# The Expert's Guide to Integrating Postgres


**AUTHORED BY:**

**Marc Linster, Ph.D.**
Chief Technology Officer, EDB

**Vibhor Kumar**
Vice President, Performance Engineering, EDB


POWER TO POSTGRES

# Contents

# 1. Introduction

# Introduction

Over the last decade, open source databases have won the hearts and minds of developers and database operators across the globe. When you double-click on the open source trends, you will clearly see that PostgreSQL® has become the database of choice. The reasons are obvious: innovation, reliability, cost, and support of a wide range of platforms.

This whitepaper takes a detailed look at the integration of PostgreSQL® with the enterprise tech stack to help accelerate development and the move to production.

A 2020 survey of EDB's enterprise customers tells us that they use PostgreSQL for systems of record, for example customer relationship management or ecommerce transactions (54%), systems of analysis, such as data marts or reporting systems (63%), and systems of engagement, such as websites or IoT applications (17%). It is important to understand how to integrate PostgreSQL with all facets of the enterprise stack.

| 54% | 63% | 17% |
|---|---|---|
| ECOMMERCE TRANSACTIONS | REPORTING SYSTEMS | SYSTEMS OF ENGAGEMENT |

**The following sections focus on PostgreSQL integration with:**

- Applications that create and consume data
- Data integration tools to ensure that PostgreSQL does not become a 'data island'
- Development tools for application developers and database developers
- Deployment tools for the software, the schema, and the data
- Security environments for authentication, authorization, auditing, and encryption
- High availability and disaster recovery tools
- Management tools for monitoring, alerting, and tuning
- Infrastructure considerations for CPU, storage, and network integration

# 2. Application Integration

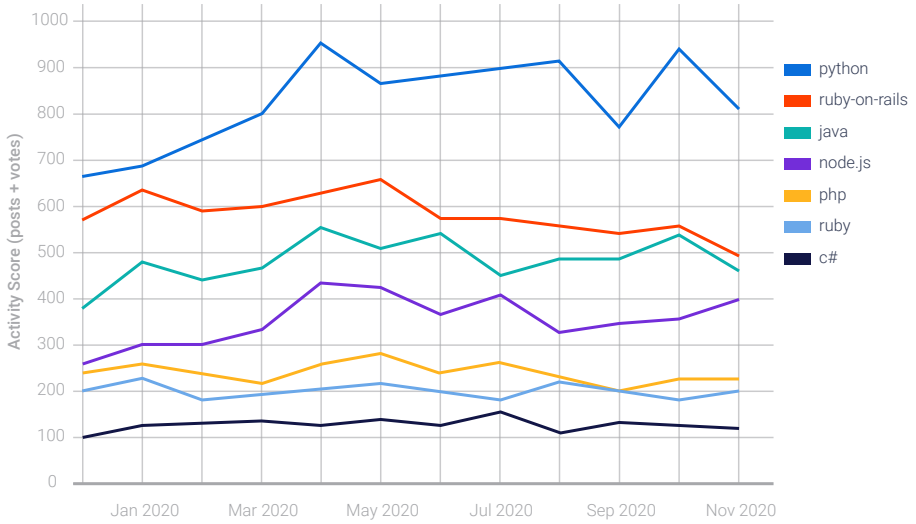
# Application Integration

PostgreSQL provides a C-based API called **libpq**, which is the recommended way to connect for applications written in C. Java-based applications leverage the **PostgreSQL JDBC driver**. ODBC and .NET drivers are also available from the PostgreSQL project. EDB provides Oracle-compatible versions of JDBC, ODBC, and .NET drivers, and OCI compatible drivers that make PostgreSQL look like Oracle to the application. These drivers can to be used in conjunction with EDB's Postgre Advanced Server.

An analysis of **Stackoverflow's PostgreSQL-related traffic in 2020** provides information about what programming languages are popular with PostgreSQL developers. Python is by far the most popular tool. Ruby on Rails appears to be declining in popularity with Java holding steady.

### Stackoverlow's PostgreSQL-related traffic in 2020

Provides information about what programming languages are popular with PostgreSQL developers.




*Stackoverflow analysis of development languages mentioned in the PostgreSQL context*

Django is an extremely popular framework for Python applications that do not rely on the psycopg2 database driver. Java, often used with Hibernate or Spring, is found largely in mature in-house developed applications.

Node.js, especially React and Vue.js, is increasingly used in microservices type architectures and for **web frameworks**.

# 3. Data Integration and Migration

# Data Integration and Migration

Successful PostgreSQL® implementations are fully integrated with the enterprise environment and do not create 'data islands'.

PostgreSQL has a native capability to integrate with the ecosystem: Foreign Data Wrappers (FDW), which are an implementation of the SQL MED standard. FDWs were first introduced in PostgreSQL 9.1 and are now widely used to integrate with remote PostgreSQL databases, Hadoop, MySQL, Mongo, Oracle, and so on. As opposed to ETL (Extract, Transform, Load) type implementation tools, FDWs do not copy or move the data; they access the data in place and push as many operations as possible to the remote (foreign) data store.

Well-implemented FDWs support aggregate and join pushdown, as well as column selection. The FDWs for HDFS, MongoDB, and MySQL are very popular and maintained by EDB. This blog has how-to instructions for using FDWs.

An increasing number of ETL vendors are adding PostgreSQL as sources and targets to their integration tools. **In a 2021 survey of EDB website visitors, the following tools proved to be very popular with PostgreSQL users (in alphabetical order):**

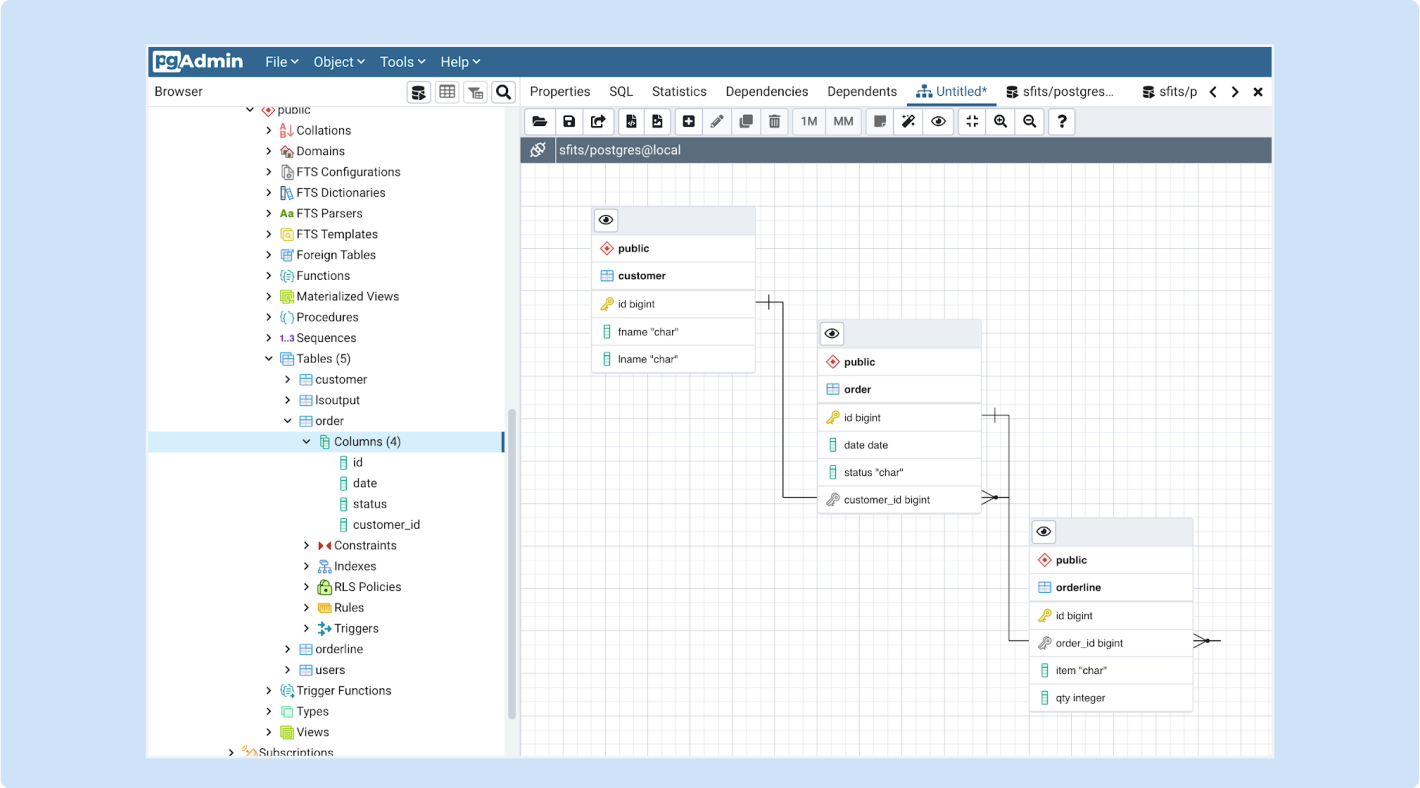
- DB*Loader
- EDB Replication Server
- IBM InfoSphere DataStage
- Informatica PowerCenter
- Microsoft SSIS
- Oracle GoldenGate
- Pentaho Data Integration
- Qlik (Attunity)
- Talend Open Studio

Migration from legacy databases, such as Oracle and SQL Server, is supported by enterprise-grade solutions, such as EDB's Migration Toolkit and EDB Replication Server, and open source tools, for instance, Orafce or OR2PG. Open source tools tend to be limited to migrations of data definitions and common data types. Specialty tools, like EDB's Migration Portal and EDB Replication Server, provide a more complete set of capabilities, including change data capture for incremental migrations.

# 4. Development Tools

# Development Tools

PostgreSQL ships with a powerful command-line client, called psql, which many developers use. The psql client can be used in interactive mode and in batch mode to run more complex scripted operations. It tends to be the preferred choice for experienced PostgreSQL DBAs.




*pgAdmin's ERD diagramming Tool*

The most popular GUI-based development environment for PostgreSQL is pgAdmin. It can be run as a standalone tool, for one developer only, or it can be configured in a server mode, where teams can collaborate on multiple database servers. The pgAdmin environment supports browsing of database objects, configurations, query development, ERD diagramming, and schema diff to identify patches that need to be rolled into production. EDB's Postgres Enterprise Manager® (PEM) builds on pgAdmin and incorporates its functionality.

For developers coming from Oracle, and who want to stay with a familiar tool, Toad Edge is also an option.

There are other development tools that support PostgreSQL too, but pgAdmin is favored by the PostgreSQL development community.

Development best practices and patterns are well-documented in The Art of PostgreSQL book.

# 5. Deployment Tools

# Deployment Tools

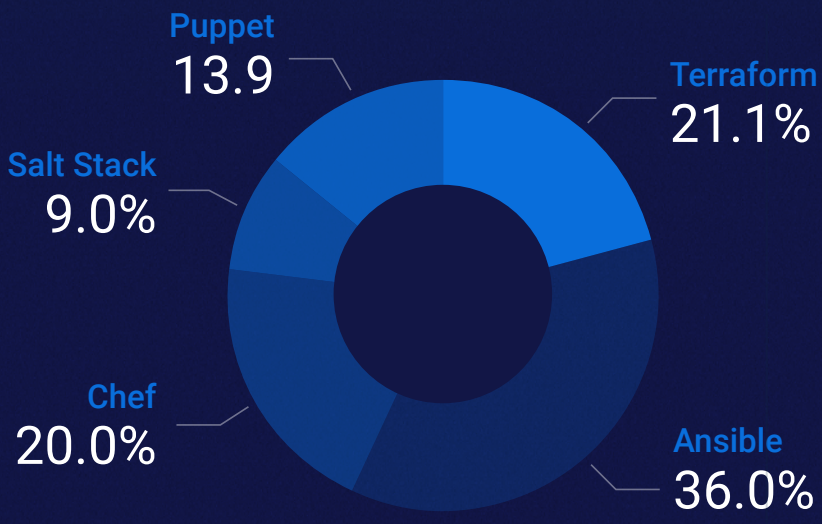
**In the deployment space, there are two major deployment types to focus on:**

- **Deployment of the software**
  That is, the PostgreSQL cluster—and any associated tools
- **Deployment of database changes**
  For example, changes to a table definition

## 5.1 Deployment of the PostgreSQL cluster and associated tools

In virtualized environments, that is on Infrastructure as a Service (IaaS) in public clouds or private clouds such as vSphere, Ansible was the most popular choice among respondents in an EDB survey in June 2020 of infrastructure as code.

### **EDB Survey June 2020:** Most popular deployment tools




*EDB Survey June 2020: Most popular deployment*

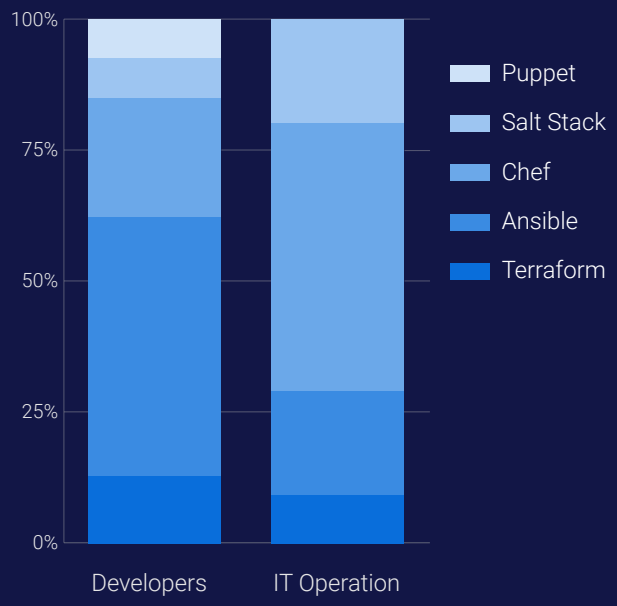
Ansible Galaxy is the preferred place to download roles and plugins, the key building blocks for **Ansible playbook**

Terraform is a popular choice to create infrastructure as code in public clouds, where it excels in provisioning compute, storage, and networking. Terraform is frequently used in conjunction with Ansible: Terraform to provision the infrastructure and Ansible to deploy and configure the software.

## Application developer and IT operations

If we look at the results from the user personas angle, it is quite evident that many developers are using Ansible. However, for IT operations, Puppet and Chef are the most popular tools.

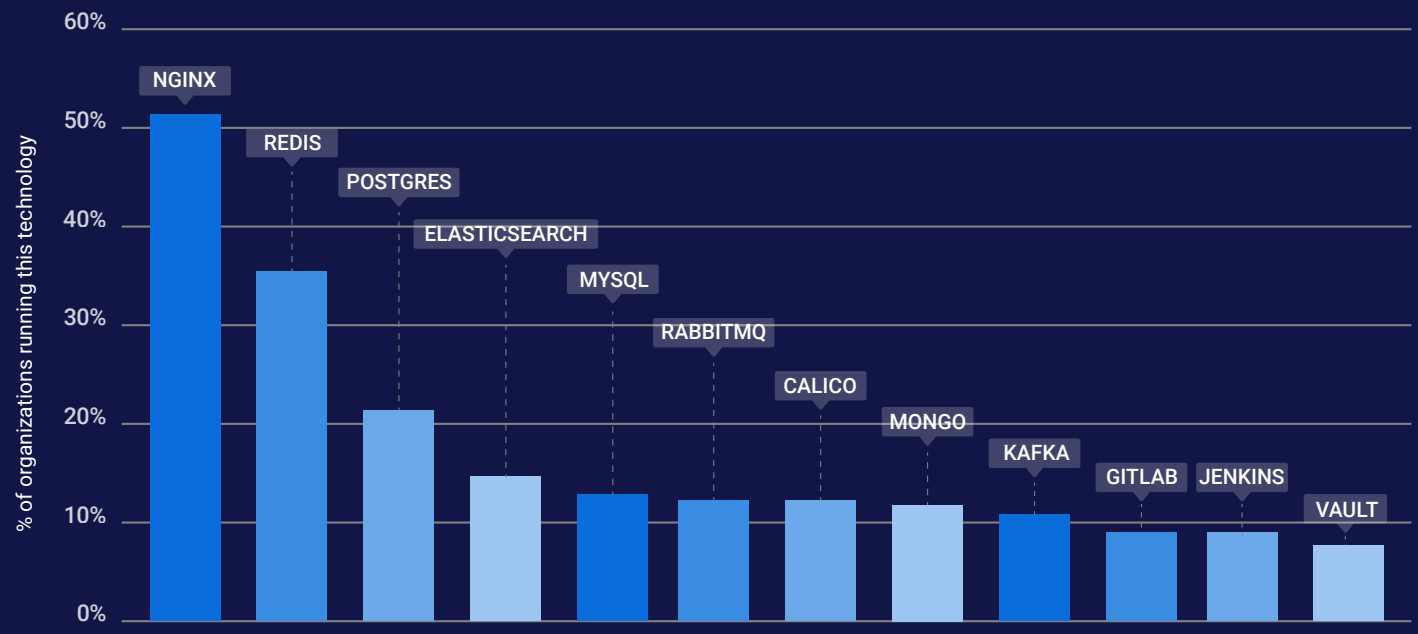
In containerized environments, Kubernetes (K8s) is the clear leader in public cloud and in private cloud orchestrations built on RedHat OpenShift or Suse Rancher.




*EDB Survey June 2020: Who uses which deployment tool*

## Application developer and IT operations

PostgreSQL is extremely popular in containerized environments. PostgreSQL is the most popular persistent database deployed in Docker containers.



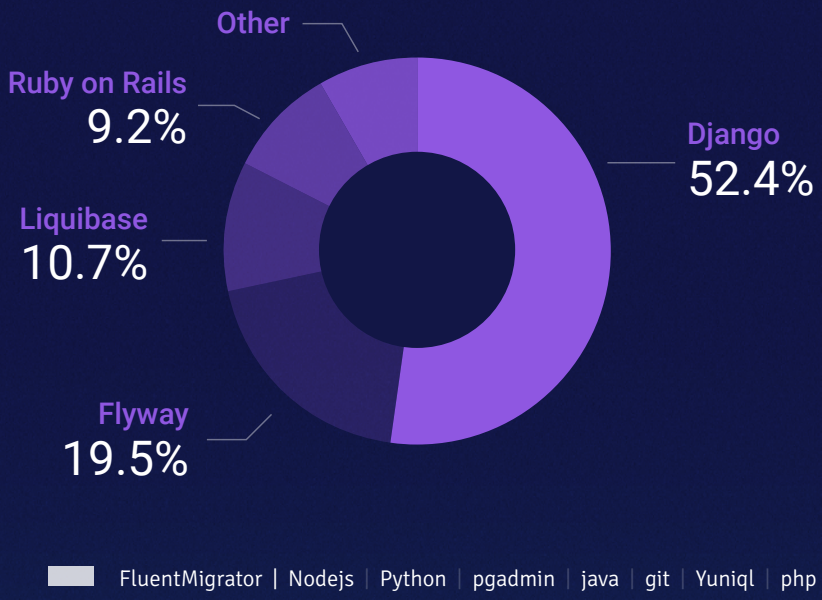

*Source: Datadog, Nov 2020, https://www.datadoghq.com/container-report/*

Stolon and Patroni are two popular first generation high availability mechanisms used in containerized environments. Stolon and Patroni coexist with Kubernetes, but they are not really cloud native; they are an add-on that makes PostgreSQL® work with Kubernetes.

More recently, proper cloud native approaches have emerged that leverage Kubernetes' built-in capabilities for high availability, failover, and other cluster operations. EDB's **Cloud Native PostgreSQL** is a leading example proving that proper leverage of the new Kubernetes capabilities results in simpler and more robust implementations with fewer moving parts.

## 5.2 Deployment of database changes

Flywheel and Liquibase are very popular frameworks for database version control, implementing **Martin Fowler's evolving database model**. Many development frameworks, such as Ruby-on-rails, Django, and others, automatically create "migrations" that can be applied to PostgreSQL too.




*March 2021 EDB Survey (top 12 identified tools)*

# 6. Security Integration

# Security Integration

PostgreSQL security should be seen in the context of the data center or cloud. In both cases, the security of the database needs to be part of a Defense in Depth (DiD) concept. For Cloud Native PostgreSQL based on Kubernetes, the 4C model (Cloud, Cluster, Container, Code) model is the emerging best practice. For PostgreSQL deployments in a data center, the best practice for a DiD is discussed in this whitepaper about **Security Best Practices for PostgreSQL.**

Within PostgreSQL, you need to consider Authentication, Authorization, Auditing, and Encryption.

For **authentication**, PostgreSQL can define users and passwords inside the database. In that case, we recommend using SCRAM authentication to protect the passwords and forgo the less secure MD5 method. If you use locally defined users and passwords, we recommend EDB PostgreSQL password profiles, which allow you to define expiration, complexity, and reuse rules.

**Let's elaborate on the tools that PostgreSQL offers to integrate with existing authentication environments.**

**SSPI** – Use this if you are on a Windows system and would like to implement Single Sign-On (SSO) authentication.

**LDAP** – LDAP is useful in situations where you have large numbers of users (###,###+) and need to manage passwords from a central location. This centralization has the advantage of keeping your pg_hba.conf file small and more manageable and gives your users a unified password experience across your infrastructure. LDAP requires solid infrastructure, as you are relying on the service and connectivity to that service to access your database.

**LDAP** should only be used if Kerberos –which includes both SSPI and GSSAPI – is out of the question. LDAP is less secure because passwords are forwarded to the LDAP server, which might be set up in an insecure way.

**RADIUS** – Radius should not be used because it has weak encryption, using md5 hashing for credentials.

**Cert** – TLS certificate authentication, –sometimes referred to as SSL, can be used for encryption of the traffic on the wire and for authentication. Certificates are often used in machine-to-machine communication.

**Authorization** is handled inside the PostgreSQL database, using roles (i.e., term regrouping users and groups) and attributes, such as LOGIN, SUPERUSER, and so on, that are used when defining privileges.

Separate audit logging is highly valued in environments with elevated compliance requirements, such as financial services. Standard PostgreSQL does not separate the audit log and the system log. EDB's Postgres Advanced Server provides a separate audit log, with different levels of logging for DDL, DML, or SELECT operations by object type.

Authentication, authorization, and auditing best practices for PostgreSQL are extensively discussed in **this blog.**

**Watch a series of short videos about PostgreSQL security features.**

**For encryption of data on disk, PostgreSQL provides three options:**

1. Full disk encryption using Red Hat Linux Unified Key Setup-on-disk-format (LUKS). LUKS bulk encrypts the hard drive partition.
2. File system-level encryption
3. Use of 3rd party tools, such as Thales Vormetric Transparent Encryption

*(1) and (2) are further described **in this blog**.*
*(3) is described further **here** and **here**.*

Authentication, authorization, auditing, and encryptions should always be considered together and used with a least-privilege approach

# 7. High Availability and Disaster Recovery Tools

# High Availability and Disaster Recovery Tools

PostgreSQL can achieve extremely high availability by leveraging its native replication capabilities. Streaming replication in combination with tools such as EDB Failover Manager (EFM), can achieve up to 99.99% availability in a zero data loss mode. EDB Bidirectional Replication (BDR) leverages logical replication and streaming replication and can achieve even higher levels of availability as it supports rolling upgrades for minor and major versions and a two-phase commit option.

In containerized environments, solutions such as Stolon or Patroni were often used in first generation architectures when Kubernetes had not yet achieved its current level of maturity. EDB's Cloud Native Postgres implements a second generation, Kubernetes-native high availability architecture that no longer relies on 3rd party ad-ons.

Prior to the maturing of PostgreSQL's native replication capabilities, hardware-based high availability solutions, such as RedHat Cluster Server with shared disks and fencing devices, were often used. Similarly, open source add-ons, such as Slony Replication or Lonediste, are rapidly disappearing from use and being replaced with PostgreSQL native capabilities.

For disaster recovery, PostgreSQL has built-in backup and recovery capabilities, which are significantly enhanced by tools such as EDB's Backup and Recovery Tool (BART), BARMAN, and pgBackrest.

# 8. Security Integration

# Management Tools for Monitoring, Alerting, and Tuning

Database server monitoring includes the following layers:

### Operating system and hardware

Tracks areas like input, output, memory utilization, network utilization, CPU, physical disk space, and component status.

### PostgreSQL server

Examines DML and DDL statements in logs, wait time, users, objects, whether backups ran successfully, and whether replication is running.

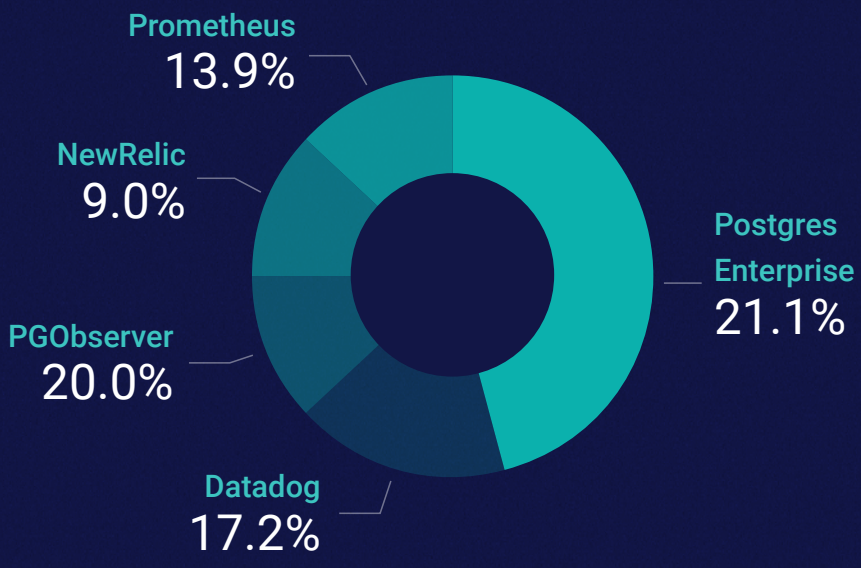
### Query performance

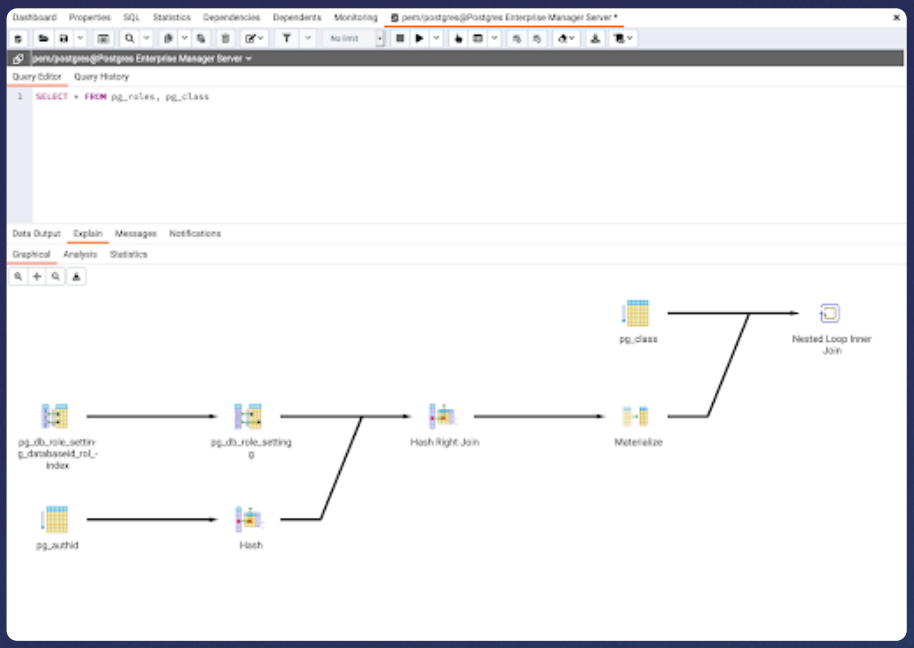
Analyzes individual SQL statements or queries, to examine throughput, latency, concurrency/load, and how errors are being dealt with.

## Monitoring Tool Results

Several general-purpose monitoring tools, like Datadog, Wavefront, SignalFX, and Prometheus with Grafana, are often used to monitor servers running PostgreSQL. Prometheus and Grafana are especially popular in containerized environments.

A **2020 survey** of 3,000 PostgreSQL visitors to the EDB website showed that Datadog, PGObserver, NewRelic and Prometheus are popular choices for users who do not choose EDB Postgres Enterprise Manager.






*EDB PostgreSQL Enterprise Manager – Query Plan Visualizer*

Specialty tools, such as EDB's Postgres Enterprise Manager®, are focused on PostgreSQL and collect specialized data sets and perform PostgreSQL-specific analytics to allow DBAs to get the most value out of PostgreSQL.


*EDB Postgres Enterprise Manager – Performance diagnostics showing different types of wait*

# 9. Infrastructure Integration

# Infrastructure Integration

PostgreSQL runs on a wide range of operating systems and hardware architectures. However, the Linux/X86 platform is by far the most popular. We estimate more than 90% of all production PostgreSQL databases in enterprises run on Linux. RedHat Enterprise Linux has the largest share, followed by Debian and Ubuntu. The percentage of Windows-based production deployments has been decreasing rapidly over the last 10 years.

Container-based deployments are increasingly getting attention, especially since Kubernetes has emerged as the top choice for container orchestration.

On the storage side, PostgreSQL leverages the operating system for disk access; it does not use direct I/O. This means that PostgreSQL greatly benefits from fast, predictable, and reliable storage access. Direct Attached Storage (DAS) and Fibre-Channel-based Storage Area Networks (SAN) are the preferred tools on premises for the data directory and the write ahead log. Backups can be stored on NFS mounts. In the cloud, SSDs with high I/O rates are highly recommended. Click **here** for a detailed discussion of storage options and RAID configurations.

PostgreSQL leverages the operating system for caching and disk access, which means a PostgreSQL tuning strategy must include key operating system settings. For an extensive description of the tuning options on Linux, click **here**, and for Windows, click **here**.

# 10. Summary

# Summary

PostgreSQL is a mature database that integrates well into the enterprise technology stack. Its extreme popularity with developers (see Stackoverflow's 2020 Developer Survey) is a testimony to the PostgreSQL community's strong support for application and database development. PostgreSQL is also well-integrated with corporate data, security, and management environments. It excels on the leading operating systems and hardware platforms and is available in all the major clouds in a SaaS or IaaS mode.

PostgreSQL has emerged as the top database platform that best enables users to get the most out of their tech stacks, get to market faster, and integrate easier into the enterprise environment.

# About EDB

EDB provides enterprise-class software and services that enable businesses and governments to harness the full power of Postgres, the world's leading open source database. With offices worldwide, EDB serves more than 1,500 customers, including leading financial services, government, media and communications and information technology organizations. As one of the leading contributors to the vibrant and fast-growing Postgres community, EDB is committed to driving technology innovation. With deep database expertise, EDB ensures high availability, reliability, security, 24x7 global support and advanced professional services, both on premises and in the cloud. This empowers enterprises to control risk, manage costs and scale efficiently. For more information, visit **www.enterprisedb.com**.

# The Expert's Guide to Integrating Postgres


EnterpriseDB Corporation
34 Crosby Drive
Suite 201
Bedford, MA 01730