

How to Install PostgreSQL 15 on Amazon Linux 2023: A Step-by-Step Guide



Hakan Bayraktar · [Follow](#)

6 min read · Nov 9, 2023

Listen

Share



Introduction

PostgreSQL is a powerful open-source relational database management system widely used for storing and managing data. If you've recently installed Amazon Linux 2023 on your AWS EC2 instance and need guidance on how to install PostgreSQL 15, you're in the right place.

Prerequisites:

Before you start, ensure you meet the following prerequisites:

- An AWS EC2 instance running Amazon Linux 2023 with administrator privileges.
- A minimum of 1GB of available hard disk space, 2GB of RAM, and a single-core CPU.

Step 1- Launching and Configuring Your EC2 Instance

[Open in app](#)

[Sign up](#)

[Sign in](#)

The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar lists navigation options: EC2 Dashboard, EC2 Global View, Events, Instances (with sub-options: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area is divided into several sections: 'Resources' (listing 0 instances, 0 auto scaling groups, 0 dedicated hosts, 0 elastic IPs, 0 instances, 0 key pairs, 0 load balancers, 0 placement groups, 0 security groups, 0 snapshots, 0 volumes), 'Account attributes' (Default VPC: vpc-0b889f3413dee8f3e, Settings: Data protection and security, Zones, EC2 Serial Console, Default credit specification, Console experiments), 'Launch instance' (button: Launch instance), 'Service health' (AWS Health Dashboard, Region: US East (N. Virginia)), and 'Explore AWS' (Save up to 90% on EC2 with Spot Instances, Get Up to 40% Better Price Performance).

You can click to “launch instance” button for create a EC2 instance

2. Configure the instance name and the OS image as follows

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start

[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-05c13eab67c5d8861 (64-bit (x86)) / ami-0840becec4971bb87 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2023 AMI 2023.2.20231030.1 x86_64 HVM kernel-6.1

Architecture

64-bit (x86)

AMI ID

ami-05c13eab67c5d8861

Verified provider

3. Scroll down and configure the instance type and key pair

▼ Instance type [Info](#)**Instance type****t2.medium**

Family: t2 2 vCPU 4 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0464 USD per Hour

On-Demand RHEL base pricing: 0.1064 USD per Hour

On-Demand Windows base pricing: 0.0644 USD per Hour

On-Demand SUSE base pricing: 0.1464 USD per Hour

 All generations[Compare instance types](#)[Additional costs apply for AMIs with pre-installed software](#)**▼ Key pair (login) [Info](#)**

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

acloud

 [Create new key pair](#)

4. Create a security group

▼ Network settings [Info](#)

[Edit](#)

Network [Info](#)

vpc-0b889f3413dee8f3e

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0



Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. **X**

5. Modify to disk size as 20 GB

▼ Storage (volumes) [Info](#) [Simple](#)

EBS Volumes [Hide details](#)

▼ Volume 1 (AMI Root) (Custom)

Storage type Info EBS	Device name - <i>required</i> Info /dev/xvda	Snapshot Info snap-05a6245e68e6545b5
Size (GiB) Info <input type="text" value="20"/>	Volume type Info <input type="text" value="gp3"/>	IOPS Info <input type="text" value="3000"/>
Delete on termination Info <input type="button" value="Yes"/>	Encrypted Info <input type="button" value="Not encrypted"/>	KMS key Info <input type="button" value="Select"/>
KMS keys are only applicable when encryption is set on this volume.		
Throughput Info <input type="text" value="125"/>		

i Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

[Add new volume](#)

File systems [Show details](#)

6. Check the Summary and click Launch instance (right pane) under the Summary section to launch your EC2 instance.

▼ **Summary**

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.2.2...[read more](#)
ami-05c13eab67c5d8861

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 20 GiB

i **Free tier:** In your first year includes X
750 hours of t2.micro (or t3.micro in
the Regions in which t2.micro is
unavailable) instance usage on free
tier AMIs per month, 30 GiB of EBS
storage, 2 million IOs, 1 GB of
snapshots, and 100 GB of bandwidth
to the internet.

[Cancel](#) Launch instance [Review commands](#)

⌚ Success
Successfully initiated launch of Instance (i-09107a02bbdc0d588)

7. Accessing Your EC2 Instance

Server is ready now select the server and click the connect button

Instances (1/2) Info		Connect	Instance state	Actions	Launch Instances		
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/> postgresql	i-09107a02bbdc0d588	Running Details Logs	t2.medium	Initializing Details	No alarms + Details	us-east-1b	ec2-54-152-82-188.co.

EC2 > Instances > i-09107a02bbdc0d588 > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-09107a02bbdc0d588 (postgresql) using any of these options

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 serial console](#)

Instance ID
 i-09107a02bbdc0d588 (postgresql)

Connection Type
 Connect using EC2 Instance Connect
 Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
 Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

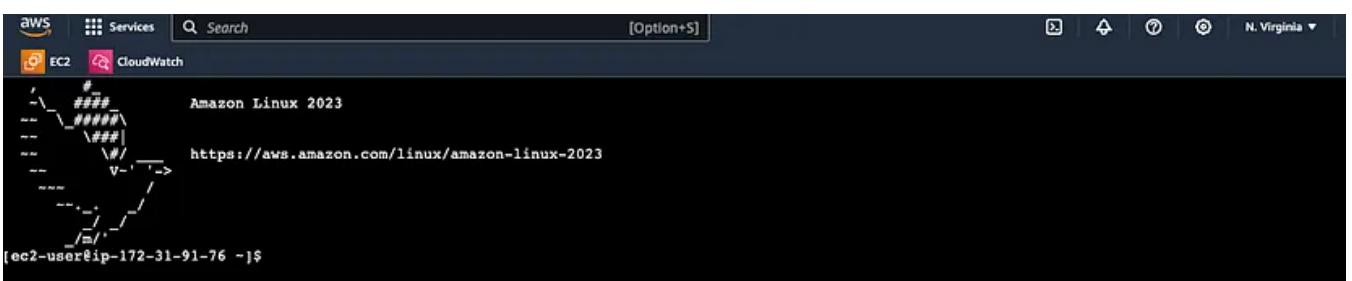
Public IP address
 54.152.82.188

User name
 Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

i Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

[Cancel](#) [Connect](#)

Congratulations! You have successfully set up EC2 Instance Connect



```

aws Services Search [Option+S] N. Virginia
EC2 CloudWatch
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-91-76 ~]$
```

Server is ready we can install the Postgresql . Please follow the below steps.

Step 2: Update Amazon Linux 2023 Packages

Before installing any packages, it's essential to update your system to ensure you have the latest updates and refresh the DNF package cache. Open your terminal or connect to your Amazon Linux instance via SSH and run the following command:

```
sudo dnf update
```

Step 3: Installing PostgreSQL

The good news is that you don't need to add any additional repositories to get PostgreSQL version 15 on your Amazon Linux 2023 because it is available through the default system repository. For a comprehensive list of packages available in the Amazon Linux 2023 repositories, you can refer to [this link](#).

<code>postgresql15</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15</code>	x86_64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-contrib</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-contrib</code>	x86_64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-docs</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-docs</code>	x86_64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-llvmjit</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-llvmjit</code>	x86_64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-plperl</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-plperl</code>	x86_64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-plpython3</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-plpython3</code>	x86_64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-pltcl</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-pltcl</code>	x86_64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-private-devel</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-private-devel</code>	x86_64	<code>postgresql15</code>	15.0-1.amzn2023.0.4
<code>postgresql15-private-libs</code>	aarch64	<code>postgresql15</code>	15.0-1.amzn2023.0.4

Run the following command to install both the client and server components of the PostgreSQL database system:

```
sudo dnf install postgresql15.x86_64 postgresql15-server -y
```

```
[ec2-user@ip-172-31-91-76 ~]$ sudo dnf install postgresql15.x86_64 postgresql15-server -y
Last metadata expiration check: 0:29:55 ago on Thu Nov 9 15:02:25 2023.
Dependencies resolved.

=====
Package           Architecture      Version        Repository    Size
=====
Installing:
  postgresql15          x86_64        15.4-1.amzn2023.0.1   amazonlinux   1.6 M
  postgresql15-server    x86_64        15.4-1.amzn2023.0.1   amazonlinux   6.1 M
Installing dependencies:
  libicu              x86_64        67.1-7.amzn2023.0.3   amazonlinux   9.6 M
  postgresql15-private-libs x86_64        15.4-1.amzn2023.0.1   amazonlinux   144 K

Transaction Summary
=====
Install 4 Packages

Total download size: 17 M
Installed size: 64 M
Downloading Packages:
(1/4): postgresql15-15.4-1.amzn2023.0.1.x86_64.rpm           13 MB/s | 1.6 MB  00:00
(2/4): postgresql15-private-libs-15.4-1.amzn2023.0.1.x86_64.rpm  1.1 MB/s | 144 KB  00:00
(3/4): libicu-67.1-7.amzn2023.0.3.x86_64.rpm                  58 MB/s | 9.6 MB  00:00
(4/4): postgresql15-server-15.4-1.amzn2023.0.1.x86_64.rpm       12 MB/s | 6.1 MB  00:00

Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/1
  Installing : postgresql15-private-libs-15.4-1.amzn2023.0.1.x86_64 1/4
  Installing : postgresql15-15.4-1.amzn2023.0.1.x86_64             2/4
  Installing : libicu-67.1-7.amzn2023.0.3.x86_64                 3/4
  Running scriptlet: postgresql15-server-15.4-1.amzn2023.0.1.x86_64 4/4
  Installing : postgresql15-server-15.4-1.amzn2023.0.1.x86_64             4/4
  Running scriptlet: postgresql15-server-15.4-1.amzn2023.0.1.x86_64 4/4
  Verifying  : postgresql15-private-libs-15.4-1.amzn2023.0.1.x86_64 1/4
  Verifying  : postgresql15-server-15.4-1.amzn2023.0.1.x86_64             2/4
  Verifying  : postgresql15-15.4-1.amzn2023.0.1.x86_64               3/4
  Verifying  : libicu-67.1-7.amzn2023.0.3.x86_64                 4/4

Installed:
  libicu-67.1-7.amzn2023.0.3.x86_64                         postgresql15-15.4-1.amzn2023.0.1.x86_64
  postgresql15-server-15.4-1.amzn2023.0.1.x86_64               postgresql15-private-libs-15.4-1.amzn2023.0.1.x86_64

Complete!
```

Step 4: Initializing PostgreSQL Database

Before starting and enabling the database service, let's initialize it. Use the `initdb` command, which will create a new PostgreSQL database cluster referring to a collection of databases managed by a single server instance:

```
sudo postgresql-setup --initdb
```

```
[ec2-user@ip-172-31-91-76 ~]$ sudo postgresql-setup --initdb
 * Initializing database in '/var/lib/pgsql/data'
 * Initialized, logs are in /var/lib/pgsql/initdb_postgresql.log
[ec2-user@ip-172-31-91-76 ~]$ █
```

Step 5: Starting and Enabling PostgreSQL Service

After completing the initialization, start and enable the PostgreSQL server service, so it starts automatically with system boot:

```
sudo systemctl start postgresql
sudo systemctl enable postgresql
sudo systemctl status postgresql
```

```
ec2-user@ip-172-31-91-76 ~]$ sudo systemctl status postgresql
postgresql.service - PostgreSQL database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: disabled)
     Active: active (running) since Thu 2023-11-09 15:33:41 UTC; 38s ago
       Main PID: 27808 (postgres)
         Tasks: 7 (limit: 4660)
        Memory: 17.0M
          CPU: 54ms
        CGroup: /system.slice/postgresql.service
            └─27808 /usr/bin/postgres -D /var/lib/pgsql/data

ov 09 15:33:41 ip-172-31-91-76.ec2.internal systemd[1]: Starting postgresql.service - PostgreSQL database server...
ov 09 15:33:41 ip-172-31-91-76.ec2.internal postgres[27808]: 2023-11-09 15:33:41.217 UTC [27808] LOG: redirecting log output to logging collector process
ov 09 15:33:41 ip-172-31-91-76.ec2.internal postgres[27808]: 2023-11-09 15:33:41.217 UTC [27808] HINT: Future log output will appear in directory "log".
ov 09 15:33:41 ip-172-31-91-76.ec2.internal systemd[1]: Started postgresql.service - PostgreSQL database server.
ec2-user@ip-172-31-91-76 ~]$
```

Step 6: Configure PostgreSQL

1. Set password for ssh postgres user and admin postgres database password

For security, set a strong password for the system user and default database admin user account. Use the following commands:

```
# Change the ssh user password:
sudo passwd postgres

# Log in using the Postgres system account:
su - postgres

# Now, change the admin database password:
psql -c "ALTER USER postgres WITH PASSWORD 'your-password';"
exit
```

```
[ec2-user@ip-172-31-91-76 ~]$ sudo passwd postgres
Changing password for user postgres.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-91-76 ~]$ su - postgres
Password:
[postgres@ip-172-31-91-76 ~]$ psql -c "ALTER USER postgres WITH PASSWORD 'Passw0rd123';"
ALTER ROLE
[postgres@ip-172-31-91-76 ~]$ exit
logout
[ec2-user@ip-172-31-91-76 ~]$
```

2. Primary Configuration File

Access the primary configuration file of PostgreSQL located at `/var/lib/pgsql/data/postgresql.conf`. Before making any changes, back up the configuration file:

```
sudo cp /var/lib/pgsql/data/postgresql.conf /var/lib/pgsql/data/postgresql.conf
```

```
ec2-user@ip-172-31-91-76 ~]$ sudo cp /var/lib/pgsql/data/postgresql.conf /var/lib/pgsql/data/postgresql.conf.bck  
ec2-user@ip-172-31-91-76 ~]$ █
```

Edit this file with a text editor:

```
sudo vi /var/lib/pgsql/data/postgresql.conf
```

By default, PostgreSQL only listens to localhost

```
listen_addresses = 'localhost'
```

if you want to listen all IP addresses:

```
listen_addresses = '*' # what IP address(es) to listen on;
```

```

#           GB = gigabytes          min = minutes
#           TB = terabytes         h   = hours
#                                     d   = days

#-----#
# FILE LOCATIONS
#-----#

# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

#data_directory = 'ConfigDir'          # use data in another directory
#                                     # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'    # host-based authentication file
#                                     # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf' # ident configuration file
#                                     # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = ''               # write an extra PID file
#                                     # (change requires restart)

#-----#
# CONNECTIONS AND AUTHENTICATION
#-----#

# - Connection Settings -

listen_addresses = '*'             # what IP address(es) to listen on;
#                                     # comma-separated list of addresses;
#                                     # defaults to 'localhost'; use '*' for all
#                                     # (change requires restart)
#port = 5432                         # (change requires restart)
max_connections = 100              # (change requires restart)
#superuser_reserved_connections = 3  # (change requires restart)
#unix_socket_directories = '/var/run/postgresql, /tmp' # comma-separated list of directories
#                                     # (change requires restart)

```

3. Authentication

For authentication, there is a separate file called `pg_hba.conf` in the same directory as the primary configuration file.

Before making any changes, back up the configuration file:

```
sudo cp /var/lib/pgsql/data/pg_hba.conf /var/lib/pgsql/data/pg_hba.conf.bck
```

```
[ec2-user@ip-172-31-91-76 ~]$ sudo cp /var/lib/pgsql/data/pg_hba.conf /var/lib/pgsql/data/pg_hba.conf.bck
[ec2-user@ip-172-31-91-76 ~]$
```

Edit this file with a text editor:

```
sudo vi /var/lib/pgsql/data/pg_hba.conf
```

```
# You can change ident as md5 To allow connections from absolutely any address
host      all      all      0.0.0.0/0      md5
```

OR

```
sudo sed -i 's/ident$/md5/' /var/lib/pgsql/data/pg_hba.conf
```

To apply all the changes, restart the PostgreSQL service using the following command.

```
sudo systemctl restart postgresql
```

```
[ec2-user@ip-172-31-91-76 ~]$ sudo sed -i 's/ident$/md5/' /var/lib/pgsql/data/pg_hba.conf
[ec2-user@ip-172-31-91-76 ~]$ sudo systemctl restart postgresql
[ec2-user@ip-172-31-91-76 ~]$ █
```

4. How to Create a User & Database

Use this section to create a new user and database on PostgreSQL:

```
# Connect to the PostgreSQL server as the Postgres user:
sudo -i -u postgres psql

# Create a new database user:
CREATE USER yourusername WITH PASSWORD 'password';

# Create a new database:
CREATE DATABASE database_name;

# Grant all privileges on the database to the user:
GRANT ALL PRIVILEGES ON DATABASE database_name TO yourusername;

# To list all available PostgreSQL users and databases:
\l
```

```
[ec2-user@ip-172-31-91-76 ~]$ sudo -i -u postgres psql
psql (15.4)
Type "help" for help.

postgres=# CREATE USER testuser WITH PASSWORD 'Passw0rd123';
CREATE ROLE
postgres=# CREATE DATABASE testdatabase;
CREATE DATABASE
postgres=# GRANT ALL PRIVILEGES ON DATABASE testdatabase TO testuser;
GRANT
postgres=# \l
      List of databases
   Name    | Owner     | Encoding | Collate | Ctype | ICU Locale | Locale Provider | Access privileges
   postgres | postgres  | UTF8     | C.UTF-8 | C.UTF-8|             | libc                |
template0 | postgres  | UTF8     | C.UTF-8 | C.UTF-8|             | libc                |
template1 | postgres  | UTF8     | C.UTF-8 | C.UTF-8|             | libc                |
testdatabase | postgres | UTF8     | C.UTF-8 | C.UTF-8|             | libc                |
(4 rows)

postgres=#
```

Step 7: Accessing the Database

You can access the database you created using the PostgreSQL client command `psql`. Local or SSH server connection users can use the following syntax:

```
psql -h localhost -U username -d database_name
```

```
[ec2-user@ip-172-31-91-76 ~]$ psql -h localhost -U testuser -d testdatabase
Password for user testuser:
psql (15.4)
Type "help" for help.

testdatabase=> []
```

Remote users can use:

```
psql -h server-ip-address -U username -d database_name
```

Replace `username` with the user you created and `database_name` with the name of the database assigned to that user.

Access the `postgres` account on your server by typing:

```
sudo -i -u postgres
```

Now you can immediately access Postgres prompt by typing:

```
$ psql
postgres.

# To list the databases:
postgres=# \l

# You can exit Postgres prompt by typing:
postgres=# \q
```

```
[ec2-user@ip-172-31-91-76 ~]$ sudo -i -u postgres
[postgres@ip-172-31-91-76 ~]$ psql
psql (15.4)
Type "help" for help.

postgres=# \l
              List of databases
   Name    | Owner     | Encoding | Collate | Ctype | ICU Locale | Locale Provider | Access privileges
----+-----+-----+-----+-----+-----+-----+-----+
postgres  | postgres  | UTF8    | C.UTF-8 | C.UTF-8 |             | libc            |
template0 | postgres  | UTF8    | C.UTF-8 | C.UTF-8 |             | libc            |
template1 | postgres  | UTF8    | C.UTF-8 | C.UTF-8 |             | libc            |
testdatabase | postgres | UTF8    | C.UTF-8 | C.UTF-8 |             | libc            |
(4 rows)

postgres=# \q
[postgres@ip-172-31-91-76 ~]$ █
```

To access your PostgreSQL database from external sources, make sure to configure your Amazon Linux EC2 security group to allow incoming traffic on port 5432, which is the default port used by PostgreSQL

Important PostgreSQL Server Configurations on Amazon Linux

Configuration Item	Value
PostgreSQL default port	5432
Default user	postgres
Config files location (postgresql.conf)	/var/lib/pgsql/data
Config files location (pg_hba.conf)	/var/lib/pgsql/data
Default database	postgres
Default data directory	/var/lib/pgsql/data

How to Uninstall PostgreSQL 15 from Amazon Linux 2023

NOTE: Backup Your Data (Important)

Before proceeding, it's crucial to back up any data you want to keep, such as databases and configuration files. Uninstalling PostgreSQL will remove these files, and data loss is irreversible.

Step 1: Stop PostgreSQL Service

Before uninstalling PostgreSQL, ensure that the service is stopped:

```
sudo systemctl stop postgresql
```

Step 2: Disable PostgreSQL Service

Prevent PostgreSQL from starting at system boot:

```
sudo systemctl disable postgresql
```

Step 3: Remove PostgreSQL Packages

Use the following commands to remove PostgreSQL 15 packages. These are the same packages you installed during the installation process:

```
sudo dnf remove postgresql15.x86_64 postgresql15-server
```

Step 4: Completely Remove PostgreSQL

After removing the packages, delete all PostgreSQL-related configuration files and data using the following command:

```
sudo rm -rf /var/lib/pgsql /var/log/postgresql /etc/postgresql
```

```

ec2-user@ip-172-31-91-76:~$ sudo dnf remove postgresql15.x86_64 postgresql15-server
Dependencies resolved.

=====
Package           Architecture   Version            Repository      Size
=====
Removing:
postgresql15      x86_64        15.4-1.amzn2023.0.1    @amazonlinux   6.5 M
postgresql15-server x86_64        15.4-1.amzn2023.0.1    @amazonlinux   25 M
Removing unused dependencies:
libicu             x86_64        67.1-7.amzn2023.0.3  @amazonlinux   33 M
postgresql15-private-libs x86_64    15.4-1.amzn2023.0.1  @amazonlinux   340 k

Transaction Summary

Remove 4 Packages

Free disk space: 64 M
Is this ok [y/N]: y
Running transaction check
  transaction check succeeded.
Running transaction test
  transaction test succeeded.
Running transaction
  Preparing :                                                 1/1
  Running scriptlet: postgresql15-server-15.4-1.amzn2023.0.1.x86_64 1/4
  Erasing   : postgresql15-server-15.4-1.amzn2023.0.1.x86_64          1/4
  Running scriptlet: postgresql15-server-15.4-1.amzn2023.0.1.x86_64 1/4
  Erasing   : postgresql15-15.4-1.amzn2023.0.1.x86_64                1/4
  Erasing   : postgresql15-private-libs-15.4-1.amzn2023.0.1.x86_64     2/4
  Erasing   : libicu-67.1-7.amzn2023.0.3.x86_64                      3/4
  Running scriptlet: libicu-67.1-7.amzn2023.0.3.x86_64                4/4
  Verifying  : libicu-67.1-7.amzn2023.0.3.x86_64                      4/4
  Verifying  : postgresql15-15.4-1.amzn2023.0.1.x86_64                1/4
  Verifying  : postgresql15-private-libs-15.4-1.amzn2023.0.1.x86_64     2/4
  Verifying  : postgresql15-server-15.4-1.amzn2023.0.1.x86_64          3/4
  Verifying  : postgresql15-server-15.4-1.amzn2023.0.1.x86_64          4/4

Removed:
libicu-67.1-7.amzn2023.0.3.x86_64          postgresql15-15.4-1.amzn2023.0.1.x86_64          postgresql15-private-libs-15.4-1.amzn2023.0.1.x86_64

Complete!
ec2-user@ip-172-31-91-76:~$ sudo rm -rf /var/lib/pgsql /var/log/postgresql /etc/postgresql
ec2-user@ip-172-31-91-76:~$ 

```

Conclusion

With these steps, you have successfully uninstalled PostgreSQL 15 from your Amazon Linux 2023 instance. Make sure you have backed up any critical data before proceeding with the uninstallation process.



[Follow](#)

Written by Hakan Bayraktar

265 Followers

More from Hakan Bayraktar

```
join 146.190.135.86:6443 --token f1h95l.u4nkex9cw8d0g63w      --disc
3d1666af50c85f060b9fadcd73f13c932e0e2a9eeef08f51f91a
e-flight checks
nfiguration from the cluster...
an look at this config file with 'kubectl -n kube-system get cm kubeadm-c
g kubelet configuration to file "/var/lib/kubelet/config.yaml"
g kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-1
ng the kubelet
g for the kubelet to perform the TLS Bootstrap...

ne cluster:
request was sent to apiserver and a response was received.
rmed of the new secure connection details.

' on the control-plane to see this node join the cluster.
```



Hakan Bayraktar

How to Install Kubernetes Cluster on Ubuntu 22.04 (Step-by-Step Guide)

Introduction

6 min read · Nov 3, 2023



269



13

```
emctl restart nfs-kernel-server
nfs-kernel-server
FS server and services
● /systemd/system/nfs-server.service; enabled
(loaded) since Fri 2023-11-03 16:53:28 UTC; 4
  StartPre=/usr/sbin/exportfs -r (code=exit
  Start=/usr/sbin/rpc.nfsd (code=exited, st
e=exited, status=0/SUCCESS)
```

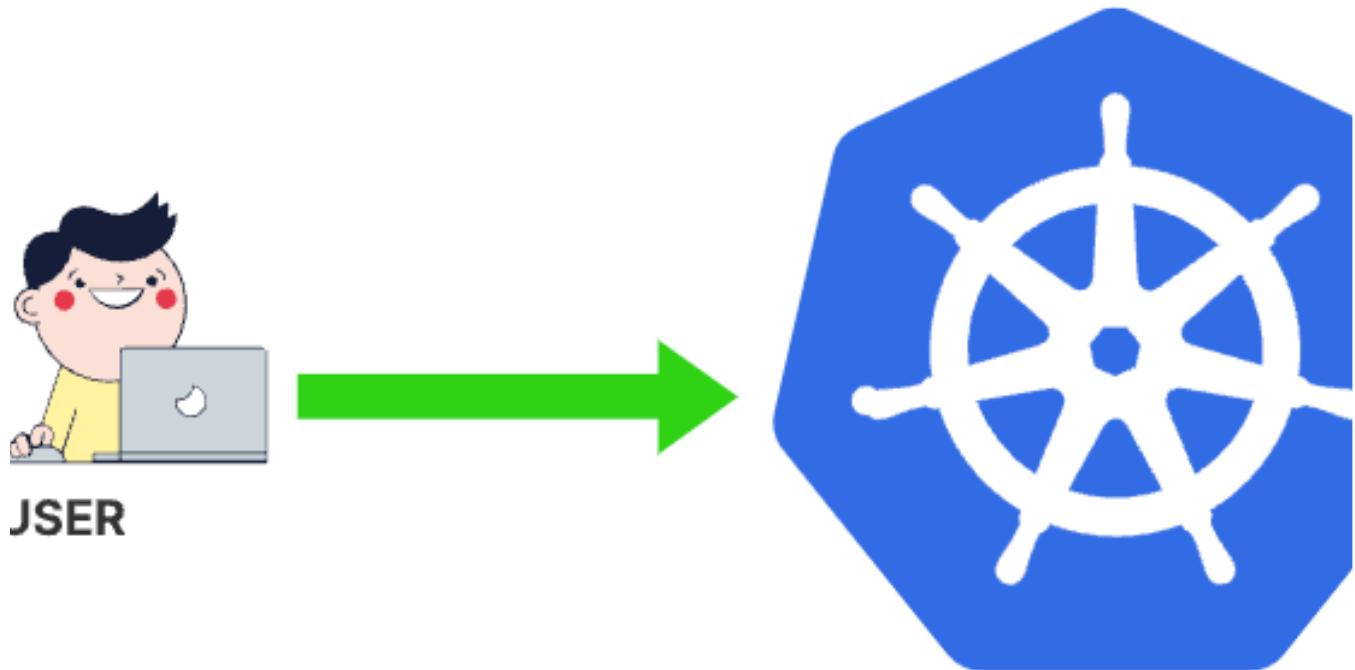
 Hakan Bayraktar

How to Setup Dynamic NFS Provisioning in a Kubernetes Cluster

Dynamic NFS storage provisioning in Kubernetes streamlines the creation and management of NFS volumes for your Kubernetes applications. It...

4 min read · Nov 3, 2023

 60 

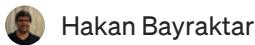


 Hakan Bayraktar

How to Create a User in a Kubernetes Cluster and Grant Access

In this detailed guide, we'll illustrate the steps required to create a user, generate necessary certificates, and configure access using a...

4 min read · Nov 18, 2023



Hakan Bayraktar

Merging Multiple kubeconfig Files into One: A Comprehensive Guide

Introduction

2 min read · Nov 7, 2023



See all from Hakan Bayraktar

Recommended from Medium



PostgreSQL



Andrey Byhalenko in DevOps Manuals and Technical Notes

How to Install and Configure PostgreSQL on Ubuntu 20.04

In this tutorial, I will show how to install PostgreSQL from the apt repository and go over the basic configurations.

5 min read · Jan 12, 2024



Sheraz Iqbal

Adding Free SSL from Let's Encrypt To Your Domain with Nginx on Ubuntu EC2

Securing Your Website Access with HTTPS

3 min read · Feb 2, 2024

54

1



Lists



Natural Language Processing

1489 stories · 1001 saves



Cumhur Akkaya

MinIO-1: Installing And Using MinIO As An Object Storage

It functions almost the same as the S3 tool in Amazon Web Services. However, MinIO is a 100% open-source object storage solution, while AWS...

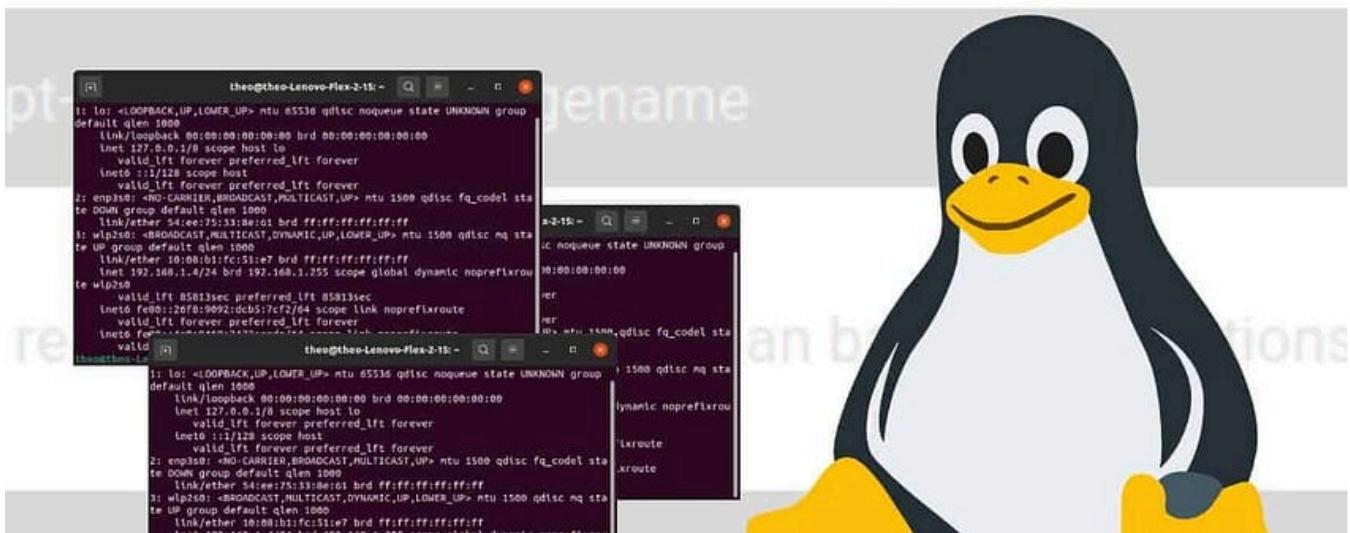
12 min read · Feb 21, 2024

25

1



Cheat Sheet



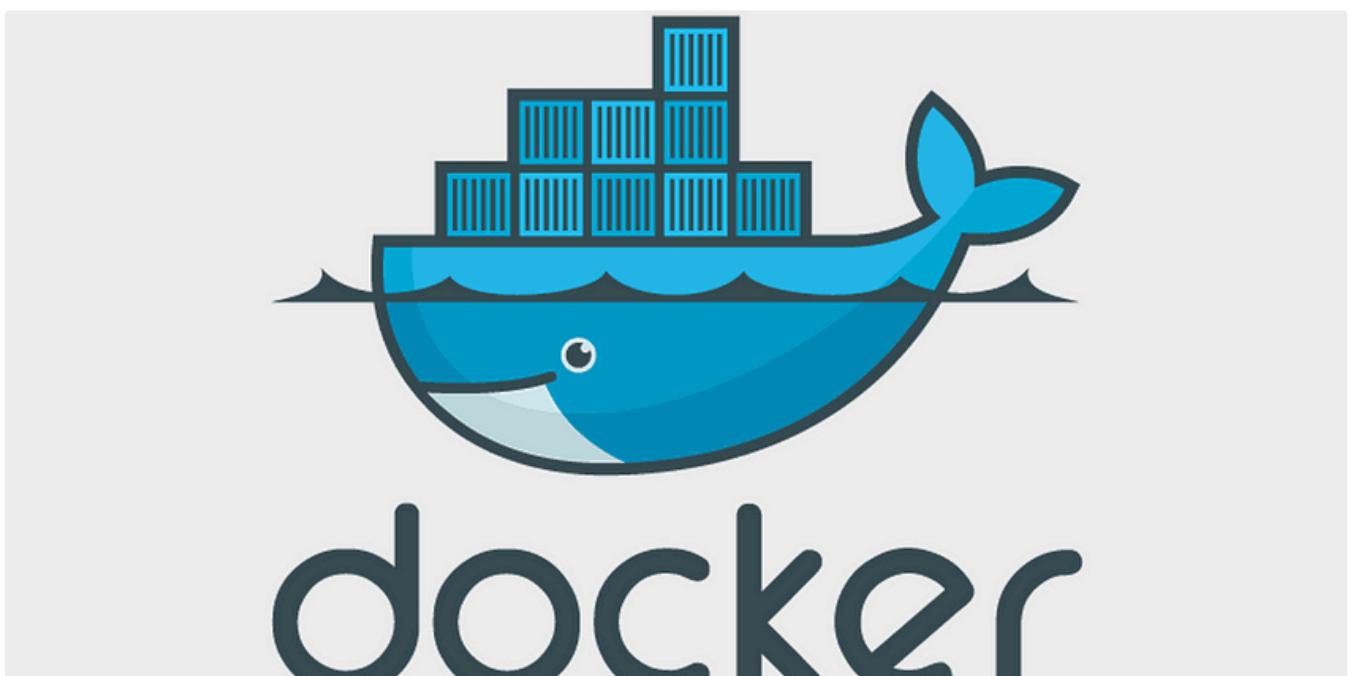
 Vinodha kumara

CheatSheet: Linux Commands for DevOps

Linux command line for effective server management, automation...

5 min read · Jan 20, 2024

 405  5



 Srija Anaparthys

Concept of Docker in detail

Concept of Docker in detail

15 min read · Feb 12, 2024

 61 1

Interview Questions & Answers

 Faisal Kuzhan

DAY_25/90 => INTERVIEW QUESTIONS ON AWS

Below are some basic AWS interview questions along with the answers. 

17 min read · Feb 12, 2024

 91 3[See more recommendations](#)