

Nov 13 17:34 • root@ip-172-31-3-86:/etc/yum.repos.d

```
[kubelet] Creating a ConfigMap "kubelet-config-1.22" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node master as control-plane by adding the labels: [node-role.kubernetes.io/master(deprecated) node-role.kubernetes.io/control-plane.node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node master as control-plane by adding the taints [node-role.kubernetes.io/master:NoSchedule]
[bootstrap-token] Using token: dzs9nc.1uw7cg9cyl4ylc6s
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] configured RBAC rules to allow CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

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 172.31.3.86:6443 --token dzs9nc.1uw7cg9cyl4ylc6s \
--discovery-token-ca-cert-hash sha256:43be655468d5a6a556508b86945f0fefc79270e27a1702ef7e9441c5e1df536e
[root@master yum.repos.d]# ]
```

Nov 13 17:49 • root@master:/home/centos

```
root@master:~# kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
master         Ready    control-plane,master   14m   v1.22.3
workernode1   Ready    <none>     5m28s  v1.22.3
workernode2   Ready    <none>     50s   v1.22.3
root@master:~# ]
```

```
Nov 13 17:51 • root@workernode1:/etc/yum.repos.d
root@workernode2:/etc/yum.repos.d
root@master:yamls

apiVersion: v1
kind: Pod
metadata:
  name: pod1
  labels:
    disk: ssdtype
    dish: mango
spec:
  containers:
    - name: c1
      image: docker.io/httpd
      ports:
        - containerPort: 80
```

```
Nov 13 17:52 • root@master:yamls

[root@master centos]# kubectl get nodes
NAME     STATUS   ROLES      AGE     VERSION
master   Ready    control-plane,master   16m    v1.22.3
workernode1 Ready    <none>    7m20s   v1.22.3
workernode2 Ready    <none>    2m42s   v1.22.3
[root@master centos]# mkdir /yamls
[root@master centos]# cd /yamls/
[root@master yamls]# vi pod1.yaml
[root@master yamls]# kubectl explain pod
KIND:   Pod
VERSION: v1

DESCRIPTION:
Pod is a collection of containers that can run on a host. This resource is
created by clients and scheduled onto hosts.

FIELDS:
apiVersion <string>
  APIVersion defines the versioned schema of this representation of an
  object. Servers should convert recognized schemas to the latest internal
  value, and may reject unrecognized values. More info:
  https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

kind <string>
  Kind is a string value representing the REST resource this object
  represents. Servers may infer this from the endpoint the client submits
  requests to. Cannot be updated. In CamelCase. More info:
  https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

metadata <Object>
  Standard object's metadata. More info:
  https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

spec <Object>
  Specification of the desired behavior of the pod. More info:
```

```
Activities Terminal Nov 13 17:53
root@workernode1:/etc/yum.repos.d root@master:/yamls
root@workernode2:/etc/yum.repos.d root@master:/yamls

created by clients and scheduled onto hosts.

FIELDS:
apiVersion <string>
  APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info:
  https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

kind <string>
  Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:
  https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

metadata <Object>
  Standard object's metadata. More info:
  https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

spec <Object>
  Specification of the desired behavior of the pod. More info:
  https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

status <Object>
  Most recently observed status of the pod. This data may not be up to date.
  Populated by the system. Read-only. More info:
  https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

[root@master yamls]# kubectl create -f pod1.yaml
pod/pod1 created
[root@master yamls]# kubectl get pod
NAME READY STATUS RESTARTS AGE
pod1 1/1 Running 0 12s
[root@master yamls]# kubectl get pod -o wide
NAME READY STATUS RESTARTS AGE IP NOMINATED NODE READINESS GATES
pod1 1/1 Running 0 19s 192.168.216.65 workernode2 <none> <none>
[root@master yamls]#
```

```
Activities Terminal Nov 13 17:54 root@master:/yaml
root@workernode1:/etc/yum.repos.d root@workernode2:/etc/yum.repos.d root@master:/yaml

APIVersion defines the versioned schema of this representation of an
object. Servers should convert recognized schemas to the latest internal
value, and may reject unrecognized values. More info:
https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

kind <string>
Kind is a string value representing the REST resource this object
represents. Servers may infer this from the endpoint the client submits
requests to. Cannot be updated. In CamelCase. More info:
https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

metadata <Object>
Standard object's metadata. More info:
https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

spec <Object>
Specification of the desired behavior of the pod. More info:
https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

status <Object>
Most recently observed status of the pod. This data may not be up to date.
Populated by the system. Read-only. More info:
https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

[root@master yaml]# kubectl create -f pod1.yaml
pod/pod1 created
[root@master yaml]# kubectl get pod
NAME READY STATUS RESTARTS AGE
pod1 1/1 Running 0 12s
[root@master yaml]# kubectl get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
pod1 1/1 Running 0 19s 192.168.216.65 workernode2 <none> <none>
[root@master yaml]# kubectl run pod2 --image=httpd
pod/pod2 created
[root@master yaml]# kubectl run pod3 --image=nginx
pod/pod3 created
[root@master yaml]#
```

```
Activities Terminal Nov 13 17:56 •
root@workernode1:/etc/yum.repos.d root@workernode2:/etc/yum.repos.d root@master/yamls
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  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
```

```
Activities Terminal Nov 13 17:56 •
root@workernode1:/etc/yum.repos.d root@workernode2:/etc/yum.repos.d root@master/yamls
pod1  1/1    Running   0       117s  192.168.216.65  workernode2  <none>      <none>
pod2  1/1    Running   0       32s   192.168.212.1   workernode1  <none>      <none>
pod3  1/1    Running   0       19s   192.168.216.66  workernode2  <none>      <none>
[root@master yamls]# ls
pod1.yaml
[root@master yamls]# vi dep.yaml
[root@master yamls]# kubectl explain deployment
KIND:  Deployment
VERSION:  apps/v1
DESCRIPTION:
Deployment enables declarative updates for Pods and ReplicaSets.
FIELDS:
  apiVersion  <string>
    APIVersion defines the versioned schema of this representation of an
    object. Servers should convert recognized schemas to the latest internal
    value, and may reject unrecognized values. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
  kind <string>
    Kind is a string value representing the REST resource this object
    represents. Servers may infer this from the endpoint the client submits
    requests to. Cannot be updated. In CamelCase. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
  metadata  <Object>
    Standard object's metadata. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
  spec <Object>
    Specification of the desired behavior of the Deployment.
  status   <Object>
    Most recently observed status of the Deployment.
[root@master yamls]#
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

```
[root@master yamls]# vi dep1.yaml
[root@master yamls]# kubectl create -f dep1.yaml
deployment.apps/nginx-deployment created
[root@master yamls]# kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1     1           1          17s
[root@master yamls]# kubectl get deploy -o wide
NAME           READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES   SELECTOR
nginx-deployment   1/1     1           1          21s   nginx       nginx:1.14.2   app=nginx
[root@master yamls]#
```

Activities Terminal Nov 13 18:00 • root@master:yamls

```
[root@master yamls]# vi dep1.yaml
[root@master yamls]# kubectl create -f dep1.yaml
deployment.apps/nginx-deployment created
[root@master yamls]# kubectl get deploy
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 1/1     1          1          17s
[root@master yamls]# kubectl get deploy -o wide
NAME      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES   SELECTOR
nginx-deployment 1/1     1          1          21s   nginx        nginx:1.14.2   app=nginx
[root@master yamls]# kubectl create deployment app1 --image=httpd
deployment.apps/app1 created
[root@master yamls]# kubectl get deploy -o wide
NAME      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES   SELECTOR
app1     0/1     1          0          5s    httpd        httpd      app=app1
nginx-deployment 1/1     1          1          66s   nginx        nginx:1.14.2   app=nginx
[root@master yamls]#
```

Activities Terminal Nov 13 18:07 • root@master:yamls

```
[root@master yamls]# kubectl get pods --show-labels
NAME                           READY   STATUS    RESTARTS   AGE   LABELS
app1-7bf48cc986-r825g         1/1    Running   0          6m37s   app=app1,pod-template-hash=7bf48cc986
nginx-deployment-66b6c48dd5-7q5r4 1/1    Running   0          7m38s   app=nginx,pod-template-hash=66b6c48dd5
pod1                          1/1    Running   0          14m    dish=mango,disk=ssdtype
pod2                          1/1    Running   0          12m    run=pod2
pod3                          1/1    Running   0          12m    run=pod3
pod4                          1/1    Running   0          4m10s   superhero=ironman
pod6                          1/1    Running   0          2m21s   superhero=hulk
pod7                          1/1    Running   0          119s   superhero=ironman
[root@master yamls]# kubectl get deploy --show-labels
NAME      READY   UP-TO-DATE   AVAILABLE   AGE   LABELS
app1     1/1     1          1          6m54s   app=app1
nginx-deployment 1/1     1          1          7m55s   app=nginx
[root@master yamls]#
```

```

Nov 13 18:10 • root@workernode1:/etc/yum.repos.d
root@workernode2:/etc/yum.repos.d
root@master/yamls

[root@master yamls]# kubectl scale deployment/app1 --replicas=15
deployment.apps/app1 scaled
[root@master yamls]# kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE      NOMINATED NODE   READINESS GATES
app1-7bf48cc986-45t2p   1/1    Running   0          97s    192.168.212.7  workernode1  <none>        <none>
app1-7bf48cc986-4pzrg   1/1    Running   0          98s    192.168.212.8  workernode1  <none>        <none>
app1-7bf48cc986-4xssq   1/1    Running   0          97s    192.168.212.9  workernode1  <none>        <none>
app1-7bf48cc986-7rvl2   1/1    Running   0          97s    192.168.212.6  workernode1  <none>        <none>
app1-7bf48cc986-86nkw   1/1    Running   0          97s    192.168.212.13  workernode1 <none>        <none>
app1-7bf48cc986-9cxtn   0/1    ContainerCreating   0          5s     <none>       workernode1  <none>        <none>
app1-7bf48cc986-bcsbw   0/1    ContainerCreating   0          5s     <none>       workernode1  <none>        <none>
app1-7bf48cc986-bh4d4   1/1    Running   0          97s    192.168.212.14  workernode1  <none>        <none>
app1-7bf48cc986-bjv6k   0/1    ContainerCreating   0          5s     <none>       workernode1  <none>        <none>
app1-7bf48cc986-fp4tj   0/1    ContainerCreating   0          5s     <none>       workernode1  <none>        <none>
app1-7bf48cc986-nzw59   1/1    Running   0          97s    192.168.212.12  workernode1  <none>        <none>
app1-7bf48cc986-p49gs   0/1    ContainerCreating   0          5s     <none>       workernode1  <none>        <none>
app1-7bf48cc986-r825g   1/1    Running   0          9m55s   192.168.212.3  workernode1 <none>        <none>
app1-7bf48cc986-xgww4   1/1    Running   0          97s    192.168.212.11  workernode1  <none>        <none>
nginx-deployment-66b6c48dd5-7q5r4  1/1    Running   0          10m   192.168.212.2  workernode1  <none>        <none>
pod1                  1/1    Running   0          17m   192.168.216.65  workernode2  <none>        <none>
pod2                  1/1    Running   0          16m   192.168.212.1  workernode1  <none>        <none>
pod3                  1/1    Running   0          15m   192.168.216.66  workernode2  <none>        <none>
pod4                  1/1    Running   0          7m28s  192.168.216.67  workernode2  <none>        <none>
pod6                  1/1    Running   0          5m39s  192.168.212.4  workernode1 <none>        <none>
pod7                  1/1    Running   0          5m17s  192.168.212.5  workernode1 <none>        <none>
[root@master yamls]# kubectl get nodes
NAME      STATUS   ROLES      AGE   VERSION
master    Ready    control-plane,master  36m  v1.22.3
workernode1 Ready    <none>      27m  v1.22.3
workernode2 Ready    <none>      22m  v1.22.3
[root@master yamls]#

```

```

Nov 13 18:11 • root@workernode1:/etc/yum.repos.d
root@workernode2:/etc/yum.repos.d
root@master/yamls

root@workernode1:/etc/yum.repos.d
root@workernode2:/etc/yum.repos.d
root@master/yamls

[root@master yamls]# kubectl scale deployment/app1 --replicas=1
deployment.apps/app1 scaled
[root@master yamls]# kubectl get nodes
NAME      STATUS   ROLES      AGE   VERSION
master    Ready    control-plane,master  36m  v1.22.3
workernode1 Ready    <none>      27m  v1.22.3
workernode2 Ready    <none>      22m  v1.22.3
[root@master yamls]# kubectl scale deployment/app1 --replicas=1
deployment.apps/app1 scaled
[root@master yamls]# kubectl get nodes
NAME      STATUS   ROLES      AGE   VERSION
master    Ready    control-plane,master  36m  v1.22.3
workernode1 Ready    <none>      27m  v1.22.3
workernode2 Ready    <none>      22m  v1.22.3
[root@master yamls]# kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE      NOMINATED NODE   READINESS GATES
app1-7bf48cc986-bh4d4   1/1    Running   0          10m   192.168.212.14  workernode1  <none>        <none>
app1-7bf48cc986-bjv6k   0/1    ContainerCreating   0          5s     <none>       workernode1  <none>        <none>
app1-7bf48cc986-fp4tj   0/1    ContainerCreating   0          5s     <none>       workernode1  <none>        <none>
app1-7bf48cc986-nzw59   1/1    Running   0          97s    192.168.212.12  workernode1  <none>        <none>
app1-7bf48cc986-p49gs   0/1    ContainerCreating   0          5s     <none>       workernode1  <none>        <none>
app1-7bf48cc986-r825g   1/1    Running   0          9m55s   192.168.212.3  workernode1 <none>        <none>
app1-7bf48cc986-xgww4   1/1    Running   0          97s    192.168.212.11  workernode1  <none>        <none>
nginx-deployment-66b6c48dd5-7q5r4  1/1    Running   0          10m   192.168.212.2  workernode1  <none>        <none>
pod1                  1/1    Running   0          17m   192.168.216.65  workernode2  <none>        <none>
pod2                  1/1    Running   0          16m   192.168.212.1  workernode1  <none>        <none>
pod3                  1/1    Running   0          15m   192.168.216.66  workernode2  <none>        <none>
pod4                  1/1    Running   0          7m28s  192.168.216.67  workernode2  <none>        <none>
pod6                  1/1    Running   0          5m39s  192.168.212.4  workernode1 <none>        <none>
pod7                  1/1    Running   0          5m17s  192.168.212.5  workernode1 <none>        <none>
[root@master yamls]#

```

```
Nov 13 18:13 • root@master:yamls
root@workernode1:/etc/yum.repos.d
root@workernode2:/etc/yum.repos.d
root@master:yamls

[root@master yamls]# kubectl get nodes
NAME      STATUS   ROLES      AGE   VERSION
master    Ready    control-plane,master 36m   v1.22.3
workernode1 Ready    <none>     27m   v1.22.3
workernode2 Ready    <none>     22m   v1.22.3

[root@master yamls]# kubectl scale deployment/app1 --replicas=1
deployment.apps/app1 scaled

[root@master yamls]# kubectl get nodes
NAME      STATUS   ROLES      AGE   VERSION
master    Ready    control-plane,master 36m   v1.22.3
workernode1 Ready    <none>     27m   v1.22.3
workernode2 Ready    <none>     22m   v1.22.3

[root@master yamls]# kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE   NOMINATED NODE   READINESS GATES
app1-7bf48cc986-r825g  1/1    Running   0          10m  192.168.212.3  workernode1  <none>        <none>
nginx-deployment-66b6c48dd5-7q5r4  1/1    Running   0          11m  192.168.212.2  workernode2  <none>        <none>
pod1          1/1    Running   0          18m  192.168.216.65  workernode2  <none>        <none>
pod2          1/1    Running   0          17m  192.168.212.1  workernode1  <none>        <none>
pod3          1/1    Running   0          16m  192.168.216.66  workernode2  <none>        <none>
pod4          1/1    Running   0          8m25s 192.168.216.67  workernode2  <none>        <none>
pod6          1/1    Running   0          6m36s 192.168.212.4  workernode1  <none>        <none>
pod7          1/1    Running   0          6m14s 192.168.212.5  workernode1  <none>        <none>

[root@master yamls]# kubectl delete pod app1-7bf48cc986-r825g
pod "app1-7bf48cc986-r825g" deleted
[root@master yamls]# kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE   NOMINATED NODE   READINESS GATES
app1-7bf48cc986-8qdd8  1/1    Running   0          5s   192.168.212.20  workernode1  <none>        <none>
nginx-deployment-66b6c48dd5-7q5r4  1/1    Running   0          13m  192.168.212.2  workernode1  <none>        <none>
pod1          1/1    Running   0          20m  192.168.216.65  workernode2  <none>        <none>
pod2          1/1    Running   0          18m  192.168.212.1  workernode1  <none>        <none>
pod3          1/1    Running   0          18m  192.168.216.66  workernode2  <none>        <none>
pod4          1/1    Running   0          10m  192.168.216.67  workernode2  <none>        <none>
pod6          1/1    Running   0          8m14s 192.168.212.4  workernode1  <none>        <none>
pod7          1/1    Running   0          7m52s 192.168.212.5  workernode1  <none>        <none>

[root@master yamls]#
```

```
Nov 13 18:17 • root@master:yamls
root@workernode1:/etc/yum.repos.d
root@workernode2:/etc/yum.repos.d
root@master:yamls

[root@master yamls]# kubectl create namespace google
namespace/google created
[root@master yamls]# kubectl create deployment project1 -n google --image=httpd
deployment.apps/project1 created
[root@master yamls]# kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE   NOMINATED NODE   READINESS GATES
app1-7bf48cc986-8qdd8  1/1    Running   0          3m14s 192.168.212.20  workernode1  <none>        <none>
nginx-deployment-66b6c48dd5-7q5r4  1/1    Running   0          16m  192.168.212.2  workernode1  <none>        <none>
pod1          1/1    Running   0          23m  192.168.216.65  workernode2  <none>        <none>
pod2          1/1    Running   0          21m  192.168.212.1  workernode1  <none>        <none>
pod3          1/1    Running   0          21m  192.168.216.66  workernode2  <none>        <none>
pod4          1/1    Running   0          13m  192.168.216.67  workernode2  <none>        <none>
pod6          1/1    Running   0          11m  192.168.212.4  workernode1  <none>        <none>
pod7          1/1    Running   0          11m  192.168.212.5  workernode1  <none>        <none>

[root@master yamls]# kubectl get deploy -o wide
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES       SELECTOR
app1          1/1     1           1           15m   httpd        httpd        app=app1
nginx-deployment  1/1     1           1           17m   nginx        nginx:1.14.2  app=nginx
[root@master yamls]# kubectl get pod -n google
NAME          READY   STATUS    RESTARTS   AGE
project1-69b7985cb-dlmhw  1/1    Running   0          103s
[root@master yamls]#
```

Activities Terminal Nov 13 18:20 • root@master/yamls

```
root@workernode1:/etc/yum.repos.d          root@workernode2:/etc/yum.repos.d          root@master:yamls
NAME      READY   STATUS    RESTARTS   AGE     IP           NODE   NOMINATED-NODE  READINESS GATES
app1-7bf48cc986-8gqd8  1/1    Running   0          3m14s  192.168.212.20  workernode1  <none>        <none>
nginx-deployment-66b6c48dd5-7q5r4  1/1    Running   0          16m    192.168.212.2  workernode1  <none>        <none>
pod1      1/1    Running   0          23m    192.168.216.65  workernode2  <none>        <none>
pod2      1/1    Running   0          21m    192.168.212.1   workernode1  <none>        <none>
pod3      1/1    Running   0          21m    192.168.216.66  workernode2  <none>        <none>
pod4      1/1    Running   0          13m    192.168.216.67  workernode2  <none>        <none>
pod6      1/1    Running   0          11m    192.168.212.4   workernode1  <none>        <none>
pod7      1/1    Running   0          11m    192.168.212.5   workernode1  <none>        <none>

[root@master yamls]# kubectl get deploy -o wide
NAME          READY  UP-TO-DATE AVAILABLE   AGE   CONTAINERS   IMAGES       SELECTOR
app1          1/1    1            1          15m   httpd        httpd       app=app1
nginx-deployment 1/1    1            1          17m   nginx        nginx:1.14.2  app=nginx
[root@master yamls]# kubectl get pod -n google
NAME      READY   STATUS    RESTARTS   AGE
project1-69b7985cb-dlmhw  1/1    Running   0          103s

[root@master yamls]# kubectl exec -it project1-69b7985cb-dlmhw bash -n google
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
root@project1-69b7985cb-dlmhw:/usr/local/apache2# ls
bin build cgi-bin conf error htdocs icons include logs modules
root@project1-69b7985cb-dlmhw:/usr/local/apache2# cd htdocs/
root@project1-69b7985cb-dlmhw:/usr/local/apache2/htdocs# ls
index.html
root@project1-69b7985cb-dlmhw:/usr/local/apache2/htdocs# cat > index.html
*****
**Hi This IS Rakesh kumar singh*****
*****
this is assem purpose only*****
*****
^C
root@project1-69b7985cb-dlmhw:/usr/local/apache2/htdocs# exit
exit
command terminated with exit code 130
[root@master yamls]# kubectl expose deployment/project1 --type=NodePort --port=80 -n google
service/project1 exposed
[root@master yamls]# kubectl get svc -n google
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
project1  NodePort  10.105.200.57  <none>          80:32330/TCP  19s
```

Nov 13 18:20

Instances | EC2 Manager | Connect to instance | EC2 | Connect to instance | EC2 | Connect to instance | EC2 | Deployments | Kubernetes | Assigning Pods to Nodes | +

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#instances:

Rakesh Kumar Singh Mumbai Support

Activities Google Chrome

Instances (1/3) Info

Search for services, features, blogs, docs, and more [Alt+S]

Launch instances

New EC2 Experience Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Capacity Reservations

Images AMIs

Elastic Block Store

Feedback English (US)

Instances (1/3)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
workernode1	i-0855439454bd2e886	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1b	ec2-65-1-107-135.ap...	65.1.107.135	-
workernode2	i-0fe5d160d35fba03b	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1b	ec2-13-233-199-56.ap...	13.233.199.56	-
master	i-00c97256cb63ea4b8	Running	t2.medium	2/2 checks passed	No alarms	+ ap-south-1b	ec2-3-110-41-100.ap...	3.110.41.100	-

Instance: i-00c97256cb63ea4b8 (master)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Public IPv4 address copied

Instance ID: i-00c97256cb63ea4b8 (master)

IPv6 address: -

Private IPv4 DNS: 172.31.3.86.ap-south-1.compute.internal

Public IPv4 address: 3.110.41.100 | open address

Instance state: Running

Instance type: t2.medium

Private IPv4 addresses: 172.31.3.86

Public IPv4 DNS: ec2-3-110-41-100.ap-south-1.compute.amazonaws.com | open address

Elastic IP addresses:

© 2008-2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Nov 13 18:20

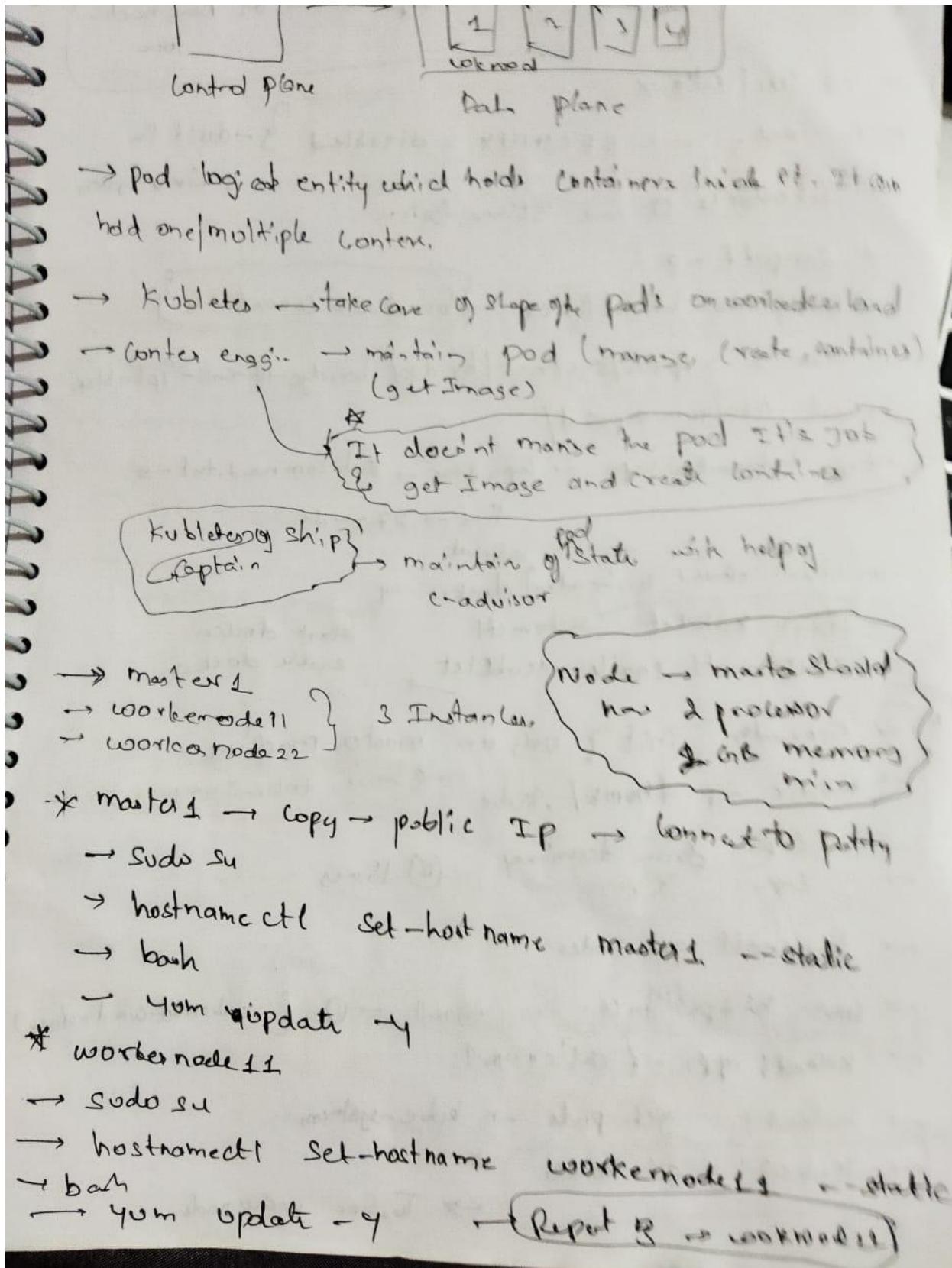
Instances | EC2 Manager | Connect to instance | EC2 | Connect to instance | EC2 | Connect to instance | EC2 | Deployments | Kubernetes | Assigning Pods to Node | 3.110.41.100:32330 | + | Reading list

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#instances:

Rakesh Kumar Singh Mumbai Support

Activities Google Chrome

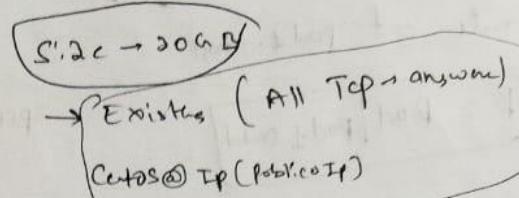
\*\*\*\*\*HI THIS IS Rakesh kumar singh\*\*\*\*\* this is assment purpose only\*\*\*\*\*



- locally
- cd /etc/selinux
  - vi config → SELINUX = disabled } → do it in  
3 Linux system
  - Setenforce 0 → 3 Linux system
  - swapoff -a
  - modprobe br\_netfilter → {for network connectivity}
  - echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
  - cd /etc/yum.repos.d/
  - vi kubernetes.repo → copy paste → kube.command.txt →  
line → 27 → 34
  - yum install kubelet docker →
  - start kubelet → systemctl
  - systemctl enable kubelet → start docker  
enable docker
  - master
    - kubeadm init } only on master machine
    - mkdir -p \$HOME/.kube → Save token somewhere
    - copy from terminal ⑧ lines
    - copy ..
    - kubectl get nodes
    - curl https://docs...yaml -O [kubernetes.yaml]
    - kubectl apply -f calico.yaml
    - kubectl get pods -n kube-system
    - kubectl get nodes \*
  - To see work node

- On worker node → uses port → token (kubeadm join ...)
- get nodes

→ machine → control



→ master → 212

→ worker1 → 250

→ worker2 → 33

→ yum install kubectl kubeadm docker →

### Pod Creating Methods

Lured now 3

Imperative  
method

declarative  
method

→ declarative method +

master node

→ mkdir /yaml

→ ed (yaml) → kubectl get pods -n kube-system

→ vi pod1.yaml

→ kubectl explain pod (for Charles version)  
(Open duplicate session to  
create if you want)

→ N<sub>i</sub> pod.yaml → Copy File  
from Pod.txt file

→ kubectl

Create -f pod.yaml

[Output: Pod/pod1 created]

→ kubectl get pod

→ kubectl get pod -o wide

→ [Create another yaml Pod] → change name: pod2 → same file

→ kubectl create -f pod2.yaml

→ kubectl get pod -o wide

→ kubectl run pod3 --image=httpd  
→ kubectl run pod4 --image=nginx

Imparative method

→ kubectl get pod -o wide

→ kubectl exec -it pod4 bash

→ [To Nodify → Enter to that particular Linux Worknode]

In worknode { To check whether docker/pod is working

→ kubectl delete pod pod4

→ docker ps → worknode

→ master

→ kubectl exec -it pod3 bash [master]  
→ docker ps | grep -i pod2 [worker node] [From here]  
→ docker exec -it k8s-c1-pod2-default[<sup>copy id</sup> ~~or get~~] bash  
→ \* kubectl run pods --image=httpd --dry-run -o yaml [master]  
→ vi cd /yaml  
→ vi pod5.yaml  
→ kubectl create -f pod5.yaml

Note: controller manager is the reason for auto scaling and auto healing of pod - using deployment

→ Replace → in yaml file

+ deployment command kind: pod → deployment: pod

→ kubectl create deployment app --image=httpd

→ kubectl get pod

→ kubectl get deploy {shows deployment app (pod)}

→ kubectl exec -it <app-d...> bash

→ Err!

→ kubectl delete pod <app-d...> [d]

→ Check In duplicate session and check there still deleted or not

→ It again creates automatically

(because of controller manager pod / deployment)

→ because of auto healing nature

→ kubectl delete deployment app

To delete the deployment

### → [for scaling] scale up

→ kubectl scale deployment/app --replicas=40

→ de-scaling (scale down)

→ kubectl scale deployment/app --replicas=5

→ kubectl run deployment/app2

### \* NAMES \*

→ kubectl get ns

→ kubectl create namespace google

→ kubectl create deployment proj1 -n google  
--image=httpd

→ kubectl get pods -n google [To see if created]

→ kubectl exec -it <pod> bash -n google

→

→ Default load balancer have to give public IP \*

[which is not secure method]

→ to loadbalancer without Linux IP

→ kubectl expose deployment/proj1 -n google --type=loadBalancer  
port=80 -n google

