

DevOps Project

Problem Statement:

Create an end-to-end CI/CD pipeline in AWS platform using Jenkins as the orchestration tool, GitHub as scm, maven as the build tool, deploy in a docker instance and create a docker image, store the docker image in ECR, Kubernetes deployment using ECR image. Build sample java web app using Maven.

Approach:

Requirements:

- **CI/CD pipeline System**
- **Git** - Local version control system
- **GitHub** - Distributed version control system
- **Jenkins** - Continuous Integration tool
- **Maven** - Build Tool
- **Docker** - Containerization
- **Kubernetes** - Container Management Tool

Step-1:

1. Set up CI/CD with GitHub, Jenkins, Maven & Tomcat:

- **Install Jenkins:**
 - Download and install Jenkins on your server or local machine.
 - Start Jenkins and complete the initial setup wizard.
- **Configure Maven and Git:**
 - Install Maven and Git on the same machine where Jenkins is installed.
 - Configure the Maven and Git paths in Jenkins under “Global Tool Configuration”.
- **Set up Tomcat Server:**
 - Download and install Apache Tomcat.
 - Configure Tomcat to run on a specific port and ensure it is running.

- **Integrate GitHub, Maven, and Tomcat Server with Jenkins:**
 - Create a GitHub repository for your project.
 - In Jenkins, create a new job and configure it to pull code from your GitHub repository.
 - Add Maven build steps to compile and package your application.
 - Configure Jenkins to deploy the packaged application to the Tomcat server.
- **Create CI and CD Jobs:**
 - Set up a Continuous Integration (CI) job to automatically build and test your code whenever changes are pushed to GitHub.
 - Set up a Continuous Deployment (CD) job to deploy the application to Tomcat after a successful build.
- **Test the Deployment:**
 - Push changes to your GitHub repository and verify that Jenkins automatically builds, tests, and deploys the application to Tomcat

Step-2:

1. **Set up CI/CD with GitHub, Jenkins, Maven & Docker:**
 - **Set up the Docker Environment:**
 - Install Docker on your server or local machine.
 - Verify the Docker installation by running a test container.
 - **Create an Image and Container on Docker Host:**
 - Write a Dockerfile to define your application's environment.
 - Build the Docker image using the Dockerfile.
 - Run a container from the Docker image to ensure it works as expected.
 - **Integrate Docker Host with Jenkins:**
 - Install the Docker plugin in Jenkins.
 - Configure Jenkins to communicate with the Docker daemon.
 - Set up credentials in Jenkins to access the Docker host.
 - **Create CI/CD Job on Jenkins to Build and Deploy on Container:**

- Create a new Jenkins job and configure it to pull code from your GitHub repository.
- Add build steps to use Maven for compiling and packaging the application.
- Add steps to build the Docker image and push it to a Docker registry.
- Configure Jenkins to deploy the Docker container using the newly built image

Step-3:

1. Build and Deploy on Container:

- **CI/CD with GitHub, Jenkins, Maven & Kubernetes:**
 - Set up a CI/CD pipeline integrating GitHub, Jenkins, Maven, and Kubernetes.
- **Set up Kubernetes (EKS):**
 - Create and configure an Amazon EKS (Elastic Kubernetes Service) cluster.
 - Ensure the cluster is properly set up and accessible.
- **Write Pod, Service, and Deployment Manifest Files:**
 - Create Kubernetes manifest files for Pods, Services, and Deployments.
 - Define the desired state and configuration for your application in these files.
- **CI/CD Job to Build Code on Jenkins & Deploy it on Kubernetes:**
 - Create a Jenkins job to pull code from your GitHub repository.
 - Add build steps to use Maven for compiling and packaging the application.
 - Configure Jenkins to apply the Kubernetes manifest files to deploy the application to the EKS cluster.

Step-4:

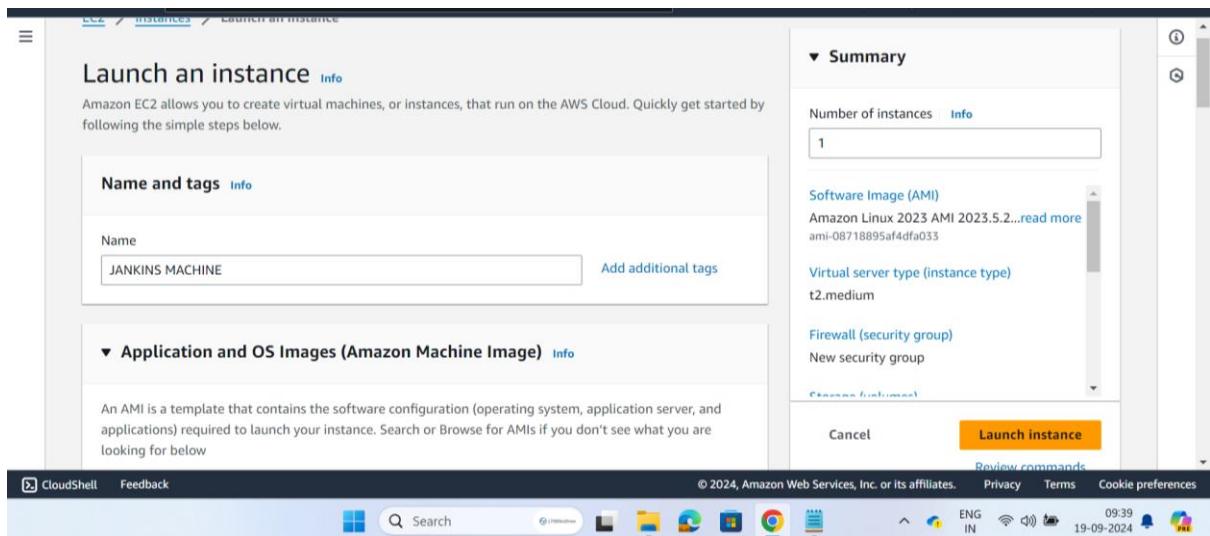
• Deploy artifacts on Kubernetes:

- Write codes in the artifacts of Docker and Kubernetes which we want to run.
- **Build the Code in Jenkins:**
 1. Create a Jenkins job to pull the latest code from your GitHub repository.
 2. Add build steps to compile and package the application using Maven.
 3. Ensure the build artifacts are correctly generated and stored.
- Check in Kubernetes if the pods are getting created or not.
- Now copy the service IP and paste it in the browser and check the output.

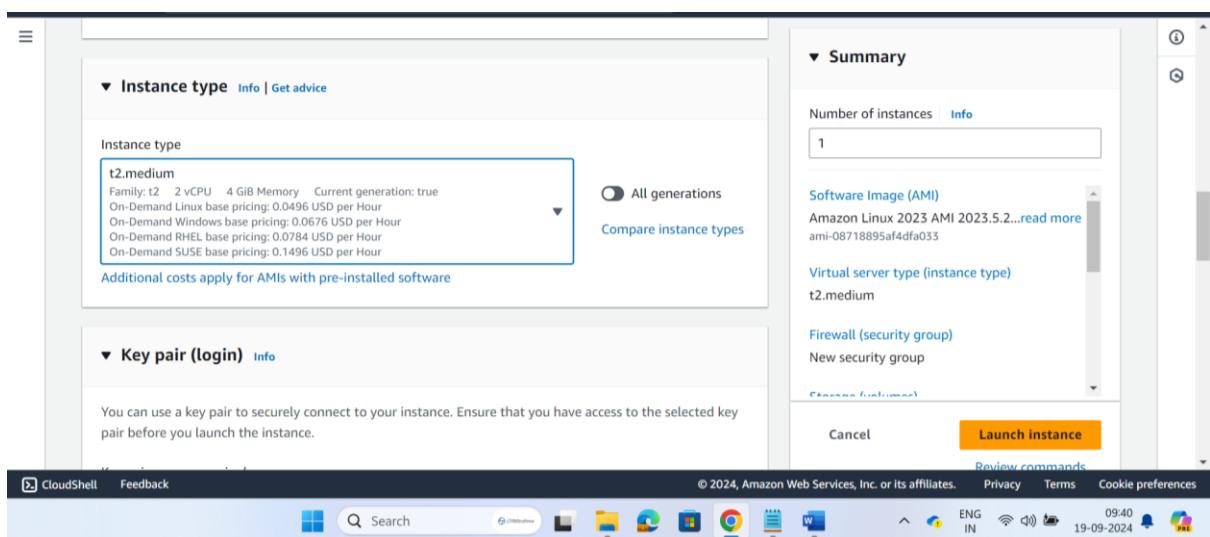
Step :

1. Setting Up EC2 Instances
 1. 1.1 Creating the EC2 Instances
 2. Jenkins Server: Choose an t2.medium instance type because Jenkins, being an orchestration tool, requires more resources for handling builds, running jobs, and managing plugins.
 3. Developer Server and Tomcat Server: Use t2.micro instances as they will be handling lighter tasks, such as serving as the development environment and hosting the application.
 4. Instance Configuration: Key Pair: Create or use an existing key pair to connect securely to these instances via SSH.

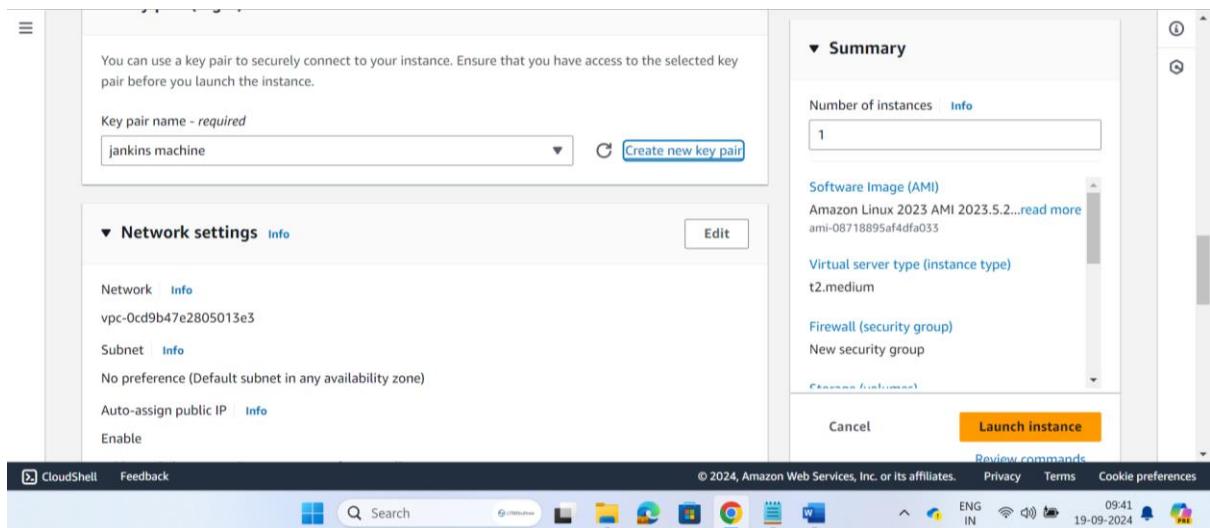
Create three instances developer,Jenkins and tomcat:



Select t2medium in jenkin machine ,t2micro for Developer and tomcat machine:



Take same key pair for all the Machine:



VPC - required [Info](#)
vpc-0cd9b47e2805013e3 (default)
Subnet [Info](#)
subnet-08f278cb89dd16890 VPC: vpc-0cd9b47e2805013e3 Owner: 559050235518 Availability Zone: ap-south-1a Zone type: Availability Zone IP addresses available: 4090 CIDR: 172.31.32.0/20
Create new subnet [Info](#)
Auto-assign public IP [Info](#)
Enable Additional charges apply when outside of free tier allowance
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
Security group name - required
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 09:42 19-09-2024

Created Security group and exposed port 8080 :

Additional charges apply when outside of free tier allowance
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
Security group name - required
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#,@[]+=;&;\$*
Description - required [Info](#)
launch-wizard-2 created 2024-09-19T04:03:29.001Z
Inbound Security Group Rules
▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) Remove
Type [Info](#) Protocol [Info](#) Port range [Info](#)
ssh TCP 22
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 09:42 19-09-2024

Created three instances for Developer,Jenkins and Tomcat:

EC2 Dashboard EC2 Global View Events
▼ Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations [New](#)
▼ Images AMIs
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 09:52 19-09-2024

Instances (3) Info					Last updated less than a minute ago	C	Connect	Instance state ▼	Actions ▼	Launch instances	▼
						Find Instance by attribute or tag (case-sensitive)		Running ▼			
	Name ▼	Instance ID	Instance state	Instance type	Status						
<input type="checkbox"/>	JANKINS MACHINE	i-02d92113adcb2285	Running Q Q	t2.medium	Running 2						
<input type="checkbox"/>	Tomcat-server	i-0a76ab91b116bc4fb	Running Q Q	t2.micro	Running 1						
<input type="checkbox"/>	Developer-server	i-0cab1360bbbc59b86	Running Q Q	t2.micro	Running 1						

Select an instance [X](#)

Created ssh-keygen

The screenshot shows a terminal window titled 'root@ip-172-31-38-164:~'. The user has run the command 'ssh-keygen' to generate an RSA key pair. The process involves saving the private key to '/root/.ssh/id_rsa' and the public key to '/root/.ssh/id_rsa.pub'. The key's randomart image is displayed as a grid of characters. The terminal also shows the SHA256 fingerprint of the key.

```
root@ip-172-31-38-164:~$ sudo su -
[ec2-user@ip-172-31-38-164 ~]$ hostnamectl set-hostname developer.example.com
[ec2-user@ip-172-31-38-164 ~]$ bash
[root@developer ~]# ssh-keygen
Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa): Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:qBi06Sglmhak5ZI38+mzJ3Klp1Bx54V7AtFmLaGRqE root@developer.example.com
The key's randomart image is:
+---[RSA 3072]---+
| .. = .
| .. + +
| E. + * .
| . + * o
| . . S + +
| .o++ . . o .
|=BB.o . . o
|O=.o+ooo.o
|+. . * ==
+---[SHA256]---+
[root@developer ~]#
[root@developer ~]#
```

Install git

The screenshot shows a terminal window titled 'root@ip-172-31-38-164:~'. The user has run the command 'yum install git -y' to install the Git version control system.

```
root@ip-172-31-38-164:~$ yum install git -y
```

Create directory and create public key

```
[root@developer ~]# mkdir /data
[root@developer ~]# cd /data
[root@developer data]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /data/.git/
[root@developer data]# cd
[root@developer ~]# cd .ssh/
[root@developer .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCmBWTvtjnGPKrnXWwvlggWmMeAyc+EafkQYN6SfUiS3+N8qlfR7+aodpMnfpQgg0yJVOXOHyCgjc0d4A2/qVDEP8IV1GhbB2QDp6aPiU/UFrQ5atiEZh9DFisZupvQSqjfDcY4toOE9pxUV4ViuiQqqwi9W5yyiuLbLzwOb3RpfZwdXb9ie9Gjrlkbbm9Vpn2tYJV85dQoW/V2pi0g9TvcTrnmeCCAHVeXb2A+IYtiN23vd1z1IKbk3V7uguiWj+gts1t4JmC1+9CLIqfSw/Y+PrvjxVQkt706u3+wijTVfkZ8f6o2WVNMKHzD4cHmNoO5pAZ7D5y5GVL8xMsxubRcnla8Q55asAv28kJXzQ5SnlwIHxaCBIQ2cURQbwCcr6n1SC+1xTijOzsC7J38MWsZv79TNV1FsKEkry2ryUi/0KD4pSRMreLs8qHSB3/1QdyV1qZolWSHGnlt1t7y6eqxQBNaq93fxCFAHTLK4s= root@developer.example.com
[root@developer .ssh]#
```

Copy jenkin machine public key and paste in github

The screenshot shows a browser window with the URL github.com/settings/ssh/new. The user is logged in as **Chiragsinghal123 (Chiragsinghal123)**. On the left, there's a sidebar with account settings like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, and **SSH and GPG keys** (which is currently selected). The main area is titled "Add new SSH Key". It has fields for "Title" (set to "project") and "Key type" (set to "Authentication Key"). Below these is a large text input field containing the copied SSH public key:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCmBWTvtjnGPKrnXWwvlggWmMeAyc+EafkQYN6SfUiS3+N8qlfR7+aodpMnfpQgg0yJVOXOHyCgjc0d4A2/qVDEP8IV1GhbB2QDp6aPiU/UFrQ5atiEZh9DFisZupvQSqjfDcY4toOE9pxUV4ViuiQqqwi9W5yyiuLbLzwOb3RpfZwdXb9ie9Gjrlkbbm9Vpn2tYJV85dQoW/V2pi0g9TvcTrnmeCCAHVeXb2A+IYtiN23vd1z1IKbk3V7uguiWj+gts1t4JmC1+9CLIqfSw/Y+PrvjxVQkt706u3+wijTVfkZ8f6o2WVNMKHzD4cHmNoO5pAZ7D5y5GVL8xMsxubRcnla8Q55asAv28kJXzQ5SnlwIHxaCBIQ2cURQbwCcr6n1SC+1xTijOzsC7J38MWsZv79TNV1FsKEkry2ryUi/0KD4pSRMreLs8qHSB3/1QdyV1qZolWSHGnlt1t7y6eqxQBNaq93fxCFAHTLK4s= root@developer.example.com|
```

Git pull java application in my developer machine by git ssh

```
[root@developer .ssh]# cd
[root@developer ~]# cd /d
data/ dev/
[root@developer ~]# cd /data
[root@developer data]# git pull git@github.com:Chiragsinghal123/clone-.git
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3vvvV6TuJhbPZisf/zLDA0zPMsvHdkx4UvC0qU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 129, done.
remote: Counting objects: 100% (129/129), done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 129 (delta 35), reused 129 (delta 35), pack-reused 0 (from 0)
Receiving objects: 100% (129/129), 12.41 MiB | 4.94 MiB/s, done.
Resolving deltas: 100% (35/35), done.
From github.com:Chiragsinghal123/clone-
 * branch      HEAD      -> FETCH_HEAD
[root@developer data]#
```

The screenshot shows a web browser window with multiple tabs open. The active tab is for the GitHub repository 'Chiragsinghal123/clone-'. The repository page displays the following details:

- Code** tab is selected.
- Branch**: main (1 Branch)
- Tags**: 0 Tags
- Files**:
 - sanjayguruji (Add files via upload)
 - server (first commit)
 - webapp (Changed Index.jsp)
 - Dockerfile (first commit)
 - README.md (Changed README)
 - Snjay_Devops_project.pdf (Add files via upload)
 - pom.xml (first commit)
 - regapp-deploy.yaml (Update regapp-deploy.yaml last year)
 - regapp-service.yaml (Update regapp-service.yaml last year)
- Clone** section:
 - Local (selected)
 - Codespaces
 - HTTPS
 - SSH (selected)
 - GitHub CLI
- SSH URL**: git@github.com:chiragsinghal123/clone-.git
- Actions**:
 - Open with GitHub Desktop
 - Download ZIP
- About**: No description, website, or topics provided.
- Readme**: Readme
- Activity**: 0 stars
- Watching**: 1 watching
- Forks**: 0 forks
- Releases**: No releases published. Create a new release
- Packages**: No packages published

Change branch name

```
From github.com:Chiragsinghal123/clone-
 * branch      HEAD      -> FETCH_HEAD
[root@developer data]# git branch
* master
[root@developer data]# git branch -M main
[root@developer data]# git branch
* main
[root@developer data]#
```

In jenkin machine change host name and install JAVA

The screenshot shows a terminal window with several tabs open. The current tab displays the output of a command that adds a host key to the known hosts file. Below that, the host name is changed to 'jenkin.example.com'. The terminal then runs a 'dnf install' command for Java 17, which includes Corretto. The package manager lists the installed packages and their details, including the Java package itself and its dependencies like fonts and libraries.

```
ED25519 key fingerprint is SHA256:qLcHGl+xIU2GwL/MuFkwby+qa8/7RId85eFIY2GZfxk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-47-129-218-82.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

[ec2-user@ip-172-31-32-175 ~]$ sudo su -
[root@ip-172-31-32-175 ~]# hostnamectl set-hostname jankin.example.com
[root@ip-172-31-32-175 ~]# bash
[root@jankin ~]#
[root@jankin ~]# dnf install java-17-amazon-corretto -y
Last metadata expiration check: 0:01:41 ago on Thu Sep 19 05:02:12 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository      Size
=====
Installing:
java-17-amazon-corretto x86_64      1:17.0.12+7-1.amzn2023.1   amazonlinux    187 k
Installing dependencies:
alsa-lib             x86_64      1.2.7.2-1.amzn2023.0.2   amazonlinux    504 k
cairo                x86_64      1.17.6-2.amzn2023.0.1   amazonlinux    684 k
dejavu-sans-fonts   noarch     2.37-16.amzn2023.0.2   amazonlinux    1.3 M
dejavu-sans-mono-fonts noarch     2.37-16.amzn2023.0.2   amazonlinux    467 k
dejavu-serif-fonts  noarch     2.37-16.amzn2023.0.2   amazonlinux    1.0 M
fontconfig           x86_64      2.13.94-2.amzn2023.0.2  amazonlinux    273 k
```

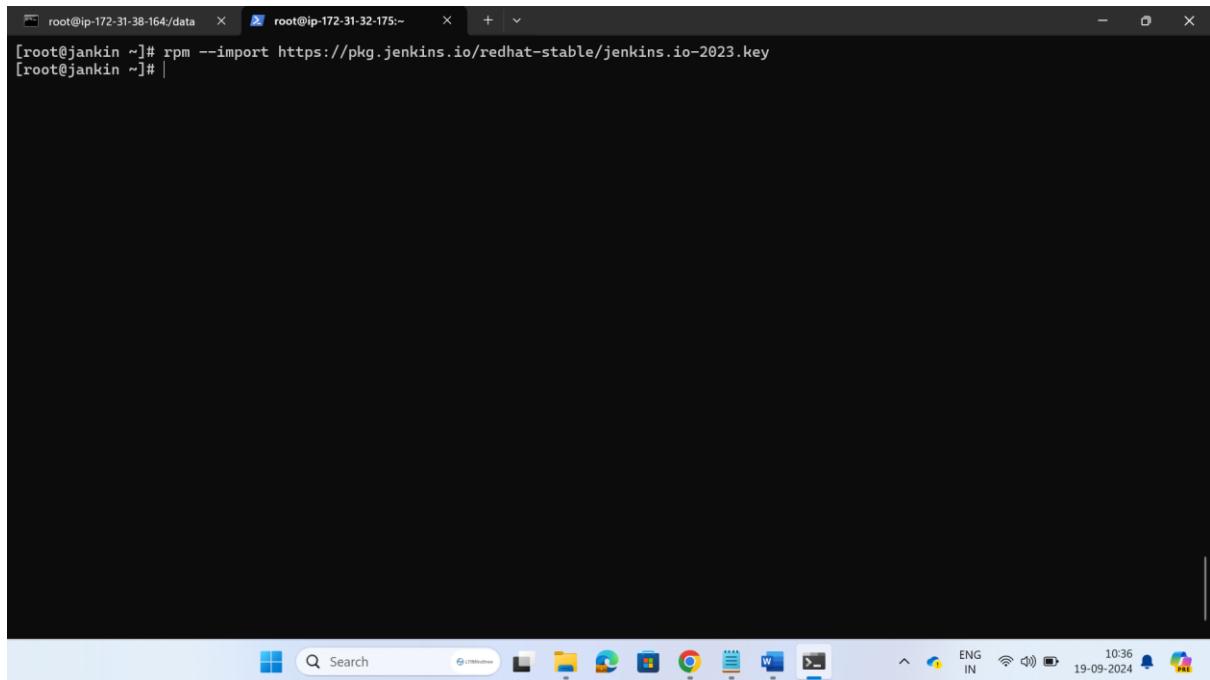
Check java version

The screenshot shows a terminal window where the Java version is checked. The command 'java -version' is run, displaying the OpenJDK version '17.0.12' and the build information '2024-07-16 LTS'. The user then proceeds to download the Jenkins repository by running 'wget' to fetch the 'jenkins.repo' file from 'pkg.jenkins.io'. The download progress is shown, and once completed, the file is saved to '/etc/yum.repos.d/jenkins.repo'.

```
[root@jankin ~]# java -version
openjdk version "17.0.12" 2024-07-16 LTS
OpenJDK Runtime Environment Corretto-17.0.12.7.1 (build 17.0.12+7-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.12.7.1 (build 17.0.12+7-LTS, mixed mode, sharing)
[root@jankin ~]# wget -O /etc/yum.repos.d/jenkins.repo \
  https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-09-19 05:05:28-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 199.232.46.133, 2a04:4e42:48::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|199.232.46.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

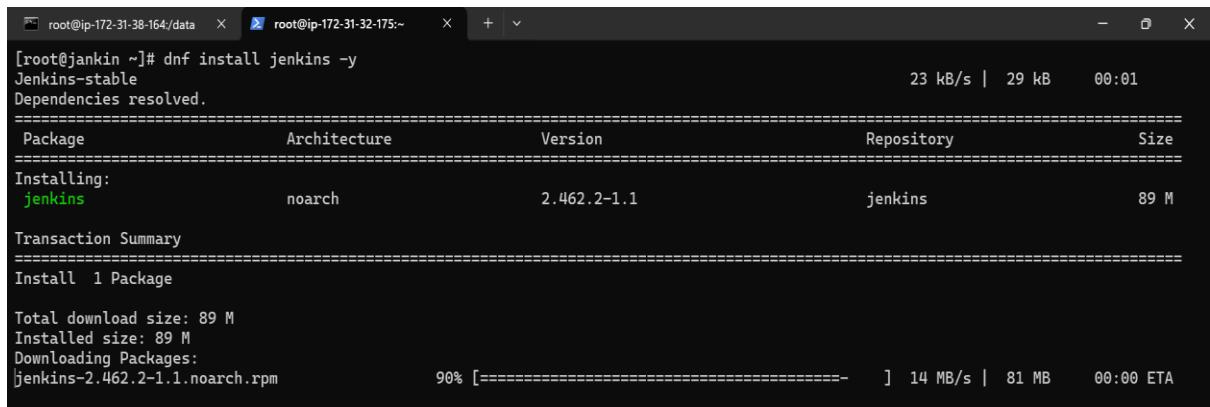
/etc/yum.repos.d/jenkins.repo    100%[=====]     85 --.-KB/s   in 0s

2024-09-19 05:05:28 (3.62 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]
[root@jankin ~]#
```



```
[root@jankin ~]# rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

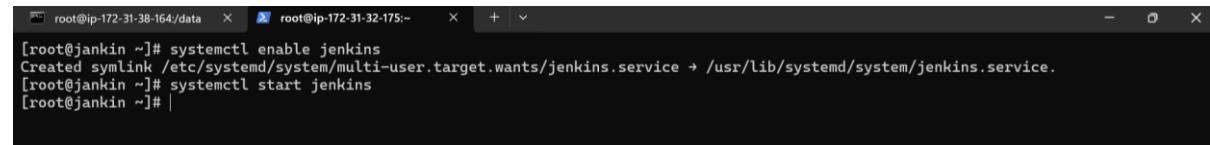
Install jenkins and java in jenkin machine



```
[root@jankin ~]# dnf install jenkins -y
Jenkins-stable
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
jenkins          noarch          2.462.2-1.1   jenkins        89 M
Transaction Summary
=====
Install 1 Package

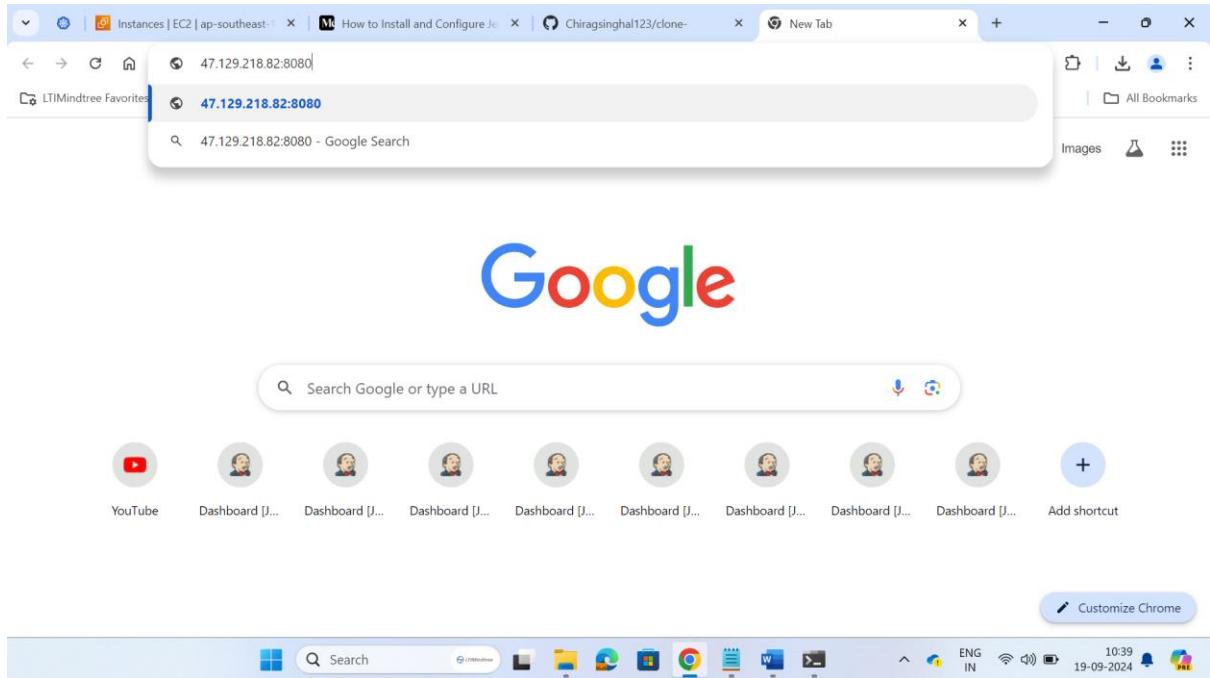
Total download size: 89 M
Installed size: 89 M
Downloading Packages:
jenkins-2.462.2-1.1.noarch.rpm      90% [=====] 14 MB/s | 81 MB  00:00 ETA
```

Jenkin machine is start and enable:

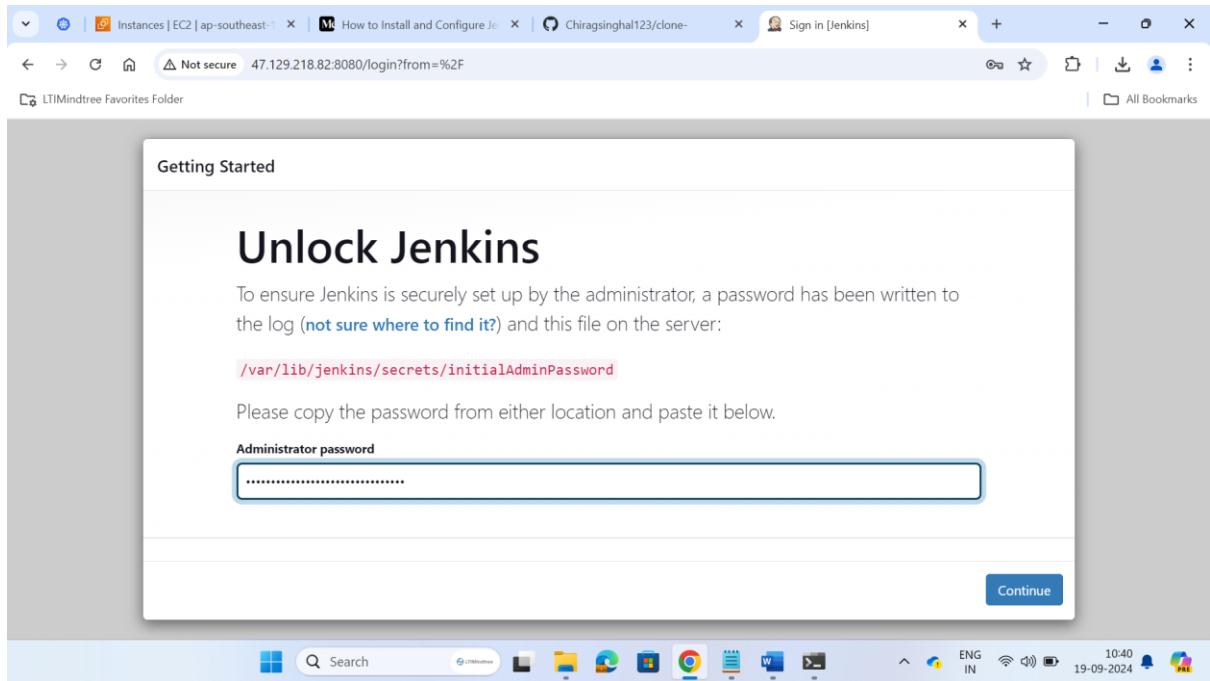


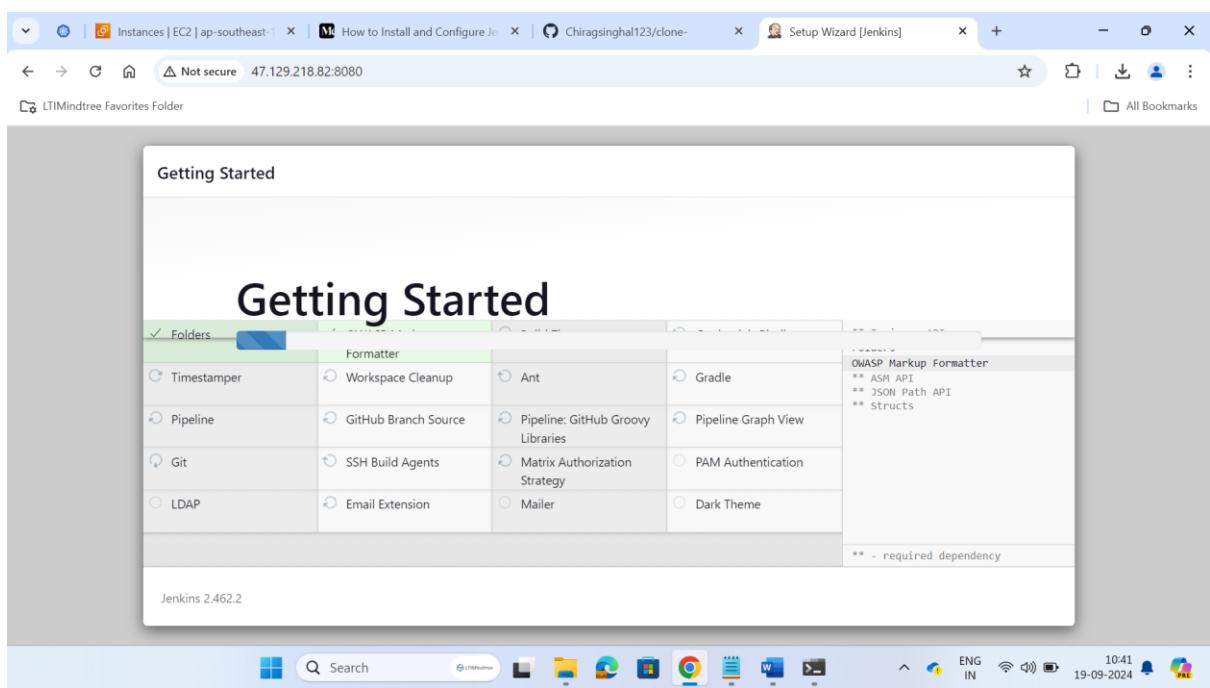
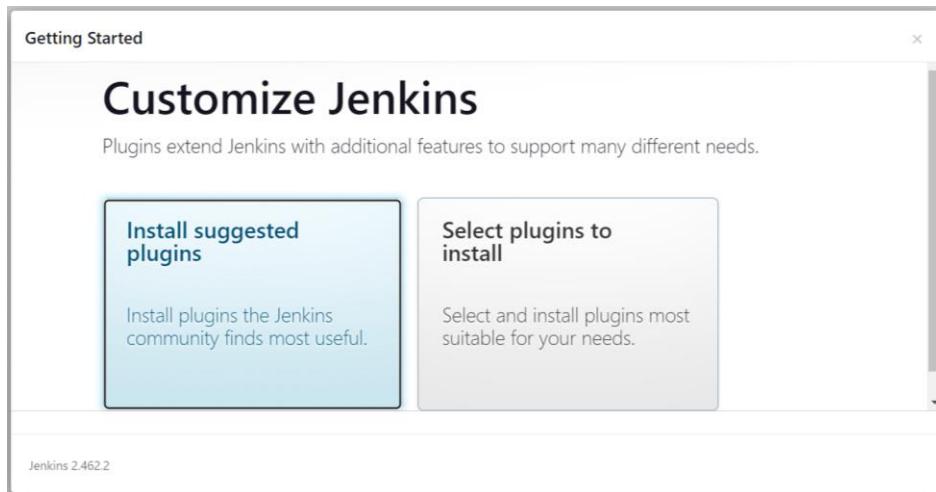
```
[root@jankin ~]# systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@jankin ~]# systemctl start jenkins
[root@jankin ~]# |
```

Paste jankin public ip and add :8080

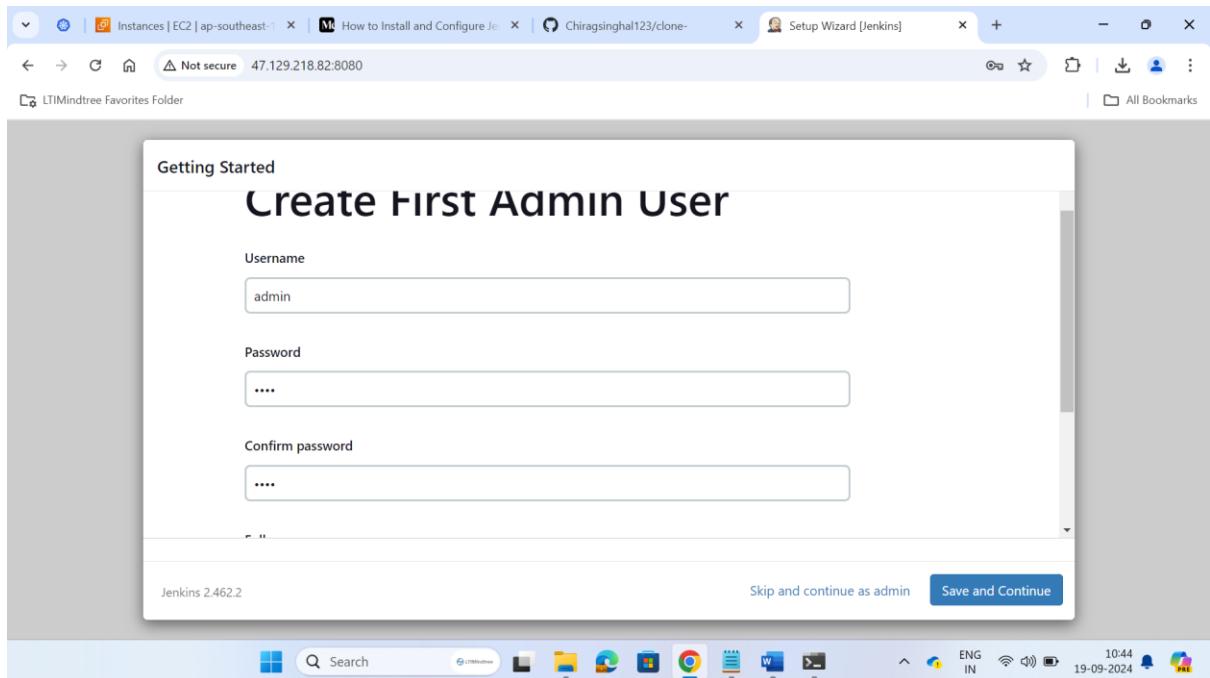


Open Jenkin by password

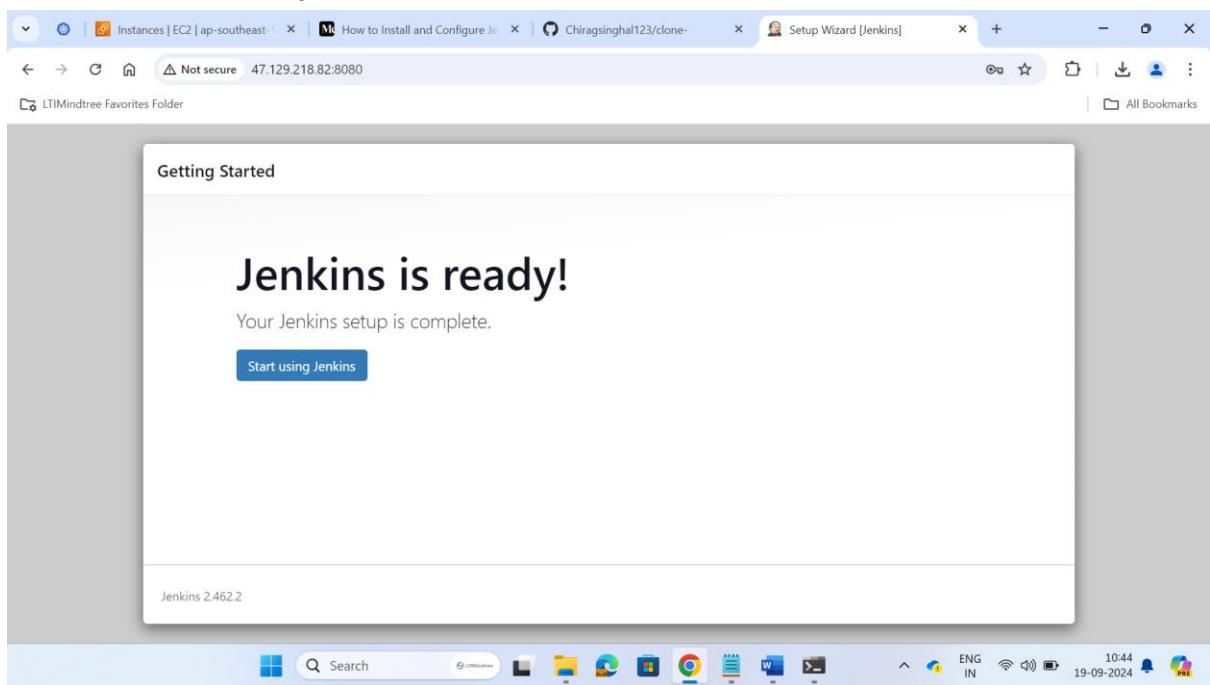




Add username,password and emailid



Jenkins server is ready



In Jenkins server select profile then select Configure and Generate Token then paste in githubs Webhook:

The screenshot shows the Jenkins 'Manage Jenkins' interface. In the top right corner, there is a dropdown menu with several options: 'Builds', 'Configure', 'My Views', and 'Credentials'. The 'Configure' option is highlighted with a light blue background. Below the menu, there is a message about building on a built-in node being a security issue, followed by three buttons: 'Set up agent', 'Set up cloud', and 'Dismiss'. On the left side, there are sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 1 Idle and 2 Idle). The URL in the browser bar is 47.129.218.82:8080/manage/.

The screenshot shows the Jenkins 'User Configuration' page for the user 'chiragsinghal'. At the top, it says 'Dashboard > chiragsinghal > Configure'. Under the 'Configure' section, there is a 'Plain text' area containing the text 'Plain text [Preview](#)'. Below this is the 'API Token' section, which includes a 'Current token(s)' table. The table has one row showing a token created on 2024-09-19T05:15:59.3 with the value 114b4a2d24118e741b39ba7c9d440d3324. A warning message states 'Copy is only supported with a secure (HTTPS) connection'. Below the table, a yellow warning icon says '⚠️ Copy this token now, because it cannot be recovered in the future.' There is also a link 'Add new Token'. At the bottom of the configuration page, there are 'Save' and 'Apply' buttons. The URL in the browser bar is 47.129.218.82:8080/user/admin/configure.

Paste Generate code in secret key

The screenshot shows the GitHub 'Webhooks / Add webhook' configuration page. On the left, a sidebar menu is open with the 'Webhooks' option selected. The main form fields include:

- Payload URL ***: http://47.129.218.82:8080/github-webhook/
- Content type ***: application/json
- Secret**: 114b4a2d24118e741b39ba7c9d440d3324
- SSL verification**: Enable SSL verification Disable (not recommended)
- Which events would you like to trigger this webhook?**: Just the push event

The status message at the bottom of the page reads: "We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation."

webhooks is created Successfully

The screenshot shows the GitHub 'Settings' page for the repository 'Chiragsinghal123 / clone-'. The 'Webhooks' section displays the newly created webhook with the URL http://47.129.218.82:8080/github-webhook/ (push) and a status message indicating 'Last delivery was successful.'

The status message at the top of the settings page reads: "Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at https://docs.github.com/webhooks/#ping-event."

The sidebar on the left includes options like Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.

Open jenkin server click new item then select Freestyle project and check jenkin job is build or not

The screenshot shows the Jenkins 'New Item' configuration page. The job name is 'jenkins-job'. The 'Freestyle project' option is selected, described as a classic general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications. An 'OK' button is visible at the bottom.

The screenshot shows a GitHub repository page for 'clone-' by 'Chiragsinghal123'. The repository contains files such as server, webapp, Dockerfile, README.md, Snjay_Devops_projectc.pdf, and pom.xml. A 'Clone' button is highlighted, showing the HTTPS URL: https://github.com/chiragsinghal123/clone-.git.

In jenkins machine install git

```
PS C:\Users\10747850\Downloads> ssh -i "newkey.pem" ec2-user@ec2-47-129-218-82.ap-southeast-1.compute.amazonaws.com
#_
###_ Amazon Linux 2023
###_ \###_ 
###_ \#_ 
###_ \#/ _-> https://aws.amazon.com/linux/amazon-linux-2023
###_ / 
###_ / 
###_ / 
Last login: Thu Sep 19 05:02:15 2024 from 167.103.3.107
[ec2-user@jankin ~]$ sudo su -
Last login: Thu Sep 19 05:02:18 UTC 2024 on pts/1
[root@jankin ~]# dnf install git
Last metadata expiration check: 0:20:50 ago on Thu Sep 19 05:06:30 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository   Size
=====
Installing:
git              x86_64          2.40.1-1.amzn2023.0.3    amazonlinux 54 k
Installing dependencies:
git-core          x86_64          2.40.1-1.amzn2023.0.3    amazonlinux 4.3 M
git-core-doc      noarch          2.40.1-1.amzn2023.0.3    amazonlinux 2.6 M
perl-Error        noarch          1:0.17029-5.amzn2023.0.2  amazonlinux 41 k
perl-File-Find    noarch          1.37-477.amzn2023.0.6    amazonlinux 26 k
perl-Git          noarch          2.40.1-1.amzn2023.0.3    amazonlinux 42 k
perl-TermReadKey x86_64          2.38-9.amzn2023.0.2     amazonlinux 36 k
perl-lib          x86_64          0.65-477.amzn2023.0.6    amazonlinux 15 k
=====
Transaction Summary
=====
```

Jenkin job is completed

The screenshot shows the Jenkins job 'jenkins-job' dashboard. At the top, there are several browser tabs: 'Instances | EC2 | ap-southeast-1', 'How to Install and Configure Jenkins', 'Chiragsinghal123/clone...', and 'jenkins-job [Jenkins]'. The main content area has a header 'Dashboard > jenkins-job >'. Below this, there are five action buttons: 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', and 'Rename'. A 'Build History' card is displayed, showing one build entry: '#1 | Sep 19, 2024, 5:28 AM' with a status of 'Atom feed for all' and 'Atom feed for failures'. The bottom of the page shows the URL '47.129.218.82:8080/job/jenkins-job/rssFailed' and the Jenkins version 'Jenkins 2.462.2'.

Jenkin Machine Install maven

```
[root@jankin ~]# dnf install maven -y
Last metadata expiration check: 0:25:51 ago on Thu Sep 19 05:06:30 2024.
Dependencies resolved.
=====
 Package           Architecture   Version      Repository  Size
=====
Installing:
 maven             noarch        1:3.8.4-3.amzn2023.0.5    amazonlinux 18 k
Installing dependencies:
 apache-commons-cli noarch        1.5.0-3.amzn2023.0.3    amazonlinux 76 k
 apache-commons-codec noarch        1.15-6.amzn2023.0.3    amazonlinux 303 k
 apache-commons-io   noarch        1:2.8.0-7.amzn2023.0.4    amazonlinux 284 k
 apache-commons-lang3 noarch        3.12.0-7.amzn2023.0.3    amazonlinux 559 k
 atinject           noarch        1.0.5-3.amzn2023.0.3    amazonlinux 23 k
 cdi-api            noarch        2.0.2-6.amzn2023.0.3    amazonlinux 54 k
 google-guice       noarch        4.2.3-8.amzn2023.0.6    amazonlinux 473 k
 guava              noarch        31.0.1-3.amzn2023.0.6    amazonlinux 2.4 M
 httpcomponents-client noarch        4.5.13-4.amzn2023.0.4    amazonlinux 657 k
 httpcomponents-core noarch        4.4.13-6.amzn2023.0.3    amazonlinux 632 k
 jakarta-annotations noarch        1.3.5-13.amzn2023.0.3    amazonlinux 46 k
 jansi               x86_64        2.4.0-3.amzn2023.0.3    amazonlinux 113 k
 java-17-amazon-corretto-devel x86_64        1:17.0.12+7-1.amzn2023.1  amazonlinux 142 k
 jcl-over-slf4j      noarch        1.7.32-3.amzn2023.0.4    amazonlinux 25 k
 jsoup               noarch        1.13.1-9.amzn2023.0.5    amazonlinux 377 k
 jsr-305             noarch        3.0.2-5.amzn2023.0.4    amazonlinux 32 k
 maven-amazon-corretto17 noarch        1:3.8.4-3.amzn2023.0.5    amazonlinux 9.4 k
 maven-lib            noarch        1:3.8.4-3.amzn2023.0.5    amazonlinux 1.5 M
 maven-resolver       noarch        1:1.7.3-3.amzn2023.0.4    amazonlinux 557 k
 maven-shared-utils   noarch        3.3.4-4.amzn2023.0.3    amazonlinux 152 k
 maven-wagon           noarch        3.4.2-6.amzn2023.0.4    amazonlinux 113 k
 plexus-cipher        noarch        1.8-3.amzn2023.0.3    amazonlinux 27 k
 plexus-classworlds   noarch        2.6.0-10.amzn2023.0.4    amazonlinux 61 k
 plexus-containers-component-annotations noarch        2.1.0-9.amzn2023.0.4    amazonlinux 19 k
=====

```

See java version and maven version

```
[root@jankin ~]# mvn -v
Apache Maven 3.8.4 (Red Hat 3.8.4-3.amzn2023.0.5)
Maven home: /usr/share/maven
Java version: 17.0.12, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en, platform encoding: UTF-8
OS name: "Linux", version: "6.1.109-118.189.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jankin ~]#
```

Click manage Jenkins then click plugins and install maven Integration

Instances | EC2 | ap-southeast-1 | How to Install and Configure Jenkins | Available plugins - Plugins [Jen...]

Not secure 47.129.218.82:8080/manage/pluginManager/available

LTMindtree Favorites Folder

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Search (CTRL+K)

Available plugins

Maven Integration 3.23

Build Tools

This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.

1 yr 1 mo ago

Config File Provider 978.v8e85886ffdc4

Groovy-related External Site/Tool Integrations Maven

Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.

9 days 20 hr ago

Jira 3.13

ENG IN 11:08 19-09-2024

Search github and turn off github branch source plugin then automatic turn on GitHub plugin

The screenshot shows the Jenkins Plugin Manager interface. On the left, there's a sidebar with options: Updates, Available plugins, **Installed plugins** (which is selected), and Advanced settings. In the main area, a search bar at the top has "github" typed into it. Below the search bar, there's a button labeled "Enabled". A list of four GitHub-related plugins is shown:

- GitHub API Plugin** 1.321-468.v6a_9f5f2d5a_7e: This plugin provides GitHub API for other plugins. Status: Enabled (green switch).
- GitHub Branch Source Plugin** 1797.v86fdb_4d57d43: Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc. Status: Disabled (gray switch).
- GitHub plugin** 1.40.0: This plugin integrates GitHub to Jenkins. Status: Enabled (green switch).
- Pipeline: GitHub Groovy Libraries** 61.v629f2cc41d83: Allows Pipeline Groovy libraries to be loaded on the fly from GitHub. Status: Enabled (green switch).

At the bottom of the list, there's a warning message: "⚠ Changes will take effect when you restart Jenkins" followed by a "Restart Once No Jobs Are Running" button.

Select New Item click maven project then ok

The screenshot shows the Jenkins "New Item" creation dialog. At the top, there's a search bar with "Search (CTRL+K)" and a user dropdown for "chiragsinghal". Below the search bar, the title "New Item" is displayed. Under "Enter an item name", the text "build maven" is entered. Under "Select an item type", two options are listed:

- Freestyle project**: Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**: Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

A blue "OK" button is located at the bottom of the dialog. The status bar at the bottom of the screen shows the date and time as 19-09-2024 11:13.

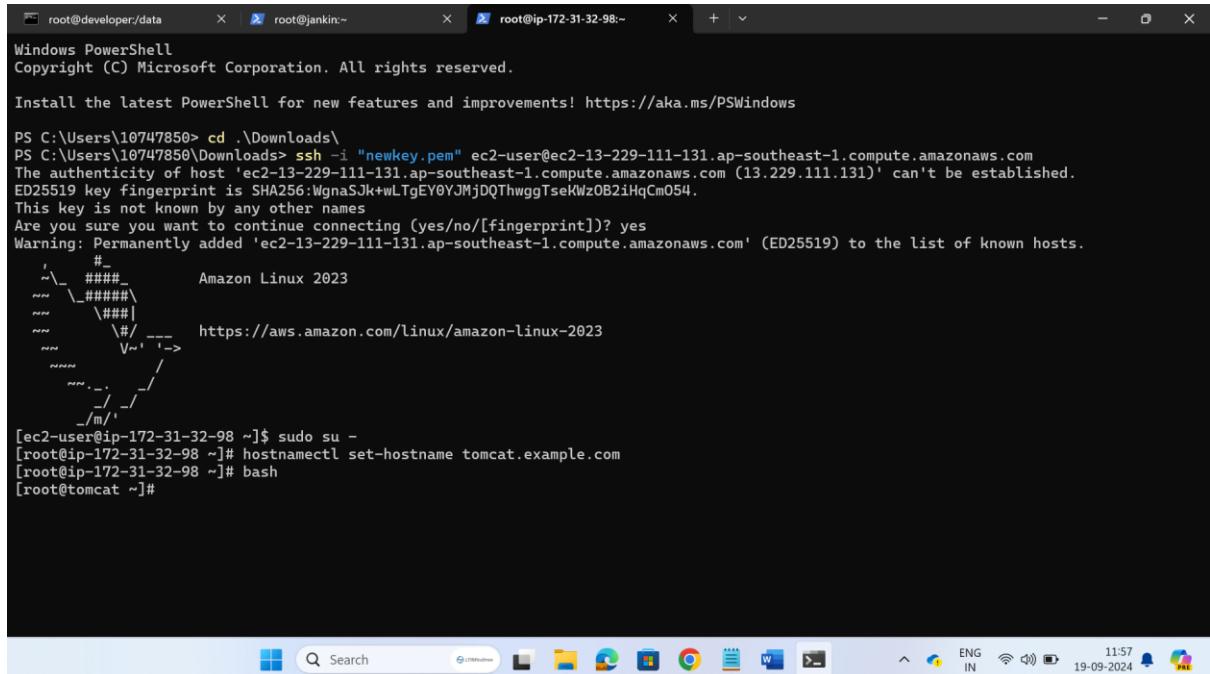
My maven project Is Successfully Build

The screenshot shows a web browser window with the URL 47.129.218.82:8080/job/build%20maven/. The page displays the Jenkins dashboard for the 'build maven' job. Under the 'Permalinks' section, there are links for 'Build Now', 'Configure', 'Delete Maven project', 'Modules', 'GitHub Hook Log', and 'Rename'. Below this is a 'Build History' section titled 'Build #1' (Sep 19, 2024, 5:44 AM). It includes links for 'Atom feed for all' and 'Atom feed for failures'. The system tray at the bottom right shows the date as 19-09-2024 and the time as 11:14.

Show Webapp.war file

The screenshot shows a web browser window with the URL 47.129.218.82:8080/job/build%20maven/ws/webapp/target/. The page displays the Jenkins workspace for the 'build maven' job. The left sidebar includes links for 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Maven project', 'Modules', 'GitHub Hook Log', and 'Rename'. The main content area is titled 'Workspace of build maven on Built-In Node' and shows a directory structure: 'build maven / webapp / target /'. Inside 'target' are 'maven-archiver', 'surefire', 'webapp', and 'webapp.war'. A download link '(all files in zip)' is available. The system tray at the bottom right shows the date as 19-09-2024 and the time as 11:18.

Start Tomcat Machine



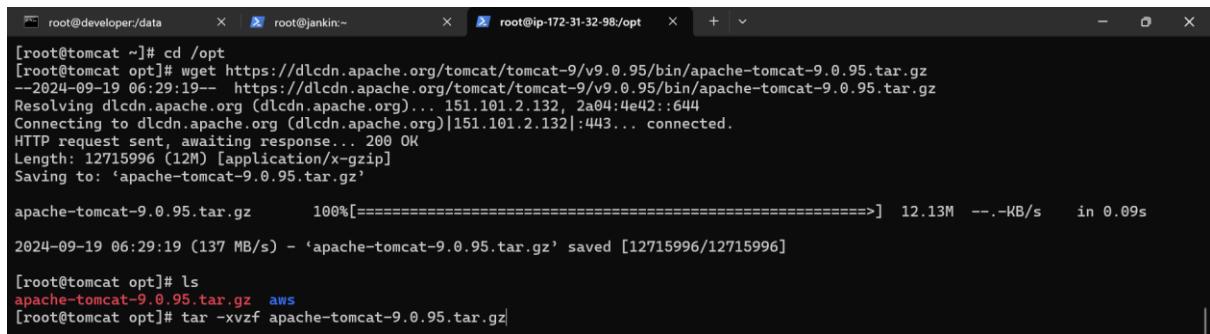
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\107477850> cd .\Downloads\
PS C:\Users\107477850\Downloads> ssh -i "newkey.pem" ec2-user@ec2-13-229-111-131.ap-southeast-1.compute.amazonaws.com
The authenticity of host 'ec2-13-229-111-131.ap-southeast-1.compute.amazonaws.com (13.229.111.131)' can't be established.
ED25519 key fingerprint is SHA256:WgnaSJk+wLTgEV0YJMjDQThwggtSeKwzOB2iHqCm054.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-229-111-131.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_\_ #####      Amazon Linux 2023
~~ \_\#\#\#\_
~~ \#\#\#
~~ \#/ ___  https://aws.amazon.com/linux/amazon-linux-2023
~~ \~' '->
~~ /_/
~~ .-' _/-
~/_/
~/m/'

[ec2-user@ip-172-31-32-98 ~]$ sudo su -
[root@ip-172-31-32-98 ~]# hostnamectl set-hostname tomcat.example.com
[root@ip-172-31-32-98 ~]# bash
[root@tomcat ~]#
```



```
[root@tomcat ~]# cd /opt
[root@tomcat opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.95/bin/apache-tomcat-9.0.95.tar.gz
--2024-09-19 06:29:19-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.95/bin/apache-tomcat-9.0.95.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12715996 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.95.tar.gz'

apache-tomcat-9.0.95.tar.gz      100%[=====] 12.13M --.-KB/s   in 0.09s

2024-09-19 06:29:19 (137 MB/s) - 'apache-tomcat-9.0.95.tar.gz' saved [12715996/12715996]

[root@tomcat opt]# ls
apache-tomcat-9.0.95.tar.gz  aws
[root@tomcat opt]# tar -xvzf apache-tomcat-9.0.95.tar.gz
```

```
root@developer:~/data          X  root@jenkin:~          X  root@ip-172-31-32-98:/opt          X  +  v
apache-tomcat-9.0.95/webapps/host-manager/WEB-INF/web.xml
apache-tomcat-9.0.95/webapps/host-manager/css/manager.css
apache-tomcat-9.0.95/webapps/host-manager/images/asf-logo.svg
apache-tomcat-9.0.95/webapps/host-manager/images/tomcat.svg
apache-tomcat-9.0.95/webapps/host-manager/index.jsp
apache-tomcat-9.0.95/webapps/manager/META-INF/context.xml
apache-tomcat-9.0.95/webapps/manager/WEB-INF/jsp/401.jsp
apache-tomcat-9.0.95/webapps/manager/WEB-INF/jsp/403.jsp
apache-tomcat-9.0.95/webapps/manager/WEB-INF/jsp/404.jsp
apache-tomcat-9.0.95/webapps/manager/WEB-INF/jsp/connectorCerts.jsp
apache-tomcat-9.0.95/webapps/manager/WEB-INF/jsp/connectorCiphers.jsp
apache-tomcat-9.0.95/webapps/manager/WEB-INF/jsp/connectorTrustedCerts.jsp
apache-tomcat-9.0.95/webapps/manager/WEB-INF/jsp/sessionDetail.jsp
apache-tomcat-9.0.95/webapps/manager/WEB-INF/jsp/sessionsList.jsp
apache-tomcat-9.0.95/webapps/manager/META-INF/web.xml
apache-tomcat-9.0.95/webapps/manager/css/manager.css
apache-tomcat-9.0.95/webapps/manager/images/asf-logo.svg
apache-tomcat-9.0.95/webapps/manager/images/tomcat.svg
apache-tomcat-9.0.95/webapps/manager/index.jsp
apache-tomcat-9.0.95/webapps/manager/status.xsd
apache-tomcat-9.0.95/webapps/manager/xform.xsl
apache-tomcat-9.0.95/bin/catalina.sh
apache-tomcat-9.0.95/bin/ciphers.sh
apache-tomcat-9.0.95/bin/configtest.sh
apache-tomcat-9.0.95/bin/daemon.sh
apache-tomcat-9.0.95/bin/digest.sh
apache-tomcat-9.0.95/bin/makebase.sh
apache-tomcat-9.0.95/bin/setclasspath.sh
apache-tomcat-9.0.95/bin/shutdown.sh
apache-tomcat-9.0.95/bin/startup.sh
apache-tomcat-9.0.95/bin/tool-wrapper.sh
apache-tomcat-9.0.95/bin/version.sh
[root@tomcat opt]# |
```



```
apache-tomcat-9.0.95/bin/version.sh
[root@tomcat opt]# ls
apache-tomcat-9.0.95 apache-tomcat-9.0.95.tar.gz aws
[root@tomcat opt]# mv apache-tomcat-9.0.95 tomcat
[root@tomcat opt]# ls
apache-tomcat-9.0.95.tar.gz aws tomcat
[root@tomcat opt]# |
```



```
root@developer:~/data          X  root@jenkin:~          X  root@ip-172-31-32-98:/opt/tc          X  +  v
[root@tomcat opt]# cd tomcat
[root@tomcat tomcat]# cd bin
[root@tomcat bin]# ls
bootstrap.jar      ciphers.sh           daemon.sh      setclasspath.bat  startup.sh        version.bat
catalina-tasks.xml commons-daemon-native.tar.gz digest.bat    setclasspath.sh  tomcat-juli.jar   version.sh
catalina.bat       commons-daemon.jar   digest.sh     shutdown.bat    tomcat-native.tar.gz
catalina.sh        configtest.bat      makebase.bat  shutdown.sh    tool-wrapper.bat
ciphers.bat        configtest.sh       makebase.sh   startup.bat   tool-wrapper.sh
[root@tomcat bin]# ./startup.sh
Neither the JAVA_HOME nor the JRE_HOME environment variable is defined
At least one of these environment variable is needed to run this program
[root@tomcat bin]# |
```

Copy Tomcat Public key and Port 8080

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Developer Quick Start

- [Tomcat Setup](#)
- [First Web Application](#)
- [Realms & AAA](#)
- [JDBC DataSources](#)
- [Examples](#)
- [Servlet Specifications](#)
- [Tomcat Versions](#)

Managing Tomcat

Documentation

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

- [tomcat-announce](#)
- [Important announcements, releases, security vulnerabilities, notifications, etc.](#)

```
java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
Sep 19, 2024 6:36:05 AM org.apache.catalina.startup.Catalina stopServer
SEVERE: Could not contact [localhost:8005] (base port [8005] and offset [0]). Tomcat may not be running.
Sep 19, 2024 6:36:05 AM org.apache.catalina.startup.Catalina stopServer
SEVERE: Error stopping Catalina
java.net.ConnectException: Connection refused
    at java.base/sun.nio.ch.Net.connect0(Native Method)
    at java.base/sun.nio.ch.Net.connect(Net.java:579)
    at java.base/sun.nio.ch.Net.connect(Net.java:568)
    at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:593)
    at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:327)
    at java.base/java.net.Socket.connect(Socket.java:633)
    at java.base/java.net.Socket.connect(Socket.java:583)
    at java.base/java.net.Socket.<init>(Socket.java:507)
    at java.base/java.net.Socket.<init>(Socket.java:287)
    at org.apache.catalina.startup.Catalina.stopServer(Catalina.java:629)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.base/java.lang.reflect.Method.invoke(Method.java:569)
    at org.apache.catalina.startup.Bootstrap.stopServer(Bootstrap.java:393)
    at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:478)

[root@tomcat bin]# ./startup.sh
Using CATALINA_BASE:  /opt/tomcat
Using CATALINA_HOME:  /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@tomcat bin]#
```

```
[root@tomcat bin]# ./startup.sh
Using CATALINA_BASE:  /opt/tomcat
Using CATALINA_HOME:  /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started
[root@tomcat bin]# cd ..
[root@tomcat tomcat]# ll
total 188
-rw-r-----. 1 root root 20913 Sep 13 18:07 BUILDING.txt
-rw-r-----. 1 root root 6166 Sep 13 18:07 CONTRIBUTING.md
-rw-r-----. 1 root root 57092 Sep 13 18:07 LICENSE
-rw-r-----. 1 root root 2333 Sep 13 18:07 NOTICE
-rw-r-----. 1 root root 3283 Sep 13 18:07 README.md
-rw-r-----. 1 root root 6901 Sep 13 18:07 RELEASE-NOTES
-rw-r-----. 1 root root 16538 Sep 13 18:07 RUNNING.txt
drwxr-x---. 2 root root 16384 Sep 19 06:30 bin
drwxr-----. 3 root root 16384 Sep 19 06:36 conf
drwxr-x---. 2 root root 16384 Sep 19 06:30 lib
drwxr-x---. 2 root root 16384 Sep 19 06:36 logs
drwxr-x---. 2 root root 30 Sep 19 06:30 temp
drwxr-x---. 7 root root 81 Sep 13 18:07 webapps
drwxr-x---. 3 root root 22 Sep 19 06:36 work
[root@tomcat tomcat]# find / -name context.xml
/opt/tomcat/conf/context.xml
/opt/tomcat/webapps/docs/META-INF/context.xml
/opt/tomcat/webapps/examples/META-INF/context.xml
/opt/tomcat/webapps/host-manager/META-INF/context.xml
/opt/tomcat/webapps/manager/META-INF/context.xml
[root@tomcat tomcat]#
```

Comment last two lines

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at
    http://www.apache.org/licenses/LICENSE-2.0

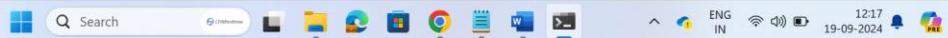
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
    <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
        sameSiteCookies="strict" />
    <!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\\.\\d+\\.\\d+\\.\\d+|1[0:0:0:0:0:1]" />!-->
    <Manager sessionAttributeValueClassNameFilter="java\\.lang\\.\\{Boolean|Integer|Long|Number|String}|org\\.apache\\.catalina\\.filters\\.C
sr|PreventionFilter\\$LruCache\\{\\$1}|java\\.util\\.\\{\\Linked\\}HashMap"/>
</Context>
~
~
~
~
~
~
-- INSERT --
```

Comment last two lines

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
<CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
<!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allows="127\\.\\d+\\.\\d+\\.\\d+::1|0:0:0:0:0:1" /-->
<Manager sessionAttributeValueClassNameFilter="java\\.lang\\.\\{Boolean|Integer|Long|Number|String}|org\\.apache\\.catalina\\.filters\\.C
srfPreventionFilter\\$LruCache\\{\\$1}\\|java\\.util\\.\\{Linked\\}HashMap"/>
</Context>
~
~
~
~
~
~
-- INSERT --
```

22,53 All



Open tomcat-users.xml file and Changes this file

```
[root@tomcat tomcat]# cd conf/
[root@tomcat conf]# ls
Catalina      catalina.properties  jaspic-providers.xml  logging.properties  tomcat-users.xml  web.xml
catalina.policy  context.xml       jaspic-providers.xsd   server.xml        tomcat-users.xsd
[root@tomcat conf]# vim tomcat-users.xml
```

Change last six lines then save this file

```
root@developer:~/data      X  root@jenkin:~      X  root@ip-172-31-32-98:/opt/tc  X  +  v
- manager-gui - allows access to the HTML GUI and the status pages
- manager-script - allows access to the HTTP API and the status pages
- manager-jmx - allows access to the JMX proxy and the status pages
- manager-status - allows access to the status pages only

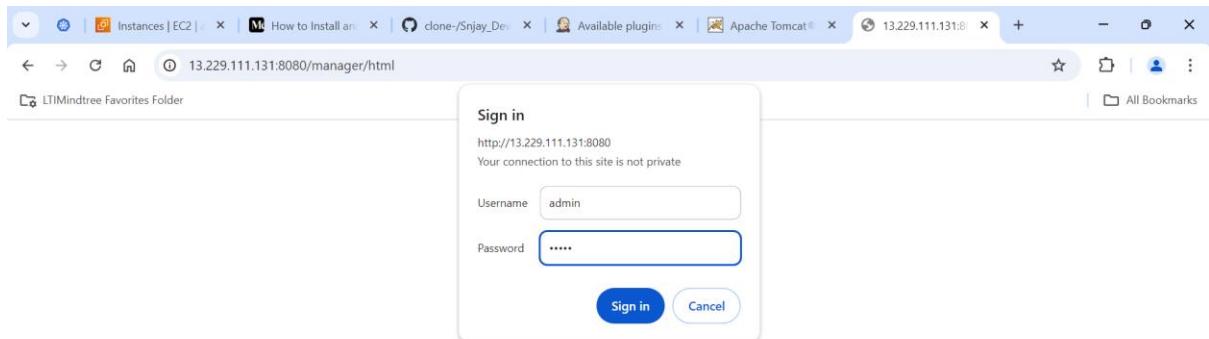
The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <!...> that surrounds them. You
will also need to set the passwords to something appropriate.
-->
<!--
<user username="admin" password="" roles="manager-gui"/>
<user username="robot" password="" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!...> that surrounds
them. You will also need to set the passwords to something appropriate.
-->

<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<user username="admin" password="admin" roles="manager-gui,manager-script,manager-jmx,manager-status"/>
<user username="deployer" password="deployer" roles="manager-script"/>
<user username="tomcat" password="tomcat" roles="manager-gui"/>

</tomcat-users>
:wq!
```

```
root@developer:~/data      X  root@jenkin:~      X  root@ip-172-31-32-98:/opt/tc  X  +  v
[root@tomcat tomcat]# cd conf/
[root@tomcat conf]# ls
Catalina      catalina.properties  jaspic-providers.xml  logging.properties  tomcat-users.xml  web.xml
catalina.policy  context.xml        jaspic-providers.xsd   server.xml          tomcat-users.xsd
[root@tomcat conf]# vim tomcat-users.xml
[root@tomcat conf]# cd bin
bash: cd: bin: No such file or directory
[root@tomcat conf]# cd ..
[root@tomcat tomcat]# cd bin
[root@tomcat bin]# ./shutdown.sh
Using CATALINA_BASE:  /opt/tomcat
Using CATALINA_HOME:  /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
[root@tomcat bin]# ./startup.sh
Using CATALINA_BASE:  /opt/tomcat
Using CATALINA_HOME:  /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@tomcat bin]# |
```

Open Tomcat Server: set username and password by using file



Show this page

The screenshot shows a web browser window with the URL `13.229.111.131:8080/manager`. The title bar indicates the page is not secure. The main content area displays the "Manager" interface. The "Applications" section lists the following deployed applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

The "Deploy" section at the bottom allows for deploying a directory or WAR file located on the server. The browser's title bar shows the URL `/manager`.

Install Deploy to Container by using Tomcat

The screenshot shows the Jenkins Plugins page. On the left, there's a sidebar with 'Updates', 'Available plugins' (which is selected), 'Installed plugins', and 'Advanced settings'. The main area has a search bar with 'deploy to con'. A table lists the 'Deploy to container' plugin, version 1.16, released 3 years and 10 months ago. It's described as allowing deployment of a war to a container after a successful build, specifically for Glassfish 3.x remote deployment. An 'Install' button is visible.

Goto Dashboard > Manage Jenkins > Credentials > System > Global Credentials then click new credentials

The screenshot shows the 'New credentials' page. Under 'Kind', 'Username with password' is selected. Under 'Scope', 'Global (Jenkins, nodes, items, all child items, etc)' is chosen. There's a 'Username' field with a note 'Treat username as secret'. At the bottom is a 'Create' button.

Instances | EC2 | How to Install an | clone-/Snjay_Dev | Apache Tomcat | /manager | New credentials | +

Not secure ec2-47-129-218-82.ap-southeast-1.compute.amazonaws.com:8080/manage/credentials/store/system/domain/_/newCredentials

LTMindtree Favorites Folder All Bookmarks

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: deployer

Treat username as secret

Password:

Create

Search bar, taskbar, system tray showing ENG IN, 12:39, 19-09-2024, and user chiragsinghal.

Add Credentials

Instances | EC2 | How to Install an | clone-/Snjay_Dev | Apache Tomcat | /manager | System » Global | +

Not secure ec2-47-129-218-82.ap-southeast-1.compute.amazonaws.com:8080/manage/credentials/store/system/domain/_/

LTMindtree Favorites Folder All Bookmarks

Jenkins Search (CTRL+K) chiragsinghal log out

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted) + Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
deployer	deployer/*****	Username with password	

Icon: S M L



Click New Item and build Tomcat Project select Maven project then ok

The screenshot shows the Jenkins 'New Item' creation interface. At the top, there is a header bar with several tabs and icons. Below the header, the URL in the address bar is `ec2-47-129-218-82.ap-southeast-1.compute.amazonaws.com:8080/view/all/newJob`. The main content area is titled 'New Item'. It has two sections: 'Enter an item name' (with 'Tomcat-project' typed into the input field) and 'Select an item type'. Under 'Select an item type', three options are listed: 'Freestyle project', 'Maven project' (which is selected), and 'Pipeline'. A large blue 'OK' button is located at the bottom of this section. The status bar at the bottom of the browser window shows the date and time as 19-09-2024 12:43.

Add GitHub url < Add Credentials

The screenshot shows the Jenkins 'Configure' screen for the 'Tomcat-project' job. The left sidebar lists various configuration categories: General, Source Code Management (which is selected and highlighted in grey), Build Triggers, Build Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The main panel is titled 'Repositories' and contains a form for adding a new repository. The 'Repository URL' field is filled with `https://github.com/Chiragsinghal123/clone-.git`. The 'Credentials' dropdown menu is open, showing the option 'deployer/*****'. There is also an 'Advanced' dropdown and a 'Save' button at the bottom of the form. The status bar at the bottom of the browser window shows the date and time as 19-09-2024 12:45.

The screenshot shows a web browser window with the URL `ec2-47-129-218-82.ap-southeast-1.compute.amazonaws.com:8080/job/Tomcat-project/configure`. The page is titled 'Configure' under 'Post-build Actions'. On the left, a sidebar lists various build steps: General, Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, **Build Settings** (which is selected), and Post-build Actions. The main content area contains a 'Deploy war/ear to a container' section. It includes fields for 'WAR/EAR files' (containing `/*.war`) and 'Context path'. Below these are sections for 'Containers' and 'Advanced'. At the bottom are 'Save' and 'Apply' buttons.

Build Tomcat Project Sucessfully

The screenshot shows a web browser window with the URL `ec2-47-129-218-82.ap-southeast-1.compute.amazonaws.com:8080/job/Tomcat-project/`. The page is titled 'Dashboard > Tomcat-project >'. On the left, a sidebar lists actions: Build Now, Configure, Delete Maven project, Modules, GitHub Hook Log, and Rename. The main content area is titled 'Build History' and shows a single entry: '#1 Sep 19, 2024, 7:18 AM'. At the bottom are links for 'Atom feed for all' and 'Atom feed for failures'. The status bar at the bottom right shows the date as 19-09-2024 and the time as 12:49.

Webapp Application is Visible in Tomcat Server

The screenshot shows the Apache Tomcat Manager interface at 13.229.111.131:8080/manager/html. The left sidebar lists deployed applications:

Application Name	Context Path	Description	Status	Actions
/docs	None specified	Tomcat Documentation	true	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/webapp	None specified	Webapp	true	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy
Deploy directory or WAR file located on server

Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path: Deploy

WAR file to deploy
Select WAR file to upload No file chosen

The screenshot shows a browser window at 13.229.111.131:8080/webapp/. The title bar says "New user Register for DevOps Learning at Virtual TechBox Youtube Channel".

Please fill in this form to create an account.

Enter Name Enter Full Name
Enter mobile Enter mobile number
Enter Email Enter Email
Password Enter Password
Repeat Password Repeat Password

By creating an account you agree to our [Terms & Privacy](#).

Already have an account? [Sign in](#).

Thank You, Happy Learning

See You Again



Click Manage Jenkin install publish over SSH by using Docker

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Search: publish

A simple Publish-Subscribe light-weight event bus for Jenkins

Infrastructure plugin for Publish Over X 0.22

Send build artifacts somewhere.

6 yr 5 mo ago

Publish Over SSH 1.25

Artifact Uploaders Build Tools

Send build artifacts over SSH

1 yr 2 mo ago

Slack Notification 741.v00f9591c586d

slack Build Notifiers

Integrates Jenkins with Slack, allows publishing build statuses, messages and files to Slack channels.

21 days ago

Email Extension Template 1.5

Build Notifiers emalext

This plugin allows administrators to create global templates for the Extended Email Publisher.

1 yr 11 mo ago

Launch instance Docker

Launch an instance

EC2 > Instances > Launch an instance

Name and tags

Name: docker

Add additional tags

Application and OS Images (Amazon Machine Image)

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

Software Image (AMI)

Amazon Linux 2023 AMI 2023.5.2...read more

ami-0aa097a5cd31450a

Virtual server type (instance type)

t2.medium

Firewall (security group)

New security group

Summary

Number of instances: 1

Cancel

Launch instance

Review commands

Select t2medium because the minimum requirement to run Docker is of 2 core CPU.

The screenshot shows the AWS Lambda console interface. A modal window is open for creating a new function. In the 'Function name' field, 'DockerTest' is entered. Under 'Runtime', 'AWS Lambda' is selected. In the 'Code' section, 'Upload a ZIP file' is chosen, and a local file named 'DockerTest.zip' is selected. The 'Handler' dropdown shows 'lambda_function.lambda_handler'. The 'Memory size' dropdown is set to '128 MB'. The 'Timeout' dropdown is set to '3 seconds'. The 'Role' dropdown is set to 'Lambda execution role'. The 'Environment variables' section is empty. The 'Advanced settings' section includes 'Environment' (set to 'Production'), 'Tracing' (set to 'None'), and 'VPC configuration' (set to 'None'). The 'Configure triggers' section is collapsed. At the bottom right of the modal, there is a large orange 'Create Function' button.

Docker machine is ready and change host name

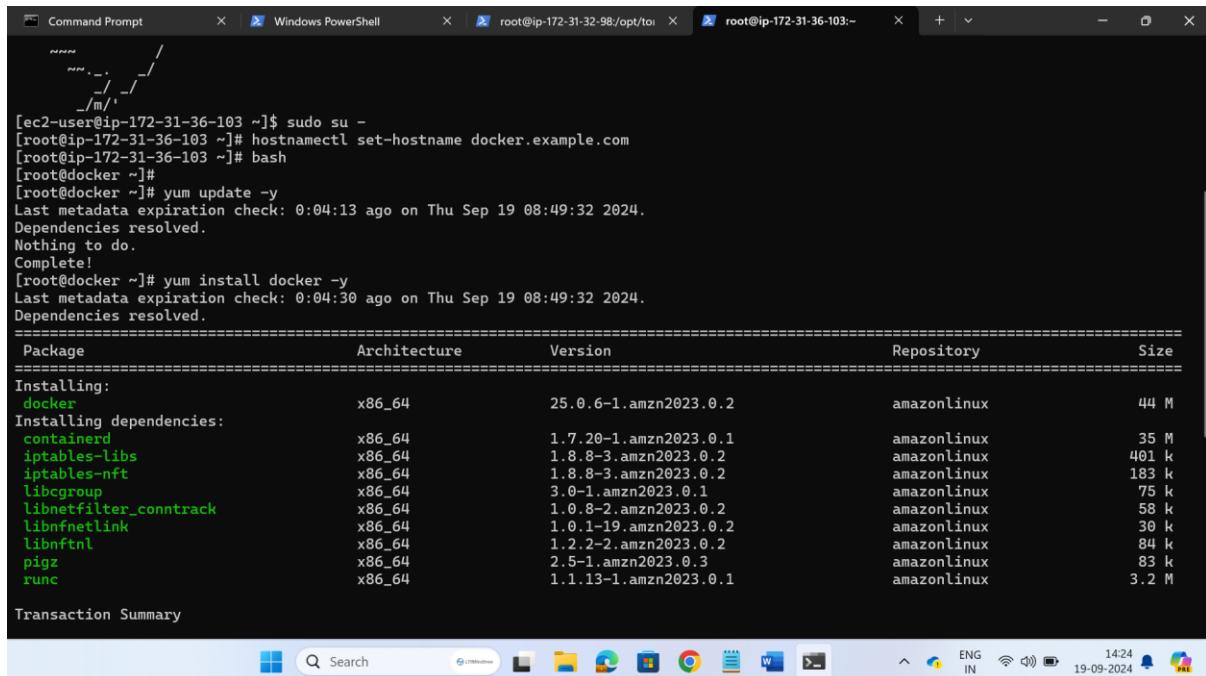
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\107447850> cd .\Downloads\
PS C:\Users\107447850\Downloads> ssh -i "newkey.pem" ec2-user@ec2-13-250-118-126.ap-southeast-1.compute.amazonaws.com
The authenticity of host 'ec2-13-250-118-126.ap-southeast-1.compute.amazonaws.com (13.250.118.126)' can't be established.
ED25519 key fingerprint is SHA256:zpXCV/yelNjTlna/VuLZBAvU07xQyTateJZ1lMiamqT0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-250-118-126.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

#_ _\_\_###_ Amazon Linux 2023
~~ \_\#\#\#\_
~~ \#\#\#
~~ \#/ _--> https://aws.amazon.com/linux/amazon-linux-2023
~~ ._. / /
~~ /_/
~/m'/[ec2-user@ip-172-31-36-103 ~]$ sudo su -
[root@ip-172-31-36-103 ~]# hostnamectl set-hostname docker.example.com
[root@ip-172-31-36-103 ~]# bash
[root@docker ~]#
```

Docker Machine install Docker



```
[ec2-user@ip-172-31-36-103 ~]$ sudo su -
[root@ip-172-31-36-103 ~]# hostnamectl set-hostname docker.example.com
[root@ip-172-31-36-103 ~]# bash
[root@docker ~]# yum update -y
Last metadata expiration check: 0:04:13 ago on Thu Sep 19 08:49:32 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@docker ~]# yum install docker -y
Last metadata expiration check: 0:04:30 ago on Thu Sep 19 08:49:32 2024.
Dependencies resolved.
=====
Package           Architecture   Version      Repository  Size
=====
Installing:
 docker           x86_64        25.0.6-1.amzn2023.0.2  amazonlinux  44 M
Installing dependencies:
 containerd       x86_64        1.7.20-1.amzn2023.0.1  amazonlinux  35 M
 iptables-libc    x86_64        1.8.8-3.amzn2023.0.2  amazonlinux  401 k
 iptables-nft     x86_64        1.8.8-3.amzn2023.0.2  amazonlinux  183 k
 libcgroup        x86_64        3.0-1.amzn2023.0.1   amazonlinux  75 k
 libnetfilter_conntrack x86_64        1.0.8-2.amzn2023.0.2  amazonlinux  58 k
 libnfnetwork     x86_64        1.0.1-19.amzn2023.0.2  amazonlinux  30 k
 libnftnl         x86_64        1.2.2-2.amzn2023.0.2  amazonlinux  84 k
 pigz             x86_64        2.5-1.amzn2023.0.3   amazonlinux  83 k
 runc             x86_64        1.1.13-1.amzn2023.0.1  amazonlinux  3.2 M
=====
Transaction Summary
=====
 0 packagesダウンロードed, 0 packagesアップグレードed, 0 packages新規インストール
 0 packages削除, 0 ファイルを残す
 Total download size: 44 M
ダウンロード: 44 M/s | 44 M 00:00:01
=====
[root@docker ~]#
```

Docker service start

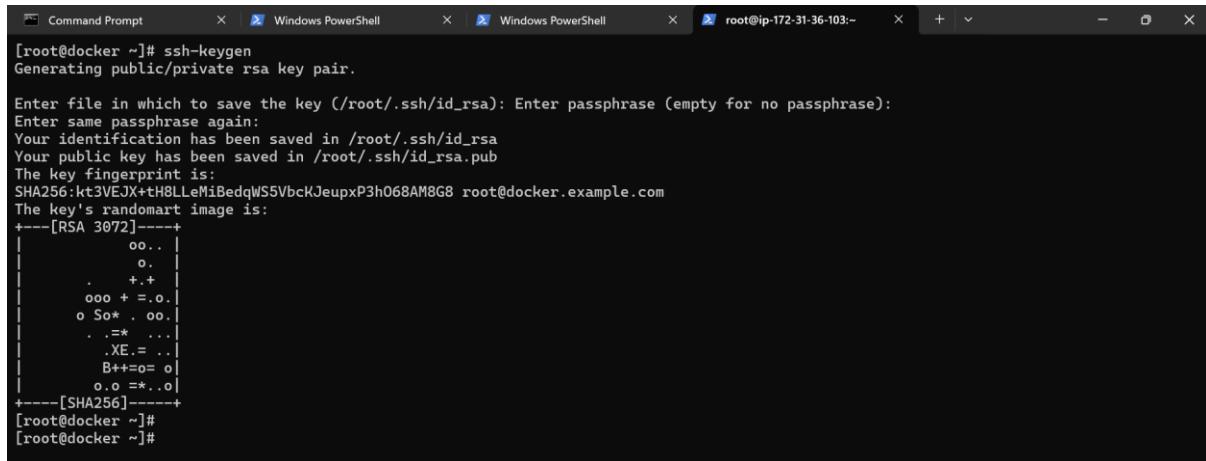


```
complete!
[root@docker ~]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@docker ~]#
```



```
[root@docker ~]# docker ps
CONTAINER ID   IMAGE    COMMAND   CREATED   STATUS    PORTS      NAMES
[root@docker ~]#
```

Create ssh-keygen



```
[root@docker ~]# ssh-keygen
Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa): Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kt3VEJX+H8LLeMiBedqWSSVbcKJeupxP3h068AM8G8 root@docker.example.com
The key's randomart image is:
+---[RSA 3072]---+
| oo.. |
| o. |
| . +. |
| ooo += =o. |
| o So* . oo. |
| . .*= ... |
| .XE.= .. |
| B++=o= o |
| o.o ==...o |
+---[SHA256]---+
[root@docker ~]#
[root@docker ~]#
```

Generate user key and Secret access key:

A screenshot of a Microsoft Excel spreadsheet titled "keychirag_accessKeys". The spreadsheet contains two rows of data:

Access key	Secret access key
AKIAYEKP5T7B13ZDAMH	32NTkQB0917MUNT4Jb+zniX2SB7oHMipDCzxUNND

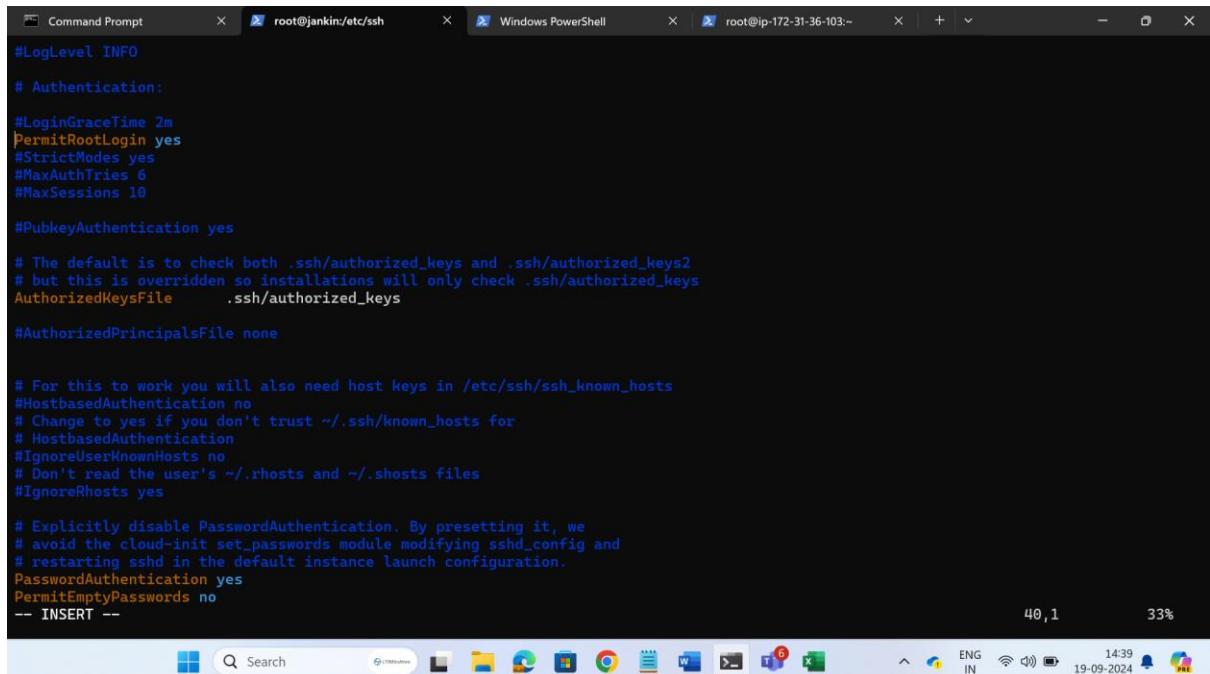
The status bar at the bottom of the Excel window shows "keychirag_accessKeys" and "1 item selected 99 bytes". The taskbar below shows various open windows including "Ansible-SANJAY-chapter2", "Ansible", "keychirag.accessKeys", and "pairchirag.pem".

Paste Access key ,Secret Access Key and Region name :

```
[root@docker ~]# ssh-keygen
Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa): Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kt3VEJX+kh8LLeMiBedqW55VbcKJeupxP3h068AM8G8 root@docker.example.com
The key's randomart image is:
+---[RSA 3072]---+
| oo... |
| o.   |
| . +.  |
| ooo + =.o.|
| o So* . oo.|
| . .*= ...|
| .XE. = ..|
| B+=o= o|
| o.o ==..o|
+---[SHA256]---+
[root@docker ~]#
[root@docker ~]# aws configure
AWS Access Key ID [None]: AKIAYEKP5T7B13ZDAMH
AWS Secret Access Key [None]: 32NTkQB0917MUNT4Jb+zniX2SB7oHMipDCzxUNND
Default region name [None]:
Default output format [None]:
[root@docker ~]# aws configure
AWS Access Key ID [*****DAMH]:
AWS Secret Access Key [*****UNND]:
Default region name [None]: ap-southeast-1
Default output format [None]:
[root@docker ~]# |
```

Command : vim /etc/ssh/sshd_config [allow permission in Jenkin machine]



```
#LogLevel INFO
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

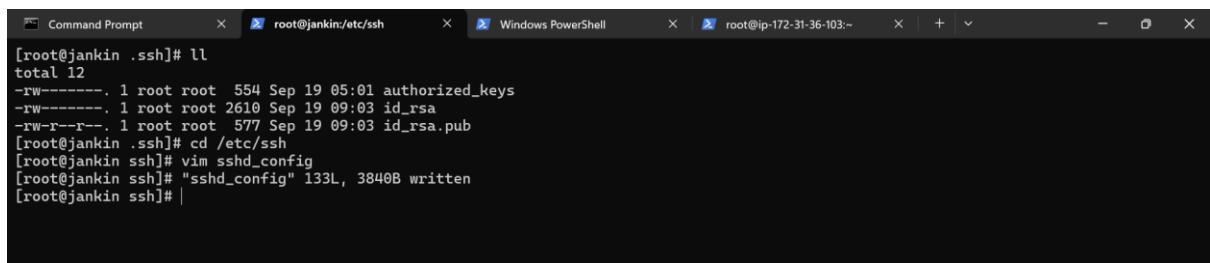
#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

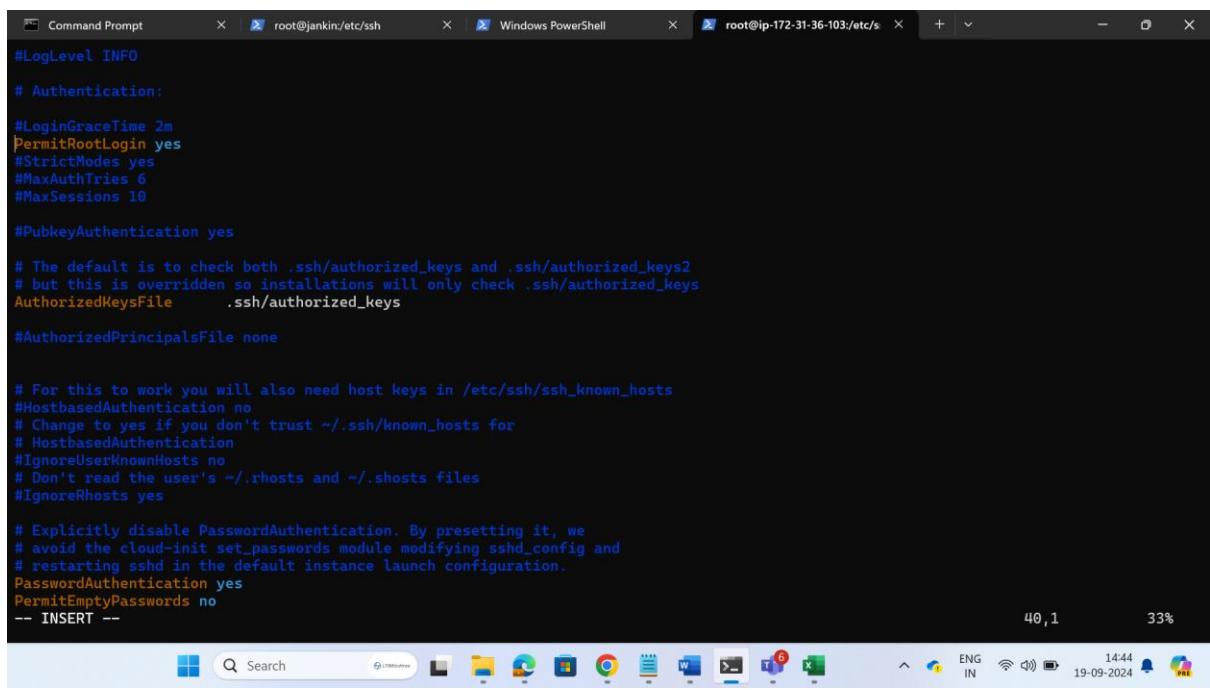
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no
-- INSERT --
```



```
[root@jankin .ssh]# ll
total 12
-rw-----. 1 root root 554 Sep 19 05:01 authorized_keys
-rw-----. 1 root root 2610 Sep 19 09:03 id_rsa
-rw-r--r--. 1 root root 577 Sep 19 09:03 id_rsa.pub
[root@jankin .ssh]# cd /etc/ssh
[root@jankin ssh]# vim sshd_config
[root@jankin ssh]# "sshd_config" 133L, 3840B written
[root@jankin ssh]# |
```

Allow Permission in Docker Machine



```
#LogLevel INFO
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no
-- INSERT --
```

Aws Configure in Docker Machine:



```
AWS Access Key ID [None]: AKIAYEKP5TZ7BI3ZDAMH
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
[root@docker ~]# aws configure
AWS Access Key ID [*****DAMH*]:
AWS Secret Access Key [*****UNND*]:
Default region name [None]: ap-southeast-1
Default output format [None]:
[root@docker ~]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host noprefixroute
                valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 06:74:92:23:03:27 brd ff:ff:ff:ff:ff:ff
        altname eni-089d583783ad7214f
        altnumber 0
        inet 172.31.36.103/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
            valid_lft 2227sec preferred_lft 2227sec
            inet6 fe80::474:92ff:fe23:327/64 scope link
                valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:84:09:6a:9a brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
[root@docker ~]# cd /etc/ssh
[root@docker ssh]# vim sshd_config
"sshd_config" 133L, 3840B written
[root@docker ssh]# systemctl restart sshd
[root@docker ssh]#
```

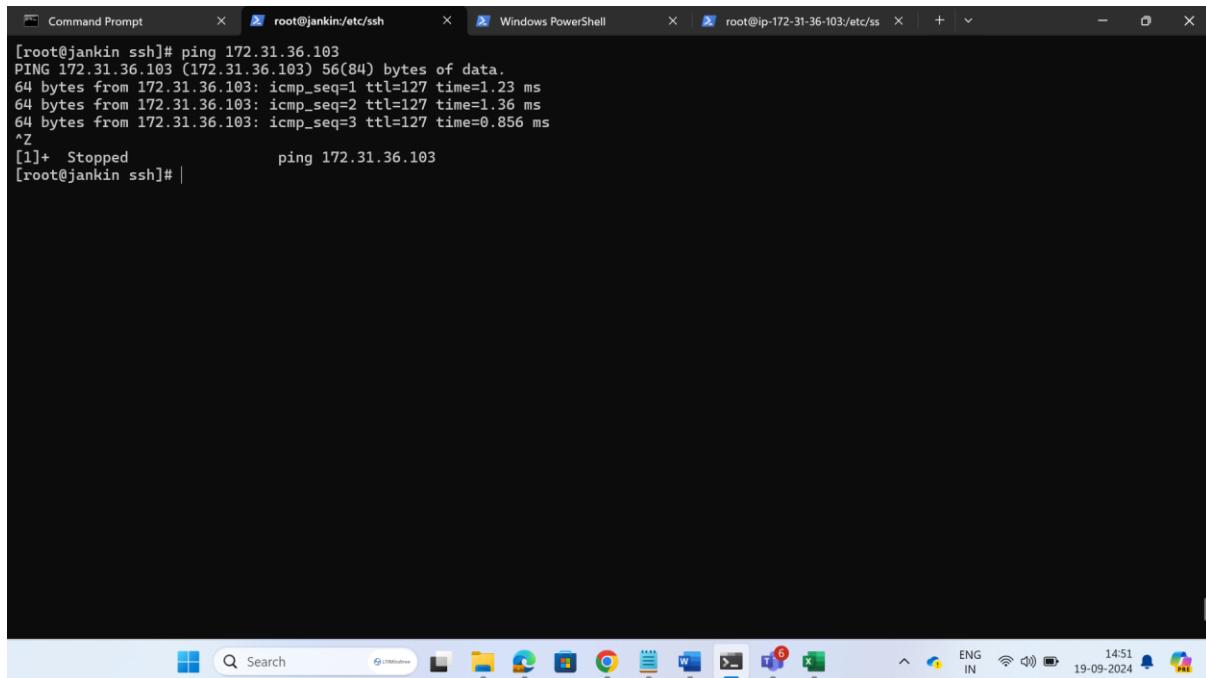


```
-rw-----. 1 root root 2610 Sep 19 09:03 id_rsa
-rw-r--r--. 1 root root 577 Sep 19 09:03 id_rsa.pub
[root@jankin .ssh]# cd /etc/ssh
[root@jankin ssh]# vim sshd_config
[root@jankin ssh]# "sshd_config" 133L, 3840B written
[root@jankin ssh]# systemctl restart sshd
sshd-keygen.target sshd.service      sshd.socket
[root@jankin ssh]# systemctl restart sshd
[root@jankin ssh]# ssh-copy-id root @172.31.36.103
/usr/bin/ssh-copy-id: ERROR: Too many arguments. Expecting a target hostname, got:
Usage: /usr/bin/ssh-copy-id [-h|-?|-f|-n|-s] [-i [identity_file]] [-p port] [-F alternative ssh_config file] [[-o <ssh -o options>] . . .] [user@]hostname
      -f: force mode -- copy keys without trying to check if they are already installed
      -n: dry run -- no keys are actually copied
      -s: use sftp -- use sftp instead of executing remote-commands. Can be useful if the remote only allows sftp
      -h|-?: print this help
[root@jankin ssh]# ssh-copy-id root@172.31.36.103
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.36.103 (172.31.36.103)' can't be established.
ED25519 key fingerprint is SHA256:zpXCV/yeNjTlna/VuLzbAvU07xQyTateJZ1lMiamqT0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.36.103's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.36.103'"
and check to make sure that only the key(s) you wanted were added.

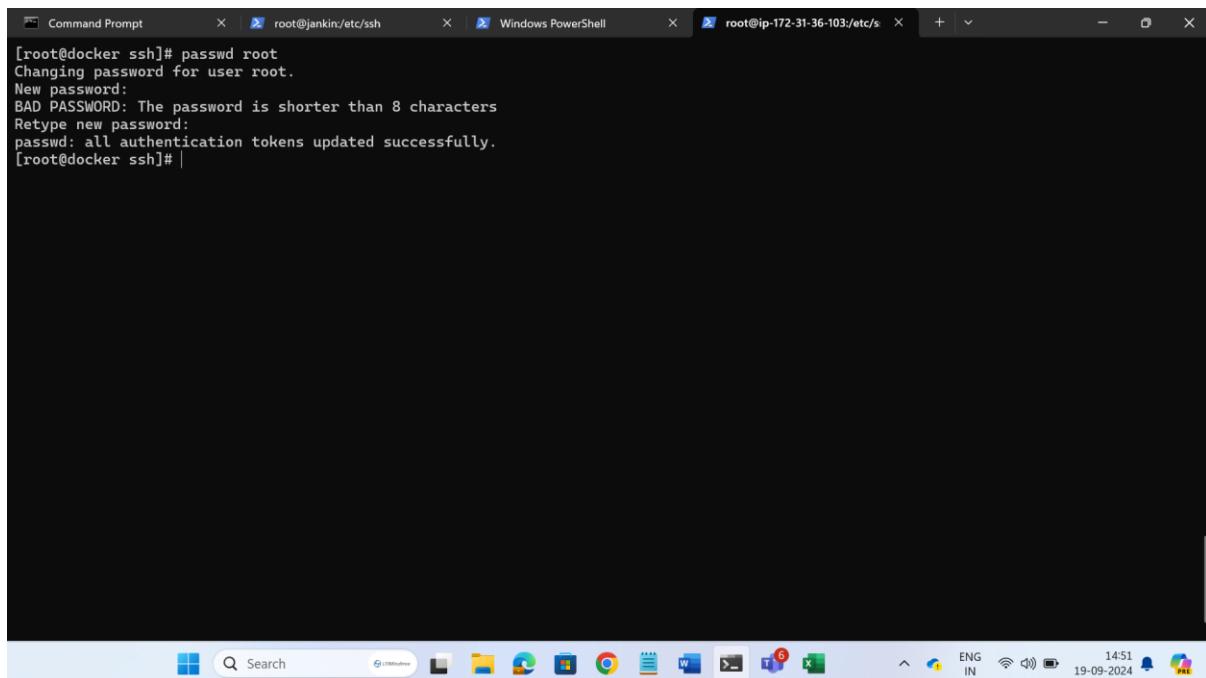
[root@jankin ssh]#
```

Check ping Docker ip Address in Jenkins Machine



```
[root@jankin ssh]# ping 172.31.36.103
PING 172.31.36.103 (172.31.36.103) 56(84) bytes of data.
64 bytes from 172.31.36.103: icmp_seq=1 ttl=127 time=1.23 ms
64 bytes from 172.31.36.103: icmp_seq=2 ttl=127 time=1.36 ms
64 bytes from 172.31.36.103: icmp_seq=3 ttl=127 time=0.856 ms
^Z
[1]+  Stopped                  ping 172.31.36.103
[root@jankin ssh]# |
```

Set Password in Docker Machine :



```
[root@docker ssh]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@docker ssh]# |
```

The screenshot shows the Jenkins System configuration page. A new SSH server is being created with the following details:

- Name:** jenkins
- Hostname:** 172.31.32.175
- Username:** root
- Remote Directory:** /root

A checkbox for "Avoid sending files that have not changed" is unchecked. At the bottom, there are "Save" and "Apply" buttons.

The screenshot shows the Jenkins System configuration page. A new SSH server is being created with the following details:

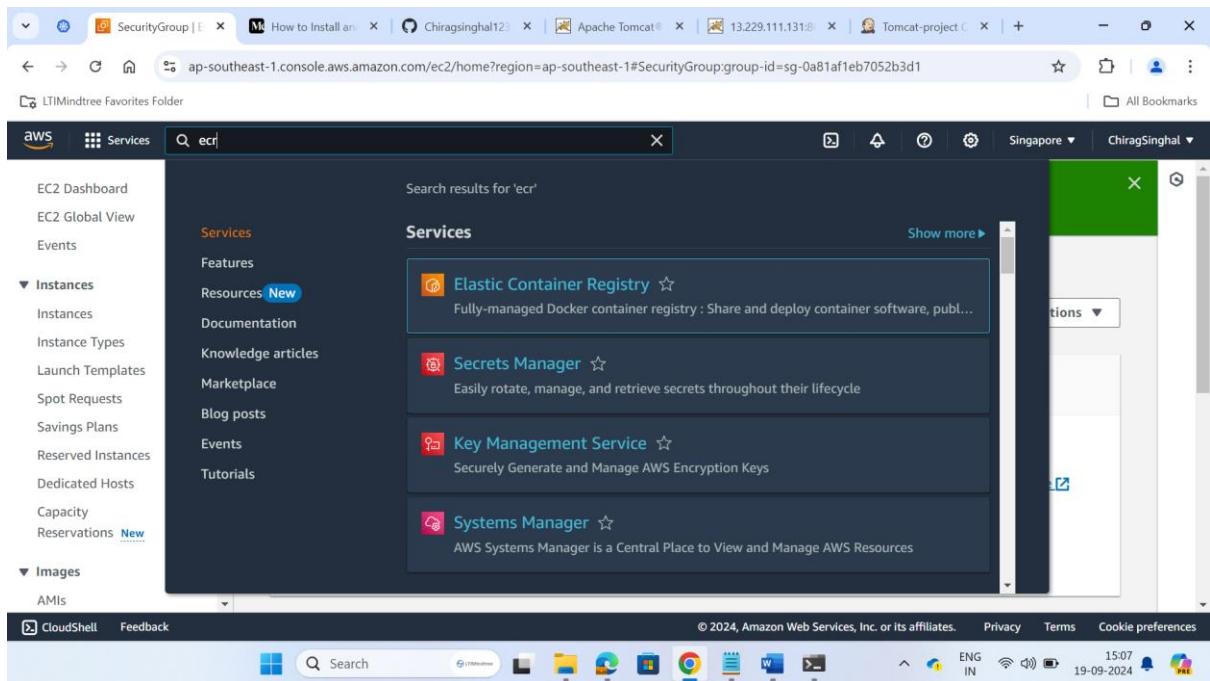
- Name:** docker
- Hostname:** 172.31.36.103
- Username:** root
- Remote Directory:** /root

A checkbox for "Avoid sending files that have not changed" is unchecked. At the bottom, there are "Save" and "Apply" buttons.

Show Docker Machine IP Address

```
[root@docker ssh]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@docker ssh]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
    valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 06:74:92:23:03:27 brd ff:ff:ff:ff:ff:ff
        altname eni-089d583783ad7214f
        altname device-number=0
        inet 172.31.36.103/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
            valid_lft 3177sec preferred_lft 3177sec
        inet6 fe80::474:92ff:fe23:327/64 scope link
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:84:d9:6a:9a brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
[root@docker ssh]# |
```

Create ECR [Elastic Container Registry]



Then Click Create

The screenshot shows the AWS CloudShell interface with multiple tabs open. The active tab is the ECR home page. The page title is "Amazon Elastic Container Registry" with the subtitle "Share and deploy container software, publicly or privately". On the right side, there is a call-to-action box with the text "Create a repository" and a large orange "Create" button. Below the main title, there is a section titled "How it works" with a brief description. To the right of this, there is a "Pricing (US)" section stating "You pay only for the amount of data you store in". At the bottom of the page, there is a table titled "Images (0)" showing "No images". The left sidebar shows navigation options for "Private registry" and "Public registry". The status bar at the bottom indicates the date and time as "19-09-2024 15:08".

Click view Push Command

The screenshot shows the AWS ECR console with the URL `ap-southeast-1.console.aws.amazon.com/ecr/repositories/private/559050235518/projectecr?region=ap-southeast-1`. The page displays instructions for pushing a Docker image to the repository. It includes four numbered steps with corresponding command snippets:

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:
`aws ecr get-login-password --region ap-southeast-1 | docker login --username AWS --password-stdin 559050235518.dkr.ecr.ap-southeast-1.amazonaws.com`
Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:
`docker build -t projectecr .`
3. After the build completes, tag your image so you can push the image to this repository:
`docker tag projectecr:latest 559050235518.dkr.ecr.ap-southeast-1.amazonaws.com/projectecr:latest`
4. Run the following command to push this image to your newly created AWS repository:
`docker push 559050235518.dkr.ecr.ap-southeast-1.amazonaws.com/projectecr:latest`

The screenshot shows the Jenkins configuration interface for a job named "Tomcat-project". The "Post-build Actions" section is selected. A modal dialog titled "Send build artifacts over SSH ?" is open, showing the "SSH Publishers" configuration. It contains an "SSH Server" section with a "Name" field set to "docker" and an "Advanced" dropdown. Below it is a "Transfers" section with a "Transfer Set" section. At the bottom of the dialog are "Save" and "Apply" buttons.

```
Command Prompt x Windows PowerShell x root@jankin:~/.ssh x root@docker:-
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.32.175's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.32.175'" and check to make sure that only the key(s) you wanted were added.

[root@docker .ssh]# cat au
cat: au: No such file or directory
[root@docker .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCBvgTjo/B5QXI403lp37ezaGT57Nod9oxv0Yk5JWHKA1xCRn7ilvRFhddibYXVnqky3WJBm+sanIzb6VcnkAKBf9VQnhd6PH0/x7kg4EKRnjtfmBZQXThns1aYrk0Jo29ijloUzYm/R3kLff80HJwQc+IS001Gvjbc5+BmtBM9y3VDibsz5EAsoo943zLqTcijwaXJbeQAKOKimuyUUF2Vbw9jp8zE709QDaiAe/n0d8hF93u+gR9dDmcM46ps+cfcPvzKf7ClcTQPFEC27T7ZCVUwShYoDe1QePhFjkchA+v7WBfm7D3irbSLcXcJpedDt6zn7F6qzr newkey
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCBvgTjo/B5QXI403lp37ezaGT57Nod9oxv0Yk5JWHKA1xCRn7ilvRFhddibYLyZFL18gDxWJky+yd88t6CKi6RNvYkQl0Xu14tgL7f4j6o98/+g7xovc7nwHAXLQRJ0sRYrcvHARxbkHMbcnr50ko0u/Nee7CySCoQZyzoRmnk5getLHw/kcUhylqFbuF+Y8YcArW0m2omnIYQz8+knsopckwCq/HusbpvHGHPxlfuzEpC1rYPP6Zfd0geir1GsyMsEvNA8d+2Pn1KA69oZhzae/sIsYP39UXHs7oPcEPiVi9oJbD2UySowjyZzjYLAQkHCc9DypUkMTLHNhG4pTDgISkSLn9oX1IDjakcX1jirn8ZbeBN/8YvigWJ0Dt02cACoUefLfzoH9NaDNbE+NFEExJXUko7h0pnOniU6+0+HSc8x1pfj3MfjjtFir0tYRdt18y9fUJcz6hm9gcYLTwb0v= root@jankin.example.com
[root@docker .ssh]# cd
[root@docker ~]# vim /etc/ssh/sshd_config
[root@docker ~]# systemctl restart sshd
[root@docker ~]# systemctl enable sshd
[root@docker ~]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
```

Add the Docker Machine's public key to the Jenkins Machine

```
Command Prompt x Windows PowerShell x root@jankin:~/.ssh x root@docker:-
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.32.175's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.32.175'" and check to make sure that only the key(s) you wanted were added.

[root@docker .ssh]# cat au
cat: au: No such file or directory
[root@docker .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCBvgTjo/B5QXI403lp37ezaGT57Nod9oxv0Yk5JWHKA1xCRn7ilvRFhddibYXVnqky3WJBm+sanIzb6VcnkAKBf9VQnhd6PH0/x7kg4EKRnjtfmBZQXThns1aYrk0Jo29ijloUzYm/R3kLff80HJwQc+IS001Gvjbc5+BmtBM9y3VDibsz5EAsoo943zLqTcijwaXJbeQAKOKimuyUUF2Vbw9jp8zE709QDaiAe/n0d8hF93u+gR9dDmcM46ps+cfcPvzKf7ClcTQPFEC27T7ZCVUwShYoDe1QePhFjkchA+v7WBfm7D3irbSLcXcJpedDt6zn7F6qzr newkey
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCBvgTjo/B5QXI403lp37ezaGT57Nod9oxv0Yk5JWHKA1xCRn7ilvRFhddibYLyZFL18gDxWJky+yd88t6CKi6RNvYkQl0Xu14tgL7f4j6o98/+g7xovc7nwHAXLQRJ0sRYrcvHARxbkHMbcnr50ko0u/Nee7CySCoQZyzoRmnk5getLHw/kcUhylqFbuF+Y8YcArW0m2omnIYQz8+knsopckwCq/HusbpvHGHPxlfuzEpC1rYPP6Zfd0geir1GsyMsEvNA8d+2Pn1KA69oZhzae/sIsYP39UXHs7oPcEPiVi9oJbD2UySowjyZzjYLAQkHCc9DypUkMTLHNhG4pTDgISkSLn9oX1IDjakcX1jirn8ZbeBN/8YvigWJ0Dt02cACoUefLfzoH9NaDNbE+NFEExJXUko7h0pnOniU6+0+HSc8x1pfj3MfjjtFir0tYRdt18y9fUJcz6hm9gcYLTwb0v= root@jankin.example.com
[root@docker .ssh]# cd
[root@docker ~]# vim /etc/ssh/sshd_config
[root@docker ~]# systemctl restart sshd
[root@docker ~]# systemctl enable sshd
[root@docker ~]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
```

“Add the Jenkin Machine’s public key to the Docker Machine

```
Command Prompt      X  Windows PowerShell      X  root@jankin:~/ssh      X  root@docke:~      -  D
drwx-----. 2 root root 48 Sep 19 11:21 ssh-copy-id.L03JNmsEDQ
[root@jankin .ssh]# cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlnNzaClzKtdjEAAAABG5vbUAAAABm9uZ0AAAAAAAABAABlwAAAAdzc2gtcn
NhAAAaWEEAAQAAYEAL9MtavEBnlisQgAa3yBFn6A1GC8R8BdtddArp+Ykz5Dh80HtG6xEr
W4P1Bw6y7zRqJzRilhPdNj2X0qHgS8sxzSPKg8ViSvsnfPE+giouktb
2JEJdF7teLYC+3!+qfpF/o08aL3058bwMS0EsdLEWk3LxwEcw5BzG3J6+tpKNLzRhIws
kggEGcszgDZOYhSwxPyNFnIcalah7hfmpGKH1jptqJpyGEM/PpJ7KD3JmAq4VlG6bxz
hZzRbxsK0Ps2D2x+m3d1Hqq9RrMjLBFXDQP0Phfj59Za0vaGycwH7CLGD9/VFx4Uu6Dz
3BD4LSpaCwH9lGLKM18mc42CwEJbwNPQ2KVJDE5RzYRkuUw4CEpEi5/aFyN042pHF9SYq5
/GW3g6fia7TtnAqFhny32a/B/TwgzlxPjRRMSV1JK04dKzjzYL0vtPh0nGvmdax
49zH4U7RyqrzWEQ7dfmV1CXM+zYznRC07cIdL9AAFKoJ808vo/NPLAAAB3NzaC1yc2
EAAAGBAJF7ET2rxRzErIAg7Bz+gBgvErwBxQK6fJM+Sw2/NB7RusREVuByAcOsKzb
+tw0BpGai4Zqzyw103Dy9lqzh6BoUfPYFkRl73zxPoIqlpE291iRCxRe7x12
Avt/1pqj3z/6Dvg19zufAcDetBen5xFity8cBhFuQcxtyevk65jS780R7sLJIKhBnLM6sw
2TmBg6sP0xSHJwUv4X5jxwCt7ybaiahcDpZ6Seyyg9TAkr+Fsxum8yc/GUW7MSkK
Ui7g8/p193SB6vUlaZ1ysRv0oDx37y+fWQd2hmHB7+wi.xg/f1RceFLug89wQ+jUj2gls
PZRIyjCPJn0NgsBCQcJz0N1lsQx0Uc2Ebi1M0dAhKR1uF2hjcu0NqrxFuMkufx/t4E3/x1+
KBVmg0TzWuAKR5t9mf010MiST40UTEldSS5juhSmc42Jt7T4DjxrsHWL+pCx+000Wks
61h0E03xZl19QlZpGbZxtg03CHS/0AAAAMBAEAAAGASEFxmD9Cfyab7Pcx26x0tOruP
His0hFSlyM0VWhuZwKrcxSzJGrn2DNwihg6huaty+Cx70ko7gYt+8masdw86ySe57qy
y7B11vRiMsUma4sXcf7PclQyCv02M0iwrjC1DbzSwNhfVLvwuo71VhIMl2+kE2
257Gv02ytwgRLzk0vF99:0U58m3yiuGHCC2JyH6wuRzaEuZ315qFkdDjdDnCMRA/auta
07Wv1fwrL70zqPphLBWnTyTdpPLzU31ys1YhrXhau42K2L6660pxylbz36AnjxS8F0sgv
Q03nVF5k+k+taIdk9dAcaUVVci4VtKORSaf09Lsr:ramyCtaOfiAm8xBt9s0YGGjYcvkmS
BjAr4ZLunKAYJ6hydWhBH43Nm7i5A82Q6x0tg/19A6eEag2+2P/iqFHNgx0D553LN1lv/
g7PjYwDPhAxhEDXTwxNy3AeD2JdJ3YDzsaa08SlsGuzaE4oDUUApdTvRwTCyGmeps9G0
Nvogh4YBIE7snsekB/Pqtgk08sLamNgT4t0CJdPZA/eW25Mtn4UvQOELGTl1C2XeN6T8
tfS1s/Ucw8K1qJ6g9IVxclWx3QFT7Am1Yk3lyfyn9qHIED2JdmFyY4p4D931N4q7hw
B/DXXXLn56cJaoRg3molDEA8Vlhaa0f0AxM/DbsopB+aqQAAMEAwGgbord/BPfVm0GU
g7P+G+10DfQzjaNbwySo/S54V70SiqJoMtTV18xDph8W0NdDH9GfCdZbcw1kLy0uH7wlH
oSFODFPP/jZjs4dYlCr60R0E4fFGPENpZMyeptkx5zV7qz5oG//IAjgJlqn7UvNwDjsZG
```

The screenshot shows a CI/CD pipeline interface with the following components:

- Build History:** A sidebar on the left lists recent builds:
 - Last build (#4), 1 hr 23 min ago
 - Last stable build (#1), 5 hr 36 min ago
 - Last successful build (#4), 1 hr 23 min ago
 - Last unstable build (#4), 1 hr 23 min ago
 - Last unsuccessful build (#4), 1 hr 23 min ago
 - Last completed build (#4), 1 hr 23 min ago
- Test Result Trend:** A chart titled "Test Result Trend" showing the status of four builds. All builds are marked as "Passed" (green dots).
 - Legend: Passed (Green dot), Skipped (Grey dot), Failed (Red dot)
 - Data: Build #1: Passed, Build #2: Passed, Build #3: Passed, Build #4: Passed
- Permalinks:** A section with a smartphone icon and the text "Latest Test Result (no failures)".
- Workspace Options:** A list of actions:
 - Workspace
 - Build Now
 - Configure
 - Delete Maven project
 - Modules
 - GitHub Hook Log
 - Rename
- Add description:** A button in the top right corner.

Create Kubernetes Cluster machine

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, and the main content area displays the following information:

- Instances (1/5) Info**: Last updated less than a minute ago.
- Find Instance by attribute or tag (case-sensitive)**: A search bar.
- Actions**: A dropdown menu with options like Launch instances, Stop, Start, and Terminate.
- Launch instances**: A large orange button.
- Instance Details**:
 - Name**: kubernetes
 - Instance ID**: i-0af93ca03513bf902
 - Instance state**: Running
 - Instance type**: t2.medium
 - Status check**: 2/2 checks passed
 - Alarm status**: View alarms
- i-0af93ca03513bf902 (kubernetes)**: The instance name.
- Details**: A tabbed view showing Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags.
- Instance summary**:
 - Instance ID**: i-0af93ca03513bf902 (kubernetes)
 - Public IPv4 address**: 54.255.250.28 | open address
 - Private IPv4 addresses**: 172.31.33.194
 - IPv6 address**:
 - Instance state**:
 - Public IPv4 DNS**:

Kubernetes Machine is Ready and Change Hostname

The screenshot shows a Windows PowerShell window with several tabs open at the top. The active tab displays the command to change the host name:

```
PS C:\Users\107477850> cd .\Downloads<br/>PS C:\Users\107477850\Downloads> ssh -i "newkey.pem" ec2-user@ec2-54-255-250-28.ap-southeast-1.compute.amazonaws.com<br/>The authenticity of host 'ec2-54-255-250-28.ap-southeast-1.compute.amazonaws.com (54.255.250.28)' can't be established.<br/>ED25519 key fingerprint is SHA256:K9Pv4y5e45HoCVOaIXzC4v50G8RH3N//0NSkRpuXFPk.<br/>This key is not known by any other names<br/>Are you sure you want to continue connecting (yes/no/[fingerprint])? yes<br/>Warning: Permanently added 'ec2-54-255-250-28.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.<br/>#_<br/>~\_\#\#\#_ Amazon Linux 2023<br/>~~ \_\#\#\#\_<br/>~~ \#\#\|<br/>~~ \#/ _-- https://aws.amazon.com/linux/amazon-linux-2023<br/>~~ \~' '-><br/>~~ .-' /<br/>~/ /'<br/>/m/'<br/>[ec2-user@ip-172-31-33-194 ~]$ sudo su -<br/>[root@ip-172-31-33-194 ~]# hostnamectl set-hostname eks.example.com<br/>[root@ip-172-31-33-194 ~]# bash<br/>[root@eks ~]#
```

The taskbar at the bottom shows various icons for Microsoft Office applications like Word, Excel, and PowerPoint.

Create ssh-keygen [Mean create public,private IP Address]

The screenshot shows a Windows PowerShell window with several tabs open at the top. The active tab displays the command to generate an SSH key pair:

```
[root@eks ~]# ssh-keygen<br/>Generating public/private rsa key pair.<br/>Enter file in which to save the key (/root/.ssh/id_rsa):<br/>Enter passphrase (empty for no passphrase):<br/>Enter same passphrase again:<br/>Your identification has been saved in /root/.ssh/id_rsa<br/>Your public key has been saved in /root/.ssh/id_rsa.pub<br/>The key fingerprint is:<br/>SHA256:ByazK+VrxOLPRg/X+FZ5IdUDw6F0LJBQqr043qbBak root@eks.example.com<br/>The key's randomart image is:<br/>+---[RSA 3072]---+<br/>| o ..o+o+o+o+ |<br/>| + ...+o+ o |<br/>| ... ++ .. |<br/>| o+ = . |<br/>| +o . S + . |<br/>| +o... . + . |<br/>| E ++.+ + |<br/>| .++0 . |<br/>| .o+ =. |<br/>+---[SHA256]---+<br/>[root@eks ~]#<br/>[root@eks ~]#
```

The taskbar at the bottom shows various icons for Microsoft Office applications like Word, Excel, and PowerPoint.

Create Roles :-

The screenshot shows the AWS IAM Roles page. The left sidebar includes options like Dashboard, Access management (Roles selected), Policies, Identity providers, and Account settings. The main area displays a table of 17 roles, each with a name and a corresponding AWS service listed next to it. A 'Create role' button is visible at the top right.

Role name	Trusted entities
AWSServiceRoleForAmazonEKS	AWS Service: eks (Service-Linked Role)
AWSServiceRoleForAmazonEKSNodegroup	AWS Service: eks-nodegroup (Service-Linked Role)
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-Linked Role)
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)
AWSServiceRoleForBackup	AWS Service: backup (Service-Linked Role)
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)

Select AWS service

The screenshot shows the 'Create role' wizard at Step 3: 'Name, review, and create'. It asks for the 'Trusted entity type'. There are five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below this, a 'Use case' section is shown with the note: 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.'

Select EC2

The screenshot shows the AWS IAM 'Create role' wizard. In Step 1, 'Service or use case' is set to 'EC2'. Under 'Use case', the 'EC2' option is selected, which is described as allowing EC2 instances to call AWS services on behalf of the user.

Then Name of the role

The screenshot shows the 'Name, review, and create' step of the wizard. The 'Role name' field contains 'project-role'. The 'Description' field contains 'Allows EC2 instances to call AWS services on your behalf.' The 'Step 1: Select trusted entities' section shows a trust policy with a single entry: "Version": "2012-10-17".

Allow Permission << AmazonEKSClusterPolicy < IAMFullAccess <
AmazonElasticContainerRegistryPublicFullAccess >>

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonEKSClusterPolicy	AWS managed	Permissions policy
AmazonElasticContainerRegistryPublicFullAccess	AWS managed	Permissions policy
IAMFullAccess	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional info

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create role

Ready for Role

Role project-role created.

Roles (16) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
project-role	AWS Service: ec2	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

- Access AWS from your non AWS workloads
- X.509 Standard
- Temporary credentials

Manage

View role

CloudShell Feedback

Kubernetes instance add role then goto action and then modify IAM Role

The screenshot shows the AWS EC2 Instances page. A single instance named "kubernetes" is listed as "Running". The "Actions" menu is open, and the "Modify IAM role" option is highlighted. The browser address bar shows the URL: ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#Instances:instanceState=running.

Allow permission in PermitRootLogin and PasswordAuthentication

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/key passwords
```

Restart and enable the sshd

```
[root@eks ~]# systemctl restart sshd
[root@eks ~]# systemctl enable sshd
[root@eks ~]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 06:48:d9:2d:09:13 brd ff:ff:ff:ff:ff:ff
    alname eni-04272d6aea8262b49
    alname device-number-0.0
    inet 172.31.33.194/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
        valid_lft 1838sec preferred_lft 1838sec
    inet6 fe80::48d9:fe2d:913/64 scope link
        valid_lft forever preferred_lft forever
[root@eks ~]#
```

Add the Docker Machine's public key to the Kubernetes Machine.”

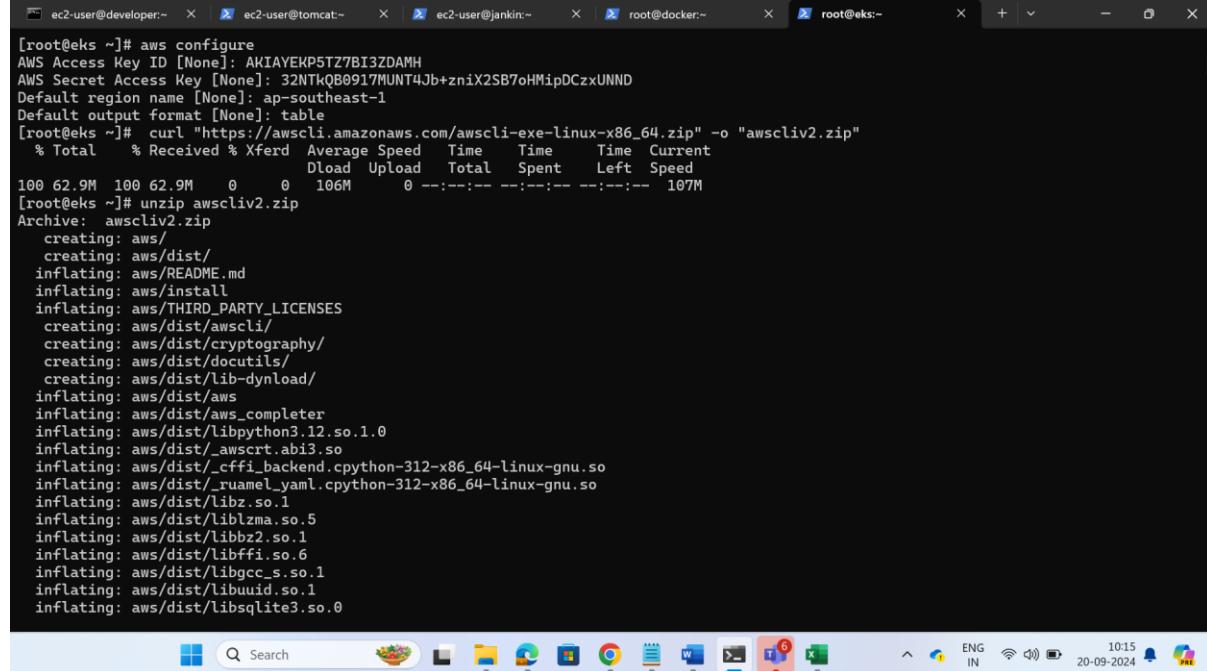
```
[ec2-user@developer ~] $ client_loop: send disconnect: Connection reset
PS C:\Users\10747850\Downloads> ssh -i "newkey.pem" ec2-user@ec2-13-250-118-126.ap-southeast-1.compute.amazonaws.com
      _#
     /_###_
Amazon Linux 2023
~~ \_#####
~~ \###_
~~ \###_
~~ \#/_ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~'`->
~~ .-/
~~ /-/
~~ /-/
~~ /-/
Last login: Fri Sep 20 04:09:04 2024 from 165.225.120.228
[ec2-user@docker ~]$ sudo su -
Last login: Thu Sep 19 12:41:56 UTC 2024 on pts/3
Last failed login: Fri Sep 20 04:11:53 UTC 2024 from 23.249.28.102 on ssh:notty
There were 6 failed login attempts since the last successful login.
[root@docker ~]# ssh-copy-id root@172.31.33.194
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.33.194' (172.31.33.194') can't be established.
ED25519 key fingerprint is SHA256:K9Pv4y5e45HoCV0aIXzC4v5G8RH3N//0NSkRpuXFPk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.33.194's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.33.194'"
and check to make sure that only the key(s) you wanted were added.

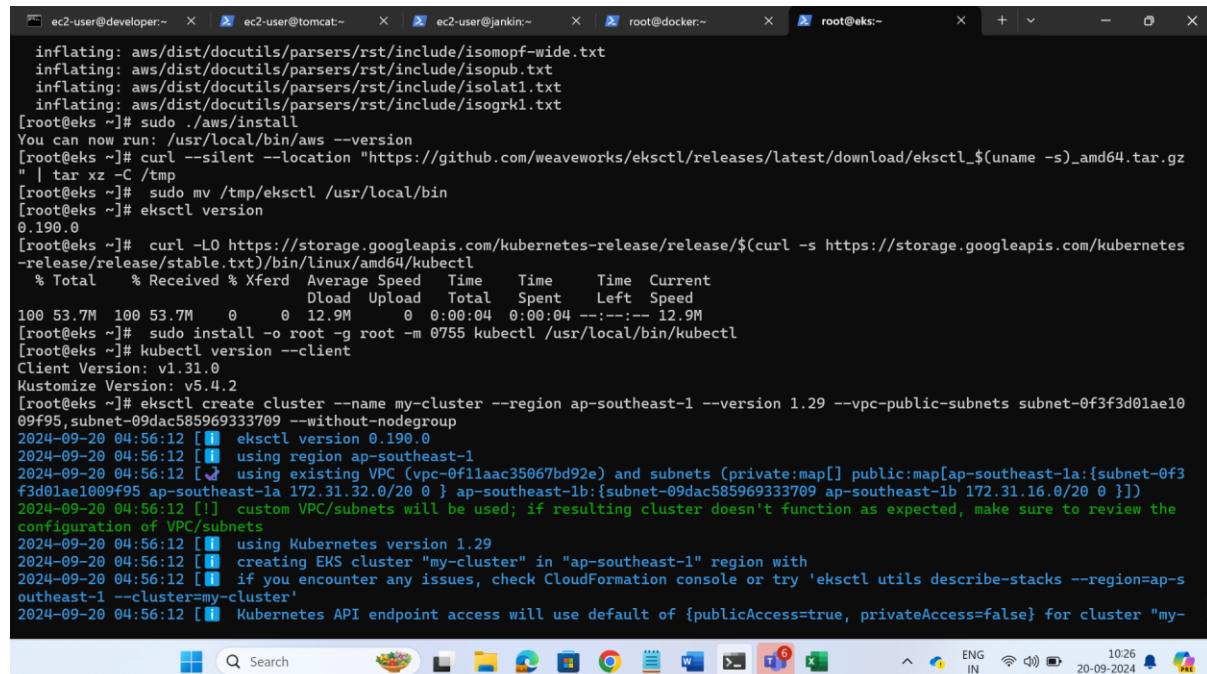
[root@docker ~]# |
```

Configure the AWS:



```
[root@eks ~]# aws configure
AWS Access Key ID [None]: AKIAYEKP5TZ7BI3ZDAMH
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-1
Default output format [None]: table
[root@eks ~]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100 62.9M  100 62.9M    0      0  106M      0 --:--:-- --:--:-- 107M
[root@eks ~]# unzip awscliv2.zip
Archive: awscliv2.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/README.md
  inflating: aws/install
  inflating: aws/THIRD_PARTY_LICENSES
  creating: aws/dist/awscli/
  creating: aws/dist/cryptography/
  creating: aws/dist/docutils/
  creating: aws/dist/lib-dynload/
  inflating: aws/dist/aws
  inflating: aws/dist/aws_completer
  inflating: aws/dist/_awscrt_abi3.so
  inflating: aws/dist/_cffi_backend.cpython-312-x86_64-linux-gnu.so
  inflating: aws/dist/_ruamel_yaml.cpython-312-x86_64-linux-gnu.so
  inflating: aws/dist/libz.so.1
  inflating: aws/dist/liblzma.so.5
  inflating: aws/dist/libbz2.so.1
  inflating: aws/dist/libffi.so.6
  inflating: aws/dist/libgcc_s.so.1
  inflating: aws/dist/libuuid.so.1
  inflating: aws/dist/libsqLite3.so.0
```

We Created a Cluster



```
[root@eks ~]# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
[root@eks ~]# curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz"
" | tar xz -C /tmp
[root@eks ~]# sudo mv /tmp/eksctl /usr/local/bin
[root@eks ~]# eksctl version
0.190.0
[root@eks ~]# curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100 53.7M  100 53.7M    0      0 12.9M      0:00:04 0:00:04 --:--:-- 12.9M
[root@eks ~]# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[root@eks ~]# kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2
[root@eks ~]# eksctl create cluster --name my-cluster --region ap-southeast-1 --version 1.29 --vpc-public-subnets subnet-0f3f3d01ae1009f95,subnet-09dac58596933709 --without-nodegroup
2024-09-20 04:56:12 [!] eksctl version 0.190.0
2024-09-20 04:56:12 [!] using region ap-southeast-1
2024-09-20 04:56:12 [!] using existing VPC (vpc-0f11aac35067bd92e) and subnets (private:map[] public:map[ap-southeast-1a:{subnet-0f3f3d01ae1009f95 ap-southeast-1a 172.31.32.0/20 0 } ap-southeast-1b:{subnet-09dac58596933709 ap-southeast-1b 172.31.16.0/20 0 }])
2024-09-20 04:56:12 [!] custom VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configuration of VPC/subnets
2024-09-20 04:56:12 [!] using Kubernetes version 1.29
2024-09-20 04:56:12 [!] creating EKS cluster "my-cluster" in "ap-southeast-1" region with
2024-09-20 04:56:12 [!] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-southeast-1 --cluster=my-cluster'
2024-09-20 04:56:12 [!] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "my-
```

```
[root@eks ~]# 
}
}
2024-09-20 04:56:12 [!] building cluster stack "eksctl-my-cluster-cluster"
2024-09-20 04:56:13 [!] deploying stack "eksctl-my-cluster-cluster"
2024-09-20 04:56:43 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 04:57:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 04:58:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 04:59:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 05:00:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 05:01:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 05:02:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 05:03:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 05:04:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 05:05:13 [!] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-20 05:05:13 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2024-09-20 05:05:13 [!] creating addon
2024-09-20 05:05:14 [!] successfully created addon
2024-09-20 05:05:14 [!] creating addon
2024-09-20 05:05:14 [!] successfully created addon
2024-09-20 05:05:15 [!] creating addon
2024-09-20 05:05:15 [!] successfully created addon
2024-09-20 05:07:15 [!] waiting for the control plane to become ready
2024-09-20 05:07:16 [!] saved kubeconfig as "/root/.kube/config"
2024-09-20 05:07:16 [!] no tasks
2024-09-20 05:07:16 [!] all EKS cluster resources for "my-cluster" have been created
2024-09-20 05:07:16 [!] created 0 nodegroup(s) in cluster "my-cluster"
2024-09-20 05:07:16 [!] created 0 managed nodegroup(s) in cluster "my-cluster"
2024-09-20 05:07:17 [!] kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
2024-09-20 05:07:17 [!] EKS cluster "my-cluster" in "ap-southeast-1" region is ready
[root@eks ~]#
```

After Creating a Cluster I created a node group

```
[root@eks ~]# eksctl create nodegroup \
--cluster my-cluster \
--region ap-southeast-1 \
--name my-node-group \
--node-ami-family Ubuntu2004 \
--node-type t2.small \
--subnet-ids subnet-0f3f3d01ae1009f95,subnet-09dac585969333709 \
--nodes 3 \
--nodes-min 2 \
--nodes-max 4 \
--ssh-access \
--ssh-public-key /root/.ssh/id_rsa.pub
2024-09-20 08:42:57 [!] will use version 1.29 for new nodegroup(s) based on control plane version
2024-09-20 08:42:57 [!] nodegroup "my-node-group" will use "ami-013fed5b876f7df77" [Ubuntu2004/1.29]
2024-09-20 08:42:57 [!] using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-my-cluster-nodegroup-my-node-group-1a:4c:e6:0b:5b:8e:69:0b:42:02:62:2b:16:c0:99:90"
2024-09-20 08:42:58 [!] 1 nodegroup (my-node-group) was included (based on the include/exclude rules)
2024-09-20 08:42:58 [!] will create a CloudFormation stack for each of 1 managed nodegroups in cluster "my-cluster"
2024-09-20 08:42:58 [!] 
2 sequential tasks: { fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "my-node-group" } } }
2024-09-20 08:42:58 [!] checking cluster stack for missing resources
2024-09-20 08:42:58 [!] cluster stack has all required resources
2024-09-20 08:42:58 [!] building managed nodegroup stack "eksctl-my-cluster-nodegroup-my-node-group"
2024-09-20 08:42:58 [!] deploying stack "eksctl-my-cluster-nodegroup-my-node-group"
2024-09-20 08:42:58 [!] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-node-group"
```

Image is Created by Docker

The screenshot shows the AWS ECR console with the URL ap-southeast-1.console.aws.amazon.com/ecr/repositories/private/559050235518/projectecr?region=ap-southeast-1. The page displays the 'projectecr' repository details. Under the 'Images' section, there is one entry for the 'latest' tag. The table shows the following information:

Image tag	Artifact type	Pushed at	Size (M)	Image	Digest
latest	Image	September 19, 2024, 18:25:55 (UTC+05.5)	227.34		sha256:934ace3...

A tooltip 'Image URI copied' is visible over the image thumbnail. The browser interface includes tabs for Billing, Elastic, How to, clone, Snjay, Apache, 13.225, Sign in, Sanjay, and others. The AWS navigation bar shows Services, Search, and the user ChiragSinghal.

This is the Deployment file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-declarative
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: 559050235518.dkr.ecr.ap-southeast-1.amazonaws.com/projectecr:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
```

The terminal window shows the deployment.yaml file with 20L and 408B. The system tray indicates ENG IN, 14:22, 20-09-2024, and a battery level of 20,20. The taskbar includes icons for File Explorer, Task View, Task Manager, and other applications.

Create Deployment file

```
[root@eks files]# vim deployment.yaml |
```

The terminal window shows the command 'vim deployment.yaml' being run in a root shell on an EKS node. The system tray indicates ENG IN, 15:54, 20-09-2024, and a battery level of 20,20. The taskbar includes icons for File Explorer, Task View, Task Manager, and other applications.

Create service file

```
[root@eks files]# vim service.yaml |
```

This is the Server file

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-app
spec:
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
  type: LoadBalancer
```

```
[root@eks files]# kubectl expose deployment/nginx-declarative --type="LoadBalancer" --port 8080|
```

Configure Kubernetes and paste it public IP Address

The screenshot shows a web browser window with the URL `ec2-47-129-218-82.ap-southeast-1.compute.amazonaws.com:8080/manage/configure`. The page displays a form for configuring a Jenkins slave node. The fields are as follows:

- Name**: eks
- Hostname**: 172.31.33.194
- Username**: root
- Remote Directory**: /root
- Avoid sending files that have not changed**: An unchecked checkbox.

At the bottom of the form, there are "Save" and "Apply" buttons. The browser's address bar shows the Jenkins URL, and the taskbar at the bottom includes icons for various Windows applications like File Explorer, Task Manager, and Edge.

In Configure Automate the process to delete the old deployment and apply new deployment whenever there is a commit :

Remove prefix ?

Remote directory ?

Exec command ?

```
cd /files
kubectl delete deployment nginx-declarative
kubectl apply -f deployment.yaml
kubectl apply -f service.yaml
```

Save Apply

LTIMindtree Billing Billing How to clone-/ Snjay_D Apache 13.229. Tomcat sanjay a5e New Tab ENG IN 17:05 21-09-2024 PRE

Deployment is Completed

Please fill in this form to create an account.

Enter Name Enter Full Name
Enter mobile Enter mobile number
Enter Email Enter Email
Password Enter Password
Repeat Password Repeat Password

By creating an account you agree to our [Terms & Privacy](#).

[Register](#)

Already have an account? [Sign in](#).

Registration App for DevOps Learning by Chirag Singhal PSID - 10747850

Thank You, Sanjay Sir

See You Again



Final output is displayed by taking the service ip:8080/webapp/ which is being hosting in the Tomcat server.

