

# Installing Jenkins

## Prerequisites

Before continuing with this tutorial, make sure you have ubuntu machine created and logged in as a [user with sudo privileges](#).

To install Jenkins on your Ubuntu system, follow these steps:

### 1. Install Java.

Since Jenkins is a Java application, the first step is to install Java. Update the package index and install the Java 8 OpenJDK package with the following commands:

```
sudo apt update  
sudo apt install openjdk-8-jdk
```

### 2. Add the Jenkins Debian repository.

Import the GPG keys of the Jenkins repository using the following [wget](#) command:

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
```

The command above should output OK which means that the key has been successfully imported and packages from this repository will be considered trusted.

Next, add the Jenkins repository to the system with:

```
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >  
/etc/apt/sources.list.d/jenkins.list'
```

### 3. Install Jenkins.

Once the Jenkins repository is enabled, update the apt package list and install the latest version of Jenkins by typing:

```
sudo apt update  
sudo apt install jenkins
```

Jenkins service will automatically start after the installation process is complete. You can verify it by printing the service status:

```
systemctl status jenkins
```

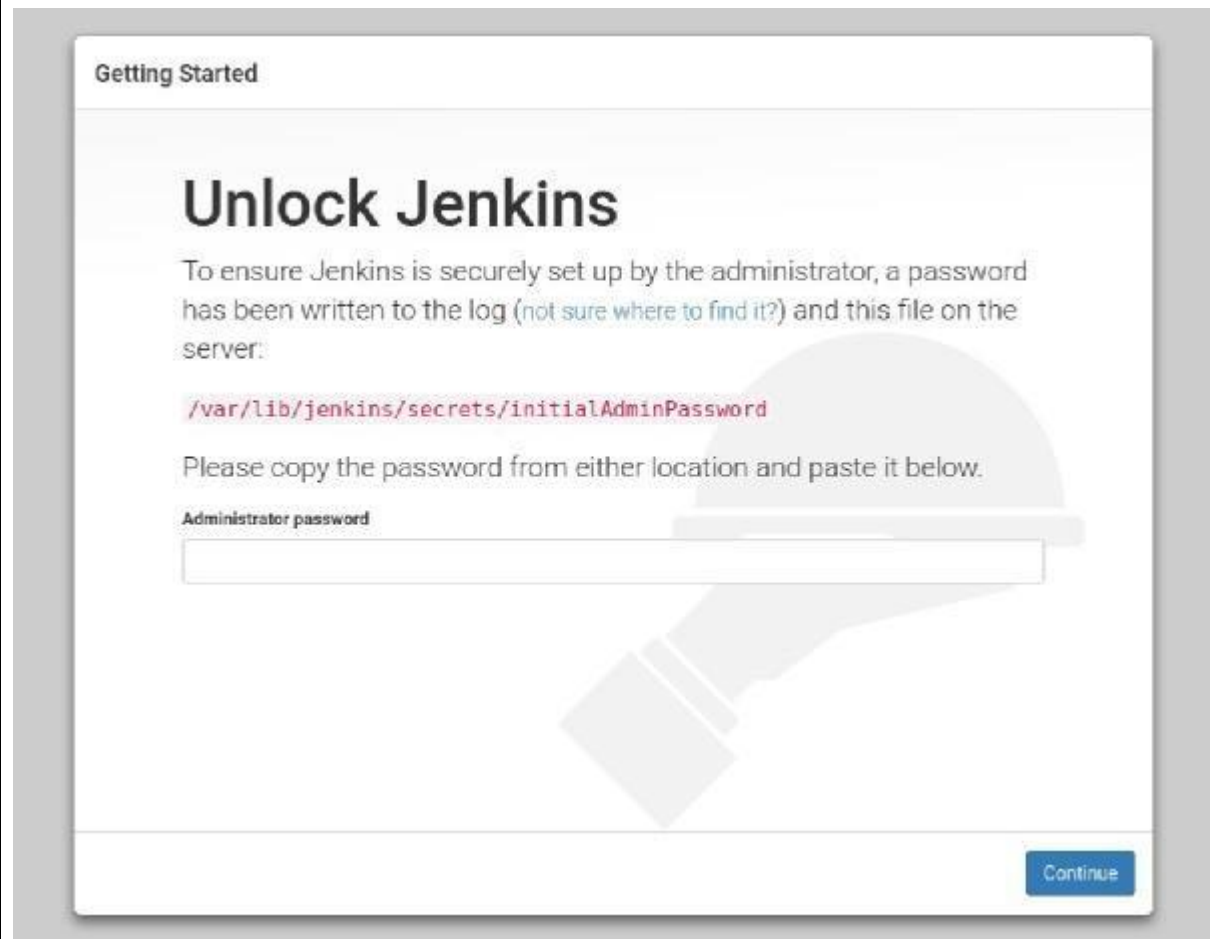
You should see something similar to this:

- jenkins.service - LSB: Start Jenkins at boot time

Loaded: loaded (/etc/init.d/jenkins; generated)  
Active: active (exited) since Wed 2019-07-06 13:08 PDT; 2min 16s ago  
Docs: man:systemd-sysv-generator(8)  
Tasks: 0 (limit: 2319)  
CGroup: /system.slice/jenkins.service

### Setting Up Jenkins

To set up your new Jenkins installation, open your browser, type your domain or IP address followed by port 8080 make sure you opened port 8080 in AWS security groups , `http://your_ip_or_domain:8080` and screen similar to the following will be displayed:



During the installation, the Jenkins installer creates an initial 32-character long alphanumeric password. Use the following command to print the password on your terminal:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
2115173b548f4e99a203ee99a8732a32
```

Copy the password from your terminal, paste it into the Administrator password field and click Continue and install Selected Plugins.

## Setting Up Docker in Jenkins Server

### 1. Install Docker

```
curl -fsSL get.docker.com | /bin/bash
```

### 2. Add Jenkins User to docker group

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

### 3. Restart Jenkins

```
sudo systemctl restart jenkins
```

## Setup Kubernetes Cluster

1. Create 2 ubuntu machines

### System Requirements

Master Machine : 4 GB RAM , 2 Core Processor

Worker Machines: 1 GB RAM , 1 Core Processor

2. Execute below commands in both master and slave machines.

=====COMMON FOR MASTER & SLAVES START =====

```
sudo apt-get update -y
```

```
sudo apt-get install -y apt-transport-https
```

```
sudo su -
```

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
```

```
cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
```

```
deb https://apt.kubernetes.io/ kubernetes-xenial main
```

```
EOF
```

```
apt-get update -y
```

```
swapoff -a
```

```
sed -i 's/ swap / s/^(.*)$/#\1/g' /etc/fstab
```

```
modprobe br_netfilter
```

```
sysctl -p
```

```
sudo sysctl net.bridge.bridge-nf-call-iptables=1
```

<b>Mithun Technologies</b> <b>+91-9980923226</b>	<b>Technology</b> <b>info@mithuntechnologies.com</b>	<b>Author</b> <b>Web site</b>	<b>Mithun Technologies</b> <b>http://mithuntechnologies.com</b>
<pre> apt install docker.io -y usermod -aG docker ubuntu  systemctl restart docker systemctl enable docker.service  apt-get install -y kubelet kubeadm kubectl kubernetes-cni  systemctl daemon-reload systemctl start kubelet systemctl enable kubelet.service  =====COMMON FOR MASTER &amp; SLAVES END=====  3. Execute below commands only in master machine.  =====In Master Node Start=====  # Execute below command as root user  kubeadm init  #exit root user &amp; execute as normal user  mkdir -p \$HOME/.kube sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config  kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=\$(kubectl version   base64   tr -d '\n')"</pre> <pre> kubectl get nodes kubectl get pods --all-namespaces # Get token  kubeadm token create --print-join-command  =====In Master Node End===== </pre>			

4. Execute kubeadm join token in worker nodes to join into cluster.

=====In Worker Nodes Start=====

copy kubeadm join token and execute in Worker Nodes to join to cluster

=====In Worker Nodes End=====

## Setup Jenkins Server to deploy applications into Kubernetes Cluster

We can deploy docker applications into Kubernetes cluster from Jenkins using below 2 approaches.

### 1) Using Kubernetes Continues Deploy Plugin

- Go to Jenkins → Manage Plugins → Available → Search for Kubernetes Continues Deploy → Select And Install.
- Add kube config information in Jenkins Credentials.  
Jenkins → Credentials → Add Credentials → Select Kind As Kubernetes Configuration ( Kubeconfig) → Select enter directly radio button → copy kubeconfig content from Kuberntes cluster
- Use KubernetesDeploy in pipeline script

Ex:

```
stage("Deploy To Kuberates Cluster"){  
    kubernetesDeploy(  
        configs: 'springBootMongo.yml',  
        kubeconfigId: 'KUBERNATES_CONFIG',  
        enableConfigSubstitution: true  
    )  
}
```

### 2) Install kubectl and add kubeconfig in Jenkins server

1. Install Kubectl in Jenkins Server

```
sudo apt-get update && sudo apt-get install -y apt-transport-https  
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
```

```
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a  
/etc/apt/sources.list.d/kubernetes.list  
sudo apt-get update  
sudo apt-get install -y kubectl
```

2. Switch to jenkins user

```
sudo -i -u jenkins
```

3. Create .kube folder in Jenkins home directory

```
cd ~  
mkdir .kube
```

4. Create config file and copy config file content from Kubernetes Cluster master machine and save the content.

```
vi .kube/config
```

5. We can use kubectl commands directly in pipe line script , kubectl commands will get executed in Kubernetes cluster directly.

```
stage("Deploy To Kuberates Cluster"){  
    sh "kubectl apply -f springBootMongo.yml"  
}
```