**Installing Kubernetes Clustors with Kubeadm in Ubuntu 16.04 Xenial**

**What is Kubeadm?**

Kubeadm helps you bootstrap a minimum viable Kubernetes cluster that conforms to best practices. Kubeadm is a tool built to provide kubeadm init and kubeadm join as best-practice "fast paths" for creating Kubernetes clusters.

**Goal**

* To Install a single master Kubernetes cluster
* To Install a high availability master Kubernetes cluster
* To Install a Pod network on the cluster so that your Pods can talk to each other.

kubeadm’s simplicity means it can serve a wide range of use cases:

* New users can start with kubeadm to try Kubernetes out for the first time.
* Users familiar with Kubernetes can spin up clusters with kubeadm and test their applications.
* Larger projects can include kubeadm as a building block in a more complex system that can also include other installer tools.

Pre-requisite

* One or more machines running a deb/rpm-compatible OS, for example Ubuntu or CentOS
* 2 GB or more of RAM per machine. Any less leaves little room for your apps.
* 2 CPUs or more on the master
* Full network connectivity among all machines in the cluster. A public or private network is fine

As part of the installation, every node (master and minions) needs:

1. Docker
2. Kubelet
3. Kubeadm
4. Kubectl
5. CNI

**Installing and Configuring single master Kubernetes cluster**

**$** apt-get update && apt-get install -y apt-transport-https

**$** 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

$ apt-get install -y docker.io kubeadm kubectl kubelet kubernetes-cni

$ apt-get install -y docker.io kubeadm kubectl kubelet kubernetes-cni

**$** kubeadm init

**Your Kubernetes master has initialized successfully!**

**To start using your cluster, you need to run the following as a regular user:**

$ mkdir -p $HOME/.kube

$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

$ sudo chown $(id -u):$(id -g) $HOME/.kube/config

**Verify Kubernetes Clustors**

$ kubectl get nodes

$ kubectl get pods --all-namespaces

**Deploy a pod network to the cluster**

You should now deploy a pod network to the cluster. Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at: https://kubernetes.io/docs/concepts/cluster-administration/addons/ Weave Net provides networking and network policy, will carry on working on both sides of a network partition, and does not require an external database.

$ 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

$ kubectl get nodes

**Installing and Adding Nodes(minions) to Kubernetes master cluster**

You can now join any number of machines by running the following on each node as root:

**$** apt-get update && apt-get install -y apt-transport-https

$ 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

$ apt-get install -y docker.io kubeadm kubectl kubelet kubernetes-cni

Note: before joining the nodes please verify the security group port opened or not in the AWS i.e 6443

$ kubeadm join 172.31.31.106:6443 --token pdn6in.r0dzhpx1ucrs69au --discovery-token-ca-cert-hash sha256:a9385951e659a3c67f55ccfbdc1169b1f660ba09aaf8cc6d5cc96d71b71900d2

If you forgot kubeadm join key please use below command to get kubeadm join

Key

$kubeadm token create --print-join-command

# **How To Monitor Linux Servers With Prometheus, Node Exporter and Grafana**

**STEP I:** First, download and unpack the current stable version of Prometheus into your home directory. You can find the latest binaries along with their checksums on the [Prometheus download page](https://prometheus.io/download/).

**$ cd ~**

**$ curl -LO** [**https://github.com/prometheus/prometheus/releases/download/v2.0.0/prometheus-2.0.0.linux-amd64.tar.gz**](https://github.com/prometheus/prometheus/releases/download/v2.0.0/prometheus-2.0.0.linux-amd64.tar.gz)

**Next, use the sha256sum command to generate a checksum of the downloaded file:**

$ sha256sum prometheus-2.0.0.linux-amd64.tar.gz

Now, unpack the downloaded archive.

**$** tar -xvzf prometheus-2.0.0.linux-amd64.tar.g z

$ mv prometheus-2.0.0.linux-amd64 prometheus-files

**STEP2: Create a Prometheus user required directories and make Prometheus user as the owner of those directories**

$ useradd --no-create-home --shell /bin/false Prometheus

$ mkdir /etc/Prometheus

$ mkdir /var/lib/Prometheus

$ chown prometheus:prometheus /etc/prometheus/

$ chown prometheus:prometheus /var/lib/prometheus/

**STEP III: Copy Prometheus and promtool binary from Prometheus-files folder to /usr/local/bin and change the ownership to Prometheus user.**

$ cp prometheus-files/prometheus /usr/local/bin/

$ cp prometheus-files/promtool /usr/local/bin/

$ chown prometheus:prometheus /usr/local/bin/Prometheus

$ chown prometheus:prometheus /usr/local/bin/promtool

**STEP 4: move the consoles and console\_libraries directories from Prometheus-files to /etc/Prometheus folder and change the ownership to Prometheus user.**

$ cp -r prometheus-files/consoles/ /etc/prometheus/

$ cp -r prometheus-files/console\_libraries/ /etc/prometheus/

$ chown -R prometheus:prometheus /etc/prometheus/consoles/

$ chown -R prometheus:prometheus /etc/prometheus/console\_libraries/

**STEP5: Setup Prometheus configuration**

**All the Prometheus configuration should be present in /etc/Prometheus/Prometheus.yml file**

**Create the Prometheus.yml file and add below contexts**

Vi /etc/Prometheus/Prometheus.yml

**Note: We should copy the original Prometheus.yml file data here and the ownership permission as shown below**

$ chown prometheus:prometheus /etc/prometheus/prometheus.yml

STEP6: Add Prometheus to Startup service

Create a Prometheus service file and add below context

**$ vi /etc/system/system/Prometheus.service**

[Unit]

Description=Prometheus

Wants=network-online.target

After=network-online.target

[Service]

User=prometheus

Group=prometheus

Tyepe=simple

ExecStart=/usr/local/bin/prometheus \

--config.file /etc/prometheus/prometheus.yml \

--storage.tsdb.path /var/lib/prometheus/ \

--web.console.templates=/etc/prometheus/consoles \

--web.console.libraries=/etc/prometheus/console\_libraries

[Install]

WantedBy=multi-user.target

**Then reload the systemd service to register the Prometheus service and start and enable the Prometheus service.**

$ systemctl daemon-relaod

$ systemctl start Prometheus

$ systemctl status Prometheus

$ systemctl enable Prometheus

**ACCESS Prometheus Web UI**

Now you will be able to access the Prometheus UI on 9090 port of the Prometheus server.

You should mention the ip address of Prometheus service in Prometheus.yml under

**static\_configs:**

**- targets: ['localhost:9090'] i.e instead of localhost mention Prometheus server ip ,**

**Then restart the Prometheus. After restart you can browse as shown below formart**

http://13.127.27.156:9090/graph

**INSTALL NODE EXPORTER**

**Node exporter install in workers machine**

**STEPI: Download the node exporter package using wget ,untar it and move the node export binary to /usr/local/bin**

**$** curl -LO https://github.com/prometheus/node\_exporter/releases/download/v0.18.1/node\_exporter-0.18.1.linux-amd64.tar.gz

$ tar -xvzf node\_exporter-0.18.1.linux-amd64.tar.gz

$ mv node\_exporter-0.18.1.linux-amd64/node\_exporter /usr/local/bin/

**STEP2: create a user node\_exporter and add node exporter as startup service**

$ useradd -rs /bin/false node\_exporter

**vi /etc/systemd/system/node\_exporter.service**

[Unit]

Description=Node Exporter

After=network.target

[Service]

User=node\_exporter

Group=node\_exporter

Type=simple

ExecStart=/usr/local/bin/node\_exporter

[Install]

WantedBy=multi-user.target

**Then reload the node exporter service to register the node exporter service and start and enable the node exporter service.**

$ systemctl daemon-reload

$ systemctl start node\_exporter.service

$ systemctl enable node\_exporter.service

$ systemctl status node\_exporter.service

**Configure the server as target on Prometheus server**

Now that’s we have the node exporter up and running on the server, we have to add this server a target on the Prometheus server configuration

Note: This configuration should be done on the Prometheus server

STEP1: login to the promethus server and open the Prometheus.yml

Vi /etc/Prometheus/Prometheus.yml

Step2 Under the scrape config section add the node exporter target as shown below .Job Name can be your hostname or IP for identification purpose

**scrape\_configs:**

**- job\_name: node**

**static\_configs:**

**- targets: ['13.127.27.156:9100']**

**$ systemctl restart Prometheus or systemctl restart prometheus.service**

**Now , if you check the target in Prometheus web UI (**[**http://PrometheusIp:9090/target**](http://PrometheusIp:9090/target)**). You wiil be able to see status.**

**Install Grafana**

**Follow the below command the for Grafana installation**

$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add –

$ sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"

$ sudo apt update

$ apt-cache policy grafana

$ sudo apt install grafana

$ sudo systemctl start grafana-server

$ sudo systemctl status grafana-server

$ sudo systemctl enable grafana-server

Browsing Grafana

[**http://13.127.27.156:3000/login**](http://13.127.27.156:3000/login)

for the first we need give user name and pwd

User: admin

Pwd: admin

Then we need to be change

**Add a dashboard**

Grafana has a long list of prebuild dashboard here:

[**Http://grafana.com/dashboard**](Http://grafana.com/dashboard)

Click on Home button

Select Addsource

Select Prometheus

Name : your choice

URL: Prometheus server Url

Clicke on Save & test Button Note : It should be success

In the left hand menu,

choose Dashboard,

Manage,

Import

In the Grafana.com dashboard input

, add the dashboard ISD we want to use : 1860 and click load

On the next screen select a name for your dashboard and select promethues as the darassourece for it and click Import