

DevOps tools install commands on Ubuntu 20.04

❖ GIT Install

- Sudo su -
- apt update
- apt install git
- git --version
- which git
- git init
- ls -a

❖ GIT Uninstall

- apt remove git -y

❖ Jenkins Install

- sudo apt install openjdk-11-jdk -y (install updated java version)

use Jenkins documentation or use this commands

- sudo wget -O /usr/share/keyrings/jenkins-keyring.asc <https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key>
- echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] <https://pkg.jenkins.io/debian-stable/binary/> | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
- sudo apt-get update
- sudo apt-get install jenkins
- sudo systemctl enable jenkins
- sudo systemctl start jenkins
- sudo systemctl status Jenkins

then, copy public ip and type :8080

❖ Maven Install

- sudo apt install openjdk-11-jdk -y (install updated java version)
- sudo apt install maven
- mvn --version
- mvn archetype:generate
- apt install tree

❖ Docker Install

- apt update
- sudo apt install openjdk-11-jdk -y (install updated java version)
- apt install docker.io -y
- docker --version
- systemctl enable docker
- systemctl start docker
- systemctl status docker

❖ Docker Compose

- sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose
- sudo chmod +x /usr/local/bin/docker-compose
- docker-compose --version

❖ Terraform Install

- sudo su -
- apt update
- sudo apt install software-properties-common gnupg2 curl
- sudo install -o root -g root -m 644 hashicorp.gpg /etc/apt/trusted.gpg.d/
- sudo apt install terraform
- terraform --version

❖ Ansible Install

On Master

- sudo su –
- apt update
- apt install ansible
- cd /etc/ansible
- ls
- vi hosts (add private ip of agent)
- ssh-keygen
- 4 times enter
- Cat /home/ubuntu/.ssh/id_rsa.pub (copy ssh key)

On Agent

- Sudo su –
- Apt update
- Ls – a
- Cd .ssh
- Ls
- Vi authorized_key (paste here key)

Again, On Master

- Ansible -m ping all
- Ssh ip address of agent

❖ Kubernetes Install

Step1: On Master & worker node

- sudo su
- apt-get update
- apt-get install docker.io -y
- service docker restart
- curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
- echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list
- apt-get update
- apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y

Step2: On Master:

- kubeadm init --pod-network-cidr=192.168.0.0/16 (Copy the token and paste it into the worker node.)

Step3: On Master:

- exit
- mkdir -p \$HOME/.kube
- sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config
- sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config

Step4: On Master:

- kubectl apply -f <https://docs.projectcalico.org/manifests/calico.yaml>
- kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml>

Our Kubernetes installation and configuration are complete

❖ Kubernetes Ingress Installation with Manifests

- `git clone https://github.com/nginxinc/kubernetes-ingress.git --branch v3.3.1`
- `cd kubernetes-ingress/deployments`
- `kubectl apply -f common/ns-and-sa.yaml`
- `kubectl apply -f rbac/rbac.yaml`
- `kubectl apply -f rbac/ap-rbac.yaml`
- `kubectl apply -f rbac/apdos-rbac.yaml`
- `kubectl apply -f ../examples/shared-examples/default-server-secret/default-server-secret.yaml`
- `kubectl apply -f common/nginx-config.yaml`
- `kubectl apply -f common/ingress-class.yaml`
- `kubectl apply -f common/crds/k8s.nginx.org_virtualservers.yaml`
- `kubectl apply -f common/crds/k8s.nginx.org_virtualserverroutes.yaml`
- `kubectl apply -f common/crds/k8s.nginx.org_transportservers.yaml`
- `kubectl apply -f common/crds/k8s.nginx.org_policies.yaml`
- `kubectl apply -f common/crds/k8s.nginx.org_globalconfigurations.yaml`
- `kubectl apply -f common/crds/appprotect.f5.com_aplogconfs.yaml`
- `kubectl apply -f common/crds/appprotect.f5.com_appolicies.yaml`
- `kubectl apply -f common/crds/appprotect.f5.com_apusersigs.yaml`
- `kubectl apply -f common/crds/appprotectdos.f5.com_apdoslogconfs.yaml`
- `kubectl apply -f common/crds/appprotectdos.f5.com_apdospolicy.yaml`
- `kubectl apply -f common/crds/appprotectdos.f5.com_dosprotectedresources.yaml`
- `kubectl apply -f deployment/appprotect-dos-arb.yaml`
- `kubectl apply -f service/appprotect-dos-arb-svc.yaml`
- `kubectl apply -f deployment/nginx-ingress.yaml`
- `kubectl apply -f deployment/nginx-plus-ingress.yaml`
- `kubectl apply -f daemon-set/nginx-ingress.yaml`
- `kubectl apply -f daemon-set/nginx-plus-ingress.yaml`
- `kubectl get pods --namespace=nginx-ingress`
- `kubectl get ns`