EDB License Management
Licensing Guide

Table of Contents

Introduction to the Licensing Guide

Welcome to the EDB Licensing Guide

At EnterpriseDB (EDB), we are committed to providing you with the tools and clarity you need to succeed with your Postgres deployments. This guide is designed to simplify licensing, set clear expectations, and empower you to confidently manage your EDB subscriptions.

Why This Guide Exists

Licensing can sometimes feel complex, but it doesn't have to be. This guide was created to:

- **Simplify the Process**: Offer clear and actionable information about EDB's licensing terms and policies.
- **Promote Transparency**: Ensure you fully understand your product entitlements and responsibilities.
- **Support Your Success**: Help you focus on business priorities while remaining compliant with licensing agreements.

Whether you're scaling deployments, renewing subscriptions, or seeking clarity on entitlements, this guide will serve as your go-to resource.

Your Responsibilities as an EDB Customer

To help us maintain a productive and seamless partnership, your key responsibilities include:

1. **Understanding Your Subscription**
   Know what your licenses include—such as limits on CPUs, instances, or entitlements.
2. **Monitoring Usage**
   Track deployments and ensure your usage aligns with the agreed licensing terms.
3. **Keeping Accurate Records**
   Maintain clear, up-to-date records of deployments and any scaling changes.
4. **Engaging with EDB**
   If you have questions or need support on scaling, renewals, or policies, we're here to help.

Our Commitment to You

At EDB, we value transparency, simplicity, and trust. By working together, we can ensure you get the full value of your EDB investment while avoiding any licensing uncertainties.

Thank you for choosing EDB as your Postgres partner. We're excited to support your continued success and growth!

Licensing Policies - Counting Cores

Physical Server / Bare Metal

- A physical server has a specified number of CPUs and cores
  - The number of cores in the server is the unit of measure for subscriptions

Virtualized Server with Known and Countable Underlying Hardware

- Full capacity scenario
  - Every physical server core is mapped to one virtual server core that is part of a virtual machine that runs Postgres
  - The total number of cores is the number of subscriptions required
  - **Example:** 8 physical cores mapped to 8 virtual cores would require 8 subscriptions
- Sub-capacity scenario
  - A part of the physical server cores are mapped to virtual server cores that are part of a virtual machine that runs Postgres
  - Only the number of mapped cores makes up the total number of cores for which a subscription is required
  - **Example:** 4 of the 8 physical cores mapped to 4 virtual cores would require 4 subscriptions
- Over-committed scenario
  - Every physical server core is mapped to one or more virtual server cores that are part of a virtual machine running Postgres
  - Only the number of physical cores makes up the total number of cores for which a subscription is required
  - **Example:** 8 physical cores mapped to 16 virtual cores would require 8 subscriptions
- *The number of cores to be considered is the lesser number resulting from the three equations above

Virtualized Server with Unknown or Uncountable Underlying Hardware

- A virtual server has a specific number of virtual CPUs
  - The total number of vCPUs is the number of subscriptions required
  - **Example:** 8 virtual CPUs would require 8 subscriptions

Cloud

• The number of virtual CPUs (vCPUs) in the cloud deployment directly determines the number of subscriptions required — a **1:1 ratio**, regardless of cores or threading
  - **Example:** 8 virtual CPUs would require 8 subscriptions

Containers & Cloud Native Postgres

- A Postgres pod cannot use more than the specified amount of CPU capacity
- CPUs can be specified in either CPU count or miliCPU, which is 1/1000 of a CPU
- The rules for over or sub-subscribing also apply to containerized deployments (see: counting for virtual servers)
- The CPU count is based on the "limits" parameter in the Postgres pod configuration
  - The "limits" parameter is multiplied by the value of the "instances" parameter of your configuration
- The resulting number is the number of subscriptions required
- **Example:**
  - A Postgres pod deployment has:
    - CPU "limits" parameter set to 2 CPUs
    - "Instances" parameter set to 4 instances
  - Total CPU Consumption:
    - 2 CPUs x 4 Instances = 8 subscriptions required

Licensing Policies - Counting Instances

For Community 360, Remote DBA, and Cloud DBA

- An instance-based subscription is required for all instances running Postgres
  - An instance means a Postgres server process:
    - On a unique combination of an IP address
    - On a TCP port
- Each replica or standby node counts as a separate instance
- **Example:**
  - Production Environment: 3 Instances (1 Primary + 2 Standby)
  - Testing Environment: 2 Instances (1 Primary + 1 Standby)
  - Development Environment: 1 Instance (1 Primary)
  - Total Instances: 3 (Production) + 2 (Testing) + 1 (Development) = 6 Instance subscriptions required

Additional Licensing Considerations

UniCore subscriptions are required for nodes that run Postgres

- Nodes only running tools like Postgres Enterprise Manager, EFM, Barman, and the like do not require uniCore subscriptions

UniCore subscriptions are required for both primary and replica standby nodes

- Nodes actively engaged in physical data streaming are classified as either primary replicas or standby nodes and require uniCore subscriptions with the same level of support as production nodes.
- There is no differentiation between warm and hot standby nodes
- A cold standby node (essentially a backup that is not engaged in physical data streaming) requires no subscriptions

Hyper-threading

- CPU hyper-threading is disregarded when the underlying hardware architecture is known. Revert to the aforementioned policies in this document

Audit and Review Process

Transparency and collaboration are key to ensuring a smooth licensing experience. Periodic license audits help both EDB and our customers maintain clarity, compliance, and alignment with licensing agreements. This section outlines what you can expect during a license audit and the responsibilities of both parties.

What to Expect During a License Audit

A license audit is a formal process that verifies your product usage aligns with your EDB licensing agreement. Here's how the process works:

1. **Notification**:
   - EDB will provide advanced written notice of the audit, including its scope, timeline, and required information.
     i. Non-standard circumstances can arise where audit notices are preempted due to real-time information that is flagged as a potential licensing issue, requiring an immediate review for further investigation.
2. **Data Collection**:
   - EDB will provide a worksheet for customers to gather and share deployment and usage details (e.g., CPU counts, instance numbers, container usage, and other relevant data).
   - Tools such as **the PGAI (or Beacon) Agent** or other monitoring systems can assist in automating this process. Customers may opt to automatically report data using the agent by configuring it with these instructions: https://www.enterprisedb.com/docs/tools/agent/usage_reporting/
3. **Review and Validation**:
   - EDB will analyze the provided data to compare usage against licensing entitlements.
4. **Results and Next Steps**:
   - EDB will share the findings and, if necessary, discuss any discrepancies.
   - Customers will have an opportunity to address and resolve issues, such as adjusting licenses and/or purchasing additional subscriptions to align with current and backdated usage within thirty days of the issuance of the final findings report.

The audit process is designed to be collaborative, efficient, and transparent, ensuring minimal disruption to your operations.

Responsibilities of Both Parties

To ensure a smooth and successful audit, both EDB and the customer have key roles to play:

Customer Responsibilities

1. **Provide Accurate Information**:
   ○ Maintain up-to-date and accurate records of deployments, usage, and scaling activities.
2. **Respond Promptly**:
   ○ Share requested data within the agreed expedited timeline to keep the process on track.
3. **Ensure Compliance**:
   ○ Work with EDB to address any discrepancies promptly and align usage with licensing agreements.
4. **Install and configure the PG AI (Beacon) Agent**
   ○ The Agent can provide valuable usage data, helping EDB understand how customers are using our products and facilitating core-based licensing and entitlements

EDB Responsibilities

1. **Communicate Clearly**:
   ○ Provide clear guidance on the audit process, required data, and expectations.
2. **Conduct Audits Fairly and Efficiently**:
   ○ Review data accurately, minimizing disruption to your operations.
3. **Support the Resolution Process**:
   ○ Collaborate with customers to address findings and provide solutions that align with both businesses.
4. **Maintain compliance tools**
   ○ When configured to send usage data, the Agent generates a small report containing environment metadata and resource usage. This includes information like:
     • Operating system details
     • Memory and CPU usage
     • Installed database distribution (e.g., PostgreSQL, EDB Postgres Advanced Server, EDB Postgres Extended Server)
     • version
   This report does not include sensitive data or user-identifiable information.

By working together during the audit process, we can ensure transparency, maintain compliance, and strengthen our partnership. At EDB, we're here to support you every step of the way. As your companies are always changing, we encourage you to leverage your EDB Sales representative to ensure continuous compliance.