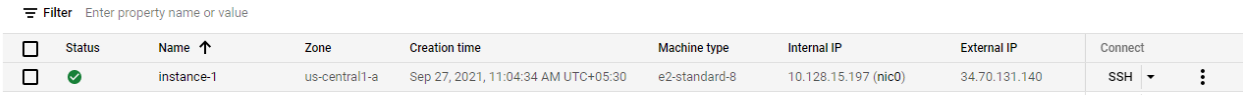
Test report - Deployment of Arktos Cluster with Mizar CNI on GCE

This document captures the steps to deploy an Arktos cluster lab with 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-27.09.2021

Create an instance on GCE

Created instance on GCE



SSH instance with credentials.

Step-1: Update kernel (If required)

To check kernel, run following command

```
uname -a
```

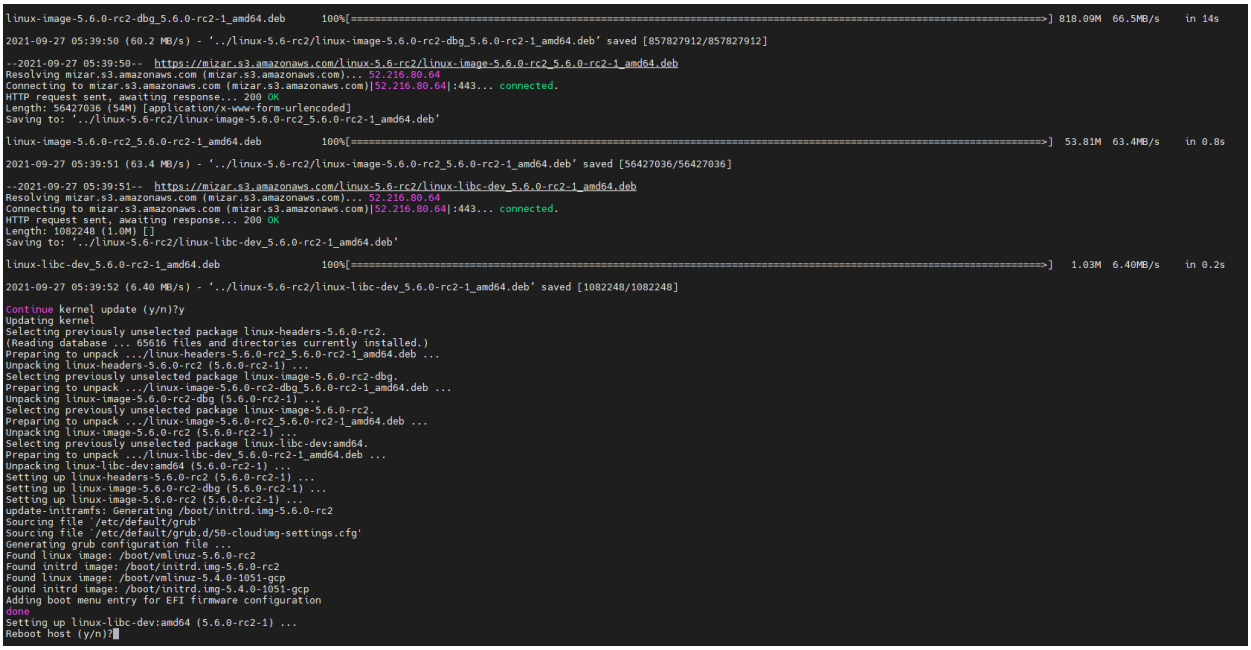
output:

```
ubuntu@instance-1:~$ uname -a
Linux instance-1 5.4.0-1051-gcp #55~18.04.1-Ubuntu SMP Sun Aug 1 20:38:04 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

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:

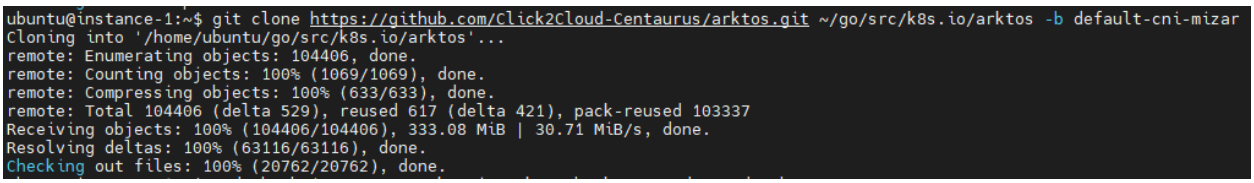


Step-2: Install dependencies

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

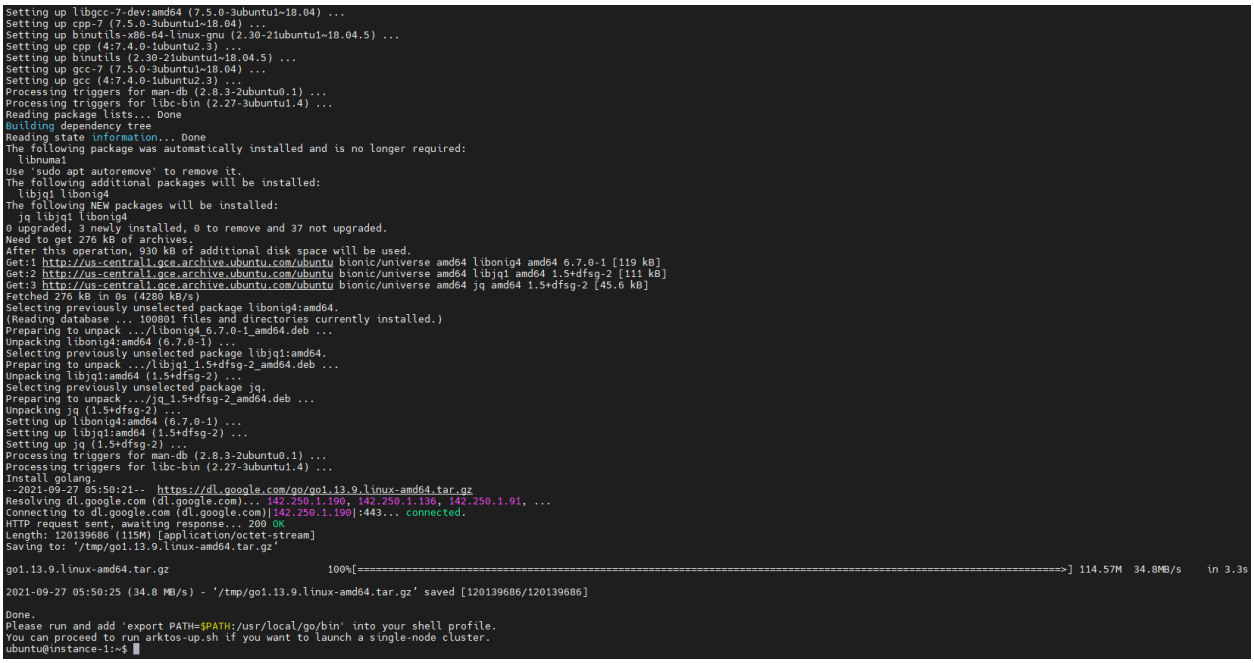
```
git clone https://github.com/Click2Cloud-Centaurus/arktos.git ~/go/src/k8s.io/arktos -b default-cni-mizar
```

output:




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

output:

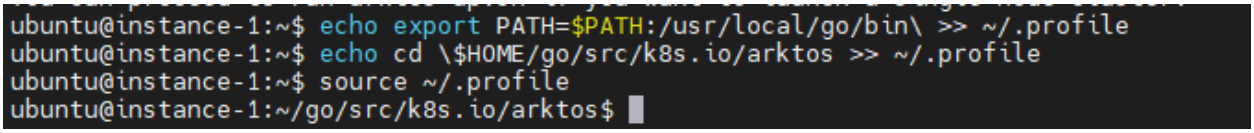



```
echo export PATH=$PATH:/usr/local/go/bin\ >> ~/.profile
```

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

```
source ~/.profile
```

output:



Step-3: Start Arktos cluster

Run following step 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:

```
clusterrolebinding.rbac.authorization.k8s.io/virtlet-crd created
serviceaccount/virtlet created
NAME      DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
virtlet   1         1         0       1            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 2.2.1 from Kata Containers (katacontainers✓) installed
* Checking Kata compatibility
No newer release available time="2021-09-27T06:05:59Z" level=error msg="CPU property not found" arch=amd64 description="Virtualization support" name=vmx pid=30026 source=runtime type=flag time="2021-09-27T06:05
:59Z" 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-runtime pid=30026 source=runtime time="2021-09-27T06:05:59Z"
 level=error msg="kernel property not found" arch=amd64 description="Kernel-based Virtual Machine" name=kvm pid=30026 source=runtime type=module time="2021-09-27T06:05:59Z" 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=30026 source=runtime time="2021-09-27T06:05:59Z" level=error msg="kernel property not fou
nd" arch=amd64 description="Host kernel accelerator for virtio" name=vhost pid=30026 source=runtime type=module time="2021-09-27T06:05:59Z" level=error msg="Module is not loaded and it can not be inserted. Plea
se consider running with sudo or as root" arch=amd64 module=vhost_net name=kata-runtime pid=30026 source=runtime time="2021-09-27T06:05:59Z" level=error msg="kernel property not found" arch=amd64 description="H
ost kernel accelerator for virtio network" name=vhost_net pid=30026 source=runtime type=module time="2021-09-27T06:05:59Z" level=error msg="Module is not loaded and it can not be inserted. Please consider runni
ng with sudo or as root" arch=amd64 module=vhost_vsock name=kata-runtime pid=30026 source=runtime time="2021-09-27T06:05:59Z" level=error msg="kernel property not found" arch=amd64 description="Host Support for
 Linux VM Sockets" name=vhost_vsock pid=30026 source=runtime type=module time="2021-09-27T06:05:59Z" level=error msg="Module is not loaded and it can not be inserted. Please consider running with sudo or as roo
t" arch=amd64 module=kvm_intel name=kata-runtime pid=30026 source=runtime time="2021-09-27T06:05:59Z" level=error msg="kernel property not found" arch=amd64 description="Intel KVM" name=kvm_intel pid=30026 sour
ce=runtime type=module time="2021-09-27T06:05:59Z" level=error msg="ERROR: System is not capable of running Kata Containers" arch=amd64 name=kata-runtime pid=30026 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://instance-1:6443 --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@instance-1:~/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
instance-1   Ready    <none>   6m32s   v0.8.0    10.128.15.197    <none>        Ubuntu 18.04.5 LTS   5.6.0-rc2        containerd://1.4.0-beta.1-29-g70b0d3cf
ubuntu@instance-1:~/go/src/k8s.io/arktos$
```

2) Check pods status

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

Output

```
ubuntu@instance-1:~/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-8bsxq                  7735477782612895179     1/1     Running             0          8m3s   10.128.15.197   instance-1   <none>           <none>
default       mizar-operator-6985d77546-82nqg     16508330519071948888    1/1     Running             0          8m3s   10.128.15.197   instance-1   <none>           <none>
default       pod-767686bb54-5bg6s                5774664824995397753     0/1     ContainerCreating   0          6m45s  <none>          instance-1   <none>           <none>
kube-system   coredns-default-54895b5bfc-qhpkm    6167013444726112653     0/1     ContainerCreating   0          8m3s   <none>          instance-1   <none>           <none>
kube-system   kube-dns-7f4bf79dc-zcjdn            4360096557658664667     0/3     ContainerCreating   0          8m3s   <none>          instance-1   <none>           <none>
kube-system   virtlet-d8gb4                       3236308487394719128     3/3     Running             0          8m2s   10.128.15.197   instance-1   <none>           <none>
ubuntu@instance-1:~/go/src/k8s.io/arktos$
```

3) Check vpc status

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

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get vpcs -Ao wide
NAMESPACE   NAME   IP          PREFIX   VNI   DIVIDERS   STATUS        CREATETIME                   PROVISIONDELAY
default     vpc0   20.0.0.0    8        1     1          Provisioned   2021-09-27T06:47:02.104597   42.064971
```

4) Check subnets

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

Output

```
ubuntu@instance-1:~/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   Provisioned   1          2021-09-27T06:47:02.208350   62.121736
ubuntu@instance-1:~/go/src/k8s.io/arktos$
```

5) Check net

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

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get net -Ao wide
NAME      TYPE    VPC                       PHASE   DNS
default   mizar   system-default-network    Ready   10.0.0.164
ubuntu@instance-1:~/go/src/k8s.io/arktos$
```

6) Check dividers

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

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get dividers -Ao wide
NAMESPACE   NAME                                   VPC    IP   MAC   DROPLET      STATUS        CREATETIME                   PROVISIONDELAY
default     vpc0-d-adb42b77-2823-402d-bbbf-904ff3ad551f   vpc0              instance-1   Provisioned   2021-09-27T06:47:44.154017   0.272695
ubuntu@instance-1:~/go/src/k8s.io/arktos$
```

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get dividers -Ao wide
No resources found.
ubuntu@instance-1:~/go/src/k8s.io/arktos$
```

7) Check bouncers

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

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get bouncers -Ao wide
NAMESPACE   NAME                                    VPC    NET    IP    MAC   DROPLET      STATUS        CREATETIME                    PROVISIONDELAY
default     net0-b-48f8d246-7aac-486e-b367-ffed1ec49a5e   vpc0   net0               instance-1   Provisioned   2021-09-27T06:48:04.321609   0.95841
ubuntu@instance-1:~/go/src/k8s.io/arktos$
```

8) Pod deployment:

Output

```
ubuntu@instance-1:~/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-8bsxq                  7735477782612895179   1/1     Running             0          8m3s    10.128.15.197   instance-1   <none>           <none>
default       mizar-operator-6985d77546-82nqg     1650833051907194888   1/1     Running             0          8m3s    10.128.15.197   instance-1   <none>           <none>
default       pod-767686bb54-5bg6s                5774664824995397753   0/1     ContainerCreating   0          6m45s   <none>          instance-1   <none>           <none>
kube-system   coredns-default-54895b5bfc-qhpkm    6167013444726112653   0/1     ContainerCreating   0          8m3s    <none>          instance-1   <none>           <none>
kube-system   kube-dns-7f4bf79dc-zcjdn            4360096557658664667   0/3     ContainerCreating   0          8m3s    <none>          instance-1   <none>           <none>
kube-system   virtlet-d8gb4                       3236308487394719128   3/3     Running             0          8m2s    10.128.15.197   instance-1   <none>           <none>
```

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