

Setting up ci/cd pipeline using jenkins to deploy on kubernetes

- The first step would be for us to set up an EC2 instance and on this instance, we will be installing -
 - JDK
 - Jenkins
 - eksctl
 - Kubectl

Create and Launch EC2 instance of type t2.medium with ubuntu 20.04 OS and with all other options as default. Also connect to the ec-2 instance using Putty for Windows or using ssh for ubuntu respectively.

The screenshot shows the AWS Launch Instance wizard at step 2: Choose an Instance Type. The user has selected the t2.medium instance type. The table lists various t2 instances with their details:

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
t3	t3.micro	2	4	EBS only	Yes	Up to 5 Gigabit	Yes

At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Instance Details'.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot Instances	
Network	vpc-b917d2d2 (default)	<input type="button"/> Create new VPC
Subnet	No preference (default subnet in any Availability Zone)	<input type="button"/> Create new subnet
Auto-assign Public IP	<input type="button"/> Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	
Domain join directory	No directory	<input type="button"/> Create new directory
IAM role	None <input type="button"/> Create new IAM role	
Shutdown behavior	Stop	
Stop - Hibernate behavior	<input type="checkbox"/> Enable hibernation as an additional stop behavior	
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring	

Review and Launch

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-09e03b16b827cccd7	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Shared file systems

You currently don't have any file systems on this instance. Select "Add file system" button below to add a file system.

Add file system

Review and Launch

How to set up AWS Kubernetes Jenkins | How to setup kubernetes jenkins pipe | Launch instance wizard | EC2 Main | Devops/Team_B_DevOps_Jenkins_Ass... | Jenkins Report - Google Docs

Spinbot.com - Artic... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Soft

aws Services Search for services, features, blogs, docs, and more [Alt+S]

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
jenkins-ec2				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Feedback English (US) ▾ © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

How to set up AWS Kubernetes Jenkins | How to setup kubernetes jenkins pipe | Launch instance wizard | EC2 Main | Devops/Team_B_DevOps_Jenkins_Ass... | Jenkins Report - Google Docs

Spinbot.com - Artic... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Soft

aws Services Search for services, features, blogs, docs, and more [Alt+S]

jaganmohan reddy g Mumbai Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group

Select an existing security group

Security group name: launch-wizard-31
Description: launch-wizard-31 created 2021-11-13T23:53:51.987+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Custom 0.0.0.0/:/0	e.g. SSH for Admin Desktop

Add Rule



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

Cancel Previous Review and Launch

Feedback English (US) ▾ © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

How to set up AWS Kubernetes Jenkins | How to setup kubernetes jenkins pipe | Launch instance wizard | EC2 Main | Devops/Team_B_Devops_Jenkins_Ass... | Jenkins Report - Google Docs

<https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard>

Spinbot.com - Article... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Soft

aws Services Search for services, features, blogs, docs, and more [Alt+S]

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

AMI Details

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0567e0d2b4b2169ae

Free tier eligible Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>). Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.medium	-	2	4	EBS only	-	Low to Moderate

Security Groups

Security group name: launch-wizard-31
Description: launch-wizard-31 created 2021-11-13T23:53:51.987+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
Custom TCP Rule	TCP	8080	0.0.0.0/0	
Custom TCP Rule	TCP	8080	::/0	

Cancel Previous Launch

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

How to set up AWS Kubernetes Jenkins | How to setup kubernetes jenkins pipe | Launch instance wizard | EC2 Main | Devops/Team_B_Devops_Jenkins_Ass... | Jenkins Report - Google Docs

<https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard>

Spinbot.com - Article... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Soft

aws Services Search for services, features, blogs, docs, and more [Alt+S]

Launch Status

Your instances are now launching. The following instance launches have been initiated: i-0629dfe26bb84b93a View launch log

Get notified of estimated charges Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances. Click [View Instances](#) to monitor your instances' status. Once your instances are in the running state, you can connect to them from the Instances screen. Find out how to connect to your instances.

Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

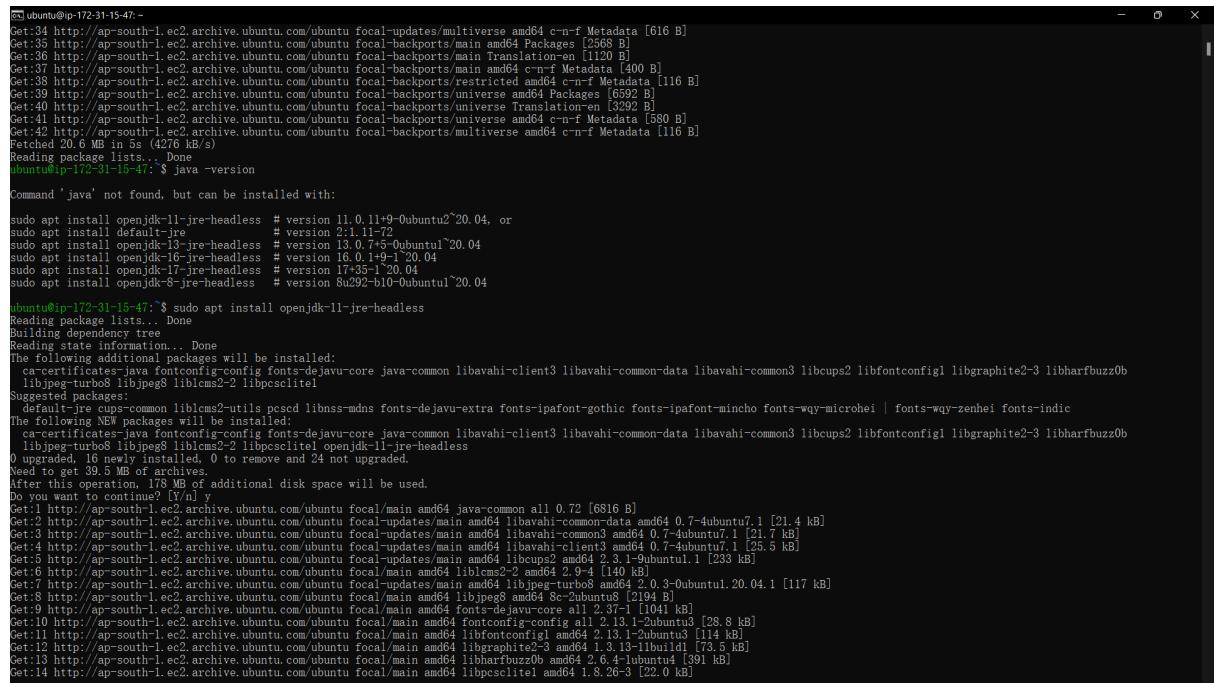
While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Install JDK on AWS EC2 Instance

```
sudo apt-get update
sudo apt install openjdk-11-jre-headless
java -version
```



The screenshot shows a terminal window on an AWS EC2 instance with the IP address 172-31-15-47. The user runs the command 'java -version' which returns an error: 'Command 'java' not found, but can be installed with:'. The user then runs 'sudo apt install openjdk-11-jre-headless' and the output shows several packages being downloaded from the 'ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe' repository. The packages include various Java-related components like libavahi-client3, libavahi-common3, libcurl3, libfontconfig1, libgraphite2-3, libharfbuzz0b, libjpeg-turbo0.1, libjpeg8, liblcms2-2, libpsclite1, and others. The download progress is shown with 'Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libpsclite1 amd64 1.8.26-3 [22.0 kB]' at the bottom.

```
Get:34 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [616 B]
Get:35 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [268 B]
Get:36 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [1120 B]
Get:37 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [400 B]
Get:38 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:39 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [6592 B]
Get:40 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [3292 B]
Get:41 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [580 B]
Get:42 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 20.6 MB in 5s (4276 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-15-47:~$ java -version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-11-jre-headless # version 11.0.11+9-Ubuntu20.04, or
sudo apt install default-jre # version 2:1.11-72
sudo apt install openjdk-13-jre-headless # version 13.0.7+5-Ubuntu1~20.04
sudo apt install openjdk-16-jre-headless # version 16.0.1+9-1~20.04
sudo apt install openjdk-17-jre-headless # version 17+35-1~20.04
sudo apt install openjdk-8-jre-headless # version 8u292-b10-Ubuntu1~20.04
ubuntu@ip-172-31-15-47:~$ sudo apt install openjdk-11-jre-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  certificates-java fontconfig-config fonts-dejavu-core java-common libavahi-client3 libavahi-common-data libavahi-common3 libcurl3 libfontconfig1 libgraphite2-3 libharfbuzz0b
  libjpeg-turbo0.1 libjpeg8 liblcms2-2 libpsclite1
Suggested packages:
  default-jre fonts-common liblcms2-utils pscd libnss-mdns fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  certificates-java fontconfig-config fonts-dejavu-core java-common libavahi-client3 libavahi-common-data libavahi-common3 libcurl3 libfontconfig1 libgraphite2-3 libharfbuzz0b
  libjpeg-turbo0.1 libjpeg8 liblcms2-2 libpsclite1 openjdk-11-jre-headless
0 upgraded, 16 newly installed, 0 to remove and 24 not upgraded.
0 upgraded, 16 newly installed, 0 to remove and 24 not upgraded.
Need to get 39.5 MB of archives.
After this operation, 178 MB of additional disk space will be used.
Do you want to continue? [y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 java-common all 0.72 [6816 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libavahi-common-data amd64 0.7-4ubuntu7.1 [21.4 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libavahi-common3 amd64 0.7-4ubuntu7.1 [21.7 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libavahi-client3 amd64 0.7-4ubuntu7.1 [25.5 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl3 amd64 2.3.1-9ubuntu1.1 [233 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 liblcms2-2 amd64 2.9.4-2 [140 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libjpeg-turbo8 amd64 2.0.3-0ubuntu1.20.04.1 [117 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libjpeg8 amd64 8c-2ubuntu8 [219 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 fontconfig-config all 2.37-1 [1041 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 fontconfig-config all 2.13.1-2ubuntu3 [28.8 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libfontconfig1 amd64 2.13.1-2ubuntu3 [114 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 liblcms2-2 amd64 1.3.13-1ubuntu1 [73.5 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libharfbuzz0b amd64 2.6.4-1ubuntu4 [391 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libpsclite1 amd64 1.8.26-3 [22.0 kB]
```

Install and Setup Jenkins

Setup jenkins

- add the Jenkins repository to the package manager

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
```

- After adding the repository link of Jenkins update the package manager

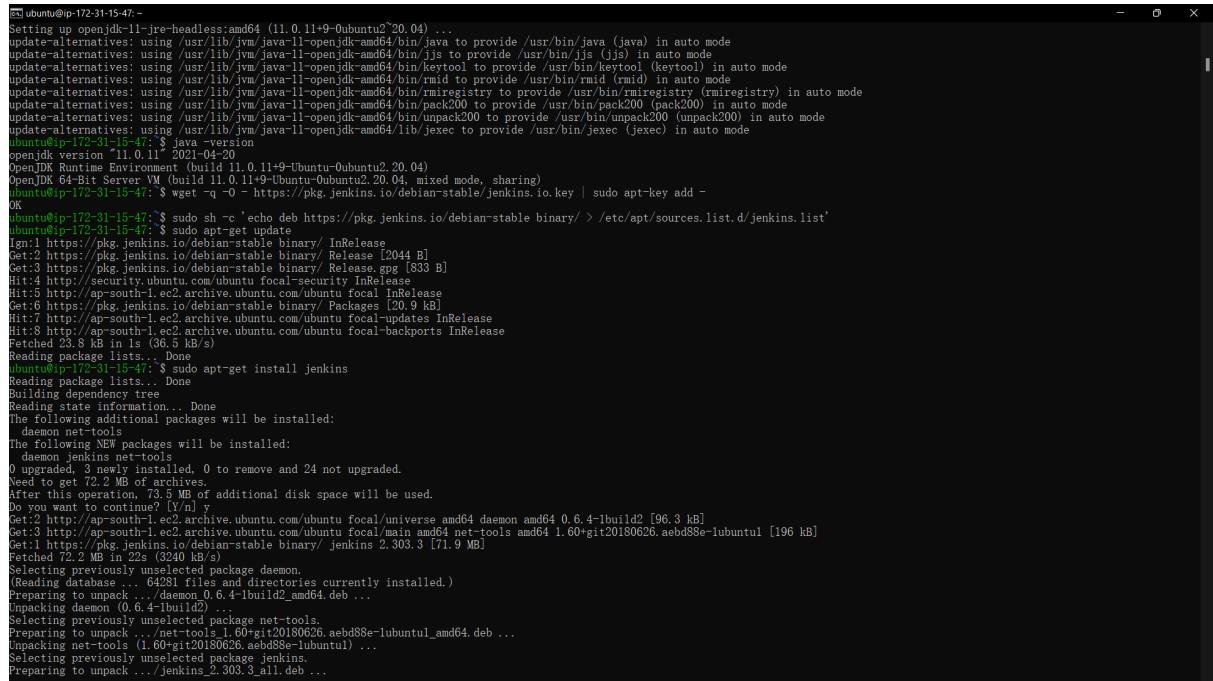
```
sudo apt-get update
```

- Then finally install Jenkins using the following command

```
sudo apt-get install jenkins
```

- On successful installation, you should see Active Status

```
sudo service jenkins status
```



```
ubuntu@ip-172-31-15-47:~$ sudo apt-get install jenkins
Setting up openjdk-11-jre-headless:amd64 (11.0.11+9-Ubuntu2.20.04) ...
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/java to provide /usr/bin/java (java) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jjs to provide /usr/bin/jjs (jjs) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/keytool to provide /usr/bin/keytool (keytool) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/rmid to provide /usr/bin/rmid (rmid) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/rmiregistry to provide /usr/bin/rmiregistry (rmiregistry) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/pack200 to provide /usr/bin/pack200 (pack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/unpack200 to provide /usr/bin/unpack200 (unpack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/lib/jexec to provide /usr/bin/jexec (jexec) in auto mode
Running java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-Ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-Ubuntu2.20.04, mixed mode, sharing)
ubuntu@ip-172-31-15-47:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
OK
ubuntu@ip-172-31-15-47:~$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
ubuntu@ip-172-31-15-47:~$ sudo apt-get update
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:2 https://pkg.jenkins.io/debian-stable/main/binary/ [2044 B]
Get:3 https://pkg.jenkins.io/debian-stable/main/binary/ InRelease [833 B]
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:6 https://pkg.jenkins.io/debian-stable binary/ Packages [20.9 kB]
Hit:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 23.8 kB in 1s (36.5 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-15-47:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon net-tools
The following NEW packages will be installed:
  daemon jenkins net-tools
0 upgraded, 3 newly installed, 0 to remove and 24 not upgraded.
Need to get 72.2 MB of archives.
After this operation, 73.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 daemon amd64 0.6.4-1build2 [96.3 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60-git20180626.aebdb88e-lubuntul [196 kB]
Get:3 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.303.3 [71.9 kB]
Fetched 72.2 MB in 22s (3240 kB/s)
Selecting previously unselected package daemon.
(Reading database ... 64281 files and directories currently installed.)
Preparing to unpack .../daemon_0.6.4-1build2_amd64.deb ...
Unpacking daemon (0.6.4-1build2) ...
Selecting previously unselected package net-tools.
Preparing to unpack .../net-tools_1.60-git20180626.aebdb88e-lubuntul_amd64.deb ...
Unpacking net-tools (1.60-git20180626.aebdb88e-lubuntul) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.303.3_all.deb ...
```

- Now we need to start using jenkins from the public ipv4 address of the instance created above. And jenkins by default runs on the port 8080.

```
<public-ipv4>:8080
```

- Now use the command to unlock jenkins to access it

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

How to set up AWS K... How to setup kubernetes Instance details | EC2... Devops/Team_8_Devops Jenkins Report - Google... Jenkins [Jenkins] New Tab Sign in [Jenkins]

Not secure | http://3.7.66.64.8080/login?from=%2F

Spinbot.com - Artic... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Soft

Getting Started

Unlock Jenkins

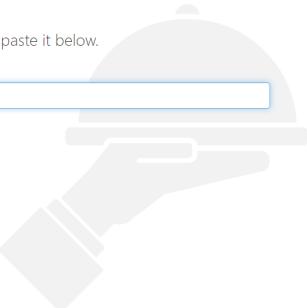
To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

.....



Continue

How to set up AWS K... How to setup kubernetes Instance details | EC2... Devops/Team_8_Devops Jenkins Report - Google... Jenkins [Jenkins] Setup Wizard [Jenkins] New Tab

Not secure | http://3.7.66.64.8080

Spinbot.com - Artic... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Soft

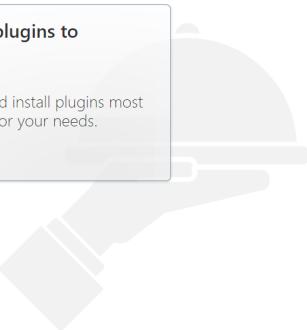
Getting Started

Customize Jenkins

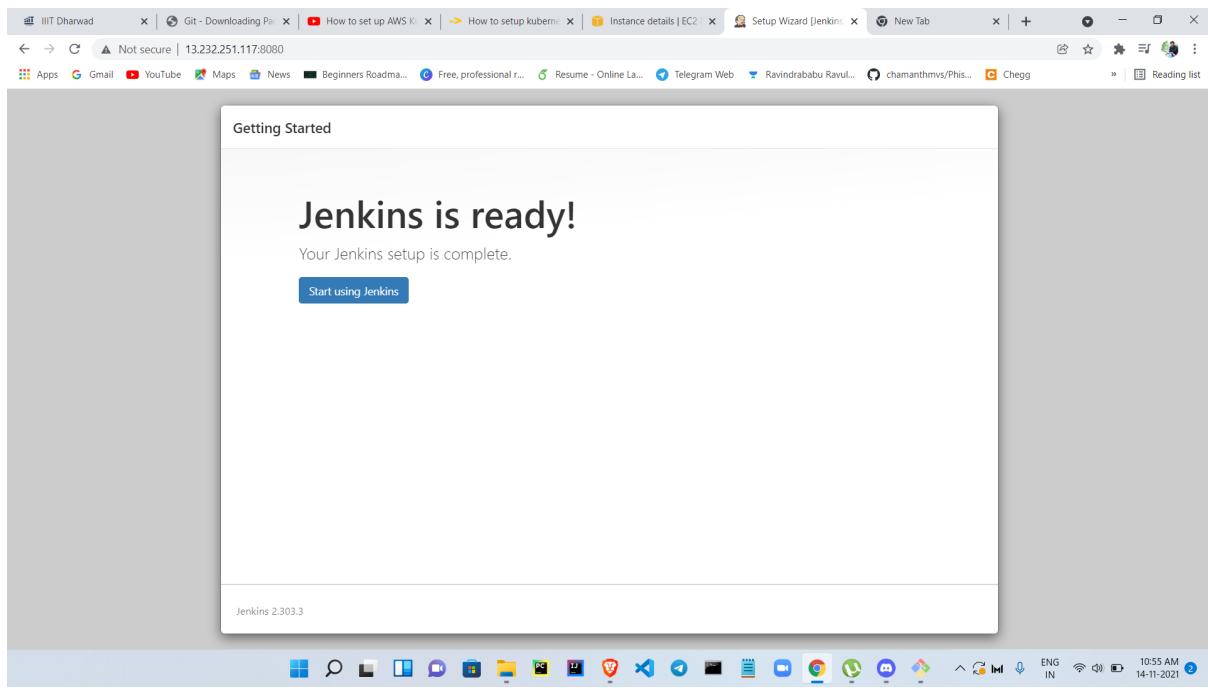
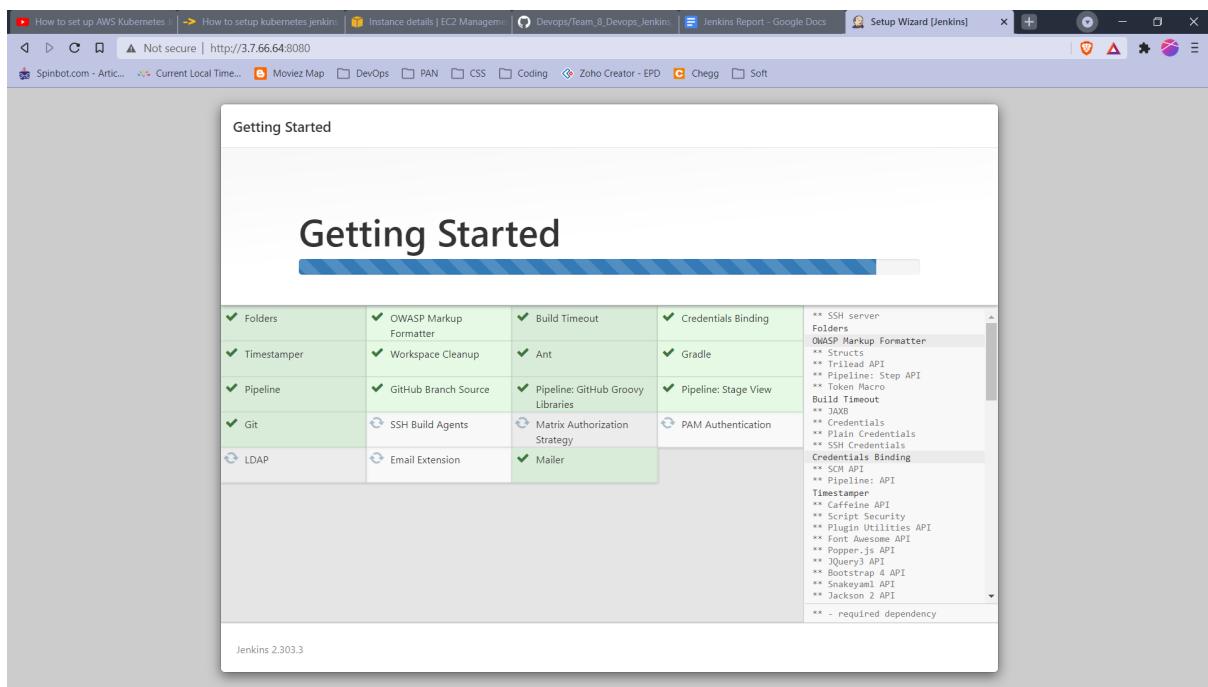
Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.

Select plugins to install
Select and install plugins most suitable for your needs.



Jenkins 2.303.3



The screenshot shows the Jenkins dashboard with the 'People' section selected. The left sidebar includes links for 'New Item', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. The main content area is titled 'People' and displays a table with one user entry:

User ID	Name	Last Commit Activity
aditya	Arepalli Sai Venkata Aditya	N/A

Below the table, it says 'Icon: S M L'. The system tray at the bottom right shows the date and time as 14-11-2021 11:04 AM.

Setup Gradle

- For setting up the gradle Goto -> Manage Jenkins -> Global Tool Configuration -> Gradle

The screenshot shows the 'Global Tool Configuration' page for 'Gradle'. Under the 'Gradle' section, there is a 'Gradle installations' section with a 'Add Gradle' button. Below it is a 'Gradle' configuration with a 'name' field set to 'default' and a checked 'Install automatically' checkbox. A 'Install from Gradle.org' sub-section shows a dropdown menu set to 'Gradle 7.1' with a 'Delete Installer' button. At the bottom of the 'Gradle' section is an 'Add Gradle' button and a note about listing installations. The 'Ant' section below has 'Save' and 'Apply' buttons. The system tray at the bottom right shows the date and time as 14-11-2021 11:05 AM.

Update visudo and assign administration privileges to jenkins user

- To interact with the Kubernetes cluster Jenkins will be executing the shell script with the Jenkins user, so the Jenkins user should have an administration(superuser) role assigned beforehand. Let's add jenkins user as an administrator and also assign NOPASSWD so that during the pipeline run it will not ask for root password. Open the file /etc/sudoers in vi mode.

```
sudo vi /etc/sudoers
```

- Add the following line at the end of the file

```
jenkins ALL=(ALL) NOPASSWD: ALL
```

- After adding the line save and quit the file. Now we can use Jenkins as root user and for that run the following command

```
sudo su - jenkins
```

Install Docker

- Now we need to install docker after installing Jenkins. The docker installation will be done by the Jenkins user because now it has root user privileges.
- Use the following command for installing the docker

```
sudo apt install docker.io
```

- After installing the docker you can verify it by simply typing the docker --version onto the terminal. It should return you with the latest version of docker. Jenkins will be accessing Docker for building the application Docker images, so we need to add the Jenkins user to the docker group.

```
sudo usermod -aG docker jenkins
```

```

ubuntu@ip-172-31-5-51:~ 
as vagrant@LAPTOP-HV0HF9M6 MINGW64 ~/OneDrive/Desktop/Devops
$ ssh -i "Devops.pem" ubuntu@ec2-13-232-251-117.ap-south-1.compute.amazonaws.com
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Nov 14 05:32:40 UTC 2021

System load: 0.0          Processes:           113
Usage of /: 28.0% of 7.69GB   Users logged in: 0
Memory usage: 38%          IPv4 address for eth0: 172.31.5.51
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.
  https://ubuntu.com/aws/pro

26 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Sun Nov 14 05:08:32 2021 from 45.118.72.187
ubuntu@ip-172-31-5-51:~$ sudo vi /etc/sudoers
ubuntu@ip-172-31-5-51:~$ sudo su - jenkins
jenkins@ip-172-31-5-51:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
bridge-utils contained dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
bridge-utils contained dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 24 not upgraded.
Need to get 74.5 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data/amd64 1.0.1-0ubuntu2.20.04.1 [4155 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.5.5-0ubuntu3-20.04.1 [33.0 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data all 2019052802 [5300 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2.2ubuntu2 [46.2 kB]

```

Install and Setup AWS CLI

- now we have our EC2 machine and Jenkins installed. Now we need to set up the AWS CLI on the EC2 machine so that we can use eksctl in the later stages. Let us get the installation done for AWS CLI

```
sudo apt install awscli
```

- Verify your AWS CLI installation by running the following command

```
aws --version
```

- It should return you with the version of CLI

```
ubuntu@ip-172-31-5-1:~$ Selecting previously unselected package docker.io.
Preparing to unpack .../0_docker.io_1:20.0.7~Ubuntu5-20.04.2_amd64.deb ...
Unpacking docker.io (20.0.7~Ubuntu5-20.04.2) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../8_ubuntu-fan_0.12.13_all.deb ...
Unpacking ubuntu-fan (0.12.13) ...
Setting up runc (1.0.1~Ubuntu2+20.04.1) ...
Setting up dns-root-data (2019052802) ...
Setting up libdnfl1:amd64 (1.33-2.2ubuntu2) ...
Setting up bridge-utils (1.6-2ubuntu1) ...
Setting up pigg (2.4-1) ...
Setting up contained (1.5.5~Ubuntu3-20.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker.io (20.0.7~Ubuntu5-20.04.2) ...
Adding group 'docker' (GID 120) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up dnsmasq-base (2.80-1.ubuntu1.4) ...
Setting up ubuntu-fan (0.12.13) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Processing triggers for systemd (245.4~Ubuntu13.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for libglib2.0-0 (2.34-0ubuntu9.2) ...
jenkins@ip-172-31-5-51:~$ docker --version
Docker version 20.10.7, build 20.10.7~Ubuntu5-20.04.2
jenkins@ip-172-31-5-51:~$ sudo usermod -aG docker jenkins
jenkins@ip-172-31-5-51:~$ sudo apt install awscli
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
awscliv2-common fontconfig fonts-droid-fallback fonts-noto-mono fonts-urw-base35 ghostscript groff gsfonts hicolor-icon-theme imagemagick imagemagick-6-common imagemagick-lsb liblilyvibre-text liblilyvibre2 liblftw3-double3 liblomp0 libbs9-common libic6 libibj5-0.35 liblmbase24 libimagequanto libjbigr libjbigr2dec0 liblqr-1-0 libmagick-lsb libmagickcore6-q16-6-extra libmagickwand6_q16-6 libmagickpmh0 libtopenexr24 libtopenp2-1 libpango-1.0-0 libpangofc2-1.0-0 libpaper-utils libpaper1 libpaper2 libtiff10 libwebp6 libwebpdmux3 libwmf6 libxaw7 libxcb-render0 libxcb-shm0 libxmu6 libxpm4 libxrender1 libxt6 netpbm poppler-data psutils python3 python3-docutils python3-jmespath python3-pygments python3-roman python3-rsa python3-s3transfer sgm1-base x11-common xml-core
Suggested packages:
fonts-noto-fonts-freefont-ttf fonts-ttx-pixops ghostscript-x imagemagick-doc autotrace cups-bsd | lpr | lprng enscript ffmpeg gimp grpnt grads graphviz povray radiance sane-utils texlive-base-bin transfig ufraw-backup xdg-utils liblftw3-bin liblftw3-dev inkscape liblxr-tools libwmf0.2-7-gtk poppler-utils fonts-japanese-mi fonts-japanese-gothic | fonts-ipafont-gothic fonts-arpthic-ukai fonts-arpthic-uming fonts-nanum docutils-doc fonts-linuxlibertine ttf-latin-libertine texlive-living-french texlive-latex-recommended python-pil-doc python3-pil-dbg python-pymgments-doc ttfb-bitstream-vera sgm1-base-doc debhelper
The following NEW packages will be installed:
awscli docutils common fontconfig fonts-droid-fallback fonts-noto-mono fonts-urw-base35 ghostscript groff gsfonts hicolor-icon-theme imagemagick imagemagick-6-common imagemagick-lsb liblilyvibre-text liblilyvibre2 liblftw3-double3 liblomp0 libbs9-common libic6 libibj5-0.35 liblmbase24 libimagequanto libjbigr libjbigr2dec0 liblqr-1-0 libmagick-lsb libmagickcore6-q16-6-extra libmagickwand6_q16-6 libmagickpmh0 libtopenexr24 libtopenp2-1 libpango-1.0-0 libpangofc2-1.0-0 libpaper-utils libpaper1 libpaper2 libtiff10 libwebp6 libwebpdmux3 libwmf6 libxaw7 libxcb-render0 libxcb-shm0 libxmu6 libxpm4 libxrender1 libxt6 netpbm poppler-data psutils python3 python3-docutils python3-jmespath python3-pygments python3-roman python3-rsa python3-s3transfer sgm1-base x11-common xml-core
```

Configure AWS CLI

- Now after installing the AWS CLI, let's configure the AWS CLI so that it can authenticate and communicate with the AWS environment. To configure the AWS the first command we are going to run is:

`aws configure`

- Once you execute the above command it will ask for the following information:
 - AWS Access Key ID [None]:
 - AWS Secret Access Key [None]:
 - Default region name [None]:
 - Default output format [None]:
 - You can find this information by going into AWS -> My Security Credentials. Then navigate to Access Keys (access key ID and secret access key). You can click on the Create New Access Key and it will let you generate - AWS Access Key ID, AWS Secret Access Key. Default region name - You can find it from the menu.

Install and Setup Kubectl

- Moving forward now we need to set up the kubectl also onto the EC2 instance where we set up the Jenkins in the previous steps. Here is the command for installing kubectl:

```
curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s
```

```
https://storage.googleapis.com/kubernetes-release/release/stable.txt) /bin/1
inux/amd64/kubectl"
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin
```

- Verify the kubectl installation by running the command kubectl version and you should see the following output

```
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.2",
GitCommit:"092fbfbf53427de67cacle9fa54aaa09a28371d7", GitTreeState:"clean",
BuildDate:"2021-06-16T12:59:11Z", GoVersion:"go1.16.5", Compiler:"gc",
Platform:"linux/amd64"}
Error from server (Forbidden): <html><head><meta http-equiv='refresh'
content='1;url=/login?from=%2Fversion%3Ftimeout%3D32s' /><script>window.loca
tion.replace('/login?from=%2Fversion%3Ftimeout%3D32s');</script></head><bod
y style='background-color:white; color:white;'>
```

Install and Setup eksctl

- The next thing which we are gonna do is to install the eksctl, which we will be using to create AWS EKS Clusters. Okay, the first command which we are gonna run to install the eksctl

```
curl --silent --location
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(una
me -s)_amd64.tar.gz" | tar xz -C /tmp
BASH
sudo mv /tmp/eksctl /usr/local/bin
```

- Verify the installation by running the command

```
eksctl version
```

- In all the previous steps we were preparing our AWS environment. Now in this step, we are going to create EKS cluster using eksctl.
- You need the following in order to run the eksctl command
 - Name of the cluster : -name jhooq-test-cluster
 - Version of Kubernetes : -version 1.17
 - Region : -name eu-central-1
 - Nodegroup name/worker nodes : worker-nodes
 - Node Type : t2.micro
 - Number of nodes: -nodes 2
- Here is the eksctl command

```
eksctl create cluster --name jhooq-test-cluster --version 1.17 --region eu-central-1 --nodegroup-name worker-nodes --node-type t2.micro --nodes 2
```

```
ubuntu@ip-172-31-5-51:~$ eksctl create cluster --name jhooq-test-cluster --version 1.17 --region ap-south-1 --nodegroup-name worker-nodes --node-type t2.micro --nodes 2
Error: Failed to create cluster "jhooq-test-cluster"
jenkins@ip-172-31-5-51:~$ eksctl create cluster --name anu --version 1.17 --region ap-south-1 --nodegroup-name worker-nodes --node-type t2.micro --nodes 2
2021-11-14 06:39:47 [ ] eksctl version 0.73.0
2021-11-14 06:39:47 [ ] using region ap-south-1
2021-11-14 06:39:47 [ ] using availability zones to [ap-south-1a ap-south-1c ap-south-1b]
2021-11-14 06:39:47 [ ] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.96.0/19
2021-11-14 06:39:47 [ ] subnets for ap-south-1c - public:192.168.32.0/19 private:192.168.128.0/19
2021-11-14 06:39:47 [ ] subnets for ap-south-1b - public:192.168.64.0/19 private:192.168.160.0/19
2021-11-14 06:39:47 [ ] using Kubernetes version 1.17
2021-11-14 06:39:47 [ ] creating EKS cluster "anu" in "ap-south-1" region with managed nodes
2021-11-14 06:39:47 [ ] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2021-11-14 06:39:47 [ ] you can enable CloudWatch logs for your cluster by trying eksctl utils describe-stacks --region=ap-south-1 --cluster=anu
2021-11-14 06:39:47 [ ] CloudWatch Logging will not be enabled for cluster "anu" in "ap-south-1"
2021-11-14 06:39:47 [ ] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=anu'
2021-11-14 06:39:47 [ ] Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "anu" in "ap-south-1"
2021-11-14 06:39:47 [ ] using 'aws' credential provider
2021-11-14 06:39:47 [ ] sequential tasks: [
2021-11-14 06:39:47 [ ]   2 sequential sub-tasks: [
2021-11-14 06:39:47 [ ]     "create cluster control plane 'anu'",

2021-11-14 06:39:47 [ ]     "sequential sub-tasks: [
2021-11-14 06:39:47 [ ]       "wait for control plane to become ready",
2021-11-14 06:39:47 [ ]       "create managed nodegroup 'worker-nodes',
2021-11-14 06:39:47 [ ]     ]
2021-11-14 06:39:47 [ ]   ]
2021-11-14 06:39:47 [ ]   building cluster stack "eksctl-anu-cluster"
2021-11-14 06:39:47 [ ]   deploying stack "eksctl-anu-cluster"
2021-11-14 06:40:17 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:40:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:41:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:42:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:43:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:44:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:45:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:46:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:47:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:48:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:49:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:50:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:51:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:52:48 [ ]   building managed nodegroup stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:53:48 [ ]   deploying stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:53:48 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:54:17 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:54:21 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:54:41 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:54:58 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:55:37 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:55:56 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:56:13 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:56:47 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:57:03 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:57:21 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:57:37 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:58:11 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:58:11 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:58:11 [ ]   waiting for the control plane availability...
2021-11-14 06:58:11 [ ]   saved kubeconfig as "/var/lib/jenkins/.kube/config"
2021-11-14 06:58:11 [ ]
```

```
ubuntu@ip-172-31-5-51:~$ eksctl create cluster --name jhooq-test-cluster --version 1.17 --region ap-south-1 --nodegroup-name worker-nodes --node-type t2.micro --nodes 2
2021-11-14 06:39:47 [ ] If you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=anu'
2021-11-14 06:39:47 [ ] CloudWatch Logging will not be enabled for cluster "anu" in "ap-south-1"
2021-11-14 06:39:47 [ ] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=anu'
2021-11-14 06:39:47 [ ] Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "anu" in "ap-south-1"
2021-11-14 06:39:47 [ ] using 'aws' credential provider
2021-11-14 06:39:47 [ ] sequential tasks: [
2021-11-14 06:39:47 [ ]   2 sequential sub-tasks: [
2021-11-14 06:39:47 [ ]     "create cluster control plane 'anu'",

2021-11-14 06:39:47 [ ]     "sequential sub-tasks: [
2021-11-14 06:39:47 [ ]       "wait for control plane to become ready",
2021-11-14 06:39:47 [ ]       "create managed nodegroup 'worker-nodes',
2021-11-14 06:39:47 [ ]     ]
2021-11-14 06:39:47 [ ]   ]
2021-11-14 06:39:47 [ ]   building cluster stack "eksctl-anu-cluster"
2021-11-14 06:39:47 [ ]   deploying stack "eksctl-anu-cluster"
2021-11-14 06:40:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:41:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:42:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:43:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:44:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:45:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:46:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:47:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:48:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:49:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:50:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:51:47 [ ]   waiting for CloudFormation stack "eksctl-anu-cluster"
2021-11-14 06:52:48 [ ]   building managed nodegroup stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:53:48 [ ]   deploying stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:53:48 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:54:05 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:54:21 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:54:41 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:54:58 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:55:18 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:55:37 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:55:56 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:56:13 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:56:47 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:57:03 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:57:21 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:57:37 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:58:11 [ ]   waiting for CloudFormation stack "eksctl-anu-nodegroup-worker-nodes"
2021-11-14 06:58:11 [ ]   waiting for the control plane availability...
2021-11-14 06:58:11 [ ]   saved kubeconfig as "/var/lib/jenkins/.kube/config"
2021-11-14 06:58:11 [ ]   no tasks
2021-11-14 06:58:11 [ ]   2 ready cluster resources for "anu" have been created
2021-11-14 06:58:11 [ ]   nodegroup "worker-nodes" has 2 node(s)
2021-11-14 06:58:11 [ ]   node "ip-192-168-19-207.ap-south-1.compute.internal" is ready
2021-11-14 06:58:11 [ ]   node "ip-192-168-86-198.ap-south-1.compute.internal" is ready
2021-11-14 06:58:11 [ ]   waiting for at least 2 node(s) to become ready in "worker-nodes"
2021-11-14 06:58:11 [ ]   nodegroup "worker-nodes" has 2 node(s)
2021-11-14 06:58:11 [ ]   node "ip-192-168-19-207.ap-south-1.compute.internal" is ready
2021-11-14 06:58:11 [ ]   node "ip-192-168-86-198.ap-south-1.compute.internal" is ready
2021-11-14 06:58:11 [ ]   kubectl command should work with "/var/lib/jenkins/.kube/config", try 'kubectl get nodes'
2021-11-14 06:58:11 [ ]   eksctl command is ready
jenkins@ip-172-31-5-51:~$
```

Verify the EKS kubernetes cluster from AWS

- You can go back to your AWS dashboard and look for Elastic Kubernetes Service -> Clusters. Click on the Cluster Name to verify the worker nodes.

The screenshot shows the AWS EKS console with the URL ap-south-1.console.aws.amazon.com/eks/home?region=ap-south-1#/clusters/anu. The left sidebar shows 'Amazon Container Services' with 'Amazon ECS' and 'Clusters' selected. The main content area shows the 'Clusters' page for 'anu'. A message box states: 'This cluster is running the oldest Kubernetes version currently supported by Amazon EKS. Ensure that your cluster is updated before the version end of support date.' with a 'Learn more' link. Below is a table of 'Nodes (2)'.

Node name	Instance type	Node Group	Created	Status
ip-192-168-19-207.ap-south-1.compute.internal	t2.micro	worker-nodes	3 minutes ago	Ready
ip-192-168-86-198.ap-south-1.compute.internal	t2.micro	worker-nodes	3 minutes ago	Ready

Add Docker and GitHub Credentials into Jenkins

- Kubernetes is a container orchestration tool and container management we are using docker. so if you are reading this line then I am assuming you have a DockerHub Account and GitHub Account. Here is the link of GitHub Repository for this project. You can set the docker credentials by going into : Goto -> Jenkins -> Manage Jenkins -> Manage Credentials -> Stored scoped to jenkins -> global -> Add Credentials.
- Now we add one more username and password for GitHub. Goto -> Jenkins -> Manage Jenkins -> Manage Credentials -> Stored scoped to jenkins -> global -> Add Credentials.

T	P	Store	Domain	ID	Name
		Jenkins	(global)	DOCKER_HUB_PASSWORD	Docker Credentials
		Jenkins	(global)	GIT_HUB_CREDENTIALS	LaxmiNarayanaK***** (Git Hub Credentials)

Icon: S M L

P	Store	Domains
	Jenkins	!(global)

Add jenkins stages

Okay, now we can start writing out the Jenkins pipeline for deploying the Spring Boot Application into the Kubernetes Cluster. Jenkins stage-1 : Checkout the GitHub Repository.

```
stage("Git Clone") {
    git credentialsId: 'GIT_HUB_CREDENTIALS', url:
    'https://github.com/rahulwagh/k8s-jenkins-aws'
}
```

- Jenkins stage-2 : Gradle compilation and build

```
stage('Gradle Build') {
    sh './gradlew build'
}
```

- Jenkins stage-3 : Create Docker Container and push to Docker Hub. After successful compilation and build let's create a Docker image and push to the docker hub.

```
stage("Docker build"){
    sh 'docker version'
    sh 'docker build -t jhooq-docker-demo .'
    sh 'docker image list'
    sh 'docker tag jhooq-docker-demo
    rahulwagh17/jhooq-docker-demo:jhooq-docker-demo'
}
```

```

stage("Push Image to Docker Hub"){
    sh 'docker push rahulwagh17/jhooq-docker-demo:jhooq-docker-demo'
}

```

Jenkins stage-4 : Kubernetes deployment

- Finally, do the Kubernetes deployment

```

stage("kubernetes deployment"){
    sh 'kubectl apply -f k8s-spring-boot-deployment.yml'
}

```

- Here is the complete final script for Jenkins pipeline

```

node {

    stage("Git Clone"){

        git credentialsId: 'GIT_HUB_CREDENTIALS', url:
'https://github.com/rahulwagh/k8s-jenkins-aws'
    }

    stage('Gradle Build') {

        sh './gradlew build'

    }

    stage("Docker build"){
        sh 'docker version'
        sh 'docker build -t jhooq-docker-demo .'
        sh 'docker image list'
        sh 'docker tag jhooq-docker-demo
rahulwagh17/jhooq-docker-demo:jhooq-docker-demo'
    }

    withCredentials([string(credentialsId: 'DOCKER_HUB_PASSWORD', variable:
'PASSWORD')]) {
        sh 'docker login -u rahulwagh17 -p $PASSWORD'
    }

    stage("Push Image to Docker Hub"){
        sh 'docker push rahulwagh17/jhooq-docker-demo:jhooq-docker-demo'
    }
}

```

Dashboard > All >

Enter an item name

>> Required field

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

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.

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

Windows Type here to search O 28°C Light rain ENG 2:43 PM 11/15/2021

Dashboard > Team12 >

General Build Triggers **Advanced Project Options** Pipeline

Advanced Project Options

Pipeline

Definition

Pipeline script

```
1+ node {  
2  stage("Git Clone"){  
3+     git credentialsId: 'GIT_HUB_CREDENTIALS', url: 'https://github.com/LaxmiNarayananK/k8s-jenkins-aws'  
4  }  
5+     stage('Gradle Build') {  
6      sh './gradlew build'  
7  }  
8+     stage("Docker build"){  
9       sh 'docker version'  
10      sh 'docker build -t jhoog-docker-demo .'  
11      sh 'docker image list'  
12  }  
13 }  
14+ stage("Docker build"){  
15   sh 'docker version'  
16   sh 'docker build -t jhoog-docker-demo .'  
17   sh 'docker image list'  
18 }
```

Use Groovy Sandbox

Pipeline Syntax

Save **Apply**

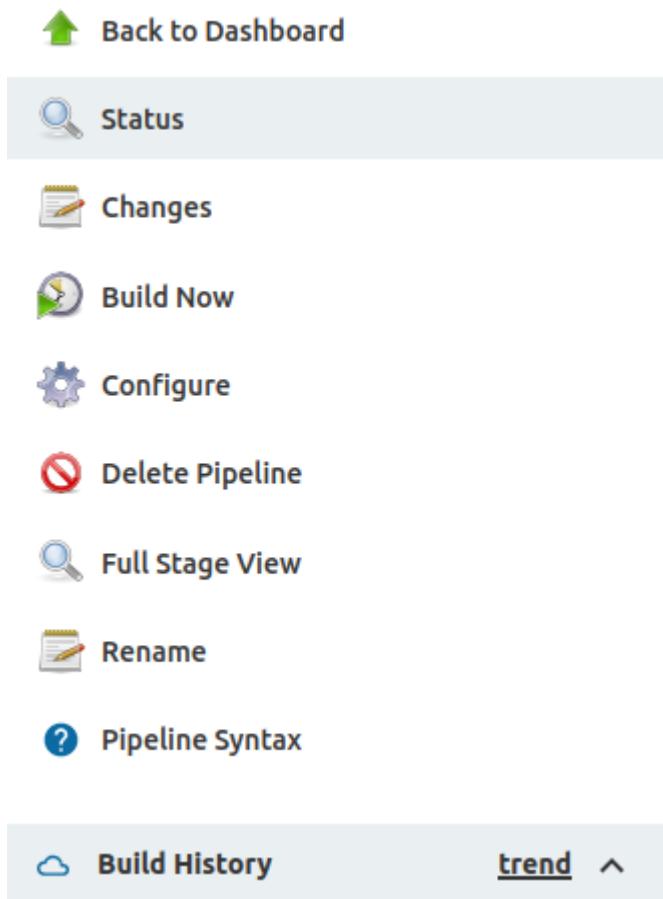
Windows Type here to search O 28°C Light rain ENG 2:48 PM 11/15/2021

Build, deploy and test CI/CD pipeline

Create new Pipeline: Goto Jenkins Dashboard or Jenkins home page click on New Item.

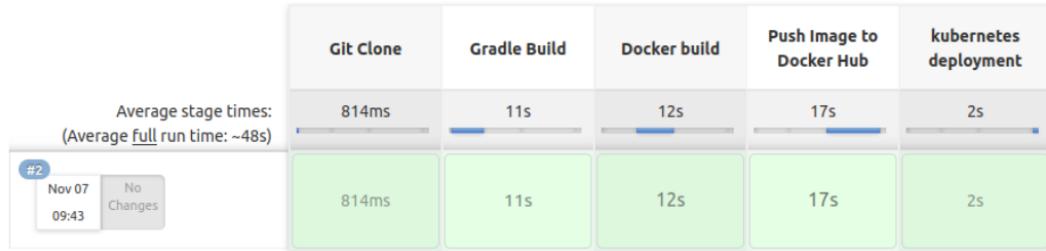
Pipeline Name: Now enter Jenkins pipeline name and select Pipeline

Add pipeline script: Goto -> Configure and then pipeline section. Copy the Jenkins script from Step 12 and paste it there. Build and Run Pipeline: Now goto pipeline and click on build now. Verify the build status:





Stage View



Permalinks

- Last build (#2), 5 min 50 sec ago
- Last stable build (#2), 5 min 50 sec ago
- Last successful build (#2), 5 min 50 sec ago
- Last completed build (#2), 5 min 50 sec ago

Verify using kubectl commands

You can also verify the Kubernetes deployment and service with `kubectl` command .e.g `kubectl get deployments, kubectl get service`

```
jenkins@ip-172-31-42-133:~$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
jhooq-springboot   2/3       3           2          33m
```

```
jenkins@ip-172-31-42-133:~$ kubectl get service
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
jhooq-springboot   LoadBalancer   10.100.230.63   a71c1183d4de24f959a0bbe37b035561-2067265724.eu-central-1.elb.amazonaws.com   80:30965/TCP   35m
kubernetes     ClusterIP   10.100.0.1    <none>        443/TCP   95m
```

You can access the rest end point from browser using the EXTERNAL-IP address



Jenkins Report - Google Docs | http://tateofficial/OctNewsletter | How to setup kubernetes jenkins pipeline | Billing Management Console | Subnets | VPC Management Console | Jenkins

Spinbot.com - Article... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Jenkins

aws Services ▾ Search for services, features, blogs, docs, and more [Alt+S] jaganmohan reddy q Global Support

Billing & Cost Management Dashboard

Spend Summary **Cost Explorer** **Month-to-Date Spend by Service** **Bill Details**

Welcome to the AWS Billing & Cost Management console. Your last month, month-to-date, and month-end forecasted costs appear below.

Current month-to-date balance for November 2021, the exchange rate for the Payment Currency is estimated.

6.97 USD which converts to **524.05 INR** at today's exchange rate of 75.19

\$12
\$9
\$6
\$3
\$0

Last Month (October 2021) Month-to-Date (November 2021) Forecast (November 2021)

\$0 **\$6.97** **\$8.94**

The chart below shows the proportion of costs spent for each service you use.

\$6.97

Service	Cost
Elastic Container Service for Kubernetes	\$3.67
Elastic Compute Cloud	\$2.24
CloudWatch	\$0.00
Data Transfer	\$0.00
Other Services	\$0.00
Tax	\$1.06
Total	\$6.97

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences