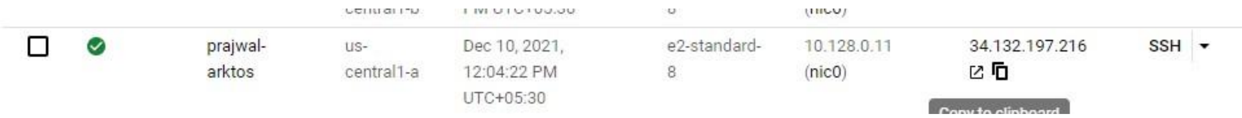
Test report - Deployment of Arktos Cluster without Mizar CNI on GCP

This document captures the steps to deploy an Arktos cluster lab without Mizar CNI. The machine in this lab used are GCE e2-standard-8 (8 vCPUs, 32 GB memory) and the storage size is 128GB), Ubuntu 18.04 LTS.

Date-10 Dec. 2021

Create an instance on GCE

Created instance on GCE



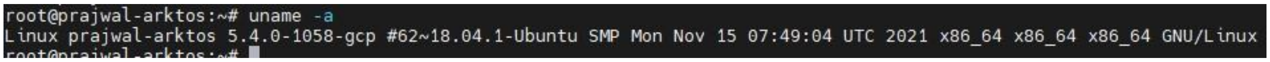
			central1-b	PM UTC+05:30	8	(nic0)		
□	✓	prajwal-arktos	us-central1-a	Dec 10, 2021, 12:04:22 PM UTC+05:30	e2-standard-8	10.128.0.11 (nic0)	34.132.197.216	SSH ▾

SSH instance with credentials.

Step-1: Update kernel (If required)

To check kernel, run following command

`uname -a` **output:**

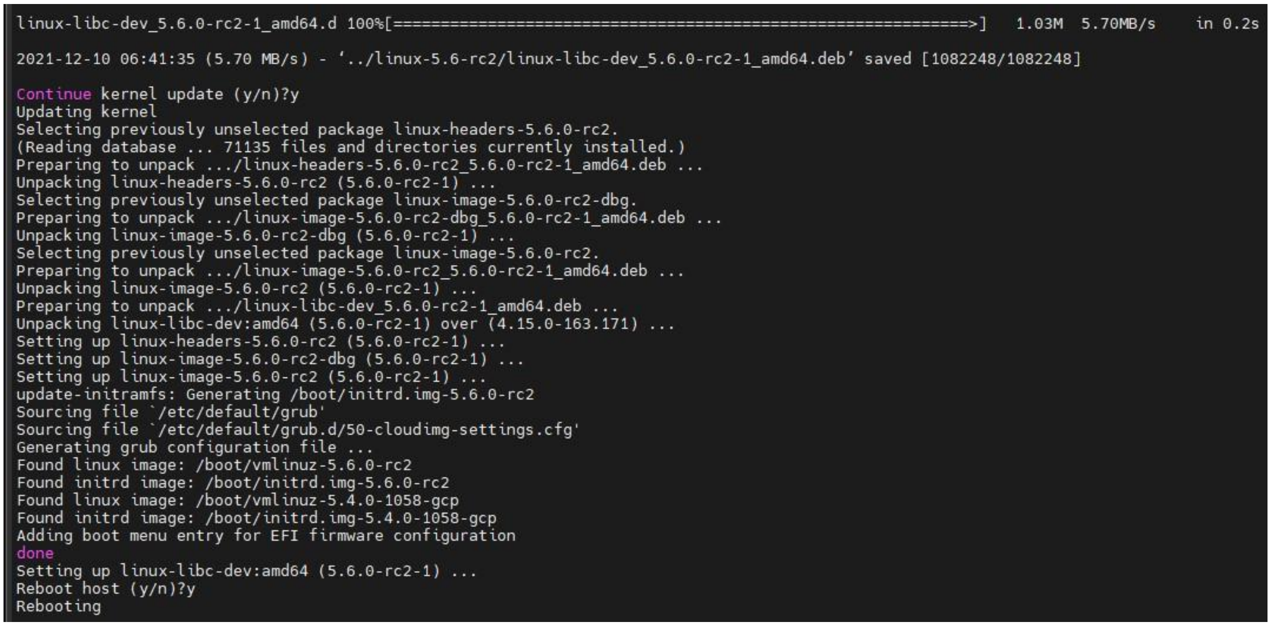


```
root@prajwal-arktos:~# uname -a
Linux prajwal-arktos 5.4.0-1058-gcp #62~18.04.1-Ubuntu SMP Mon Nov 15 07:49:04 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
root@prajwal-arktos:~#
```

Here kernel version is 5.4.0-1051-gcp which is less than the required kernel version, so to update the kernel version to 5.6.0-rc2, we used the following steps :

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

output:



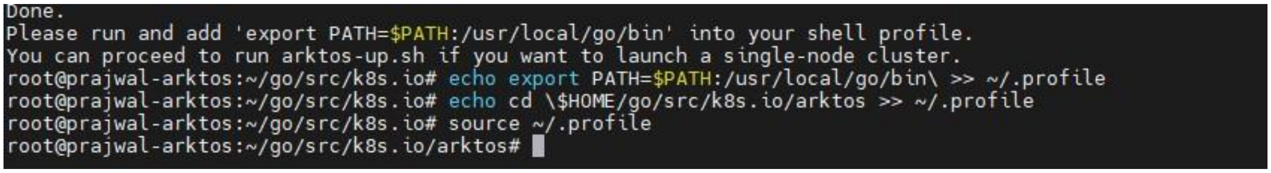
```
linux-libc-dev_5.6.0-rc2-1_amd64.d 100%[=============================================================>]   1.03M  5.70MB/s    in 0.2s

2021-12-10 06:41:35 (5.70 MB/s) - ‘../linux-5.6-rc2/linux-libc-dev_5.6.0-rc2-1_amd64.deb’ saved [1082248/1082248]

Continue kernel update (y/n)?y
Updating kernel
Selecting previously unselected package linux-headers-5.6.0-rc2.
(Reading database ... 71135 files and directories currently installed.)
Preparing to unpack .../linux-headers-5.6.0-rc2_5.6.0-rc2-1_amd64.deb ...
Unpacking linux-headers-5.6.0-rc2 (5.6.0-rc2-1) ...
Selecting previously unselected package linux-image-5.6.0-rc2-dbg.
Preparing to unpack .../linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb ...
Unpacking linux-image-5.6.0-rc2-dbg (5.6.0-rc2-1) ...
Selecting previously unselected package linux-image-5.6.0-rc2.
Preparing to unpack .../linux-image-5.6.0-rc2_5.6.0-rc2-1_amd64.deb ...
Unpacking linux-image-5.6.0-rc2 (5.6.0-rc2-1) ...
Preparing to unpack .../linux-libc-dev_5.6.0-rc2-1_amd64.deb ...
Unpacking linux-libc-dev:amd64 (5.6.0-rc2-1) over (4.15.0-163.171) ...
Setting up linux-headers-5.6.0-rc2 (5.6.0-rc2-1) ...
Setting up linux-image-5.6.0-rc2-dbg (5.6.0-rc2-1) ...
Setting up linux-image-5.6.0-rc2 (5.6.0-rc2-1) ...
update-initramfs: Generating /boot/initrd.img-5.6.0-rc2
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/50-cloudimg-settings.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.6.0-rc2
Found initrd image: /boot/initrd.img-5.6.0-rc2
Found linux image: /boot/vmlinuz-5.4.0-1058-gcp
Found initrd image: /boot/initrd.img-5.4.0-1058-gcp
Adding boot menu entry for EFI firmware configuration
done
Setting up linux-libc-dev:amd64 (5.6.0-rc2-1) ...
Reboot host (y/n)?y
Rebooting
```

Step-2: Install dependencies

Run the following steps to install dependencies required for arktos deployment:

```
git clone https://github.com/Click2Cloud-Centaurus/arktos.git ~/go/src/k8s.io/arktos
-b default-cni-mizar  bash

$HOME/go/src/k8s.io/arktos/hack/setup-dev-node.sh  echo

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

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

source ~/.profile
```

output:



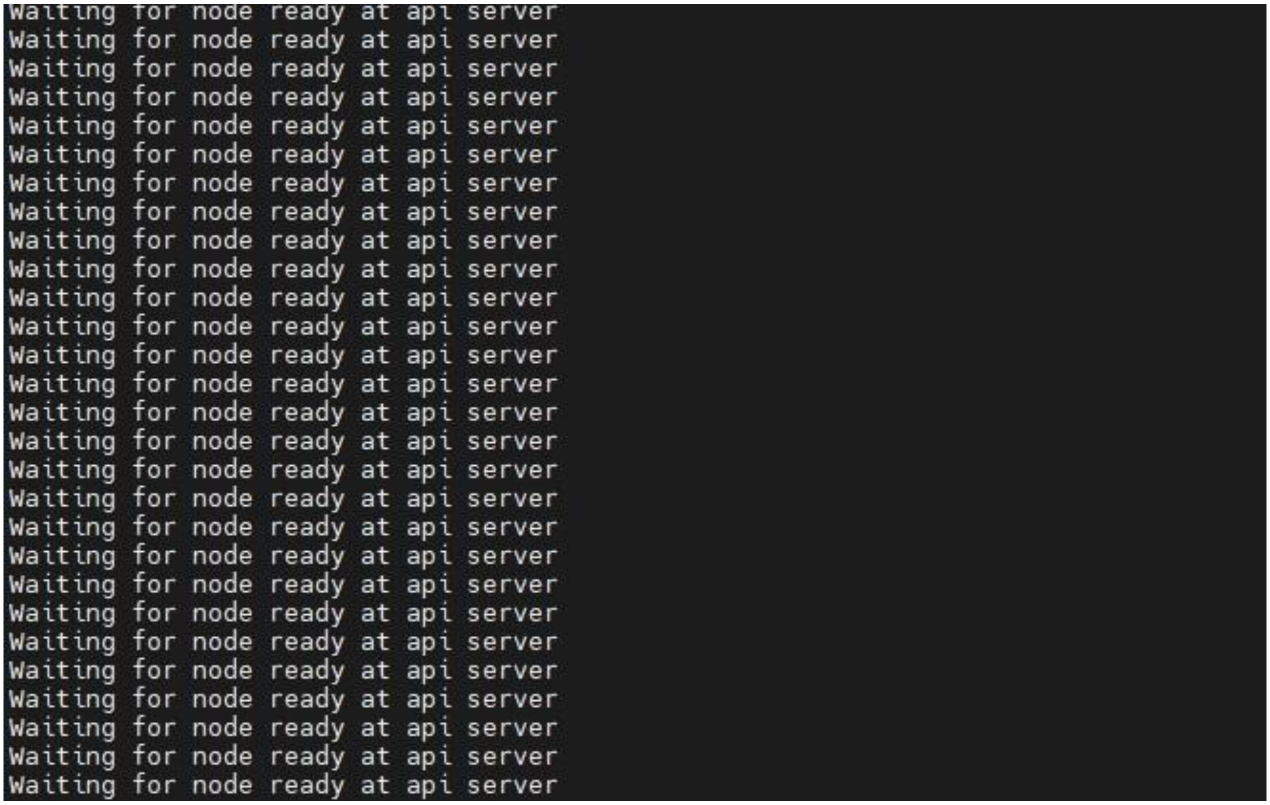
```
Done.
Please run and add 'export PATH=$PATH:/usr/local/go/bin' into your shell profile.
You can proceed to run arktos-up.sh if you want to launch a single-node cluster.
root@prajwal-arktos:~/go/src/k8s.io# echo export PATH=$PATH:/usr/local/go/bin\ >> ~/.profile
root@prajwal-arktos:~/go/src/k8s.io# echo cd \$HOME/go/src/k8s.io/arktos >> ~/.profile
root@prajwal-arktos:~/go/src/k8s.io# source ~/.profile
root@prajwal-arktos:~/go/src/k8s.io/arktos# █
```

Step-3: Start Arktos cluster

Login to instance and run following steps to deploy arktos cluster without Mizar as CNI:

```
./hack/arktos-up.sh
```

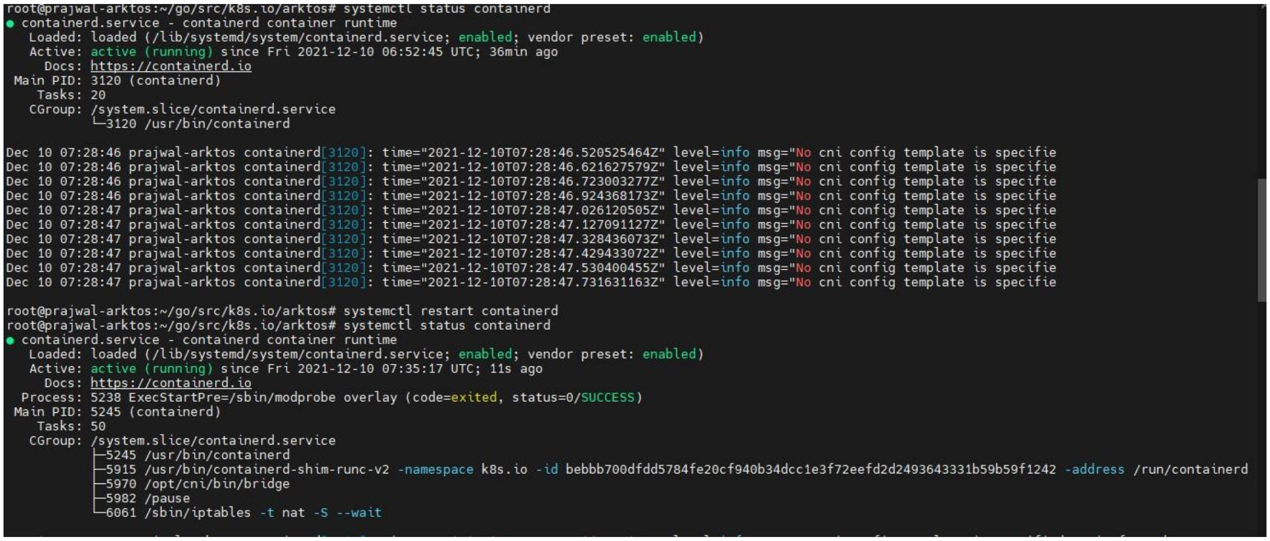
The terminal was stuck in this state.



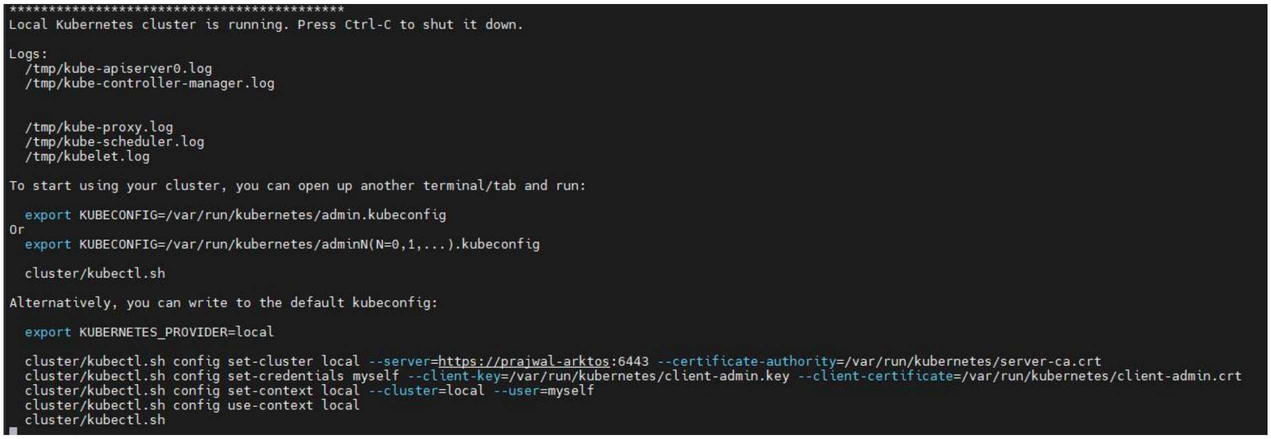
```
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting for node ready at api server
Waiting 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@prajwal-arktos:~/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 06:52:45 UTC; 36min ago
     Docs: https://containerd.io
 Main PID: 3120 (containerd)
    Tasks: 20
   CGroup: /system.slice/containerd.service
           └─3120 /usr/bin/containerd

Dec 10 07:28:46 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:46.520525464Z" level=info msg="No cni config template is specifie
Dec 10 07:28:46 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:46.621627579Z" level=info msg="No cni config template is specifie
Dec 10 07:28:46 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:46.723003277Z" level=info msg="No cni config template is specifie
Dec 10 07:28:46 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:46.924368173Z" level=info msg="No cni config template is specifie
Dec 10 07:28:47 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:47.026120505Z" level=info msg="No cni config template is specifie
Dec 10 07:28:47 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:47.127091127Z" level=info msg="No cni config template is specifie
Dec 10 07:28:47 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:47.328436073Z" level=info msg="No cni config template is specifie
Dec 10 07:28:47 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:47.429433072Z" level=info msg="No cni config template is specifie
Dec 10 07:28:47 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:47.530400455Z" level=info msg="No cni config template is specifie
Dec 10 07:28:47 prajwal-arktos containerd[3120]: time="2021-12-10T07:28:47.731631163Z" level=info msg="No cni config template is specifie

root@prajwal-arktos:~/go/src/k8s.io/arktos# systemctl restart containerd
root@prajwal-arktos:~/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 07:35:17 UTC; 11s ago
     Docs: https://containerd.io
  Process: 5238 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
 Main PID: 5245 (containerd)
    Tasks: 50
   CGroup: /system.slice/containerd.service
           ├─5245 /usr/bin/containerd
           ├─5915 /usr/bin/containerd-shim-runc-v2 -namespace k8s.io -id bebbb700dfdd5784fe20cf940b34dcc1e3f72eefd2d2493643331b59b59f1242 -address /run/containerd
           ├─5970 /opt/cni/bin/bridge
           ├─5982 /pause
           └─6061 /sbin/iptables -t nat -S --wait
```



```
********************************************
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
```

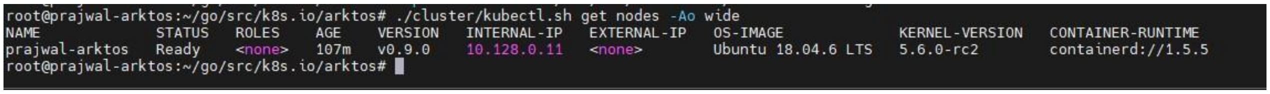
Leave this terminal here as it is (do not close the terminal) and open new terminal of same instance

Open new terminal for same instance and run following commands:

1) Check node status

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

Output



```
root@prajwal-arktos:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh get nodes -Ao wide
NAME             STATUS   ROLES    AGE    VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
prajwal-arktos   Ready    <none>   107m   v0.9.0    10.128.0.11   <none>        Ubuntu 18.04.6 LTS   5.6.0-rc2        containerd://1.5.5
root@prajwal-arktos:~/go/src/k8s.io/arktos# 
```

2) Check Deployed pods status

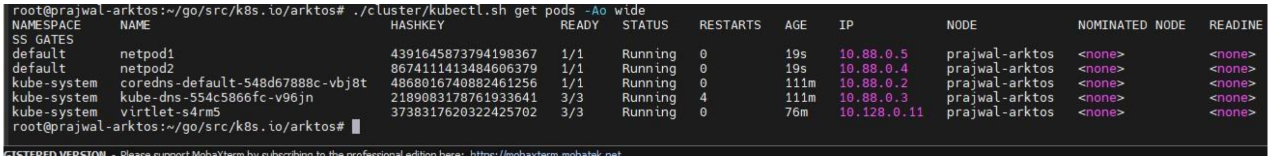
Deploy test pods:

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

Check deployed pods:

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

Output



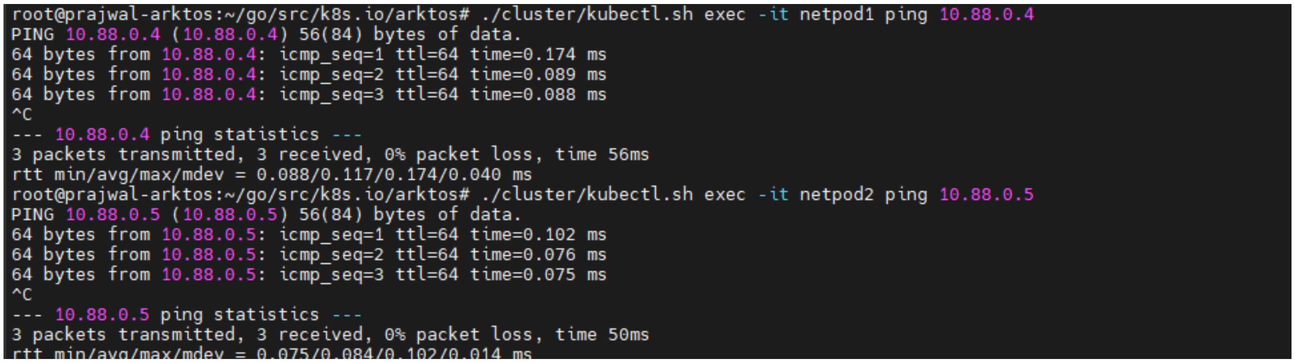
```
root@prajwal-arktos:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                                HASHKEY               READY   STATUS    RESTARTS   AGE    IP            NODE             NOMINATED NODE   READINE
SS GATES
default       netpod1                             4391645873794198367   1/1     Running   0          19s    10.88.0.5     prajwal-arktos   <none>           <none>
default       netpod2                             8674111413484606379   1/1     Running   0          19s    10.88.0.4     prajwal-arktos   <none>           <none>
kube-system   coredns-default-548d67888c-vbj8t    4868016740882461256   1/1     Running   0          111m   10.88.0.2     prajwal-arktos   <none>           <none>
kube-system   kube-dns-554c5866fc-v96jn           2189083178761933641   3/3     Running   4          111m   10.88.0.3     prajwal-arktos   <none>           <none>
kube-system   virtlet-s4rm5                       3738317620322425702   3/3     Running   0          76m    10.128.0.11   prajwal-arktos   <none>           <none>
root@prajwal-arktos:~/go/src/k8s.io/arktos# 
```

GISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

3) Check ping of deployed pods

```
./cluster/kubectl.sh exec -it netpod1 ping 10.88.0.4
./cluster/kubectl.sh exec -it netpod2 ping 10.88.0.5
```

Output



```
root@prajwal-arktos:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh exec -it netpod1 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=0.174 ms
64 bytes from 10.88.0.4: icmp_seq=2 ttl=64 time=0.089 ms
64 bytes from 10.88.0.4: icmp_seq=3 ttl=64 time=0.088 ms
^C
--- 10.88.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 56ms
rtt min/avg/max/mdev = 0.088/0.117/0.174/0.040 ms
root@prajwal-arktos:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh exec -it netpod2 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=0.102 ms
64 bytes from 10.88.0.5: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 10.88.0.5: icmp_seq=3 ttl=64 time=0.075 ms
^C
--- 10.88.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 50ms
rtt min/avg/max/mdev = 0.075/0.084/0.102/0.014 ms
```