

Launch an instance [Info](#)

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

Name and tags [Info](#)

Name: Jenkins-server [Add additional tags](#)

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

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

Recent [Quick Start](#)

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux [Browse more AMIs](#) Including AMIs from AWS, Marketplace and

[CloudShell](#) [Feedback](#)

Summary

Number of instances: [Info](#)

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2... [read more](#)
ami-0182f573e66f89c85

Virtual server type (instance type)
t2.micro

Firewall (security group)
Jenkins_Security_Group

Storage (volumes)
1 volume(s) - 25 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and

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

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Instance type [Info](#) | [Get advice](#)

Instance type

t2.large
Family: t2 2 vCPU 8 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.1208 USD per Hour
On-Demand RHEL base pricing: 0.1216 USD per Hour
On-Demand SUSE base pricing: 0.1938 USD per Hour
On-Demand Linux base pricing: 0.0928 USD per Hour

[All generations](#) [Compare instance types](#)

Key pair (login) [Info](#)

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

Key pair name - required
 [Create new key pair](#)

Network settings [Info](#)

VPC - required [Info](#)
 (default) [Change](#)

Subnet [Info](#)

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

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Screenshot of the AWS CloudFormation console showing the creation of a new instance.

Network settings

- VPC - required: [Info](#) **vpc-0671076c5968bda50** (default) [Edit](#)
- Subnet: [Info](#) No preference [Create new subnet](#)
- Auto-assign public IP: [Info](#) Enable
- Additional charges apply when outside of [free tier allowance](#)
- Firewall (security groups): [Info](#) Jenkins_Security_Group
- Storage (volumes): 1 volume(s) - 25 GiB

Configure storage

- Root volume (Not encrypted): 1x 25 GiB gp3

Summary

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2... [read more](#) ami-0182f373e66f89c85
- Virtual server type (instance type): t2.large
- Firewall (security group): Jenkins_Security_Group
- Storage (volumes): 1 volume(s) - 25 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and

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

Screenshot of the AWS EC2 Security Groups console showing the details of a security group named Jenkins_Security_Group.

Details

Security group name: Jenkins_Security_Group	Security group ID: sg-0ee1e494325a12211	Description: launch-wizard-13 created 2024-08-14T02:24:24.058Z	VPC ID: vpc-0671076c5968bda50
Owner: 754173272413	Inbound rules count: 2 Permission entries	Outbound rules count: 1 Permission entry	

Inbound rules (2)

Name	Security group rule...	IP versi...	Type	Protocol	Port range	Source
-	sgr-0a695fc53edb34c3a	IPv4	Custom TCP	TCP	8080	0.0.0.0/0
-	sgr-0734b66ebfb5b576c8	IPv4	SSH	TCP	22	0.0.0.0/0

Go to Jenkins downloads → Redhat

SSH into your ec2 instance. Change the name

```
vi /etc/hostname
```

```
change to Jenkins-server
```

```
press esc :wq
```

```
init 6
```

```
ssh again
```

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

```
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

```
yum install fontconfig java-17-openjdk (this wont work so)
```

```
sudo dnf install java-17-amazon-corretto-devel
```

```
to check java --version
```

```
yum install Jenkins
```

```
systemctl status Jenkins
```

```
systemctl enable Jenkins
```

```
systemctl start Jenkins
```

Jenkins cd ▾

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

Jenkins Redhat Packages

To use this repository, run the following command:

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

If you've previously imported the key from Jenkins, the `rpm --import` will fail because you already have a key. Please ignore that and move on.

```
yum install fontconfig java-17-openjdk
yum install jenkins
```

The rpm packages were signed using this key:

```
pub rsa4096 2023-03-27 [SC] [expires: 2026-03-26]
  63667EE74BBA1F0A08A698725BA31057EF5975CA
uid Jenkins Project
sub rsa4096 2023-03-27 [E] [expires: 2026-03-26]
```

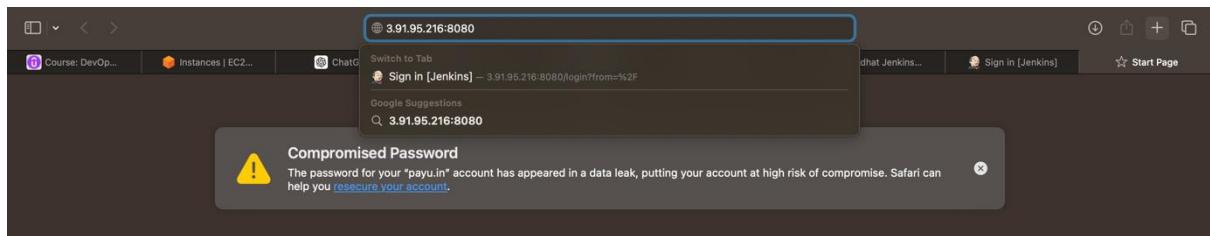
You will need to explicitly install a supported Java runtime environment (JRE), either from your distribution (as described above) or another Java vendor (e.g., [Adoptium](#)).

Weekly Release Line

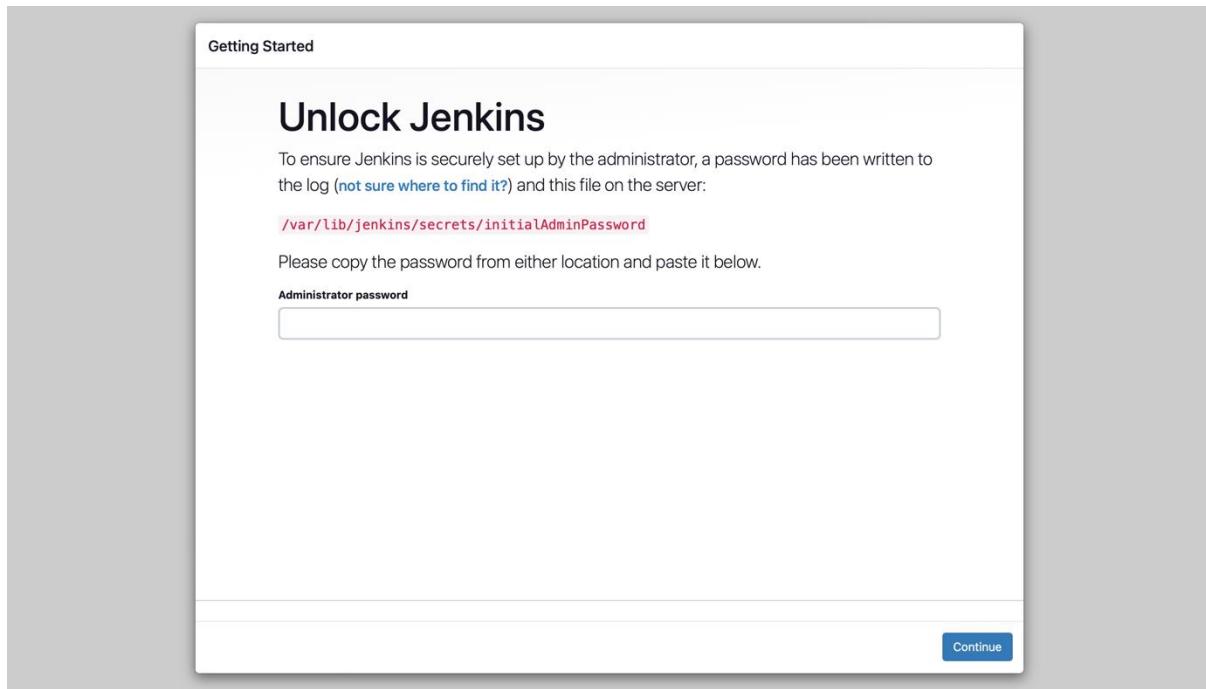
Supported Java versions for the weekly release line are:

2.463 (June 2024) and newer
Java 17 or Java 21

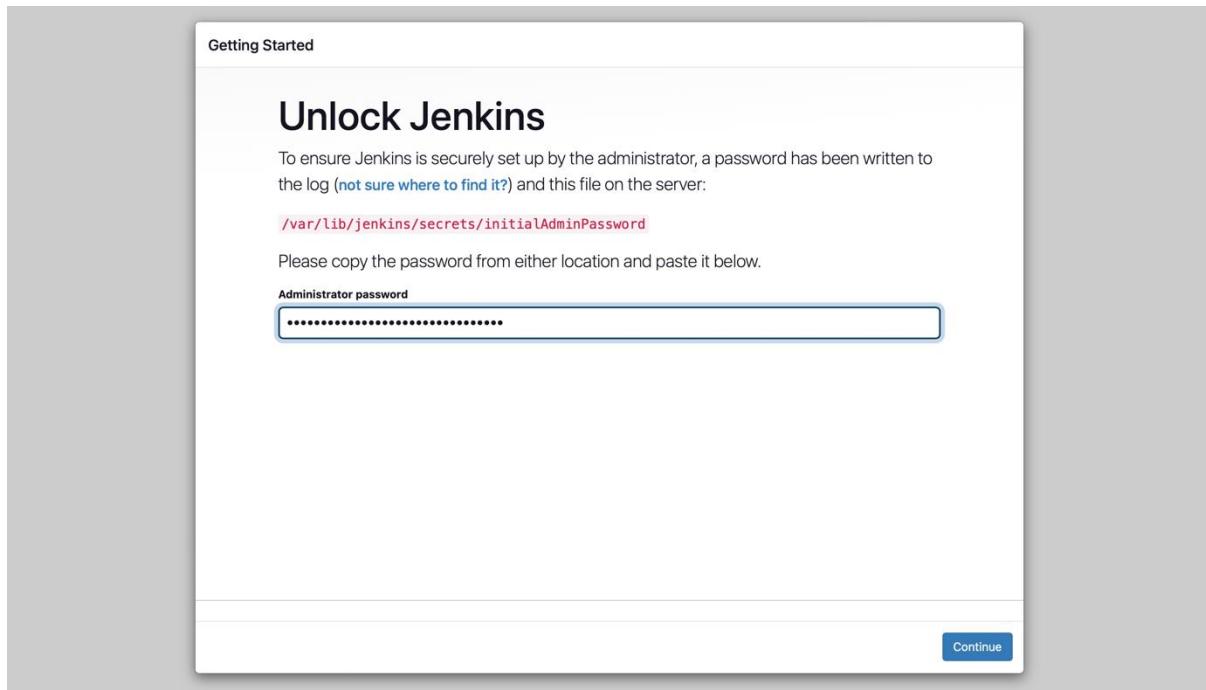
copy paste public ip of ec2 with :8080



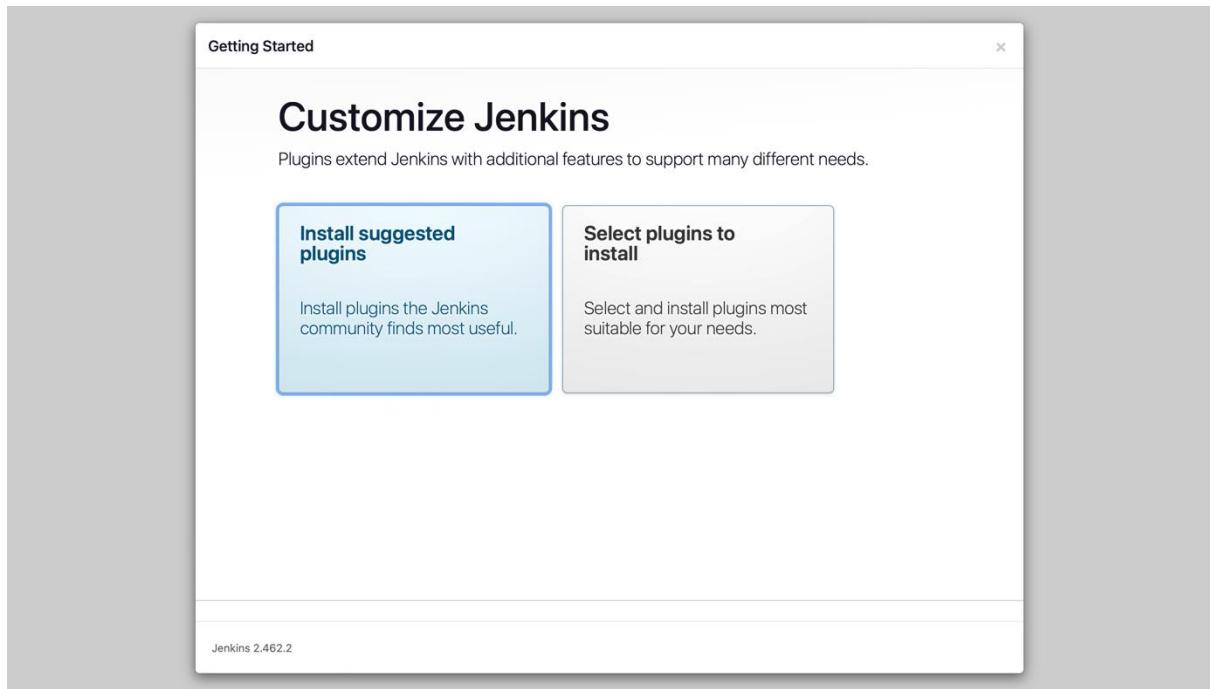
You will get Jenkins dashboard. To get password copy that in screen



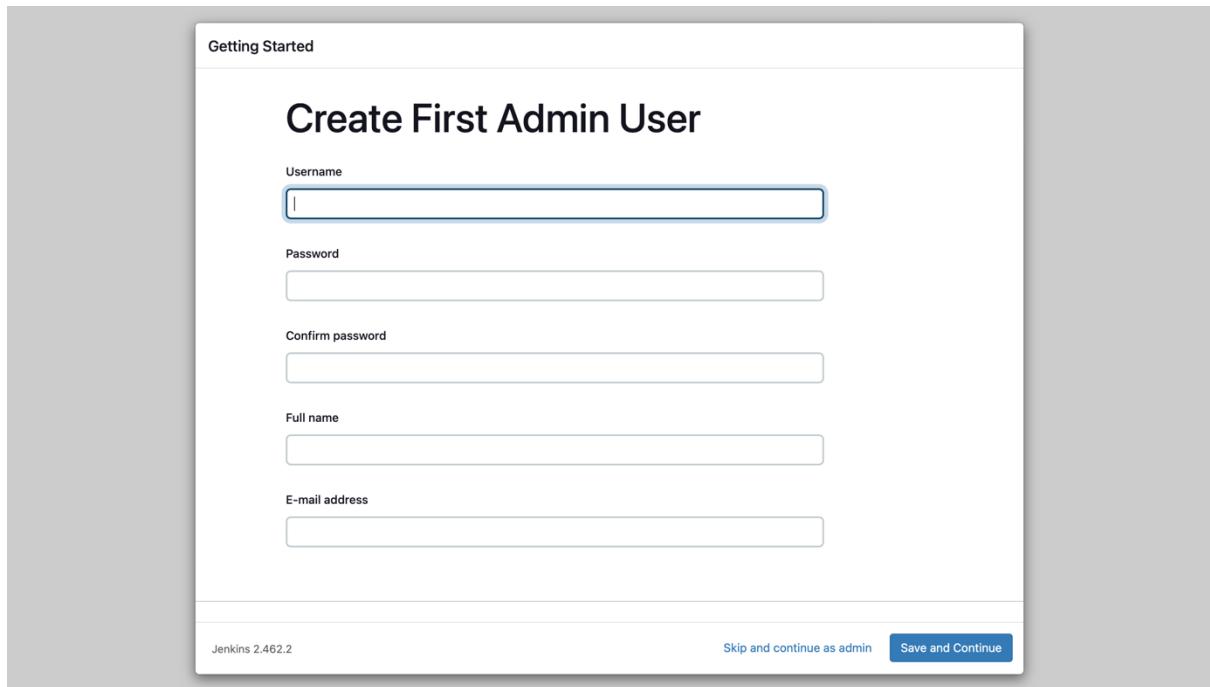
In terminal, cat <paste> you will get password
Paste the password from terminal here



Install suggested plugins



Skip



The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links: '+ New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are two sections: 'Build Queue' (empty) and 'Build Executor Status' (2 Idle). On the right, there's a central area titled 'Welcome to Jenkins!'. It includes a search bar, a 'Create a job' button, and several informational cards: 'Set up a distributed build' (with 'Set up an agent' and 'Configure a cloud' buttons), 'Start building your software project' (with a 'Create a job' button), and 'Learn more about distributed builds'. At the bottom right, it says 'REST API' and 'Jenkins 2.462.2'.

Go to manage Jenkins → plugins → github integration → install

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. The left sidebar has tabs: 'Updates', 'Available plugins' (selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area shows a search bar with 'github integration' and an 'Install' button. A table lists the 'GitHub Integration' plugin: Version 0.7.0, released 6 mo 23 days ago, with 'emailext' and 'Build Triggers' listed under it. The 'Released' column shows a blue checkmark icon.

Next we gotta let our Jenkins know wheres is our git.
Run “whereis git” on terminal
Next run “git”
Apply and save

Dashboard > Manage Jenkins > Tools

Git installations

Git

Name: Git

Path to Git executable: git

Install automatically

Add Git ▾

Gradle installations

Add Gradle

Ant installations

Save **Apply**

Maven integration

Go to google → maven download → copy that binary tar link

The screenshot shows the Apache Maven Project website at <http://maven.apache.org/>. The main navigation bar includes links for Welcome, License, About Maven, What is Maven?, Features, Download, Use, Release Notes, Documentation, Maven Plugins, Maven Extensions, Index (category), User Centre, Plugin Developer Centre, Maven Repository Centre, Maven Developer Centre, Books and Resources, Security, Community, and Project Overview.

The 'Download' menu is currently selected. Below it, there are sections for 'System Requirements' and 'Files'.

In the 'System Requirements' section, it lists Java Development Kit (JDK) requirements (Maven 3.9+ requires JDK 8 or above), memory requirements (No minimum requirement), and disk requirements (Approximately 10MB required for the Maven installation itself).

The 'Files' section displays three download options: 'Binary tar.gz archive', 'Binary zip archive', and 'Source tar.gz archive'. The 'Binary tar.gz archive' link has a context menu open, with 'Copy Link' being the selected option.

Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.9.9-bin.tar.gz.sha512	apache-maven-3.9.9-bin.tar.gz.asc
Binary zip archive	apache-maven-3.9.9-bin.zip.sha512	apache-maven-3.9.9-bin.zip.asc
Source tar.gz archive	apache-maven-3.9.9-src.tar.gz.sha512	apache-maven-3.9.9-src.tar.gz.asc

On terminal :

```
See 'git help git' for an overview of the system.
[root@jenkins-server ~]# cd /opt
[root@jenkins-server opt]# wget https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
--2024-09-10 18:06:25 https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org) [151.181.2.132]:443... connected.
Connecting to dlcdn.apache.org (dlcdn.apache.org) [151.181.2.132]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9102945 (8.7M) [application/x-gzip]
Saving to: 'apache-maven-3.9.9-bin.tar.gz'

apache-maven-3.9.9-bin.tar.gz      100%[=====]  8.68M --.-KB/s   in 0.1s

2024-09-10 18:06:26 (75.9 MB/s) - 'apache-maven-3.9.9-bin.tar.gz' saved [9102945/9102945]

[root@jenkins-server opt]# ll
total 8892
drwxr-xr-x.  1 root root 9102945 Aug 17 18:44 apache-maven-3.9.9-bin.tar.gz
[root@jenkins-server opt]# tar -xvf apache-maven-3.9.9-bin.tar.gz
apache-maven-3.9.9/README.txt
apache-maven-3.9.9/LICENSE
apache-maven-3.9.9/NOTICE
apache-maven-3.9.9/NOTICE
apache-maven-3.9.9/lib/apachalliance.license
apache-maven-3.9.9/lib/commons-cli.license
apache-maven-3.9.9/lib/commons-codec.license
apache-maven-3.9.9/lib/failureaccess.license
apache-maven-3.9.9/lib/guava.license
apache-maven-3.9.9/lib/httpclient.license
apache-maven-3.9.9/lib/httpcore.license
apache-maven-3.9.9/lib/jansi.license
apache-maven-3.9.9/lib/javax.annotation-api.license
apache-maven-3.9.9/lib/javax.inject.license
apache-maven-3.9.9/lib/jboss-logging.license
apache-maven-3.9.9/lib/org.eclipse.sisu.inject.license
apache-maven-3.9.9/lib/org.eclipse.sisu.plexus.license
apache-maven-3.9.9/lib/plexus-cipher.license
apache-maven-3.9.9/lib/plexus-component-annotations.license
apache-maven-3.9.9/lib/plexus-interpolation.license
apache-maven-3.9.9/lib/plexus-maven-watcher.license
apache-maven-3.9.9/lib/plexus-utils.license
apache-maven-3.9.9/lib/plexus-xml.license
apache-maven-3.9.9/lib/slf4j-api.license
apache-maven-3.9.9/boot/
apache-maven-3.9.9/boot/plexus-classworlds.license
apache-maven-3.9.9/bin/ant.7z
apache-maven-3.9.9/bin/ansi-native/Windows/
apache-maven-3.9.9/bin/ansi-native/Windows/arm64/
apache-maven-3.9.9/bin/ansi-native/Windows/x86/
apache-maven-3.9.9/bin/ansi-native/Windows/x86_64/
apache-maven-3.9.9/bin/ansi-native/Windows/x86/jansi.so
apache-maven-3.9.9/bin/ansi-native/Windows/x86/jansi.dll
apache-maven-3.9.9/bin/ansi-native/Windows/x86_64/jansi.dll
apache-maven-3.9.9/bin/m2.conf
apache-maven-3.9.9/bin/mvn.cmd
apache-maven-3.9.9/bin/mvnDebug.cmd
apache-maven-3.9.9/bin/mvn
apache-maven-3.9.9/bin/mvnDebug
apache-maven-3.9.9/bin/mvnvp
apache-maven-3.9.9/conf/
apache-maven-3.9.9/conf/logging/
apache-maven-3.9.9/conf/logging/simplelogger.properties
```

Pwd

Cd /opt

wget <https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz>

ll to see

next we extract that tar

tar -xvf apache-maven-3.9.9-bin.tar.gz

ll

we rename it to maven just to make it simple

mv apache-maven-3.9.9 maven

cd maven

ll

if you go inside bin

cd bin

ll

you will see mvn

here if you run ./mvn – v , you can see it

but if you go outside of this directory and try it doesn't work

cd ..

./mvn – v

```

apache-maven-3.9.9/lib/maven-slf4j-provider-3.9.9.jar
apache-maven-3.9.9/lib/plexus-interpolation-3.27.jar
apache-maven-3.9.9/lib/maven-resolver-transport-http-1.9.22.jar
apache-maven-3.9.9/lib/maven-model-3.9.9.jar
apache-maven-3.9.9/lib/plexus-xml-3.0.1.jar
apache-maven-3.9.9/lib/maven-resolver-spi-1.9.22.jar
apache-maven-3.9.9/lib/maven-resolver-transport-3.9.9.jar
apache-maven-3.9.9/lib/commons-codec-5.1.0.jar
apache-maven-3.9.9/lib/commons-codec-1.17.1.jar
apache-maven-3.9.9/lib/lavax/inject-1.jar
apache-maven-3.9.9/lib/org/eclipse/sisu/plexus-0.9.0.M3.jar
apache-maven-3.9.9/lib/jansi-2.4.1.jar
apache-maven-3.9.9/lib/plexus-threads-api-1.0.12-jlocks-1.9.22.jar
apache-maven-3.9.9/lib/maven-plugin-api-3.9.9.jar
apache-maven-3.9.9/lib/httpclient-4.5.14.jar
apache-maven-3.9.9/lib/maven-resolver-transport-file-1.9.22.jar
apache-maven-3.9.9/lib/plexus-component-annotations-2.1.8.jar
apache-maven-3.9.9/lib/maven-resolver-connector-basic-1.9.22.jar
apache-maven-3.9.9/lib/plexus-utils-3.6.1.jar
[root@jenkins-server opt]\$ ll
[root@jenkins-server opt]\$ ll
total 892
drwxr-xr-x. 6 root root 99 Sep 10 18:06 apache-maven-3.9.9
-rw-r--r--. 1 root root 9102945 Aug 17 18:44 apache-maven-3.9.9-bin.tar.gz
drwxr-xr-x. 3 root root 33 Sep 3 21:10 maven
[root@jenkins-server opt]\$ mv apache-maven-3.9.9 maven
[root@jenkins-server opt]\$ cd maven
[root@jenkins-server maven]\$ ll
total 48
-rw-r--r--. 1 root root 18929 Aug 14 08:48 LICENSE
-rw-r--r--. 1 root root 5834 Aug 14 08:48 NOTICE
-rw-r--r--. 1 root root 12000 Aug 14 08:48 RELEASE-NOTES.txt
drwxr-xr-x. 2 root root 97 Sep 10 18:06 bin
drwxr-xr-x. 2 root root 76 Sep 10 18:06 boot
drwxr-xr-x. 3 root root 63 Aug 14 08:48 conf
drwxr-xr-x. 4 root root 16384 Sep 10 18:06 lib
[root@jenkins-server maven]\$ cd bin
[root@jenkins-server bin]\$ ll
total 32
-rw-r--r--. 1 root root 327 Aug 14 08:48 m2.conf
-rwxr-xr-x. 1 root root 5917 Aug 14 08:48 mvn
-rw-r--r--. 1 root root 6324 Aug 14 08:48 mvn.cmd
-rwxr-xr-x. 1 root root 1684 Aug 14 08:48 mvnDebug
-rw-r--r--. 1 root root 1684 Aug 14 08:48 mvnDebug.cmd
-rwxr-xr-x. 1 root root 1611 Aug 14 08:48 mvnJsp
[root@jenkins-server bin]\$ ./mvn -v
Apache Maven 3.9.9 (8e8579a97e7fd8d15ee5ec7bfcdc97d260186937)
Maven home: /opt/maven
Java version: 17.0.12, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en_US.UTF-8, platform encoding: UTF-8
OS name: "linux", version: "6.1.186-116.188.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins-server bin]\$
[root@jenkins-server bin]\$ ./mvn -v
Apache Maven 3.9.9 (8e8579a97e7fd8d15ee5ec7bfcdc97d260186937)
Maven home: /opt/maven
Java version: 17.0.12, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en_US.UTF-8, platform encoding: UTF-8
OS name: "linux", version: "6.1.186-116.188.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins-server bin]\$
[root@jenkins-server bin]\$ cd ..
[root@jenkins-server maven]\$ ./mvn -v
:bash: ./mvn: No such file or directory
[root@jenkins-server maven]\$
```

That's the reason we have to set it up in environment variables.

To set it up in the environment variables, we can either update the dot bash underscore profile of the user, which it is going to use, or else we can update in the environment variables directly.

But I would like to make it available only for the root user.
That is where I'm going to update it under .bash_profile.

So .bash_profile will be available off here.
Root user home directory.

```
Cd ~
ll -a
```

(nothing but hidden files it is going to display. You can see here, bash underscore profile is there. We need to edit it.)

```
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: "6.1.106-116.188.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins-server bin]#
[root@jenkins-server bin]# cd ..
[root@jenkins-server maven]# ./mvn -v
-bash: ./mvn: No such file or directory
[root@jenkins-server maven]#
[root@jenkins-server maven]# cd ~
[root@jenkins-server ~]# pwd
/root
[root@jenkins-server ~]# ll -a
total 28
dr-xr-x---. 3 root root 137 Sep 10 17:34 .
dr-xr-xr-x. 18 root root 237 Sep 3 21:09 ..
-rw-r--r--. 1 root root 18 Feb 2 2023 .bash_logout
-rw-r--r--. 1 root root 141 Feb 2 2023 .bash_profile
-rw-r--r--. 1 root root 429 Feb 2 2023 .bashrc
-rw-r--r--. 1 root root 100 Feb 2 2023 .cshrc
drwx-----. 2 root root 29 Sep 10 16:10 .ssh
-rw-r--r--. 1 root root 129 Feb 2 2023 .tcshrc
-rw-----. 1 root root 747 Sep 10 17:32 .viminfo
-rw-r--r--. 1 root root 164 Sep 10 17:34 .wget-hsts
[root@jenkins-server ~]# vi .bash_profile
```

```
vi .bash_profile
```

For this we need java path. So close this

Esc :wq

In root,

find / -name jvm
It will give 3, in that /usr/lib/jvm

```
cd /usr/lib/jvm  
||
```

you will get java and jvm path

copy it

you can also use

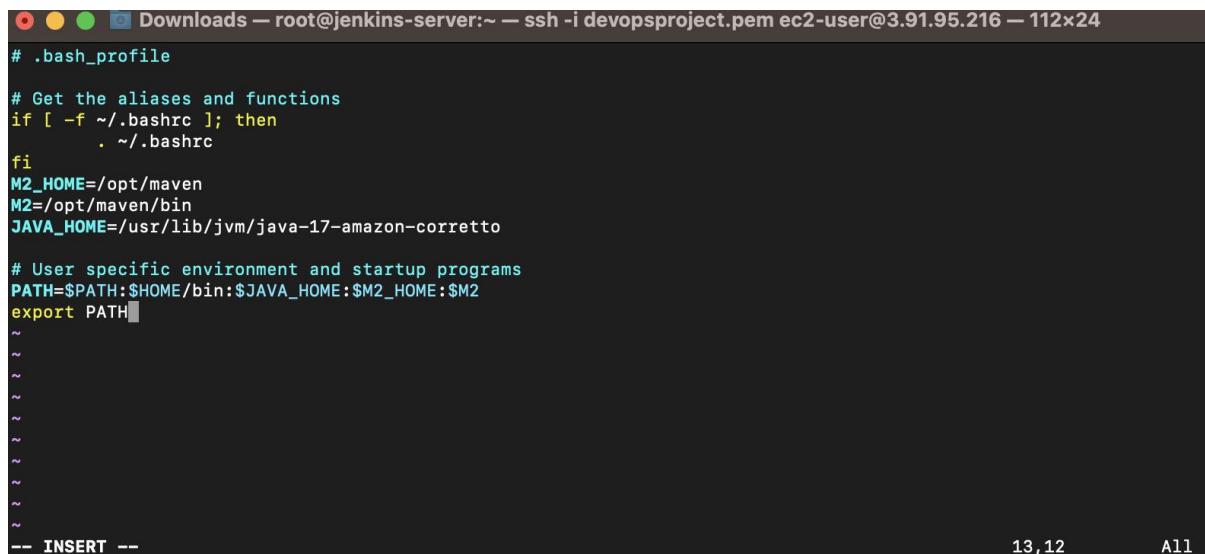
```
find / -name java-17*
```

it will suggest java-17-amazon-corretto
find / -name java-17-amazon-corretto

```
cd ~  
ll -a  
vi .bash_profile
```

Here we gotta add our java path, as well as M2 and M2_HOME

M2_HOME – This is the path of where our apache maven is available
M2 – this is the binary directory
JAVA_HOME – Java home path is nothing but where your jdk is available



```
# .bash_profile  
  
# Get the aliases and functions  
if [ -f ~/.bashrc ]; then  
    . ~/.bashrc  
fi  
M2_HOME=/opt/maven  
M2=/opt/maven/bin  
JAVA_HOME=/usr/lib/jvm/java-17-amazon-corretto  
  
# User specific environment and startup programs  
PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2  
export PATH
```

At last in path, we tell to load these value.

```
PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2
```

If I don't specify \$ it will take the same value, if you specify it will replace the new value we have specified.

```
# .bash_profile  
  
# Get the aliases and functions  
if [ -f ~/.bashrc ]; then  
    . ~/.bashrc  
fi  
M2_HOME=/opt/maven  
M2=/opt/maven/bin  
JAVA_HOME=/usr/lib/jvm/java-17-amazon-corretto  
  
# User specific environment and startup programs
```

```
PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2
```

```
export PATH
```

Save it.

Let's save this file. And if I check dollar path to validate whether our changes are affected or not, so far it hasn't affected.

Why? Because we have updated it in the Bash under score profile.

It will be loaded. In case if you re-log into server usr root, okay?

Now what we can do, either we can log out from the root and log in back.

Or else we can just execute, source dot bash underscore profile, okay?

Source command is going to read the Bash underscore profile again.

Load the values which are updated. Now, if I check the echo \$PATH, now you can see Java path as well as the Maven and Maven Home Path, sorry.

Maven Home Path as well as Maven. Okay?

Now if I do mvn -v,

it should be able to execute from anywhere. I'm not in the actual directory of MVN. Now I'm outside even though I could be able to execute it.

```
-rw-r--r--, 1 root root 100 Feb 2 2023 .cshrc
drwx-----, 2 root root 29 Sep 10 16:10 .ssh
-rw-r--r--, 1 root root 129 Feb 2 2023 .tshrc
-rw-r--r--, 1 root root 129 Feb 2 2023 .viminfo
-rw-r--r--, 1 root root 164 Sep 10 17:54 wget-hsts
[root@jenkins-server ~]# vi .bash_profile
[root@jenkins-server ~]#
[root@jenkins-server ~]# find / -name jvm
/etc/jvm
/usr/lib/jvm
/usr/share/jvm
[root@jenkins-server ~]# cd /usr/lib/jvm
[root@jenkins-server jvm]# ll
total 0
lrwxrwxrwx. 1 root root 26 Sep 10 17:35 java -> /etc/alternatives/java_sdk
lrwxrwxrwx. 1 root root 43 Sep 10 17:35 java-17-amazon-corretto -> /etc/alternatives/java-17-amazon-corretto
drwxr-xr-x. 8 root root 94 Sep 10 17:35 java-17-amazon-corretto.x86_64
lrwxrwxrwx. 1 root root 37 Sep 10 17:35 java-17-openjdk -> /etc/alternatives/java_sdk_17_openjdk
lrwxrwxrwx. 1 root root 21 Sep 10 17:35 jre -> /etc/alternatives/jre
lrwxrwxrwx. 1 root root 24 Sep 10 17:35 jre-17 -> /etc/alternatives/jre_17
lrwxrwxrwx. 1 root root 32 Sep 10 17:35 jre-17-openjdk -> /etc/alternatives/jre_17_openjdk
lrwxrwxrwx. 1 root root 32 Sep 10 17:35 jre-openjdk -> /etc/alternatives/jre_openjdk
[root@jenkins-server jvm]#
[root@jenkins-server jvm]# find / -name java-17*
find: paths must precede expression: 'java-17-amazon-corretto'
find: possible unquoted pattern after predicate '-name'?
[root@jenkins-server jvm]# java-17-amazon-corretto
-bash: java-17-amazon-corretto: command not found
[root@jenkins-server jvm]# find / -name java-17-amazon-corretto
/etc/alternatives/java-17-amazon-corretto
/usr/lib/jvm/java-17-amazon-corretto
[root@jenkins-server jvm]#
[root@jenkins-server jvm]# vi .bash_profile
[root@jenkins-server jvm]# cd .
[root@jenkins-server ~]# ll -a
total 28
dr-xr-x---. 3 root root 137 Sep 10 18:43 ..
dr-xr-xr--. 18 root root 237 Sep 3 21:09 ..
-rw-r--r--. 1 root root 18 Feb 2 2023 .bash_logout
-rw-r--r--. 1 root root 1408 Sep 10 18:43 .bash_profile
-rw-r--r--. 1 root root 429 Feb 2 2023 .bashrc
-rw-r--r--. 1 root root 180 Feb 2 2023 .cshrc
drwx-----, 2 root root 29 Sep 10 16:10 .ssh
-rw-r--r--, 1 root root 129 Feb 2 2023 .tshrc
-rw-r--r--, 1 root root 129 Feb 2 2023 .viminfo
-rw-r--r--, 1 root root 164 Sep 10 17:54 wget-hsts
[root@jenkins-server ~]# vi .bash_profile
[root@jenkins-server ~]# echo SPATH
/root/.local/bin:/root/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
[root@jenkins-server ~]# source .bash_profile
[root@jenkins-server ~]# echo SPATH
/root/.local/bin:/root/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/usr/lib/jvm/java-17-amazon-corretto:/opt/maven:/opt/maven/bin
[root@jenkins-server ~]#
[root@jenkins-server ~]# pwd
/
[root@jenkins-server ~]# mvn -v
Apache Maven 3.9.4 (8e8579a976f7d015ee5ec7bfcdc97d266186937)
Maven home: /opt/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: "6.1.106-116.186.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins-server ~]#
```

Next thing is we need to tell to our Jenkins JUI that Maven is available under this location and Java is available under this location.

For that, first we need to install Maven plugin. Then we should configure it.

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Search: maven

Install

Install	Name	Released
<input checked="" type="checkbox"/>	Maven Integration 3.23	Build Tools 1 yr 1 mo ago
<input type="checkbox"/>	Config File Provider 978.v8e85886ffdc4	Groovy-related External Site/Tool Integrations Maven 1 day 9 hr ago
<input type="checkbox"/>	Jira 3.13	External Site/Tool Integrations Maven jira 6 mo 0 days ago
<input type="checkbox"/>	Pipeline Maven Integration 1421.v610fa_b_e2d60e	pipeline Maven 3 mo 5 days ago
<input type="checkbox"/>	Artifactory 4.0.8	pipeline This plugin allows your build jobs to download artifacts and resolve dependencies to and from Artifactory.

Install

Click Go back to the top page.

Dashboard > Manage Jenkins > Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Plugin	Status
Pipeline Graph View	Success
Git	Success
EDDSA API	Success
Trilead API	Success
SSH Build Agents	Success
Matrix Authorization Strategy	Success
PAM Authentication	Success
LDAP	Success
Email Extension	Success
Mailer	Success
Theme Manager	Success
Dark Theme	Success
Loading plugin extensions	Success
GitHub Integration	Success
Loading plugin extensions	Success
Javadoc	Success
JSch dependency	Success
Maven Integration	Success
Loading plugin extensions	Success

→ [Go back to the top page](#)
(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.462.2

Lets first add JDK

Name – java-17
Java_home : /usr/lib/jvm/java-17-amazon-corretto

Dashboard > Manage Jenkins > Tools

JDK installations

Add JDK

≡ JDK

Name: java-17

JAVA_HOME: /usr/lib/jvm/java-17-amazon-corretto

Install automatically ?

Add JDK

Git installations

≡ Git

Name:

Save **Apply**

This screenshot shows the Jenkins 'Tools' configuration page. It displays two sections: 'JDK installations' and 'Git installations'. In the 'JDK installations' section, a new entry named 'java-17' is being added with the path '/usr/lib/jvm/java-17-amazon-corretto'. There is an option to 'Install automatically'. In the 'Git installations' section, a new entry named 'Git' is being added. At the bottom, there are 'Save' and 'Apply' buttons.

Scroll down, Add maven. Unselect install automatically.

Dashboard > Manage Jenkins > Tools

Add Ant

Maven installations

Add Maven

≡ Maven

Name: maven-3.9.9

MAVEN_HOME: /opt/maven

Install automatically ?

Add Maven

Save **Apply**

Jenkins 2.462.2

This screenshot shows the Jenkins 'Tools' configuration page with the 'Maven installations' section selected. A new entry named 'maven-3.9.9' is being added with the path '/opt/maven'. The 'Install automatically' checkbox is unchecked. At the bottom, there are 'Save' and 'Apply' buttons. A Jenkins watermark 'Jenkins 2.462.2' is visible at the bottom right.

Name – maven-3.9.9
Maven_Path - /opt/maven

Don't give full path /opt/maven/bin cuz it doesn't accept full path.
It is saying that it is not like a Maven directory,

just tell /opt/maven
 That is what even we have given,
 M2_Home, right?
 So same thing and let it be.

Dashboard > Manage Jenkins > Tools

Add Ant

Maven installations

Add Maven

Maven

Name: maven-3.9.9

MAVEN_HOME: /opt/maven/bin

Info: /opt/maven/bin doesn't look like a Maven directory

Install automatically ?

Add Maven

Save **Apply**

Apply-Save

Docker:

EC2 > Instances > Launch an instance

Launch an instance Info

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

Name and tags Info

Name: Docker-Host

Application and OS Images (Amazon Machine Image) Info

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

Recent **Quick Start**

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE	<input type="button" value="Browse more AMIs"/>
						Including AMIs from AWS, Marketplace and

Summary

Number of instances:

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI... read more
ami-0a5c3558529277641

Virtual server type (instance type): t2.micro

Firewall (security group): Devops_Security_Group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and

Launch instance **Review commands**

Amazon Machine Image (AMI)

- Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
- Free tier eligible
- ami-0a5c3558529277641 (64-bit (x86)) / ami-05a48cb826fb15fc (64-bit (Arm))
- Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Architecture

- 64-bit (x86)
- AMI ID: ami-0a5c3558529277641
- Verified provider

Instance type

t2.micro

- Family: t2 1 vCPU 1 GiB Memory Current generation: true
- On-Demand Windows base pricing: 0.0162 USD per Hour
- On-Demand SUSE base pricing: 0.0116 USD per Hour
- On-Demand RHEL base pricing: 0.026 USD per Hour
- On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

Launch an instance

Key pair (login)

devopsproject

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more

Virtual server type (instance type): t2.micro

Firewall (security group): Devops_Security_Group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and

Launch instance

Key pair (login)

devopsproject

Network settings

Network: vpc-0671076c5968bda50

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups): Select existing security group

Common security groups: Devops_Security_Group sg-001c2fe96782d7928

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more

Virtual server type (instance type): t2.micro

Firewall (security group): Devops_Security_Group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and

Launch instance

SSH into your ec2 instance. Change the name

```
vi /etc/hostname
change to dockerhost-new
press esc :wq
init 6
ssh again
```

Install Docker

```
sudo su -  
yum install docker -y
```

```
systemctl status docker  
systemctl enable docker  
systemctl start docker
```

```
ec2-user:x:1000:  
[root@dockerhost-new ~]# yum install docker -y  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amzn2-core | 3.6 kB 00:00  
Resolving Dependencies  
--> Running transaction check  
--> Package docker.x86_64 0:25.0.6-1.amzn2.0.2 will be installed  
--> Pre-Trans: Docker: containerd.x86_64 0:1.7.20-1.amzn2.0.1 will be installed  
--> Pre-Trans: libcgroup.x86_64 0:4.41-21.amzn2 will be installed  
--> Processing Dependency: libcgroup >= 0.40.rcl-1.15 for package: docker-25.0.6-1.amzn2.0.2.x86_64  
--> Processing Dependency: runc >= 1.0.0 for package: docker-25.0.6-1.amzn2.0.2.x86_64  
--> Processing Dependency: pigz for package: docker-25.0.6-1.amzn2.0.2.x86_64  
--> Running transaction check  
--> Package containerd.x86_64 0:1.7.20-1.amzn2.0.1 will be installed  
--> Package libcgroup.x86_64 0:4.41-21.amzn2 will be installed  
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed  
--> Package runc.x86_64 0:1.1.13-1.amzn2 will be installed  
--> Finished Dependency Resolution  
Dependencies Resolved  
=====  
Package Arch Version Repository Size  
=====  
Installing:  
  docker x86_64 25.0.6-1.amzn2.0.2 amzn2extra-docker 45 M  
Installing For dependencies:  
  containerd x86_64 1.7.20-1.amzn2.0.1 amzn2extra-docker 38 M  
  libcgroup x86_64 0.41-21.amzn2 amzn2-core 66 k  
  pigz x86_64 2.3.4-1.amzn2.0.1 amzn2-core 81 k  
  runc x86_64 1.1.13-1.amzn2 amzn2extra-docker 3.1 M  
Transaction Summary  
=====  
Install 1 Package (+4 Dependent packages)  
Total download size: 78 M  
Installed size: 285 M  
Downloading packages:  
(1/5): docker-25.0.6-1.amzn2.0.2.x86_64.rpm | 45 M 00:00  
(2/5): containerd-1.7.20-1.amzn2.0.1.x86_64.rpm | 38 M 00:00  
(3/5): docker-25.0.6-1.amzn2.0.2.x86_64.rpm | 66 kB 00:00  
(4/5): libcgroup-0.41-21.amzn2.0.1.x86_64.rpm | 3.1 MB 00:00  
(5/5): runc-1.1.13-1.amzn2.x86_64.rpm | 81 k 00:00  
Total: 73 MB/s | 78 MB 00:01  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : runc-1.1.13-1.amzn2.x86_64 1/5  
  Installing : containerd-1.7.20-1.amzn2.0.1.x86_64 2/5  
  Installing : libcgroup-0.41-21.amzn2.0.1.x86_64 3/5  
  Installing : docker-25.0.6-1.amzn2.0.2.x86_64 4/5  
  Verifying : runc-1.1.13-1.amzn2.x86_64 1/5  
  Verifying : docker-25.0.6-1.amzn2.0.2.x86_64 2/5  
  Verifying : libcgroup-0.41-21.amzn2.0.1.x86_64 3/5  
  Verifying : pigz-2.3.4-1.amzn2.0.1.x86_64 4/5  
  Verifying : containerd-1.7.20-1.amzn2.0.1.x86_64 5/5  
Installed:  
  docker.x86_64 0:25.0.6-1.amzn2.0.2
```

```
[root@dockerhost-new ~]# docker.x86_64 0:25.0.6-1.amzn2.0.2  
Dependency Installed:  
  containerd.x86_64 0:1.7.20-1.amzn2.0.1    libcgroup.x86_64 0:0.41-21.amzn2  
  pigz.x86_64 0:2.3.4-1.amzn2.0.1           runc.x86_64 0:1.1.13-1.amzn2  
Complete!  
[root@dockerhost-new ~]# systemctl status docker  
● docker.service - Docker Application Container Engine  
  Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)  
  Active: inactive (dead)  
    Docs: https://docs.docker.com  
[root@dockerhost-new ~]# systemctl enable docker  
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.  
[root@dockerhost-new ~]# systemctl status docker  
● docker.service - Docker Application Container Engine  
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)  
  Active: active (running) since Tue 2024-09-10 22:46:48 UTC; 2s ago  
    Docs: https://docs.docker.com  
  Process: 3480 ExecStart=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)  
  Process: 3481 ExecStart=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)  
 Main PID: 3485 (dockerd)  
   Tasks: 9  
    Memory: 30.8M  
     CPU: /system.slice/docker.service  
          └─ 3485 /usr/bin/dockerd -H fd:// --containerid=/run/containerd/cont...  
Sep 10 22:46:47 dockerhost-new systemd[1]: Starting Docker Application Conta...  
Sep 10 22:46:47 dockerhost-new dockerd[3485]: time="2024-09-10T22:46:47.6763..."  
Sep 10 22:46:47 dockerhost-new dockerd[3485]: time="2024-09-10T22:46:47.7239..."  
Sep 10 22:46:48 dockerhost-new dockerd[3485]: time="2024-09-10T22:46:48.0074..."  
Sep 10 22:46:48 dockerhost-new dockerd[3485]: time="2024-09-10T22:46:48.0222..."  
Sep 10 22:46:48 dockerhost-new dockerd[3485]: time="2024-09-10T22:46:48.0222..."  
Sep 10 22:46:48 dockerhost-new dockerd[3485]: time="2024-09-10T22:46:48.0662..."  
Sep 10 22:46:48 dockerhost-new systemd[1]: Started Docker Application Contai...  
Hint: Some lines were ellipsized, use -l to show in full.  
[root@dockerhost-new ~]#  
[root@dockerhost-new ~]# cat /etc/passwd  
root:x:0:0:root:/root/:/bin/bash  
bin:x:1:1:bin:/bin/:/sbin/nologin  
daemon:x:2:2:daemon:/sbin/:/sbin/nologin  
adm:x:3:4:adm:/var/adm/:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd/:/sbin/nologin  
sync:x:5:9:sync:/sbin/:/sbin/nologin  
shutdown:x:6:9:shutdown:/sbin/:/sbin/nologin  
halt:x:7:9:halt:/sbin/:/sbin/nologin  
mail:x:8:12:mail:/var/spool/mail/:/sbin/nologin  
operator:x:11:0:operator:/root/:/sbin/nologin  
games:x:12:00:games:/usr/games/:/sbin/nologin  
ftp:x:44:50:FTP User:/var/ftp/:/sbin/nologin  
nobody:x:99:99:Nobody:/var/empty/:/sbin/nologin  
systemnetworkx:x:92:192:system Network Management:/sbin/nologin  
dbus:x:81:81:System message bus:/sbin/nologin  
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind/:/sbin/nologin  
libstoragemgmt:x:99:997:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin  
sshd:x:74:74:SSH Daemon: /etc/empty/:/sbin/nologin  
rndisfilter:x:996:RNDIS Filter:/var/run/rndisfilter/:/sbin/nologin  
rootfsobody:x:65534:65534:(Anonymous NFS User):/var/lib/nfs/:/sbin/nologin  
ec2-instance-connect:x:997:995:/home/ec2-instance-connect:/sbin/nologin  
postfix:x:89:89:/var/spool/postfix/:/sbin/nologin  
chrony:x:996:994:/var/lib/chrony/:/sbin/nologin
```

Now Integrate Docker with Jenkins:

To integrate Docker with the Jenkins, first thing is we need to create a dedicated user. Of course, we can use the EC2- user but it is not recommended. And make sure that this dockeradmin user is added to the Docker group. I will show you how to do that. Next thing, we would like to copy artifacts from Jenkins to Docker host, right?

For that we need to use Publish Over SSH plugin.

Even we can use SCP or some other plugins but Publish Over SSH is the easiest way to manage it. At last, we need to add Docker host to Jenkins configure systems so that we can able to communicate with the Docker host from the Jenkins.

- 1.Create a dockeradmin user**
- 2. Install “Publish over ssh” plugin**
- 3. Add dockerhost to Jenkins “configure systems”**

cat /etc/passwd (to see what all users we have)
cat /etc/group (to see what all groups we have)

```
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
[root@dockerhost-new ~]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:ec2-user
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:ec2-user
cdrom:x:11:
mail:x:12:postfix
nntp:x:15:
dialin:x:18:
floppy:x:19:
games:x:20:
tapes:x:33:
video:x:39:
fprint:x:40:
lock:x:54:
audio:x:63:
nobody:x:99:
users:x:100:
utmp:x:22:
utmpxfrt:x:36:
inotify:x:99:
systemd-journal:x:108:ec2-user
systemd-network:x:192:
dbus:x:81:
ssh_keys:x:998:
rpc:x:32:
libnsl:x:997:
sshd:x:74:
rngd:x:996:
rprocuser:x:29:
nfsnobody:x:65534:
ec2-identityconnect:x:995:
slabd:x:21:
postdrop:x:98:
postfix:x:89:
chrony:x:994:
stapusr:x:186:
stapsys:x:187:
stapd:x:188:
screen:x:84:
tcpdump:x:72:
ec2-user:x:1000:
cgrid:x:993:
docker:x:992:
[root@dockerhost-new ~]# useradd dockeradmin
[root@dockerhost-new ~]# passwd dockeradmin
Changing password for user dockeradmin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@dockerhost-new ~]# id dockeradmin
uid=1001(dockeradmin) gid=1001(dockeradmin)
[root@dockerhost-new ~]# usermod -aG docker dockeradmin
[root@dockerhost-new ~]# vi /etc/ssh/sshd_config
```

Now we should create a user and we need to add him to the Docker group. Then only he can perform the activities as expected. For that, I'm going to create a user called the dockeradmin.

useradd dockeradmin (create user)
passwd dockeradmin (create password for him)(admin is the password)

Next thing, we need to add this user to Docker group, right? If you check the dockeradmin user, (id dockeradmin) he is belongs to only dockeradmin group, which created by default. But we should add him to the Docker group.

For that, we need to modify this user, usermod is the command, -aG docker. Add the additional group called the Docker for dockeradmin user.

Usermod -aG docker dockeradmin

Now it's time to check that whether we can able to log in.

Let's open duplicate session and let me increase font size. Let me do dockeradmin. We'll see what happens. You can see "Server refused our key" and "Not supported authentication method available."

What it is trying to say is whatever key you have loaded, it is not right key and I could not able to log in, which means that it is not asking for the password.

Why I am trying to show you this one? Because by default, EC2 instances doesn't allow password-based authentication. We should explicitly enable it.

To do that we gotta update /etc/ssh/sshd_config

Vi /etc/ssh/sshd_config

You will get

```
OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naday Exp $  
# This is the sshd server system-wide configuration file. See  
# sshd_config(5) for more information.  
  
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin  
  
# The strategy used for options in the default sshd_config shipped with  
# OpenSSH is to specify options with their default value where  
# possible, but leave them commented. Uncommented options override the  
# default value.  
  
# If you want to change the port on a SELinux system, you have to tell  
# SELinux about this change.  
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER  
#  
#Port 22  
#AddressFamily any  
#ListenAddress 0.0.0.0  
#ListenAddress ::  
  
HostKey /etc/ssh/ssh_host_rsa_key  
HostKey /etc/ssh/ssh_host_dsa_key  
HostKey /etc/ssh/ssh_host_ecdsa_key  
HostKey /etc/ssh/ssh_host_ed25519_key  
  
# Ciphers and keys  
#RekeyLimit default none  
  
# Logging  
#SyslogFacility AUTH  
SyslogFacility AUTHPRIV  
#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  
  
# To disable tunneled clear text passwords, change to no here!  
#PasswordAuthentication yes  
#PermitEmptyPasswords no
```

1,1 Top

Scroll down

```

# 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

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes

# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials no
#GSSAPIStrictAcceptorCheck yes
#GSSAPITKeyExchange no
#GSSAPIEnforceUsers no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication will be bypassed if ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
# WARNING: 'UsePAM no' is not supported in Red Hat Enterprise Linux and may cause several
# problems.
#UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#UseLogin no
#UserPrivilegeSeparation sandbox
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 6
#ClientAliveCountMax 3

```

61,1 62%

```

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no

```

you can see there is a password authentication, yes,
but it is commented out, which means that by default you should not use the password. You
need to go with the key-based authentication, okay?

However, I want to use the password-based authentication. That is the reason I'm just
uncommenting this. And one more entry,

we have password-based authentication, no? This, we need to comment it out
so that we are accepting password-based authentication.

Now you may ask that "Can't I able to log in without password?"

Yes, you can, but for that you need to create SSH keys. I will show you how you can do that
as well.

Now, once you have updated the configuration of this file,
you just need to restart the services,
service sshd reload, okay?

Don't stop and start because if you stop it,
you are going to lose the connection to Docker host
because you are connected with the SSH service itself.

service sshd reload

To login into dockeradmin

sudo su – dockeradmin

```

chrony:x:994:
stapusr:x:156:
stapsys:x:157:
stapdev:x:158:
screen:x:84:
tcpdump:x:72:
ec2-user:x:1000:
crgred:x:993:
docker:x:992:
[root@dockerhost-new ~]# useradd dockeradmin
[root@dockerhost-new ~]# passwd dockeradmin
Changing password for user dockeradmin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@dockerhost-new ~]# id dockeradmin
uid=1001(dockeradmin) gid=1001(dockeradmin) groups=1001(dockeradmin)
[root@dockerhost-new ~]# usermod -aG docker dockeradmin
[root@dockerhost-new ~]# vi /etc/ssh/sshd_config
[root@dockerhost-new ~]# vi /etc/ssh/sshd_config
[root@dockerhost-new ~]# service sshs reload
Redirecting to /bin/systemctl reload sshs.service
Failed to reload sshs.service: Unit not found.
[root@dockerhost-new ~]# service sshd reload
Redirecting to /bin/systemctl reload sshd.service
[root@dockerhost-new ~]#
[root@dockerhost-new ~]#
[root@dockerhost-new ~]# dockeradmin
-bash: dockeradmin: command not found
[root@dockerhost-new ~]# sudo su - dockeradmin
[dockeradmin@dockerhost-new ~]#

```

Now we would like to integrate this Docker host with the Jenkins, right? [For that, we need to install your plugin.](#)

The screenshot shows the Jenkins Plugins page. The left sidebar has tabs for 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area has a search bar with 'publish over ssh' and an 'Install' button. A table lists the 'Publish Over SSH' plugin by version 1.25, released 1 yr 2 mo ago, under the 'Artifact Updaters' and 'Build Tools' categories. The 'Install' checkbox is checked. The bottom right corner shows 'REST API' and 'Jenkins 2.462.2'.

Install	Name ↓	Released
<input checked="" type="checkbox"/>	Publish Over SSH 1.25 Artifact Updaters Build Tools Send build artifacts over SSH	1 yr 2 mo ago

Next,

Dashboard > Manage Jenkins > System >

Publish over SSH

Jenkins SSH Key ?

Passphrase ?
[Text Input Field]

Path to key ?
[Text Input Field]

Key ?
[Text Input Field]

Disable exec ?

SSH Servers

Add

Advanced ▾

GitHub Pull Requests

Click on add servers,

Dashboard > Manage Jenkins > System >

Disable exec ?

SSH Servers

SSH Server

Name ?
dockerhost

Hostname ?
172.31.82.210

Username ?
dockeradmin

Remote Directory ?
[Text Input Field]

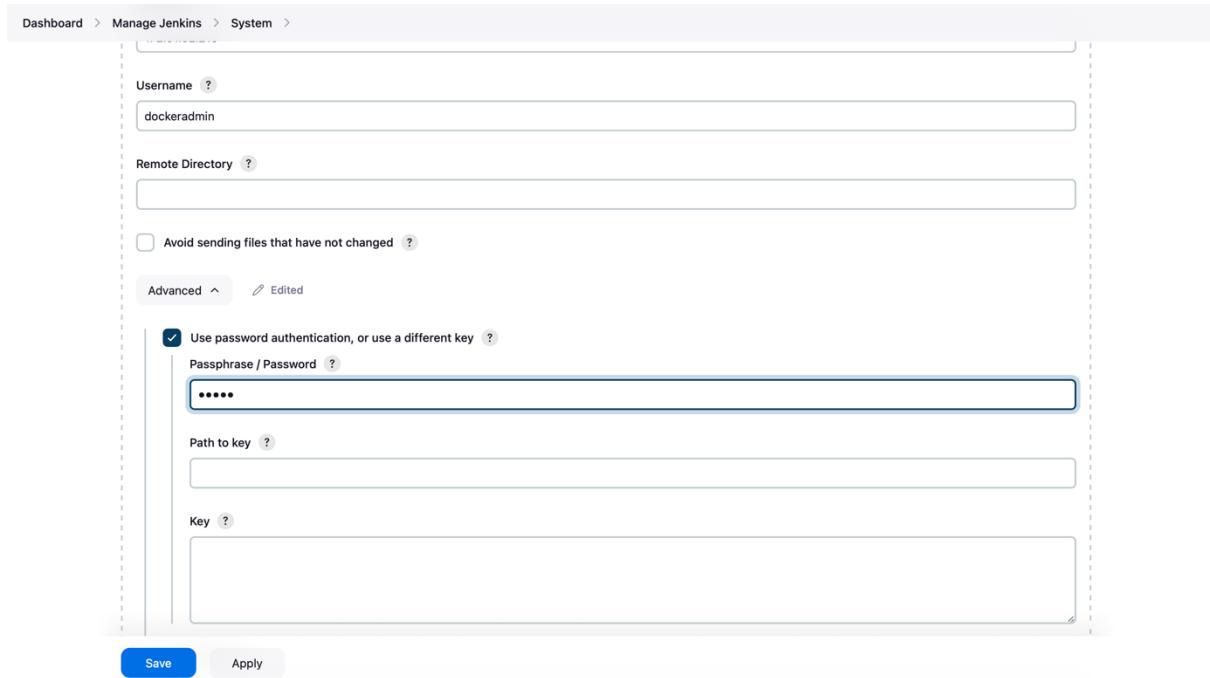
Avoid sending files that have not changed ?

Advanced ▾

Use password authentication, or use a different key ?

Copy paste the dockerhost ec2 – private ip address here in hostname.

Click on advanced, select use password authentication.



And in case if we have the key, we can load the key here. Even we can copy the key into this location. Either way, it'll work. How to get this key? Of course, we can log into our dockeradmin user and we can generate a key. [How to generate the key? ssh-keygen, okay](#)

```

chrony:x:994:
stapusr:x:196:
stapkey:x:197:
stapobj:x:198:
screen:x:84:
tcpdump:x:72:
ec2-user:x:1000:
cgradm:x:992:
[dockeradmin:x:992:
[root@dockerhost-new ~]# useradd dockeradmin
[root@dockerhost-new ~]# passwd dockeradmin
Changing password for user dockeradmin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Re-enter new password:
passwd: all authentication tokens updated successfully.
[root@dockerhost-new ~]# id dockeradmin
uid=1001(dockeradmin) gid=1001(dockeradmin) groups=1001(dockeradmin)
[root@dockerhost-new ~]# usermod -aG docker dockeradmin
[root@dockerhost-new ~]# vi /etc/ssh/sshd_config
[root@dockerhost-new ~]# service sshd reload
[root@dockerhost-new ~]# service ssh reload
Redirecting to /bin/systemctl reload sshs.service
Failed to reload sshs.service: Unit not found.
[root@dockerhost-new ~]# service sshd reload
Redirecting to /bin/systemctl reload sshd.service
[root@dockerhost-new ~]#
[root@dockerhost-new ~]# dockeradmin
-bash: dockeradmin: command not found
[root@dockerhost-new ~]# sudo su - dockeradmin
[dockeradmin@dockerhost-new ~]$
[dockeradmin@dockerhost-new ~]$ ssh-keygen
Generating public/private rsa key pair...
Enter file in which to save the key (/home/dockeradmin/.ssh):
Created directory '/home/dockeradmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identity has been saved in /home/dockeradmin/.ssh/id_rsa.
Your public key has been saved in /home/dockeradmin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tD9duXizw/23uX7bKbHblRpGTMJ0ifK0zaCCXukx5k dockeradmin@dockerhost-new
The key's randomart image is:
+---[RSA 2048]---+
|          |
|          |
|          |
|          |
|          |
|          |
|          |
|          |
|          |
+---[SHA256]---+
[dockeradmin@dockerhost-new ~]$ cd /home/dockeradmin/.ssh
[dockeradmin@dockerhost-new ~]$ ll
total 8
[dockeradmin@dockerhost-new ~]$ cd /home/dockeradmin/.ssh
[dockeradmin@dockerhost-new ~]$ ll
total 8
-rw-r--r-- 1 dockeradmin dockeradmin 1679 Sep 10 23:19 id_rsa
-rw-r--r-- 1 dockeradmin dockeradmin 408 Sep 10 23:19 id_rsa.pub
[dockeradmin@dockerhost-new ~]$ 

```

Now scroll down , click test configuration. Success

Dashboard > Manage Jenkins > System >

Disable exec

Proxy type

Proxy host

Proxy port

Proxy user

Proxy password

Success

Test Configuration

Add

Save

Apply

This screenshot shows the 'System' configuration page in Jenkins. It includes fields for 'Proxy type', 'Proxy host', 'Proxy port', 'Proxy user', and 'Proxy password'. There are also 'Success' and 'Test Configuration' buttons, and standard Jenkins navigation links like 'Add', 'Save', and 'Apply'.

Till now we used Jenkins as build and deploy tool. Now we gonna separate it by ansible

Video 32. we could see Jenkins as a build and deployment tool.

But is there any better way to handle it?

Yes, that is where we can introduce deployment tool.

In our case, we already knew that we are using Git
to update any changes.

Let's keep it aside for some time.

Now I would like to introduce deployment tool.

We are going to use Ansible as a deployment tool,
so that Jenkins need not to do
the administrative kind of activities.

Why?

Because Jenkins is more efficiently work as a build tool.

Along with Ansible we are using Docker Hub.

Our applications are containerized at this moment.

We need a repository.

In case if it is a non containerized applications,
I mean to say which we can deploy directly on VM.

Maybe we can use Nexus or Artifactory.

But we have already reached to the containerization level.

That is the reason we are going to use Docker Hub.

With the help of Ansible and Docker Hub,
we are going to make this CI-CD pipeline more smoother.

So in this case Jenkins is going
to take the code from GitHub
and build artifacts
and copy those artifacts onto Ansible server.

Now, it is Ansible task to create images

and deploy the containers.

Ansible is going to take the artifact

and with help of Docker file it creates a Docker image.

This Docker image we can commit it into the Docker Hub

because Docker hub is a repository to store Docker images.

Now whenever we execute any Ansible playbook to deploy

a container this Docker host communicates with Docker hub,

pull the image whatever we mentioned in our playbook,

and create a container out of it.

PREPARE ANSIBLE SERVER

1. Setup EC2
2. Setup hostname
3. Create ansadmin user
4. Add user to sudoers file
5. Generate ssh keys
6. Enable password based login
7. Install Ansible

Ssh into ansible ec2

vi /etc/hostname

change to ansible-server-new

press esc :wq

init 6

ssh again

sudo su -

useradd ansadmin

passwd ansadmin (password is admin)

Next we gotta add this user to sudoers file

visudo

```
Last login: Tue Sep 18 19:16:37 on ttys007
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT200805.
(base) Nitheesh-MBP:~ apple$ cd downloads
(base) Nitheesh-MBP:downloads apple$ ssh -j devopsproject.pem ec2-user@98.81.255.84
Last login: Wed Sep 11 08:18:02 2024 from 107.139.62.209
# 
` \_ ##### Amazon Linux 2
` \_ ##### AL2 End of Life is 2025-06-30.
` \_ ##### V:-->
` \_ / A newer version of Amazon Linux is available!
` \_ / Amazon Linux 2023, GA and supported until 2028-03-15.
` \_ / https://aws.amazon.com/linux/amazon-linux-2023/
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-83-187 ~]$ sudo su -
Last login: Wed Sep 11 08:18:06 UTC 2024 on pts/0
[root@ip-172-31-83-187 ~]# vi /etc/hostname
[root@ip-172-31-83-187 ~]# hostnamectl
Connection to 98.81.255.84 closed by remote host.
Connection to 98.81.255.84 closed.
(base) Nitheesh-MBP:downloads apple$ ssh -j devopsproject.pem ec2-user@98.81.255.84
Last login: Wed Sep 11 08:23:36 2024 from 107.139.62.209
# 
` \_ ##### Amazon Linux 2
` \_ ##### AL2 End of Life is 2025-06-30.
` \_ ##### V:-->
` \_ / A newer version of Amazon Linux is available!
` \_ / Amazon Linux 2023, GA and supported until 2028-03-15.
` \_ / https://aws.amazon.com/linux/amazon-linux-2023/
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ansible-server-new ~]$ sudo su -
Last login: Wed Sep 11 08:23:33 UTC 2024 on pts/0
[root@ansible-server-new ~]# useradd ansadmin
[root@ansible-server-new ~]# passwd ansadmin
Changing password for user ansadmin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
password successfully updated.
[root@ansible-server-new ~]# visudo
```

You will get something like this

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users or groups.
##
## This file must be edited with the 'visudo' command.

## Host Aliases
## Groups of machines. You may prefer to use hostnames (perhaps using
## aliases for private domains) or IP addresses instead.
# Host_Alias FILESERVERS = fs1, fs2
# Host_Alias MAILSERVERS = smtp, smtp2

## User Aliases
## These aren't often necessary, as you can use regular groups
## (e.g., files, LOAD_NIS, etc) in this file - just use %groupname
## rather than USERALIAS
# User_Alias ADMINS = jsmith, mikem

## Command Aliases
## These are groups of related commands...

## Networking
# Cmd_Alias NETWORKING = /sbin/route, /sbin/ifconfig, /bin/ping, /sbin/dhclient, /usr/bin/net, /sbin/iptables, /usr/bin/rfcomm, /usr/bin/wvdial, /sbin/iwconfig, /sbin/mii-tool

## Installation and management of software
# Cmd_Alias SOFTWARE = /bin/rpm, /usr/bin/up2date, /usr/bin/yum

## Services
# Cmd_Alias SERVICES = /sbin/service, /sbin/chkconfig, /usr/bin/systemctl start, /usr/bin/systemctl stop, /usr/bin/systemctl reload, /usr/bin/systemctl restart, /usr/bin/systemctl status, /usr/bin/systemctl enable, /usr/bin/systemctl disable

## Updating the locate database
# Cmd_Alias LOCATE = /usr/bin/updatedb

## Storage
# Cmd_Alias STORAGE = /sbin/fdisk, /sbin/sfdisk, /sbin/parted, /sbin/partprobe, /bin/mount, /bin/umount

## Delegating permissions
# Cmd_Alias DELEGATING = /usr/sbin/visudo, /bin/chown, /bin/chmod, /bin/chgrp

## Processes
# Cmd_Alias PROCESSES = /bin/nice, /bin/kill, /usr/bin/kill, /usr/bin/killall

## Drivers
# Cmd_Alias DRIVERS = /sbin/modprobe

## Defaults specification

#
# Refuse to run if unable to disable echo on the tty.
#
Defaults !visiblepw

#
# Preserving HOME has security implications since many programs
# search for configuration files. Note that HOME
# is already set when the env_reset option is enabled, so
# this option is only effective for configurations where either
```

2,1 Top

Scroll down and add ansadmin. I'm just adding my user and giving privileges that I can execute any command without any password. That is the meaning of this and save this file.

```

# Prior to version 1.8.15, groups listed in sudoers that were not
# found in the system group database were passed to the group
# plugin, if any. Starting with 1.8.15, only groups of the form
# %group are resolved via the group plugin by default.
# We enable always_query_group_plugin to restore old behavior.
# Disable this option for new behavior.
Defaults    always_query_group_plugin

Defaults    env_reset
Defaults    env_keep = "COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS"
Defaults    env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults    env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
Defaults    env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults    env_keep += "HOME"

Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
## user    MACHINE=COMMANDS
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root      ALL=(ALL)          ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys  ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel    ALL=(ALL)          ALL

## Same thing without a password
# %wheel    ALL=(ALL)          NOPASSWD: ALL
##
## Allows members of the users group to mount and umount the
## cdrom as root
# %users   ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users   localhost:/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includein /etc/sudoers.d
~
~
~
~
~
~
-- INSERT --

```

111,1 Bot

Type here and add this

```

## Same thing without a password
# %wheel    ALL=(ALL)          NOPASSWD: ALL
ansadmin  ALL=(ALL)          NOPASSWD: ALL
## Allows members of the users group to mount and umount the
## cdrom as root
# %users   ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

```

Now we gotta enable password based authentication

`vi /etc/ssh/sshd_config`

```

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no

```

`service sshd reload`

Now we gotta create keys for our ansadmin user.

`Sudo su – ansadmin`

`Ssh-keygen`

It is going to create ssh keys,
while creating our public and private keys.
`/home/ansadmin/.ssh/id_rsa`

```
[lansadmin@ansible-server-new ~]$ ssh-keygen
Generating public/private rsa key pair.
[Enter file in which to save the key (/home/ansadmin/.ssh/id_rsa):
Created directory '/home/ansadmin/.ssh'.
[Enter passphrase (empty for no passphrase):
[Enter same passphrase again:
Your identification has been saved in /home/ansadmin/.ssh/id_rsa.
Your public key has been saved in /home/ansadmin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:C3Wzp9A1fDQD0JKsp91sIx9DhehsjgNp05ayiD4ApQI ansadmin@ansible-server-new
The key's randomart image is:
+---[RSA 2048]---+
|          ..=.o+
| E .      =.o..o |
| .o       ..+o.+..
| +       +..o=+.o |
| o     ..+SBo+.
| .   . =.+o+oB
| ... . + ....+ +
| ... .       .
| ...
+---[SHA256]---+
```

It stores by default under `/home/ansadmin`.

Under that it creates a `.ssh` directory

and Private key stores as `id_rsa`

and publicly stored as `id_rsa.pub`.

Now if you see your identification has been saved

in `/home/ansadmin`, this is my home directory

under `ssh` directory and `id_rsa`.

Now, whatever server we wish to access we need to copy this `id.rsa.pub` key onto target system. In target system, which location we should copy in `users home directory` we will have `.ssh`. Under that we should copy this one onto your authorized keys.

Now we install ansible. So, become your root as your `ansadmin`.

I can become your root. Why? Because, I have root privileges. I mean to say administrative privileges.

```
sudo su -
```

```
yum install ansible (it doesn't work cuz amazon linux system right)
```

(usually it will show “amazon-linux-extras install ansible2” but didn’t show for me, so run the command below)

```
amazon-linux-extras install ansible2 (So its pulling packages from amazon-linux-extras)
```

```

+---[RSA 2048]---+
| E . .+o.. |
| .o ..+o..o |
| + .+o+..o |
| o ..+SBo+ |
| . . =+ooB+ |
| . . + ....+ |
| ... . . |
| ... |
+---[SHA256]---+
[admin@ansible:~]$
[admin@ansible:~]# ssh -v root@192.168.1.11
Last login: Sat Sep 11 00:26:28 UTC 2024 on pts/0
[root@ansible-server-new ~]# yum install ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No package ansible available.
Error: Nothing to do.
[root@ansible-server-new ~]#
[root@ansible-server-new ~]# amazon-linux-extras install ansible2
Topic ansible2 has end-of-support date of 2023-09-30
Installing ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repository amzn2-core amzn2extra-ansible2 amzn2extra-docker amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-ansible2
amzn2extra-docker
amzn2extra-kernel-5.10
(1/9): amzn2-core/2/x86_64/group.gz
(2/9): amzn2-core/2/x86_64/updateinfo
(3/9): amzn2extra-docker/2/x86_64-primary_db
(4/9): amzn2extra-kernel-5.10/2/x86_64-primary_db
(5/9): amzn2extra-ansible2/2/x86_64-primary_db
(6/9): amzn2extra-ansible2/2/x86_64-updateinfo
(7/9): amzn2extra-docker/2/x86_64-primary_db
(8/9): amzn2extra-kernel-5.10/2/x86_64-primary_db
(9/9): amzn2-core/2/x86_64-primary_db
Resolving dependencies
--> Running transaction check
--> Package ansible.noarch 0:2.9.23-1.amzn2 will be installed
--> Processing Dependency: python-crypto for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-httplib2 for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-keyczar for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-paramiko for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: sshpass for package: ansible-2.9.23-1.amzn2.noarch
--> Running transaction check
--> Package python-keyczar.noarch 0:0.71c-2.amzn2 will be installed
--> Package python2-crypto.x86_64 0:2.6.1-1.amzn2.0.3 will be installed
--> Processing Dependency: libtomcrypt.so.1()(64bit) for package: python2-crypto-2.6.1-13.amzn2.0.3.x86_64
--> Package python2-httplib2.noarch 0:0.1.3-3.amzn2 will be installed
--> Package python2-paramiko.noarch 0:2.9.23-1.amzn2.0.3 will be installed
--> Processing Dependency: python-crypto for package: python2-paramiko-1.16.1-3.amzn2.0.3.noarch
--> Package sshpass.x86_64 0:1.06-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package libtomcrypt.x86_64 0:1.18.2-1.amzn2.0.1 will be installed
--> Processing Dependency: libtommath >= 1.0 for package: libtomcrypt-1.18.2-1.amzn2.0.1.x86_64
--> Processing Dependency: libtommath.so.1()(64bit) for package: libtomcrypt-1.18.2-1.amzn2.0.1.x86_64
--> Package python2-ecdsa.noarch 0:0.19.3-1.amzn2.0.1 will be installed

```

For ansible we need python. By default our ec2 comes with python.

To verify,
 python --version
 ansible --version

```

18 libreoffice           available \
 [ =0.6.2.15 =5.3.6.1 =stable ]
19 gimp                 available [ =2.8.22 ]
20 fdocker-latest        enabled
   =17.12.1 =18.03.1 =18.03.1 =18.09.9 =stable ]
21 mate-terminal          available \
 [ =1.19.0 =1.20.0 =stable ]
22 GraphicsMagick1.3     available \
 [ =1.3.29 =1.3.32 =1.3.34 =stable ]
23 epel                  available [ =7.11 =stable ]
24 testing               available [ =1.0 =stable ]
25 ces                   available [ =stable ]
27 correcto8             available \
 [ =1.8.0.192 =1.8.0.202 =1.8.0.212 =1.8.0.222 =1.8.0.232
   =1.8.0.242 =stable ]
32 lustre2.10            available \
 [ =1.10.5 =2.10.8 =stable ]
33 java-openjdk11         available [ =11 =stable ]
34 lynis                available [ =stable ]
36 BCC                  available [ =8.x =stable ]
37 mono                 available [ =5.x =stable ]
38 nginx1              available [ =stable ]
40 cockpit              available [ =stable ]
43 lsofpatch             available [ =stable ]
44 python3.8              available [ =stable ]
45 haproxy2              available [ =stable ]
46 collected             available [ =stable ]
47 aws-nitro-enclaves-cli available [ =stable ]
48 kernel                available [ =stable ]
   =kernel-5.4
50 selinux-ng             available [ =stable ]
52 tomcat9              available [ =stable ]
53 unbound1.13            available [ =stable ]
54 mariadb10.5             available [ =stable ]
55 mariadb10.5-latest    enabled [ =stable ]
56 redis6                available [ =stable ]
58 postgresql12            available [ =stable ]
59 postgresql13            available [ =stable ]
60 mock2                 available [ =stable ]
61 dnsmasq2.85             available [ =stable ]
62 kernel-5.15             available [ =stable ]
63 mariadb10.5             available [ =stable ]
64 firefox               available [ =stable ]
65 lustre                available [ =stable ]
66 rphb8.1                available [ =stable ]
67 awsccli               available [ =stable ]
68 rphb8.2                available [ =stable ]
69 amznssqd              available [ =stable ]
70 unbound1.17            available [ =stable ]
72 collectd-python3       available [ =stable ]
* Extra topic has reached end of support.
† Note on end-of-support. Use 'info' subcommand.
[root@ansible-server-new ~]#
[root@ansible-server-new ~]# python --version
Python 2.7.18
[root@ansible-server-new ~]# ansible --version
ansible 2.9.23
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/usr/share/ansible/plugins/modules', u'/usr/lib/python2.7/site-packages/ansible']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Dec 18 2023, 22:08:43) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[root@ansible-server-new ~]# []

```

Now next thing is, we need to prepare our Ansible system to create docker images. For that, of course we need to install docker and do some additional configuration.

Manage Dockerhost with Ansible. Video – 34

On Docker Host:

- 1.Create ansadmin
- 2.Add ansadmin to sudoers files
- 3.Enable password based login

On Ansible Node:

1. Add to hosts file
2. Copy ssh keys
3. Test the connection

In this lecture,

we are going to add our DockerHost to Ansible as a managed node.

So that our Ansible control node can enable to manage our DockerHost.

Why we need to do that one?

Because this image is built. We need to initialize our DockerHost to create a container.

Right?

That we are going to do with the Ansible control node.

I mean to say we are going to write here playbook. That playbook is going to tell to our DockerHost, how to create a container.

For that of course, we need to add our DockerHost as a client Or slave to our Ansible system. To enable these steps, we need to do some activities on DockerHost.

First if we see, what steps we need to do on our Docker host,

we need to create a ansadmin. Then add this ansadmin to sudoers files, and enable password based authentication. What we have done on Ansible system, same steps. Why because with the ansadmin user itself, Ansible is going to manage our DockerHost. Once this is done, our docker system is ready to managed by Ansible. For that on Ansible node,

we need to add DockerHost IP address in the inventory file, nothing but hosts file.

Next, we need to copy ssh keys. We will be copying Ansible server public key onto DockerHost ansadmin user, so that passwordless authentication get enabled.

At last we need to test the connection for that. Maybe we can execute some ping command. or finding out the uptime of our Docker system to validate it.

Let's go and configure our DockerHost system first.

Go to Dockerhost EC2.

Become root. cd ~

```
Sudo su –  
useradd ansadmin  
passwd ansadmin (password is admin)
```

```
visudo
```

```
## Same thing without a password
# %wheel      ALL=(ALL)      NOPASSWD: ALL
ansadmin  ALL=(ALL)      NOPASSWD: ALL
## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
```

Esc :wq save it

Now we gotta enable password based authentication. But remember we have already done this.

Now we need to enable password based authentication,
but we have already done that.

If you want to check, you can just grep Password.

I'm just gripping this command, slash etc ssh sshd config.

What exactly does it do?

It looks for password pattern.

Nothing but wherever this word is there in this file.

It list out all the lines and some of them are.

You can see password authentication we have already enabled,
and password authentication now is disabled.

I mean to say it is commented out.

We have enabled it at the time

of integrating with Jenkins right?

All right I can say required configuration changes
has been done on DockerHost.

If something is needed, we will come back over here.

Let's jump back to our Ansible server.

To check

```
grep Password /etc/ssh/sshd_config
```

```
[[root@dockerhost-new opt]#
[[root@dockerhost-new opt]# cd ~
[[root@dockerhost-new ~]#
[[root@dockerhost-new ~]# useradd ansadmin
[[root@dockerhost-new ~]# passwd ansadmin
Changing password for user ansadmin.
[New password:
BAD PASSWORD: The password is shorter than 8 characters
[Retype new password:
passwd: all authentication tokens updated successfully.
[[root@dockerhost-new ~]#
[[root@dockerhost-new ~]# visudo
[[root@dockerhost-new ~]# grep Password /etc/ssh/sshd_config
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication, then enable this but set PasswordAuthentication
[root@dockerhost-new ~]# ]]
```

2.Now on Ansible server

This is our Ansible server. Here, we need to add our DockerHost as a managed node. Managed node nothing, but which is managed by Ansible. Those we call as a managed nodes. And we can do that in the inventory, that is default inventory is available on etc Ansible hosts. Okay? This is the default inventory file. By default it has some information, how to define our hosts information. But anyway, we don't record it. Let's delete it. And a let's add our DockerHost IP address.

Vi /etc/ansible/hostname

```
● ○ ● ● Downloads — root@ansible-server-new:~ — ssh -i devopsproject.pem ec2-user@98.81.255.84 — 115x35
50 selinux-ng      available  [ =stable ]
52 tomcat9        available  [ =stable ]
53 unbound1.13    available  [ =stable ]
54 tmariadb10.5   available  [ =stable ]
55 kernel-5.10=latest enabled   [ =stable ]
56 redis6         available  [ =stable ]
58 tpostgresql12  available  [ =stable ]
59 tpostgresql13  available  [ =stable ]
60 mock2          available  [ =stable ]
61 dnsmasq2.85   available  [ =stable ]
62 kernel-5.15   available  [ =stable ]
63 tpostgresql14  available  [ =stable ]
64 firefox         available  [ =stable ]
65 lustre          available  [ =stable ]
66 tphp8.1        available  [ =stable ]
67 awscli1        available  [ =stable ]
68 tphp8.2        available  [ =stable ]
69 dnsmasq         available  [ =stable ]
70 unbound1.17    available  [ =stable ]
72 collectd-python3 available  [ =stable ]
* Extra topic has reached end of support.
† Note on end-of-support. Use 'info' subcommand.
[[root@ansible-server-new ~]#]
[[root@ansible-server-new ~]# python --version
Python 2.7.18
[[root@ansible-server-new ~]# ansible --version
ansible 2.9.23
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Dec 18 2023, 22:08:43) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[[root@ansible-server-new ~]#
[[root@ansible-server-new ~]# vi /etc/ansible/hosts
[[root@ansible-server-new ~]# ]]
```

Go to docker ec2

`ipconfig` (you will get its private ip or take it from aws console)

```

Downloads — root@dockerhost-new:~ — ssh -i devopsproject.pem ec2-user@44.211.221.22 — 118x36
[root@dockerhost-new ~]# visudo
[root@dockerhost-new ~]# grep Password /etc/ssh/sshd_config
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication, then enable this but set PasswordAuthentication
[root@dockerhost-new ~]#
[root@dockerhost-new ~]# ifconfig
docke0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
        ether 02:42:e1:d3:56:f8 txqueuelen 0 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.82.210 netmask 255.255.240.0 broadcast 172.31.95.255
        inet6 fe80::10ff:59ff:fe5b:2c7 prefixlen 64 scopeid 0x20<link>
            ether 12:ff:59:b0:02:c7 txqueuelen 1000 (Ethernet)
            RX packets 71240 bytes 95484272 (91.0 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 12126 bytes 1088909 (1.0 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@dockerhost-new ~]# 

```

```

Downloads — root@ansible-server-new:~ — ssh -i devopsproject.pem ec2-user@98.81.255.84 — 115x35
172.31.82.210

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

~  

-- INSERT --          1,14          All

```

Now we need to copy our ansadmin user keys onto target ansadmin user account.
For that, we should switch onto a ansadmin user.
Okay?

I'm going back to ansadmin user, and if I check for my keys, it should be available in our home directory.

Okay?

Let's go inside ssh.

And if you see, this is private key and this is public key.

Sudo su – ansadmin

ll -la (cuz .ssh is hidden)

cd .ssh

ll

```
○ ○ ○ Downloads — ansadmin@ansible-server-new:~/ssh — ssh -i devopsproject.pem ec2-user@98.81.255.84 — 115...
68 tphp8.2           available  [ =stable ]
69 dnsmasq          available  [ =stable ]
70 unbound1.17       available  [ =stable ]
72 collectd-python3  available  [ =stable ]
* Extra topic has reached end of support.
† Note on end-of-support. Use 'info' subcommand.
[[root@ansible-server-new ~]#
[[root@ansible-server-new ~]# python --version
Python 2.7.18
[[root@ansible-server-new ~]# ansible --version
ansible 2.9.23
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Dec 18 2023, 22:08:43) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[[root@ansible-server-new ~]#
[[root@ansible-server-new ~]# vi /etc/ansible/hosts
[[root@ansible-server-new ~]#
[[root@ansible-server-new ~]# sudo su - ansadmin
Last login: Wed Sep 11 00:38:00 UTC 2024 on pts/0
[[ansadmin@ansible-server-new ~]$ ll -la
total 12
drwx----- 3 ansadmin ansadmin 74 Sep 11 00:38 .
drwxr-xr-x  4 root      root    38 Sep 11 00:26 ..
-rw-r--r--  1 ansadmin ansadmin 18 Jul 15 2020 .bash_logout
-rw-r--r--  1 ansadmin ansadmin 193 Jul 15 2020 .bash_profile
-rw-r--r--  1 ansadmin ansadmin 231 Jul 15 2020 .bashrc
drwx----- 2 ansadmin ansadmin 38 Sep 11 00:38 .ssh
[[ansadmin@ansible-server-new ~]$ cd .ssh
[[ansadmin@ansible-server-new .ssh]$ ll
total 8
-rw----- 1 ansadmin ansadmin 1679 Sep 11 00:38 id_rsa
-rw-r--r-- 1 ansadmin ansadmin 409 Sep 11 00:38 id_rsa.pub
[ansadmin@ansible-server-new .ssh]$
```

We need to keep our private key safe.

Why because if somebody have this private key, they can able to log into our system as a ansadmin.

Anyway, we have a simple command

to copy this public key under target system, that is ssh copy id and target system IP address.

We already got the IP address, right?

This is the private IP.

I'm going to use private IP.

Why because these all are in the same VPC.

If you are aware of VPC concept,

```
cd .. (go back from .ssh)
```

```
ssh-copy-id <private ip of dockerhost>
```

It asks password for ansadmin (I guess it's the ansadmin created on dockerhost, cuz check that ip address)

```
Downloads — ansadmin@ansible-server-new:~ — ssh -i devopsproject.pem ec2-user@98.81.255.84 — 115x35
[root@ansible-server-new ~]#
[root@ansible-server-new ~]# sudo su - ansadmin
Last login: Wed Sep 11 00:38:00 UTC 2024 on pts/0
[ansadmin@ansible-server-new ~]$ ll -la
total 12
drwx----- 3 ansadmin ansadmin 74 Sep 11 00:38 .
drwxr-xr-x 4 root      root   38 Sep 11 00:26 ..
-rw-r--r-- 1 ansadmin ansadmin 18 Jul 15 2020 .bash_logout
-rw-r--r-- 1 ansadmin ansadmin 193 Jul 15 2020 .bash_profile
-rw-r--r-- 1 ansadmin ansadmin 231 Jul 15 2020 .bashrc
drwx----- 2 ansadmin ansadmin 38 Sep 11 00:38 .ssh
[ansadmin@ansible-server-new ~]$ cd .ssh
[ansadmin@ansible-server-new .ssh]$ ll
total 8
-rw----- 1 ansadmin ansadmin 1679 Sep 11 00:38 id_rsa
-rw-r--r-- 1 ansadmin ansadmin 409 Sep 11 00:38 id_rsa.pub
[ansadmin@ansible-server-new .ssh]$
[ansadmin@ansible-server-new .ssh]$ cd ..
[ansadmin@ansible-server-new ~]$ ssh-copy-id 172.31.82.210
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host '172.31.82.210 (172.31.82.210)' can't be established.
ECDSA key fingerprint is SHA256:YtqJHXQgFpKNwDNnK5WsGoUQyNjtzZTKOjzz6hSfXYM.
ECDSA key fingerprint is MD5:e1:fe:01:27:3b:e4:15:ae:44:9e:12:bc:b6:fe:60:ee.
Are you sure you want to continue connecting (yes/no)? 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 key
s
ansadmin@172.31.82.210's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '172.31.82.210'"
and check to make sure that only the key(s) you wanted were added.

[ansadmin@ansible-server-new ~]$
```

To verify, go to docker host.

if I go and check it on ansadmin,

ll - la, okay? Because .ssh a hidden file go inside to this directory.

```
cd .ssh
```

Now you can see here, there is a authorized keys file.

```
sudo su – ansadmin
```

```
ll -la
```

```
cd .ssh
```

```
ll
```

```

Downloads — ansadmin@dockerhost-new:~/ssh — ssh -i devopsproject.pem ec2-user@44.211.221.22 — 118x36
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12126 bytes 1088909 (1.0 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@dockerhost-new ~]#
[root@dockerhost-new ~]#
[root@dockerhost-new ~]# sudo su - ansadmin
[ansadmin@dockerhost-new ~]$ ll
total 0
[ansadmin@dockerhost-new ~]$ ll -la
total 12
drwx----- 3 ansadmin ansadmin 74 Sep 11 01:29 .
drwxr-xr-x 5 root      root   57 Sep 11 01:03 ..
-rw-r--r-- 1 ansadmin ansadmin 18 Jul 15 2020 .bash_logout
-rw-r--r-- 1 ansadmin ansadmin 193 Jul 15 2020 .bash_profile
-rw-r--r-- 1 ansadmin ansadmin 231 Jul 15 2020 .bashrc
drwx----- 2 ansadmin ansadmin 29 Sep 11 01:29 .ssh
[ansadmin@dockerhost-new ~]$ cd .ssh
[ansadmin@dockerhost-new .ssh]$ ll
total 4
-rw----- 1 ansadmin ansadmin 409 Sep 11 01:29 authorized_keys
[ansadmin@dockerhost-new .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQD3w0iY/Mv8/XqedNS8qJRaLnTMSPjFaP6Nne2WWPA4g/JV3m0t7kkg8xfbonpRE+CdTtH9S5FM/T2lt
2jNgmNgha3j4eeeHC6L3Uhf7sJt8gy1b8uGX1hRPqUigpuZNf38UqP/H3U0D11xUQVQz3XYxt3rokX/uqGd5Q8Uax5cL4BaEZdCDAYtGr15iP+ufV4AJ
1LgM5WEVynzb94brG1an1BmWHQ04Fyeeg0UzxXrOsziokMWeVWKBQ1AL/d9ZQh72m5JM7hTVW5K27ujEF8x1/fT9ZqbLI57qrCZjtfCrBNPF6fKeuGPf
1pJ001Wa4ONkCc2Ap3bwaBDAjf ansadmin@ansible-server-new
[ansadmin@dockerhost-new .ssh]$ 

```

and if I open this one
you can see the key of the Ansible server.

cat authorized_keys (on dockerhost)

And even if I open the same key over here on ansible server.

cat .ssh/id_rsa.pub

```

Downloads — ansadmin@ansible-server-new:~ — ssh -i devopsproject.pem ec2-user@98.81.255.84 — 141x40
[root@ansible-server-new ~]# vi /etc/ansible/hosts
[root@ansible-server-new ~]#
[root@ansible-server-new ~]# sudo su - ansadmin
Last login: Wed Sep 11 00:38:00 UTC 2024 on pts/0
[ansadmin@ansible-server-new ~]$ ll -la
total 12
drwx----- 3 ansadmin ansadmin 74 Sep 11 00:38 .
drwxr-xr-x 4 root      root   38 Sep 11 00:26 ..
-rw-r--r-- 1 ansadmin ansadmin 18 Jul 15 2020 .bash_logout
-rw-r--r-- 1 ansadmin ansadmin 193 Jul 15 2020 .bash_profile
-rw-r--r-- 1 ansadmin ansadmin 231 Jul 15 2020 .bashrc
drwx----- 2 ansadmin ansadmin 38 Sep 11 00:38 .ssh
[ansadmin@ansible-server-new ~]$ cd .ssh
[ansadmin@ansible-server-new .ssh]$ ll
total 8
-rw----- 1 ansadmin ansadmin 1679 Sep 11 00:38 id_rsa
-rw-r--r-- 1 ansadmin ansadmin 409 Sep 11 00:38 id_rsa.pub
[ansadmin@ansible-server-new .ssh]$ cd ..
[ansadmin@ansible-server-new ~]$ ssh-copy-id 172.31.82.210
/usr/bin/ssh-copy-id: INFO: Source of key_id to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host '172.31.82.210 (172.31.82.210)' can't be established.
ECDSA key fingerprint is SHA256:tqJHXQgFpNWDNnK6WsGoUQyNjtZTKoJzz6hSfxYXM.
ECDSA key fingerprint is MD5:e1:f0:01:27:3b:a4:15:a0:44:9e:12:bc:b6:fe:60:ee.
Are you sure you want to continue connecting (yes/no)? 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
ansadmin@172.31.82.210's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '172.31.82.210'"
and check to make sure that only the key(s) you wanted were added.

[ansadmin@ansible-server-new ~]$ 
[ansadmin@ansible-server-new ~]$ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQD3w0iY/Mv8/XqedNS8qJRaLnTMSPjFaP6Nne2WWPA4g/JV3m0t7kkg8xfbonpRE+CdTtH9S5FM/T2lt2jNgmNgha3j4eeeHC6L3Uhf
7sJt8gy1b8uGX1hRPqUigpuZNf38UqP/H3U0D11xUQVQz3XYxt3rokX/uqGd5Q8Uax5cL4BaEZdCDAYtGr15iP+ufV4AJ1LgM5WEVynzb94brG1an1BmWHQ04Fyeeg0UzxXrOsziok
MWeVWKBQ1AL/d9ZQh72m5JM7hTVW5K27ujEF8x1/fT9ZqbLI57qrCZjtfCrBNPF6fKeuGPf1pJ001Wa4ONkCc2Ap3bwaBDAjf ansadmin@ansible-server-new
[ansadmin@ansible-server-new ~]$ 

```

So, Public key (from ansible), We have copied onto remote system. (dockerhost)

This is the public key.

Same key.

`cat .ssh/id_rsa → (on ansible server, this will show the private key but remember you gotta keep this safe)`

Anyway, let's switch back to home directory.

One last thing,

we need to validate that

whether I can able to connect to my remote system.

I mean to say, manage the system without any issues or not.

For that, we can just execute Ansible all minus m ping.

Okay, so what does it mean that, all nothing,

but whatever hosts are there in the inventory file.

Try to connect to all those systems.

We have only one system right in the host file.

It connects to the system, and if you are able to connect,
you will get this successful message.

If you are not able to connect,

you can see the failed misses.

Anyway, the connection is established successfully.

To find out uptime,

we need to give Ansible all - m command, minus a uptime.

Okay?

Instead of a, I have given m,

but we can ignore this minus m command as well.

If you don't give minus m command also by default,
it's going to take command module.

Okay, let's execute.

And you can see,

and it is up from five hours, 38 minutes,

two users has been logged in.

`ansible all -m ping`

`ansible all -m command -a uptime`

```

Downloads — ansadmin@ansible-server-new:~ — ssh -i devopsproject.pem ec2-user@98.81.255.84 — 141x29

Now try logging into the machine, with: "ssh '172.31.82.210'"
and check to make sure that only the key(s) you wanted were added.

[ansadmin@ansible-server-new ~]$ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQABAAQD3w0iY/Mv8/XqedNS8qJRaLnTMSPjFaP6Nne2WPA4g/JV3m0t7kkg8xfbonpRE+CdTtH9S5FM/T2lt2jNgmNgHa3j4eeeHC6L3Uhf7sBgy1b8uGx1hRPqUigpuZNf38UqP/H3UOD11xUQVQz3XYxt3rokX/uqGd5Q8Uax5cL4BaEZdCDAYtGr15iP+ufV4AJ1gM5WEVyNzbd94brG1an1BmWHQ4Fyeeg0UzxXrOsziokMWeVWKbQ1AL/d9ZQh72m5JM7hTVW5K27ujEF8x1/fT9ZqbLI57qrCZjfdCrBNPF6fKeuGPf1pJ001Wa4ONkCc2Ap3bwA0DAjf ansadmin@ansible-server-new
[ansadmin@ansible-server-new ~]$ cd ~
[ansadmin@ansible-server-new ~]$ ansible all -m ping
[WARNING]: Platform linux on host 172.31.82.210 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
172.31.82.210 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[ansadmin@ansible-server-new ~]$ ansible all -m command -a uptime
[WARNING]: Platform linux on host 172.31.82.210 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
172.31.82.210 | CHANGED | rc=0 >>
 01:50:33 up 3:16, 2 users, load average: 0.00, 0.00, 0.00
[ansadmin@ansible-server-new ~]$ 

```

Even you can check same thing over here.

Let me come out from here and uptime. If I execute, you can see the same output.

Alright? Now I can say that our Ansible could able to communicate with our DockerHost, without any credentials Nothing but with Passwordless authentication.

So that I can instruct my DockerHost from Ansible to do the activities.

Cd ..

Uptime

```

Downloads — ansadmin@dockerhost-new:~ — ssh -i devopsproject.pem ec2-user@44.211.221.22 — 118x15

drwx----- 2 ansadmin ansadmin 29 Sep 11 01:29 .ssh
[[ansadmin@dockerhost-new ~]$ cd .ssh
[[ansadmin@dockerhost-new .ssh]$ ll
total 4
-rw----- 1 ansadmin ansadmin 409 Sep 11 01:29 authorized_keys
[[ansadmin@dockerhost-new .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAQABAAQD3w0iY/Mv8/XqedNS8qJRaLnTMSPjFaP6Nne2WPA4g/JV3m0t7kkg8xfbonpRE+CdTtH9S5FM/T2lt2jNgmNgHa3j4eeeHC6L3Uhf7sBgy1b8uGx1hRPqUigpuZNf38UqP/H3UOD11xUQVQz3XYxt3rokX/uqGd5Q8Uax5cL4BaEZdCDAYtGr15iP+ufV4AJ1gM5WEVyNzbd94brG1an1BmWHQ4Fyeeg0UzxXrOsziokMWeVWKbQ1AL/d9ZQh72m5JM7hTVW5K27ujEF8x1/fT9ZqbLI57qrCZjfdCrBNPF6fKeuGPf1pJ001Wa4ONkCc2Ap3bwA0DAjf ansadmin@ansible-server-new
[[ansadmin@dockerhost-new ~]$ 
[[ansadmin@dockerhost-new ~]$ cd ..
[[ansadmin@dockerhost-new ~]$ uptime
 01:50:52 up 3:17, 1 user, load average: 0.00, 0.00, 0.00
[ansadmin@dockerhost-new ~]$ 

```

Video – 35

Integrate Ansible with Jenkins

In this lecture, we are going to see how to integrate Ansible with Jenkins so that Jenkins can able to copy artifacts onto Ansible systems,

Ansible can able to create image or it can deploy the containers on Docker host.

By doing this one, we can delegate the activities, like Jenkins can able to do only build activities, Ansible can able to take care of deployment activities.

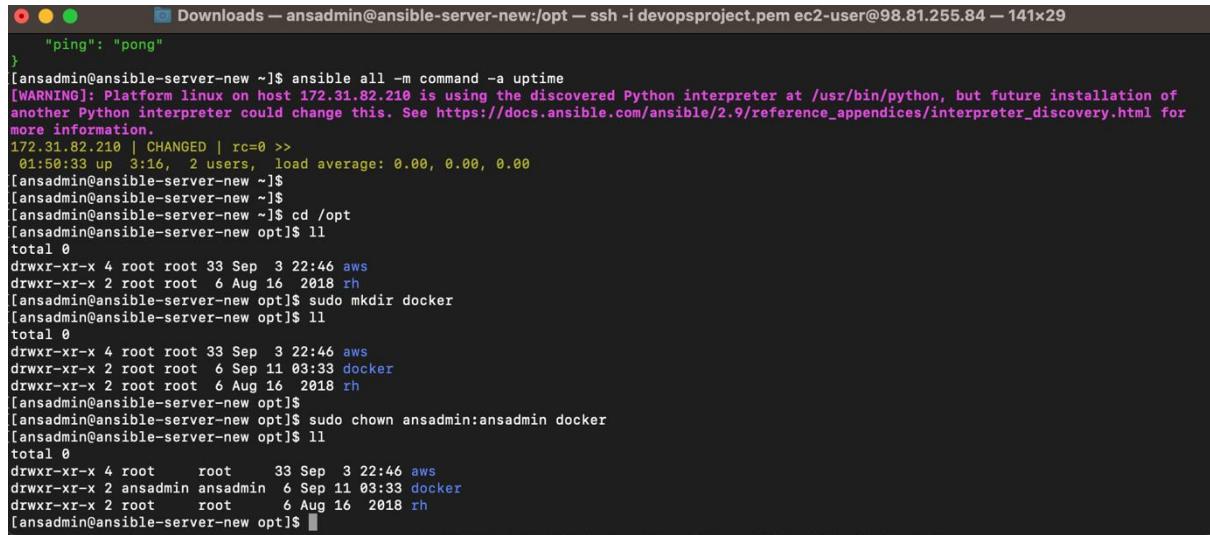
In Jenkins GUI, Manage Jenkins , Add publish over ssh server.

The screenshots show the Jenkins 'Manage Jenkins > System' configuration interface. The top section, titled 'SSH Server', contains fields for 'Name' (set to 'ansible-server-new'), 'Hostname' (set to '172.31.83.187'), 'Username' (set to 'ansadmin'), and 'Remote Directory'. An 'Advanced' section is expanded, showing a checked checkbox for 'Use password authentication, or use a different key?' and a password field containing '*****'. The bottom section, titled 'Proxy', contains fields for 'Proxy type' (a dropdown menu), 'Proxy host', 'Proxy port', 'Proxy user', and 'Proxy password'. A 'Success' message is displayed above the 'Test Configuration' button. Both sections have 'Save' and 'Apply' buttons at the bottom.

Let's go and create a directory called **docker** in ansible server, so whenever the job runs the war file or artificatory will get deposited here.

I'm jumping onto "cd/opt", and "mkdir docker". I don't have privileges, let me add sudo command "mkdir docker". Now if I check "LL", you can see Docker has been created but it is run by the root user. If it is run by the root user, while copying artifacts from Jenkins server, it uses ANS admin, right? So you will get here permission issue. To overcome that one, we must grant access to this directory to ANS admin user. For that, we should change the ownership. Sudo, CH own, ANS admin, ANS admin, onto Docker. Okay? This is how we can change the ownership of your directory. It is run by the ANS admin user now.

```
Cd /opt  
Ll  
Sudo mkdir docker  
Ll  
Sudo chown ansadmin:ansadmin docker  
Ll
```



```
Downloads — ansadmin@ansible-server-new:/opt — ssh -i devopsproject.pem ec2-user@98.81.255.84 — 141x29  
} "ping": "pong"  
[ansadmin@ansible-server-new ~]$ ansible all -m command -a uptime  
[WARNING]: Platform linux on host 172.31.82.210 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.  
172.31.82.210 | CHANGED | rc=0 >>  
 01:50:33 up 3:16, 2 users, load average: 0.00, 0.00, 0.00  
[ansadmin@ansible-server-new ~]$  
[ansadmin@ansible-server-new ~]$  
[ansadmin@ansible-server-new ~]$ cd /opt  
[ansadmin@ansible-server-new opt]$ ll  
total 0  
drwxr-xr-x 4 root root 33 Sep 3 22:46 aws  
drwxr-xr-x 2 root root 6 Aug 16 2018 rh  
[ansadmin@ansible-server-new opt]$ sudo mkdir docker  
[ansadmin@ansible-server-new opt]$ ll  
total 0  
drwxr-xr-x 4 root root 33 Sep 3 22:46 aws  
drwxr-xr-x 2 root root 6 Sep 11 03:33 docker  
drwxr-xr-x 2 root root 6 Aug 16 2018 rh  
[ansadmin@ansible-server-new opt]$  
[ansadmin@ansible-server-new opt]$ sudo chown ansadmin:ansadmin docker  
[ansadmin@ansible-server-new opt]$ ll  
total 0  
drwxr-xr-x 4 root      root    33 Sep 3 22:46 aws  
drwxr-xr-x 2 ansadmin ansadmin 6 Sep 11 03:33 docker  
drwxr-xr-x 2 root      root    6 Aug 16 2018 rh  
[ansadmin@ansible-server-new opt]$
```

In this video, Jenkins was able to copy the artifact or war file onto ansible in /opt/docker. In next video we will create image out of this war file.

Which means that we could able to successfully copy the artifacts onto Ansible system.

Now our Ansible job is creating an image

by using this WAR file

and push it into the Docker hub.

Now you may ask that

why we need to push image onto Docker hub?

You can see the problem going forward

to copy images onto different systems.

Then obviously it enforce us to use the Docker hub.

However, for now,

we could able to get artifacts onto Ansible system.

Now it's time to create image out of this

on Ansible system.

Video - 36

In previous lecture, we have integrated our Ansible with Jenkins and copied artifacts by using Jenkins job. Now it's time to create your docker image with help of the artifacts which we have copied onto Ansible.

This activity again we have done previously on our docker host.

Same thing we are going to do.

For that, of course we need to install docker on our Ansible system.

Cat /etc/group (no docker group)

Currently we are in the docker directory we created, don't know why he installed docker from there itself and not in ~

Sudo yum install docker -y

Cat /etc/group (you can see docker group)

```
[ansadmin@ansible-server-new opt]$ [ansadmin@ansible-server-new opt]$ sudo chown ansadmin:ansadmin docker
[ansadmin@ansible-server-new opt]$ ls
total 0
drwxr-xr-x 4 root root 33 Sep 3 22:46 aws
drwxr-xr-x 2 ansadmin ansadmin 6 Sep 11 03:38 docker
drwxr-xr-x 1 root root 16 Aug 16 2018 rh
[ansadmin@ansible-server-new opt]$ [ansadmin@ansible-server-new opt]$ cd docker/
[ansadmin@ansible-server-new docker]$ sudo yum install docker -y
[Loaded: plugins=extras_suggestions, langpacks, priorities, update-motd
amzn2-core]
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:25.0.6-1.amzn2.0.2 will be installed
--> Processing Dependency: containerd >= 1.3.2 for package: docker-25.0.6-1.amzn2.0.2.x86_64
--> Processing Dependency: libcgroup >= 0.46.rcl-5.15 for package: docker-25.0.6-1.amzn2.0.2.x86_64
--> Processing Dependency: runc >= 1.0.0 for package: docker-25.0.6-1.amzn2.0.2.x86_64
--> Processing Dependency: pigz for package: docker-25.0.6-1.amzn2.0.2.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.7.20-1.amzn2.0.1 will be installed
--> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.1.13-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository        Size
=====
Installing:
docker            x86_64    25.0.6-1.amzn2.0.2   amzn2extra-docker  45 M
Installing for dependencies:
containerd         x86_64    1.7.20-1.amzn2.0.1   amzn2extra-docker  38 M
libcgroup          x86_64    0.41-21.amzn2       amzn2-core          66 k
pigz              x86_64    2.3.4-1.amzn2.0.1   amzn2-core          81 k
runc              x86_64    1.1.13-1.amzn2      amzn2extra-docker  3.1 M

Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 78 M
Installed size: 285 M
Downloading packages:
(1/5): libcgroup-0.41-21.amzn2.x86_64.rpm                                | 66 kB  00:00:00
(2/5): pigz-2.3.4-1.amzn2.0.1.x86_64.rpm                                    | 81 kB  00:00:00
(3/5): containerd-1.7.20-1.amzn2.0.1.x86_64.rpm                           | 30 MB   00:00:00
(4/5): docker-25.0.6-1.amzn2.0.2.x86_64.rpm                               | 45 MB   00:00:00
(5/5): runc-1.1.13-1.amzn2.x86_64.rpm                                       | 3.1 MB  00:00:00

Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing: runc-1.1.13-1.amzn2.x86_64
  Installing : containerd-1.7.20-1.amzn2.0.1.x86_64
  Installing : pigz-2.3.4-1.amzn2.0.1.x86_64
  Installing : libcgroup-0.41-21.amzn2.x86_64
  Installing : docker-25.0.6-1.amzn2.0.2.x86_64
  Verifying  : runc-1.1.13-1.amzn2.x86_64
                                                               1/5
                                                               2/5
                                                               3/5
                                                               4/5
                                                               5/5
                                                               1/5
```

```
[Installed:
 docker.x86_64 0:25.0.6-1.amzn2.0.2
Dependency Installed:
 containerd.x86_64 0:1.7.20-1.amzn2.0.1  libcgroup.x86_64 0:0.41-21.amzn2  pigz.x86_64 0:2.3.4-1.amzn2.0.1  runc.x86_64 0:1.1.13-1.amzn2

Completed!
[ansadmin@ansible-server-new docker]$ cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:ec2-user
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:ec2-user
cdrom:x:11:
mail:x:12:postfix
man:x:15:
dialout:x:18:
floppy:x:19:
games:x:20:
tts:x:23:
videotx:x:39:
ftor:x:80:
lock:x:54:
audio:x:63:
nobody:x:99:
users:x:100:
utmp:x:22:
utempter:x:35:
input:x:99:
systemd-journal:x:198:ec2-user
systemd-network:x:192:
dm:x:81:
ssh:x:998:
rpc:x:32:
libstoragemgmt:x:997:
sshd:x:74:
rndis:x:995:
rpctester:x:29:
netfilterdyi:x:165534:
ec2-instance-connect:x:995:
slocate:x:21:
postdrop:x:98:
postfix:x:89:
crond:x:73:
stapusr:x:156:
stapsys:x:157:
stadev:x:158:
screen:x:84:
tcpdump:x:72:
tcpdump:x:1000:
ec2-ssm:x:1001:
ansadmin:x:1001:
cgred:x:993:
docker:x:992:
[ansadmin@ansible-server-new docker]$ sudo usermod -G docker ansadmin
[ansadmin@ansible-server-new docker]$ id ansadmin
uid=1001(ansadmin) gid=1001(ansadmin) groups=1001(ansadmin),992(docker)
[ansadmin@ansible-server-new docker]$ systemctl status docker
```

Now I should add our ansadmin to the docker group. Then only we can execute docker commands as ansadmin. Otherwise, it will be continue to throw errors.

```
sudo usermod -aG docker ansadmin
```

Next start docker,

When you tried to start docker you faced an issue

```
[ansadmin@ansible-server-new docker]$ systemctl start docker
Failed to start docker.service: The name org.freedesktop.PolicyKit1 was not provided by any
.service files
See system logs and 'systemctl status docker.service' for details.
```

So you chatgpt and ran

```
sudo yum install polkit -y
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
```

it worked.

```
cgrdx:993:
docker:x:992,
[ansadmin@ansible-server-new docker]$ sudo usermod -aG docker ansadmin
[ansadmin@ansible-server-new docker]$ id ansadmin
uid=1001(ansadmin) gid=1001(ansadmin) groups=1001(ansadmin),992(docker)
[ansadmin@ansible-server-new docker]$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
       Docs: man:docker(8)
[ansadmin@ansible-server-new docker]$ sudo docker
Failed to execute operation: The name org.freedesktop.PolicyKit1 was not provided by any .service files
[ansadmin@ansible-server-new docker]$ systemctl start docker
Failed to start docker.service: The name org.freedesktop.PolicyKit1 was not provided by any .service files
See system logs and 'systemctl status docker.service' for details.
[ansadmin@ansible-server-new docker]$ systemctl status docker
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
       Docs: https://docs.docker.com
[ansadmin@ansible-server-new docker]$ service docker start
Redirecting to /bin/systemctl start docker.service
Failed to start docker.service: The name org.freedesktop.PolicyKit1 was not provided by any .service files
See system logs and 'systemctl status docker.service' for details.
[ansadmin@ansible-server-new docker]$ sudo systemctl start docker
Last login: Wed Sep 11 01:27:08 UTC 2024 on pts/0
Last boot: Wed Sep 11 00:56:18 UTC 2024 from 18.137.159.28 on ssh:root
There were 152 failed login attempts since the last successful login.
[root@ansible-server-new ~]# sudo su - ansadmin
Last login: Wed Sep 11 01:27:08 UTC 2024 on pts/0
[ansadmin@ansible-server-new ~]$ service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
       Docs: https://docs.docker.com
[ansadmin@ansible-server-new ~]$ service docker start
Redirecting to /bin/systemctl start docker.service
Failed to start docker.service: The name org.freedesktop.PolicyKit1 was not provided by any .service files
See system logs and 'systemctl status docker.service' for details.
[ansadmin@ansible-server-new ~]$ sudo yum install polkit -y
Loading plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package polkit.x86_64 0:0.112-26.amzn2.1 will be installed
--> Processing Dependency: libmozjs-17.0.so(mozjs-17.0)(64bit) for package: polkit-0.112-26.amzn2.1.x86_64
--> Processing Dependency: polkit-pkla-compat for package: polkit-0.112-26.amzn2.1.x86_64
--> Processing Dependency: libmozjs-17.0.so()(64bit) for package: polkit-0.112-26.amzn2.1.x86_64
--> Running transaction check
--> Package polkit.x86_64 0:0.1-4.amzn2.0.2 will be installed
--> Package polkit-pkla-compat.x86_64 0:0.1-4.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository        Size
=====
Installing:
polkit            x86_64   0.112-26.amzn2.1      amzn2-core      178 k
Installing for dependencies:
```

```

Package          Arch    Version      Repository   Size
=====
Installing:    polkit           x86_64  0.112-26.amzn2.1      amzn2-core     176 k
Installing for dependencies:
  mozs17          x86_64  17.0.0-20.amzn2.0.1      amzn2-core     1.4 M
  polkit-pkla-compat x86_64  0.1-4.amzn2.0.2      amzn2-core     39 k
Transaction Summary
=====
Install 1 Package (+2 Dependent packages)

Total download size: 1.6 M
Installed size: 4.5 M
Downloading packages:
(1/3): polkit-0.112-26.amzn2.1.x86_64.rpm          | 170 kB  00:00:00
(2/3): mozs17-17.0.0-20.amzn2.0.1.x86_64.rpm       | 1.4 MB  00:00:00
(3/3): polkit-pkla-compat-0.1-4.amzn2.0.2.x86_64.rpm | 39 kB   00:00:00
Total                                         9.7 MB/s | 1.6 MB 00:00:00

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mozs17-17.0.0-20.amzn2.0.1.x86_64          1/3
  Installing : polkit-0.112-26.amzn2.1.x86_64          2/3
  Installing : polkit-pkla-compat-0.1-4.amzn2.0.2.x86_64 3/3
  Verifying  : polkit-pkla-compat-0.1-4.amzn2.0.2.x86_64 1/3
  Verifying  : polkit-0.112-26.amzn2.1.x86_64          2/3
  Verifying  : mozs17-17.0.0-20.amzn2.0.1.x86_64          3/3

Installed:
  polkit.x86_64 0:0.112-26.amzn2.1

Dependency Installed:
  mozs17.x86_64 0:17.0.0-20.amzn2.0.1               polkit-pkla-compat.x86_64 0:0.1-4.amzn2.0.2

Complete!
[ansadmin@ansible-server-new ~]$ sudo systemctl start docker
sudo systemctl start docker
[ansadmin@ansible-server-new ~]$ sudo systemctl enable docker
[Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.]
[ansadmin@ansible-server-new ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2024-09-18 03:58:09 UTC; 20s ago
     Docs: https://docs.docker.com
   Main PID: 30621 (dockerd)
     CGroup: /system.slice/docker.service
             └─30621 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Sep 11 03:58:09 ansible-server-new systemd[1]: Starting Docker Application Container Engine...
Sep 11 03:58:09 ansible-server-new dockerd[30621]: time="2024-09-11T03:58:09.185527874Z" level=info msg="Starting up"
Sep 11 03:58:09 ansible-server-new dockerd[30621]: time="2024-09-11T03:58:09.331988095Z" level=info msg="Loading containers: start."
Sep 11 03:58:09 ansible-server-new dockerd[30621]: time="2024-09-11T03:58:09.6072672Z" level=info msg="Loading containers: done."
Sep 11 03:58:09 ansible-server-new dockerd[30621]: time="2024-09-11T03:58:09.611520527Z" level=warning msg="Warning: bridge-nf-call-iptables is disabled"
Sep 11 03:58:09 ansible-server-new dockerd[30621]: time="2024-09-11T03:58:09.612283232Z" level=error msg="WARNING: bridge-nf-call-...abled"
Sep 11 03:58:09 ansible-server-new dockerd[30621]: time="2024-09-11T03:58:09.612400103Z" level=info msg="Docker daemon" commit=b08a5...25 0.6
Sep 11 03:58:09 ansible-server-new dockerd[30621]: time="2024-09-11T03:58:09.612969673Z" level=info msg="Daemon has completed initialization"
Sep 11 03:58:09 ansible-server-new dockerd[30621]: time="2024-09-11T03:58:09.646511180Z" level=info msg="API listen on /run/docker.sock"
Sep 11 03:58:09 ansible-server-new systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
[ansadmin@ansible-server-new ~]$ 

```

In the next few minutes, we created a dockerfile, prof copied the content of dockerfile from dockerhost which you didn't create remember and created one in ansible and created image and container from it manually.

VIDEO – 37

So far, we have created a Ansible server and we have integrated it with Jenkins, so Jenkins could able to copy artifacts.

With those artifacts, we have created a Docker image, right, by using manual commands.

Now, this image is available on Ansible system.

Now, how can I make this is available

to our target environment, which means that a Docker host, that is where Docker Hub comes into the picture.

If we use Docker Hub, we can push our image onto Docker Hub and this Docker Hub can be accessible by any Docker system.

In this case, we have Docker host, right, even it can able to access our Docker host.

Now, from our Ansible, we can instruct to this Docker host that, okay, go and pull the image from Docker Hub and create a container.

This is how Docker host can pull the image and create a container out of it.

Now, this image is customized by Ansible, which means that with the artifacts which we have created,

so we can see our application is running on Docker host
and our end users could be accessing this Docker host.
And moreover, it may be a one Docker host,
or hundreds of Docker hosts.

Now, let's jump into our Ansible system
and create a playbook to create your image.
Alright, I have logged into my Ansible server
and currently, I'm a ec2-user.
Either I can log into Ansible server directly as a ansadmin,
or else I can switch as a ansadmin,
which means that, equal to,
I have logged in as ansadmin itself
and clear the screen and let's go to /opt

Before creating a Ansible Playbook,
I will quickly run through how Ansible Playbook does work.
To run the Ansible Playbook,
we need to use a command called ansible-playbook.
However, whenever we run this playbook, we need to provide
where we are running this Ansible Playbook.
That we can provide with the -i option,
nothing but inventory.
Where is the inventory exist?
If we don't give -i option, by default,
it is going to take a default inventory file.
What is the default inventory file?
That is what /etc/ansible/hosts.
Even though we call it as a hosts, it is a inventory file.
Whatever servers are there,
on these servers it is going to execute.
But in our case, we want to execute our Ansible Playbook
on Ansible server.
But whatever IP address we could see over here,
this is a Docker host IP address.
So what we can do,
we can add our Ansible server IP address to this one
and we can tell to our Ansible Playbook
that you should execute only on Ansible server.

```

Sep 11 03:58:09 ansible-server-new systemd[1]: Starting Docker Application Container Engine...
Sep 11 03:58:09 ansible-server-new dockerd[38621]: time=2024-09-11T03:58:09.185527874Z level=info msg="Starting up"
Sep 11 03:58:09 ansible-server-new dockerd[38621]: time=2024-09-11T03:58:09.331988095Z level=info msg="Loading containers: start."
Sep 11 03:58:09 ansible-server-new dockerd[38621]: time=2024-09-11T03:58:09.597047267Z level=info msg="Loading containers: done."
Sep 11 03:58:09 ansible-server-new dockerd[38621]: time=2024-09-11T03:58:09.611912536Z level=warning msg="WARNING: bridge-nf-call-...abled"
Sep 11 03:58:09 ansible-server-new dockerd[38621]: time=2024-09-11T03:58:09.612283232Z level=warning msg="WARNING: bridge-nf-call-...abled"
Sep 11 03:58:09 ansible-server-new dockerd[38621]: time=2024-09-11T03:58:09.612406610Z level=info msg="Docker daemon" commit=b88a5...25.0.6
Sep 11 03:58:09 ansible-server-new dockerd[38621]: time=2024-09-11T03:58:09.612464732Z level=info msg="Daemon has completed initialization"
Sep 11 03:58:09 ansible-server-new dockerd[38621]: time=2024-09-11T03:58:09.644111104Z level=info msg="API listen on /run/docker.sock"
Sep 11 03:58:09 ansible-server-new systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.

[ansadmin@ansible-server-new ~]$ ls
[ansadmin@ansible-server-new ~]$ ls cd /opt
[ansadmin@ansible-server-new ~]$ ll
total 0
drwx--xr-x 4 root root 33 Sep 3 22:46 aws
drwx--x--x 4 root root 28 Sep 11 03:58 containerd
drwxr--xr-x 6 ansadmin ansadmin 6 Sep 11 03:33 docker
drwxr--xr-x 2 root root 16 Aug 16 2018 rh
[ansadmin@ansible-server-new opt]$ ls
[ansadmin@ansible-server-new opt]$ cd docker
[ansadmin@ansible-server-new docker]$ ll
total 0
[ansadmin@ansible-server-new docker]$ ansible-playbook -i*C
[ansadmin@ansible-server-new docker]$ cat /etc/ansible/hosts
172.31.82.210

[ansadmin@ansible-server-new docker]$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 brd 172.17.0.1 scope host docker0
        link layer 1000b4 brd 172.17.0.1
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9000
    inet 172.31.83.187 brd 255.255.24.0 broadcast 172.31.95.255
        link layer 00:0c:29:00:00:00 brd 00:0c:29:ff:ff:ff
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 brd 127.0.0.1 scope host lo
        link layer 00:00:00:00:00:00 brd 00:00:00:00:00:00
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[ansadmin@ansible-server-new docker]$ sudo vi /etc/ansible/hosts
[ansadmin@ansible-server-new docker]$ cat /etc/ansible/hosts
[dockerhost]
172.31.82.210

[ansible]
172.31.83.187

```

```

[ansadmin@ansible-server-new docker]$ sudo vi /etc/ansible/hosts
[ansadmin@ansible-server-new docker]$ cat /etc/ansible/hosts
[dockerhost]
172.31.82.210

[ansible]
172.31.83.187

```

-- INSERT --

I have updated inventory file and just to validate,
if we check it once again,
we could see dockerhost IP address, as well
as ansible IP address, but to make work with the Ansible,
you should copy your SSH keys onto Ansible server itself.
Otherwise, you cannot able to communicate with this system
and just to test it out, this is our one IP address, right?
If I do, ansible all -m uptime, if I give, you can see,
it is going to successfully communicate with this system - dockerhost,
but not with this system - ansible.

ansible all -m uptime (wrong command)

```
ansible all -a uptime
```

That is the reason we should copy our own keys onto our own system.

For that, again, we need to execute the same command.

So ssh-copy-id onto this one.

So we are just copying the keys.

I'm giving the password, which is successfully copied.

So ssh-copy-id <private ip of ansible ec2>

It will ask for ansible ansadmin's password I guess

Even we can use the localhost, as well.

Alright, we have given the localhost, as well.

And anyway, once we have used this one , - (ssh-copy-id 172.31.83.187)
same thing we are using over here - (ssh-copy-id localhost).

Nothing to copy.

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                broadcast 127.0.0.1
                link-layer brd 0x000000000000
                brd 0x000000000000
                txqueuelen 1000  (Local Loopback)
                RX packets 0 bytes 0 (0.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 0 bytes 0 (0.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[ansadmin@ansible-server-new docker]$ sudo vi /etc/ansible/hosts
[ansadmin@ansible-server-new docker]$ cat /etc/ansible/hosts
[dockerhost]
172.31.82.210

[ansible]
172.31.83.187
[ansadmin@ansible-server-new docker]$ ansible all -m uptime
The authenticity of host '172.31.83.187 (172.31.83.187)' can't be established.
ECDSA key fingerprint is SHA256:HNXMNNE5E1ZUpjRgYXNvainLbnuus148kZ6xybfJ/bEI.
ECDSA key fingerprint is MD5:cbb:f7:3f:a8:f4:f1:b5:ca:ac:52:62:14:71:86:7b:89.
Are you sure you want to continue connecting (yes/no)? 172.31.82.210 [FAILED] => {
    msg: "The module uptime was not found in configured module paths"
}
^C [ERROR]: User interrupted execution
[ansadmin@ansible-server-new docker]$ ansible all -a uptime
The authenticity of host '172.31.83.187 (172.31.83.187)' can't be established.
ECDSA key fingerprint is SHA256:HNXMNNE5E1ZUpjRgYXNvainLbnuus148kZ6xybfJ/bEI.
ECDSA key fingerprint is MD5:cbb:f7:3f:a8:f4:f1:b5:ca:ac:52:62:14:71:86:7b:89.
Are you sure you want to continue connecting (yes/no)? [WARNING]: Platform linux on host 172.31.82.210 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
172.31.82.210 | CHANGED | rc=0 >
0:29:59 up 5:55, 2 users,  load average: 0.08, 0.02, 0.01
^C [WARNING]: User interrupted execution
[ansadmin@ansible-server-new docker]$ 
[ansadmin@ansible-server-new docker]$ ssh-copy-id 172.31.83.187
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host '172.31.83.187 (172.31.83.187)' can't be established.
ECDSA key fingerprint is SHA256:HNXMNNE5E1ZUpjRgYXNvainLbnuus148kZ6xybfJ/bEI.
ECDSA key fingerprint is MD5:cbb:f7:3f:a8:f4:f1:b5:ca:ac:52:62:14:71:86:7b:89.
Are you sure you want to continue connecting (yes/no)? 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
ansadmin@172.31.83.187's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh '172.31.83.187'"
and check to make sure that only the key(s) you wanted were added.

[ansadmin@ansible-server-new docker]$ ssh-copy-id localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:HNXMNNE5E1ZUpjRgYXNvainLbnuus148kZ6xybfJ/bEI.
ECDSA key fingerprint is MD5:cbb:f7:3f:a8:f4:f1:b5:ca:ac:52:62:14:71:86:7b:89.
Are you sure you want to continue connecting (yes/no)? 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: WARNING: All keys were skipped because they already exist on the remote system.
    (if you think this is a mistake, you may want to use -f option)

[ansadmin@ansible-server-new docker]$
```

Now if I check uptime both should work

```
Number of key(s) added: 1

Now try logging into the machine, with: "ssh '172.31.83.187'"
and check to make sure that only the key(s) you wanted were added.

[ansadmin@ansible-server-new docker]$ ssh-copy-id localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:HNxdMNE5E1ZUpjRgvYXNvaihLbnus14BKz6xybfj/bEI.
ECDSA key fingerprint is MD5:cb:f7:3f:a8:f4:1b:5fc:ac:52:62:14:71:86:7b:89.
Are you sure you want to continue connecting (yes/no)? 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: WARNING: All keys were skipped because they already exist on the remote system.
          (if you think this is a mistake, you may want to use -f option)

[ansadmin@ansible-server-new docker]$ ansible all -a uptime
[WARNING]: Platform linux on host 172.31.82.210 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
172.31.82.210 | CHANGED | rc=0 >>
 04:37:39 up 6:04, 2 users,  load average: 0.00, 0.00, 0.00
[WARNING]: Platform linux on host 172.31.83.187 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
172.31.83.187 | CHANGED | rc=0 >>
 04:37:39 up 4:12, 2 users,  load average: 0.00, 0.00, 0.00
[ansadmin@ansible-server-new docker]$
```

Now time to write ansible playbook.

Before that you create dockerfile from video 36. Cuz here there comes an issue and then you change permissions. So to be safe you do this step. /var/run/docker.sock

Sudo chmod 777 /var/run/docker.sock

In ansadmin in ansible server

Cd /opt

Cd docker

Vi Dockerfile

```
● ○ ● ◻ Downloads — ansadmin@ansible-server-new:/opt/docker — ssh -i devopsproject.pem ec2-user@54.1...
FROM tomcat:latest
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
COPY ./*.war /usr/local/tomcat/webapps
~
```

FROM tomcat:latest

```
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
COPY ./*.war /usr/local/tomcat/webapps
```

The issue should have come here. Permission denied `/var/run/docker.sock`
And after this you change permission using `Sudo chmod 777 /var/run/docker.sock`

But you didn't face this issue, the docker build was successful.

Prof next created a container out of it.
`Docker run -t -name regapp-container -p 8081:8080 regapp:v1`

But you went with `-d`

```
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ansible-server-new ~]$ 
[ec2-user@ansible-server-new ~]$ sudo su - ansible
Last login: Wed Sep 11 04:37:39 UTC 2024 from ip-172-31-83-187.ec2.internal on pts/3
[ansible@ansible-server-new ~]$ ls
total 0
[ansible@ansible-server-new ~]$ cd /opt
[ansible@ansible-server-new ~]$ ls
total 0
drwxr-xr-x 4 root root 33 Sep 3 22:46 aws
drwxr-xr-x 4 root root 28 Sep 11 03:58 containerd
drwxr-xr-x 2 ansible ansible 6 Sep 11 03:33 docker
drwxr-xr-x 2 root root 6 Aug 16 2018 rh
[ansible@ansible-server-new ~]$ docker ls
[ansible@ansible-server-new ~]$ docker ls -v Dockerfile
[ansible@ansible-server-new ~]$ docker build -t regapp:v1 .
[+] Building 13.1s (8/8) FINISHED                                            docker:default
=> (internal) load build definition from Dockerfile                         0.0s
=> (internal) load metadata for docker.io/library/tomcat:latest            0.0s
=> (internal) load build context                                             0.3s
=> (internal) resolve image configuration                                     0.0s
=> transferring context: 28                                                 0.0s
=> 1/31: FROM docker.io/library/tomcat:latest@sha256:221f95c5c6bbd428f2c7c70e8a421f668f21844f3e9 11.7s
=> 2/31:  resolve docker.io/library/tomcat@sha256:221f95c5c6bbd428f2c7c70e8a421f668f21844f3e9 0.0s
=> 3/31:  transfer manifest: 1678                                           0.0s
=> 4/31:  resolve https://index.docker.io/v1/repositories/docker/library/tomcat@sha256:221f95c5c6bbd428f2c7c70e8a421f668f21844f3e9 0.4s
=> 5/31:  transfer manifest: 1678                                           0.0s
=> 6/31:  resolve https://index.docker.io/v1/repositories/docker/library/tomcat@sha256:221f95c5c6bbd428f2c7c70e8a421f668f21844f3e9 0.0s
=> 7/31:  transfer blob: 1678                                           0.0s
=> 8/31:  transfer manifest: 1678                                           0.0s
=> 9/31:  transfer blob: 1678                                           0.0s
=> 10/31:  transfer manifest: 1678                                         0.0s
=> 11/31:  transfer blob: 1678                                         0.0s
=> 12/31:  transfer manifest: 1678                                         0.0s
=> 13/31:  transfer blob: 1678                                         0.0s
=> 14/31:  transfer manifest: 1678                                         0.0s
=> 15/31:  transfer blob: 1678                                         0.0s
=> 16/31:  transfer manifest: 1678                                         0.0s
=> 17/31:  transfer blob: 1678                                         0.0s
=> 18/31:  transfer manifest: 1678                                         0.0s
=> 19/31:  transfer blob: 1678                                         0.0s
=> 20/31:  transfer manifest: 1678                                         0.0s
=> 21/31:  transfer blob: 1678                                         0.0s
=> 22/31:  transfer manifest: 1678                                         0.0s
=> 23/31:  transfer blob: 1678                                         0.0s
=> 24/31:  transfer manifest: 1678                                         0.0s
=> 25/31:  transfer blob: 1678                                         0.0s
=> 26/31:  transfer manifest: 1678                                         0.0s
=> 27/31:  transfer blob: 1678                                         0.0s
=> 28/31:  transfer manifest: 1678                                         0.0s
=> 29/31:  transfer blob: 1678                                         0.0s
=> 30/31:  transfer manifest: 1678                                         0.0s
=> 31/31:  transfer blob: 1678                                         0.0s
=> 32/31:  transfer manifest: 1678                                         0.0s
=> 33/31:  transfer blob: 1678                                         0.0s
=> 34/31:  transfer manifest: 1678                                         0.0s
=> 35/31:  transfer blob: 1678                                         0.0s
=> 36/31:  transfer manifest: 1678                                         0.0s
=> 37/31:  transfer blob: 1678                                         0.0s
=> 38/31:  transfer manifest: 1678                                         0.0s
=> 39/31:  transfer blob: 1678                                         0.0s
=> 40/31:  transfer manifest: 1678                                         0.0s
=> 41/31:  transfer blob: 1678                                         0.0s
=> 42/31:  transfer manifest: 1678                                         0.0s
=> 43/31:  transfer blob: 1678                                         0.0s
=> 44/31:  transfer manifest: 1678                                         0.0s
=> 45/31:  transfer blob: 1678                                         0.0s
=> 46/31:  transfer manifest: 1678                                         0.0s
=> 47/31:  transfer blob: 1678                                         0.0s
=> 48/31:  transfer manifest: 1678                                         0.0s
=> 49/31:  transfer blob: 1678                                         0.0s
=> 50/31:  transfer manifest: 1678                                         0.0s
=> 51/31:  transfer blob: 1678                                         0.0s
=> 52/31:  transfer manifest: 1678                                         0.0s
=> 53/31:  transfer blob: 1678                                         0.0s
=> 54/31:  transfer manifest: 1678                                         0.0s
=> 55/31:  transfer blob: 1678                                         0.0s
=> 56/31:  transfer manifest: 1678                                         0.0s
=> 57/31:  transfer blob: 1678                                         0.0s
=> 58/31:  transfer manifest: 1678                                         0.0s
=> 59/31:  transfer blob: 1678                                         0.0s
=> 60/31:  transfer manifest: 1678                                         0.0s
=> 61/31:  transfer blob: 1678                                         0.0s
=> 62/31:  transfer manifest: 1678                                         0.0s
=> 63/31:  transfer blob: 1678                                         0.0s
=> 64/31:  transfer manifest: 1678                                         0.0s
=> 65/31:  transfer blob: 1678                                         0.0s
=> 66/31:  transfer manifest: 1678                                         0.0s
=> 67/31:  transfer blob: 1678                                         0.0s
=> 68/31:  transfer manifest: 1678                                         0.0s
=> 69/31:  transfer blob: 1678                                         0.0s
=> 70/31:  transfer manifest: 1678                                         0.0s
=> 71/31:  transfer blob: 1678                                         0.0s
=> 72/31:  transfer manifest: 1678                                         0.0s
=> 73/31:  transfer blob: 1678                                         0.0s
=> 74/31:  transfer manifest: 1678                                         0.0s
=> 75/31:  transfer blob: 1678                                         0.0s
=> 76/31:  transfer manifest: 1678                                         0.0s
=> 77/31:  transfer blob: 1678                                         0.0s
=> 78/31:  transfer manifest: 1678                                         0.0s
=> 79/31:  transfer blob: 1678                                         0.0s
=> 80/31:  transfer manifest: 1678                                         0.0s
=> 81/31:  transfer blob: 1678                                         0.0s
=> 82/31:  transfer manifest: 1678                                         0.0s
=> 83/31:  transfer blob: 1678                                         0.0s
=> 84/31:  transfer manifest: 1678                                         0.0s
=> 85/31:  transfer blob: 1678                                         0.0s
=> 86/31:  transfer manifest: 1678                                         0.0s
=> 87/31:  transfer blob: 1678                                         0.0s
=> 88/31:  transfer manifest: 1678                                         0.0s
=> 89/31:  transfer blob: 1678                                         0.0s
=> 90/31:  transfer manifest: 1678                                         0.0s
=> 91/31:  transfer blob: 1678                                         0.0s
=> 92/31:  transfer manifest: 1678                                         0.0s
=> 93/31:  transfer blob: 1678                                         0.0s
=> 94/31:  transfer manifest: 1678                                         0.0s
=> 95/31:  transfer blob: 1678                                         0.0s
=> 96/31:  transfer manifest: 1678                                         0.0s
=> 97/31:  transfer blob: 1678                                         0.0s
=> 98/31:  transfer manifest: 1678                                         0.0s
=> 99/31:  transfer blob: 1678                                         0.0s
=> 100/31:  transfer manifest: 1678                                         0.0s
=> 101/31:  transfer blob: 1678                                         0.0s
=> 102/31:  transfer manifest: 1678                                         0.0s
=> 103/31:  transfer blob: 1678                                         0.0s
=> 104/31:  transfer manifest: 1678                                         0.0s
=> 105/31:  transfer blob: 1678                                         0.0s
=> 106/31:  transfer manifest: 1678                                         0.0s
=> 107/31:  transfer blob: 1678                                         0.0s
=> 108/31:  transfer manifest: 1678                                         0.0s
=> 109/31:  transfer blob: 1678                                         0.0s
=> 110/31:  transfer manifest: 1678                                         0.0s
=> 111/31:  transfer blob: 1678                                         0.0s
=> 112/31:  transfer manifest: 1678                                         0.0s
=> 113/31:  transfer blob: 1678                                         0.0s
=> 114/31:  transfer manifest: 1678                                         0.0s
=> 115/31:  transfer blob: 1678                                         0.0s
=> 116/31:  transfer manifest: 1678                                         0.0s
=> 117/31:  transfer blob: 1678                                         0.0s
=> 118/31:  transfer manifest: 1678                                         0.0s
=> 119/31:  transfer blob: 1678                                         0.0s
=> 120/31:  transfer manifest: 1678                                         0.0s
=> 121/31:  transfer blob: 1678                                         0.0s
=> 122/31:  transfer manifest: 1678                                         0.0s
=> 123/31:  transfer blob: 1678                                         0.0s
=> 124/31:  transfer manifest: 1678                                         0.0s
=> 125/31:  transfer blob: 1678                                         0.0s
=> 126/31:  transfer manifest: 1678                                         0.0s
=> 127/31:  transfer blob: 1678                                         0.0s
=> 128/31:  transfer manifest: 1678                                         0.0s
=> 129/31:  transfer blob: 1678                                         0.0s
=> 130/31:  transfer manifest: 1678                                         0.0s
=> 131/31:  transfer blob: 1678                                         0.0s
=> 132/31:  transfer manifest: 1678                                         0.0s
=> 133/31:  transfer blob: 1678                                         0.0s
=> 134/31:  transfer manifest: 1678                                         0.0s
=> 135/31:  transfer blob: 1678                                         0.0s
=> 136/31:  transfer manifest: 1678                                         0.0s
=> 137/31:  transfer blob: 1678                                         0.0s
=> 138/31:  transfer manifest: 1678                                         0.0s
=> 139/31:  transfer blob: 1678                                         0.0s
=> 140/31:  transfer manifest: 1678                                         0.0s
=> 141/31:  transfer blob: 1678                                         0.0s
=> 142/31:  transfer manifest: 1678                                         0.0s
=> 143/31:  transfer blob: 1678                                         0.0s
=> 144/31:  transfer manifest: 1678                                         0.0s
=> 145/31:  transfer blob: 1678                                         0.0s
=> 146/31:  transfer manifest: 1678                                         0.0s
=> 147/31:  transfer blob: 1678                                         0.0s
=> 148/31:  transfer manifest: 1678                                         0.0s
=> 149/31:  transfer blob: 1678                                         0.0s
=> 150/31:  transfer manifest: 1678                                         0.0s
=> 151/31:  transfer blob: 1678                                         0.0s
=> 152/31:  transfer manifest: 1678                                         0.0s
=> 153/31:  transfer blob: 1678                                         0.0s
=> 154/31:  transfer manifest: 1678                                         0.0s
=> 155/31:  transfer blob: 1678                                         0.0s
=> 156/31:  transfer manifest: 1678                                         0.0s
=> 157/31:  transfer blob: 1678                                         0.0s
=> 158/31:  transfer manifest: 1678                                         0.0s
=> 159/31:  transfer blob: 1678                                         0.0s
=> 160/31:  transfer manifest: 1678                                         0.0s
=> 161/31:  transfer blob: 1678                                         0.0s
=> 162/31:  transfer manifest: 1678                                         0.0s
=> 163/31:  transfer blob: 1678                                         0.0s
=> 164/31:  transfer manifest: 1678                                         0.0s
=> 165/31:  transfer blob: 1678                                         0.0s
=> 166/31:  transfer manifest: 1678                                         0.0s
=> 167/31:  transfer blob: 1678                                         0.0s
=> 168/31:  transfer manifest: 1678                                         0.0s
=> 169/31:  transfer blob: 1678                                         0.0s
=> 170/31:  transfer manifest: 1678                                         0.0s
=> 171/31:  transfer blob: 1678                                         0.0s
=> 172/31:  transfer manifest: 1678                                         0.0s
=> 173/31:  transfer blob: 1678                                         0.0s
=> 174/31:  transfer manifest: 1678                                         0.0s
=> 175/31:  transfer blob: 1678                                         0.0s
=> 176/31:  transfer manifest: 1678                                         0.0s
=> 177/31:  transfer blob: 1678                                         0.0s
=> 178/31:  transfer manifest: 1678                                         0.0s
=> 179/31:  transfer blob: 1678                                         0.0s
=> 180/31:  transfer manifest: 1678                                         0.0s
=> 181/31:  transfer blob: 1678                                         0.0s
=> 182/31:  transfer manifest: 1678                                         0.0s
=> 183/31:  transfer blob: 1678                                         0.0s
=> 184/31:  transfer manifest: 1678                                         0.0s
=> 185/31:  transfer blob: 1678                                         0.0s
=> 186/31:  transfer manifest: 1678                                         0.0s
=> 187/31:  transfer blob: 1678                                         0.0s
=> 188/31:  transfer manifest: 1678                                         0.0s
=> 189/31:  transfer blob: 1678                                         0.0s
=> 190/31:  transfer manifest: 1678                                         0.0s
=> 191/31:  transfer blob: 1678                                         0.0s
=> 192/31:  transfer manifest: 1678                                         0.0s
=> 193/31:  transfer blob: 1678                                         0.0s
=> 194/31:  transfer manifest: 1678                                         0.0s
=> 195/31:  transfer blob: 1678                                         0.0s
=> 196/31:  transfer manifest: 1678                                         0.0s
=> 197/31:  transfer blob: 1678                                         0.0s
=> 198/31:  transfer manifest: 1678                                         0.0s
=> 199/31:  transfer blob: 1678                                         0.0s
=> 200/31:  transfer manifest: 1678                                         0.0s
=> 201/31:  transfer blob: 1678                                         0.0s
=> 202/31:  transfer manifest: 1678                                         0.0s
=> 203/31:  transfer blob: 1678                                         0.0s
=> 204/31:  transfer manifest: 1678                                         0.0s
=> 205/31:  transfer blob: 1678                                         0.0s
=> 206/31:  transfer manifest: 1678                                         0.0s
=> 207/31:  transfer blob: 1678                                         0.0s
=> 208/31:  transfer manifest: 1678                                         0.0s
=> 209/31:  transfer blob: 1678                                         0.0s
=> 210/31:  transfer manifest: 1678                                         0.0s
=> 211/31:  transfer blob: 1678                                         0.0s
=> 212/31:  transfer manifest: 1678                                         0.0s
=> 213/31:  transfer blob: 1678                                         0.0s
=> 214/31:  transfer manifest: 1678                                         0.0s
=> 215/31:  transfer blob: 1678                                         0.0s
=> 216/31:  transfer manifest: 1678                                         0.0s
=> 217/31:  transfer blob: 1678                                         0.0s
=> 218/31:  transfer manifest: 1678                                         0.0s
=> 219/31:  transfer blob: 1678                                         0.0s
=> 220/31:  transfer manifest: 1678                                         0.0s
=> 221/31:  transfer blob: 1678                                         0.0s
=> 222/31:  transfer manifest: 1678                                         0.0s
=> 223/31:  transfer blob: 1678                                         0.0s
=> 224/31:  transfer manifest: 1678                                         0.0s
=> 225/31:  transfer blob: 1678                                         0.0s
=> 226/31:  transfer manifest: 1678                                         0.0s
=> 227/31:  transfer blob: 1678                                         0.0s
=> 228/31:  transfer manifest: 1678                                         0.0s
=> 229/31:  transfer blob: 1678                                         0.0s
=> 230/31:  transfer manifest: 1678                                         0.0s
=> 231/31:  RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
=> 232/31: COPY ./*.war /usr/local/tomcat/webapps
=> 233/31: docker run -d --name regapp-container -p 8081:8080 regapp:v1
=> 234/31: exporting to image
=> 235/31: exporting layers
=> 236/31: writing image sha256:7287ad6c8e95fa81b2b261858c9ba6364adad49389a4cf8dcfd376d51f00ba
=> 237/31:  /usr/bin/docker save -o /tmp/regapp.tar regapp:v1
[ansible@ansible-server-new ~]$ ls
total 0
[ansible@ansible-server-new ~]$
```

To delete container `docker rm -f <container id>`

To delete images `docker image prune` or `docker rmi <image name:tag>` if latest don't have to give any. Check video 40 4:50 mins

So this step created a container too.

By using this one we can create a container.

So to create a container, `docker run`,

I'm going to name it as a

`docker run` minus `t`, I'm not using detached mode just I'm going to use terminal minus minus name registration app server minus `p`.

I'm going to use port number 8081

because this is entirely different system.

So this port is freely available

because we haven't used any ports

on our Ansible system so far.

You built container and copied ansible public ip with :8081 and it worked, i.e tomcat homepage was visible. But when you gave /webapp you saw http error cuz remmebr you have run Jenkins job yet and its not pulling your code

The screenshot shows the Apache Tomcat 10.1.28 homepage. At the top, there's a navigation bar with links for Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. Below the navigation is a banner that says "If you're seeing this, you've successfully installed Tomcat. Congratulations!" It features a cartoon cat icon and links to recommended reading: Security Considerations How-To, Manager Application How-To, and Clustering/Session Replication How-To. To the right of the banner are three buttons: Server Status, Manager App, and Host Manager. The main content area is divided into several sections: "Developer Quick Start" (with links to Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, and Servlet Specifications); "Documentation" (with links to Tomcat 10.1 Documentation, Configuration, and Wiki); and "Getting Help" (with links to FAQ and Mailing Lists, and information about mailing lists like tomcat-announce, tomcat-users, tomcat-dev, and tomcat-qa). At the bottom, there are links for Other Downloads (Tomcat Connectors, Tomcat Native), Other Documentation (Tomcat Connectors, mod_jk Documentation), Get Involved (Overview, Source Repositories), Miscellaneous (Contact, Legal), and Apache Software Foundation (Who We Are).

The screenshot shows an HTTP Status 404 - Not Found error page. The title bar says "Not Secure – 54.197.96.232". The main content area has a dark header "HTTP Status 404 – Not Found". Below it, there's a "Type Status Report" section. Under "Message", it says "The requested resource [/webapp] is not available". Under "Description", it says "The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.". At the bottom, a dark footer bar says "Apache Tomcat/10.1.28".

In next lecture we will create an Ansible playbook which can do all these activities through the playbook.

So since you logged in again the next day you wanted to be sure if all is good so you ran ansible all -a uptime (its all good)

Next we gotta push images to dockerhub so for that we gotta login into dockerhub account from our ansible

docker login

Username – nitheshskmr

Password – divadiaz1

So you can see here I have logged into my Docker hub successfully.

But in the real world, we may not use Docker hub

We can create our own Docker registries

in our local system like (indistinct) Artifactory

or else Image and Docker registry, or Azure Docker registry.

Anything we can use, it will just hold your image.

Video 38, prof logged into dockerhub from ansible and tried to push image to dockerhub but remember you gotta push and mention your username with it if not how will dockerhub know which account to send, so he first

Docker tag <image id> valaxyv/<image name:<latest or the image tag i.e v1 for example>

Docker images (you can see new image has created in the name of valaxy/regapp)

Docker push valaxy/regapp:latest

You didn't do above 3 steps cuz , this is all manual in next you will do it inside playbook

Also prof saud make sure with which user you are running your ansible playbook, with the same user you gotta login into dockerhub only then you can push images. i.e in here ansadmin or root. Just go with ansadmin.

Another thing creating image should only happen on ansible server. That's the reason we have given hosts: ansible in regapp.yml

If you want to specify explicitly,
we can use the option called limits
while executing our Ansible playbook.

Ansible-playbook regapp.yml --limit <ip address of ansible>

Okay, it is creating image and pushing the image
and the Docker
images; I hope you don't see any changes over there, why?
Because we have already created these images sometime back,
so there won't be any change.

And here also,
just you can see only a few seconds ago,
it has been updated.

However, there is no change in the Docker file,
but this timestamp update, just giving indication
that our Ansible playbook is working fine.

Now, I don't want to execute this Ansible playbook as well.
Then what we can do, we can give this command
to our Jenkins server.

So Jenkins server will initiate this Ansible playbook
whenever there is new code come in.

That is where we need to create a new image, right?
Otherwise, there is no point of creating image.

So now what we can do is we will go to our Jenkins job
and update it, and we'll add this Ansible playbook
to execute it whenever there is a change in the code, right?

So that this Ansible playbook get executed
and creates a new image, okay?

Nothing to change; only here, we need to give this command,
whatever we have executed.

And we don't need to mention limit, why?
Because we have already mentioned in our playbook
that use only Ansible group.

All right, so that is how we can simplify the customization.

He means he has already executed a job right, he is editing in that, remember in exec command he gives ansible-playbook /opt/docker/regapp.yml (location of the playbook as well)

Video – 40

Here we create another playbook to deploy it on dockerhost from ansible.

So here prof runs the deploy playbook manually, gets permission error, so he goes to docker host and gives permission

chmod 777 /var/run/docker.sock

Ansible playbook without minus minus check option.
Now what it should do, it should pull the image from the remote repository, I mean to say docker hub and create a container out of it.

Let's see whether it can able to do it or not.

Okay? It's failed because of the permission issue on var run docker dot sock.

Okay? We had a similar kind of issue.

Why? Because we are running as a ANS admin user.

So what we can do it is quite simple.

We will go to this server - dockerhost and we will own the not one, change the permission chmod 777 /var/run/docker.sock on this specific directory.

So prof tries to run it again it works on browser too.

Now again he tried to run the playbook and gets error cuz same name, so in next video he will edit the playbook again. Like first remove container, remove image and then create new container, error handling.

Kubernetes

We are gonna setup kubernetes using eksctl. (eksctl is a command line utility)

As prerequisites, we need an EC2 instance, and this EC2 instance, we are going to use it as a bootstrap image.

Bootstrap image, nothing but.

We are going to set up Kubernetes with the help of this bootstrap image,

and this EC2 instance going to communicate with the EKS service through the AWSCLI. That is the reason we need to make sure that we have installed AWSCLI.

Next thing is setting up the kubectl. We are going to use kubectl commands to instruct our Kubernetes cluster,

name: eks-bootstrap-server-new

USE LINUX 2023 AMI

The screenshot shows the AWS CloudFormation console interface. At the top, there's a navigation bar with 'Services', a search bar, and user information. Below it, a message says 'following the simple steps below.' The main area has two tabs: 'Name and tags' (selected) and 'Info'. Under 'Name and tags', the name 'eks-bootstrap-ec2-new' is entered. To the right, there's a button to 'Add additional tags'. Below this, there's a section titled 'Application and OS Images (Amazon Machine Image)' with a sub-section 'Info'. It describes what an AMI is and provides a search bar. Below the search bar, there are two tabs: 'Recents' (selected) and 'Quick Start'. Under 'Recents', several AMI icons are shown: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. To the right of these, there's a 'Browse more AMIs' button with a note about including AMIs from AWS Marketplace and the Community. A tooltip for the 'Free tier' is displayed, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and'. On the right side of the screen, there's a 'Summary' section with a table showing 'Number of instances' set to 1. Other details include 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. At the bottom, there are 'Cancel', 'Launch instance' (which is highlighted in orange), and 'Review commands' buttons. The footer contains links for CloudShell, Feedback, and various AWS terms like Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS CloudFormation console interface. A new stack named 'HelloWorld' is being created. The 'General Information' section shows the stack name, region (N. Virginia), and a summary of the resources: 1 Lambda function, 1 CloudWatch Log group, and 1 CloudWatch Metrics metric. The 'Resources' section lists the Lambda function with its details: Handler (index.js), Role (HelloWorldRole), and Test events (HelloWorldTestEvent). The 'Outputs' section shows the ARN of the Lambda function.

Ssh into it

Aws –version

We download aws cli latest version

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

So amazon 2023 AMI is installed with latest. Prof had to do it cuz he did it on linux 2 ami.

Next install kubectl

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.2/2024-07-12/bin/linux/arm64/kubectl
```

Let's download the packages and download it.

Now we are downloading kube CTL packages.

Once it is downloaded, we need to add execution permissions
to this CH mode plus X

and kube CTL

and we need to move it into slash c vsr local bin.

While we are moving, if you see the echo dollar path,
we could able to see that USR local bin in our path.

So whenever we execute any command,

it is going to validate in this location as well.

So kube CTL --version if I give it is going to display the version.

You can see here client version.

It is 1.21 nothing but kube CTL is successfully installed.

You did a mistake here by installing arm architecture instead of AMD. So when you ran
kubectl version you got error. You chatgptd and it gave solution.

Final steps:

```
aws --version
```

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

```
aws --version
```

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.2/2024-07-12/bin/linux/amd64/kubectl
```

||

```
chmod +x kubectl
sudo mv kubectl /usr/local/bin
kubectl version --client
kubectl version
```

```
echo $PATH
```

```
curl -LO https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_\$\(uname -s\)\_amd64.tar.gz
```

```
tar -xzf eksctl_$(uname -s)_amd64.tar.gz  
sudo mv eksctl /usr/local/bin  
eksctl version  
cd /usr/local/bin  
||  
mv eksctl /tmp  
cd /tmp  
||
```

```

inflating: aws/dist/docutils/parsers/rst/include/isogrk4.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrk4.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrb2.txt
inflating: aws/dist/docutils/parsers/rst/include/isodia.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrk4-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrub.txt
inflating: aws/dist/docutils/parsers/rst/include/iconv.txt
inflating: aws/dist/docutils/parsers/rst/include/isomarx.txt
inflating: aws/dist/docutils/parsers/rst/include/isocyr2.txt
inflating: aws/dist/docutils/parsers/rst/include/isomac.txt
inflating: aws/dist/docutils/parsers/rst/include/isocr1.txt
[root@eks-bootstrapper ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.2/2024-07-12/bin/linux/arm64/kubectl
[root@eks-bootstrapper ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.2/2024-07-12/bin/linux/arm64/kubectl
aws-cliv2.15.18. Python/3.9.16 Linux/x86_64_amzn2023.x86_64 source/x86_64.amzn.2023 prompt/off
[root@eks-bootstrapper ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.2/2024-07-12/bin/linux/arm64/kubectl
[Total % Received Xferd Average Speed Time Current
          Dload Upload Total Spent Left Speed
100 47.4M 100 47.4M 0 0:00:06 0:00:06 --:-- 8225k
[root@eks-bootstrapper ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.2/2024-07-12/bin/linux/arm64/kubectl
total 108216
drwxr-xr-x 3 root root    78 Sep 11 18:21 aws
-rw-r--r-- 1 root root 46868288 Sep 11 20:42 awscliv2.zip
-rw-r--r-- 1 root root 49938584 Sep 11 20:46 kubectl
[root@eks-bootstrapper ~]# chmod +x kubectl
[root@eks-bootstrapper ~]# sudo mv kubectl /usr/local/bin
[root@eks-bootstrapper ~]# echo $PATH
:/root/.local/bin:/root/bin:/usr/local/bin:/usr/bin:/sbin:/usr/bin
[root@eks-bootstrapper ~]# kubectl version
-bash: /usr/local/bin/kubectl: cannot execute binary file: Exec format error
[root@eks-bootstrapper ~]# kubectl version --client
-bash: /usr/local/bin/kubectl: cannot execute binary file: Exec format error
[root@eks-bootstrapper ~]# sudo mv kubectl /usr/local/bin
[root@eks-bootstrapper ~]# sudo mv kubectl /usr/local/bin
mv: cannot stat 'kubectl': No such file or directory
[root@eks-bootstrapper ~]# kubectl version --client
-bash: /usr/local/bin/kubectl: cannot execute binary file: Exec format error
[root@eks-bootstrapper ~]# sudo mv kubectl /usr/local/bin
[root@eks-bootstrapper ~]# uname -m
x86_64
[root@eks-bootstrapper ~]# file /usr/local/bin/kubectl
/usr/local/bin/kubectl: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), statically linked, Go BuildID=73Ju0ihNz-xTkRcjg6CV_rHCKbgexX9Clqu8S85../Hu3KfijCfmJv@A551My/nU-l1c..mxECFvJWZp3a, stripped
[Total % Received Xferd Average Speed Time Current
          Dload Upload Total Spent Left Speed
100 49.8M 100 49.8M 0 0:00:06 0:00:06 --:-- 7978k
[root@eks-bootstrapper ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.2/2024-07-12/bin/linux/amd64/kubectl
total 109496
drwxr-xr-x 3 root root    78 Sep 11 18:21 aws
-rw-r--r-- 1 root root 46868288 Sep 11 20:42 awscliv2.zip
-rw-r--r-- 1 root root 51454164 Sep 11 20:55 kubectl
[root@eks-bootstrapper ~]# chmod +x kubectl
[root@eks-bootstrapper ~]# sudo mv kubectl /usr/local/bin
[root@eks-bootstrapper ~]# kubectl version --client
Client Version: v1.30.2-eks-1552ad9
Kustomize Version: v5.0.4-e-20230601165947-ace0bf390e83
[root@eks-bootstrapper ~]# kubectl version
Client Version: v1.30.2-eks-1552ad9
Kustomize Version: v5.0.4-e-20230601165947-ace0bf390e83
The connection to the server localhost:8000 was refused - did you specify the right host or port?
[root@eks-bootstrapper ~]#

```

ChatGPT

Memory Full

[root@eks-bootstrap-server-new ~]# kubectl version
-bash: /usr/local/bin/kubectl: cannot execute binary file: Exec format error
[root@eks-bootstrap-server-new ~]#

The error `cannot execute binary file: Exec format error` typically occurs when you try to run a binary compiled for a different architecture than your system's architecture (for example, running an ARM binary on an `x86_64` machine).

To resolve this issue, follow these steps:

1. **Check the system architecture:** Run the following command to see your system's architecture:

```
bash
```

`uname -m`
Copy code
2. **Verify the architecture of the `kubectl` binary:** Check if the `kubectl` binary is for the correct architecture:

```
bash
```

`file /usr/local/bin/kubectl`
Copy code

Message ChatGPT

ChatGPT can make mistakes. Check important info.

ChatGPT

Memory Full ⓘ

[root@eks-bootstrap-server-new ~]# uname -m
x86_64
[root@eks-bootstrap-server-new ~]# file /usr/local/bin/kubectl
/usr/local/bin/kubectl: ELF 64-bit LSB executable, ARM aarch64,
version 1 (SYSV), statically linked, Go BuildID=7JJu0lhNz-
xTkRcqj6CV/_rHCKbgeAX9Clqu8SB5/_hUt3KfJCfmJv0A55IMy/nU-
lIc_mxCEFVJWZp3a, stripped
[root@eks-bootstrap-server-new ~]#

It looks like the `kubectl` binary you have is built for ARM architecture (`aarch64`), but your system is `x86_64`, which is Intel/AMD architecture. That's why you're encountering the `Exec format error`.

You need to download the `kubectl` binary that's compatible with your `x86_64` architecture. Here's how you can do it:

1. Remove the incorrect binary (if necessary):
bash
sudo rm /usr/local/bin/kubectl
2. Download the correct `kubectl` binary for `x86_64`:
bash
curl -Ls https://dl.k8s.io/release/stable/bin/linux/amd64/kubectl

Message ChatGPT

ChatGPT can make mistakes. Check important info.

ChatGPT

Memory Full ⓘ

1. Remove the incorrect binary (if necessary):
bash
sudo rm /usr/local/bin/kubectl
2. Download the correct `kubectl` binary for `x86_64`:
bash
curl -LO "https://dl.k8s.io/release/\$(curl -L -s https://dl.k8s.io/release/stable/bin/linux/amd64/kubectl")"
3. Make the new binary executable:
bash
chmod +x kubectl
4. Move the binary to `/usr/local/bin/`:
bash
sudo mv kubectl /usr/local/bin/kubectl
5. Verify the installation:
bash
kubectl version --client

Message ChatGPT

ChatGPT can make mistakes. Check important info.

Next eksctl

Prof downloading it to /tmp file.

```

-bash: /usr/local/bin/kubectl: cannot execute binary file: Exec format error
[root@eks-bootstrap-server-new ~]# sudo mv kubectl /usr/local/bin
mv: cannot stat 'kubectl': No such file or directory
[root@eks-bootstrap-server-new ~]# kubectl version --client
-bash: /usr/local/bin/kubectl: cannot execute binary file: Exec format error
[root@eks-bootstrap-server-new ~]# kubectl version
-bash: /usr/local/bin/kubectl: cannot execute binary file: Exec format error
[root@eks-bootstrap-server-new ~]# uname -a
x86_64
[root@eks-bootstrap-server-new ~]# file /usr/local/bin/kubectl
/usr/local/bin/kubectl: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), statically linked, Go BuildID=7JuOinNz-xTkRcjg6CV_rHCKbgeAX9Clqu8S85_hUt3KfijCfmJv0A55lMy/nU-llc_mxECEfJVWZp3a, stripped
d
[root@eks-bootstrap-server-new ~]#
[root@eks-bootstrap-server-new ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.2/2024-07-12/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
100 49.0M 100 49.0M 0 0 7327k 0 0:00:06 0:00:06 --:--:-- 7978k
[root@eks-bootstrap-server-new ~]# ll
total 198352
drwxr-xr-x 3 root root 78 Sep 11 18:21 aws
-rw-r--r-- 1 root root 68686288 Sep 11 20:42 awscliv2.zip
-rw-r--r-- 1 root root 51454184 Sep 11 20:55 kubectl
[root@eks-bootstrap-server-new ~]# chmod +x kubectl
[root@eks-bootstrap-server-new ~]# sudo mv kubectl /usr/local/bin
[root@eks-bootstrap-server-new ~]# kubectl version --client
Client Version: v1.30.2-eks-1552ad0
Kustomize Version: v5.0.4-0.20230601165947-4ce0bf398cc3
[root@eks-bootstrap-server-new ~]# kubectl version
Client Version: v1.30.2-eks-1552ad0
Kustomize Version: v5.0.4-0.20230601165947-4ce0bf398cc3
The connection to the server https://127.0.0.1:10000 was refused - did you specify the right host or port?
[root@eks-bootstrap-server-new ~]# echo $PATH
/root/.local/bin:/root/bin:/usr/local/sbin:/usr/local/bin:/usr/bin
[root@eks-bootstrap-server-new ~]# curl -L "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz"
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0:--:--:--:--:--:--:--:--:--:--:--:--:--:0
0 0 0 0 0 0 0 0:--:--:--:--:--:--:--:--:--:--:--:--:0
0 0 0 0 0 0 0 0:--:--:--:--:--:--:--:--:--:--:--:--:0
100 34.0M 100 34.0M 0 0 49.6M 0:--:--:--:--:--:--:--:--:49.6M
[root@eks-bootstrap-server-new ~]# tar -xzf eksctl_$(uname -s)_amd64.tar.gz
[root@eks-bootstrap-server-new ~]# sudo mv eksctl /usr/local/bin
[root@eks-bootstrap-server-new ~]# eksctl version
0.190.0
[root@eks-bootstrap-server-new ~]# cd /usr/local/bin
[root@eks-bootstrap-server-new bin]# ll
total 198352
lrwxrwxrwx 1 root root 37 Sep 11 20:42 aws -> /usr/local/aws-cli/v2/current/bin/aws
lrwxrwxrwx 1 root root 68 Sep 11 20:42 aws_completer -> /usr/local/aws-cli/v2/current/bin/aws_completer
-rwxr-xr-x 1 1001 127 143462480 Sep 4 11:12 kubectl
-rwxr-xr-x 1 root root 51454184 Sep 11 20:55 kubectl
[root@eks-bootstrap-server-new bin]# mv eksctl /tmp
[root@eks-bootstrap-server-new bin]# cd /tmp
[root@eks-bootstrap-server-new tmp]# ll
total 140106
-rwxr-xr-x 1 1001 127 143462480 Sep 4 11:12 eksctl
drwxr----- 3 root root 68 Sep 11 20:37 systemd-private-e271e300a4b1a7cac15788e81287b28-chronynd.service-lidDBHU
drwxr----- 3 root root 68 Sep 11 20:37 systemd-private-e271e300a4b1a7cac15788e81287b28-dbus-broker.service-LQ0byb
drwxr----- 3 root root 68 Sep 11 20:37 systemd-private-e271e300a4b1a7cac15788e81287b28-policy-routes@enX0.service-oLOCPn
drwxr----- 3 root root 68 Sep 11 20:37 systemd-md-private-e271e300a4b1a7cac15788e81287b28-systemd-logind.service-VYbgk
drwxr----- 3 root root 68 Sep 11 20:37 systemd-md-private-e271e300a4b1a7cac15788e81287b28-systemd-resolved.service-tbdRy6
[root@eks-bootstrap-server-new tmp]# 
```

Next we gotta create an IAM role

Since you already created one, you use it again.

Creation date	ARN	Instance profile ARN
September 03, 2024, 16:14 (UTC-05:00)	arn:aws:iam::754173272413:role/eksctl_role	arn:aws:iam::754173272413:instance-profile/eksctl_role

Last activity	Maximum session duration
7 days ago	1 hour

Permissions

Permissions policies (4) [Info](#)

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AdministratorAccess	AWS managed - job function	2
AmazonEC2FullAccess	AWS managed	3
AWSCloudFormationFullAccess	AWS managed	1
IAMFullAccess	AWS managed	1

Attach it.

Instances (1/4) [Info](#) Last updated less than a minute ago

[Find Instance by attribute or tag \(case-sensitive\)](#)

[All states](#)

[Launch instances](#)

[Connect](#)

[View details](#)

[Manage instance state](#)

[Instance settings](#)

[Networking](#)

[Security](#)

[Change security groups](#)

[Get Windows password](#)

[Modify IAM role](#)

[Image and templates](#)

[Monitor and troubleshoot](#)

Name	Instance ID	Instance state	Instance type
ansible-server-new	i-026de678694f4616b	Running	t2.micro
eks-bootstrap-ec2-new	i-0af3e1e83b3479337	Running	t2.micro
Jenkins-server	i-07ab8daeb4dcb7eb	Running	t2.micro
Docker-Host	i-0eddf466bff854197	Running	t2.micro

[Instances](#)

[Instances](#)

[Instance Types](#)

[Launch Templates](#)

[Spot Requests](#)

[Savings Plans](#)

[Reserved Instances](#)

[Dedicated Hosts](#)

[Capacity Reservations](#)

[Images](#)

[AMIs](#)

[AMI Catalog](#)

[Elastic Block Store](#)

[Volumes](#)

[Snapshots](#)

[Lifecycle Manager](#)

[Network & Security](#)

[Security Groups](#)

[Elastic IP's](#)

[CloudShell](#)

[Feedback](#)

[© 2024, Amazon Web Services, Inc. or its affiliates.](#)

[Privacy](#)

[Terms](#)

[Cookie preferences](#)

EC2 > Instances > i-0af3e1e83b3479337 > Modify IAM role

Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID
i-0af3e1e83b3479337 (eks-bootstrap-ec2-new)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

eksctl_role

Create new IAM role

Cancel [Update IAM role](#)

[Modify IAM role](#)

[CloudShell](#)

[Feedback](#)

[© 2024, Amazon Web Services, Inc. or its affiliates.](#)

[Privacy](#)

[Terms](#)

[Cookie preferences](#)

Now, this bootstrap server have access to almost all the A W S services because we have given the administrative privileges. Otherwise we need to give I A M EC2 and cloud formation access. Now from the bootstrap image, I can able to create the cluster. How to create the cluster we are going to see in the next lecture.

Next create cluster;

First you gave this as prof but error

```
eksctl create cluster --name valaxy \
--region us-east-1 \
--node-type t2.small
-bash: /usr/local/bin/eksctl: No such file or directory
```

Error so,

```
/tmp/eksctl create cluster --name nitheshskmr-cluster --region us-east-1 --node-type t2.small
```

This worked

```
[root@eks-bootstrap-server-new tmp]# /tmp/eksctl create cluster --name nitheshskmr-cluster --region us-east-1 --node-type t2.small
2024-09-11 22:42:50 [!] eksctl version 0.35.1
2024-09-11 22:42:50 [!] running on us-east-1
2024-09-11 22:42:51 [!] setting availability zones to [us-east-1a us-east-1b]
2024-09-11 22:42:51 [!] subnets for us-east-1a - public:192.168.0.0/19 private:192.168.64.0/19
2024-09-11 22:42:51 [!] subnets for us-east-1b - public:192.168.32.0/19 private:192.168.96.0/19
2024-09-11 22:42:51 [!] nodegroup "ng-109b585a" will use "" ([AmazonLinux2/1.30]
2024-09-11 22:42:51 [!] using Kubernetes version 1.30
2024-09-11 22:42:51 [!] creating EKS cluster "nitheshskmr-cluster" in "us-east-1" region with managed nodes
2024-09-11 22:42:51 [!] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2024-09-11 22:42:51 [!] if you encounter any issues, check CloudFormation console or try `eksctl utils describe-stacks --region=us-east-1 --cluster=nitheshskmr-cluster`
2024-09-11 22:42:51 [!] Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "nitheshskmr-cluster" in "us-east-1"
2024-09-11 22:42:51 [!] CloudWatch logging will not be enabled for cluster "nitheshskmr-cluster" in "us-east-1"
2024-09-11 22:42:51 [!] you can enable it with `eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE} (e.g. all) --region=us-east-1 --cluster=nitheshskmr-cluster`
2024-09-11 22:42:51 [!] default addons vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2024-09-11 22:42:51 [!] sequential tasks: { create cluster control plane "nitheshskmr-cluster",
  2 sequential sub-tasks:
    2 sequential sub-tasks:
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "ng-109b585a",
  }
)
2024-09-11 22:42:51 [!] building cluster stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:42:51 [!] deploying stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:43:21 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:43:51 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:44:51 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:46:01 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:46:31 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:47:31 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:48:31 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:49:51 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:50:51 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:51:51 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:52:51 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-cluster"
2024-09-11 22:52:52 [!] 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'
2024-09-11 22:52:52 [!] creating addon
2024-09-11 22:52:53 [!] successfully created addon
2024-09-11 22:52:53 [!] creating addon
2024-09-11 22:52:53 [!] successfully created addon
2024-09-11 22:52:53 [!] creating addon
2024-09-11 22:52:54 [!] successfully created addon
2024-09-11 22:52:54 [!] building managed nodegroup stack "eksctl-nitheshskmr-cluster-nodegroup-ng-109b585a"
2024-09-11 22:54:54 [!] deploying stack "eksctl-nitheshskmr-cluster-nodegroup-ng-109b585a"
2024-09-11 22:54:54 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-nodegroup-ng-109b585a"
2024-09-11 22:55:17 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-nodegroup-ng-109b585a"
2024-09-11 22:56:17 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-nodegroup-ng-109b585a"
2024-09-11 22:57:34 [!] waiting for CloudFormation stack "eksctl-nitheshskmr-cluster-nodegroup-ng-109b585a"
2024-09-11 22:57:34 [!] waiting for the control plane to become ready
2024-09-11 22:57:38 [!] saved kubeconfig as "/root/.kube/config"
2024-09-11 22:57:38 [!] no tasks
2024-09-11 22:57:38 [!] no pending cluster resources; for "nitheshskmr-cluster" have been created
2024-09-11 22:57:38 [!] created 0 nodegroup(s) in cluster "nitheshskmr-cluster"
2024-09-11 22:57:38 [!] nodegroup "ng-109b585a" has 2 node(s)
2024-09-11 22:57:38 [!] node "ip-192-168-40-35.ec2.internal" is ready
2024-09-11 22:57:38 [!] node "ip-192-168-5-203.ec2.internal" is ready
2024-09-11 22:57:38 [!] waiting for at least 2 node(s) to become ready in "ng-109b585a"
2024-09-11 22:57:38 [!] nodegroup "ng-109b585a" has 2 node(s)
```

To create a deployment.

This step running the yaml file manually is not required cuz we will be running the yaml file in ansible or Jenkins for automation

```
○ ○ ○ Downloads — root@eks-bootstrap-server-new:~ — ssh -i devopsproject.pem ec2-user@54.205.151.15...
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nitheshskmr-regapp
  labels:
    app: regapp

spec:
  replicas: 3
  selector:
    matchLabels:
      app: regapp

template:
  metadata:
    labels:
      app: regapp
  spec:
    containers:
      - name: regapp
        image: nitheshskmr/regapp
        imagePullPolicy: Always
        ports:
          - containerPort: 8080
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 1
~
~
~
-- INSERT --
```

```

current-context: i-0af3e1e83b3479337@nitheshskmr-cluster.us-east-1.eksctl.io
kind: Config
preferences: {}
users:
- name: i-0af3e1e83b3479337@nitheshskmr-cluster.us-east-1.eksctl.io
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1beta1
      args:
        - eks
        - get-token
        - --output
        - json
        - --cluster-name
        - nitheshskmr-cluster
        - --region
        - us-east-1
      command: aws
      env:
        - name: AWS_STS_REGIONAL_ENDPOINTS
          value: regional
      provideClusterInfo: false
[root@eks-bootstrap-server-new ~]# 
[root@eks-bootstrap-server-new ~]# kubectl get all
NAME                           CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
services/kubernetes            10.96.0.1    <none>        443/TCP   16m
[root@eks-bootstrap-server-new ~]# cd -
[root@eks-bootstrap-server-new ~]# ls
aws  awsecli2.zip  eksctl_Linux_amd64.tar.gz
[root@eks-bootstrap-server-new ~]#
[root@eks-bootstrap-server-new ~]# vi regapp-deployment.yaml
[root@eks-bootstrap-server-new ~]# cat regapp-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nitheshskmr-regapp
  labels:
    app: regapp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: regapp
  template:
    metadata:
      labels:
        app: regapp
    spec:
      containers:
        - name: regapp
          image: nitheshskmr/regapp
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
[root@eks-bootstrap-server-new ~]# 

```

You can see here, three parts has been created and it is creating.

And service, we haven't created yet.

And a deployment, one deployment.

And so far, none of the parts are ready.

That is the reason 0 by three is showing.

And replica set, it is trying to create three.

Three containers are not yet ready.

One more time, let's run it.

This time, it should be running.

Yes, it is running.

And deployments 3 by 3 and replicaset 3 are ready.

Okay, so now our application is running.

Now I would like to access our registry app application.

54. Integrating ansible and Kubernetes

First we create a user ansadmin

Create password

Add him to sudoers file visudo

Vi /etc/ssh/sshd_config

```
[root@eks-bootstrap-server-new ~]# 
[root@eks-bootstrap-server-new ~]# 
[root@eks-bootstrap-server-new ~]# useradd ansadmin
[root@eks-bootstrap-server-new ~]# passwd ansadmin
Changing password for user ansadmin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@eks-bootstrap-server-new ~]# visudo
[root@eks-bootstrap-server-new ~]# vi /etc/ssh/sshd_config
[root@eks-bootstrap-server-new ~]# service sshd reload
Redirecting to /bin/systemctl reload sshd.service
[root@eks-bootstrap-server-new ~]# 
```

Downloads — root@eks-bootstrap-server-new:~ — ssh -i devopsproject.pem ec2-user@54.205.151.15...

```
GNU nano 5.8                               /etc/sudoers.tmp                         Modified
## Syntax:
##
##      user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root      ALL=(ALL)        ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys   ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel    ALL=(ALL)        ALL

## Same thing without a password
# %wheel    ALL=(ALL)        NOPASSWD: ALL
ansadmin  ALL=(ALL)        NOPASSWD: ALL
## Allows members of the users group to mount and umount the
## cdrom as root
# %users   ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users   localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute      ^C Location     M-U Undo
^X Exit         ^R Read File     ^\ Replace       ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo
```

```
● ○ ● ○ Downloads — root@eks-bootstrap-server-new:~ — ssh -i devopsproject.pem ec2-user@54.205.151.15...
#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 no
PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
```

68,1

40%

```
# 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
```

Now, I need to add Bootstrap image to the hosts file.
In this case, I'm going to create inventory file
here itself, that is hosts

Go to ansible ec2

Add the private ips of Kubernetes and ansible here

Next we gotta copy keys.

Ssh-copy-id <Kubernetes private ip>

It asks password of Kubernetes ansadmin

Now, I can test it by using the

ansible all -a uptime

I'm just checking the uptime of all our systems.
So, sorry, it'll go and check the **/etc/ansible** file.

Now, I need to pass

ansible -i hosts all -a uptime

hosts all, which means that in the current location,
we have hosts file.

In this host file, whatever server is there,
ping to that server.

```
● ● ● Downloads — ansadmin@ansible-server-new:/opt/docker — ssh -i devopsproject.pem ec2-user@54.197.96.232 — 132x37
[ansadmin@ansible-server-new docker]$ vi hosts
[ansadmin@ansible-server-new docker]$ ssh-copy-id 172.31.95.236
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host '172.31.95.236 (172.31.95.236)' can't be established.
ECDSA key fingerprint is SHA256:S/jvthYlaYyX6E9mtjueGEdyHOpkz4BLbio1X6fZn4.
ECDSA key fingerprint is MD5:a2:b5:9d:9e:13:f8:54:97:29:16:af:96:0f:c9:db:18.
Are you sure you want to continue connecting (yes/no)? 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
ansadmin@172.31.95.236's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '172.31.95.236'"
and check to make sure that only the key(s) you wanted were added.

[ansadmin@ansible-server-new docker]$
[ansadmin@ansible-server-new docker]$ ansible -i hosts all -a uptime
[WARNING]: Platform linux on host 172.31.95.236 is using the discovered Python interpreter at
/usr/bin/python3, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
172.31.95.236 | CHANGED | rc=0 >>
 00:44:30 up 4:06, 3 users, load average: 0.00, 0.00, 0.00
[WARNING]: Platform linux on host localhost is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
localhost | CHANGED | rc=0 >>
 00:44:30 up 8:26, 3 users, load average: 0.08, 0.02, 0.01
[WARNING]: Platform linux on host 172.31.83.187 is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
172.31.83.187 | CHANGED | rc=0 >>
 00:44:30 up 8:26, 2 users, load average: 0.08, 0.02, 0.01
[ansadmin@ansible-server-new docker]$
```

Next we need to copy our ansadmin keys
onto the root user,
otherwise it cannot able to log in as a root user.
For that, ssh-copy-id onto root user of this server.
This is our bootstrap server, okay?
So let's give enter and it is asking for password
of root user of k8s ec2. (password is admin)

ssh-copy-id root@ <priv ip of Kubernetes ec2>

So video 55 6 mins. You faced a big issue. When you did above step and enter password it kept throwing permission denied but you gave ansadmin password but here you had to create a password for root. So you went to eks ec2 created a password for root, came back gave

ssh-copy-id root@ <priv ip of Kubernetes ec2>
again permission denied, so you chatgptd, it said to open
vi /etc/ssh/sshd_config and change PermitRootLogin prohibit-password to PermitRootLogin,
then you reloaded it sudo systemctl restart sshd and to verify sudo systemctl status sshd.
Then it worked

```
[root@eks-bootstrapserver-new ~]#
[root@eks-bootstrapserver-new ~]#
[root@eks-bootstrapserver-new ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Re-type new password:
passwd: all authentication tokens updated successfully.
[root@eks-bootstrapserver-new ~]#
[root@eks-bootstrapserver-new ~]# vi /etc/ssh/sshd_config
[root@eks-bootstrapserver-new ~]# vi /etc/ssh/sshd_config
[root@eks-bootstrapserver-new ~]# sudo systemctl restart sshd
[root@eks-bootstrapserver-new ~]# sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-09-12 01:11:10 UTC; 1s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
 Main PID: 13331 (sshd)
   Tasks: 3 (limit: 1132)
  Memory: 7.9M
    CPU: 54ms
   CGroup: /system.slice/sshd.service
           └─13331 sshd: /usr/sbin/sshd -D [listener] 1 of 10-100 startups*
             ├─13394 sshd: unknown [priv]
             ├─13395 sshd: unknown [net]
             └─13396 sshd: unknown [net]

Sep 12 01:11:13 eks-bootstrapserver-new sshd[13390]: Failed none for invalid user debian from 54.188.246.123 port 55802
Sep 12 01:11:14 eks-bootstrapserver-new sshd[13390]: Connection closed by invalid user debian 54.188.246.123 port 55802
Sep 12 01:11:14 eks-bootstrapserver-new sshd[13392]: Invalid user admin from 54.188.246.123 port 55802
Sep 12 01:11:14 eks-bootstrapserver-new sshd[13392]: pam_unix(sshd:auth): check pass: user unknown
Sep 12 01:11:14 eks-bootstrapserver-new sshd[13392]: pam_unix(sshd:auth): authentication failure; logname=uid=0 euid=0 tty=pts/0 ruser= rhost= user=
Sep 12 01:11:16 eks-bootstrapserver-new sshd[13392]: Failed password for invalid user admin from 54.188.246.123 port 55802
Sep 12 01:11:18 eks-bootstrapserver-new sshd[13392]: Connection closed by invalid user admin 54.188.246.123 port 55802
Sep 12 01:11:19 eks-bootstrapserver-new sshd[13394]: Invalid user admin from 54.188.246.123 port 55804
Sep 12 01:11:19 eks-bootstrapserver-new sshd[13394]: pam_unix(sshd:auth): check pass; user unknown
Sep 12 01:11:19 eks-bootstrapserver-new sshd[13394]: pam_unix(sshd:auth): authentication failure; logname=uid=0 euid=0 tty=pts/0 ruser= rhost= user=
[root@eks-bootstrapserver-new ~]#
```

Jenkins issue resolved

So your jenkins was slow right everytime you stopped and started. its not cuz cpu and memory utilization.

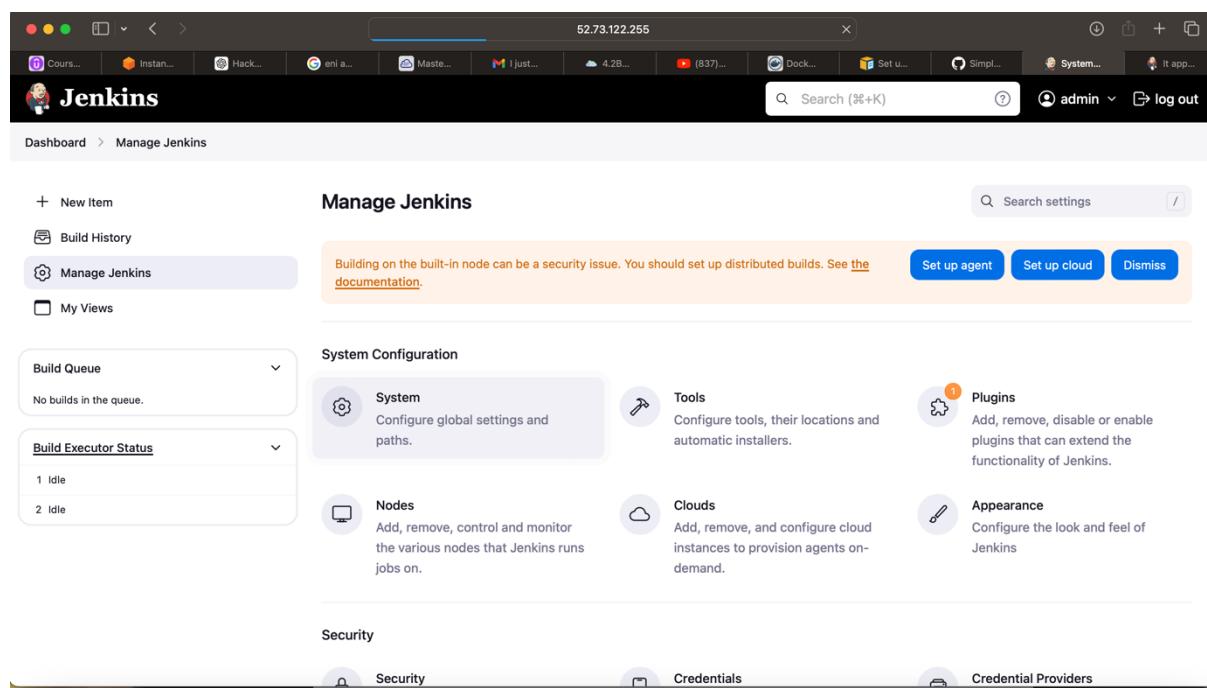
So i opened jenkins gui it was slow, and then on manage jenkins, my jenkins dashboard threwed “ it appears that your reverse proxy set up is broken” so i chatgptd and

Go to Jenkins → Manage Jenkins → System

Update Jenkins URL:

- Find the Jenkins URL field

It showed the old public ip in the url, i mean the public ip i first created the ec2. and then since you had stopped and restarted it the public ip has changed, so i changed the url with latest public ip. and then boom super fast



Jenkins Location

Jenkins URL ?
http://3.91.95.216:8080/

System Admin e-mail address ?
address not configured yet <nobody@nowhere>

Serve resource files from another domain

Resource Root URL ?
[empty input field]

Favourites

Frequently Visited

Privacy Report

In the last seven days, Safari has prevented 139

Global properties

Disable deferred wipeout on this node ?

Save Apply

Old ip

Not Secure – 52.73.122.255

Jenkins Location

Jenkins URL ?
http://3.91.95.216:8080/

System Admin e-mail address ?
address not configured yet <nobody@nowhere>

Serve resource files from another domain

Resource Root URL ?
[empty input field]

Without a resource root URL, resources will be served from the Jenkins URL with Content-Security-Policy set.

Global properties

Disable deferred wipeout on this node ?

Save Apply

New ip

The screenshot shows the Jenkins System configuration page. It includes fields for Jenkins URL (http://52.73.122.255:8080/), System Admin e-mail address (address not configured yet <nobody@nowhere>), and Resource Root URL (empty). There is also a checkbox for Disable deferred wipeout on this node. At the bottom are Save and Apply buttons.

So create a new job freestyle cuz theres no build in CD, Also no need of GIT or anything, just click post build action → send build artifacts over over ssh → ansible server

The screenshot shows the Jenkins New Item creation page. The item name is set to Deploy_on_Kubernetes. The item type is selected as Freestyle project. Other options shown include Maven project, Pipeline, Multi-configuration project, and Folder. An OK button is at the bottom.

Jenkins

Dashboard > Deploy_on_kubernetes > Configuration

Configure

General

Enabled

Description
Deploy on kubernetes

Plain text [Preview](#)

Discard old builds [?](#)

GitHub project

This project is parameterized [?](#)

Throttle builds [?](#)

Execute concurrent builds if necessary [?](#)

Advanced [▼](#)

Source Code Management

[Save](#) [Apply](#)

Dashboard > Deploy_on_kubernetes > Configuration

Advanced [▼](#)

Configure

General

Source Code Management

None

Git [?](#)

Build Triggers

Trigger builds remotely (e.g., from scripts) [?](#)

Build after other projects are built [?](#)

Build periodically [?](#)

GitHub Branches

GitHub Pull Requests [?](#)

GitHub hook trigger for GITScm polling [?](#)

Poll SCM [?](#)

Build Environment

Delete workspace before build starts

[Save](#) [Apply](#)

The screenshot shows the Jenkins configuration interface for a job named 'Deploy_on_kubernetes'. The 'Build Steps' section is currently selected. A dropdown menu titled 'Add post-build action' is open, listing various Jenkins actions. The 'Send build artifacts over SSH' option is highlighted with a light blue background.

So, Jenkins job is going to execute the playbooks.
Playbooks is going to execute the deployment.

Open ansible ec2 and create kube_deploy.yml

```
---
- hosts: kubernetes
# become: true
user: root

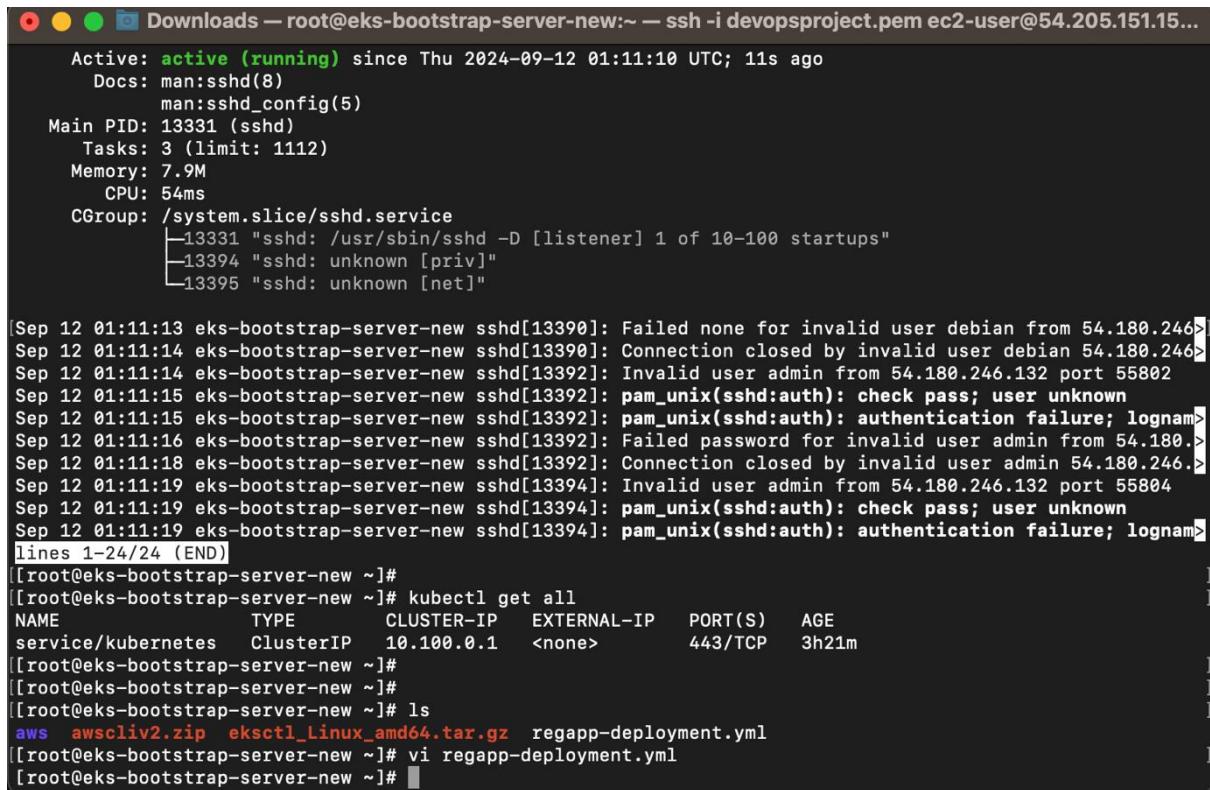
tasks:
  - name: deploy regapp on kubernetes
    command: kubectl apply -f regapp-deployment.yml

  - name: create service for regapp
    command: kubectl apply -f regapp-service.yml

  - name: update deployment with new pods if image updated in docker hub
    command: kubectl rollout restart deployment.apps/vs/regapp
```

Don't add this here itself cuz, only after you deploy once and see containers on eks ec2, there only you will find your "deployment.apps/vs/regapp" name. Means your deployments name. Until you deploy once you wont get that deployment name.

So until now you just have created regapp-deployment.yml on eks ec2 but you also need service.yml. These 2 will be combined and run inside a single playbook → kube_deploy.yml.



The screenshot shows a terminal window with the following content:

```
● ● ● ● Downloads — root@eks-bootstrap-server-new:~ — ssh -i devopsproject.pem ec2-user@54.205.151.15...
Active: active (running) since Thu 2024-09-12 01:11:10 UTC; 11s ago
  Docs: man:sshd(8)
        man:sshd_config(5)
Main PID: 13331 (sshd)
  Tasks: 3 (limit: 1112)
 Memory: 7.9M
   CPU: 54ms
CGroup: /system.slice/sshd.service
└─13331 "sshd: /usr/sbin/sshd -D [listener] 1 of 10-100 startups"
  ├─13394 "sshd: unknown [priv]"
  └─13395 "sshd: unknown [net]"

[Sep 12 01:11:13 eks-bootstrap-server-new sshd[13390]: Failed none for invalid user debian from 54.180.246.132 port 55802]
[Sep 12 01:11:14 eks-bootstrap-server-new sshd[13390]: Connection closed by invalid user debian 54.180.246.132 port 55802]
[Sep 12 01:11:14 eks-bootstrap-server-new sshd[13392]: Invalid user admin from 54.180.246.132 port 55802]
[Sep 12 01:11:15 eks-bootstrap-server-new sshd[13392]: pam_unix(sshd:auth): check pass; user unknown]
[Sep 12 01:11:15 eks-bootstrap-server-new sshd[13392]: pam_unix(sshd:auth): authentication failure; logname=""]
[Sep 12 01:11:16 eks-bootstrap-server-new sshd[13392]: Failed password for invalid user admin from 54.180.246.132 port 55802]
[Sep 12 01:11:18 eks-bootstrap-server-new sshd[13392]: Connection closed by invalid user admin 54.180.246.132 port 55802]
[Sep 12 01:11:19 eks-bootstrap-server-new sshd[13394]: Invalid user admin from 54.180.246.132 port 55804]
[Sep 12 01:11:19 eks-bootstrap-server-new sshd[13394]: pam_unix(sshd:auth): check pass; user unknown]
[Sep 12 01:11:19 eks-bootstrap-server-new sshd[13394]: pam_unix(sshd:auth): authentication failure; logname=""]
Lines 1-24/24 (END)

[[root@eks-bootstrap-server-new ~]# 
[[root@eks-bootstrap-server-new ~]# kubectl get all
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP  10.100.0.1   <none>       443/TCP   3h21m
[[root@eks-bootstrap-server-new ~]# 
[[root@eks-bootstrap-server-new ~]# 
[[root@eks-bootstrap-server-new ~]# ls
aws awscliv2.zip eksctl_Linux_amd64.tar.gz regapp-deployment.yml
[[root@eks-bootstrap-server-new ~]# vi regapp-deployment.yml
[[root@eks-bootstrap-server-new ~]# ]]
```

regapp-deployment.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nitheshskmr-regappnew
  labels:
    app: regappnew

spec:
  replicas: 3
  selector:
    matchLabels:
      app: regappnew

  template:
    metadata:
      labels:
        app: regappnew
    spec:
      containers:
        - name: regappnew
          image: nitheshskmr/regappnew
          imagePullPolicy: Always
          ports:
```

```
- containerPort: 8080
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 1
```

regapp-service.yml

```
apiVersion: v1
kind: Service
metadata:
  name: nitheshskmr-service
  labels:
    app: regappnew
spec:
  selector:
    app: regappnew

  ports:
    - port: 8080
      targetPort: 8080

  type: LoadBalancer
```

NOTE: names must be same

Regapp-deployment.yml

```
labels:
  app: regappnew
```

Regapp-service.yml

```
selector:
  app: regappnew
```

Regapp-deployment.yml

```
○ ○ ○ Downloads — root@eks-bootstrap-server-new:~ — ssh -i devopsproject.pem ec2-user@54.205.151.15...
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nitheshskmr-regappnew
  labels:
    app: regappnew

spec:
  replicas: 3
  selector:
    matchLabels:
      app: regappnew

  template:
    metadata:
      labels:
        app: regappnew
    spec:
      containers:
        - name: regappnew
          image: nitheshskmr/regappnew
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
~
~
~
-- INSERT --
```

Regapp-service.yml

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions**

Name ?

Advanced ▾

Transfers

Transfer Set

Source files ?

Either Source files, Exec command or both must be supplied

Remove prefix ?

Remote directory ?

Exec command ?

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

Advanced ▾

Save **Apply**

ansible-playbook -i /opt/docker/hosts /opt/docker/kube_deploy.yml;

You renamed Deploy_on_Kubernetes to Regapp_CD_job

New Item

Enter an item name

Select an item type

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.

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

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

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

OK

Jenkins

Dashboard > Regapp_CI_job > Configuration

Configure

General

Enabled

Description
Build code with the help of Maven and create an image on Ansible push it onto Dockerhub

Plain text [Preview](#)

Discard old builds [?](#)

GitHub project

This project is parameterized [?](#)

Throttle builds [?](#)

Execute concurrent builds if necessary [?](#)

Advanced [▼](#)

Source Code Management

[Save](#) [Apply](#)

This screenshot shows the 'General' configuration page for a Jenkins job named 'Regapp_CI_job'. The 'Enabled' checkbox is checked. The 'Description' field contains the text 'Build code with the help of Maven and create an image on Ansible push it onto Dockerhub'. Below the description are several checkboxes for build options: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. There is also an 'Advanced' dropdown menu. At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > Regapp_CI_job > Configuration

Configure

General

Execute concurrent builds if necessary [?](#)

Advanced [▼](#)

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)
https://github.com/Nithesh67/hello-world.git

Credentials [?](#)
- none -

+ Add [▼](#)

Advanced [▼](#)

Add Repository

[Save](#) [Apply](#)

This screenshot shows the 'Source Code Management' configuration page. It has a 'Git' repository selected. The 'Repository URL' is set to 'https://github.com/Nithesh67/hello-world.git'. The 'Credentials' dropdown is currently empty, showing '- none -'. There is an '+ Add' button for adding new credentials. An 'Advanced' dropdown is also present. At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > Regapp_CI_job > Configuration

Configure

Pre Steps

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps**
- Build
- Post Steps
- Build Settings
- Post-build Actions

Add pre-build step ▾

Build

Root POM ?
pom.xml

Goals and options ?
clean install

Advanced ▾

Post Steps

Run only if build succeeds

Run only if build succeeds or is unstable

Run regardless of build result

Save **Apply**

Dashboard > Regapp_CI_job > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps
- Build**
- Post Steps
- Build Settings
- Post-build Actions**

SSH Server

Name ?
ansible-server-new

Advanced ▾

Transfers

Transfer Set

Source files ?
webapp/target/*.war

Remove prefix ?
webapp/target

Remote directory ?
/opt/docker

Exec command ?
ansible-playbook /opt/docker/create_image_regapp.yml;

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

Advanced ▾

Save **Apply**

You ran CI job. ➔ SUCCESS

Jenkins

Dashboard > Regapp_CI_job >

Maven project Regapp_CI_job

Build code with the help of Maven and create an image on Ansible push it onto Dockerhub

Status Changes Workspace Build Now Configure Delete Maven project Modules Rename

Permalinks

Build History trend Filter... / #1 12-Sep-2024, 4:09 am Atom feed for all Atom feed for failures

REST API Jenkins 2.462.2

Before

Docker Hub Explore Repositories Organizations Usage Search Docker Hub

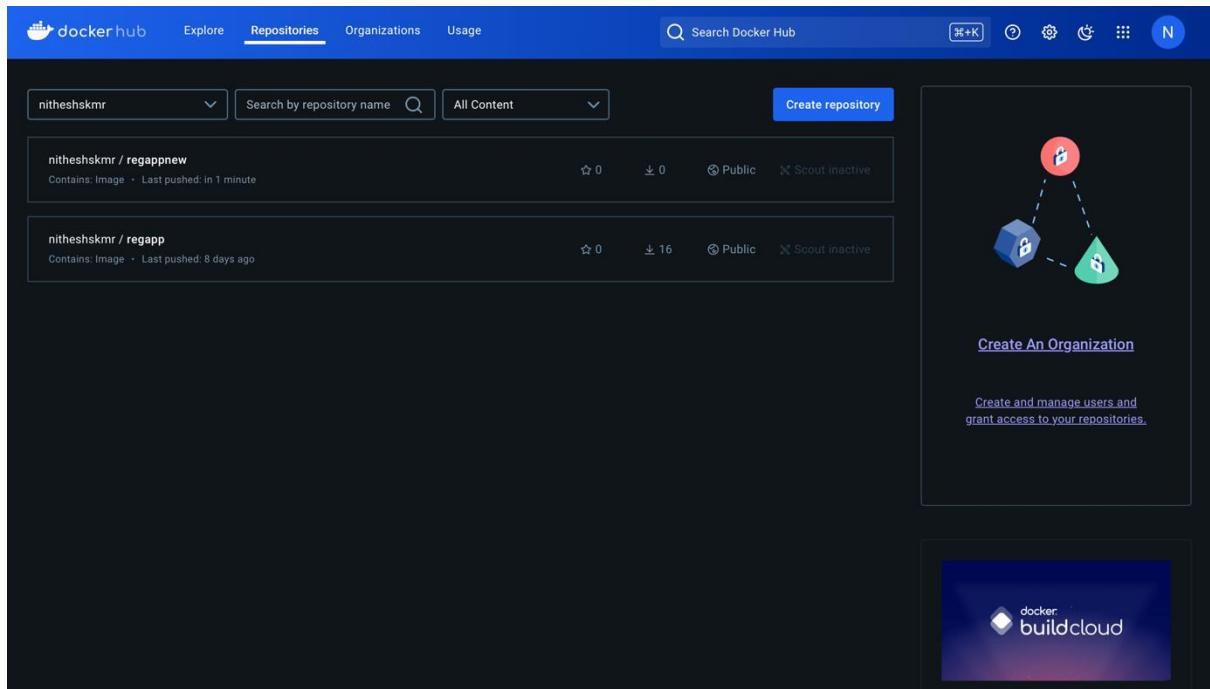
nitheshskmr Search by repository name All Content Create repository

nitheshskmr / regapp Contains: Image · Last pushed: 8 days ago 0 16 Public Scout inactive

Create An Organization Create and manage users and grant access to your repositories.

docker buildcloud

After



```
Downloads — ansadmin@ansible-server-new:/opt/docker — ssh -i devopsproject.pem ec2-user@54.197.96.232 — 132x29
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
/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@172.31.95.236's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@172.31.95.236'"
and check to make sure that only the key(s) you wanted were added.

[ansadmin@ansible-server-new docker]$ 
[ansadmin@ansible-server-new docker]$ vi kube_deploy.yml
[ansadmin@ansible-server-new docker]$ ls
Dockerfile  hosts  kube_deploy.yml
[ansadmin@ansible-server-new docker]$ 
[ansadmin@ansible-server-new docker]$ vi kube_deploy.yml
[ansadmin@ansible-server-new docker]$ vi create_image_regapp.yml
[ansadmin@ansible-server-new docker]$ vi kube_deploy.yml
[ansadmin@ansible-server-new docker]$ vi kube_deploy.yml
[ansadmin@ansible-server-new docker]$ vi create_image_regapp.yml
[ansadmin@ansible-server-new docker]$ ll
total 20
-rw-rw-r-- 1 ansadmin ansadmin  347 Sep 12 04:07 create_image_regapp.yml
-rw-rw-r-- 1 ansadmin ansadmin 128 Sep 11 17:55 Dockerfile
-rw-rw-r-- 1 ansadmin ansadmin   63 Sep 12 00:39 hosts
-rw-rw-r-- 1 ansadmin ansadmin 247 Sep 12 04:03 kube_deploy.yml
-rw-rw-r-- 1 ansadmin ansadmin 2365 Sep 12 04:09 webapp.war
[ansadmin@ansible-server-new docker]$
```

Now you go enable Poll scm and post build actions to trigger Regapp_CD_job.

Dashboard > Regapp_CI_job > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

Build Triggers

Build whenever a SNAPSHOT dependency is built ?
 Schedule build when some upstream has no successful builds ?
 Trigger builds remotely (e.g., from scripts) ?
 Build after other projects are built ?
 Build periodically ?
 GitHub Branches
 GitHub Pull Requests ?
 GitHub hook trigger for GITScm polling ?

Poll SCM ?
 Schedule ?
 **** *

No schedules so will only run due to SCM changes if triggered by a post-commit hook

Ignore post-commit hooks ?

Build Environment

[Save](#) [Apply](#)

Dashboard > Regapp_CI_job > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

Filter

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Deploy artifacts to Maven repository
- GitHub PR: add labels
- GitHub PR: close PR
- GitHub PR: post comment
- GitHub PR: remove labels
- GitHub PR: set PR status
- Record fingerprints of files to track usage
- Git Publisher
- Editable Email Notification
- Send build artifacts over SSH
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Delete workspace when build is done

Add post-build action ^

[Save](#) [Apply](#)

Dashboard > Regapp_CI_job > Configuration

Build Settings

Configure

E-mail Notification

Post-build Actions

Build other projects

Projects to build: Regapp_CD_job, ! No such project 'R'. Did you mean 'Regapp_CD_job'?

Trigger only if build is stable
 Trigger even if the build is unstable
 Trigger even if the build fails

Send build artifacts over SSH

SSH Publishers

SSH Server

Name: aws2_ec2_ip

Save **Apply**

EC2 > Security Groups > sg-0ee1e494325a12211 - Jenkins_Security_Group > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-0a695fc53edb34c3a	Custom TCP	TCP	8080	Custom	<input type="text"/> 0.0.0.0/0 X
sgr-0734b66ebf5b576c8	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0 X
-	Custom TCP	TCP	8081	Anyw...	<input type="text"/> 0.0.0.0/0 X
-	Custom TCP	TCP	8082	Anyw...	<input type="text"/> 0.0.0.0/0 X
-	Custom TCP	TCP	8083	Anyw...	<input type="text"/> 0.0.0.0/0 X
-	Custom TCP	TCP	8084	Anyw...	<input type="text"/> 0.0.0.0/0 X

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

Just to verify nothing has been deployed yet

```
● ○ ● ○ Downloads — root@eks-bootstrap-server-new:~ — ssh -i devopsproject.pem ec2-user@54.205.151.15...
[root@eks-bootstrap-server-new ~]# kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP  10.100.0.1   <none>     443/TCP   3h21m
[root@eks-bootstrap-server-new ~]#
[root@eks-bootstrap-server-new ~]#
[root@eks-bootstrap-server-new ~]# ls
aws awscliv2.zip eksctl_Linux_amd64.tar.gz regapp-deployment.yml
[root@eks-bootstrap-server-new ~]# vi regapp-deployment.yml
[root@eks-bootstrap-server-new ~]# vi create regapp-service.yml
2 files to edit
[root@eks-bootstrap-server-new ~]#
[root@eks-bootstrap-server-new ~]# vi regapp-deployment.yml
[root@eks-bootstrap-server-new ~]# vi regapp-service.yml
[root@eks-bootstrap-server-new ~]# ls
aws awscliv2.zip eksctl_Linux_amd64.tar.gz regapp-deployment.yml regapp-service.yml
[root@eks-bootstrap-server-new ~]# rm create
rm: remove regular file 'create'? y
[root@eks-bootstrap-server-new ~]# ls
aws awscliv2.zip eksctl_Linux_amd64.tar.gz regapp-deployment.yml regapp-service.yml
[root@eks-bootstrap-server-new ~]# vi regapp-service.yml
[root@eks-bootstrap-server-new ~]# vi regapp-deployment.yml
[root@eks-bootstrap-server-new ~]# vi regapp-service.yml
[root@eks-bootstrap-server-new ~]# ls
aws awscliv2.zip eksctl_Linux_amd64.tar.gz regapp-deployment.yml regapp-service.yml
[root@eks-bootstrap-server-new ~]#
[root@eks-bootstrap-server-new ~]#
[root@eks-bootstrap-server-new ~]# kubectl get all
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP  10.100.0.1   <none>     443/TCP   5h22m
[root@eks-bootstrap-server-new ~]# kubectl get all
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP  10.100.0.1   <none>     443/TCP   5h35m
[root@eks-bootstrap-server-new ~]#
```

Now you open local git and modify code.

```
● ○ ● ○ webapp — --bash — 104x32
[master b96904c] Rampa name entry
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) Nithehs-MBP:webapp apple$ git push origin master
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 538 bytes | 269.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Nithesh67/hello-world.git
  fea9895..b96904c master -> master
(base) Nithehs-MBP:webapp apple$ vi index.jsp
(base) Nithehs-MBP:webapp apple$ git add .
(base) Nithehs-MBP:webapp apple$ git commit -m "Raman paru's name entry "
[master e9c6a68] Raman paru's name entry
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) Nithehs-MBP:webapp apple$ git push origin master
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 539 bytes | 23.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Nithesh67/hello-world.git
  b96904c..e9c6a68 master -> master
(base) Nithehs-MBP:webapp apple$ 
(base) Nithehs-MBP:webapp apple$ 
(base) Nithehs-MBP:webapp apple$ 
(base) Nithehs-MBP:webapp apple$ ls
WEB-INF      index.jsp
(base) Nithehs-MBP:webapp apple$ vi index.jsp
(base) Nithehs-MBP:webapp apple$
```

```

webapp — vi index.jsp — 104x32
<form action="action_page.php">
  <div class="container">
    <h1>New user Register</h1>
    <p>Please fill in this form to create an account.</p>
    <hr>

    <label for="Name"><b>Enter Name</b></label>
    <input type="text" placeholder="Enter Last Name" name="Name" id="Name" required>
    <br>

    <label for="mobile"><b>Enter mobile</b></label>
    <input type="text" placeholder="Enter mobile number" name="mobile" id="mobile" required>
    <br>

    <label for="email"><b>Enter Email</b></label>
    <input type="text" placeholder="Enter Email" name="email" id="email" required>
    <br>

    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="psw" id="psw" required>
    <br>

    <label for="psw-repeat"><b>Repeat Password</b></label>
    <input type="password" placeholder="Repeat Password" name="psw-repeat" id="psw-repeat" required>
    <hr>
    <br>
    <p>By creating an account you agree to our <a href="#">Terms & Privacy</a>.</p>
    <button type="submit" class="registerbtn">Register</button>
  </div>
  <div class="container signin">
    <p>Already have an account? <a href="#">Sign in</a>.</p>
-- INSERT --

```

Success. It automatically executed CI job, then triggered CD job and success.

The screenshot shows the Jenkins dashboard for the Maven project **Regapp_CI_job**. The left sidebar includes options like Status, Changes, Workspace, Build Now, Configure, Delete Maven project, Modules, Git Polling Log, Rename, and a Build History section showing builds #2 and #1. The main content area displays the latest test result (no failures), downstream projects (Regapp_CD_job), and a Test Result Trend chart. The trend chart shows two green bars representing successful builds (#1 and #2) from 12-Sep-2024, with no red or grey bars indicating failed or skipped tests.

Jenkins

Dashboard > Regapp_CD_job >

Status: **Regapp_CD_job** (green checkmark)

Deploy on kubernetes

Upstream Projects

Regapp_CI_job (green checkmark)

Permalinks

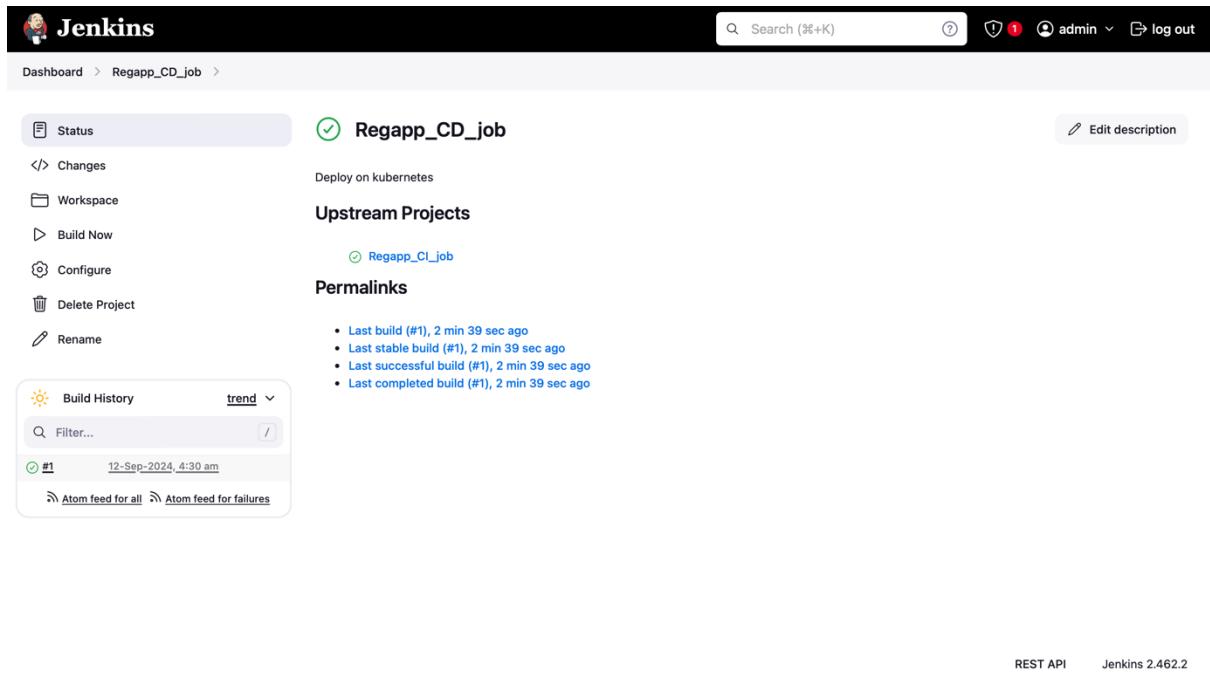
- Last build (#1), 2 min 39 sec ago
- Last stable build (#1), 2 min 39 sec ago
- Last successful build (#1), 2 min 39 sec ago
- Last completed build (#1), 2 min 39 sec ago

Build History (trend dropdown) | Filter... | /

#1 12-Sep-2024, 4:30 am

Atom feed for all Atom feed for failures

REST API Jenkins 2.462.2



Docker Hub

Explore Repositories Organizations Usage Search Docker Hub

nitheshskmr Search by repository name All Content Create repository

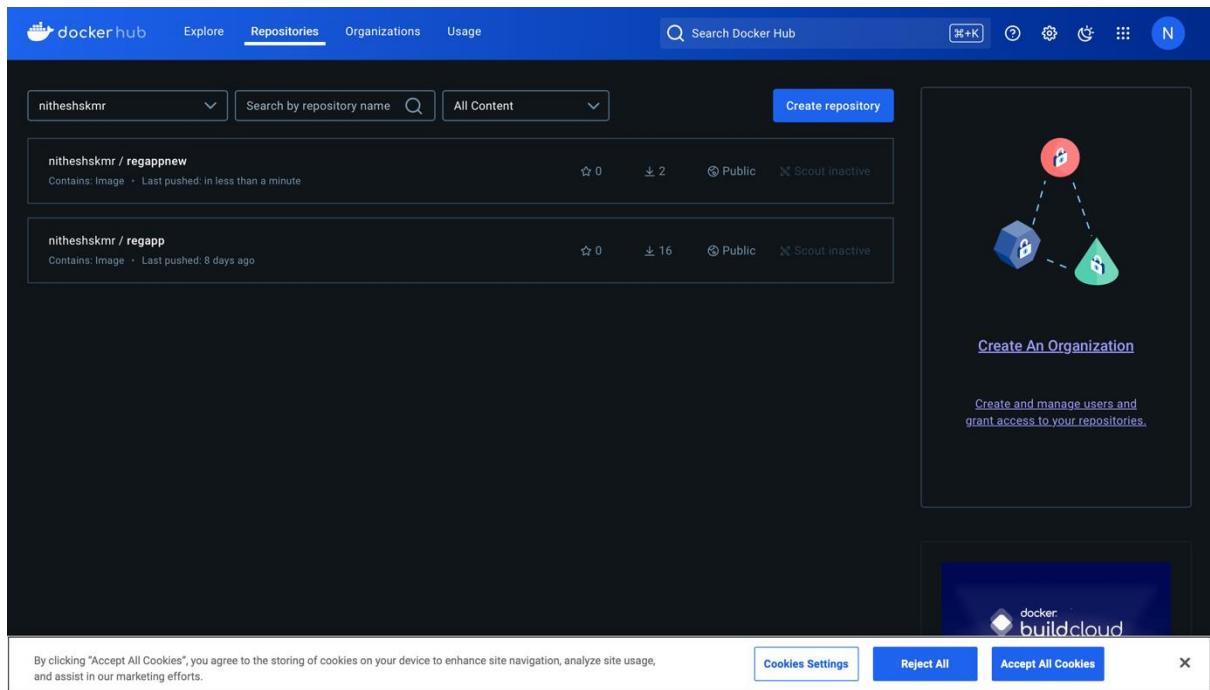
nitheshskmr / regappnew
Contains: Image · Last pushed: in less than a minute

nitheshskmr / regapp
Contains: Image · Last pushed: 8 days ago

Create An Organization
Create and manage users and grant access to your repositories.

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

Cookies Settings Reject All Accept All Cookies X



```

Downloads — root@eks-bootstrap-server-new:~ — ssh -i devopsproject.pem ec2-user@54.205.151.15...
[[root@eks-bootstrap-server-new ~]# vi regapp-service.yml
[[root@eks-bootstrap-server-new ~]# vi regapp-deployment.yml
[[root@eks-bootstrap-server-new ~]# vi regapp-service.yml
[[root@eks-bootstrap-server-new ~]# ls
aws awscli v2.zip eksctl_Linux_amd64.tar.gz regapp-deployment.yml regapp-service.yml
[[root@eks-bootstrap-server-new ~]#
[[root@eks-bootstrap-server-new ~]#
[[root@eks-bootstrap-server-new ~]# kubectl get all
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes   ClusterIP  10.100.0.1 <none>       443/TCP   5h22m
[[root@eks-bootstrap-server-new ~]# kubectl get all
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes   ClusterIP  10.100.0.1 <none>       443/TCP   5h35m
[[root@eks-bootstrap-server-new ~]#
[[root@eks-bootstrap-server-new ~]# kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/nitheshskmr-regappnew-ffbb8b55-8qv2x  1/1    Running   0          87s
pod/nitheshskmr-regappnew-ffbb8b55-dj6fb  1/1    Running   0          87s
pod/nitheshskmr-regappnew-ffbb8b55-jlrsx  1/1    Running   0          87s
[[root@eks-bootstrap-server-new ~]# kubectl get all
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP
PORT(S)        AGE
service/kubernetes   ClusterIP  10.100.0.1 <none>
443/TCP        5h43m
service/nitheshskmr-service   LoadBalancer  10.100.162.146 a34fdf8bbb0b14851a100e5ea6014e33-272552752.us-east-1.elb.amazonaws.com  8080:31706/TCP  86s
[[root@eks-bootstrap-server-new ~]# kubectl get all
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nitheshskmr-regappnew  3/3     3           3          87s
[[root@eks-bootstrap-server-new ~]# kubectl get all
NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/nitheshskmr-regappnew-ffbb8b55  3         3         3      87s
[[root@eks-bootstrap-server-new ~]# ]

```

The screenshot shows the AWS CloudShell interface with a terminal window displaying the execution of several commands. The commands include creating deployment and replicaset files, extracting an AWS CLI zip file, and running Kubernetes commands like `vi` to edit files, `ls` to list files, `kubectl` to get and apply configurations, and `aws` to create a load balancer. The output shows the creation of services, pods, and a load balancer named 'a34fdf8bbb0b14851a100e5ea6014e33-272552752.us-east-1.elb.amazonaws.com'.

With load balancer url:8080/webapp

New user Register

Please fill in this form to create an account

Enter Name	Enter Last 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](#).

Thankyou, Happy Learning

Now if you make changes again, the containers will not change, for that you gotta make changes to your `kube_deploy.yml`

Here we have your option that is update deployment with the new pod if the image is updated on Docker hub. So that is kube CTL rollout command.

Just add it to your playbook that's your kube_deploy.

Add this but with your deployment name

- name: update deployment with new pods if image updated in docker hub
command: kubectl rollout restart **deployment.apps/valaxy-regapp**

```
Downloads -- ansadmin@ansible-server-new:/opt/docker - ssh -i devopsproject.pem ec2-user@54.205.151.15...  
--  
- hosts: kubernetes  
# become: true  
user: root  
  
tasks:  
- name: deploy regapp on kubernetes  
  command: kubectl apply -f regapp-deployment.yml  
  
- name: create service for regapp  
  command: kubectl apply -f regapp-service.yml  
  
- name: update deployment with new pods if image updated in docker hub  
  command: kubectl rollout restart deployment.apps/nitheshskmr-regappnew  
  
[root@eks-bootstrap-server-new ~]# aws s3cp s3://eksbootstrap.s3.amazonaws.com/ekstclv2.zip eksctl_Linux_amd64.tar.gz regapp-deployment.yml regapp-service.yml  
[root@eks-bootstrap-server-new ~]# [root@eks-bootstrap-server-new ~]# kubectl get all  
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE  
service/kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP  5h22m  
[root@eks-bootstrap-server-new ~]# kubectl get all  
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE  
service/kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP  5h35m  
[root@eks-bootstrap-server-new ~]# kubectl get all  
NAME          READY  STATUS    RESTARTS   AGE  
pod/nitheshskmr-regappnew-ffbb8b55-8qv2x  1/1   Running   0          87s  
pod/nitheshskmr-regappnew-ffbb8b55-dj6fb  1/1   Running   0          87s  
pod/nitheshskmr-regappnew-ffbb8b55-jirsx  1/1   Running   0          87s  
  
[root@eks-bootstrap-server-new ~]# kubectl get all  
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE  
service/kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP  5h43m  
service/nitheshskmr-service  LoadBalancer  10.100.162.146  a34fdf8bb0b14851a100e5ea014e33-272552752.u...  
s-east-1.elb.amazonaws.com  8080:31706/TCP  86s  
  
[root@eks-bootstrap-server-new ~]# kubectl get all  
NAME          READY  UP-TO-DATE  AVAILABLE  AGE  
deployment.apps/nitheshskmr-regappnew  3/3   3           3          87s  
  
[root@eks-bootstrap-server-new ~]# kubectl get all  
NAME          DESIRED  CURRENT  READY  AGE  
replicaset.apps/nitheshskmr-regappnew-ffbb8b55  3       3       3       87s  
[root@eks-bootstrap-server-new ~]#
```

New user Register

Please fill in this form to create an account.

Enter Name	Enter Last 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](#).