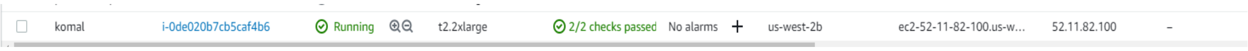
Test report -Deploying Arktos cluster with Mizar CNI on AWS

This document captures the steps to deploy an Arktos cluster lab with mizar cni. The machines in this lab used are AWS EC2 t2.2xlarge (8 CPUs, 32GB mem), Ubuntu 18.04 LTS.

Date: 24.09.2021

Created an instance on AWS



□	komal	i-0de020b7cb5caf4b6	Running	t2.2xlarge	2/2 checks passed	No alarms	+	us-west-2b	ec2-52-11-82-100.us-w...	52.11.82.100	–

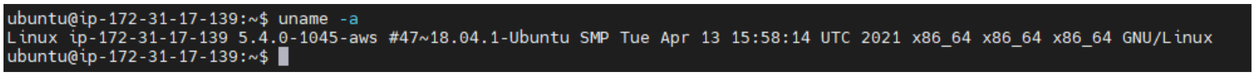
SSH instance using credentials

Step-1: Update kernel version

- Check kernel version:

```
uname -a
```

Output

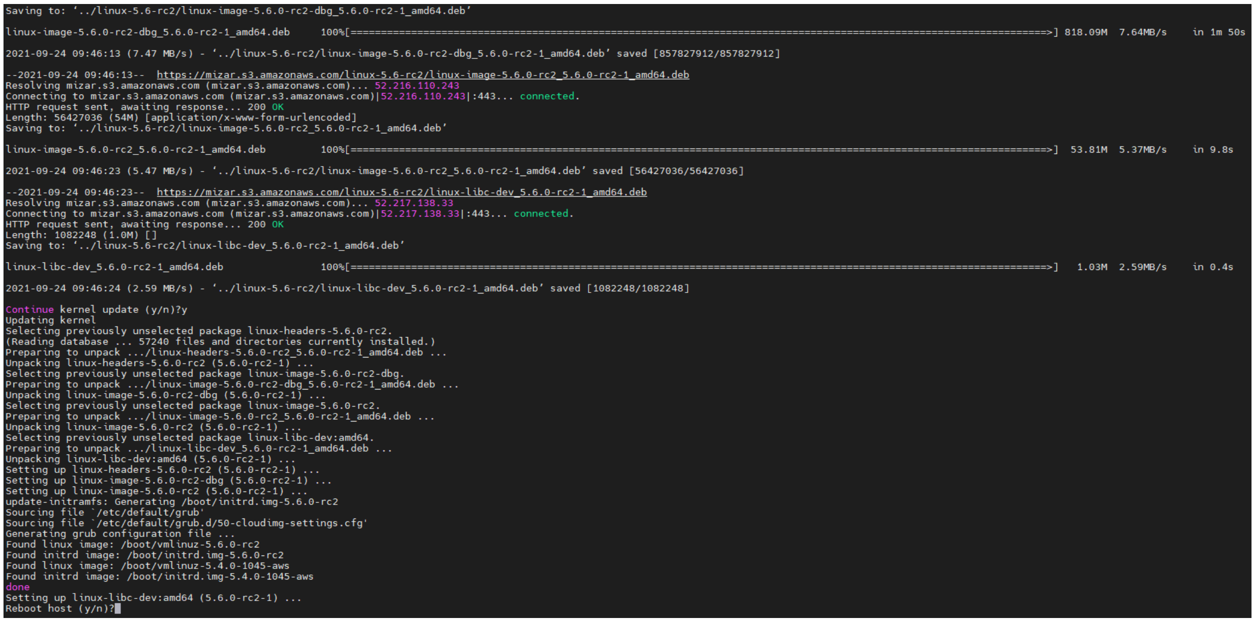


```
ubuntu@ip-172-31-17-139:~$ uname -a
Linux ip-172-31-17-139 5.4.0-1045-aws #47~18.04.1-Ubuntu SMP Tue Apr 13 15:58:14 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
ubuntu@ip-172-31-17-139:~$
```

Here kernel version was `5.4.0-1045-aws` hence, to update kernel version to `5.6.0-rc2`, we used following steps :

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

Output



```
Saving to: '../linux-5.6-rc2/linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb'

linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb     100%[==========================================================================>] 818.09M  7.64MB/s    in 1m 50s

2021-09-24 09:46:13 (7.47 MB/s) - '../linux-5.6-rc2/linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb' saved [857827912/857827912]

--2021-09-24 09:46:13--  https://mizar.s3.amazonaws.com/linux-5.6-rc2/linux-image-5.6.0-rc2_5.6.0-rc2-1_amd64.deb
Resolving mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)... 52.216.110.243
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.216.110.243|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 56427036 (54M) [application/x-www-form-urlencoded]
Saving to: '../linux-5.6-rc2/linux-image-5.6.0-rc2_5.6.0-rc2-1_amd64.deb'

linux-image-5.6.0-rc2_5.6.0-rc2-1_amd64.deb         100%[==========================================================================>]  53.81M  5.37MB/s    in 9.8s

2021-09-24 09:46:23 (5.47 MB/s) - '../linux-5.6-rc2/linux-image-5.6.0-rc2_5.6.0-rc2-1_amd64.deb' saved [56427036/56427036]

--2021-09-24 09:46:23--  https://mizar.s3.amazonaws.com/linux-5.6-rc2/linux-libc-dev_5.6.0-rc2-1_amd64.deb
Resolving mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)... 52.217.138.33
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.217.138.33|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1082248 (1.0M) []
Saving to: '../linux-5.6-rc2/linux-libc-dev_5.6.0-rc2-1_amd64.deb'

linux-libc-dev_5.6.0-rc2-1_amd64.deb                100%[==========================================================================>]   1.03M  2.59MB/s    in 0.4s

2021-09-24 09:46:24 (2.59 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 ... 57240 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) ...
Selecting previously unselected package linux-libc-dev:amd64.
Preparing to unpack .../linux-libc-dev_5.6.0-rc2-1_amd64.deb ...
Unpacking linux-libc-dev:amd64 (5.6.0-rc2-1) ...
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-5.4.0-1045-aws
Found initrd image: /boot/initrd.img-5.4.0-1045-aws
done
Setting up linux-libc-dev:amd64 (5.6.0-rc2-1) ...
Reboot host (y/n)?
```

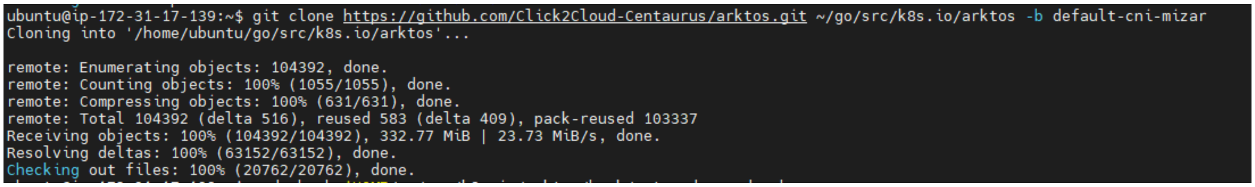
Step-2:Install dependencies

Relogin the instance and run following steps to install dependencies required for arktos deployment:

- Clone the Arktos repository

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

Output

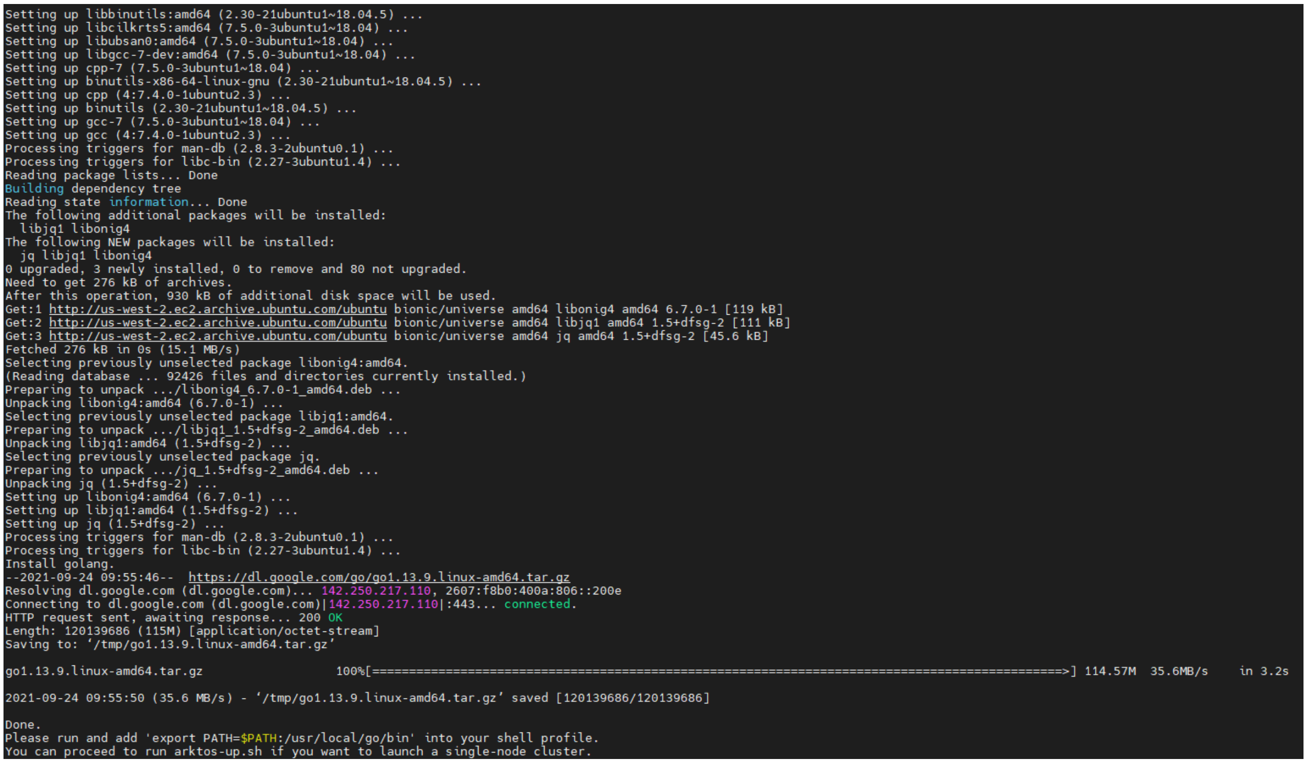


```
ubuntu@ip-172-31-17-139:~$ git clone https://github.com/Click2Cloud-Centaurus/arktos.git ~/go/src/k8s.io/arktos -b default-cni-mizar
Cloning into '/home/ubuntu/go/src/k8s.io/arktos'...

remote: Enumerating objects: 104392, done.
remote: Counting objects: 100% (1055/1055), done.
remote: Compressing objects: 100% (631/631), done.
remote: Total 104392 (delta 516), reused 583 (delta 409), pack-reused 103337
Receiving objects: 100% (104392/104392), 332.77 MiB | 23.73 MiB/s, done.
Resolving deltas: 100% (63152/63152), done.
Checking out files: 100% (20762/20762), done.
```

Then installed prerequisites required for Arktos cluster suing following command

```
sudo bash $HOME/go/src/k8s.io/arktos/hack/setup-dev-node.sh
```

Output

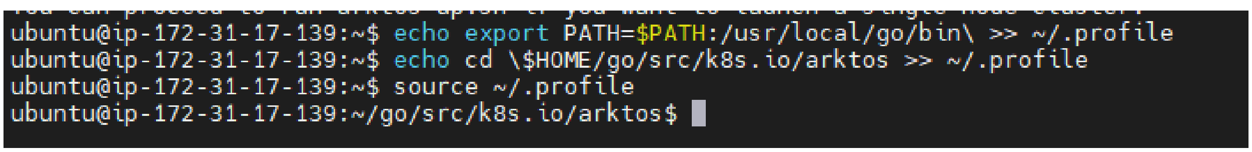


```
Setting up libbinutils:amd64 (2.30-21ubuntu1~18.04.5) ...
Setting up libcilkrts5:amd64 (7.5.0-3ubuntu1~18.04) ...
Setting up libubsan0:amd64 (7.5.0-3ubuntu1~18.04) ...
Setting up libgcc-7-dev:amd64 (7.5.0-3ubuntu1~18.04) ...
Setting up cpp-7 (7.5.0-3ubuntu1~18.04) ...
Setting up binutils-x86-64-linux-gnu (2.30-21ubuntu1~18.04.5) ...
Setting up cpp (4:7.4.0-1ubuntu2.3) ...
Setting up binutils (2.30-21ubuntu1~18.04.5) ...
Setting up gcc-7 (7.5.0-3ubuntu1~18.04) ...
Setting up gcc (4:7.4.0-1ubuntu2.3) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libjq1 libonig4
The following NEW packages will be installed:
  jq libjq1 libonig4
0 upgraded, 3 newly installed, 0 to remove and 80 not upgraded.
Need to get 276 kB of archives.
After this operation, 930 kB of additional disk space will be used.
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 libonig4 amd64 6.7.0-1 [119 kB]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 libjq1 amd64 1.5+dfsg-2 [111 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 jq amd64 1.5+dfsg-2 [45.6 kB]
Fetched 276 kB in 0s (15.1 MB/s)
Selecting previously unselected package libonig4:amd64.
(Reading database ... 92426 files and directories currently installed.)
Preparing to unpack .../libonig4_6.7.0-1_amd64.deb ...
Unpacking libonig4:amd64 (6.7.0-1) ...
Selecting previously unselected package libjq1:amd64.
Preparing to unpack .../libjq1_1.5+dfsg-2_amd64.deb ...
Unpacking libjq1:amd64 (1.5+dfsg-2) ...
Selecting previously unselected package jq.
Preparing to unpack .../jq_1.5+dfsg-2_amd64.deb ...
Unpacking jq (1.5+dfsg-2) ...
Setting up libonig4:amd64 (6.7.0-1) ...
Setting up libjq1:amd64 (1.5+dfsg-2) ...
Setting up jq (1.5+dfsg-2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
Install golang.
--2021-09-24 09:55:46--  https://dl.google.com/go/go1.13.9.linux-amd64.tar.gz
Resolving dl.google.com (dl.google.com)... 142.250.217.110, 2607:f8b0:400a:806::200e
Connecting to dl.google.com (dl.google.com)|142.250.217.110|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 120139686 (115M) [application/octet-stream]
Saving to: ‘/tmp/go1.13.9.linux-amd64.tar.gz’

go1.13.9.linux-amd64.tar.gz              100%[=============================================================================================>] 114.57M  35.6MB/s    in 3.2s

2021-09-24 09:55:50 (35.6 MB/s) - ‘/tmp/go1.13.9.linux-amd64.tar.gz’ saved [120139686/120139686]

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.
```

and then run the following commands:

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

Output



```
You can proceed to run arktos-up.sh if you want to launch a single-node cluster.
ubuntu@ip-172-31-17-139:~$ echo export PATH=$PATH:/usr/local/go/bin\ >> ~/.profile
ubuntu@ip-172-31-17-139:~$ echo cd \$HOME/go/src/k8s.io/arktos >> ~/.profile
ubuntu@ip-172-31-17-139:~$ source ~/.profile
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ 
```

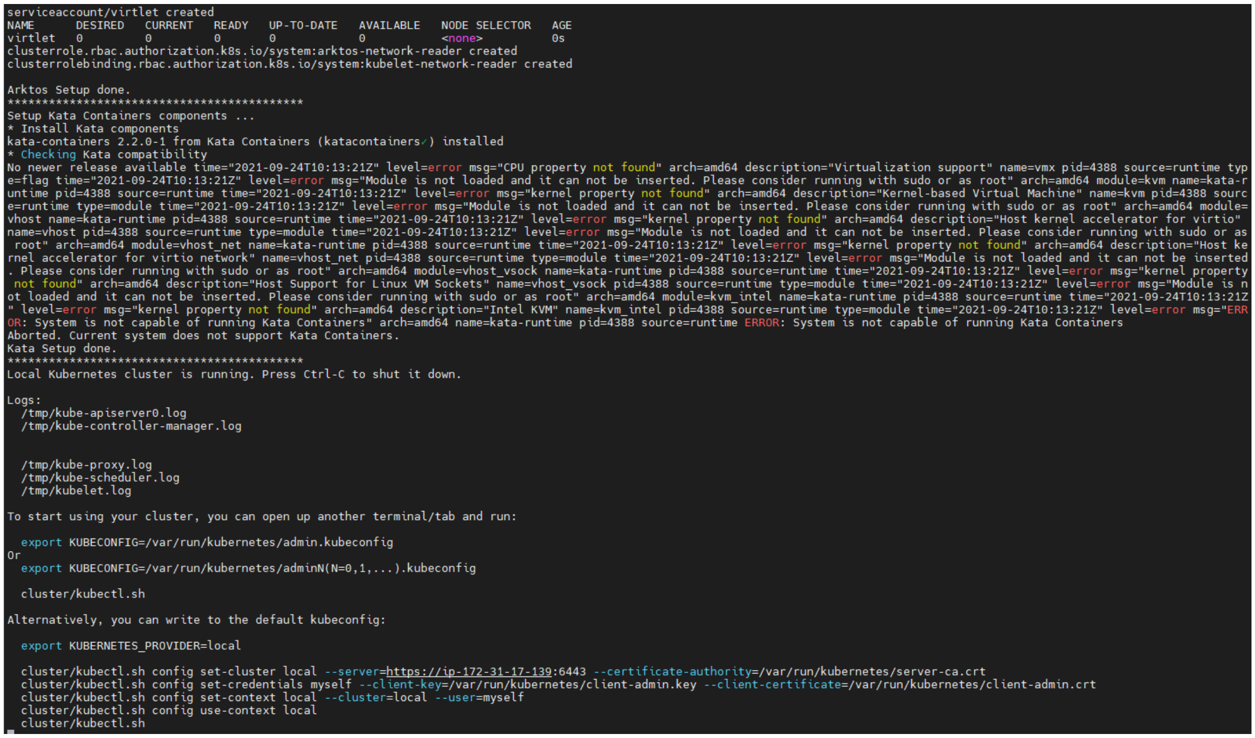
Step-3: Start Arktos cluster

Run following steps to deploy arktos cluster with Mizar as CNI

```
CNIPLUGIN=mizar ./hack/arktos-up.sh
```

Finally we got following output, which indicates that arktos cluster created successfully with Mizar as CNI

Output



```
serviceaccount/virtlet created
NAME      DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
virtlet   0         0         0       0            0           <none>          0s
clusterrole.rbac.authorization.k8s.io/system:arktos-network-reader created
clusterrolebinding.rbac.authorization.k8s.io/system:kubelet-network-reader created

Arktos Setup done.
*******************************************
Setup Kata Containers components ...
* Install Kata components
kata-containers 2.2.0-1 from Kata Containers (katacontainers✓) installed
* Checking Kata compatibility
No newer release available time="2021-09-24T10:13:21Z" level=error msg="CPU property not found" arch=amd64 description="Virtualization support" name=vmx pid=4388 source=runtime typ
e=flag time="2021-09-24T10:13:21Z" level=error msg="Module is not loaded and it can not be inserted. Please consider running with sudo or as root" arch=amd64 module=kvm name=kata-r
untime pid=4388 source=runtime time="2021-09-24T10:13:21Z" level=error msg="kernel property not found" arch=amd64 description="Kernel-based Virtual Machine" name=kvm pid=4388 sourc
e=runtime type=module time="2021-09-24T10:13:21Z" level=error msg="Module is not loaded and it can not be inserted. Please consider running with sudo or as root" arch=amd64 module=
vhost name=kata-runtime pid=4388 source=runtime time="2021-09-24T10:13:21Z" level=error msg="kernel property not found" arch=amd64 description="Host kernel accelerator for virtio"
name=vhost pid=4388 source=runtime type=module time="2021-09-24T10:13:21Z" level=error msg="Module is not loaded and it can not be inserted. Please consider running with sudo or as
 root" arch=amd64 module=vhost_net name=kata-runtime pid=4388 source=runtime time="2021-09-24T10:13:21Z" level=error msg="kernel property not found" arch=amd64 description="Host ke
rnel accelerator for virtio network" name=vhost_net pid=4388 source=runtime type=module time="2021-09-24T10:13:21Z" level=error msg="Module is not loaded and it can not be inserted
. Please consider running with sudo or as root" arch=amd64 module=vhost_vsock name=kata-runtime pid=4388 source=runtime time="2021-09-24T10:13:21Z" level=error msg="kernel property
 not found" arch=amd64 description="Host Support for Linux VM Sockets" name=vhost_vsock pid=4388 source=runtime type=module time="2021-09-24T10:13:21Z" level=error msg="Module is n
ot loaded and it can not be inserted. Please consider running with sudo or as root" arch=amd64 module=kvm_intel name=kata-runtime pid=4388 source=runtime time="2021-09-24T10:13:21Z
" level=error msg="kernel property not found" arch=amd64 description="Intel KVM" name=kvm_intel pid=4388 source=runtime type=module time="2021-09-24T10:13:21Z" level=error msg="ERR
OR: System is not capable of running Kata Containers" arch=amd64 name=kata-runtime pid=4388 source=runtime ERROR: System is not capable of running Kata Containers
Aborted. Current system does not support Kata Containers.
Kata Setup done.
*******************************************
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://ip-172-31-17-139:6443 --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

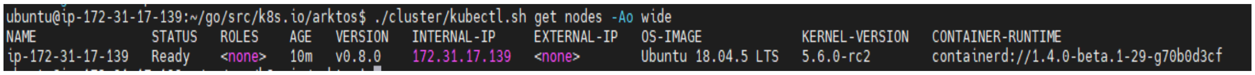
Step-4 Check Cluster health

Open new terminal for same instance and run following commands:

1) Check node status

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

Output

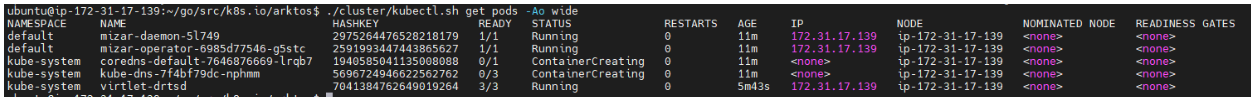


```
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get nodes -Ao wide
NAME               STATUS   ROLES    AGE   VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-17-139   Ready    <none>   10m   v0.8.0    172.31.17.139   <none>        Ubuntu 18.04.5 LTS   5.6.0-rc2        containerd://1.4.0-beta.1-29-g70b0d3cf
```

2) Check pods status

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

Output

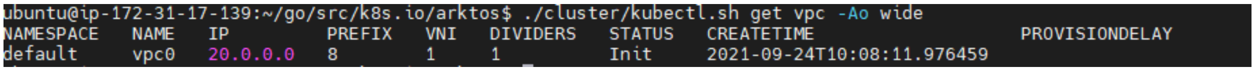


```
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                                 HASHKEY               READY   STATUS              RESTARTS   AGE     IP              NODE                NOMINATED NODE   READINESS GATES
default       mizar-daemon-5l749                   2975264476528218179   1/1     Running             0          11m     172.31.17.139   ip-172-31-17-139    <none>           <none>
default       mizar-operator-6985d77546-g5stc      2591993447443865627   1/1     Running             0          11m     172.31.17.139   ip-172-31-17-139    <none>           <none>
kube-system   coredns-default-7646876669-lrqb7     1940585041135008088   0/1     ContainerCreating   0          11m     <none>          ip-172-31-17-139    <none>           <none>
kube-system   kube-dns-7f4bf79dc-nphmm             5696724946622562762   0/3     ContainerCreating   0          11m     <none>          ip-172-31-17-139    <none>           <none>
kube-system   virtlet-drtsd                        7041384762649019264   3/3     Running             0          5m43s   172.31.17.139   ip-172-31-17-139    <none>           <none>
```

3) Check vpc status

```
./cluster/kubectl.sh get vpc -Ao wide
```

Output

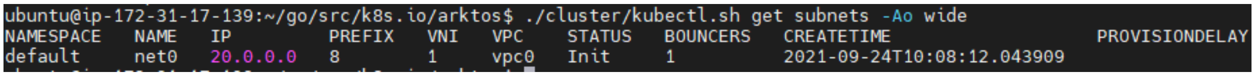


```
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get vpc -Ao wide
NAMESPACE   NAME   IP         PREFIX   VNI   DIVIDERS   STATUS   CREATETIME                   PROVISIONDELAY
default     vpc0   20.0.0.0   8        1     1          Init     2021-09-24T10:08:11.976459
```

4) Check subnets

```
./cluster/kubectl.sh get subnets -Ao wide
```

Output

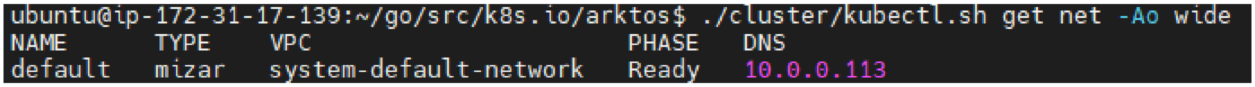


```
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get subnets -Ao wide
NAMESPACE   NAME   IP         PREFIX   VNI   VPC    STATUS   BOUNCERS   CREATETIME                   PROVISIONDELAY
default     net0   20.0.0.0   8        1     vpc0   Init     1          2021-09-24T10:08:12.043909
```

5) Check net

```
./cluster/kubectl.sh get net -Ao wide
```

Output

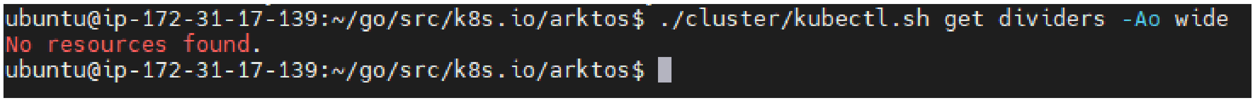


```
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get net -Ao wide
NAME      TYPE    VPC                      PHASE   DNS
default   mizar   system-default-network   Ready   10.0.0.113
```

6) Check dividers

```
./cluster/kubectl.sh get dividers -Ao wide
```

Output

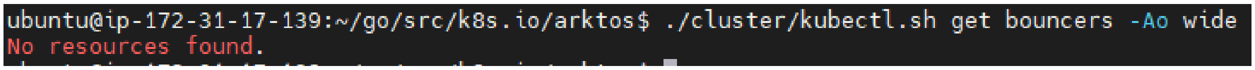


```
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get dividers -Ao wide
No resources found.
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$
```

7) Check bouncers

```
./cluster/kubectl.sh get bouncers -Ao wide
```

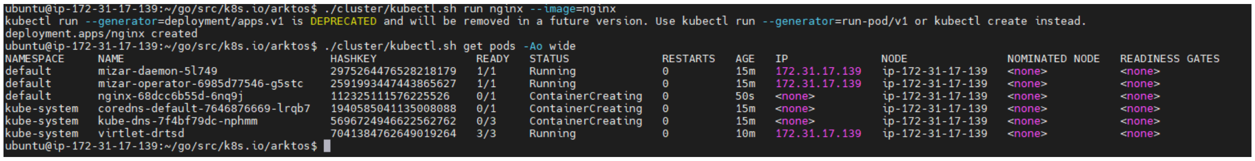
Output



```
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get bouncers -Ao wide
No resources found.
```

8) Pod deployment:

Output



```
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh run nginx --image=nginx
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version. Use kubectl run --generator=run-pod/v1 or kubectl create instead.
deployment.apps/nginx created
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                               HASHKEY               READY   STATUS              RESTARTS   AGE   IP              NODE               NOMINATED NODE   READINESS GATES
default       mizar-daemon-5l749                 2975264476528218179   1/1     Running             0          15m   172.31.17.139   ip-172-31-17-139   <none>           <none>
default       mizar-operator-6985d77546-g5stc    2591993447443865627   1/1     Running             0          15m   172.31.17.139   ip-172-31-17-139   <none>           <none>
default       nginx-68dcc6b55d-6nq9j             112325111576225526    0/1     ContainerCreating   0          50s   <none>          ip-172-31-17-139   <none>           <none>
kube-system   coredns-default-7646876669-lrqb7   1940585041135008088   0/1     ContainerCreating   0          15m   <none>          ip-172-31-17-139   <none>           <none>
kube-system   kube-dns-7f4bf79dc-nphmm           5696724946622562762   0/3     ContainerCreating   0          15m   <none>          ip-172-31-17-139   <none>           <none>
kube-system   virtlet-drtsd                      7041384762649019264   3/3     Running             0          10m   172.31.17.139   ip-172-31-17-139   <none>           <none>
ubuntu@ip-172-31-17-139:~/go/src/k8s.io/arktos$
```

Pod getting stuck in ***ContainerCreating*** *state.*