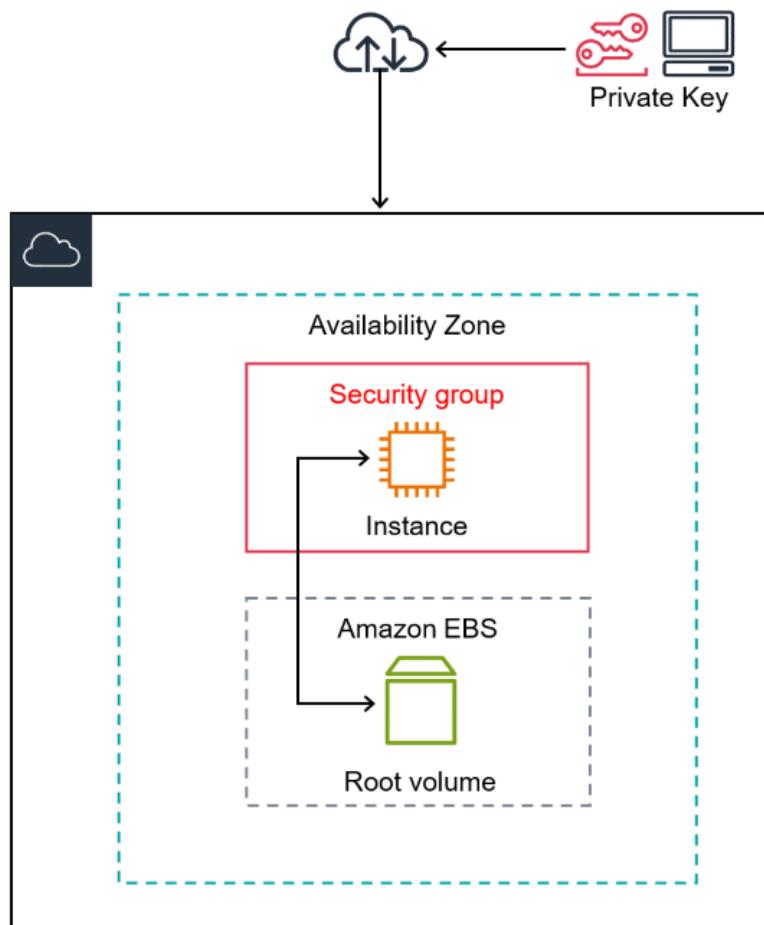


## Overview:

The instance launched in this tutorial is an Amazon EC2 select an Availability Zone for you. Availability Zones are multiple, isolated locations within each Region. You can think of an Availability Zone as an isolated data center. When you launch your instance, you secure it by specifying a key pair (to prove your identity) and a security group (which acts as a virtual firewall to control ingoing and outgoing traffic). When you connect to your instance, you must provide the private key of the key pair that you specified when you launched your instance.



## To launch an instance:

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- From the EC2 console dashboard, in the **Launch instance** box, choose **Launch instance**, and then choose **Launch instance** from the options that appear.
- Under **Name and tags**, for **Name**:Project-1A.

- Under **Application and OS Images (Amazon Machine Image)**, do the following:
  - Choose Amazon Linux. This is the operating system (OS) for your instance.
  - From **Amazon Machine Image (AMI)**, select an Amazon Linux 2. Notice that these AMIs are marked **Free tier eligible**.
- Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the `t2.micro` instance type, which is selected by default. The `t2.micro` instance type is eligible for the free tier.
- Under **Key pair (login)**, for **Project-1A**, choose the key pair that you created when getting set up.
- Warning.
  - Keep the default selections for the other configuration settings for your instance.
  - A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console.
  - On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running`.
  - It can take a few minutes for the instance to be ready for you to connect.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main content area displays the 'Instances (1/2)' table. Two instances are listed: 'terraform' (stopped, t2.micro) and 'project-1A' (stopped, t2.micro). The 'project-1A' row is highlighted with a blue selection bar. Below the table, a modal window titled 'Instance: i-0ff2f6506f0d7f3e4 (project-1A)' is open, showing the 'Inbound rules' section. It lists two rules:

Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-074ebfc0ed0e48618	22	TCP	0.0.0.0/0	launch-wizard-24
-	sgr-06d40246eec33f150	8080	TCP	0.0.0.0/0	launch-wizard-24

(1733) How to Push docker ima x | Ashokrekha/How-to-Push-dock x | Connect to instance | EC2 ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-0ff2f6506f0d7f3e4

aws Services Search [Alt+S] Mumbai Ashok

## Connect to instance Info

Connect to your instance i-0ff2f6506f0d7f3e4 (project-1A) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID  
i-0ff2f6506f0d7f3e4 (project-1A)

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  
3.111.245.25

Username  
root  
Use "root"

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

Cancel Connect

https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0ff2f6506f0d7f3e4&osUser=root&sshPort=22 © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 11:20 AM 1/27/2024

Instances | EC2 | ap-south-1 x | EC2 Instance Connect | ap-south-1 x | (1733) How to Push docker ima x | Ashokrekha/How-to-Push-dock x | +

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0ff2f6506f0d7f3e4&osUser=root&region=ap-south-1&sshPort=22#

aws Services Search [Alt+S] Mumbai Ashok

```
Last login: Thu Jan 4 17:30:32 2024 from ec2-13-233-177-5.ap-south-1.compute.amazonaws.com
,#
~\### Amazon Linux 2
~~\### AL2 End of Life is 2025-06-30.
~~ \#/
~~ \~'-->
~~ / A newer version of Amazon Linux is available!
~~ . / Amazon Linux 2023, GA and supported until 2028-03-15.
~/m/ / https://aws.amazon.com/linux/amazon-linux-2023/
```

[root@ip-172-31-34-36 ~]#

i-0ff2f6506f0d7f3e4 (project-1A)

PublicIPs: 3.111.245.25 PrivateIPs: 172.31.34.36

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 11:21 AM 1/27/2024

## Jenkins installation and configure:

- I have installed Jenkins through yum on Red Hat Enterprise Linux, Alma Linux, Rocky Linux, Oracle Linux, and other Red Hat based distributions.
  - Long Term Support release:

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install fontconfig java-17-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
```
  - You can enable the Jenkins service to start at boot with the command: `sudo systemctl enable jenkins`
  - You can start the Jenkins service with the command: `sudo systemctl start jenkins`
  - You can check the status of the Jenkins service using the command: `sudo systemctl status jenkins`
- Browse to <http://localhost:8080> and wait until the **Unlock Jenkins** page appears.
- From the Jenkins console log output, copy the automatically-generated alphanumeric password.
- On the **Unlock Jenkins** page, paste this password into the **Administrator password** field and click **Continue**.
- After [unlocking Jenkins](#), the **Customize Jenkins** page appears. Here you can install any number of useful plugins as part of your initial setup.
  - **Install suggested plugins** - to install the recommended set of plugins, which are based on most common use cases.
- When the **Create First Admin User** page appears, specify the details for your administrator user in the respective fields and click **Save and Finish**.
- When the **Jenkins is ready** page appears, click **Start using Jenkins**.

Instances | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | (1733) How to Push docker image | Ashokrekha/How-to-Push-docker | Linux

jenkins.io/doc/book/installing/linux/#red-hat-centos

**Jenkins** cd

Blog Success Stories Contributor Spotlight Documentation Plugins Community Subprojects Security About Download Search

> User Documentation Home

**User Handbook**

- [User Handbook Overview](#)
- **Installing Jenkins**
  - Docker
  - Kubernetes
  - **Linux**
  - macOS
  - Windows
  - Other Systems
  - WAR file
  - Other Servlet Containers
  - Offline Installations
  - Initial Settings
- [Platform Information](#)
- [Using Jenkins](#)
- [Pipeline](#)
- [Blue Ocean](#)
- [Managing Jenkins](#)
- [Securing Jenkins](#)
- [System Administration](#)
- [Scaling Jenkins](#)
- [Troubleshooting Jenkins](#)
- [Glossary](#)

**Tutorials**

BASH | Copy to clipboard

## Long Term Support release

A [LTS \(Long-Term Support\) release](#) is chosen every 12 weeks from the stream of regular releases as the stable release for that time period. It can be installed from the [redhat-stable](#) yum repository.

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install fontconfig java-17-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
```

## Weekly release

A new release is produced weekly to deliver bug fixes and features to users and plugin developers. It can be installed from the [redhat](#) yum repository.

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install fontconfig java-17-openjdk
sudo yum install jenkins
```

ENG IN 11:27 AM 1/27/2024

Instances | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | (1733) How to Push docker image | Ashokrekha/How-to-Push-docker | Linux

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-Offf2f6506f0d7f3e4&osUser=root&region=ap-south-1&sshPort=22/#

Services Search [Alt+S]

Mumbai Ashok

```
[root@ip-172-31-34-36 ~]# sudo wget -O /etc/yum.repos.d/jenkins.repo \
>     https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-01-27 05:59:09--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.154.133, 2a04:4e42:24::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.154.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

100%[=====] 85      --.-K/s   in 0s

2024-01-27 05:58:09 (5.61 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@ip-172-31-34-36 ~]# sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@ip-172-31-34-36 ~]# sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package PyYAML.x86_64 0:3.10-11.amzn2.0.2 will be updated
--> Package PyYAML.x86_64 0:3.10-11.amzn2.0.3 will be an update
--> Package binutils.x86_64 0:2.29.1-31.amzn2 will be updated
--> Package binutils.x86_64 0:2.29.1-31.amzn2.0.1 will be an update
--> Package curl.x86_64 0:8.3.0-1.amzn2.0.4 will be updated
--> Package curl.x86_64 0:8.3.0-1.amzn2.0.5 will be an update
--> Package dbus.x86_64 1:1.10.24-7.amzn2.0.3 will be updated
--> Package dbus.x86_64 1:1.10.24-7.amzn2.0.4 will be an update
--> Package dbus-libs.x86_64 1:1.10.24-7.amzn2.0.3 will be updated
--> Package dbus-libs.x86_64 1:1.10.24-7.amzn2.0.4 will be an update

i-Offf2f6506f0d7f3e4 (project-1A)
PublicIPs: 3.111.245.25 PrivateIPs: 172.31.34.36
```

CloudShell Feedback

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 11:29 AM 1/27/2024

Instances | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | (1733) How to Push docker image | Ashokrekha/How-to-Push-docker | Linux

```

aws Services Search [Alt+S] Mumbai Ashok
Verifying : curl-0.3.0-1.amzn2.0.4.x86_64
Verifying : 2:vim-filesystem-9.0.2120-1.amzn2.0.1.noarch
Verifying : 2:vim-data-9.0.2120-1.amzn2.0.1.noarch

Installed:
kernel.x86_64 0:5.10.205-195.807.amzn2

Updated:
PyYAML.x86_64 0:3.10-11.amzn2.0.3
curl.x86_64 0:8.3.0-1.amzn2.0.5
dbus-libs.x86_64 1:1.10.24-7.amzn2.0.4
freetype.x86_64 0:2.8-14.amzn2.1.2
java-17-amazon-corretto-devel.x86_64 1:17.0.10+7-1.amzn2.1
kernel-tools.x86_64 0:5.10.205-195.807.amzn2
libseccomp.x86_64 0:2.5.2-1.amzn2.0.1
ncurses-base.noarch 0:6.0-8.20170212.amzn2.1.7
nss-softokn.x86_64 0:3.90.0-6.amzn2.0.1
postfix.x86_64 2:2.10.1-6.amzn2.0.4
python-devel.x86_64 0:2.7.18-1.amzn2.0.8
python-lxml.x86_64 0:3.2.1-4.amzn2.0.5
python-urllib3.noarch 0:1.25.9-1.amzn2.0.3
traceroute.x86_64 3:2.0.22-2.amzn2.0.2
vim-common.x86_64 2:9.0.2153-1.amzn2.0.1
vim-enhanced.x86_64 2:9.0.2153-1.amzn2.0.1
vim-minimal.x86_64 2:9.0.2153-1.amzn2.0.1

binutils.x86_64 0:2.29.1-31.amzn2.0.1
dbus.x86_64 1:1.10.24-7.amzn2.0.4
dmidecode.x86_64 1:3.2-5.amzn2.1.1
java-17-amazon-corretto.x86_64 1:17.0.10+7-1.amzn2.1
java-17-amazon-corretto-headless.x86_64 1:17.0.10+7-1.amzn2.1
libcurl.x86_64 0:8.3.0-1.amzn2.0.5
ncurses.x86_64 0:6.0-8.20170212.amzn2.1.7
ncurses-libs.x86_64 0:6.0-8.20170212.amzn2.1.7
nss-softokn-freebl.x86_64 0:3.90.0-6.amzn2.0.1
python.x86_64 0:2.7.18-1.amzn2.0.8
python-libs.x86_64 0:2.7.18-1.amzn2.0.8
python-pillow.x86_64 0:2.0.0-23.gidic6db8.amzn2.0.9
tar.x86_64 2:1.26-35.amzn2.0.3
tzdata.noarch 0:2023d-1.amzn2.0.1
vim-data.noarch 2:9.0.2153-1.amzn2.0.1
vim-filesystem.noarch 2:9.0.2153-1.amzn2.0.1
xxd.x86_64 2:9.0.2153-1.amzn2.0.1

Complete!
[root@ip-172-31-34-36 ~]#

```

i-Offf2f6506f0d7f3e4 (project-1A)  
Public IPs: 3.111.245.25 Private IPs: 172.31.34.36

- yum install java -y

Instances | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | (1733) How to Push docker image | Ashokrekha/How-to-Push-docker | Linux

```

aws Services Search [Alt+S] Mumbai Ashok
[root@ip-172-31-34-36 ~]# yum install java -y
Loaded plugins: extras suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package java-17-amazon-corretto.x86_64 1:17.0.10+7-1.amzn2.1 will be installed
--> Processing Dependency: java-17-amazon-corretto-headless(x86-64) = 1:17.0.10+7-1.amzn2.1 for package: 1:java-17-amazon-corretto-17.0.10+7-1.amzn2.1.x86_64
--> Running transaction check
--> Package java-17-amazon-corretto-headless.x86_64 1:17.0.10+7-1.amzn2.1 will be installed
--> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.10+7-1.amzn2.1.x86_64
--> Running transaction check
--> Package javapackages-tools.noarch 0:3.4.1-11.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository      Size
=====
Installing:
java-17-amazon-corretto          x86_64    1:17.0.10+7-1.amzn2.1      amzn2-core   175 k
Installing for dependencies:
java-17-amazon-corretto-headless  x86_64    1:17.0.10+7-1.amzn2.1      amzn2-core   94 M
javapackages-tools                noarch   3.4.1-11.amzn2             amzn2-core   73 k

Transaction Summary
=====
Install 1 Package (+2 Dependent packages)

i-Offf2f6506f0d7f3e4 (project-1A)
Public IPs: 3.111.245.25 Private IPs: 172.31.34.36

```

i-Offf2f6506f0d7f3e4 (project-1A)  
Public IPs: 3.111.245.25 Private IPs: 172.31.34.36

Instances | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | (1733) How to Push docker image | Ashokrekhya/How-to-Push-docker-image | Linux

Total download size: 94 M  
 Installed size: 238 M  
 Downloading packages:  
 (1/3): java-17-amazon-corretto-17.0.10+7-1.amzn2.1.x86\_64.rpm | 175 KB 00:00:00  
 (2/3): javapackages-tools-3.4.1-11.amzn2.noarch.rpm | 73 KB 00:00:00  
 (3/3): java-17-amazon-corretto-headless-17.0.10+7-1.amzn2.1.x86\_64.rpm | 94 MB 00:00:01

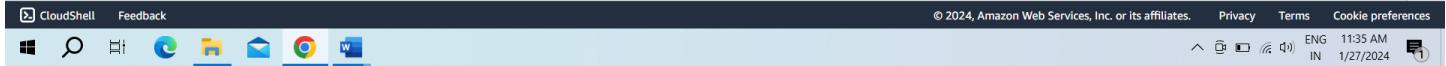
Total 66 MB/s | 94 MB 00:00:01

Running transaction check  
 Running transaction test  
 Transaction test succeeded  
 Running transaction  
 Installing : javapackages-tools-3.4.1-11.amzn2.noarch 1/3  
 Installing : i:java-17-amazon-corretto-headless-17.0.10+7-1.amzn2.1.x86\_64 2/3  
 Installing : i:java-17-amazon-corretto-17.0.10+7-1.amzn2.1.x86\_64 3/3  
 Verifying : i:java-17-amazon-corretto-headless-17.0.10+7-1.amzn2.1.x86\_64 1/3  
 Verifying : i:java-17-amazon-corretto-17.0.10+7-1.amzn2.1.x86\_64 2/3  
 Verifying : javapackages-tools-3.4.1-11.amzn2.noarch 3/3

Installed:  
 java-17-amazon-corretto.x86\_64 1:17.0.10+7-1.amzn2.1  
 Dependency Installed:  
 java-17-amazon-corretto-headless.x86\_64 1:17.0.10+7-1.amzn2.1  
 javapackages-tools.noarch 0:3.4.1-11.amzn2  
 Complete!

[root@ip-172-31-34-36 ~]#

i-Offf2f6506f0d7f3e4 (project-1A)  
 Public IPs: 3.111.245.25 Private IPs: 172.31.34.36



- **systemctl enable --now Jenkins.**
- **systemctl start Jenkins.**
- **systemctl status Jenkins.**

Instances | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | (1733) How to Push docker image | Ashokrekhya/How-to-Push-docker-image | Linux

[root@ip-172-31-34-36 ~]# systemctl enable --now jenkins  
 Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.  
 [root@ip-172-31-34-36 ~]# systemctl start jenkins  
 [root@ip-172-31-34-36 ~]# systemctl status jenkins  
 ● jenkins.service - Jenkins Continuous Integration Server  
 Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)  
 Active: active (running) since Sat 2024-01-27 06:09:07 UTC; 1min 13s ago  
 Main PID: 16063 (java)  
 CGroup: /system.slice/jenkins.service  
 └─16063 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

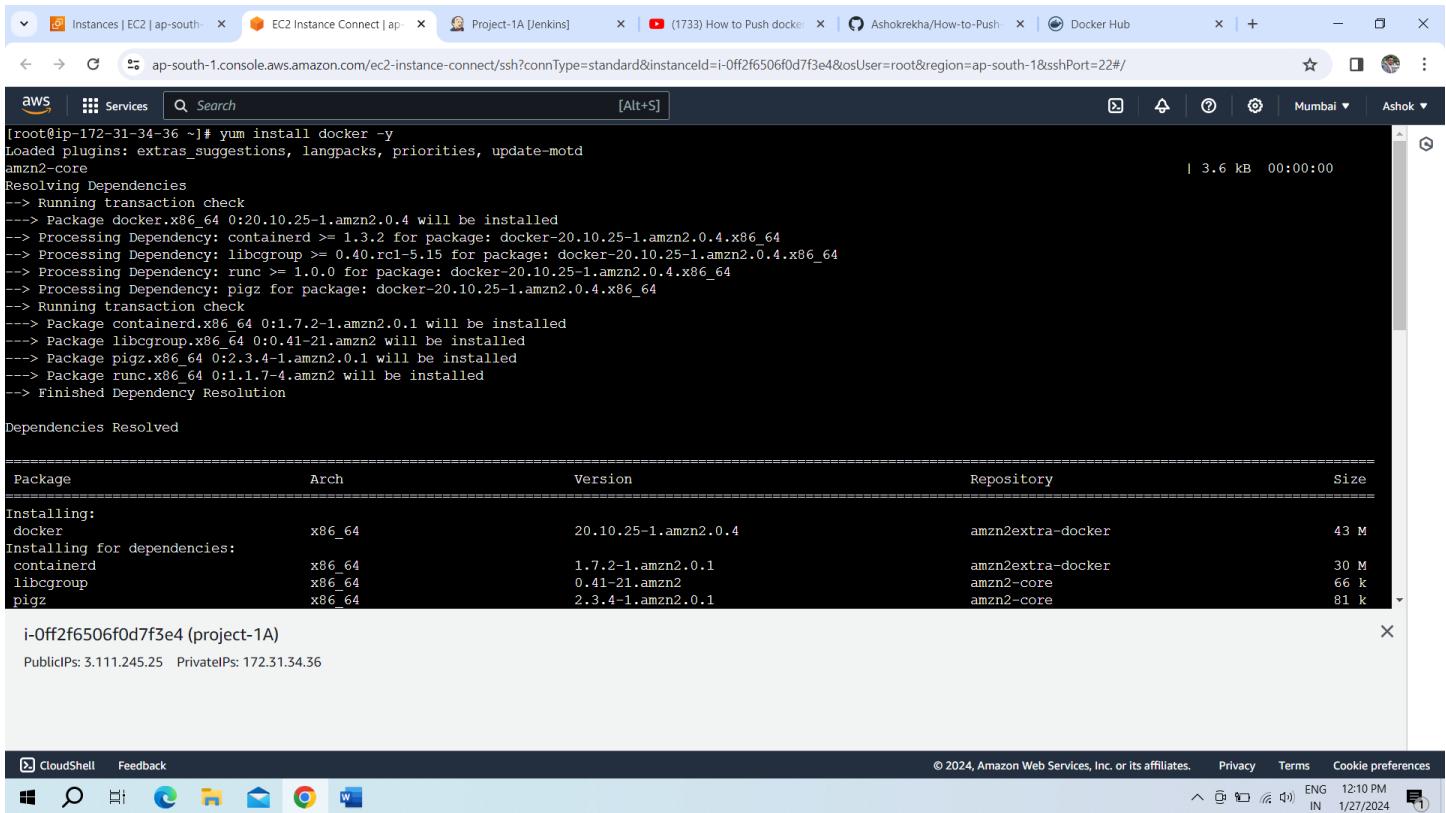
```
Jan 27 06:09:05 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:05.437+0000 [id=32] INFO h.p.b.g.GlobalTimeOutConf...ot set
Jan 27 06:09:06 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:06.655+0000 [id=32] INFO jenkins.InitReactorRunner...nsions
Jan 27 06:09:07 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:07.195+0000 [id=32] INFO jenkins.InitReactorRunner...loaded
Jan 27 06:09:07 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:07.196+0000 [id=32] INFO jenkins.InitReactorRunner...apted
Jan 27 06:09:07 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:07.233+0000 [id=32] INFO jenkins.InitReactorRunner...l jobs
Jan 27 06:09:07 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:07.238+0000 [id=31] INFO jenkins.InitReactorRunner...dated
Jan 27 06:09:07 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:07.445+0000 [id=32] INFO j.install.InstallState$Op...avior
Jan 27 06:09:07 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:07.453+0000 [id=32] INFO jenkins.InitReactorRunner...zation
Jan 27 06:09:07 ip-172-31-34-36.ap-south-1.compute.internal jenkins[16063]: 2024-01-27 06:09:07.541+0000 [id=25] INFO hudson.lifecycle.Lifecycl...unning
Jan 27 06:09:07 ip-172-31-34-36.ap-south-1.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.

Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-34-36 ~]# cd /var/lib/
[root@ip-172-31-34-36 lib]# ls
alternatives authconfig cloud dhclient gssproxy initramfs logrotate misc nfs plymouth rpcbind rpm-state stateless update-motd yum
amazon chrony dbus games hibinit-agent jenkins machines mlocate os-prober postfix rpm rsyslog systemd xfsdump
[root@ip-172-31-34-36 lib]#
```

i-Offf2f6506f0d7f3e4 (project-1A)  
 Public IPs: 3.111.245.25 Private IPs: 172.31.34.36



- `yum install docker -y`



```
[root@ip-172-31-34-36 ~]# yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.25-1.amzn2.0.4 will be installed
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: libcgroup >= 0.40.rc1-5.15 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: pigz for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.7.2-1.amzn2.0.1 will be installed
--> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.1.7-4.amzn2 will be installed
--> Finished Dependency Resolution

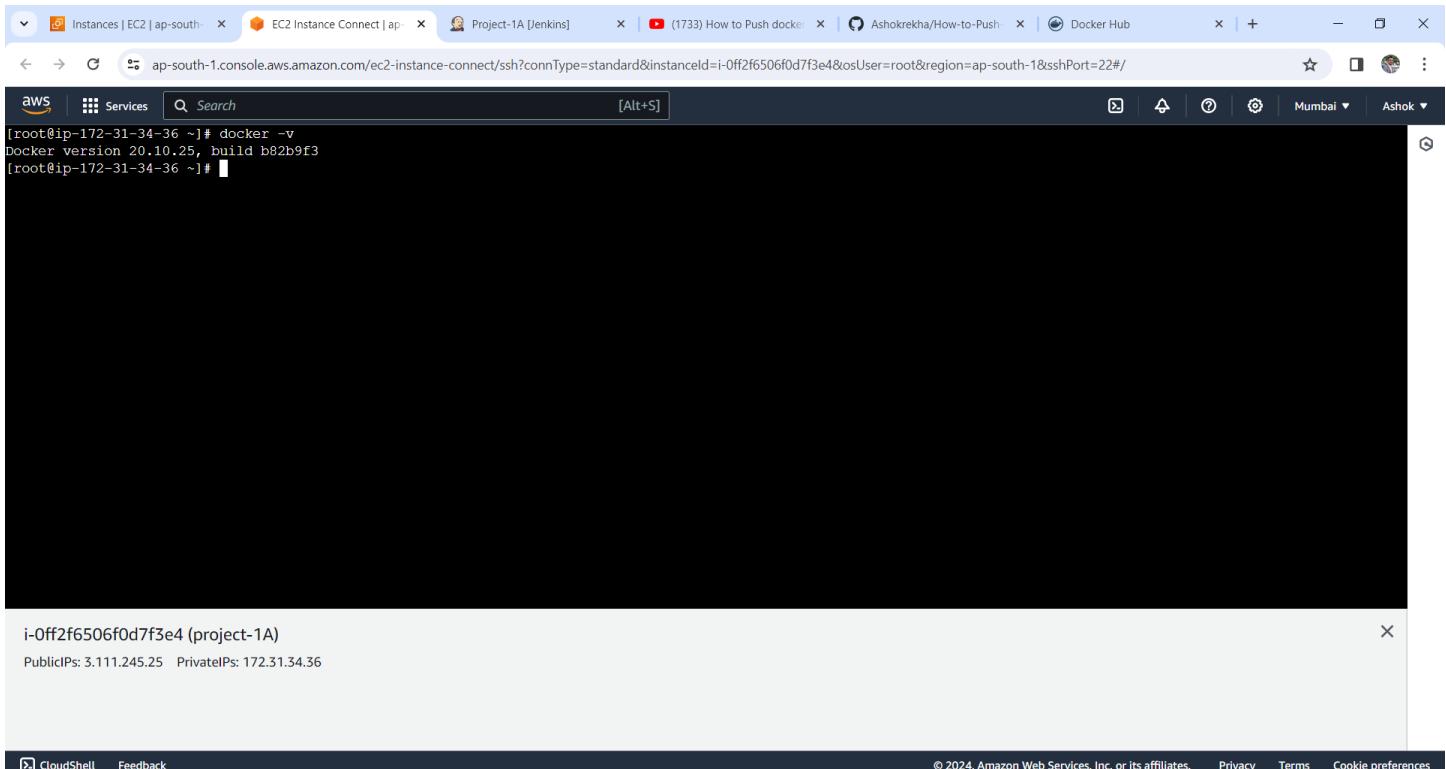
Dependencies Resolved

=====
Package           Arch      Version            Repository      Size
=====
Installing:
docker           x86_64   20.10.25-1.amzn2.0.4    amzn2extra-docker  43 M
Installing for dependencies:
containerd        x86_64   1.7.2-1.amzn2.0.1     amzn2extra-docker  30 M
libcgroup         x86_64   0.41-21.amzn2       amzn2-core          66 k
pigz              x86_64   2.3.4-1.amzn2.0.1    amzn2-core          81 k

i-Offf2f6506f0d7f3e4 (project-1A)
PublicIPs: 3.111.245.25 PrivateIPs: 172.31.34.36
```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 12:10 PM IN 1/27/2024

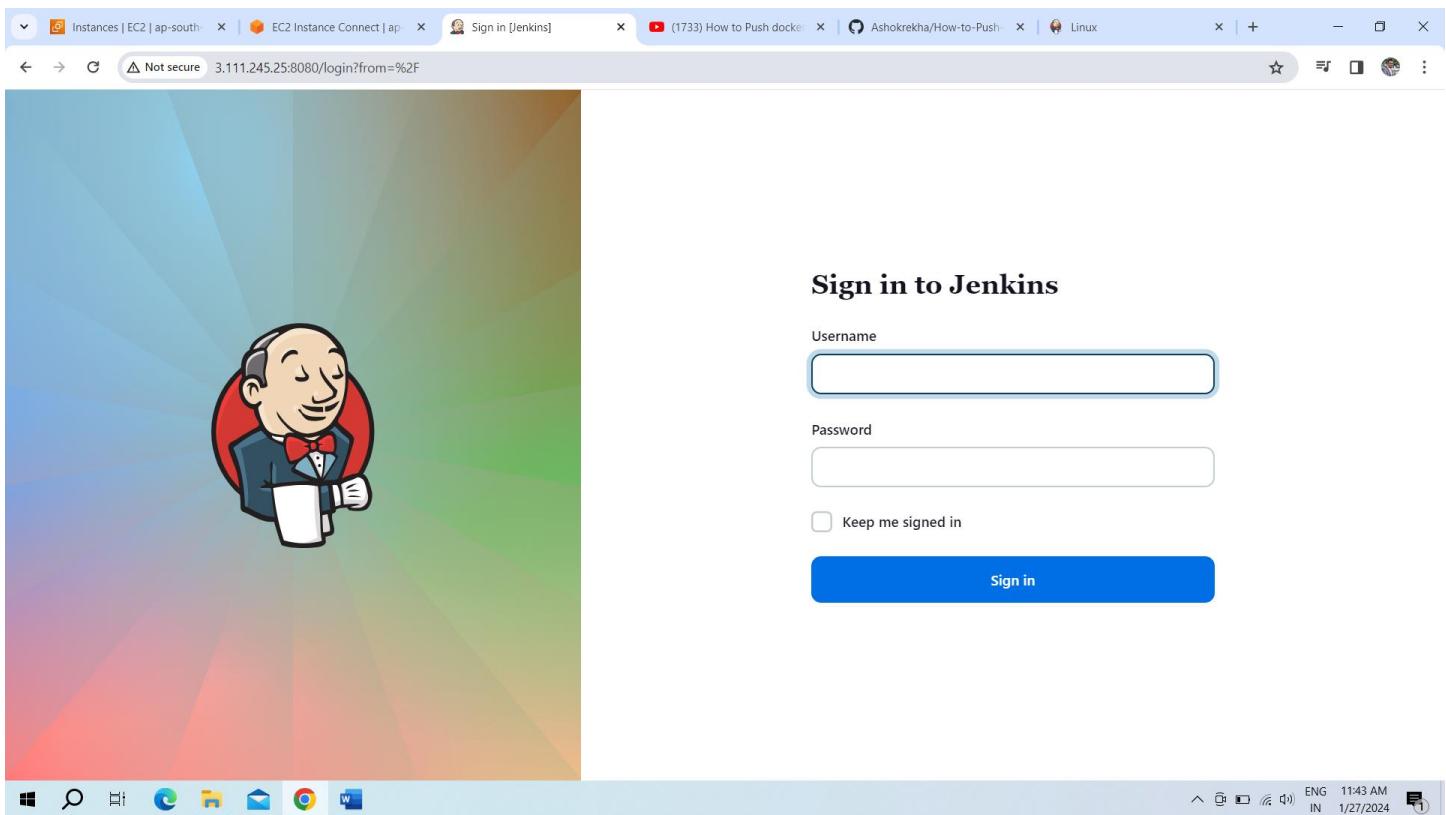
- `docker -v (To check docker version).`



```
[root@ip-172-31-34-36 ~]# docker -v
Docker version 20.10.25, build b82b9f3
[root@ip-172-31-34-36 ~]#
```

i-Offf2f6506f0d7f3e4 (project-1A)
PublicIPs: 3.111.245.25 PrivateIPs: 172.31.34.36

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 12:11 PM IN 1/27/2024



- Jenkins Dashboard

A screenshot of the Jenkins dashboard. The header includes the Jenkins logo, a search bar, and user information for 'Ashok'. The main area has a dark header with 'Welcome to Jenkins!' and a sub-header 'Start building your software project'. It features a 'Create a job' button and sections for 'Set up a distributed build' (with links for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'). On the left, there are navigation links like 'New Item', 'People', 'Build History', etc., and sections for 'Build Queue' (empty) and 'Build Executor Status' (1 Idle, 2 Idle). The footer shows the Jenkins version '2.426.3' and the date '1/27/2024 11:45 AM'.

- Jenkins job with pipeline

Enter an item name

Project-1A  
» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**OK** **branch Pipeline**  
Create a set of Pipeline projects according to detected branches in one SCM repository.

- Access Token for Dockerhub to integrate with Jenkins.

ashok223

User Joined September 20, 2023

Description	Source	Scope	Last Used	Created
jenkins	MANUAL	Read, Write, Delete	Jan 28, 2024 17:29:41	Jan 28, 2024 17:17:10

**Two-Factor Authentication**

Two-factor authentication is not enabled yet.  
Two-factor authentication adds an extra layer of security to your account by requiring more than just a password to sign in. [Learn more](#)

- Credentials for Dockerhub.

The screenshot shows the Jenkins web interface at [43.205.143.13:8080/manage/credentials/](http://43.205.143.13:8080/manage/credentials/). The page title is "Jenkins > Credentials". The main section is titled "Credentials" and lists one item:

T	P	Store ↓	Domain	ID	Name
		System	(global)	dockerhub	ashok223/******** (dockerhub)

Below this, there's a section titled "Stores scoped to Jenkins" which lists:

P	Store ↓	Domains
	System	(global)

At the bottom, there are icons for S, M, and L, and status information: REST API Jenkins 2.426.3, ENG IN 5:45 PM 1/28/2024.

**Jenkins Pipeline:** Jenkins Pipeline is a suite of plugins that allows to define and manage continuous delivery pipelines as code in Jenkins. A pipeline is a set of automated processes that allows to model, orchestrate, and visualize the steps involved in building, testing, and deploying your software. The idea behind Jenkins Pipeline is to enable the creation of complex, multi-step workflows in a more structured and maintainable way.

Jenkins Pipeline is a powerful tool for implementing continuous integration and continuous delivery (CI/CD) practices, helping teams automate and streamline their software delivery processes.

**Access Token:** Docker Hub uses access tokens for authentication and authorization purposes. Access tokens are required to interact with the Docker Hub API and perform actions such as pulling or pushing Docker images.

**Credentials:** In Jenkins, we can manage Docker Hub credentials to securely authenticate and interact with Docker Hub as part of our CI/CD pipeline. This is typically done by configuring Docker Hub credentials in Jenkins so that Jenkins can use them when building, pushing, or pulling Docker images.

- Jenkins pipeline script.

The screenshot shows the Jenkins Pipeline configuration page for a project named "Project-1A". The "Pipeline" tab is selected in the left sidebar. The main area displays the Pipeline script:

```

1 * pipeline{
2     agent any
3     environment {
4         DOCKERHUB_CREDENTIALS=credentials('dockerhub')
5     }
6     stages {
7         stage('gitclone') {
8             steps {
9                 git branch: 'main', url: 'https://github.com/Ashokrekha/live01.git'
10            }
11        }
12    }
13 }
14
15
16
17

```

Below the script, there is a checkbox labeled "Use Groovy Sandbox" which is checked. At the bottom are "Save" and "Apply" buttons.

- Pipeline Syntax.

The screenshot shows the Jenkins Pipeline Syntax configuration page for a project named "Project-1A #2". The "Pipeline Syntax" tab is selected in the left sidebar. On the left, there is a sidebar with links: "Global Variables Reference", "Online Documentation", "Examples Reference", and "IntelliJ IDEA GDSL".

The main area is titled "Sample Step" and shows a "git: Git" step configuration. It includes fields for "Repository URL" (set to "https://github.com/Ashokrekha/live01.git"), "Branch" (set to "main"), and "Credentials" (set to "- none -"). There are also checkboxes for "Include in polling?" and "Include in changelog?", both of which are checked.

At the bottom, there is a "Generate Pipeline Script" button, and the resulting Groovy script is displayed in a code editor:

```
git branch: 'main', url: 'https://github.com/Ashokrekha/live01.git'
```

- Copying of code from Github for Pipeline syntax

The screenshot shows a Microsoft Edge browser window with the GitHub repository 'live01' open. The 'Clone' menu is displayed, showing options for 'Local' (GitHub Desktop) and 'Codespaces'. Under 'Clone', there are three tabs: 'HTTPS' (selected), 'SSH', and 'GitHub CLI'. Below the tabs is a text input field containing the URL <https://github.com/Ashokrekha/live01.git>. A 'Copy url to clipboard' button is visible next to the URL. The GitHub interface shows a list of files: 'src/main' (Update index.html), 'Dockerfile' (Dockerfile), 'Jenkinsfile' (Update Jenkinsfile), 'README.md' (code added), and 'pom.xml' (Update pom.xml). On the right side, there are sections for 'About', 'Releases', 'Packages', and 'Languages'. The status bar at the bottom indicates the system is in English (IN) and the date is 1/28/2024.

- Dockerfile in live01 Repository.

The screenshot shows a Microsoft Edge browser window with the GitHub file 'Dockerfile' in the 'live01' repository open. The left sidebar shows the repository structure with 'main' selected. The main area displays the 'Dockerfile' content, which includes the following code:

```

1 FROM tomcat:9
2 COPY target/*.war /usr/local/tomcat/webapps/ROOT.war

```

The GitHub interface also shows other files like 'src/main', 'Dockerfile', 'Jenkinsfile', 'README.md', and 'pom.xml' in the sidebar. The status bar at the bottom indicates the system is in English (IN) and the date is 1/28/2024.

- **Pipeline Script:**

```
pipeline{

    agent any

    environment {

        DOCKERHUB_CREDENTIALS=credentials('dockerhub')

    }

    stages {

        stage('gitclone') {

            steps {

                git branch: 'main', url: 'https://github.com/Ashokrekha/live01.git'

            }

        }

        stage('Build') {

            steps {

                sh 'docker build -t ashok223/project-1a .'

            }

        }

        stage('Login') {

            steps {

                sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u
$DOCKERHUB_CREDENTIALS_USR --password-stdin'

            }

        }

        stage('Push') {

            steps {

                sh 'docker push ashok223/project-1a'

            }

        }

        stage('deploy') {

            steps {

                sh 'docker run -d -p 8081:8080 --name project-1a ashok223/project-1a'

            }

        }

    }

}
```

```

    }
}

post {
    always {
        sh 'docker logout'
    }
}

```

- **Dockerfile:**

```

FROM tomcat:9
COPY target/*.war /usr/local/tomcat/webapps/ROOT.war

```

- Build status of project-1A.

gitclone	Build	Login	Push	deploy	Declarative: Post Actions
807ms	9s	1s	7s	676ms	432ms
842ms	17s	2s	15s	1s	498ms
773ms	400ms failed	74ms failed	58ms failed	53ms failed	366ms

Average stage times: (Average full run time: ~39s)

Stage View

gitclone Build Login Push deploy Declarative: Post Actions

#2 Jan 28 21:44 No Changes

#1 Jan 28 21:43 No Changes

Project-1A

Status Changes Build Now Configure Delete Pipeline Full Stage View Rename Pipeline Syntax

Add description Disable Project

Build History trend Filter builds... #2 Jan 28, 2024, 4:14 PM #1 Jan 28, 2024, 4:13 PM

Atom feed for all Atom feed for failures

Permalinks

Last build (#2) 9 min 25 sec ago

- Console output.

```

48a6a6a380ec: Mounted from library/tomcat
3889857e8e17: Mounted from library/tomcat
334771ac3946: Mounted from library/tomcat
9d23663b4305: Mounted from library/tomcat
d221051zedb4: Mounted from library/tomcat
8e87ff28f1b5: Mounted from library/tomcat
7142b04460fb: Pushed
latest: digest: sha256:d337e765daf90fe84b598c86a9cc517d23f96a9cd1124e05364c881d59a11e7d size: 2207
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] {
[Pipeline] { (deploy)
[Pipeline] sh
+ docker run -d -p 8001:8080 --name project-1a ashok223/project-1a
c228b00470745b0a67cf109ddaf5cd927c401cb5503745c1373bb46975a58cf8
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] {
[Pipeline] { (Declarative: Post Actions)
[Pipeline] sh
+ docker logout
Removing login credentials for https://index.docker.io/v1/
[Pipeline]
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

- Pushed image to Dockerhub (ashok223/project-1a).

**ashok223 / project-1a**

**Description**  
This repository does not have a description

**Last pushed:** 10 minutes ago

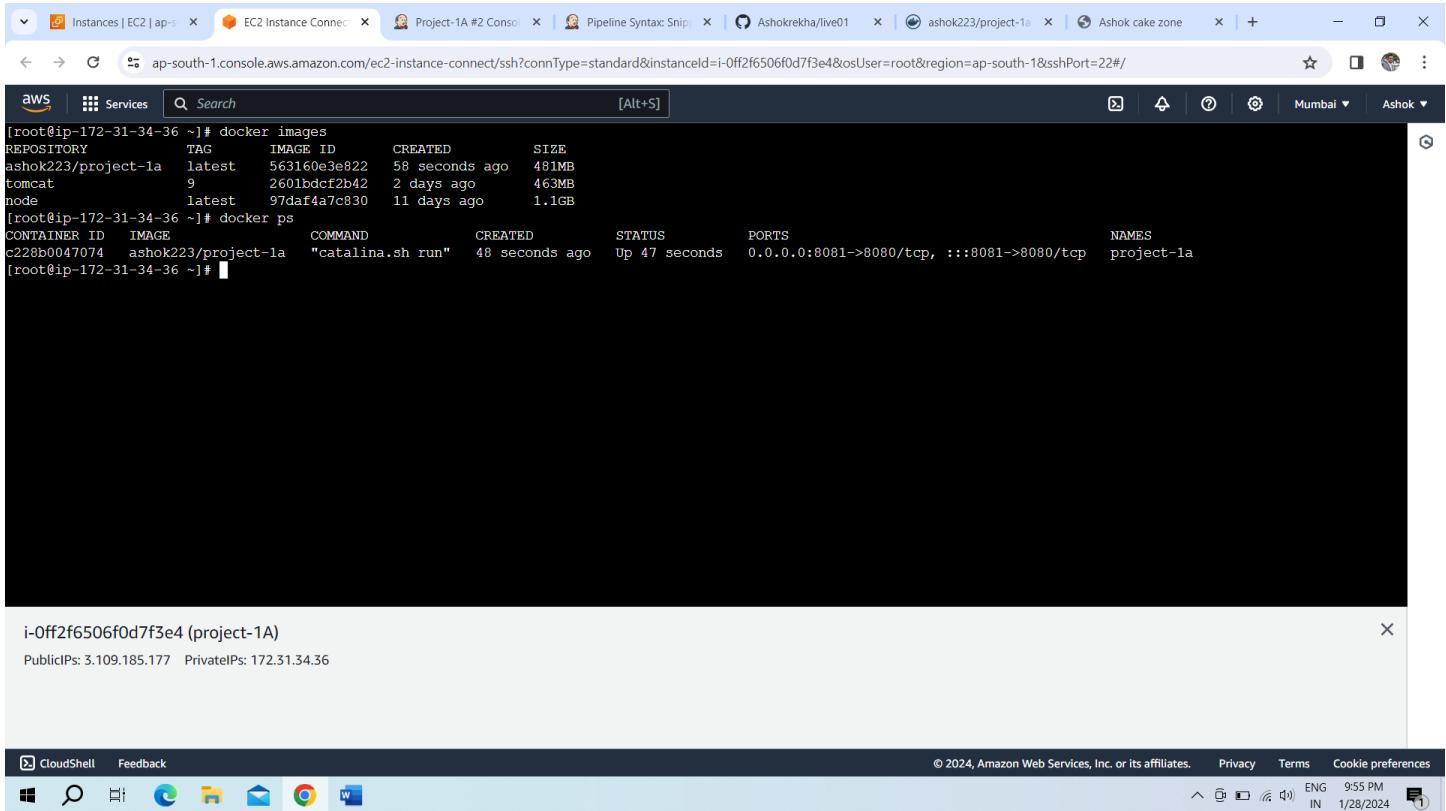
**Tags**  
This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	--	10 minutes ago

**Automated Builds**  
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.  
Available with Pro, Team and Business subscriptions. [Read more about automated builds](#).

**Docker commands**  
To push a new tag to this repository:  
`docker push ashok223/project-1a:tagname`

- Created docker images (ashok223/project-1a, tomcat) and Container (project-1a).



```
[root@ip-172-31-34-36 ~]# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
ashok223/project-1a  latest   563160e3e822  58 seconds ago  481MB
tomcat              9        2601bdcf2b42  2 days ago    463MB
node                latest   97daf4a7c830  11 days ago   1.1GB
[root@ip-172-31-34-36 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
c228b0047074        ashok223/project-1a "catalina.sh run"  48 seconds ago   Up 47 seconds   0.0.0.0:8081->8080/tcp, :::8081->8080/tcp   project-1a
[root@ip-172-31-34-36 ~]#
```

i-Offf2f6506f0d7f3e4 (project-1A)  
PublicIPs: 3.109.185.177 PrivateIPs: 172.31.34.36

**Docker Image:** Docker images are a fundamental component in the Docker ecosystem, a platform for developing, shipping, and running applications in containers. Containers provide a lightweight and consistent environment for applications to run, allowing them to be easily deployed across different environments.

**Dockerfile:** The Dockerfile is a script that contains instructions for building a Docker image. It specifies the base image, sets up the environment, copies files, installs dependencies, and defines other configurations needed for the application.

**Base Image:** The base image is the starting point for a Docker image. It could be a minimal Linux distribution or a specialized image with certain tools pre-installed. Images are often built on top of other images, forming a hierarchy.

**Docker Container:** A Docker container is an instance of a Docker image. It is a runnable, isolated process that encapsulates an application and its dependencies. Containers are created from images and can be started, stopped, moved, and deleted independently of one another.

**Docker Hub:** Docker Hub is a public registry that allows users to share and distribute Docker images. It hosts a vast collection of pre-built images that can be used as base images for various applications.

- Prot number 8081 for project-1a container.

Screenshot of the AWS CloudShell interface showing the 'Edit inbound rules' section for a security group. The table lists three existing rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-070b947b30479fd00	Custom TCP	TCP	8081	Custom	0.0.0.0/0
sgr-074ebfc0ed0e48618	SSH	TCP	22	Custom	0.0.0.0/0
sgr-06d40246eec33f150	Custom TCP	TCP	8080	Custom	0.0.0.0/0

**Add rule**

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

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 9:56 PM 1/28/2024

- Instance public ip:8081 (CAKEZONE Application).

Screenshot of a web browser displaying the CAKEZONE website. The page features a large image of a chocolate cake with cherries and white frosting. The navigation menu includes HOME, ABOUT US, MENU & PRICING, MASTER CHEFS, PAGES, and CONTACT US.

EMAIL US ashokrekha223@gmail.com

CALL US +012 345 6789

Super Crispy

# CAKEZONE

THE BEST CAKE IN LONDON

Read More

Play Video

Not secure 3.109.185.177:8081

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 9:56 PM 1/28/2024