Arktos deployment without Mizar CNI (On Premise)

Date-10 Dec. 2021

This document is intended for new users to install the Arktos platform without Mizar as the underlying network technology.

Prepare a machine of 16 Gb RAM, 8 vCPUs, 128G Storage, Ubuntu 18.04 LTS.

1. Check the kernel version:

Command:

uname -a

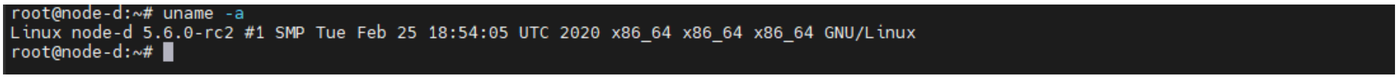
Update the kernel if the kernel version is below `5.6.0-rc2`

wget https://raw.githubusercontent.com/CentaurusInfra/mizar/dev-next/kernelupdate.sh

sudo bash kernelupdate.sh

uname -a

Output:



```
root@node-d:~# uname -a
Linux node-d 5.6.0-rc2 #1 SMP Tue Feb 25 18:54:05 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
root@node-d:~#
```

2. Clone the Arktos repository and install the required dependencies:

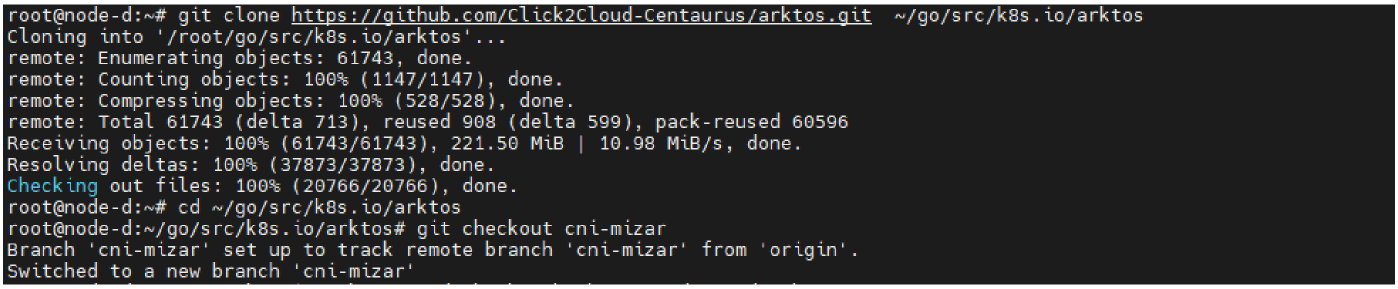
git clone https://github.com/Click2Cloud-Centaurus/arktos.git ~/go/src/k8s.io/arktos

cd ~/go/src/k8s.io/arktos

git checkout cni-mizar

sudo bash ./hack/setup-dev-node.sh

Output:



```
root@node-d:~# git clone https://github.com/Click2Cloud-Centaurus/arktos.git  ~/go/src/k8s.io/arktos
Cloning into '/root/go/src/k8s.io/arktos'...
remote: Enumerating objects: 61743, done.
remote: Counting objects: 100% (1147/1147), done.
remote: Compressing objects: 100% (528/528), done.
remote: Total 61743 (delta 713), reused 908 (delta 599), pack-reused 60596
Receiving objects: 100% (61743/61743), 221.50 MiB | 10.98 MiB/s, done.
Resolving deltas: 100% (37873/37873), done.
Checking out files: 100% (20766/20766), done.
root@node-d:~# cd ~/go/src/k8s.io/arktos
root@node-d:~/go/src/k8s.io/arktos# git checkout cni-mizar
Branch 'cni-mizar' set up to track remote branch 'cni-mizar' from 'origin'.
Switched to a new branch 'cni-mizar'
```

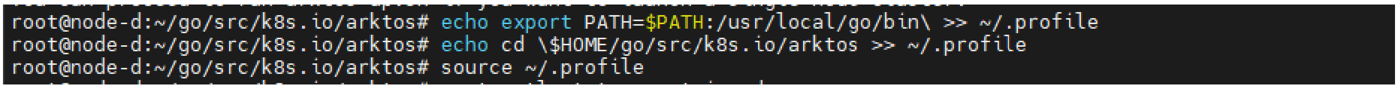
Command:

echo export PATH=$PATH:/usr/local/go/bin\ >> ~/.profile

echo cd \$HOME/go/src/k8s.io/arktos >> ~/.profile

source ~/.profile

Output:



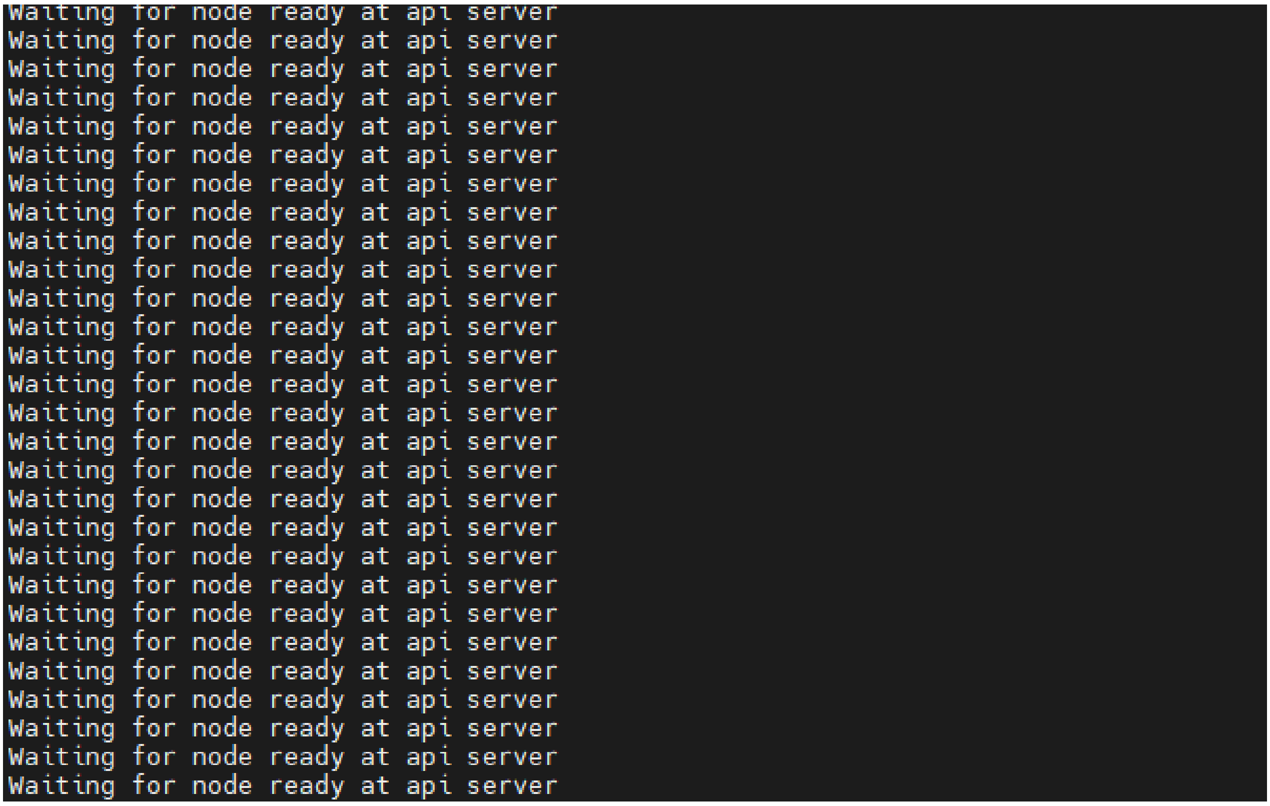
```
root@node-d:~/go/src/k8s.io/arktos# echo export PATH=$PATH:/usr/local/go/bin\ >> ~/.profile
root@node-d:~/go/src/k8s.io/arktos# echo cd \$HOME/go/src/k8s.io/arktos >> ~/.profile
root@node-d:~/go/src/k8s.io/arktos# source ~/.profile
```

3. Start Arktos cluster

Command:

`./hack/arktos-up.sh`

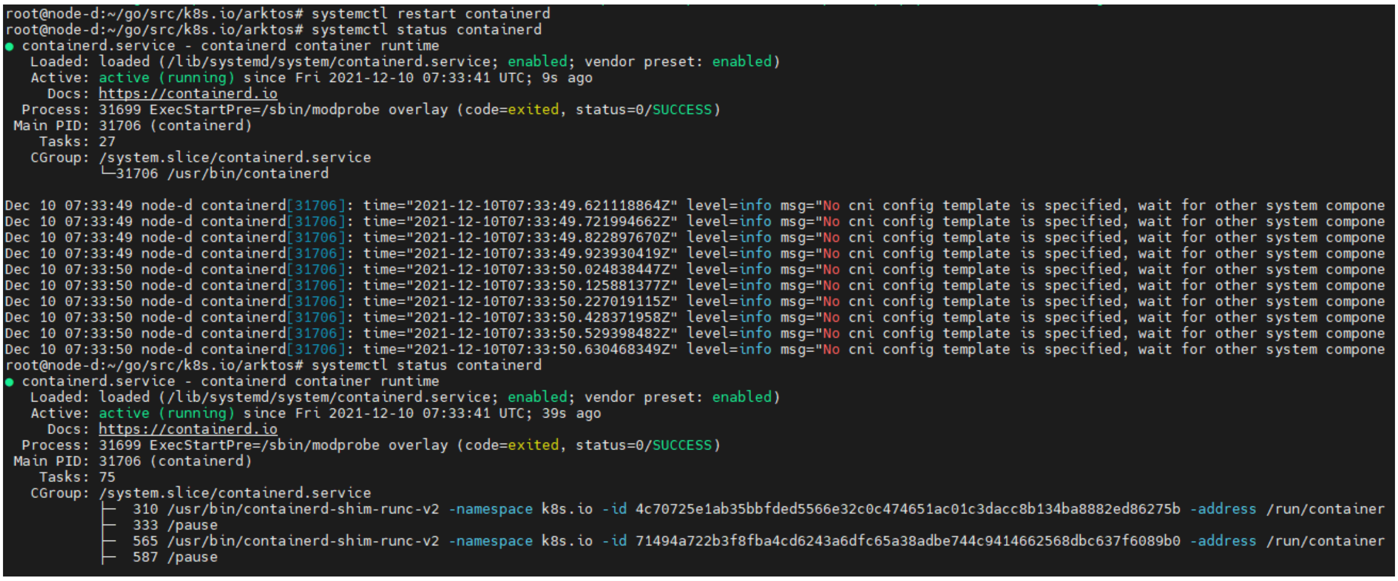
The terminal was stuck in this state.



```
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
```

After restarting the containerd we got the output:

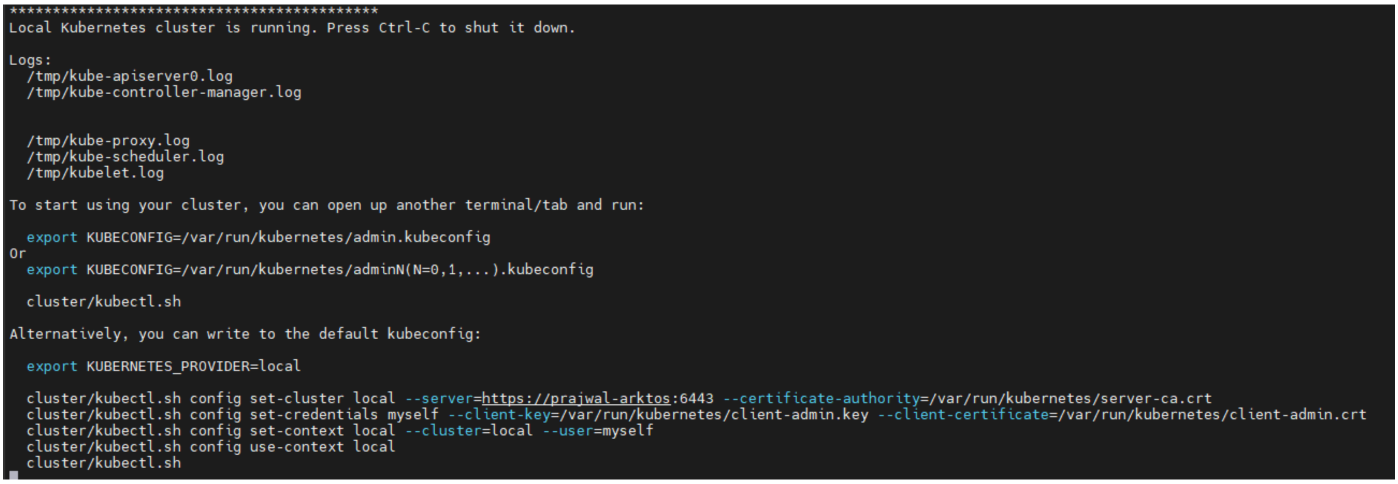
systemctl restart containerd



```
root@node-d:~/go/src/k8s.io/arktos# systemctl restart containerd
root@node-d:~/go/src/k8s.io/arktos# systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/lib/systemd/system/containerd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-12-10 07:33:41 UTC; 9s ago
     Docs: https://containerd.io
  Process: 31699 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
 Main PID: 31706 (containerd)
    Tasks: 27
   CGroup: /system.slice/containerd.service
           └─31706 /usr/bin/containerd

Dec 10 07:33:49 node-d containerd[31706]: time="2021-12-10T07:33:49.621118864Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:49 node-d containerd[31706]: time="2021-12-10T07:33:49.721994662Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:49 node-d containerd[31706]: time="2021-12-10T07:33:49.822897670Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:49 node-d containerd[31706]: time="2021-12-10T07:33:49.923930419Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:50 node-d containerd[31706]: time="2021-12-10T07:33:50.024838447Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:50 node-d containerd[31706]: time="2021-12-10T07:33:50.125881377Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:50 node-d containerd[31706]: time="2021-12-10T07:33:50.227019115Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:50 node-d containerd[31706]: time="2021-12-10T07:33:50.428371958Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:50 node-d containerd[31706]: time="2021-12-10T07:33:50.529398482Z" level=info msg="No cni config template is specified, wait for other system compone
Dec 10 07:33:50 node-d containerd[31706]: time="2021-12-10T07:33:50.630468349Z" level=info msg="No cni config template is specified, wait for other system compone
root@node-d:~/go/src/k8s.io/arktos# systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/lib/systemd/system/containerd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-12-10 07:33:41 UTC; 39s ago
     Docs: https://containerd.io
  Process: 31699 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
 Main PID: 31706 (containerd)
    Tasks: 75
   CGroup: /system.slice/containerd.service
           ├─  310 /usr/bin/containerd-shim-runc-v2 -namespace k8s.io -id 4c70725e1ab35bbfded5566e32c0c474651ac01c3dacc8b134ba8882ed86275b -address /run/container
           ├─  333 /pause
           ├─  565 /usr/bin/containerd-shim-runc-v2 -namespace k8s.io -id 71494a722b3f8fba4cd6243a6dfc65a38adbe744c9414662568dbc637f6089b0 -address /run/container
           ├─  587 /pause
```

Output:



```
*******************************************
Local Kubernetes cluster is running. Press Ctrl-C to shut it down.

Logs:
  /tmp/kube-apiserver0.log
  /tmp/kube-controller-manager.log


  /tmp/kube-proxy.log
  /tmp/kube-scheduler.log
  /tmp/kubelet.log

To start using your cluster, you can open up another terminal/tab and run:

  export KUBECONFIG=/var/run/kubernetes/admin.kubeconfig
Or
  export KUBECONFIG=/var/run/kubernetes/adminN(N=0,1,...).kubeconfig

  cluster/kubectl.sh

Alternatively, you can write to the default kubeconfig:

  export KUBERNETES_PROVIDER=local

  cluster/kubectl.sh config set-cluster local --server=https://prajwal-arktos:6443 --certificate-authority=/var/run/kubernetes/server-ca.crt
  cluster/kubectl.sh config set-credentials myself --client-key=/var/run/kubernetes/client-admin.key --client-certificate=/var/run/kubernetes/client-admin.crt
  cluster/kubectl.sh config set-context local --cluster=local --user=myself
  cluster/kubectl.sh config use-context local
  cluster/kubectl.sh
```

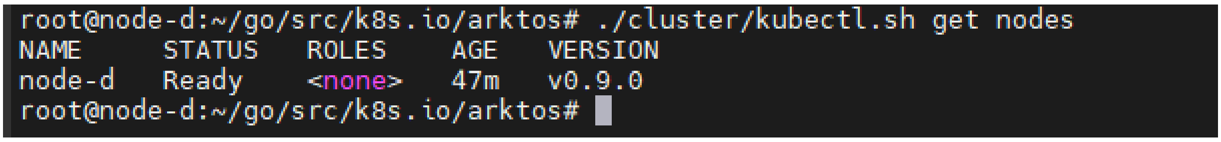
4. Leave the "arktos-up.sh" terminal and open another terminal to the master node.

Check nodes:

Command:

`./cluster/kubectl.sh get nodes`

Output:



```
root@node-d:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh get nodes
NAME     STATUS   ROLES    AGE   VERSION
node-d   Ready    <none>   47m   v0.9.0
root@node-d:~/go/src/k8s.io/arktos#
```

Deploy test pods:

Command:

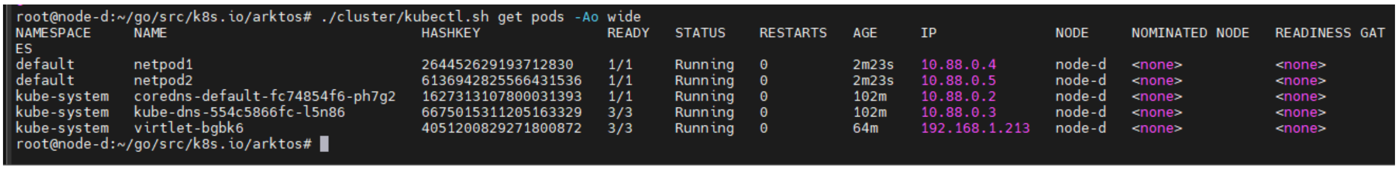
./cluster/kubectl.sh apply -f https://github.com/Click2Cloud-Centaurus/Documentation/blob/main/test-yamls/test_pods.yaml

Check deployed pods:

Command:

./cluster/kubectl.sh get pods -Ao wide

Output:



```
root@node-d:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                              HASHKEY               READY   STATUS    RESTARTS   AGE     IP             NODE     NOMINATED NODE   READINESS GAT
ES
default       netpod1                           264452629193712830    1/1     Running   0          2m23s   10.88.0.4      node-d   <none>           <none>
default       netpod2                           6136942825566431536   1/1     Running   0          2m23s   10.88.0.5      node-d   <none>           <none>
kube-system   coredns-default-fc74854f6-ph7g2   1627313107800031393   1/1     Running   0          102m    10.88.0.2      node-d   <none>           <none>
kube-system   kube-dns-554c5866fc-l5n86         6675015311205163329   3/3     Running   0          102m    10.88.0.3      node-d   <none>           <none>
kube-system   virtlet-bgbk6                     4051200829271800872   3/3     Running   0          64m     192.168.1.213  node-d   <none>           <none>
root@node-d:~/go/src/k8s.io/arktos#
```

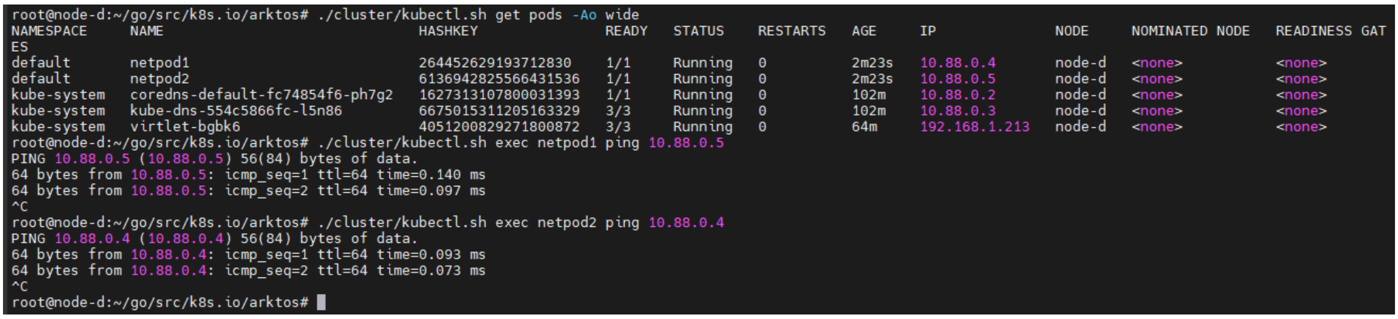
Check ping deployed pods:

Command:

./cluster/kubectl.sh exec netpod1 ping 10.88.0.5

./cluster/kubectl.sh exec netpod2 ping 10.88.0.4

Output:



```
root@node-d:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                              HASHKEY               READY   STATUS    RESTARTS   AGE     IP              NODE     NOMINATED NODE   READINESS GAT
ES
default       netpod1                           2644526291937128300   1/1     Running   0          2m23s   10.88.0.4       node-d   <none>           <none>
default       netpod2                           6136942825566431536   1/1     Running   0          2m23s   10.88.0.5       node-d   <none>           <none>
kube-system   coredns-default-fc74854f6-ph7g2   1627313107800031393   1/1     Running   0          102m    10.88.0.2       node-d   <none>           <none>
kube-system   kube-dns-554c5866fc-l5n86         6675015311205163329   3/3     Running   0          102m    10.88.0.3       node-d   <none>           <none>
kube-system   virtlet-bgbk6                     4051200829271800872   3/3     Running   0          64m     192.168.1.213   node-d   <none>           <none>
root@node-d:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh exec netpod1 ping 10.88.0.5
PING 10.88.0.5 (10.88.0.5) 56(84) bytes of data.
64 bytes from 10.88.0.5: icmp_seq=1 ttl=64 time=0.140 ms
64 bytes from 10.88.0.5: icmp_seq=2 ttl=64 time=0.097 ms
^C
root@node-d:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh exec netpod2 ping 10.88.0.4
PING 10.88.0.4 (10.88.0.4) 56(84) bytes of data.
64 bytes from 10.88.0.4: icmp_seq=1 ttl=64 time=0.093 ms
64 bytes from 10.88.0.4: icmp_seq=2 ttl=64 time=0.073 ms
^C
root@node-d:~/go/src/k8s.io/arktos# 
```