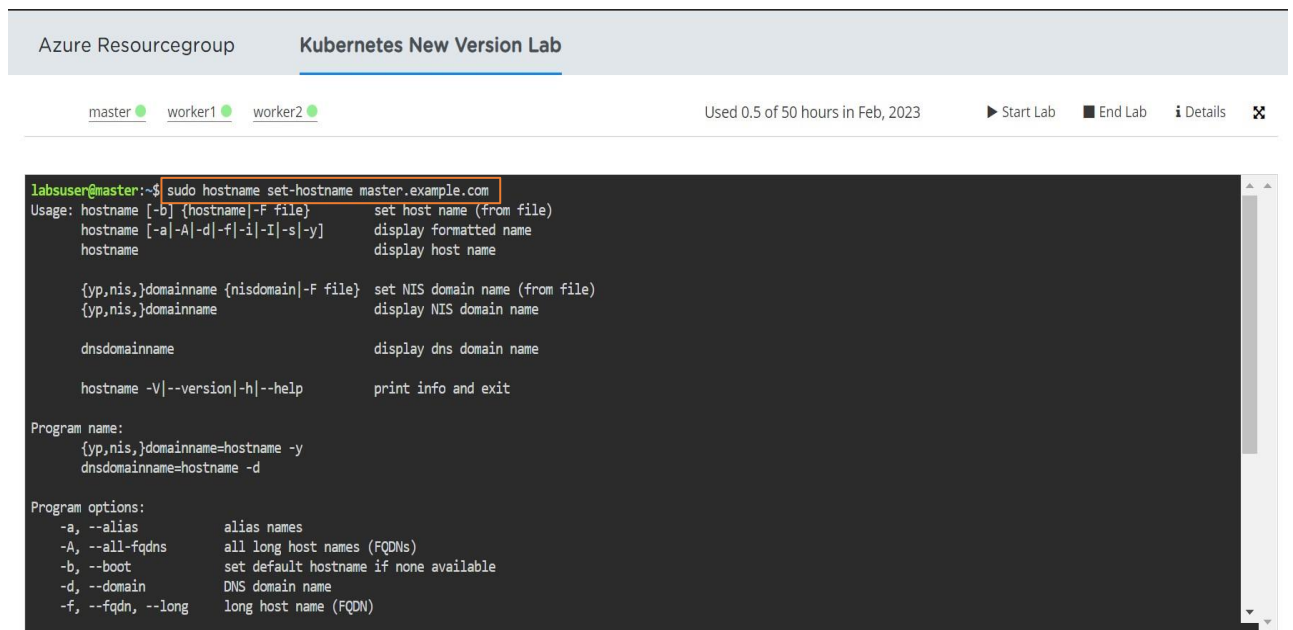


SETTING MASTER NODE IN KUBERNETES

1. Setting hostname for master-plane (Master Node)



```
labsuser@master:~$ sudo hostname set-hostname master.example.com
Usage: hostname [-b] {hostname|-F file}      set host name (from file)
        hostname [-a|-A|-d|-f|-i|-I|-s|-y]    display formatted name
        hostname                                display host name

        {yp,nis,}domainname {nisdomain|-F file} set NIS domain name (from file)
        {yp,nis,}domainname                    display NIS domain name

        dnsdomainname                          display dns domain name

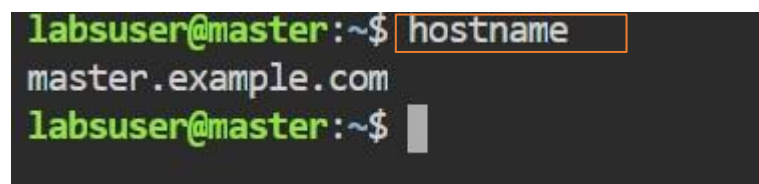
        hostname -V|--version|-h|--help        print info and exit

Program name:
{yp,nis,}domainname=hostname -y
dnsdomainname=hostname -d

Program options:
-a, --alias                alias names
-A, --all-fqdns            all long host names (FQDNs)
-b, --boot                 set default hostname if none available
-d, --domain               DNS domain name
-f, --fqdn, --long         long host name (FQDN)
```

Description: Hostname has been set executing command “sudo hostname set-hostname master.example.com”.

OUTPUT: Checking hostname using command “hostname”



```
labsuser@master:~$ hostname
master.example.com
labsuser@master:~$
```

2. Checking Docker status for configuring Master node using command “sudo service docker status”.

```
Azure Resourcegroup Kubernetes New Version Lab

master worker1 worker2

Used 0.5 of 50 hours in Feb, 2023 ▶ Start Lab ■ End Lab ⓘ Details ✕

labsuser@master:~$ sudo service docker status
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-02-01 10:07:32 UTC; 12min ago
     TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 729 (dockerd)
   Tasks: 15
   Memory: 123.9M
   CGroup: /system.slice/docker.service
           └─729 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 01 10:07:32 master.example.com dockerd[729]: time="2023-02-01T10:07:32.030398130Z" level=info msg="API listen on /run/docker.sock"
Feb 01 10:13:06 master.example.com dockerd[729]: time="2023-02-01T10:13:06.515855545Z" level=info msg="ignoring event" container=389b80b47d7c5fed7b37ebfd45cefd769afdb5
Feb 01 10:13:06 master.example.com dockerd[729]: time="2023-02-01T10:13:06.906734515Z" level=info msg="ignoring event" container=6989e1962b587abddb47e75fce91ca2024395f8
Feb 01 10:13:17 master.example.com dockerd[729]: time="2023-02-01T10:13:17.088254486Z" level=info msg="Container failed to exit within 10s of signal 15 - using the force"
Feb 01 10:13:17 master.example.com dockerd[729]: time="2023-02-01T10:13:17.138816183Z" level=info msg="ignoring event" container=d87e1cb3de4a0aaca0b5c1fed9c1d2dd82891b
Feb 01 10:13:17 master.example.com dockerd[729]: time="2023-02-01T10:13:17.375569248Z" level=info msg="ignoring event" container=99c974b9c55310ff1351eb317a960c83715842b
Feb 01 10:13:17 master.example.com dockerd[729]: time="2023-02-01T10:13:17.542867275Z" level=info msg="ignoring event" container=34d6300e9d2e4b8717ff85827fea215c658ea1b
Feb 01 10:13:17 master.example.com dockerd[729]: time="2023-02-01T10:13:17.710952162Z" level=info msg="ignoring event" container=49b61831beb198980e8397557d596dead37b85b
Feb 01 10:13:17 master.example.com dockerd[729]: time="2023-02-01T10:13:17.866334365Z" level=info msg="ignoring event" container=25857f5076c1af26ec36738e7ff63f422660db5
Feb 01 10:13:18 master.example.com dockerd[729]: time="2023-02-01T10:13:18.043644927Z" level=info msg="ignoring event" container=9ce8d5e1f99ca534d0d31d63d670420a4889ad5
```

OUTPUT : Reflects docker as active(running).

3. Checking kubelet status using command “sudo service kubelet status”.

```
Azure Resourcegroup Kubernetes New Version Lab

master worker1 worker2

Used 0.5 of 50 hours in Feb, 2023 ▶ Start Lab ■ End Lab ⓘ Details ✕

● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/kubelet.service.d
           └─10-kubeadm.conf
   Docs: https://kubernetes.io/docs/home/
   Main PID: 5349 (kubelet)
   Tasks: 16 (limit: 9405)
   Memory: 37.5M
   CGroup: /system.slice/kubelet.service
           └─5349 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/

Feb 01 10:20:51 master.example.com kubelet[5349]: E0201 10:20:51.383406 5349 kubelet.go:2347] "Container runtime network not ready" networkReady="NetworkReady=false"
Feb 01 10:20:55 master.example.com kubelet[5349]: I0201 10:20:55.412360 5349 cni.go:240] "Unable to update cni config" err="no networks found in /etc/cni/net.d"
Feb 01 10:20:56 master.example.com kubelet[5349]: E0201 10:20:56.396389 5349 kubelet.go:2347] "Container runtime network not ready" networkReady="NetworkReady=false"
Feb 01 10:21:00 master.example.com kubelet[5349]: I0201 10:21:00.413098 5349 cni.go:240] "Unable to update cni config" err="no networks found in /etc/cni/net.d"
Feb 01 10:21:01 master.example.com kubelet[5349]: E0201 10:21:01.405955 5349 kubelet.go:2347] "Container runtime network not ready" networkReady="NetworkReady=false"
Feb 01 10:21:05 master.example.com kubelet[5349]: I0201 10:21:05.414050 5349 cni.go:240] "Unable to update cni config" err="no networks found in /etc/cni/net.d"
Feb 01 10:21:06 master.example.com kubelet[5349]: E0201 10:21:06.416560 5349 kubelet.go:2347] "Container runtime network not ready" networkReady="NetworkReady=false"
Feb 01 10:21:10 master.example.com kubelet[5349]: I0201 10:21:10.415462 5349 cni.go:240] "Unable to update cni config" err="no networks found in /etc/cni/net.d"
Feb 01 10:21:11 master.example.com kubelet[5349]: E0201 10:21:11.426711 5349 kubelet.go:2347] "Container runtime network not ready" networkReady="NetworkReady=false"
lines 1-21
```

OUTPUT: Prompts Kubelet status as active (running).

4. Activating the Master control – plane (Master node) using command “sudo kubeadm init”.

Azure Resourcegroup

Kubernetes New Version Lab

master worker1 worker2

Used 0.5 of 50 hours in Feb, 2023

Start Lab

End Lab

Details

```
labsuser@master:~$ sudo kubeadm init
10:20:10:27:26.174843 8903 version.go:255] remote version is much newer: v1.26.1; falling back to: stable-1.23
[init] Using Kubernetes version: v1.23.16
[preflight] Running pre-flight checks
[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 [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local master.example.com] and IPs [10.96.0.1 172.31.54.228]
[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
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [localhost master.example.com] and IPs [172.31.54.228 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [localhost master.example.com] and IPs [172.31.54.228 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
```

Azure Resourcegroup

Kubernetes New Version Lab

master worker1 worker2

Used 0.5 of 50 hours in Feb, 2023

Start Lab

End Lab

Details

```
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [localhost master.example.com] and IPs [172.31.54.228 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [localhost master.example.com] and IPs [172.31.54.228 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
```

Azure Resourcegroup

Kubernetes New Version Lab

master worker1 worker2

Used 0.5 of 50 hours in Feb, 2023

Start Lab

End Lab

Details

```
Your Kubernetes control-plane 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

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

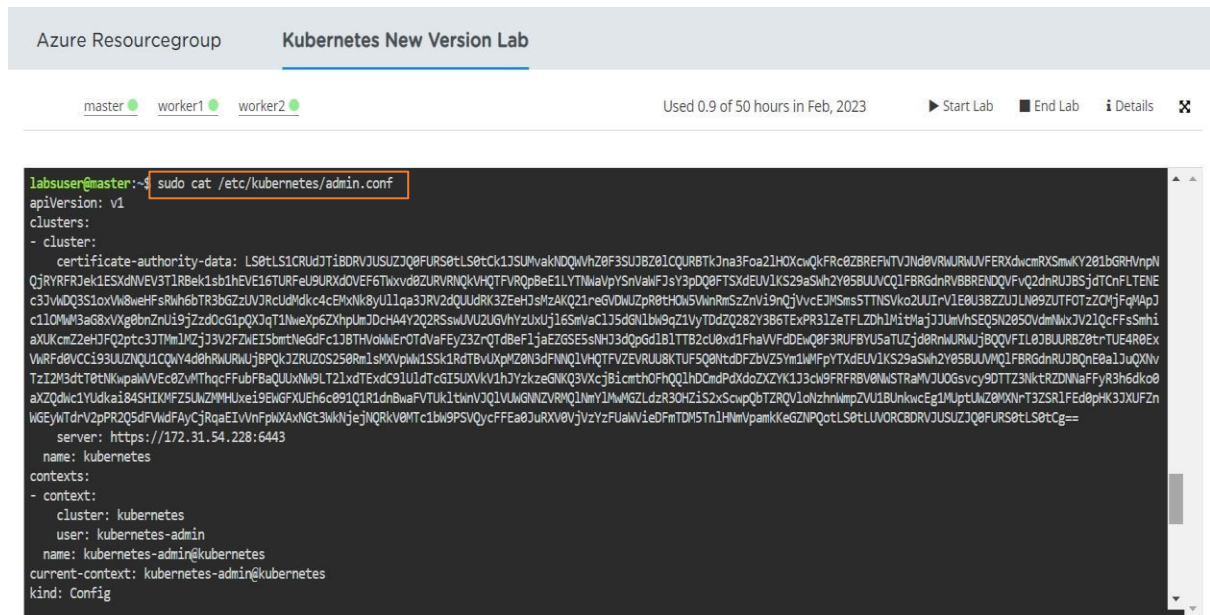
You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

kubeadm join 172.31.54.228:6443 --token 8xs07.1ewm6sc1nwcgboh \
--discovery-token-ca-cert-hash sha256:1b7f71b22185f3799696a7551e5f5d627a510bf3d86998554791dfafbee8388a
labsuser@master:~$
```

OUTPUT: Prompts “your Kubernetes control-plane has initialized successfully!”

5. Checking API server IP address and port number using command “sudo cat /etc/Kubernetes/admin.conf” (Fetching from configuration file).



The screenshot shows the Azure Resource Group console for 'Kubernetes New Version Lab'. It displays three nodes: master, worker1, and worker2. The master node is selected, and the command 'sudo cat /etc/kubernetes/admin.conf' is executed. The output shows the Kubernetes configuration, including the API server address and port.

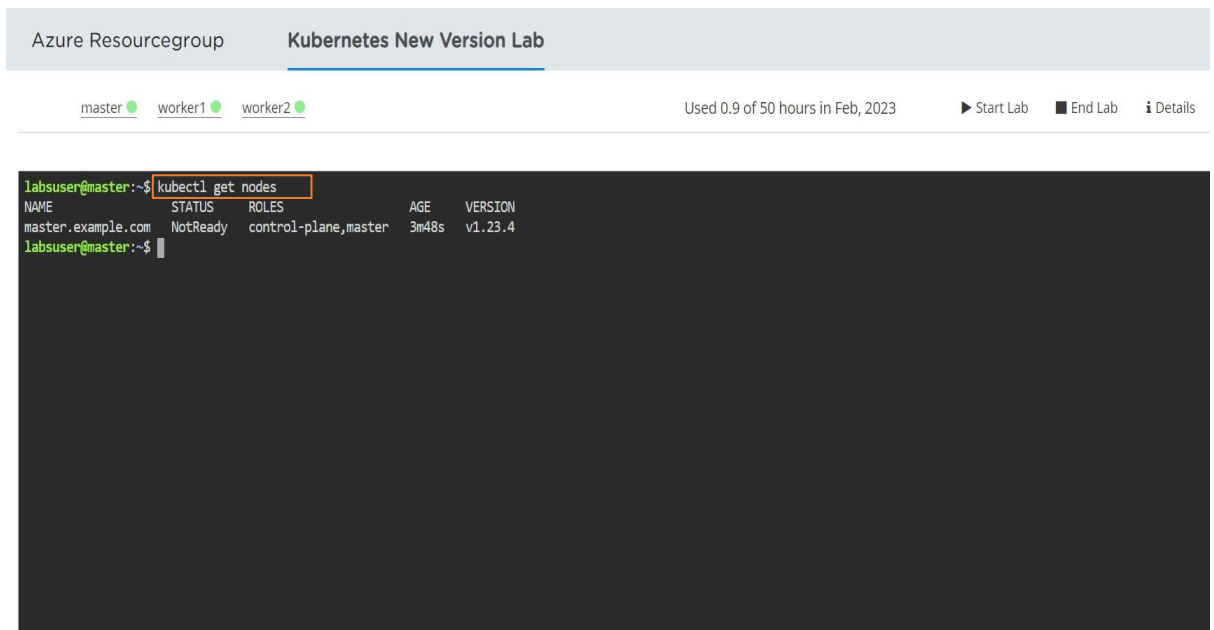
```
labsuser@master:~$ sudo cat /etc/kubernetes/admin.conf
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiB0dXJUSUJzQ0FURSB0tLS0tCk1JSUMvakhQWWhZ0F3SUJBZ0lCQURBTk3na3Foa2lHdXcwQkFRc0ZBREFlVTV3d0VRWURWUFRXdwcmRXSmwKY201bGRHVnpN
    QjRyRFRJek1ESXdlMVEVSTlRbek1sb1hEVEI6TURFeU9URXdoVEF6Tlxxvd8ZURVRNQRVHQTFRQpBeE1LYTNMaVpYSnVhVWJ3Y3pDQ0FTSXdEUVlKS29aSWZhY05BUUVCQlFBRGdnRVBBRNDQVlVQ2dnRUJBSjdtCnFLTENE
    c3JvMQ3S1oxVW8weHFsRWh6bTR3bGZzUVRjRldmMk44cEMxNk8yU1lqa3JRV2dQUUdRk3ZEhHs3ZAKQ21reGVdWUJpR0tHOM5VWnRmSzZnV19nQjVvcEJMSes5TTNSVko2UUIrV1E0U3BZZUJlN09ZUTFTOTzChJfQdMqApJ
    c1lQWMM3aG8xVWg0bnZnU19jZzd0G1pQXQjTlNweXp6ZkhpUmJ0dChA4Y2Q2R5swUJ2UGVhYzUxUj16SmVaC1J3dGVLbW9qZ1VvTDZQ282Y3B6TEpPR3lZeTFLZDhJMiThMajJUmVhSEQ5N205OVdmMw3V2lQcFFsSmhi
    aXUKcmZ2eHJFQ2ptc3JlTm1WZj3V2F2NlEISbmtNeGdFc1J0THVWolWlErOTdVaFyZ3ZrQTdBeF1jaE2GSE5sNHJ3dGp6d1B1TT82clU8xd1FhaVWFd0EwQ0F3RUF8YU5aTUZjd08nMURWUjB0QVFL0JBUUR8Z0trTUE4R0Ex
    VmRf0VCCi93UUNQZlU1CQWY4d0hRWURWUjBPQkJZRUZOS250Rm1sMkVpMw1SSk1RdtBVUkplZ0N3dFNQ1VHQTFRVZVUURU8KUF5Q8NtdDFZbVZ5Ym1MFPYTYxdEUVlKS29aSWZhY05BUUVCQlFBRGdnRVB0QnE0a1JuQXlv
    TzI2N3dtT0tNkpaWVVe0ZVMTgqFFubF8aQUUxW9LT2lxdTExdC9lUldTcG1SUUxVXV1h3YzZkeGNKQ3VXc3JlcmthOFhQ1hDcmRpDkd0ZVYK1J3cl9FRFRB0W5TRaMvJU0Gsvcy9DTT3NktRZDNlNaFFyR3h6dk00
    aXZQdmc1YUdka184SHIKFZ5UWMMHUXei9EwGFUXEH6c091Q1R1dnBwaFVUk1tWnVjQ1VUdGNNZVRhQ1NwY1MwMGZLdzR3OHZiS2xScwpQbTZRQV10nzhnWpZVU1BUnkwcEg1MUpTlUW0W0N-T3ZSR1FEd0pK3JXUFZn
    W6EYwtdrV2pPR2Q5dFVmdFayCjRqaEIVnFpKAXwGt3WkNjeJNQRKv8MTc1bW9PSVQycFFEa0JuRXV0VjVzYzYUaWVieDFmTDN5TnJHNMvpamkKegZNPQotLS0tLUVORC0RDRVJUSUJzQ0FURSB0tLS0tCg==
  server: https://172.31.54.228:6443
name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
```

OUTPUT: Prompts API server address as “<https://172.31.54.228:6443>”. The port number is “6443”

6. Changing root permissions.

```
labsuser@master:~$ sudo mkdir -p $HOME/.kube
labsuser@master:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
labsuser@master:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

7. Checking either master node is set – up using command “kubectl get nodes”.



The screenshot shows a terminal window within a lab environment titled "Kubernetes New Version Lab". The terminal displays the command `kubectl get nodes` and its output. The output is a table with columns: NAME, STATUS, ROLES, AGE, and VERSION. The table contains one entry: `master.example.com` with status `NotReady`, roles `control-plane,master`, age `3m48s`, and version `v1.23.4`. The terminal prompt is `labsuser@master:~$`.

NAME	STATUS	ROLES	AGE	VERSION
master.example.com	NotReady	control-plane,master	3m48s	v1.23.4

OUTPUT: The master node has been set successfully! The status reflects to be not ready as the worker nodes are to be connected.