**Jenkins server Installation**

**Prerequisites**

1. EC2 Linux 7.x Instance
2. Java v11

**Install Java**

We will be using open java for our demo, Get latest version from <http://openjdk.java.net/install/>

yum install java-11\*

**Confirm Java Version**

Lets install java and set the java home

java -version

find / -name java-11\* | head -n 4

/etc/alternatives/java-11-amazon-corretto

/etc/alternatives/java-11

/usr/lib/jvm/java-11-amazon-corretto.x86\_64

/usr/lib/jvm/java-11-amazon-corretto

vi .bash\_profile

JAVA\_HOME=/usr/lib/jvm/java-11-amazon-corretto.x86\_64

export JAVA\_HOME

PATH=$PATH:$JAVA\_HOME

# To set it permanently update your .bash\_profile

source ~/.bash\_profile

*The output should be something like this,*

[root@~]# java -version

openjdk version "11.0.17" 2022-10-18 LTS

OpenJDK Runtime Environment Corretto-11.0.17.8.1 (build 11.0.17+8-LTS)

OpenJDK 64-Bit Server VM Corretto-11.0.17.8.1 (build 11.0.17+8-LTS, mixed mode)

**Install Jenkins**

You can install jenkins using the rpm or by setting up the repo. We will setup the repo so that we can update it easily in future. Get latest version of jenkins from <https://pkg.jenkins.io/redhat-stable/>

yum -y install wget

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

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

yum -y install jenkins

**Start Jenkins**

# Start jenkins service

systemctl start jenkins

# Setup Jenkins to start at boot,

systemctl enable jenkins

**Accessing Jenkins**

By default jenkins runs at port 8080, You can access jenkins at

http://YOUR-SERVER-PUBLIC-IP:8080

**Configure Jenkins**

* The default Username is admin
* Grab the default password
  + Password Location:/var/lib/jenkins/secrets/initialAdminPassword
* Skip Plugin Installation; *We can do it later*
* Change admin password
  + Admin > Configure > Password
* Configure java path
  + Manage Jenkins > Global Tool Configuration > JDK
* Create another admin user id

**Test Jenkins Jobs**

1. Create “new item”
2. Enter an item name – My-First-Project
   * Chose Freestyle project
3. Under Build section Execute shell : echo "Welcome to Jenkins Demo"
4. Save your job
5. Build job
6. Check "console output"

## Install Docker

```sh

yum install docker

systemctl start docker

systemctl enable docker

```

## provide permissions to jenkins user in jenkins server to access docker

```sh

sudo groupadd docker

sudo usermod -aG docker jenkins

sudo chmod 777 /var/run/docker.sock

```

## Add Jenkins user into sudoers file to get sudo access

```sh

vi /etc/sudoers

jenkins ALL=(ALL) NOPASSWD: ALL

```

### Jenkins server setup with Helm to deploy into Kubernetes cluster

## Download and Install helm

curl -fsSL -o get\_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3

chmod 700 get\_helm.sh

./get\_helm.sh

## Test with helm command

helm version

helm list

## Copy config file from EKS Management host to Jenkins home directory

mkdir /var/lib/jenkins/.kube

copy config file under .kube directory with jenkins ownership

# Install and Configure Maven & git in Jenkins server

## Install Maven

yum install wget

wget https://mirror.lyrahosting.com/apache/maven/maven-3/3.8.1/binaries/apache-maven-3.8.1-bin.tar.gz

tar -xvzf apache-maven-3.8.1-bin.tar.gz

export M2\_HOME=/opt/apache-maven-3.8.1

export M2=$M2\_HOME/bin

PATH=$PATH:$M2

# To set it permanently update your .bash\_profile

source ~/.bash\_profile

Validate Maven

mvn version

## Install git

yum install git

## Assign shell to jenkins user

vi /etc/passwd

change shell from /bin/false to /bin/bash

**Setup Kubernetes on Amazon EKS**

You can follow same procedure in the official AWS document [Getting started with Amazon EKS – eksctl](https://docs.aws.amazon.com/eks/latest/userguide/getting-started-eksctl.html)

**Pre-requisites:**

* an EC2 Instance (Kubernetes Management Host)

1. Install and setup kubectl on Management host a. Download kubectl version 1.19.6 b. Grant execution permissions to kubectl executable  
   c. Move kubectl onto /usr/local/bin  
   d. Test that your kubectl installation was successful
2. curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl
3. chmod +x ./kubectl
4. mv ./kubectl /usr/local/bin

kubectl version --short --client

1. Install and setup eksctl on Management Host  
   a. Download and extract the latest release  
   b. Move the extracted binary to /usr/local/bin  
   c. Test that your eksclt installation was successful
2. curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp
3. sudo mv /tmp/eksctl /usr/local/bin

eksctl version

1. Create an IAM Role and attache it to EC2 instance Management Host  
   Note: create IAM user with programmatic access if your bootstrap system is outside of AWS  
   IAM user should have access to  
   IAM  
   EC2  
   VPC  
   CloudFormation
2. Create EKS cluster and nodes from EC2 Management Host
3. eksctl create cluster --name cluster-name \
4. --region region-name \
5. --node-type instance-type \
6. --nodes-min 2 \
7. --nodes-max 2 \
8. --zones <AZ-1>,<AZ-2>
9. example:
10. eksctl create cluster --name cloudfreak-cluster \
11. --region ap-south-1 \

--node-type t2.medium \

1. To delete the EKS clsuter

eksctl delete cluster cloudfreak-cluster --region ap-south-1

1. Validate your cluster using by creating by checking nodes and by creating a pod
2. kubectl get nodes

kubectl run pod tomcat --image=tomcat

# Add stable to helm

helm repo add stable https://charts.helm.sh/stable

helm repo search <chartname>

helm install repo stable/<chartname> <releasename>

helm pull <chartname>

helm package <chartname>

helm uninstall RELEASE\_NAME