|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Mithun Technologies**  **+91-9980923226** | **Technology** [**info@mithuntechnologies.com**](mailto:info@mithuntechnologies.com) | **Author** | **Mithun Technologies** |  |
| **Web site** | [**http://mithuntechnologies.com**](http://mithuntechnologies.com/) |  |
|  | | | | | |
| 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 de[b http://pkg.jenkins.io/debian-stable](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 | | | | | |
|  |  | | | | | |
|  | **© 2019 all rights Reserved by Mithun Technologies** [**devopstrainingblr@gmail.com**](mailto:devopstrainingblr@gmail.com) | | | | | |

|  |  |  |
| --- | --- | --- |
| **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: | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mithun Technologies**  **+91-9980923226** | **Technology** [**info@mithuntechnologies.com**](mailto:info@mithuntechnologies.com) | **Author** | **Mithun Technologies** |  |
| **Web site** | [**http://mithuntechnologies.com**](http://mithuntechnologies.com/) |  |
| 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 1308 PDT; 2min 16s ago Docs: man:systemd-sysv-generator(8)  Tasks: 0 (limit: 2319)  CGroup: /system.slice/jenkins.service  **© 2019 all rights Reserved by Mithun Technologies** [**devopstrainingblr@gmail.com**](mailto:devopstrainingblr@gmail.com) | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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: | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mithun Technologies**  **+91-9980923226** | **Technology** [**info@mithuntechnologies.com**](mailto:info@mithuntechnologies.com) | **Author** | **Mithun Technologies** |  |
| **Web site** | [**http://mithuntechnologies.com**](http://mithuntechnologies.com/) |  |
| 1. Install Docker   curl -fsSL get.docker.com | /bin/bash   1. Add Jenkins User to docker group   **© 2019 all rights Reserved by Mithun Technologies** [**devopstrainingblr@gmail.com**](mailto:devopstrainingblr@gmail.com) | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 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** |  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mithun Technologies**  **+91-9980923226** | **Technology** [**info@mithuntechnologies.com**](mailto:info@mithuntechnologies.com) | **Author** | **Mithun Technologies** |  |
| **Web site** | [**http://mithuntechnologies.com**](http://mithuntechnologies.com/) |  |
| 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 Processer Worker Machines: 1 GB RAM , 1 Core Processer   1. 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 '/ swap / s/^\(.\*\)$/#\1/g' /etc/fstab  modprobe br\_netfilter sysctl -p  sudo sysctl net.bridge.bridge-nf-call-iptables=1  **© 2019 all rights Reserved by Mithun Technologies** [**devopstrainingblr@gmail.com**](mailto:devopstrainingblr@gmail.com) | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mithun Technologies**  **+91-9980923226** | **Technology** [**info@mithuntechnologies.com**](mailto:info@mithuntechnologies.com) | **Author** | **Mithun Technologies** |  |
| **Web site** | [**http://mithuntechnologies.com**](http://mithuntechnologies.com/) |  |
| 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 & 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 & 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')"  kubectl get nodes  kubectl get pods --all-namespaces # Get token  kubeadm token create --print-join-command  =========In Master Node End====================  **© 2019 all rights Reserved by Mithun Technologies** [**devopstrainingblr@gmail.com**](mailto:devopstrainingblr@gmail.com) | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mithun Technologies**  **+91-9980923226** | **Technology** [**info@mithuntechnologies.com**](mailto:info@mithuntechnologies.com) | **Author** | **Mithun Technologies** |  |
| **Web site** | [**http://mithuntechnologies.com**](http://mithuntechnologies.com/) |  |
| 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 Manage Credentials  Add Credentials  Select Kind As Kubernates Configuration ( Kubeconfig)  Select enter directly radio button  copy kubeconfig content from Kubenertes cluster(Get Kubernates Master’s CMD🡺 cat .kube/congif )   * + Use KubernetesDeploy in pipeline script Ex:   stage("Deploy To Kuberates Cluster"){ kubernetesDeploy(  configs: 'springBootMongo.yml', kubeconfigId: 'KUBERNATES\_CONFIG', enableConfigSubstitution: true  )  }   1. **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 -**  **© 2019 all rights Reserved by Mithun Technologies** [**devopstrainingblr@gmail.com**](mailto:devopstrainingblr@gmail.com) | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mithun Technologies**  **+91-9980923226** | **Technology** [**info@mithuntechnologies.com**](mailto:info@mithuntechnologies.com) | **Author** | **Mithun Technologies** |  |
| **Web site** | [**http://mithuntechnologies.com**](http://mithuntechnologies.com/) |  |
| **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**   1. Switch to jenkins user   **sudo -i -u jenkins**   1. Create .kube folder in Jenkins home directory   **cd ~**  **mkdir .kube**   1. 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"  }  **© 2019 all rights Reserved by Mithun Technologies** [**devopstrainingblr@gmail.com**](mailto:devopstrainingblr@gmail.com)  **Notes:**   |  | | --- | | **🡺Follow this version while installing below plugin ( Insatll manually I know**  **that)**   * **Install 🡺** Kubernetes Continuous Deploy V[2.3.0](http://15.207.107.199:8080/pluginManager/plugin/kubernetes-cd/thirdPartyLicenses) * **Insatll 🡺** Jackson 2 API V [2.11.0](http://15.207.107.199:8080/pluginManager/plugin/jackson2-api/thirdPartyLicenses) |  * **If you get below error : Make sure for all instances(Master & slave) Security Group for Inbound Traffic should be open for all ports (Security Group 🡺Add Inbound🡺 Select All traffic)**      * **Once Master & slave setup is done Follow below Instructions:**  1. Create job in Jenkins with pipeline 2. Clone github URL (<https://github.com/Alsamad-GH/spring-boot-mongo-docker.git>) in Pipeline script      |  | | --- | | stage("Git Clone"){  git credentialsId: 'Git\_Credentials', url: 'https://github.com/Alsamad-GH/spring-boot-mongo-docker.git'  } |  1. Install Maven in Jenkins  * Go to Manage Jenkins🡺Global Tool Config🡺install Maven  |  | | --- | | stage("Maven Clean Build"){  def mavenHome = tool name: "Maven-3.6.3", type: "maven"  def mavenCMD = "${mavenHome}/bin/mvn"  sh "${mavenCMD} clean package"  } |  1. Build Docker Image  |  | | --- | | stage("Build Docker Image"){  sh "docker build -t alsamad0786/spring-boot-mango ."  } |  1. Push Docker image  * First Create Secret for Password in Pipeline syntax  1. Click on Pipeline syntax 2. Select 🡺 withCredentials: Bind credentials to variables 3. Select Add 🡺 4. Kind🡺secret text 5. Add🡺Secret🡺ID (DOCKER\_HUB\_CREDENTIALS ) 🡺Description 6. Add 7. Variable🡺 DOCKER\_HUB\_CREDENTIALS (ID)      |  | | --- | | stage("Docker Push"){  withCredentials([string(credentialsId: DOCKER\_HUB\_CREDENTIALS, variable: 'DOCKER\_HUB\_CREDENTIALS')]) {  sh "docker login -u alsamad0786 -p ${DOCKER\_HUB\_CREDENTIALS}"  }  sh "docker push alsamad0786/spring-boot-mango"  }  alsamad0786 🡺 Docker hub User Name  ${DOCKER\_HUB\_CREDENTIALS} 🡺 Secret Password of Docker hub |      1. Now follow last step(check up) 🡺 **Setup Jenkins Server to deploy applications into Kubernetes Cluster**   Finally Page is Up | | | | | |
|  | | | | | |