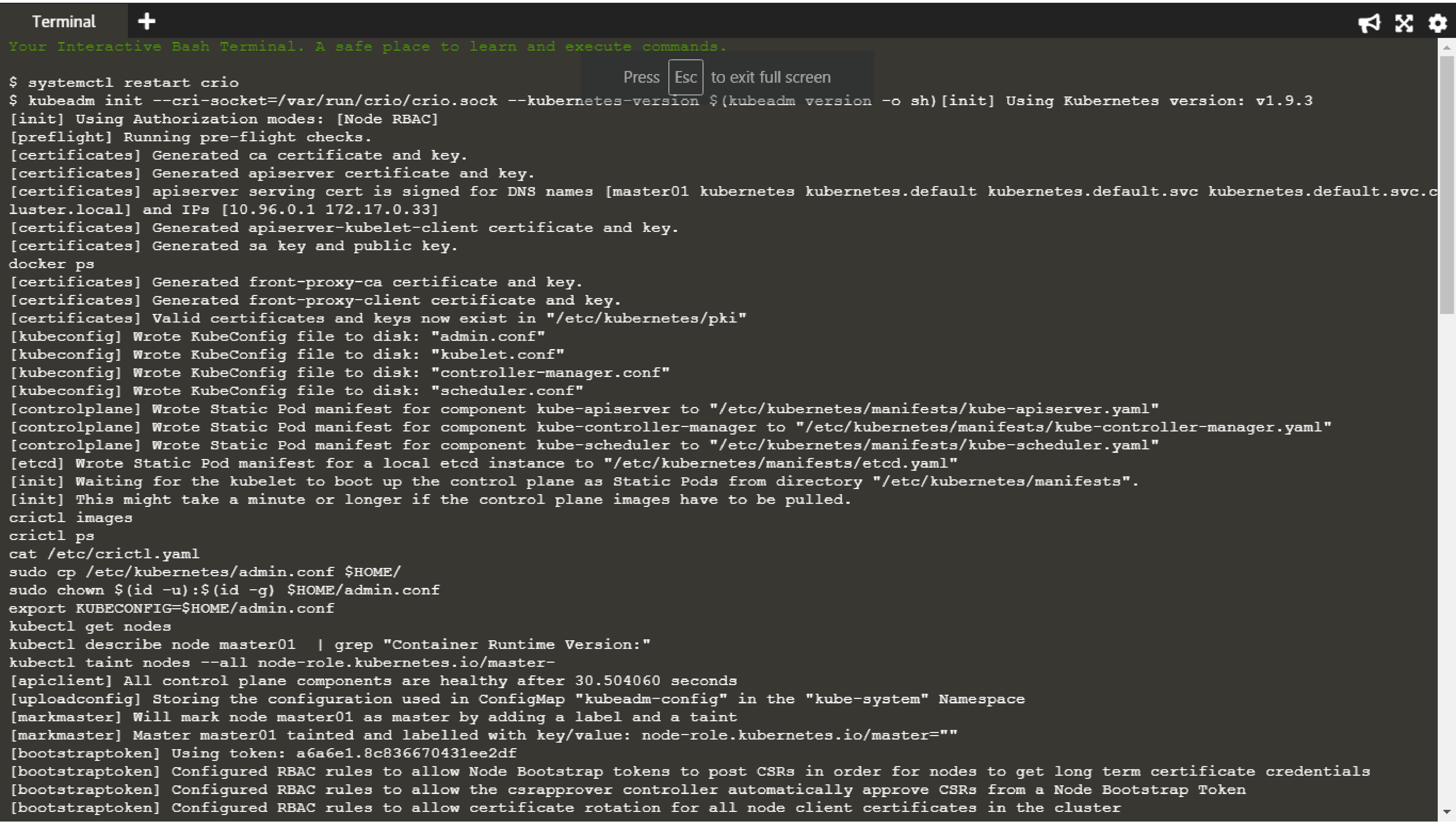
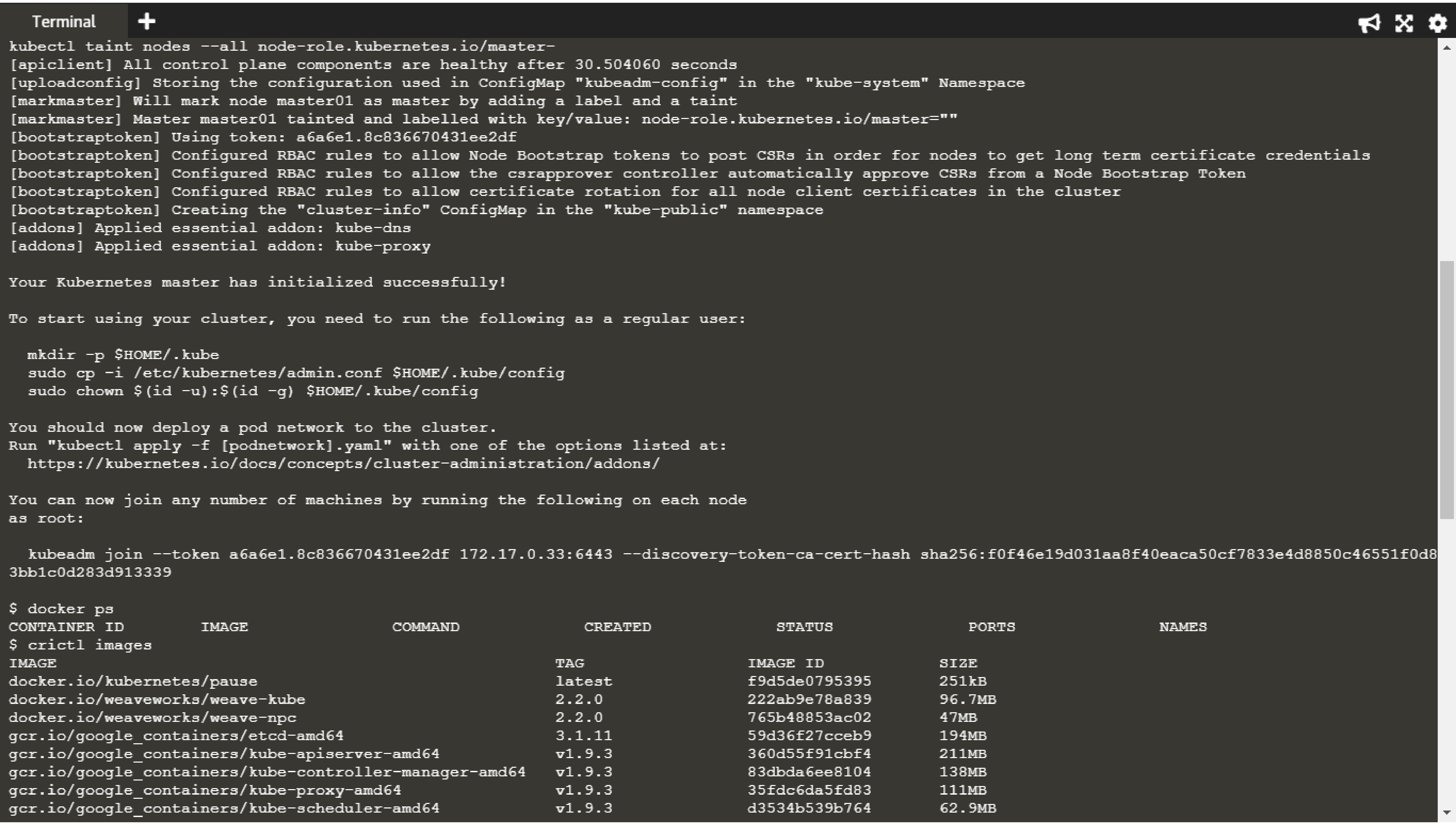


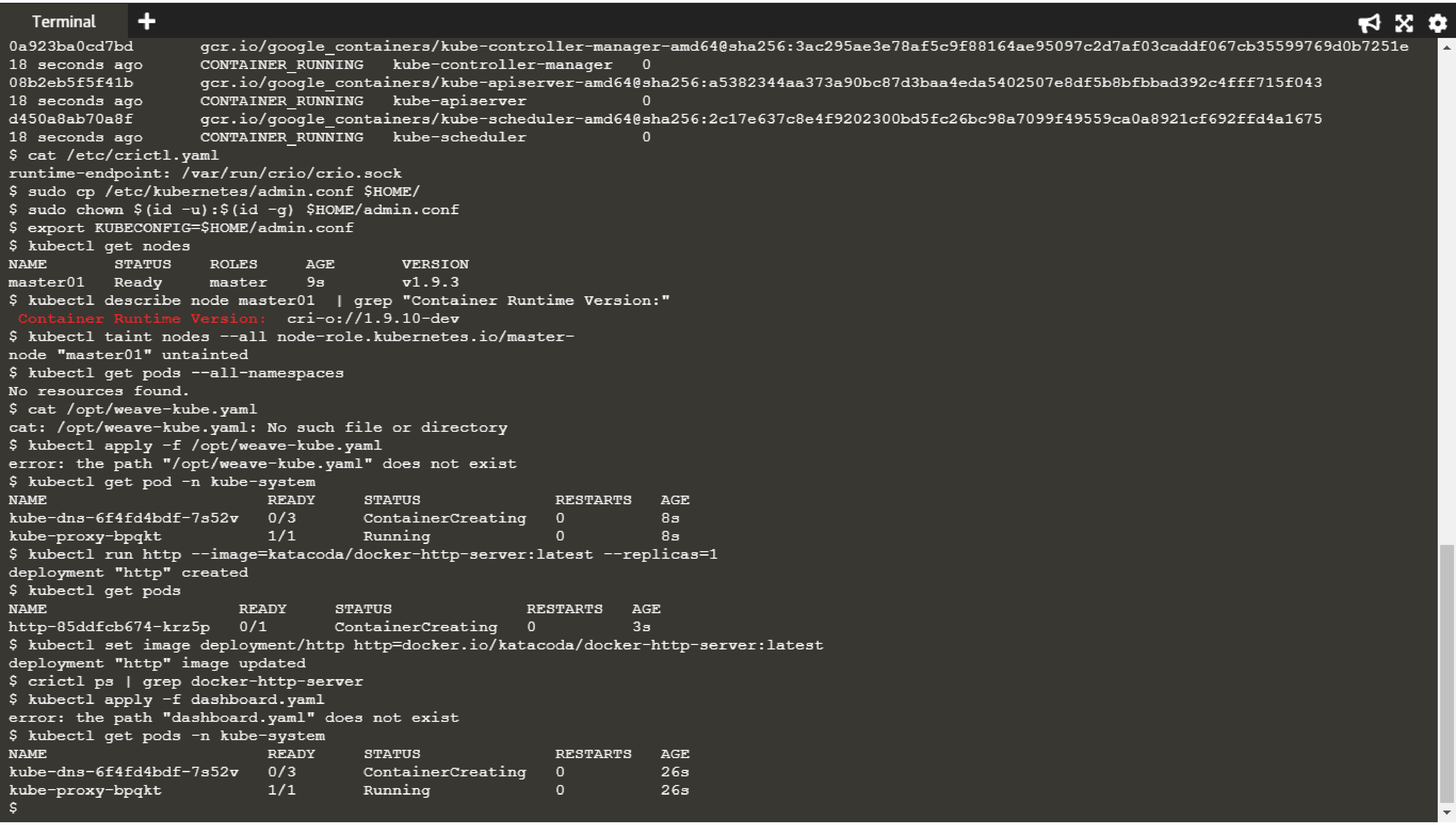
Terminal

Press Esc to exit full screen

```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ systemctl restart crio
$ kubeadm init --cri-socket=/var/run/crio/crio.sock --kubernetes-version $(kubeadm version -o sh)[init] Using Kubernetes version: v1.9.3
[init] Using Authorization modes: [Node RBAC]
[preflight] Running pre-flight checks.
[certificates] Generated ca certificate and key.
[certificates] Generated apiserver certificate and key.
[certificates] apiserver serving cert is signed for DNS names [master01 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.c
luster.local] and IPs [10.96.0.1 172.17.0.33]
[certificates] Generated apiserver-kubelet-client certificate and key.
[certificates] Generated sa key and public key.
docker ps
[certificates] Generated front-proxy-ca certificate and key.
[certificates] Generated front-proxy-client certificate and key.
[certificates] Valid certificates and keys now exist in "/etc/kubernetes/pki"
[kubeconfig] Wrote KubeConfig file to disk: "admin.conf"
[kubeconfig] Wrote KubeConfig file to disk: "kubelet.conf"
[kubeconfig] Wrote KubeConfig file to disk: "controller-manager.conf"
[kubeconfig] Wrote KubeConfig file to disk: "scheduler.conf"
[controlplane] Wrote Static Pod manifest for component kube-apiserver to "/etc/kubernetes/manifests/kube-apiserver.yaml"
[controlplane] Wrote Static Pod manifest for component kube-controller-manager to "/etc/kubernetes/manifests/kube-controller-manager.yaml"
[controlplane] Wrote Static Pod manifest for component kube-scheduler to "/etc/kubernetes/manifests/kube-scheduler.yaml"
[etcd] Wrote Static Pod manifest for a local etcd instance to "/etc/kubernetes/manifests/etcd.yaml"
[init] Waiting for the kubelet to boot up the control plane as Static Pods from directory "/etc/kubernetes/manifests".
[init] This might take a minute or longer if the control plane images have to be pulled.
crictl images
crictl ps
cat /etc/crictl.yaml
sudo cp /etc/kubernetes/admin.conf $HOME/
sudo chown $(id -u):$(id -g) $HOME/admin.conf
export KUBECONFIG=$HOME/admin.conf
kubectl get nodes
kubectl describe node master01  | grep "Container Runtime Version:"
kubectl taint nodes --all node-role.kubernetes.io/master-
[apiclient] All control plane components are healthy after 30.504060 seconds
[uploadconfig] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[markmaster] Will mark node master01 as master by adding a label and a taint
[markmaster] Master master01 tainted and labelled with key/value: node-role.kubernetes.io/master=""
[bootstraptoken] Using token: a6a6e1.8c836670431ee2df
[bootstraptoken] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstraptoken] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstraptoken] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
```



```
kubectl taint nodes --all node-role.kubernetes.io/master-
[apiclient] All control plane components are healthy after 30.504060 seconds
[uploadconfig] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[markmaster] Will mark node master01 as master by adding a label and a taint
[markmaster] Master master01 tainted and labelled with key/value: node-role.kubernetes.io/master=""
[bootstraptoken] Using token: a6a6e1.8c836670431ee2df
[bootstraptoken] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstraptoken] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstraptoken] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstraptoken] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[addons] Applied essential addon: kube-dns
[addons] Applied essential addon: kube-proxy

Your Kubernetes master has initialized successfully!

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

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

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/

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

  kubeadm join --token a6a6e1.8c836670431ee2df 172.17.0.33:6443 --discovery-token-ca-cert-hash sha256:f0f46e19d031aa8f40eaca50cf7833e4d8850c46551f0d8
3bb1c0d283d913339

$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
$ crictl images
IMAGE                                                 TAG                 IMAGE ID            SIZE
docker.io/kubernetes/pause                            latest              f9d5de0795395       251kB
docker.io/weaveworks/weave-kube                       2.2.0               222ab9e78a839       96.7MB
docker.io/weaveworks/weave-npc                        2.2.0               765b48853ac02       47MB
gcr.io/google_containers/etcd-amd64                   3.1.11              59d36f27cceb9       194MB
gcr.io/google_containers/kube-apiserver-amd64         v1.9.3              360d55f91cbf4       211MB
gcr.io/google_containers/kube-controller-manager-amd64   v1.9.3           83dbda6ee8104       138MB
gcr.io/google_containers/kube-proxy-amd64             v1.9.3              35fdc6da5fd83       111MB
gcr.io/google_containers/kube-scheduler-amd64         v1.9.3              d3534b539b764       62.9MB
```



```
0a923ba0cd7bd       gcr.io/google_containers/kube-controller-manager-amd64@sha256:3ac295ae3e78af5c9f88164ae95097c2d7af03caddf067cb35599769d0b7251e
18 seconds ago      CONTAINER_RUNNING   kube-controller-manager   0
08b2eb5f5f41b       gcr.io/google_containers/kube-apiserver-amd64@sha256:a5382344aa373a90bc87d3baa4eda5402507e8df5b8bfbbad392c4fff715f043
18 seconds ago      CONTAINER_RUNNING   kube-apiserver            0
d450a8ab70a8f       gcr.io/google_containers/kube-scheduler-amd64@sha256:2c17e637c8e4f9202300bd5fc26bc98a7099f49559ca0a8921cf692ffd4a1675
18 seconds ago      CONTAINER_RUNNING   kube-scheduler            0
$ cat /etc/crictl.yaml
runtime-endpoint: /var/run/crio/crio.sock
$ sudo cp /etc/kubernetes/admin.conf $HOME/
$ sudo chown $(id -u):$(id -g) $HOME/admin.conf
$ export KUBECONFIG=$HOME/admin.conf
$ kubectl get nodes
NAME       STATUS    ROLES     AGE       VERSION
master01   Ready     master    9s        v1.9.3
$ kubectl describe node master01  | grep "Container Runtime Version:"
 Container Runtime Version:  cri-o://1.9.10-dev
$ kubectl taint nodes --all node-role.kubernetes.io/master-
node "master01" untainted
$ kubectl get pods --all-namespaces
No resources found.
$ cat /opt/weave-kube.yaml
cat: /opt/weave-kube.yaml: No such file or directory
$ kubectl apply -f /opt/weave-kube.yaml
error: the path "/opt/weave-kube.yaml" does not exist
$ kubectl get pod -n kube-system
NAME                       READY     STATUS              RESTARTS   AGE
kube-dns-6f4fd4bdf-7s52v   0/3       ContainerCreating   0          8s
kube-proxy-bpqkt           1/1       Running             0          8s
$ kubectl run http --image=katacoda/docker-http-server:latest --replicas=1
deployment "http" created
$ kubectl get pods
NAME                    READY     STATUS              RESTARTS   AGE
http-85ddfcb674-krz5p   0/1       ContainerCreating   0          3s
$ kubectl set image deployment/http http=docker.io/katacoda/docker-http-server:latest
deployment "http" image updated
$ crictl ps | grep docker-http-server
$ kubectl apply -f dashboard.yaml
error: the path "dashboard.yaml" does not exist
$ kubectl get pods -n kube-system
NAME                       READY     STATUS              RESTARTS   AGE
kube-dns-6f4fd4bdf-7s52v   0/3       ContainerCreating   0          26s
kube-proxy-bpqkt           1/1       Running             0          26s
$
```