

# Building Kubernetes From Source

---

# Prerequisite

---

- An ubuntu master server (version 18.04+)
- More than 4.5GB of memory (preferably 8GB+)
- 60GB of free space

# Building K8s From Source - 1

---

## - Install docker

```
sudo apt update
```

```
sudo apt install -y docker.io
```

```
sudo usermod -aG docker $USER
```

```
newgrp docker
```

```
sudo systemctl enable docker
```

## - Install compilation and other tools

```
sudo apt install -y gcc make socat curl git
```

# Building K8s From Source - 2

---

## - **Compile and build K8s**

```
git clone https://github.com/kubernetes/kubernetes  
cd kubernetes  
make quick-release
```

# Building K8s From Source - 2

---

## - Load Binary as Docker Image

```
cd _output/release-tars
tar -xvf kubernetes-server-linux-amd64.tar.gz
cd kubernetes/server/bin
docker load < kube-controller-manager.tar
docker images
```

```
eunchan@ubuntu:~/Desktop/kubernetes/_output/release-tars/kubernetes/server/bin$ docker images
REPOSITORY                                TAG
k8s.gcr.io/kube-controller-manager-amd64  v1.19.0-alpha.2.223_961a5ed144cf0f-dirty
```

# Install Kubernetes

---

# Prerequisite

---

- An ubuntu slave server (version 18.04+)
- More than 2GB of memory
- 40GB of free space

# Install K8s - 1

---

- Install Docker on both master / slave node(Slide 3)
- Install Kubeadm on both nodes

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add  
  
sudo apt-add-repository "deb https://apt.kubernetes.io/ kubernetes-xenial main"  
  
sudo apt install kubeadm
```



# Install K8s - 2

---

- Disable swap memory on both nodes

```
sudo swapoff -a
```

- Set Hostnames for both nodes

```
(master node) sudo hostnamectl set-hostname master-node
```

```
(slave node) sudo hostnamectl set-hostname slave-node
```

# Install K8s - 3

---

- Initialize Kubernetes on master node as root

```
kubeadm init --pod-network-cidr=10.244.0.0/16
```

- Apply config on master node

```
mkdir -p $HOME/.kube
```

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

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

# Install K8s - 4

---

- Join the master node on slave node

```
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/  
  
Then you can join any number of worker nodes by running the following on each as root:  
kubeadm join 192.168.0.9:6443 --token 3mt02c.en60f3fe1iip99n8 \  
  --discovery-token-ca-cert-hash sha256:e9e383fbd2fa3f9b525adf6837cf8e7915e433f6e512bf9f80cf919df64c91c2
```



Copy and run on slave node

# Install K8s - 5

---

kubectl get nodes

```
eunchan@ubuntu:~/Desktop/kubernetes/_output/release-tars/kubernetes/server/bin$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master-node	Ready	master	30m	v1.18.2
slave-node	Ready	<none>	28m	v1.18.2

# Install K8s - 6

---

## - Apply built binary

```
sudo vim /etc/kubernetes/manifests/kube-controller-manager
```

```
27  - --use-service-account-credentials=true
28  image: k8s.gcr.io/kube-controller-manager-amd64:v1.19.0-alpha.2.223_961a5ed144cf0f-dirty
29  imagePullPolicy: IfNotPresent
```

```
Conditions:
  Type           Status  Reason
  ----           -
  AbleToScale    True    SucceededGetScale
  ScalingActive  False   FailedGetResourceMetric: the HPA was unable to compute the replica count: unable to get metrics for resource cpu: no metrics re
turned from resource metrics API
Events:
```

# Working with Kubernetes

---

# DashBoard (GUI)

---

**Deploy dashboard:**

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-beta8/aio/deploy/recommended.yaml
```

```
kubectl proxy
```

Enter <http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/> on a browser to enter the dashboard

# DashBoard (GUI) - Token

---

admin\_user\_account.yml

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
```

admin\_user\_role.yml

```
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
```



# DashBoard (GUI) - Token

```
kubectl apply -f admin_user_account.yml
```

```
kubectl apply -f admin_user_role.yml
```

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

token: eyJhbGciOiJSUzI1NiIsImtpZCI6IjFyOXBzSEE3RXo1WS1mMWRpTTNNUGVddlRUS0dw  
dEpLdDR4RlhFR0FvQXcifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZ  
XRlcyc5pby9zZXJ2aWNlYWNib3VudC9uYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsImt1Y  
mVybmV0ZXMuaW8vc2VydmljZWJfY291bnQvc2VjcmlvLm5lbWUoiOiJhZG1pb11c2VyLXRva2VuLW5jd  
25oIiwia3ViZXJuZXRlcyc5pby9zZXJ2aWNlYWNib3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkb  
WluLXVzZXIiLCJrdWJlcm5ldGVzLm1vL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNib3VudC51aWQiO  
iJJNTAyMTMxNi1mZWVKLTQxYWETyWZjNi00YzFhOTViNDUyZDgiLCJzdWIiOiJzeXN0ZW06c2VydmljZ  
WFjY291bnQ6a3ViZXJuZXRlcyc1kYXNoYm9hcmQ6YWRtaW4tdXNlciJ9.kcvMqivKDRd0J3ipiAuPV\_eO  
6kv\_CFYrSa1INzyu3xx47QCq148pPkVskLHFDZBQhue8Vhu5gzqSVz4G1Tar\_WWL4eyJLDYRuSUXM-Am  
sxHBF6ooos60SGoit5is90IXxl7o46DpCmUKYBSgM8TSnKu3BDqnr\_kTnIT63ScB-JAaMMeGMBNXIEw1  
pa7qiWKqY-Vqth9RMXodN\_xDbwGWYYLZocDv1K-C9T\_6UHdu\_wcWiUy\_TJMFJQLoQWBENCvZie6bn8qq  
ki2xXOPz9njxLb9GeuLT\_mha-jklInj5wqDOTVFq5wXGVPohxKAZYRR1MX-K2QvNe3ELvgJh\_vzskQ

# Metric Server

---

Fetch metric server:

wget <https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.3.6/components.yaml>

```
88     args:
89         - --cert-dir=/tmp
90         - --secure-port=4443
91         - --kubelet-insecure-tls
92         - --kubelet-preferred-address-types=InternalIP
93     ports:
94     - name: main-port
95       containerPort: 4443
```

← Add to  
components.yaml

kubectl apply -f components.yaml