `database_name` specifies the name of the database on EDB Ark.

Then, invoke the `pg_restore` utility:

```
pg_restore -Ft -U <db_superuser> /<path>/<archive_name.tar> -d <target_db_name>
```

Where:

- `db_superuser` specifies the name of the database superuser. On an Advanced Server cluster, the default is `enterprisedb`; on a PostgreSQL cluster, the default is `postgres`.
- `path` is the pathname to the archive on the Ark.
- `archive_name.tar` is the name of the archived database.
- `target_db_name` is the name of the target database on the Ark.

Include:

- the `-Ft` flag to specify that the file is an archive
- the `-U` flag to specify the name of a database superuser.
- the `-d target_db_name` flag to specify the name of the target database
Step Seven - Confirm that the Move was Successful

After performing the restore, you can use the psql client to connect to the EDB Ark and confirm that the database has been transferred:

```bash
psql -U <database_superuser> -d <target_db_name>
```

Where:

- `db_superuser` specifies the name of the database superuser. On an Advanced Server cluster, the default is `enterprisedb`; on a PostgreSQL cluster, the default is `postgres`.
- `target_db_name` is the name of the target database.

Use the `\dt` command to view a list of database objects in the current database.

To exit the psql client, enter `\q`; to exit the ssh session, type `exit` and `Return`.

For more information about using the psql client, please see the PostgreSQL core documentation.

For more information about using PostgreSQL utilities to move an existing database into EDB Ark, please see the PostgreSQL core documentation.
Many of the features of a Postgres server may be influenced by settings specified in configuration files:

- The `postgresql.conf` file determines Postgres server behavior as it pertains to auditing, authentication, file locations, resource usage, query planning, statistic gathering, error handling and more.

- The `pg_hba.conf` file controls the type of authentication that should be used when a client application connects to an EDB Ark service. By default, the `pg_hba.conf` file is configured to require clients to provide a valid md5-encrypted password.

- The `pg_ident.conf` file contains user mappings for external authentication methods (like LDAP or GSS-API). Each entry within the `pg_ident.conf` file maps an external user name to his corresponding Postgres user name.

- The `pgpool.conf` file determines the behavior of EDB Ark as it pertains to load balancing.

To modify a configuration file:

1. ssh to the node of the cluster that contains the file you wish to modify.
2. Use your choice of editor to modify the files.
3. Reload or restart the server.

When you add or remove nodes from a cluster, EDB Ark takes a backup of your `pg_hba.conf` and `pgpool.conf` configuration files. Configuration file backups are appended with the date that the backup was taken and a unique identifier; for example, `pg_hba.conf.20140319-140903` identifies a backup of the `pg_hba.conf` file.

When modifying a configuration file, you should make changes only to those files that are not appended with a timestamp and identifier.
13.2.1 Best Practices for Modifying Configuration Files

Please note that changing parameter settings can have unintended consequences, ranging from degraded performance to system crashes. Consequently, we recommend that only an advanced user who accepts these risks, and has experience with both Postgres and cloud environments modify parameter settings.

There are several ways that you can minimize the risks involved when making parameter changes:

- Always make a snapshot backup of your data before making parameter changes.
- Always use a test cluster to test parameter changes, to ensure they have the intended effect before deploying them to your production environment. Create a test environment that mirrors the final target environment as much as possible - this is easy to accomplish by restoring a production backup into a similar size cluster as the original.
- Only change one parameter at a time (or as few as possible when dealing with interdependent settings) and monitor its effect until you are comfortable with the result.
- Make parameter changes on a copy of the existing configuration that is in use for the master or replicas. That way, if the parameter change proves detrimental, it will be easy for you to re-apply the original settings. Keep a backup of the original configuration, so they can be easily restored if necessary.

When adjusting parameters, be mindful of the fact that the master node in the cluster processes both read and write requests, while the replica nodes in the cluster accept only read requests. You can tune the master node and the replica nodes independently to quickly have an impact (either positive or negative) on your write or read performance.

For more information about modifying Postgres server parameters, please see the PostgreSQL core documentation.
13.3 Controlling the Database Server

You can use your platform-specific service controller to control a Postgres database. The service name of the database server in an Ark cluster is clouddb.

Controlling a Service on CentOS 7.x

If your cluster resides on CentOS version 7.x, you can use the systemctl command to control the service. The systemctl command must be in your search path and must be invoked with superuser privileges. To use the command, open a command line, and enter:

```
systemctl <action> clouddb
```

Where `action` specifies the action taken by the service command. Specify:

- `status` to discover the current status of the service.
- `start` to start the service.
- `stop` to stop the service.
- `restart` to stop and then start the service.

Controlling a Service on CentOS 6.x

On CentOS version 6.x, you can control a service at the command line with the service command. The Linux service controller mechanism allows you to start and stop the server gracefully. The command must be in your search path and must be invoked with superuser privileges. Open a command line, and issue the command:

```
service clouddb <action>
```

Where `action` specifies the action taken by the service command. Specify:

- `status` to discover the current status of the service.
- `start` to start the service.
- `stop` to stop the service.
- `condstop` to stop the service without displaying a notice if the server is already stopped.
- `restart` to stop and then start the service.
- `condrestart` to restart the service without displaying a notice if the server is already stopped.
- `try-restart` to restart the service without displaying a notice if the server is already stopped.
13.4 Updating the Server Version on the EDB Ark Cluster

When an update becomes available for a package installed on your cluster, the Ark console will display an alert symbol in the UP column of the Details panel for the cluster, and in the UP column of the DNSNAME table adjacent to the node that requires an update:

![Fig. 13.1: The DNSNAME table.](image)

The overall cluster status (displayed in the top section of the Clusters tab) is based on the values of the nodes within the cluster:

- If all of the nodes within the cluster are up-to-date, the UP column displays a green checkmark.
- If one or more nodes require a non-critical update, the UP column displays a yellow alert symbol.
- If one or more nodes require a critical update, the UP column for the cluster displays a red error symbol.
- If one or more nodes have an unknown package status, the UP column for the cluster displays a grey checkmark.

You can use the Upgrade icon (located on the Clusters tab) to access a dialog that allows you to update the server version on each node within the cluster:

![Fig. 13.2: The cluster upgrade dialog.](image)

Select the radio button next to an option to:

- Select OS and DB Minor Version Upgrade to invoke a `yum update` command and update any outdated packages and perform a minor database version upgrade on each node of the cluster.
- Select DB Major Version Upgrade to select a version and perform a major version upgrade of the server. Please note that this functionality is restricted to users that are not required to use a template when deploying a cluster.

After making a selection, click Upgrade to continue.
13.4.1 Performing a Minor Version Upgrade

If you select the radio button next to OS and DB Minor Version Upgrade and click the Upgrade button, the Ark console will invoke the yum update command on each node of the cluster. The yum update command will update all installed packages to the most recent version available of the same release (i.e., if you are running a 9.6 database server, yum will update your database server to the most recent version of 9.6).

![Upgrade Cluster dialog](image)

Fig. 13.3: The Upgrade Cluster dialog.

Before performing the update, EDB Ark will perform a backup. During the upgrade process, all clients will be disconnected from the server. The updated server will retain the IP address used by the original server. When the update has completed, clients may once again connect.

After performing a yum update, the node will be rebooted, initiating any kernel updates required. When the update completes, EDB Ark will send an email notification that contains a list of the updated packages.

If one or more nodes in your cluster are currently displaying an unknown status, EDB Ark will display an error message. You must correct the problem that is causing the unknown status before EDB Ark can perform an update.

Please note that if the yum update command fails during the upgrade process, EDB Ark will terminate the process and yum update will not be run on any remaining nodes, leaving the cluster partially upgraded.
13.4.2 Performing a Major Version Upgrade

You can use the Upgrade Cluster dialog to upgrade the Postgres server installed on your Ark cluster; the upgrade must be to a more recent version of the same server type and must use the same server image as the current database engine. For example, you may upgrade an Advanced Server version 9.6 database server that resides on a CentOS 6.x host to Advanced Server 11, but you cannot move the server onto a CentOS 7.x host. Similarly, you may not upgrade a PostgreSQL 9.6 database server that resides on a CentOS 7.x host to use an Advanced Server 11 server on a CentOS 7.x host. The server type and host operating system version must remain the same. Please note:

- a major version upgrade may not be performed by a template-only user.
- if over half of the data space allocated to a cluster is used, you must add storage to the cluster before performing the upgrade.

To upgrade a running cluster, select the radio button next to Major Version Upgrade, and then use the drop-down listbox to select a server version. Click Upgrade to continue.

Fig. 13.4: The Upgrade Cluster dialog.

A popup will open, asking you to confirm that you wish to upgrade the server; click the Upgrade button to perform an upgrade. The server will be briefly unavailable during the upgrade process. The upgrade does not change the IP address and listening port of the server.
This section provides helpful troubleshooting information; if you still have unanswered questions after reviewing this section, you can also find solutions through EnterpriseDB:

If you have purchased support, you can log a support ticket:

- in the Customer Portal: http://www.enterprisedb.com/support
- via email: mailto:support@enterprisedb.com
- or by phone: +1-732-331-1320 or 1-800-235-5891 (US Only)

If you have not purchased support, and would like to, you can view your support options at:

http://www.enterprisedb.com/cloud-database/support

You are always welcome to log an issue via email; when time permits, our customer support experts will respond to inquiries from customers that have not purchased support.

You can also find free help on a wide variety of topics in the EnterpriseDB User Forums, at:


Postgres documentation and helpful tutorials are available from the EDB Ark bookshelf, located on the Dashboard tab of the management console.
14.1 Frequently Asked Questions

Problem: Logging into the Console sometimes takes a long time.

This can be attributed to delays in the connection time to the cloud provider. When you log in, the Console Manager must pass your credentials to the provider to log in; any delays at the service provider may slow your connection time.

Problem: I am attempting to connect to my cluster, but don’t know my default database name.

- The name of the default database in an Advanced Server cluster is edb.
- The name of the default database in a PostgreSQL cluster is postgres.

Problem: unable to connect to the load balancing port (9999).

If you are having difficulty connecting to the load balancing port, you should:

- Make sure you are connecting to the master server’s DNS name, rather than a replica’s DNS name; the load balancer resides on the master node of an EDB Ark cluster.
- Make sure that your client application is providing an MD-5 encrypted password when attempting to connect to the load balancing port. The username:password-md5 combination is stored in pgpool_passwd.conf, and is automatically updated when a user changes password, or when a new user is created.

Problem: pgpool keeps issuing the following error: make_persistent_db_connection: s_do_auth failed.

pgpool attempts to connect to each node to perform replication lag checking. This happens unconditionally if pgpool is configured in a master-slave mode and streaming replication is being used (which is the case for EDB Ark). The pgpool community has been alerted to this behavior; please ignore these messages.

Question: How do I stop the Postgres server on a cluster node without triggering a failover process?

To safely stop a Postgres server without triggering failover, you can use either the service command or the pg_ctl utility. For more information, see Controlling the Database Server.

Problem: I am attempting to connect to my Advanced Server database with the psql client, and am getting an error

```
(03/23/2012 13:36:53)--> psql --host=192.0.43.10 -p 9999 -U enterprisedb
Password for user enterprisedb: psql: FATAL: database "postgres" does not exist
```

The psql client expects the default database to be named postgres; the edb-psql client expects the default database to be named edb. If you attempt to connect to an Advanced Server cluster with the psql client without specifying the name of the database to which it should connect, the client will fail to connect.

You can include the -d or --dbname flag, followed by the database name when invoking either client to specify the database to which the client should connect.

Question: I’m trying to drop a database from a cluster, but I am getting an error that there are open sessions.

There are no clients connected. How can I terminate any leftover backend sessions?

It may be that pgpool is retaining a connection to the database. You can use the pg_cancel_backend() or pg_terminate_backend() functions to selectively close connections to the database you wish to drop.

Question: Why do I have to restart pgPool before it will recognize new users that I’ve added to the database server?

pgPool does not check for new Postgres users. EDB Ark has a periodic update process that updates the user list every 20 seconds; if the update process identifies a new user, it sends a reload signal to the pgPool process. After the reload, pgPool will allow new users to login.
Instead of reloading, simply waiting for 20 seconds between the CREATE USER statement and the CREATEDB statement should solve the problem.

**Question: Why are scheduled backups not working?**

If you invoke the `pg_start_backup()` function before performing a manual backup your database, you must remember to invoke the `pg_stop_backup()` function when the backup has completed, or EDB Ark scheduled backups will fail.
14.2 The EDB Ark Email Notification System

EDB Ark invokes an email notification system that will alert you if your cluster changes or encounters a problem. Email notifications are sent to the address used to log in to the management console.

EDB Ark will send an email:

- When a new cluster is created.
- If a server stops (or is terminated).
- When a replica is added to a cluster.
- When memory is scaled up.
- When failover is invoked on a master or a replica.
- If a backup fails.
- If the password associated with your user account changes.

The Notification Email field (on the User tab) allows you to change the notification email associated with your user account; for more information, see *The User Tab*. 
AWS Security Policy

When you define a new Amazon role, you are required to provide a security policy. The following text is an example of a security policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssignPrivateIpAddresses",
        "ec2:Associate*",
        "ec2:Attach*",
        "ec2:AuthorizeSecurityGroup*",
        "ec2:Copy*",
        "ec2:Create*",
        "ec2:DeleteInternetGateway",
        "ec2:DeleteNetworkAcl",
        "ec2:DeleteNetworkAclEntry",
        "ec2:DeleteNetworkInterface",
        "ec2:DeletePlacementGroup",
        "ec2:DeleteRoute",
        "ec2:DeleteRouteTable",
        "ec2:DeleteSecurityGroup",
        "ec2:DeleteSnapshot",
        "ec2:DeleteSubnet",
        "ec2:DeleteTags",
        "ec2:DeleteVolume"
      ]
    }
  ]
}
```
"ec2:DeleteVpc",
"ec2:DeleteKeypair",
"ec2:Describe*",
"ec2:Detach*",
"ec2:DisassociateAddress",
"ec2:DisassociateRouteTable",
"ec2:EnableVolumeIO",
"ec2:GetConsoleOutput",
"ec2:ModifyImageAttribute",
"ec2:ModifyInstanceAttribute",
"ec2:ModifyNetworkInterfaceAttribute",
"ec2:ModifySnapshotAttribute",
"ec2:ModifyVolumeAttribute",
"ec2:ModifyVpcAttribute",
"ec2:MonitorInstances",
"ec2:ReleaseAddress",
"ec2:ReplaceNetworkAclAssociation",
"ec2:ReplaceNetworkAclEntry",
"ec2:ReplaceRoute",
"ec2:ReplaceRouteTableAssociation",
"ec2:ReportInstanceStatus",
"ec2:ResetImageAttribute",
"ec2:ResetInstanceAttribute",
"ec2:ResetNetworkInterfaceAttribute",
"ec2:ResetSnapshotAttribute",
"ec2:RevokeSecurityGroup*",
"ec2:RunInstances",
"ec2:StartInstances",
"ec2:UnassignPrivateIpAddresses",
"ec2:UnmonitorInstances"
],
"Resource": "*",
"Effect": "Allow",
"Sid": "Stmt1407961327680"
},
{
"Action": [
"iam:PassRole"
],
"Resource": "*",
"Effect": "Allow",
"Sid": "Stmt1407961362664"
},
{
"Action": [
"s3:CreateBucket",
"s3:Get*",
"s3:List*"
]
AWS User Trust Policy

When you define an Amazon role, you are required to provide a security policy. The following text is an example of a trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "sts.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```
"Sid": "",
"Effect": "Allow",
"Principal": {
  "AWS": "arn:aws:iam::747919436152:root"
},
"Action": "sts:AssumeRole",
"Condition": {
  "StringEquals": {
    "sts:ExternalId": "EDB-PPCD-CONSOLE"
  }
}
}
EDB design, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres Ark where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.

EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.

EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.
A
Accessing the Ark Console, 11
Adding an Extension, 83
Adding an Extension to a New Cluster, 83
Adding Replicas, 72
Adding Storage, 72
Administrative Settings, 51
Architecture Overview, 5
Automatic Failover, 69
Automatic Scaling, 78
Automatic Scaling Thresholds, 78
AWS Policies, 99
AWS Security Policy, 99
AWS User Trust Policy, 101

C
Clone cluster with template, 49
Cloning a Server Cluster, 46
Conclusion, 103
Connecting to an EDB Ark Cluster, 54
Connecting with psql, 56
Console Switcher, 23
Controlling the Database Server, 90
Create Cluster, 43
Creating a Server Cluster, 37
Customizing Your Cluster, 83

D
Database Management, 85
Details Panel, 27

E
EDB Ark - Overview, 4
Email Notification System, 97
Events Panel, 32

I
iptables rules, 59

K
keyword one, 55, 62

L
Load Balancing, 80

M
Managing Backups and Recovery, 61
Manual Scaling, 72
Manually Adding Replicas and Storage, 72
Manually Changing the Server Class, 75
Manually Creating a Cluster, 38
Manually Promoting a Standby, 70
Manually Removing Replicas and Storage, 74
Modifying a Cluster's Administrative Settings, 51
Modifying Configuration Files, 88, 89
Monitoring Panel, 31
Moving an Existing Database into a New Cluster, 85

N
New Features, 2

P
Password, 35
Performing a Base Backup for PITR, 61
Performing a Major Version Upgrade, 93
Performing a Minor Version Upgrade, 92
PITR Base Backup, 61
promote, 70
psql, 56

R
Reload Configuration, 90
Removing Replicas, 74
Removing Storage, 74
Restart Server, 90
Restoring a Cluster from Backup, 63
S
Self-Registration on AWS, 12

T
Template Only User, 43
Template Restore from Backup, 67
Template-only User, 7
The Backups Tab, 34
The Clusters Tab, 25
The Dashboard Tab, 23
The User Tab, 35
Troubleshooting, 95

U
UP column, 91
Updating on AWS, 35
Updating the Server Version, 91
Using a Federated Console, 8
Using a Template to Clone a Cluster, 49
Using a Template to Create a Cluster, 43
Using a Template to Restore from Backup, 67
Using iptables Rules, 59
Using the Ark Console, 22

W
What’s New, 2