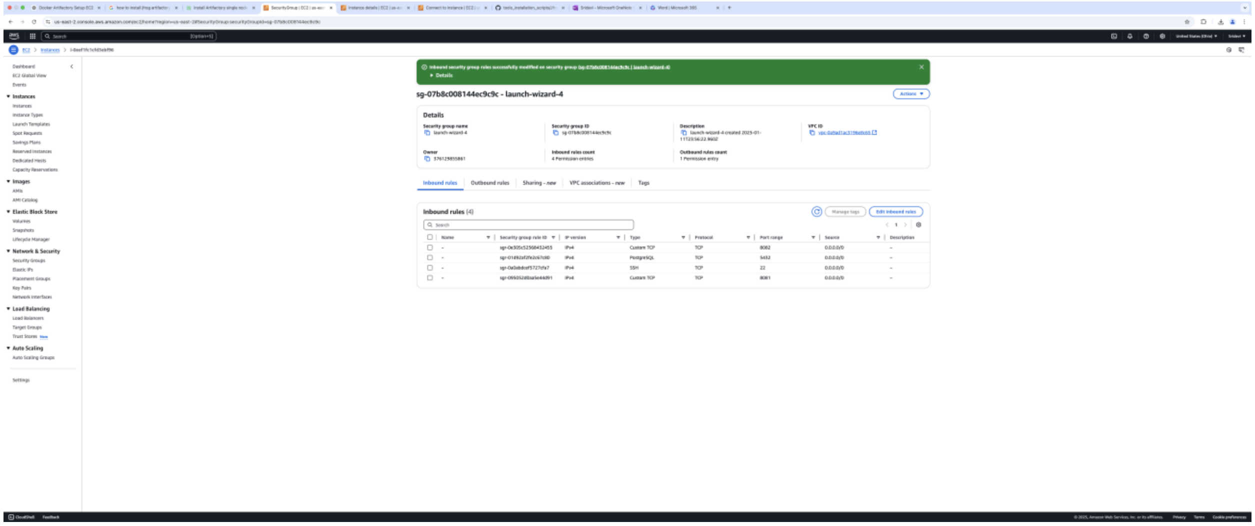
Install JFROG Artifactory using Docker

Login to your EC2 instance.
From AWS console enable inbound port rules for 8081, 8082 for Jfrog artifactory and 5432 for postgresql respectively.



Inbound security group rules successfully modified on security group (sg-07b8c008144ec9c9c | launch-wizard-4)

sg-07b8c008144ec9c9c - launch-wizard-4

Details

Security group name: launch-wizard-4
Security group ID: sg-07b8c008144ec9c9c
Description: launch-wizard-4 created 2025-01-11T23:56:22.960Z
Owner: 376129855861
Inbound rules count: 4 Permission entries
Outbound rules count: 1 Permission entry

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0e505c52568452455	IPv4	Custom TCP	TCP	8082	0.0.0.0/0	-
-	sgr-01d92af2fe2c67c80	IPv4	PostgreSQL	TCP	5432	0.0.0.0/0	-
-	sgr-0a0abdcaf5727cfa7	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-099052d0aa5e44d91	IPv4	Custom TCP	TCP	8081	0.0.0.0/0	-

```
sridevisandeeppochu@Sridevis-MacBook-Pro Downloads % ssh -i "jfrog.pem" ubuntu@ec2-18-191-14-99.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-191-14-99.us-east-2.compute.amazonaws.com (18.191.14.99)' can't be established.
ED25519 key fingerprint is SHA256:RtDQgP0L5QcsgUwBZ71Qbs93x8ph7sSZi4OGZ7tPrv0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-191-14-99.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)
ubuntu@ip-172-31-13-46:~$
```

Then switch to the root user and execute the following commands to install Docker & Postgresql & Jfrog artifactory:
Update the package list and upgrade the system.

```
ubuntu@ip-172-31-13-46:~$ sudo su -
root@ip-172-31-13-46:~# sudo apt update && sudo apt upgrade -y
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 P
```

Install the required dependencies:

```
root@ip-172-31-13-46:~# sudo apt install apt-transport-https ca-certificates curl software-properties-common -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Add Docker’s official GPG key:

```
root@ip-172-31-13-46:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

Add the Docker repository:

```
root@ip-172-31-13-46:~# sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable" -y
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable'
Description:
Archive for codename: bionic components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
```

Install Docker:

```
sudo apt update -y
sudo apt-cache policy docker-ce
sudo apt install docker-ce –y
```

```
root@ip-172-31-13-46:~# sudo apt update -y
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
root@ip-172-31-13-46:~# sudo apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:24.0.2-1~ubuntu.18.04~bionic
  Version table:
     5:24.0.2-1~ubuntu.18.04~bionic 500
root@ip-172-31-13-46:~# sudo apt install docker-ce -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
root@ip-172-31-13-46:~# chmod 777 /var/run/docker.sock
```

Enable and start the Docker service:

```
sudo systemctl enable docker
sudo systemctl start docker
```

```
root@ip-172-31-13-46:~# sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
root@ip-172-31-13-46:~# sudo systemctl start docker
root@ip-172-31-13-46:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-01-12 00:00:33 UTC; 2s ago
```

```
TriggeredBy: ● docker.socket
```

Once after Docker installed, refer the JFROG official documentation page ref: https://jfrog.com/help/r/jfrog-installation-setup-documentation/install-artifactory-single-node-with-docker

Add your user to the docker group:

```
root@ip-172-31-13-46:~# sudo usermod -aG docker $USER
```

Create your Artifactory home directory and an empty system.yaml file. The user creating the folder should be the user running the docker run.

```
root@ip-172-31-13-46:~# mkdir -p /opt/jfrog/artifactory/var/etc/
root@ip-172-31-13-46:~# cd /opt/jfrog/artifactory/var/etc/
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# touch ./system.yaml
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# vi system.yaml
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# cat system.yaml
shared:
  database:
    driver: org.postgresql.Driver
    type: postgresql
    # add your ec2 public ip
    url: jdbc:postgresql://18.191.14.99:5432/artifactorydb
    username: artifactory
    password: password

root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# chown -R 1030:1030 /opt/jfrog/artifactory/var
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# chmod -R 777 /opt/jfrog/artifactory/var
```

Check master.key if it is missing create new one as follows:

The file is typically located in the $JFROG_HOME/artifactory/var/etc/security/ directory.
Since you're running Artifactory in a Docker container, you need to check the persistent volume mapped to:
If the file is missing, Artifactory cannot start properly.

```
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# mkdir security
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# cd security/
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# touch master.key
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# openssl rand -hex 32 > /opt/jfrog/artifactory/var/etc/security/master.key
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# ls -l
total 4
-rw-r--r-- 1 root root 65 Jan 12 00:02 master.key

root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# chown 1030:1030 /opt/jfrog/artifactory/var/etc/security/master.key
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# chmod 600 /opt/jfrog/artifactory/var/etc/security/master.key
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# ls -l
total 4
-rwxrwxrwx 1 1030 1030 65 Jan 12 00:02 master.key
```

Start an PostgreSQL container on the same machine as Artifactory Container:

```
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# docker run --name postgres -itd -e
POSTGRES_USER=artifactory -e POSTGRES_PASSWORD=password -e POSTGRES_DB=artifactorydb -p 5432:5432
library/postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
fd674058ff8f: Pull complete
1eab12a50bdf: Pull complete
5a81b4aedb94: Pull complete
```

```
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                       NAMES
1e3565be6b96   postgres   "docker-entrypoint.s…"   5 seconds ago   Up 3 seconds   0.0.0.0:5432->5432/tcp, :::5432-
>5432/tcp   postgres
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# telnet 18.191.14.99 5432
Trying 18.191.14.99...
Connected to 18.191.14.99.
```

Once Postgresql installed run the JFROG Artifactory:

```
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# docker run --name artifactory -v
/opt/jfrog/artifactory/var/:/var/opt/jfrog/artifactory -d -p 8081:8081 -p 8082:8082 releases-
docker.jfrog.io/jfrog/artifactory-pro:7.90.7
Unable to find image 'releases-docker.jfrog.io/jfrog/artifactory-pro:7.90.7' locally
7.90.7: Pulling from jfrog/artifactory-pro
969cd2eb857f: Pull complete
799f114b0db9: Pull complete
fbb77d0f94ec: Pull complete
ad3bf0aa5078: Pull complete
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security# docker ps -a
CONTAINER ID   IMAGE                                          COMMAND                  CREATED         STATUS         PORTS
NAMES
da7c76e4d92a   releases-docker.jfrog.io/jfrog/artifactory-pro:7.90.7   "/entrypoint-artifac…"   4 seconds ago   Up 4
seconds   0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp   artifactory
1e3565be6b96   postgres                                       "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:5432-
>5432/tcp, :::5432->5432/tcp                    postgres
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc/security#
```

Check the Artifactory log:

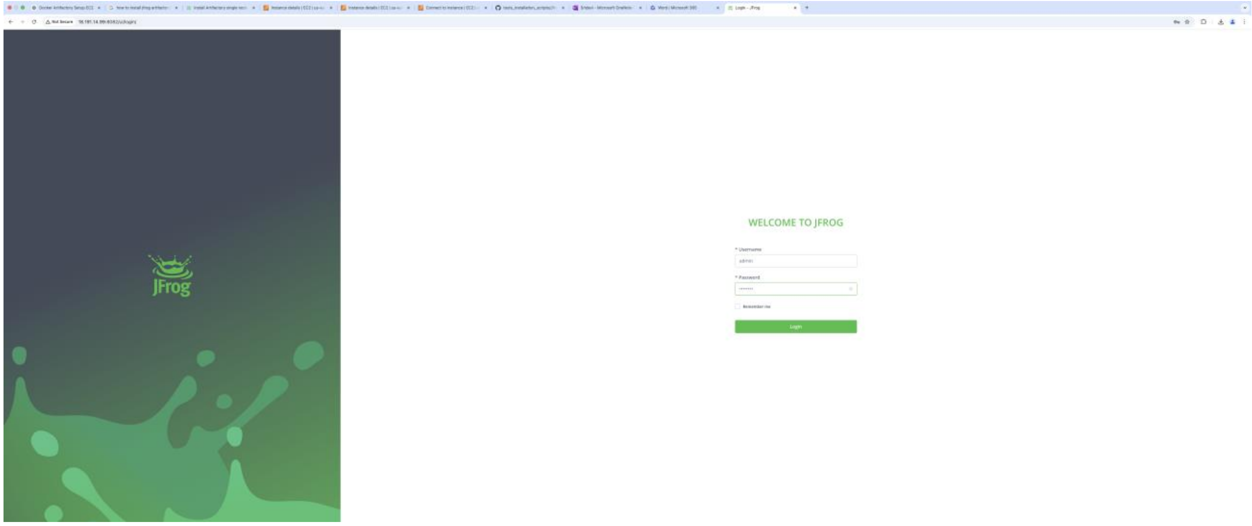
```
root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc# docker logs -f artifactory
Preparing to run Artifactory in Docker
Running as uid=1030(artifactory) gid=1030(artifactory) groups=1030(artifactory)
Dockerfile for this image can found inside the container.
To view the Dockerfile: 'cat /docker/artifactory-pro/Dockerfile.artifactory'.
Resolved .shared.database.type (postgresql) from /opt/jfrog/artifactory/var/etc/system.yaml
Resolved .shared.database.url (__sensitive_key_hidden___) from /opt/jfrog/artifactory/var/etc/system.yaml
Waiting for DB postgresql to be ready on host.docker.internal/5432 for 30 seconds


2025-01-12T00:17:13.397Z [jfrou] [INFO ] [09e7d88f86fd4084] [local_topology.go:385       ] [main             ] [] -
###############################################################
###   All services started successfully in 80.089 seconds   ###
###############################################################
```

```
2025-01-12T00:17:13.443Z [jfac ] [INFO ] [09e7d88f86fd4084] [CertificateFileHandlerBase:167] [c-default-executor-3] -
Loading ca certificate from database.
2025-01-12T00:17:13.618Z [jfac ] [INFO ] [09e7d88f86fd4084] [CertificateFileHandlerBase:200] [c-default-executor-3] -
[ACCESS BOOTSTRAP] Saved new ca certificate at: /opt/jfrog/artifactory/var/etc/access/keys/ca.crt
2025-01-12T00:17:13.619Z [jfac ] [INFO ] [09e7d88f86fd4084] [CertificateFileHandlerBase:167] [c-default-executor-3] -
Loading root certificate from database.
2025-01-12T00:17:13.728Z [jfac ] [INFO ] [09e7d88f86fd4084] [CertificateFileHandlerBase:200] [c-default-executor-3] -
[ACCESS BOOTSTRAP] Saved new root certificate at: /opt/jfrog/artifactory/var/etc/access/keys/root.crt
2025-01-12T00:17:15.086Z [jfrt ] [INFO ] [                ] [o.j.c.ConfigWrapperImpl:336  ] [e-watcher-notifier-1] - [Node ID:
3e4016da0674]  detected local  modify for config 'artifactory/config/security/access/access.admin.token'
2025-01-12T00:17:15.412Z [jfrou] [INFO ] [318eeedf4680db03] [external_topology.go:272     ] [main               ] [] -
External Topology changed: 3e4016da0674 state changed (jfac@01jhbxxa5qhdww1exwn4d90d4f
HEALTHY,jfcon@01jhbxxa5qhdww1exwn4d90d4f HEALTHY,jfevt@01jhbxxa5qhdww1exwn4d90d4f
HEALTHY,jffe@01jhbxxa5qhdww1exwn4d90d4f HEALTHY,jfmd@01jhbxx6yw23x1qhwws7j5dvxw
HEALTHY,jfob@01jhbxxn2zebnf1t0zyy7n149z HEALTHY,jfrt@01jhbxxn2zebnf1t0zyy7n149z HEALTHY)
2025-01-12T00:17:15.416Z [jfrou] [INFO ] [318eeedf4680db03] [routing_handler.go:410       ] [main               ] [] -
External topology is changed, updating traefik external routes
2025-01-12T00:17:23.152Z [jfrt ] [INFO ] [                ] [o.j.c.ConfigWrapperImpl:336  ] [e-watch
2025-01-12T00:17:39.567Z [jfrt ] [INFO ] [f5ff621a0e4d6adb] [onStatusStorageServiceImpl:124] [onitor-migration-job] -
Migration for 'event-table-repo-key-migration' has finished.

2025-01-12T00:17:39.570Z [jfrt ] [INFO ] [f5ff621a0e4d6adb] [ntTableRepoKeyMigrationDao:185] [onitor-migration-
job] - successfully finished execute events table repo key migration
(node:4275) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to
[Server]. Use emitter.setMaxListeners() to increase limit
(Use `node --trace-warnings ...` to show where the warning was created)

root@ip-172-31-13-46:/opt/jfrog/artifactory/var/etc#
```

Access Artifactory from your browser:

http://18.191.14.99:8081/artifactory/

Default credentials:
User: admin, Password: password



* Username: admin

* Password: ••••••••

□ Remember me