Deploy arktos with mizar cni test

The preferred OS is Ubuntu 18.04.

If you are using AWS, the recommended instance size is t2.2xlarge and the storage size is 128GB or more.

Steps:

1. Check the kernel version:

Command:

```
uname -a
```

Output:

```
ubuntu@ip-172-31-31-176:~$ uname -a
Linux ip-172-31-31-176 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
```

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

Command:

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

sudo bash kernelupdate.sh
```

Output:

```
ubuntu@ip-172-31-31-176:~$ wget https://raw.githubusercontent.com/CentaurusInfra/mizar/dev-next/kernelupdate.sh
--2021-10-19 12:04:18--  https://raw.githubusercontent.com/CentaurusInfra/mizar/dev-next/kernelupdate.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 791 [text/plain]
Saving to: 'kernelupdate.sh'

kernelupdate.sh                   100%[==========================================================================>]     791  --.-KB/s    in 0s

2021-10-19 12:04:19 (41.2 MB/s) - 'kernelupdate.sh' saved [791/791]

ubuntu@ip-172-31-31-176:~$
ubuntu@ip-172-31-31-176:~$ sudo bash kernelupdate.sh
--2021-10-19 12:04:19--  https://mizar.s3.amazonaws.com/linux-5.6-rc2/linux-headers-5.6.0-rc2_5.6.0-rc2-1_amd64.deb
Resolving mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)... 52.217.70.204
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.217.70.204|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7621020 (7.3M) []
Saving to: '../linux-5.6-rc2/linux-headers-5.6.0-rc2_5.6.0-rc2-1_amd64.deb'

linux-headers-5.6.0-rc2_5.6.0-rc2-1 100%[==========================================================================>]   7.27M  5.25MB/s    in 1.4s

2021-10-19 12:04:20 (5.25 MB/s) - '../linux-5.6-rc2/linux-headers-5.6.0-rc2_5.6.0-rc2-1_amd64.deb' saved [7621020/7621020]

--2021-10-19 12:04:20--  https://mizar.s3.amazonaws.com/linux-5.6-rc2/linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb
Resolving mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)... 52.217.36.156
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.217.36.156|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 857827912 (818M) [application/x-www-form-urlencoded]
Saving to: '../linux-5.6-rc2/linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb'

age-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64  95%[==========================================================================>   ] 783.57M  8.78MB/s    eta 5s
```

```
ubuntu@ip-172-31-31-176:~$ uname -a
Linux ip-172-31-31-176 5.6.0-rc2 #1 SMP Tue Feb 25 18:54:05 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

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

Command:

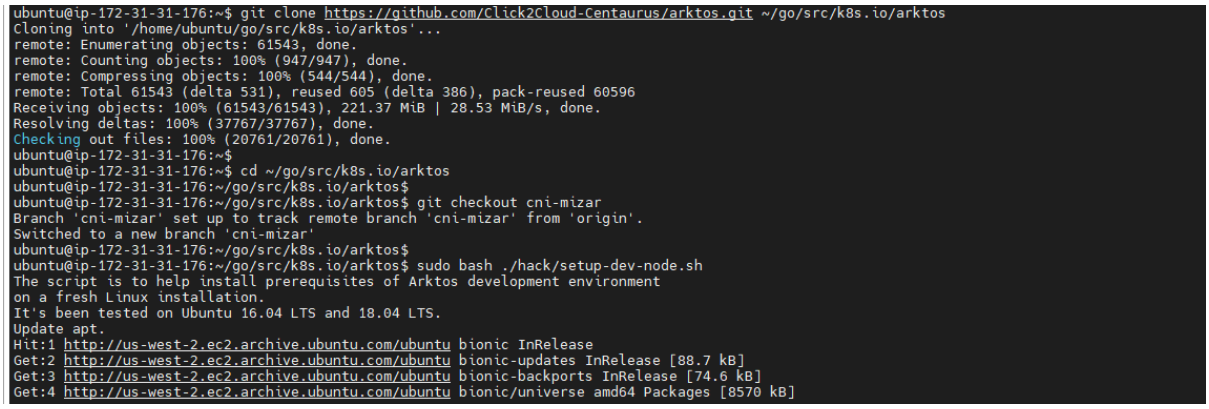
```
git clone https://github.com/Click2Cloud-Centaurus/arktos.git
~/go/src/k8s.io/arktos

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

git checkout cni-mizar

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

Output:



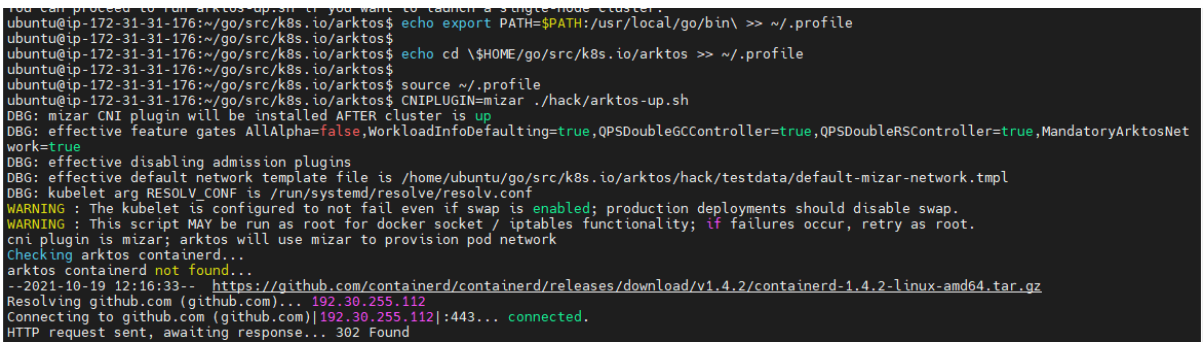
3 Command:

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

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

source ~/.profile
```

Output:

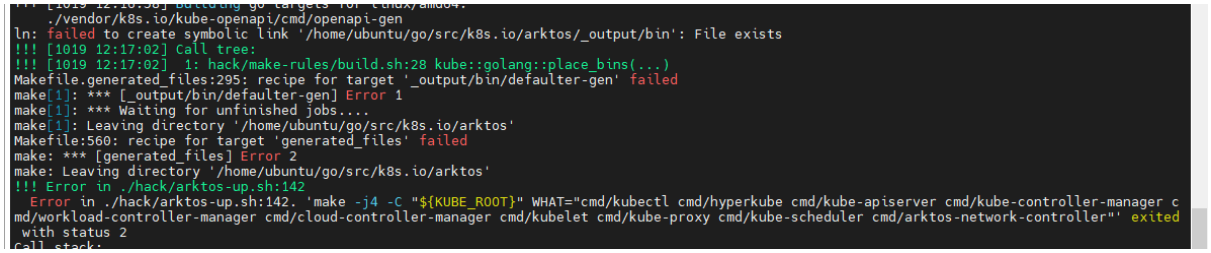


4 Start Arktos cluster with error

Command:

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

Output:

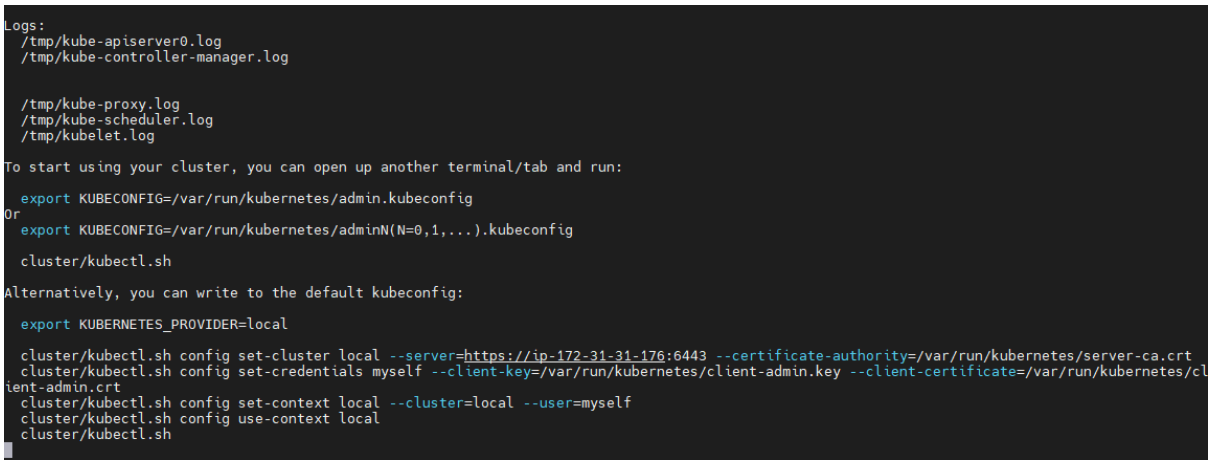


Start Arktos cluster without error

Command:

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

Output:



5. Verify mizar pods i.e. mizar-operator and mizar-daemon pods are in running state, for that run:

Command:

```
./cluster/kubectl.sh get pods
```

Output:

```
ubuntu@ip-172-31-31-176:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get pods
NAME                                HASHKEY                READY   STATUS    RESTARTS   AGE
mizar-daemon-pfpb8                  7084511080620833354    1/1     Running   0          11m
mizar-operator-79bf96959-fk4hj      5618739021028152324    1/1     Running   0          11m
ubuntu@ip-172-31-31-176:~/go/src/k8s.io/arktos$
```

deploy test pod

Command

```
./cluster/kubectl.sh apply -f https://raw.githubusercontent.com/Click2Cloud-Centaurus/Documentation/main/test-yamls/test_pods.yaml
```

Output:

```
ubuntu@ip-172-31-31-176:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh apply -f https://raw.githubusercontent.com/Click2Cloud-Centaurus/Documentation/main/test-yamls/test_pods.yaml
pod/netpod1 created
pod/netpod2 created
ubuntu@ip-172-31-31-176:~/go/src/k8s.io/arktos$
```

Command:

```
./cluster/kubectl.sh get pods -A
```

```
ubuntu@ip-172-31-31-176:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get pods -A
NAMESPACE     NAME                                HASHKEY                READY   STATUS              RESTARTS   AGE
default       mizar-daemon-pfpb8                  7084511080620833354    1/1     Running             0          16m
default       mizar-operator-79bf96959-fk4hj      5618739021028152324    1/1     Running             0          16m
default       netpod1                             4164102727579144295    0/1     ContainerCreating   0          95s
default       netpod2                             5510621305704266765    0/1     ContainerCreating   0          95s
kube-system   coredns-default-5f7f97949d-2lmmd    2531655630568084883    0/1     ContainerCreating   0          16m
kube-system   kube-dns-7f4bf79dc-nz2qs            3275302932980669175    0/3     ContainerCreating   0          16m
kube-system   virtlet-6d5ks                       6121837199323958649    3/3     Running             0          11m
```

Comment:

Pods stuck in creating container state

get VPCs

Command

```
./cluster/kubectl.sh get vpc -A
```

Output:

```
ubuntu@ip-172-31-31-176:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get vpc -A
NAMESPACE   NAME   IP          PREFIX   VNI   DIVIDERS   STATUS   CREATETIME                    PROVISIONDELAY
default     vpc0   20.0.0.0    8        1     1          Init     2021-10-19T12:21:14.102260
ubuntu@ip-172-31-31-176:~/go/src/k8s.io/arktos$
```

get subnet

command

```
./cluster/kubectl.sh get subnet -A
```

Output: