***Installation Guide***

**Jenkins Installation on AWS EC2**

Prerequisites: 8080 port will be opened for inbound traffic.

JAVA Installation

1. yum install java-1.8\*
2. java -version

find /usr/lib/jvm/java-1.8\* | head -n 3

1. set the JAVA home path.

Vi .bash\_profile

JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86\_64

M2\_HOME=/opt/maven

M2=/opt/maven/bin

PATH=$PATH:$HOME/bin:$JAVA\_HOME:$M2:$M2\_HOME

export PATH

Jenkins Installation

1. yum -y install wget

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.key

yum -y install Jenkins

1. Start Jenkins –

systemctl status jenkins

Systemctl start jenkins

**Maven Installation on Jenkins server**

Download maven packages https://maven.apache.org/download.cgi onto Jenkins server.

In this case, I am using /opt/maven as my installation directory

Link : https://maven.apache.org/download.cgi

# Creating maven directory under /opt

mkdir /opt/maven

cd /opt/maven

# downloading maven version 3.6.0

wget http://mirrors.estointernet.in/apache/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz

tar -xvzf apache-maven-3.6.1-bin.tar.gz

Setup M2\_HOME and M2 paths in .bash\_profile of the user and add these to the path variable

# User specific environment and startup programs

M2\_HOME=/opt/maven

M2=/opt/maven/bin

PATH=$PATH:$HOME/bin:$JAVA\_HOME:$M2:$M2\_HOME

export PATH

Git Installation

yum install git –y

**Tomcat Installation**

1. Download tomcat packages from https://tomcat.apache.org/download-80.cgi onto /opt on EC2 instance

# Create tomcat directory

cd /opt

wget http://mirrors.fibergrid.in/apache/tomcat/tomcat-8/v8.5.35/bin/apache-tomcat-8.5.35.tar.gz

tar -xvzf /opt/apache-tomcat-8.5.35.tar.gz

1. Tomcat application can be used at port 8080
2. search for context.xml

find / -name context.xml

1. Update users information in the tomcat-users.xml file goto tomcat home directory and Add below users to conf/tomcat-user.xml file

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

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

<user username="deployer" password="deployer" roles="manager-script"/>

<user username="tomcat" password="s3cret" roles="manager-gui"/>

1. Restart serivce and try to login to tomcat application from the browser

**Docker Installation**

1. yum install docker -y

docker –version

1. start docker

systemctl status docker

systemctl start docker

1. Create a user called dockeradmin

useradd dockeradmin

passwd dockeradmin

1. add a user to docker group to manage docker

usermod -aG docker dockeradmin

**Ansible Installation**

Prerequisite: Launch EC2

Install python and python-pip

yum install python

yum install python-pip

Install ansible using pip

pip install ansible

ansible --version

Create a user called ansadmin

useradd ansadmin

passwd ansadmin

add ansadmin to sudoers file or

echo "ansadmin ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

Log in as a ansadmin user on master and generate ssh key

sudo su - ansadmin

ssh-keygen

Copy keys onto all ansible managed hosts

ssh-copy-id ansadmin@<node ip address >

Ansible server used to create images and store on docker registry. Hence install docker, start docker services and add ansadmin to the docker group.

yum install docker

# start docker services

Systemctl status docker

Systemctl start docker

# add user to docker group

usermod -aG docker ansadmin

Create a directory /etc/ansible and create an inventory file called "hosts" add control node and managed hosts IP addresses to it.

Test Ansible setup ansible all -m ping