

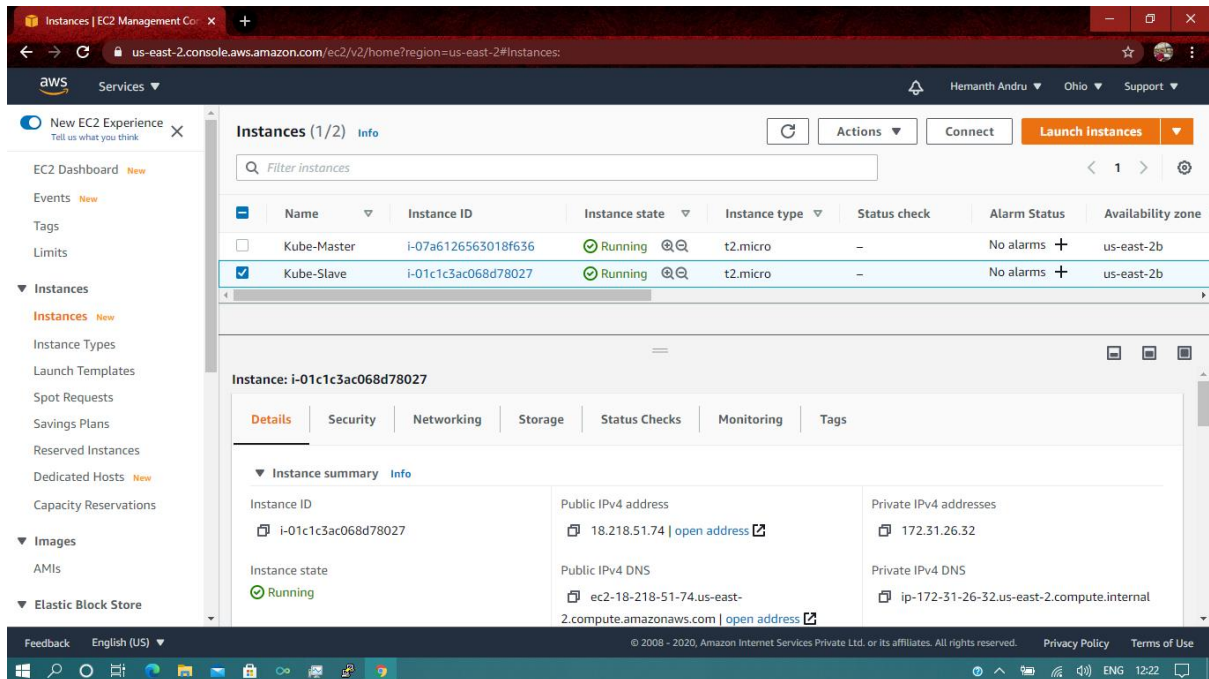


SCHOOL OF COMPUTER SCIENCE
LOVELY PROFESSIONAL UNIVERSITY
PHAGWARA, PUNJAB(INDIA)-144402

KUBERNETES

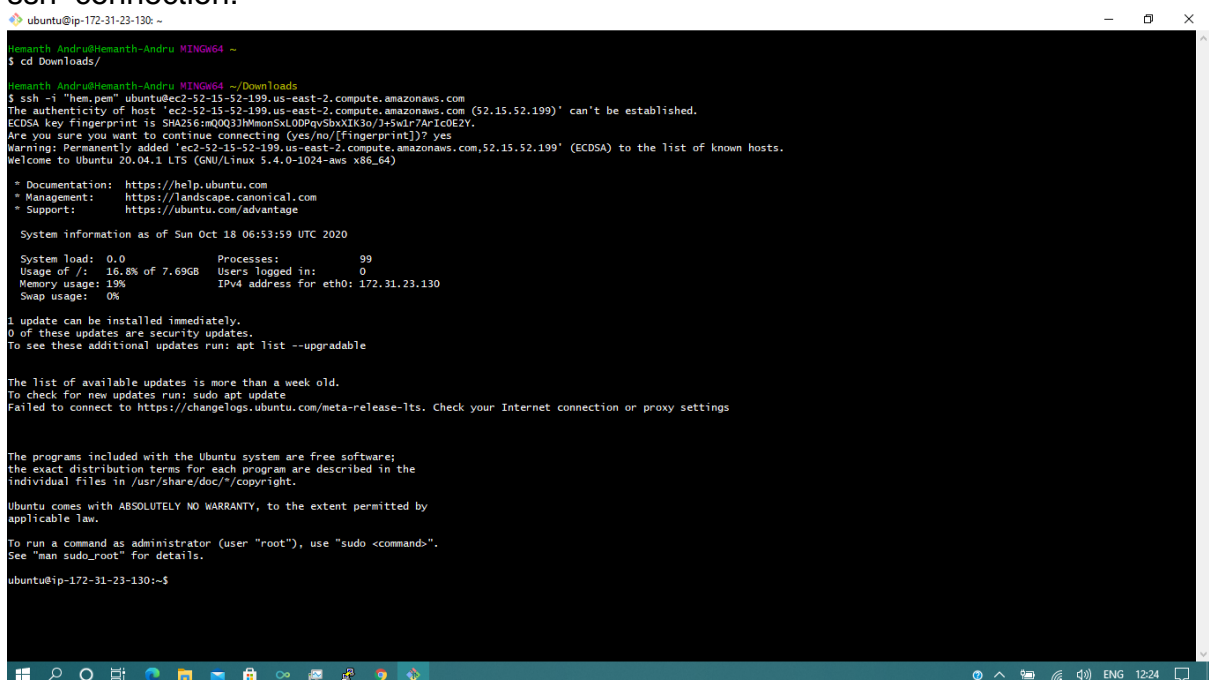
Practical :- 1 Kubernetes installation.

Step 1:- Create a two aws ec2 instance in aws using ubuntu operating system. And then name them as Kube-Master and kube-Slave. As shown in the figure below.

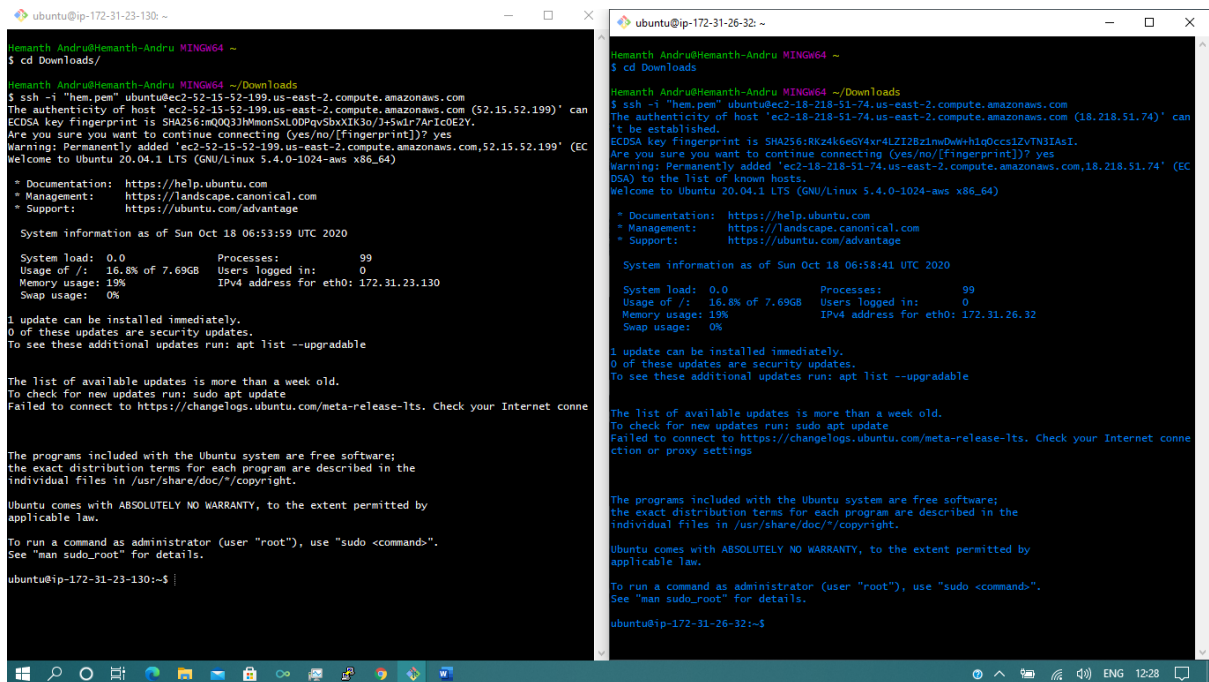


Note :- Here kube-Master is my master and kube-slave is my slave.

Step 2 :- Now start two ec2 instances and open gitbash .And then connect it with ssh connection.



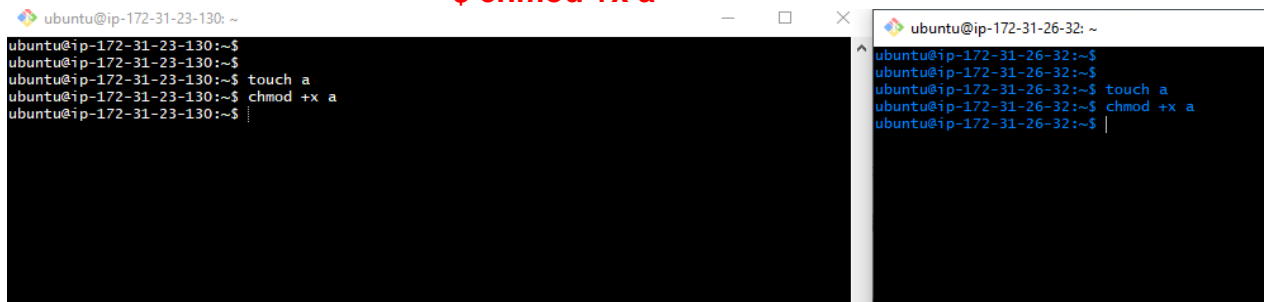
Note :- For master my gitbash in white colour and for slave my gitbash is in blue colour.



```
ubuntu@ip-172-31-23-130: ~  
Hemanth Andru@Hemanth-Andru MINGW64 ~  
$ cd Downloads/  
Hemanth Andru@Hemanth-Andru MINGW64 ~/Downloads  
$ ssh -i "hen.pem" ubuntu@ec2-52-15-52-199.us-east-2.compute.amazonaws.com  
The authenticity of host 'ec2-52-15-52-199.us-east-2.compute.amazonaws.com (52.15.52.199)' can  
be established.  
ECDSA key fingerprint is SHA256:M00331MwonSxL00PqySbxIX3o/145uL7Ar1cdE2y.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-52-15-52-199.us-east-2.compute.amazonaws.com,52.15.52.199' (EC  
DSA) to the list of known hosts.  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1024-aws x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:        https://ubuntu.com/advantage  
  
System information as of Sun Oct 18 06:53:59 UTC 2020  
  
System load: 0.0      Processes:            99  
Usage of /:  16.6% of 7.69GB   Users logged in:       0  
Memory usage: 19%      IPv4 address for eth0: 172.31.23.130  
Swap usage:  0%  
  
1 update can be installed immediately.  
0 of these updates are security updates.  
To see these additional updates run: apt list --upgradable  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connec  
tion or proxy settings  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-23-130:~$ |  
ubuntu@ip-172-31-26-32: ~  
Hemanth Andru@Hemanth-Andru MINGW64 ~  
$ cd Downloads/  
Hemanth Andru@Hemanth-Andru MINGW64 ~/Downloads  
$ ssh -i "hen.pem" ubuntu@ec2-18-218-51-74.us-east-2.compute.amazonaws.com  
The authenticity of host 'ec2-18-218-51-74.us-east-2.compute.amazonaws.com (18.218.51.74)' can  
be established.  
ECDSA key fingerprint is SHA256:RKd4L6eGY4vr4LZT2B2mDwH-H1q0ccs12VTN3IA5I.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-18-218-51-74.us-east-2.compute.amazonaws.com,18.218.51.74' (EC  
DSA) to the list of known hosts.  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1024-aws x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:        https://ubuntu.com/advantage  
  
System information as of Sun Oct 18 06:58:41 UTC 2020  
  
System load: 0.0      Processes:            99  
Usage of /:  16.6% of 7.69GB   Users logged in:       0  
Memory usage: 19%      IPv4 address for eth0: 172.31.26.32  
Swap usage:  0%  
  
1 update can be installed immediately.  
0 of these updates are security updates.  
To see these additional updates run: apt list --upgradable  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connec  
tion or proxy settings  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-26-32:~$
```

Step 3 :- Now create file with any name and give execute permission to it in both master and slave.

\$ sudo su
\$ touch a
\$ chmod +x a



```
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ touch a  
ubuntu@ip-172-31-23-130:~$ chmod +x a  
ubuntu@ip-172-31-23-130:~$ |  
ubuntu@ip-172-31-26-32:~$  
ubuntu@ip-172-31-26-32:~$  
ubuntu@ip-172-31-26-32:~$ touch a  
ubuntu@ip-172-31-26-32:~$ chmod +x a  
ubuntu@ip-172-31-26-32:~$ |
```

Now open the file using the command nano .

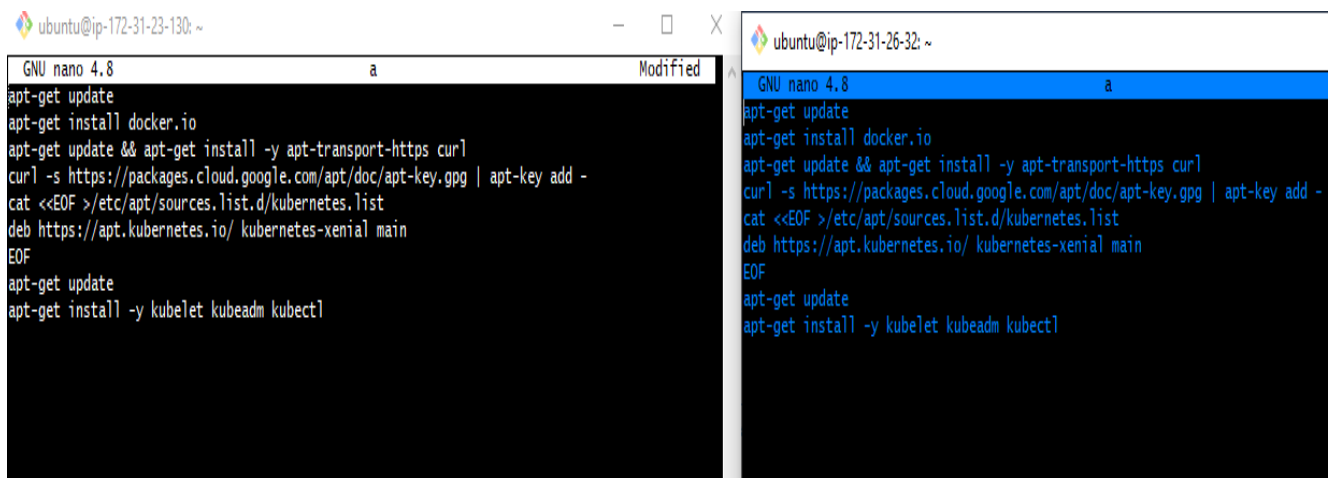
\$ nano a



```
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ touch a  
ubuntu@ip-172-31-23-130:~$ chmod +x a  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ nano a  
ubuntu@ip-172-31-26-32:~$  
ubuntu@ip-172-31-26-32:~$  
ubuntu@ip-172-31-26-32:~$ touch a  
ubuntu@ip-172-31-26-32:~$ chmod +x a  
ubuntu@ip-172-31-26-32:~$  
ubuntu@ip-172-31-26-32:~$ nano a
```

Step 4 :- After opening the files now copy the commands and paste in that file and save it and close and then run the file.

```
$ apt-get update
$ apt-get install docker.io
$ apt-get update && apt-get install -y apt-transport-https curl
$ 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 kubelet kubeadm kubectl
```



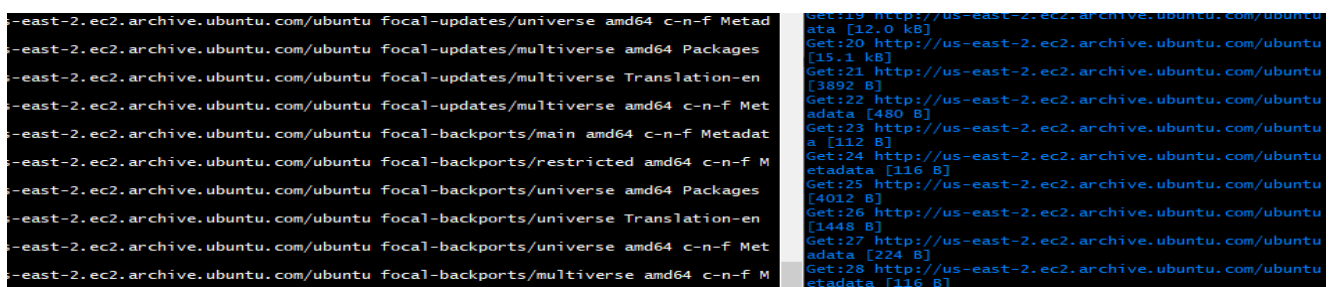
The image shows two terminal windows side-by-side. Both windows have a title bar that says 'ubuntu@ip-172-31-23-130: ~' and 'GNU nano 4.8'. The left window shows the commands being pasted into a file named 'a'. The right window shows the same commands being pasted into a file named 'a'.

Now save the file using `ctrl+s` and exit using `ctrl+x`. And then run it with following command.

`$. /a`



The image shows two terminal windows side-by-side. Both windows have a title bar that says 'ubuntu@ip-172-31-23-130: ~' and 'ubuntu@ip-172-31-26-32: ~'. The left window shows the execution of the script 'a' with the following commands: `touch a`, `chmod +x a`, `nano a`, and `./a`. The right window shows the execution of the script 'a' with the following commands: `touch a`, `chmod +x a`, `nano a`, and `./a`.



The image shows a terminal window with the output of the `apt-get update` command. The output shows the progress of the update, including the download of the package list and the calculation of the package sizes. The output is as follows:

```
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metad
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages
Get:21 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Met
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadat
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f M
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Met
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f M
```

```
root@ip-172-31-23-130:/home/ubuntu#
Get:3 https://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 socat amd64 1.7.3-3-2 [23 kB]
Get:4 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 cri-tools amd64 1.13.0-0-1 [8775 kB]
Get:5 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubernetescni amd64 0.8.7-00 [25.0 MB]
Get:6 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubelet amd64 1.19.3-00 [18.2 MB]
Get:7 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubect1 amd64 1.19.3-00 [8350 kB]
Get:8 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubeadm amd64 1.19.3-00 [7758 kB]
Fetched 68.5 MB in 19s (3552 kB/s)
Selecting previously unselected package conract.
(Reading database ... 60036 files and directories currently installed.)
Preparing to unpack .../0-conract_1x3al.4.5-2.amd64.deb ...
Unpacking conract (1:1.4.5-2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.13.0-0-1.amd64.deb ...
Unpacking cri-tools (1.13.0-0-1) ...
Selecting previously unselected package etables.
Preparing to unpack .../2-etables_2.0.11-3bu1d1.amd64.deb ...
Unpacking etables (2.0.11-3bu1d1) ...
Selecting previously unselected package kubernetescni.
Preparing to unpack .../3-kubernetescni_0.8.7-00.amd64.deb ...
Unpacking kubernetescni (0.8.7-00) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.3.3-2.amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.19.3-00.amd64.deb ...
Unpacking kubelet (1.19.3-00) ...
Selecting previously unselected package kubect1.
Preparing to unpack .../6-kubect1_1.19.3-00.amd64.deb ...
Unpacking kubect1 (1.19.3-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.19.3-00.amd64.deb ...
Unpacking kubeadm (1.19.3-00) ...
Setting up conract (1:1.4.5-2) ...
Setting up kubect1 (1.19.3-00) ...
Setting up etables (2.0.11-3bu1d1) ...
Setting up socat (1.7.3.3-2) ...
Setting up kubernetescni (0.8.7-00) ...
Setting up kubelet (1.19.3-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service -> /lib/systemd/system/kubelet.service.
Setting up kubeadm (1.19.3-00) ...
Processing triggers for man-db (2.9.1-1) ...
root@ip-172-31-23-130:/home/ubuntu#

root@ip-172-31-26-32:/home/ubuntu#
Get:3 https://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 socat amd64 1.7.3-3-2 [23 kB]
Get:4 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 cri-tools amd64 1.13.0-0-1 [8775 kB]
Get:5 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubernetescni amd64 0.8.7-00 [25.0 MB]
Get:6 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubelet amd64 1.19.3-00 [18.2 MB]
Get:7 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubect1 amd64 1.19.3-00 [8350 kB]
Get:8 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubeadm amd64 1.19.3-00 [7758 kB]
Fetched 68.5 MB in 11s (6058 kB/s)
Selecting previously unselected package conract.
(Reading database ... 60036 files and directories currently installed.)
Preparing to unpack .../0-conract_1x3al.4.5-2.amd64.deb ...
Unpacking conract (1:1.4.5-2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.13.0-0-1.amd64.deb ...
Unpacking cri-tools (1.13.0-0-1) ...
Selecting previously unselected package etables.
Preparing to unpack .../2-etables_2.0.11-3bu1d1.amd64.deb ...
Unpacking etables (2.0.11-3bu1d1) ...
Selecting previously unselected package kubernetescni.
Preparing to unpack .../3-kubernetescni_0.8.7-00.amd64.deb ...
Unpacking kubernetescni (0.8.7-00) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.3.3-2.amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.19.3-00.amd64.deb ...
Unpacking kubelet (1.19.3-00) ...
Selecting previously unselected package kubect1.
Preparing to unpack .../6-kubect1_1.19.3-00.amd64.deb ...
Unpacking kubect1 (1.19.3-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.19.3-00.amd64.deb ...
Unpacking kubeadm (1.19.3-00) ...
Setting up conract (1:1.4.5-2) ...
Setting up kubect1 (1.19.3-00) ...
Setting up etables (2.0.11-3bu1d1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.13.0-0-1) ...
Setting up kubernetescni (0.8.7-00) ...
Setting up kubelet (1.19.3-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service -> /lib/systemd/system/kubelet.service.
Setting up kubeadm (1.19.3-00) ...
Processing triggers for man-db (2.9.1-1) ...
root@ip-172-31-26-32:/home/ubuntu#
```

Step 5 :-Now we have to run kubeadm init command in master to generate a token to connect with slave.Here my master private ip is 172.31.23.130 .

```
$ kubeadm init --apiserver-advertise-address=<private ip-address-of-master> --pod-network-cidr=192.168.0.0/16 --ignore-preflight-errors=all
```

```
root@ip-172-31-23-130: /home/ubuntu#
root@ip-172-31-23-130: /home/ubuntu#
root@ip-172-31-23-130: /home/ubuntu#
root@ip-172-31-23-130: /home/ubuntu#
root@ip-172-31-23-130: /home/ubuntu# kubeadm init --apiserver-advertise-address=172.31.23.130 --pod-network-cidr=192.168.0.0/16 --ignore-preflight-errors=all
W1018 07:10:15.685360 16651 configset.go:348] WARNING: kubeadm cannot validate component configs for API groups [kubelet.config.k8s.io kubeproxy.config.k8s.io]
[init] Using Kubernetes version: v1.19.3
[preflight] Running pre-flight checks
        [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
        [WARNING Service-Docker]: docker service is not enabled, please run 'systemctl enable docker.service'
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Please follow the guide at https://kubernetes.io/docs/setup/operating-system/container-runtime/#use-systemd-as-cgroup-driver
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-23-130 kubernet.es kubernet.es.default kubernet.es.default.svc kubernet.es.default.svc.cluster.local] and IPs [10.1.1.130]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
```

It will generate a token. copy it and paste in slave to connect the slave in cluster.

Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 172.31.23.130:6443 --token w5pztD.3q74vm38qpil7rju \
--discovery-token-ca-cert-hash sha256:ceb725e2a71da49012fc12adfc86ccf46c36a83630f19f5044a0712ce5823311
root@ip-172-31-23-130:/home/ubuntu#
```



In slave ,

```
root@ip-172-31-26-32: /home/ubuntu#
root@ip-172-31-26-32:/home/ubuntu#
root@ip-172-31-26-32:/home/ubuntu#
root@ip-172-31-26-32:/home/ubuntu# kubeadm join 172.31.23.130:6443 --token wSpztd.3q74vm38qpil7rju \
--discovery-token-ca-cert-hash sha256:cebf25e2a71da49012fc12adfc86ccf46c36a83630f19f5044a0712ce5823311
[preflight] Running pre-flight checks
[WARNING Service-Docker]: docker service is not enabled, please run 'systemctl enable docker.service'
[WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "cgroupv2"
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -oyaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

root@ip-172-31-26-32:/home/ubuntu#
```

Step 6 :- Now we have to create a directory for kubernets cluster to start. In master.

\$ mkdir -p \$HOME/.kube

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

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

```
ubuntu@ip-172-31-23-130: ~
root@ip-172-31-23-130:/home/ubuntu#
root@ip-172-31-23-130:/home/ubuntu#
root@ip-172-31-23-130:/home/ubuntu# exit
exit
ubuntu@ip-172-31-23-130:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-23-130:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-23-130:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-23-130:~$ |
```

Step 7 :- Now we can use the cluster. But if we write kubectl get nodes on master it shows the nodes are not ready in state. Because there is no network plugin in cluster. To install network plugin the following commands are

\$ kubectl apply -f <https://docs.projectcalico.org/v3.3/getting-started/kubernetes/installation/hosted/rbac-kdd.yaml>

```
ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl apply -f https://docs.projectcalico.org/v3.3/getting-started/kubernetes/installation/hosted/rbac-kdd.yaml
Warning: rbac.authorization.k8s.io/v1beta1 ClusterRole is deprecated in v1.17+, unavailable in v1.22+; use rbac.authorization.k8s.io/v1 ClusterRole
clusterrole.rbac.authorization.k8s.io/calico-node created
Warning: rbac.authorization.k8s.io/v1beta1 ClusterRoleBinding is deprecated in v1.17+, unavailable in v1.22+; use rbac.authorization.k8s.io/v1 ClusterRoleBinding
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
ubuntu@ip-172-31-23-130:~$
```


\$ kubectl apply -f <https://docs.projectcalico.org/v3.3/getting-started/kubernetes/installation/hosted/kubernetes-datastore/calico-networking/1.7/calico.yaml>

```
ubuntu@ip-172-31-23-130: ~  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ kubectl apply -f https://docs.projectcalico.org/v3.3/getting-started/kubernetes/installation/hosted/kubernetes-datastore/calico-networking/1.7/calico.yaml  
configmap/calico-config created  
service/calico-typha created  
poddisruptionbudget.policy/calico-typha created  
serviceaccount/calico-node created  
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition  
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/bgppolicies.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created  
unable to recognize "https://docs.projectcalico.org/v3.3/getting-started/kubernetes/installation/hosted/kubernetes-datastore/calico-networking/1.7/calico.yaml": no matches for kind "De  
in version "apps/v1beta1"  
unable to recognize "https://docs.projectcalico.org/v3.3/getting-started/kubernetes/installation/hosted/kubernetes-datastore/calico-networking/1.7/calico.yaml": no matches for kind "De  
in version "extensions/v1beta1"  
ubuntu@ip-172-31-23-130:~$
```

Step 8 :- The previous command will take some time to take effect. After 4-5 mins, try the following command, if both the nodes are in the ready state, Installation is successful!

\$ kubectl get nodes

```
ubuntu@ip-172-31-23-130: ~  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ kubectl get nodes  
NAME                 STATUS    ROLES    AGE     VERSION  
ip-172-31-23-130     Ready     master   8m3s    v1.19.3  
ip-172-31-26-32      Ready     <none>    6m42s   v1.19.3  
ubuntu@ip-172-31-23-130:~$
```

Practical :- 2 Kubernetes Depolyment.

Step 1 :- Now for creating deployment we have to create a yaml file.The command for creating yaml file.

\$ nano hello.yaml

```
ubuntu@ip-172-31-23-130: ~  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ nano hello.yaml  
ubuntu@ip-172-31-23-130:~$
```

Step 2 :- Now the yaml file is created and it will be opened. In the file you have paste the following content.

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: hello-depo  
  labels:  
    app: nginx  
spec:  
  replicas: 3  
  selector:  
    matchLabels:  
      app: nginx  
  template:  
    metadata:  
      labels:  
        app: nginx  
    spec:  
      containers:  
        - name: nginx  
          image: nginx:1.14.2  
          ports:  
            - containerPort: 80
```


ubuntu@ip-172-31-23-130: ~

```
GNU nano 4.8
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-depo
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

Step 3 :- For create deployment ,we have to run kubectl create of yaml file.

\$ kubectl create -f hello.yaml

ubuntu@ip-172-31-23-130: ~

```
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl create -f hello.yaml
deployment.apps/hello-depo created
ubuntu@ip-172-31-23-130:~$ |
```

Deployment is created .

Step 4 :- we can check the replicas where created or not and they are running in running state or not.

\$ kubectl get po

ubuntu@ip-172-31-23-130: ~

```
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
hello-depo-66b6c48dd5-pllsm        1/1     Running   0           50s
hello-depo-66b6c48dd5-sg7sq        1/1     Running   0           50s
hello-depo-66b6c48dd5-w929l        1/1     Running   0           50s
ubuntu@ip-172-31-23-130:~$ |
```

Step 5 :- we get internal cluster ip using the following command.

\$ kubectl get po -o wide

```
ubuntu@ip-172-31-23-130: ~  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ kubectl get po -o wide  
NAME                                READY   STATUS    RESTARTS   AGE   IP            NODE              NOMINATED NODE   READINESS GATES  
hello-depo-66b6c48dd5-p1lsm        1/1     Running   0           113s  10.32.0.5     ip-172-31-26-32   <none>           <none>  
hello-depo-66b6c48dd5-sg7sq        1/1     Running   0           113s  10.32.0.4     ip-172-31-26-32   <none>           <none>  
hello-depo-66b6c48dd5-w929l        1/1     Running   0           113s  10.32.0.6     ip-172-31-26-32   <none>           <none>  
ubuntu@ip-172-31-23-130:~$
```

Step 6 :- Now take the ip address and use curl command to check our depolyments are working properly.

\$ curl 10.32.0.5

```
ubuntu@ip-172-31-23-130: ~  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ curl 10.32.0.5  
<!DOCTYPE html>  
<html>  
<head>  
<title>Welcome to nginx!</title>  
<style>  
  body {  
    width: 35em;  
    margin: 0 auto;  
    font-family: Tahoma, Verdana, Arial, sans-serif;  
  }  
</style>  
</head>  
<body>  
<h1>Welcome to nginx!</h1>  
<p>If you see this page, the nginx web server is successfully installed and  
working. Further configuration is required.</p>  
  
<p>For online documentation and support please refer to  
<a href="http://nginx.org/">nginx.org</a>.<br/>  
Commercial support is available at  
<a href="http://nginx.com/">nginx.com</a>.</p>  
  
<p><em>Thank you for using nginx.</em></p>  
</body>  
</html>  
ubuntu@ip-172-31-23-130:~$ |
```

It seems it is working properly.

Practical :- 3 Kubernetes Services.

Why we need service in Kubernetes ? :

We need services in Kubernetes is to access our application from outside of cluster. ip address of the pods are used inside of the cluster and it cannot work outside. For getting accessible outside we have use external load balancer which is used to listen on particular port and forwards it to access.

Step 1 :- Now for create service we have use kubectl command and we have mention tcp port. It will create a new port in which we access our application from outside of our cluster.

ubuntu@ip-172-31-23-130: ~

```
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ kubectl create service nodeport nginx --tcp=80:80  
service/nginx created  
ubuntu@ip-172-31-23-130:~$
```

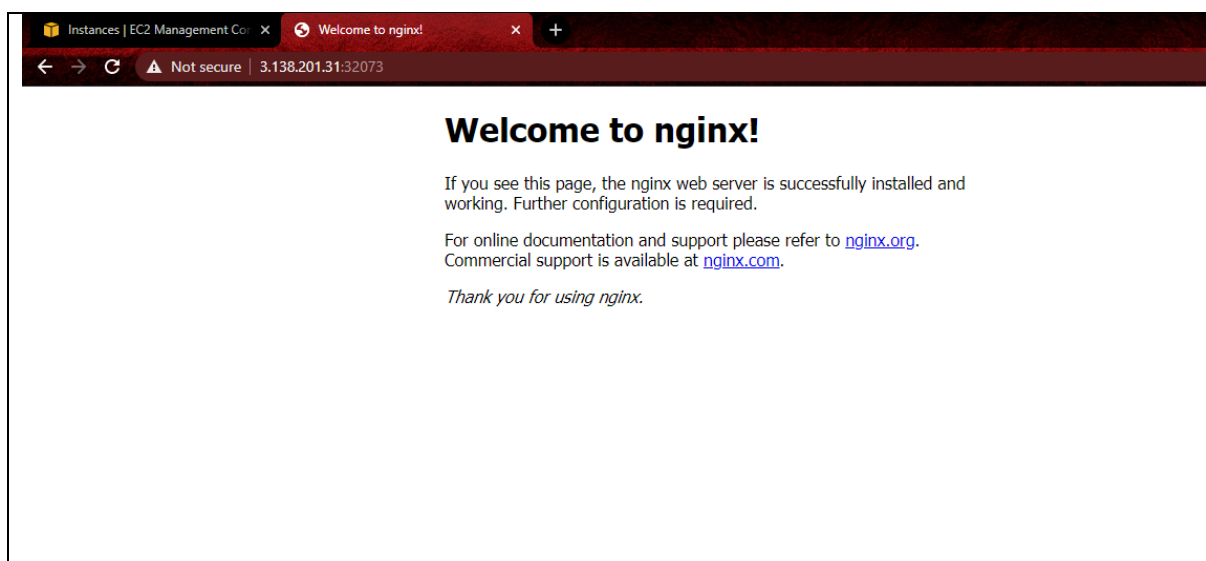
Step 2 :- Now we check the port in which we can access the application.

ubuntu@ip-172-31-23-130: ~

```
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ kubectl get svc nginx  
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE  
nginx     NodePort    10.110.147.230   <none>           80:32073/TCP     38s  
ubuntu@ip-172-31-23-130:~$
```

Step 3 :- Now we to take master-ip address and go ot browser ,then paste the ip address of followed by port number shown in above picture.

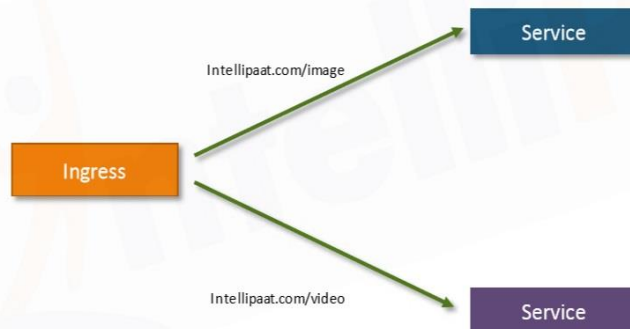
<https://3.138.201.31:32073>



Practical :- 4 Kubernetes Ingress.

What is an Ingress?

Kubernetes ingress is a collection of routing rules that govern how external users access services running in a Kubernetes cluster.



Step 1 :- we have install bare-metal for ingress. Which will install ingress controller and ingress service.

\$ kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.40.2/deploy/static/provider/baremetal/deploy.yaml>

```
ubuntu@ip-172-31-23-130: ~  
ubuntu@ip-172-31-23-130:~$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.40.2/deploy/static/provider/baremetal/deploy.yaml  
namespace/ingress-nginx created  
serviceaccount/ingress-nginx created  
configmap/ingress-nginx-controller created  
clusterrole.rbac.authorization.k8s.io/ingress-nginx created  
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx created  
role.rbac.authorization.k8s.io/ingress-nginx created  
rolebinding.rbac.authorization.k8s.io/ingress-nginx created  
service/ingress-nginx-controller-admission created  
service/ingress-nginx-controller created  
deployment.apps/ingress-nginx-controller created  
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission created  
serviceaccount/ingress-nginx-admission created  
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission created  
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created  
role.rbac.authorization.k8s.io/ingress-nginx-admission created  
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created  
job.batch/ingress-nginx-admission-create created  
job.batch/ingress-nginx-admission-patch created  
ubuntu@ip-172-31-23-130:~$
```

Step 2 :- now we can check weather ingress controller is created or not .

\$ kubectl get svc --all-namespaces

```
ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl get svc --all-namespaces
NAMESPACE      NAME                TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
default        kubernet            ClusterIP  10.96.0.1        <none>           443/TCP          7d6h
default        nginx               NodePort   10.110.147.230   <none>           80:32073/TCP     6d22h
ingress-nginx  ingress-nginx-controller  NodePort   10.97.234.209    <none>           80:32225/TCP,443:31012/TCP  2m45s
ingress-nginx  ingress-nginx-controller-admission  ClusterIP  10.106.211.205   <none>           443/TCP          2m45s
kube-system    calico-typha        ClusterIP  10.103.25.221    <none>           5473/TCP         7d6h
kube-system    kube-dns             ClusterIP  10.96.0.10       <none>           53/UDP,53/TCP,9153/TCP  7d6h
kubernetes-dashboard  dashboard-metrics-scraper  ClusterIP  10.98.68.208     <none>           8000/TCP         6d22h
kubernetes-dashboard  kubernetes-dashboard        NodePort   10.108.175.87    <none>           443:31704/TCP     6d22h
ubuntu@ip-172-31-23-130:~$ |
```

Step 3 :- Now we have to create our application in cluster ip. But we can see that our nginx is nodeport. So, we have to delete it and create new service with cluster ip.

```
ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl get svc --all-namespaces
NAMESPACE      NAME                TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
default        kubernet            ClusterIP  10.96.0.1        <none>           443/TCP          7d6h
default        nginx               NodePort   10.110.147.230   <none>           80:32073/TCP     6d22h
ingress-nginx  ingress-nginx-controller  NodePort   10.97.234.209    <none>           80:32225/TCP,443:31012/TCP  2m45s
ingress-nginx  ingress-nginx-controller-admission  ClusterIP  10.106.211.205   <none>           443/TCP          2m45s
kube-system    calico-typha        ClusterIP  10.103.25.221    <none>           5473/TCP         7d6h
kube-system    kube-dns             ClusterIP  10.96.0.10       <none>           53/UDP,53/TCP,9153/TCP  7d6h
kubernetes-dashboard  dashboard-metrics-scraper  ClusterIP  10.98.68.208     <none>           8000/TCP         6d22h
kubernetes-dashboard  kubernetes-dashboard        NodePort   10.108.175.87    <none>           443:31704/TCP     6d22h
ubuntu@ip-172-31-23-130:~$ |
```

\$ kubectl delete service nginx

```
ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl delete service nginx
service "nginx" deleted
ubuntu@ip-172-31-23-130:~$
```

\$ kubectl create service clusterip nginx --tcp=80:80

```
ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl create service clusterip nginx --tcp=80:80
service/nginx created
ubuntu@ip-172-31-23-130:~$ |
```

```

ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl get svc --all-namespaces

```

NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
default	kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	7d6h
default	nginx	ClusterIP	10.99.17.146	<none>	80/TCP	26s
ingress-nginx	ingress-nginx-controller	NodePort	10.97.234.209	<none>	80:32225/TCP, 443:31012/TCP	5m23s
ingress-nginx	ingress-nginx-controller-admission	ClusterIP	10.106.211.205	<none>	443/TCP	5m23s
kube-system	calico-typha	ClusterIP	10.103.25.221	<none>	5473/TCP	7d6h
kube-system	kube-dns	ClusterIP	10.96.0.10	<none>	53/UDP, 53/TCP, 9153/TCP	7d6h
kubernetes-dashboard	dashboard-metrics-scraper	ClusterIP	10.98.68.208	<none>	8000/TCP	6d22h
kubernetes-dashboard	kubernetes-dashboard	NodePort	10.108.175.87	<none>	443:31704/TCP	6d22h

```

ubuntu@ip-172-31-23-130:~$

```

Step 4 :- Next, we will have to create an ingress rule, create an ingress.yaml file with the below code.

```

ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ touch ingress.yaml
ubuntu@ip-172-31-23-130:~$

```

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
    - http:
        paths:
          - path: /hemanth
            pathType: Prefix
            backend:
              service:
                name: test
                port:
                  number: 80

```

ubuntu@ip-172-31-23-130: ~

```
GNU nano 4.8
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /hemanth
        pathType: Prefix
        backend:
          service:
            name: test
            port:
              number: 80
```

Step 5 :- Finally, create the ingress rule using the following command

\$ kubectl create -f ingress.yaml

ubuntu@ip-172-31-23-130: ~

```
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl create -f ingress.yaml
ingress.networking.k8s.io/minimal-ingress created
ubuntu@ip-172-31-23-130:~$ |
```

Step 6: Let's verify if ingress is working or not, by checking the nodeport of the ingress service, for checking the nodeport use the following command

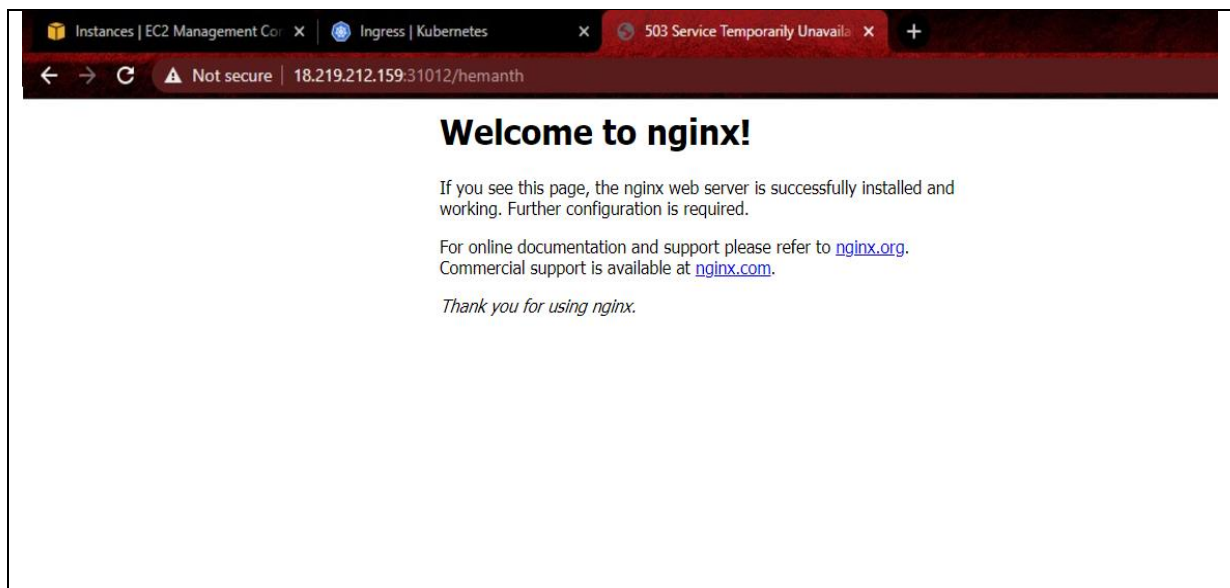
\$ kubectl get svc -n ingress-nginx

ubuntu@ip-172-31-23-130: ~

```
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl get svc --all-namespaces
NAMESPACE   NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)                  AGE
default     kubernetes          ClusterIP   10.96.0.1    <none>        443/TCP                  7d6h
default     nginx               ClusterIP   10.99.17.146 <none>        80/TCP                   13m
ingress-nginx ingress-nginx-controller NodePort    10.97.234.209 <none>        80:32225/TCP,443:31012/TCP 17m
ingress-nginx ingress-nginx-controller-admission ClusterIP   10.106.211.205 <none>        443/TCP                  17m
kube-system calico-typha        ClusterIP   10.103.25.221 <none>        5473/TCP                 7d6h
kube-system kube-dns             ClusterIP   10.96.0.10   <none>        53/UDP,53/TCP,9153/TCP   7d6h
kubernetes-dashboard dashboard-metrics-scraper ClusterIP   10.98.68.208 <none>        8000/TCP                  6d22h
kubernetes-dashboard kubernetes-dashboard NodePort    10.108.175.87 <none>        443:31704/TCP            6d22h
ubuntu@ip-172-31-23-130:~$ |
```

Step 7 :- Now finally we have to browse using master ip address and ingress-controller port number.

<https://18.219.212.159:31012/hemanth>



Practical :- 5 Kubernetes UI.

Dashboard is a web-based Kubernetes user interface. You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application, and manage the cluster resources.

You can use Dashboard to get an overview of applications running on your cluster, as well as for creating or modifying individual Kubernetes resources (such as Deployments, Jobs, DaemonSets, etc).

Dashboard also provides information on the state of Kubernetes resources in your cluster and on any errors that may have occurred.

Step 1 :- The Dashboard UI is not deployed by default. To deploy it, run the following command.

\$ kubectl apply -f

<https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml>

```
ubuntu@ip-172-31-23-130: ~  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$  
ubuntu@ip-172-31-23-130:~$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml  
namespace/kubernetes-dashboard created  
serviceaccount/kubernetes-dashboard created  
service/kubernetes-dashboard created  
secret/kubernetes-dashboard-certs created  
secret/kubernetes-dashboard-csrf created  
secret/kubernetes-dashboard-key-holder created  
configmap/kubernetes-dashboard-settings created  
role.rbac.authorization.k8s.io/kubernetes-dashboard created  
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created  
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created  
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created  
deployment.apps/kubernetes-dashboard created  
service/dashboard-metrics-scraper created  
deployment.apps/dashboard-metrics-scraper created  
ubuntu@ip-172-31-23-130:~$
```

Step 2 :- For Accessing the Dashboard UI,

To protect your cluster data, Dashboard deploys with a minimal RBAC configuration by default. Currently, Dashboard only supports logging in with a Bearer Token. We are creating Service Account with name admin-user in namespace kubernetes-dashboard first.

```
cat <<EOF | kubectl apply -f -  
apiVersion: v1  
kind: ServiceAccount  
metadata:  
  name: admin-user  
  namespace: kubernetes-dashboard  
EOF
```

```

ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ cat <<EOF | kubectl apply -f -
> apiVersion: v1
> kind: ServiceAccount
metadata:
> metadata:
>   name: admin-user
>   namespace: kubernetes-dashboard
> EOF
serviceaccount/admin-user created
ubuntu@ip-172-31-23-130:~$

```

Step 3 :- Now we have to create a ClusterRoleBinding

```

cat <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
namespace: kubernetes-dashboard
EOF

```

```

ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ cat <<EOF | kubectl apply -f -
> apiVersion: rbac.authorization.k8s.io/v1
> kind: ClusterRoleBinding
> metadata:
>   name: admin-user
> roleRef:
>   apiGroup: rbac.authorization.k8s.io
>   kind: ClusterRole
>   name: cluster-admin
> subjects:
> - kind: ServiceAccount
>   name: admin-user
>   namespace: kubernetes-dashboard
> EOF
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
ubuntu@ip-172-31-23-130:~$

```

```
$ kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-  
dashboard get secret | grep admin-user | awk '{print $1}')
```

My generated token is :

Step 5 :- Now for accessing dashboard we have to edit cluster ip to node port because it was a recommended way for accessing dashboard .

```
$ kubectl -n kubernetes-dashboard edit service kubernetes-dashboard
```

```

ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl -n kubernetes-dashboard edit service kubernetes-dashboard
service/kubernetes-dashboard edited
ubuntu@ip-172-31-23-130:~$

```

Now the new yml file is opened in it we have to change Cluster ip to NodePort.

```

ubuntu@ip-172-31-23-130: ~
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"k8s-app":"kubernetes-dashboard"},"name":"kubernetes-dashboard","namespace":"kubernetes-dashboard"},"spec":{"ports":[{"port":443,"targetPort":8443}],"selector":{"k8s-app":"kubernetes-dashboard"}}}
  creationTimestamp: "2020-10-18T15:27:32Z"
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
  resourceVersion: "14584"
  selfLink: /api/v1/namespaces/kubernetes-dashboard/services/kubernetes-dashboard
  uid: 0879b458-1283-40f5-831b-fabb25952f6
spec:
  clusterIP: 10.108.175.87
  externalTrafficPolicy: Cluster
  ports:
  - nodePort: 31704
    port: 443
    protocol: TCP
    targetPort: 8443
  selector:
    k8s-app: kubernetes-dashboard
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}

```

Step 5 :- Next we need to check port on which Dashboard was exposed.

\$ kubectl -n kubernetes-dashboard get service kubernetes-dashboard

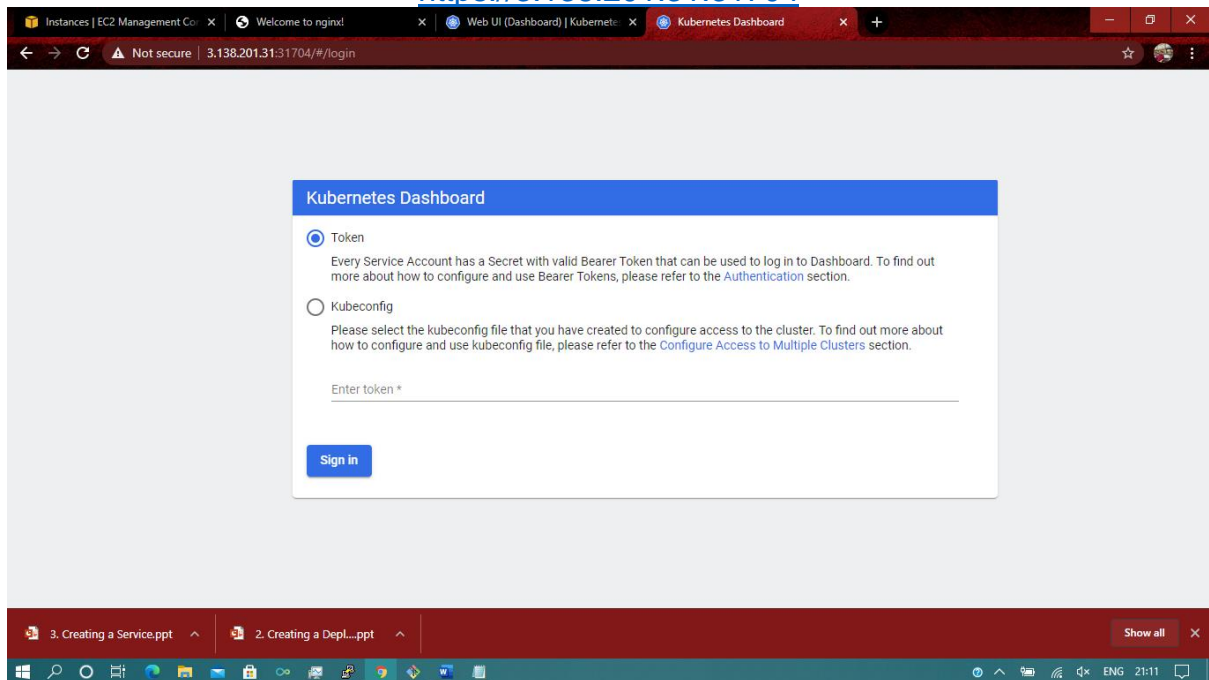
```

ubuntu@ip-172-31-23-130: ~
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$
ubuntu@ip-172-31-23-130:~$ kubectl -n kubernetes-dashboard get service kubernetes-dashboard
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes-dashboard NodePort    10.108.175.87 <none>         443:31704/TCP    11m
ubuntu@ip-172-31-23-130:~$ |

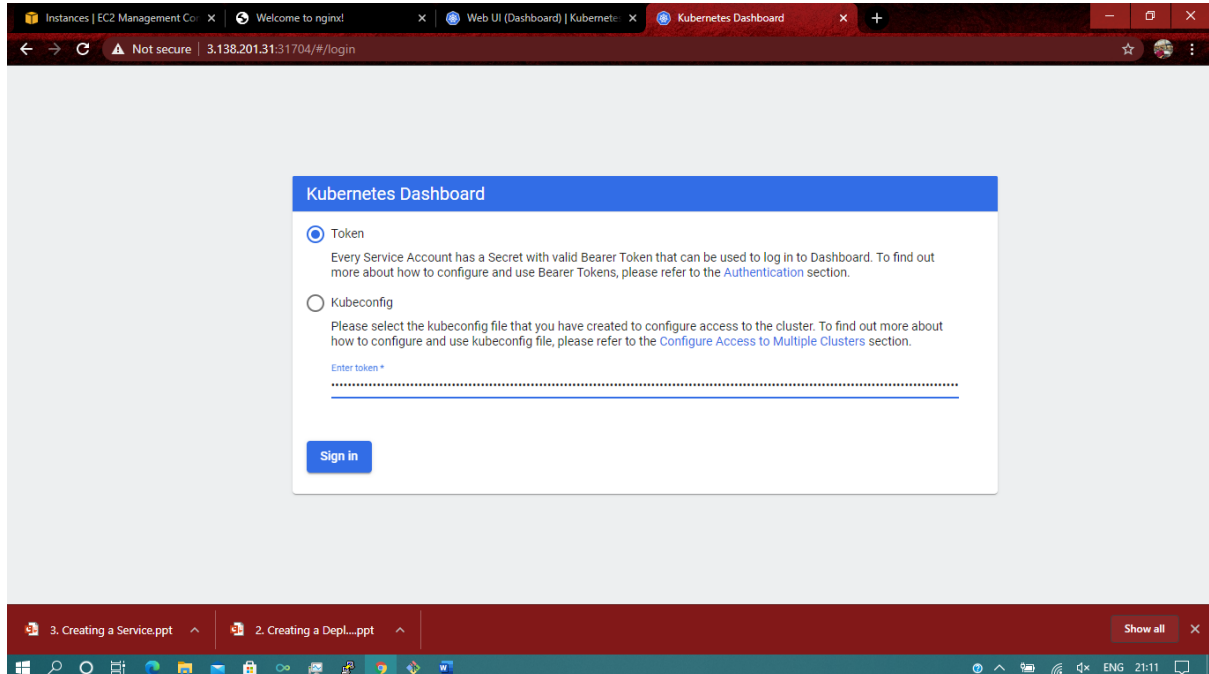
```

Step 6 :- Now copy the ip-address. Paste it in browser along with port number shown in above picture .

<https://3.138.201.31:31704>



Step 7 :- Now paste the token generated by cluster binding in the login page to enter into the bash-board ui .



After pasting token just press sign in.

Instances | EC2 Management Co x Welcome to nginx! x Web UI (Dashboard) | Kubernetes: x Kubernetes Dashboard x +

← → ↻ ⚠ Not secure | 3.138.201.31:31704/#/overview?namespace=default

kubernetes Search + 🔔 👤

Overview

Cluster

- Cluster Roles
- Namespaces
- Nodes
- Persistent Volumes
- Storage Classes

Namespace

default

Overview

Workloads

- Cron Jobs
- Daemon Sets
- Deployments

Workloads

Workload Status

Deployments Pods Replica Sets

Deployments

Name	Namespace	Labels	Pods	Created	Images
------	-----------	--------	------	---------	--------

3. Creating a Service.ppt 2. Creating a Depl...ppt Show all

Windows taskbar: Search, File Explorer, Edge, VS Code, Docker Desktop, Kubernetes Dashboard, 21:11

kubernetes Search + 🔔 👤

Overview

Cluster

- Cluster Roles
- Namespaces
- Nodes
- Persistent Volumes
- Storage Classes

Namespace

default

Overview

Workloads

- Cron Jobs
- Daemon Sets
- Deployments

Pods

Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
hello-depo-66b6c48dd5-plism	default	app: nginx pod-template-hash: 66b6c48dd5	ip-172-31-26-32	Running	0	-	-	49 minutes ago
hello-depo-66b6c48dd5-sg7sq	default	app: nginx pod-template-hash: 66b6c48dd5	ip-172-31-26-32	Running	0	-	-	49 minutes ago
hello-depo-66b6c48dd5-w929l	default	app: nginx pod-template-hash: 66b6c48dd5	ip-172-31-26-32	Running	0	-	-	49 minutes ago

1 - 3 of 3 |< < > >|

Replica Sets

3. Creating a Service.ppt 2. Creating a Depl...ppt Show all

Windows taskbar: Search, File Explorer, Edge, VS Code, Docker Desktop, Kubernetes Dashboard, 21:12

Practical 6: Splunk

Step 1 :- Now create two instances, name them one as master and other as universal forwarder. And connect master using ssh command in gitbash.

ubuntu@ip-172-31-46-43: ~

```
Hemanth Andru@Hemanth-Andru MINGW64 ~
$ cd Downloads/

Hemanth Andru@Hemanth-Andru MINGW64 ~/Downloads
$ ssh -i "hem1.pem" ubuntu@ec2-13-126-194-168.ap-south-1.compute.amazonaws.com
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1024-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Oct 26 04:15:05 UTC 2020

System load:  0.21           Processes:           102
Usage of /:   16.8% of 7.69GB Users logged in:       0
Memory usage: 20%           IPv4 address for eth0: 172.31.46.43
Swap usage:   0%

1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Oct 26 04:14:06 2020 from 157.48.140.191
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-46-43:~$ |
```

Step 2 :- Now just enter into super user mode using the following command :

\$ sudo su

root@ip-172-31-46-43: /home/ubuntu

```
ubuntu@ip-172-31-46-43:~$
ubuntu@ip-172-31-46-43:~$
ubuntu@ip-172-31-46-43:~$ sudo su
root@ip-172-31-46-43:/home/ubuntu# |
```

Step 3 :- Now just enter into opt folder using following command :

\$ cd /opt

```

root@ip-172-31-46-43: /opt
root@ip-172-31-46-43: /#
root@ip-172-31-46-43: /#
root@ip-172-31-46-43: /#
root@ip-172-31-46-43: /# cd /opt/
root@ip-172-31-46-43: /opt#

```

Step 4 :- Now just update ubuntu using the following command :

\$ apt update

```

root@ip-172-31-46-43: /opt
root@ip-172-31-46-43: /opt#
root@ip-172-31-46-43: /opt#
root@ip-172-31-46-43: /opt#
root@ip-172-31-46-43: /opt# apt update
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [342 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [78.5 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [4996 B]
Get:8 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [65.8 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [10.8 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [509 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [64.2 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [8668 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [1256 B]
Get:15 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [540 B]
Get:16 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [116 B]
32% [13 Packages 271 kB/8628 kB 38%]

```

Step 5 :- Now we have to download the splunk package from online using the following command :

**\$ wget -O splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz
'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.0.6&product=splunk&filename=splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz&wget=true'**

```

root@ip-172-31-46-43: /opt
root@ip-172-31-46-43: /opt#
root@ip-172-31-46-43: /opt#
root@ip-172-31-46-43: /opt# wget -O splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.0.6&product=splunk&filename=splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz&wget=true'
--2020-10-26 04:20:03-- https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.0.6&product=splunk&filename=splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz&wget=true
Resolving www.splunk.com (www.splunk.com)... 23.63.110.202, 23.63.110.145
Connecting to www.splunk.com (www.splunk.com)|23.63.110.202|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://download.splunk.com/products/splunk/releases/8.0.6/linux/splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz [following]
--2020-10-26 04:20:05-- https://download.splunk.com/products/splunk/releases/8.0.6/linux/splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz
Resolving download.splunk.com (download.splunk.com)... 13.227.138.113, 13.227.138.69, 13.227.138.129, ...
Connecting to download.splunk.com (download.splunk.com)|13.227.138.113|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 513438808 (490M) [application/x-gzip]
Saving to: 'splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz'

splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz 100%[=====] 489.65M
2020-10-26 04:20:51 (10.7 MB/s) - 'splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz' saved [513438808/513438808]

root@ip-172-31-46-43: /opt# |

```

Step 6 :- Now just check wheater the splunk zip file is downloaded or not.

\$ ls

```
root@ip-172-31-46-43: /opt
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt# ls
splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz
root@ip-172-31-46-43:/opt#
```

Step 7 :- Now unzip the file to get splunk documentation.

\$ tar -xvf (file name after ls)

```
root@ip-172-31-46-43: /opt
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt# tar -xvf splunk-8.0.6-152fb4b2bb96-Linux-x86_64.tgz |
root@ip-172-31-46-43: /opt
splunk/share/splunk/search_mrsparkle/exposed/js/shim/splunk.jquery.csrf.js
splunk/share/splunk/search_mrsparkle/exposed/xml/
splunk/share/splunk/search_mrsparkle/exposed/xml/print.xml
splunk/share/splunk/search_mrsparkle/exposed/robots.txt
splunk/share/splunk/search_mrsparkle/exposed/README.txt
splunk/share/splunk/search_mrsparkle/modules.new/
splunk/share/splunk/search_mrsparkle/modules.new/AbstractModule.conf
splunk/share/splunk/search_mrsparkle/modules.new/DispatchingModule.conf
splunk/share/splunk/search_mrsparkle/modules.new/DispatchingModule.js
splunk/share/splunk/search_mrsparkle/modules.new/nav/
splunk/share/splunk/search_mrsparkle/modules.new/nav/AccountBar.js
splunk/share/splunk/search_mrsparkle/modules.new/nav/LiteBar.js
splunk/share/splunk/search_mrsparkle/modules.new/nav/LiteBar.conf
splunk/share/splunk/search_mrsparkle/modules.new/nav/AccountBar.css
splunk/share/splunk/search_mrsparkle/modules.new/nav/AccountBar.conf
splunk/share/splunk/search_mrsparkle/modules.new/nav/ManagerBar.css
splunk/share/splunk/search_mrsparkle/modules.new/AbstractModule.js
splunk/share/splunk/search_mrsparkle/modules.new/messaging/
splunk/share/splunk/search_mrsparkle/modules.new/messaging/Message.css
splunk/share/splunk/search_mrsparkle/modules.new/messaging/Message.js
splunk/share/splunk/search_mrsparkle/modules.new/messaging/Message.html
splunk/share/splunk/search_mrsparkle/modules.new/messaging/Message.conf
splunk/share/splunk/search_mrsparkle/modules.new/paginator/
splunk/share/splunk/search_mrsparkle/modules.new/paginator/paginator.css
splunk/share/splunk/mbtiles/
splunk/share/splunk/mbtiles/splunk-tiles-dark.mbtiles
splunk/share/splunk/mbtiles/splunk-tiles.mbtiles
splunk/share/splunk/migration/
splunk/share/splunk/migration/app_contents_windows.tar.gz
splunk/share/splunk/migration/app_contents_SplunkDeploymentMonitor.tar.gz
splunk/share/splunk/migration/app_contents_unix.tar.gz
splunk/share/GeoLite2-City.mmbd
splunk/share/mongo-c-driver/
splunk/share/mongo-c-driver/uninstall.sh
splunk/share/iso3166.csv
splunk/share/copyright.txt
splunk/openssl/
splunk/openssl/openssl.cnf
splunk/openssl/misc/
splunk/openssl/misc/c_hash
splunk/openssl/misc/c_issuer
splunk/openssl/misc/c_info
splunk/openssl/misc/CA.sh
splunk/openssl/misc/tsget
splunk/openssl/misc/c_name
splunk/openssl/misc/CA.pl
splunk/openssl/copyright.txt
splunk/splunk-8.0.6-152fb4b2bb96-linux-2.6-x86_64-manifest
splunk/copyright.txt
root@ip-172-31-46-43:/opt#
```

Step 8 :- Now check the folder is created or not using ls command. And then give

permission to the file.

\$ ls

\$ chmod 777 splunk

```
root@ip-172-31-46-43: /opt
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt# chmod 777 splunk
root@ip-172-31-46-43:/opt#
```

Step 9 :- now move to splunk bin folder where all applications of splunk can be worked.

\$ cd splunk/bin

```
root@ip-172-31-46-43: /opt/splunk/bin
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt#
root@ip-172-31-46-43:/opt# cd splunk/bin/
root@ip-172-31-46-43:/opt/splunk/bin#
```

```
root@ip-172-31-46-43: /opt/splunk/bin
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin# ls
ColdStorageArchiver.py  dbmanipulator.py  importtool  mongod_cc  prigraypng  python3  scripts  splunk-optimize-lex  untar.py
bloom  exporttool  installit.py  node  pripalpng  python3.7  scrubber.py  splunkd  walklex
bottle.py  fill_summary_index.py  jars  openssl  priplantopng  python3.7m  searchtest  splunkmon
bttool  genAuditKeys.py  jp.py  parse_xml_buckets.py  prianglisch  pyenv  setSplunkEnv  srm
btprobe  genRootCA.sh  jsmin  parsetest  pripgtopam  pyenv-3.7  shc_upgrade_template.py  tarit.py
bzip2  genSignedServerCert.py  locktest  pcregextest  priweavepng  recover-metadata  signtool  tocsv.py
classify  genSignedServerCert.sh  locktool  pid_check.sh  python  rest_handler.py  slm  tsidx_scan.py
collatorFrozenExample.py  genWebCert.py  mongod-3.4  prichumpng  python2  runScript.py  splunk  tsidxprobe
copyright.txt  genWebCert.sh  mongod  priforgepng  python2.7  safe_restart_cluster_master.py  splunk-optimize  tsidxprobe_plo
root@ip-172-31-46-43:/opt/splunk/bin#
```

Step 10 :- for starting splunk ,we have to accept license of the splunk.

\$./splunk start --accept-license

```
root@ip-172-31-46-43: /opt/splunk/bin
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin# ./splunk start --accept-license|
```

Now it will ask us to create user name and password to create account.

```
root@ip-172-31-46-43: /opt/splunk/bin
root@ip-172-31-46-43: /opt/splunk/bin#
root@ip-172-31-46-43: /opt/splunk/bin#
root@ip-172-31-46-43: /opt/splunk/bin# ./splunk start --accept-license

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: hemanth
Password must contain at least:
  * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
ERROR: Passwords did not match.
Please enter a new password:
Please confirm new password:
ERROR: Password did not meet complexity requirements. Password must contain at least:
  * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
ERROR: Password did not meet complexity requirements. Password must contain at least:
  * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
Copying '/opt/splunk/etc/openldap/ldap.conf.default' to '/opt/splunk/etc/openldap/ldap.conf'.
Generating RSA private key, 2048 bit long modulus
.....+++++
e is 65537 (0x10001)
writing RSA key

Generating RSA private key, 2048 bit long modulus
.....+++++
e is 65537 (0x10001)
writing RSA key

Moving '/opt/splunk/share/splunk/search_mrsparkle/modules.new' to '/opt/splunk/share/splunk/search_mrsparkle/modules'.

Splunk> Now with more code!

Checking prerequisites...
  Checking http port [8000]: open
  Checking mgmt port [8089]: open
  Checking appserver port [127.0.0.1:8065]: open
  Checking kvstore port [8191]: open
  Checking configuration... Done.
    Creating: /opt/splunk/var/lib/splunk
  Checking mgmt port [8089]: open
  Checking appserver port [127.0.0.1:8065]: open
  Checking kvstore port [8191]: open
  Checking configuration... Done.
    Creating: /opt/splunk/var/lib/splunk
    Creating: /opt/splunk/var/run/splunk
    Creating: /opt/splunk/var/run/splunk/appserver/i18n
    Creating: /opt/splunk/var/run/splunk/appserver/modules/static/css
    Creating: /opt/splunk/var/run/splunk/upload
    Creating: /opt/splunk/var/run/splunk/search_telemetry
    Creating: /opt/splunk/var/spool/splunk
    Creating: /opt/splunk/var/spool/dirmoncache
    Creating: /opt/splunk/var/lib/splunk/authDb
    Creating: /opt/splunk/var/lib/splunk/hashDb
New certs have been generated in '/opt/splunk/etc/auth'.
  Checking critical directories... Done
  Checking indexes...
    Validated: _audit _internal _introspection _metrics _metrics_rollup _telemetry _thefishbucket history main summary
    Done
  Checking filesystem compatibility... Done
  Checking conf files for problems...
    Done
  Checking default conf files for edits...
    Validating installed files against hashes from '/opt/splunk/splunk-8.0.6-152fb4b2bb96-linux-2.6-x86_64-manifest'
    All installed files intact.
    Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'privKeySecure.pem'
-----
Signature ok
subject=CN=ip-172-31-46-43/0=SplunkUser
Getting CA Private Key
writing RSA key
Done

Waiting for web server at http://127.0.0.1:8000 to be available..... Done

If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

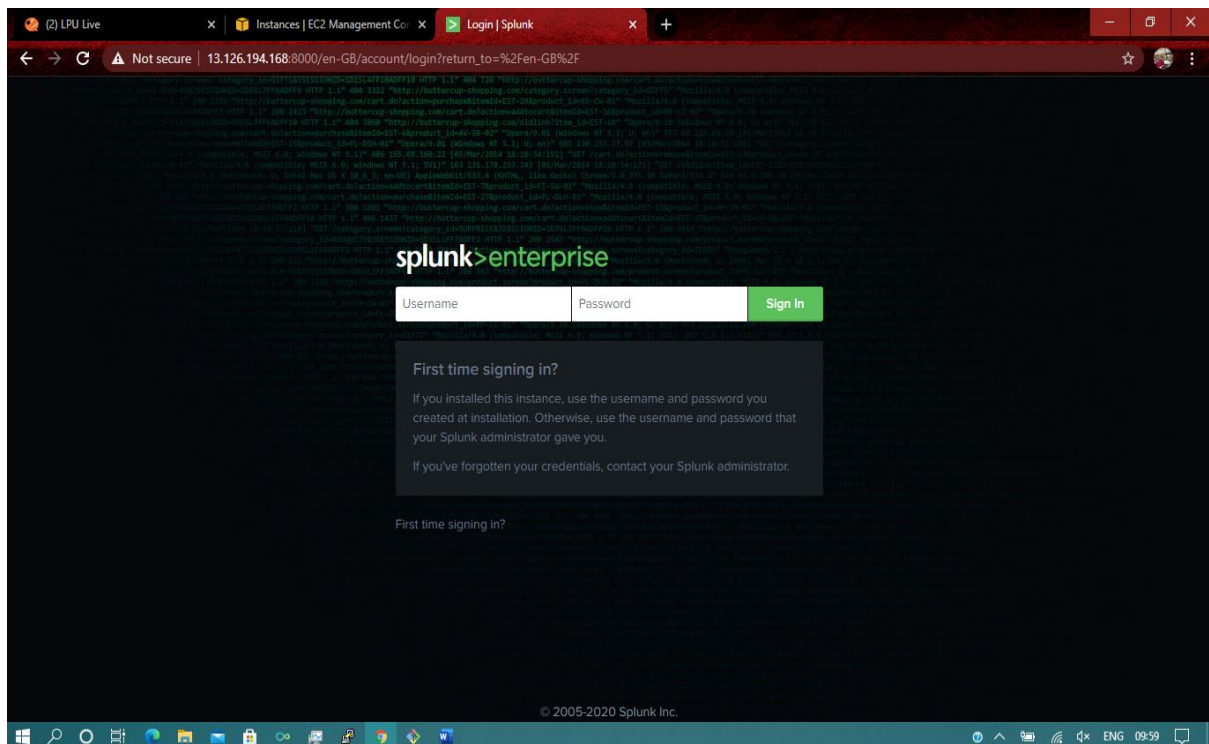
The Splunk web interface is at http://ip-172-31-46-43:8000

root@ip-172-31-46-43: /opt/splunk/bin |
```

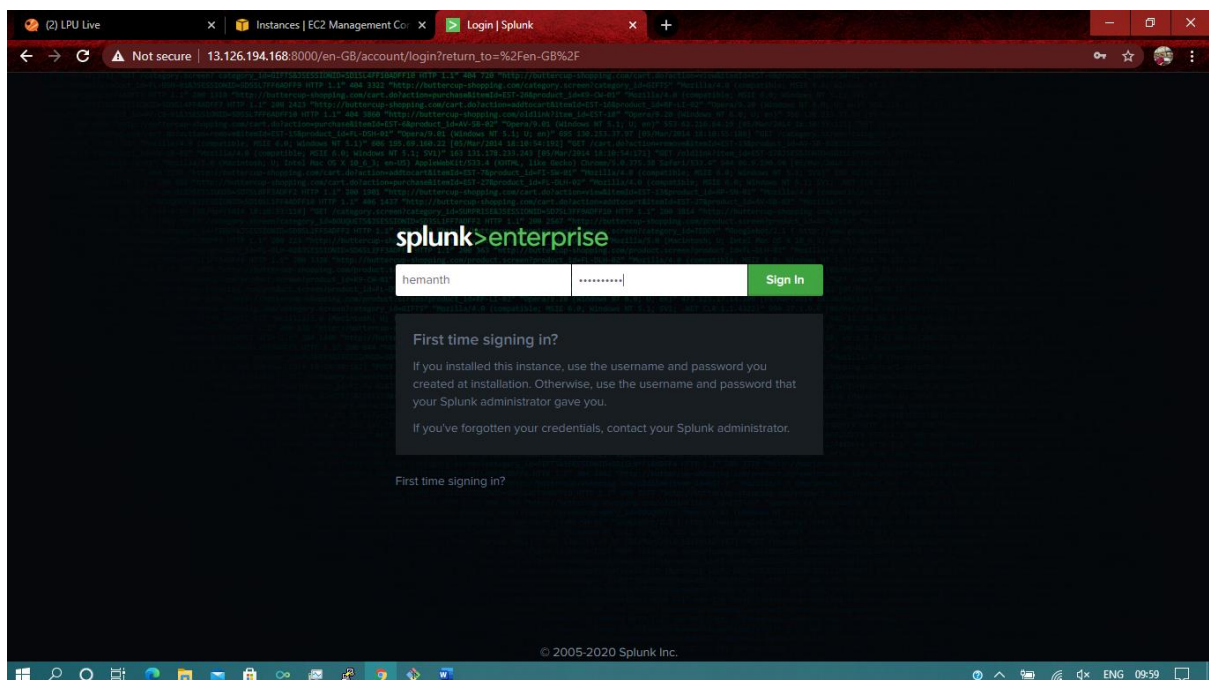
Step 11 :- now copy the ip-address of master from ec2 instance and paste it in

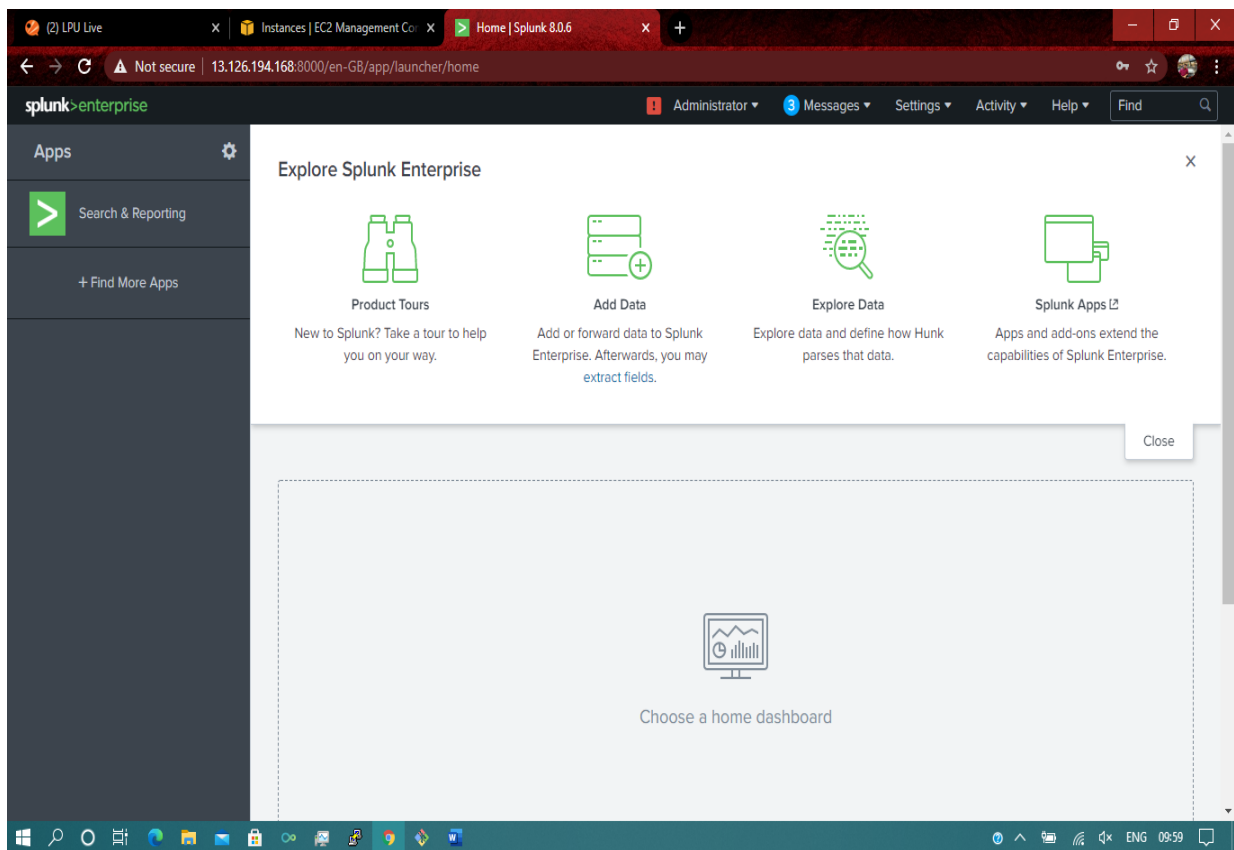
browser following with port 8000.

<https://13.126.194.168:8000>



Now enter the details of username and password created before.





Now lets start with installing splunk frowder..

Step 1 :- Now open universal forwarder ec2 instance that named before and connect with ssh command.


```
ubuntu@ip-172-31-41-90: ~  
Hemanth Andru@Hemanth-Andru MINGW64 ~  
$ cd Downloads/  
Hemanth Andru@Hemanth-Andru MINGW64 ~/Downloads  
$ ssh -i "hem1.pem" ubuntu@ec2-13-127-112-149.ap-south-1.compute.amazonaws.com  
The authenticity of host 'ec2-13-127-112-149.ap-south-1.compute.amazonaws.com (13.127.112.149)' can't be established.  
ECDSA key fingerprint is SHA256:BnKrvL0yS423kpfQa50kFxaTGxgqYsIvM1Gx5Kck4Qs.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-13-127-112-149.ap-south-1.compute.amazonaws.com,13.127.112.149' (ECDSA) to the list of known hosts.  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1024-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Mon Oct 26 04:34:24 UTC 2020  
  
System load:  0.0          Processes:            100  
Usage of /:   16.7% of 7.69GB   Users logged in:     0  
Memory usage: 20%          IPv4 address for eth0: 172.31.41.90  
Swap usage:   0%  
  
1 update can be installed immediately.  
0 of these updates are security updates.  
To see these additional updates run: apt list --upgradable  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-41-90:~$
```

Step 2 :- Enter into super using sudo su and enter into opt folder using following command :

\$ sudo su

\$cd /opt

```
root@ip-172-31-41-90: /opt  
ubuntu@ip-172-31-41-90:~$  
ubuntu@ip-172-31-41-90:~$  
ubuntu@ip-172-31-41-90:~$  
ubuntu@ip-172-31-41-90:~$ sudo su  
root@ip-172-31-41-90:/home/ubuntu#  
root@ip-172-31-41-90:/home/ubuntu# cd /opt/  
root@ip-172-31-41-90:/opt#
```

Step 3 :- update the cli using the following command :

\$ apt update

```

root@ip-172-31-41-90:/opt# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [342 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [78.5 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [4996 B]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [65.8 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [10.8 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [509 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [64.2 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [8668 B]
Get:14 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [1256 B]
Get:15 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [540 B]
Get:16 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [116 B]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [618 kB]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [156 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [10.4 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [78.2 kB]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [12.4 kB]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [348 B]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [673 kB]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [127 kB]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [12.3 kB]
Get:31 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [15.1 kB]
Get:32 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [3892 B]
Get:33 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [480 B]
Get:34 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [112 B]
Get:35 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:36 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [4012 B]
Get:37 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [1448 B]
Get:38 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [224 B]
Get:39 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 17.4 MB in 6s (2854 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
68 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-41-90:/opt#

```

Step 4 :- now we have download the splunk-forwarder zip file from online using wget command :

```

$ wget -O splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.0.3&product=universalforwarder&filename=splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz&wget=true'

```

```

root@ip-172-31-41-90:/opt
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt# wget -O splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.0.3&product=universalforwarder&filename=splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz&wget=true'

```

```

root@ip-172-31-41-90: /opt
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt# wget -O splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz 'https://www.splunk.com/bin/splunk/products/universalforwarder/releases/8.0.3/linux/splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz&wget=true'
--2020-10-26 04:36:34-- https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux
Resolving www.splunk.com (www.splunk.com)... 23.63.110.202, 23.63.110.145
Connecting to www.splunk.com (www.splunk.com)|23.63.110.202|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://download.splunk.com/products/universalforwarder/releases/8.0.3/linux/splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz
--2020-10-26 04:36:35-- https://download.splunk.com/products/universalforwarder/releases/8.0.3/linux/splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz
Resolving download.splunk.com (download.splunk.com)... 13.227.138.114, 13.227.138.113, 13.227.138.69, ...
Connecting to download.splunk.com (download.splunk.com)|13.227.138.114|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27023654 (26M) [application/x-gzip]
Saving to: 'splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz'

splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64 100%[=====]
2020-10-26 04:36:40 (7.11 MB/s) - 'splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz' saved [27023654/27023654]

```

Step 5 :- Now just check wheater the splunk-forwarder zip file is downloaded **or not**.

\$ ls

```

root@ip-172-31-41-90: /opt
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt# ls
splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz
root@ip-172-31-41-90:/opt#

```

Step 6 :- Now just check wheater the splunk-forwarder zip file is downloaded or not.

\$ tar -xvf (file name)

```

root@ip-172-31-41-90: /opt
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt# tar -xvf splunkforwarder-8.0.3-a6754d8441bf-Linux-x86_64.tgz

```

```

root@ip-172-31-41-90: /opt
splunkforwarder/etc/system/default/literals.conf
splunkforwarder/etc/system/default/sourcetypes.conf
splunkforwarder/etc/system/default/federated.conf
splunkforwarder/etc/system/default/conf.conf
splunkforwarder/etc/system/default/app.conf
splunkforwarder/etc/system/default/outputs.conf
splunkforwarder/etc/system/default/source-classifier.conf
splunkforwarder/etc/system/default/authorize.conf
splunkforwarder/etc/system/bin/
splunkforwarder/etc/system/bin/gnome_keyring.py
splunkforwarder/etc/splunk.version
splunkforwarder/etc/log-debug.cfg
splunkforwarder/etc/log.cfg
splunkforwarder/etc/deployment-apps/
splunkforwarder/etc/deployment-apps/README
splunkforwarder/etc/splunk-launch.conf.default
splunkforwarder/etc/log-dfs-search.cfg
splunkforwarder/etc/shcluster/
splunkforwarder/etc/shcluster/users/
splunkforwarder/etc/shcluster/users/README
splunkforwarder/etc/shcluster/apps/
splunkforwarder/etc/shcluster/apps/README
splunkforwarder/etc/licenses/
splunkforwarder/etc/licenses/forwarder/
splunkforwarder/etc/licenses/forwarder/splunkforwarder.lic
splunkforwarder/etc/modules/
splunkforwarder/etc/modules/parsing/
splunkforwarder/etc/modules/parsing/config.xml
splunkforwarder/etc/modules/input/
splunkforwarder/etc/modules/input/tailfile/
splunkforwarder/etc/modules/input/tailfile/config.xml
splunkforwarder/etc/modules/input/structuredparsing/
splunkforwarder/etc/modules/input/structuredparsing/config.xml
splunkforwarder/etc/modules/input/fschangemanager/
splunkforwarder/etc/modules/input/fschangemanager/config.xml
splunkforwarder/etc/modules/input/TCP/
splunkforwarder/etc/modules/input/TCP/config.xml
splunkforwarder/etc/modules/input/FIFO/
splunkforwarder/etc/modules/input/FIFO/config.xml
splunkforwarder/etc/modules/input/UDP/
splunkforwarder/etc/modules/input/UDP/config.xml
splunkforwarder/etc/modules/input/exec/
splunkforwarder/etc/modules/input/exec/config.xml
splunkforwarder/etc/modules/input/RemoteQueue/
splunkforwarder/etc/modules/input/RemoteQueue/config.xml
splunkforwarder/etc/prettyprint.xsl
splunkforwarder/etc/log-btool.cfg
splunkforwarder/include/
splunkforwarder/include/copyright.txt
root@ip-172-31-41-90:/opt#

```

Step 7 :- Now check the folder is created or not using ls command .And the give permission to the folder.

\$ ls

\$ chmod 777 splunkforwarder

```

root@ip-172-31-41-90: /opt
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt# chmod 777 splunkforwarder
root@ip-172-31-41-90:/opt#

```

Step 8 :- now move to splunk-forwarder bin folder where all applications of splunkforwarder can be worked.

\$ cd splunkforwarder/bin/

```
root@ip-172-31-41-90: /opt/splunkforwarder/bin
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt#
root@ip-172-31-41-90:/opt# cd splunkforwarder/bin/
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
```

Step 9 :- we check the files that are included in bin folder .

\$ ls

```
root@ip-172-31-41-90: /opt/splunkforwarder/bin
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin# ls
bttool  bzip2  copyright.txt  genSignedServerCert.sh  openssl  prichunkpng  prigrey.png  pripamtopng  pripngtopam  scripts
btprobe  classify  genRootCA.sh  genWebCert.sh  pid_check.sh  priforge.png  pripal.png  pripnglsch  priweave.png  setSplunkEnv
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
```

Step 10 :- for starting splunk-forwarder ,we have to accept license of the splunk. In it we have to create username and password.

\$./splunk start --accept-license

```
root@ip-172-31-41-90: /opt/splunkforwarder/bin
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin# ./splunk start --accept-license

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: hemanth
Password must contain at least:
* 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:

Splunk> Now with more code!

Checking prerequisites...
  Checking mgmt port [8089]: open
    Creating: /opt/splunkforwarder/var/lib/splunk
    Creating: /opt/splunkforwarder/var/run/splunk
    Creating: /opt/splunkforwarder/var/run/splunk/appserver/i18n
    Creating: /opt/splunkforwarder/var/run/splunk/appserver/modules/static/css
    Creating: /opt/splunkforwarder/var/run/splunk/upload
    Creating: /opt/splunkforwarder/var/run/splunk/search_telemetry
    Creating: /opt/splunkforwarder/var/spool/splunk
    Creating: /opt/splunkforwarder/var/spool/dirmoncache
    Creating: /opt/splunkforwarder/var/lib/splunk/authDb
    Creating: /opt/splunkforwarder/var/lib/splunk/hashDb
New certs have been generated in '/opt/splunkforwarder/etc/auth'.
  Checking conf files for problems...
    Done
  Checking default conf files for edits...
  Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-8.0.3-a6754d8441bf-linux-2.6-x86_64-manifest'
  All installed files intact.
    Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
```

Step 11 :- now we add our universal forwarder to your master using master-ip address in following command :

\$/splunk add forward-server 13.127.149:8000

```

root@ip-172-31-41-90: /opt/splunkforwarder/bin
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin# ./splunk add forward-server 13.127.112.149:8000
Splunk username: hemanth
Password:
Added forwarding to: 13.127.112.149:8000.
root@ip-172-31-41-90:/opt/splunkforwarder/bin#

```

Step 12 :- now we have to add the log that has to be observed and indexed by splunk.

\$./splunk add monitor /var/log/syslog -index main -sourcetype MyUFServerLogs

```

root@ip-172-31-41-90: /opt/splunkforwarder/bin
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin#
root@ip-172-31-41-90:/opt/splunkforwarder/bin# ./splunk add monitor /var/log/syslog -index main -sourcetype MyUFServerLogs
Added monitor of '/var/log/syslog'.
root@ip-172-31-41-90:/opt/splunkforwarder/bin#

```

Step 13 :- Now open the master instance and type the enable command and listen on particular port. it will ask username and password.

\$./splunk enable listen 9997

```

root@ip-172-31-46-43: /opt/splunk/bin
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin#
root@ip-172-31-46-43:/opt/splunk/bin# ./splunk enable listen 8000
Splunk username: hemanth
Password:
Parameter name: TCP port 8000 is not available.
root@ip-172-31-46-43:/opt/splunk/bin# ./splunk enable listen 9997
Listening for Splunk data on TCP port 9997.
root@ip-172-31-46-43:/opt/splunk/bin#

```

Step 14 :- Now open the browser and open the login splunk using master ip and then go to search and type index="main".

Instances | EC2 Management Co...

Search | Splunk 8.0.6

← → ↻ ⚠ Not secure | 65.0.80.33:8000/en-GB/app/search/search

splunk>enterprise

App: Search & Reporting ▾

Administrator ▾ 6 Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

Search Analytics Datasets Reports Alerts Dashboards

Search & Reporting

Search

index="main"

Last 24 hours ▾ 🔍

No Event Sampling ▾ Smart Mode ▾

How to Search

If you are not familiar with the search features, or want to learn more, see one of the following resources.

Documentation 🔗

Tutorial 🔗

What to Search

3 Events

INDEXED

13 minutes ago

EARLIEST EVENT

13 minutes ago

LATEST EVENT

Data Summary

> Search History