Test report - Arktos deployment with Mizar CNI

Create an instance on AWS

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.

SSH instance with credentials:

output:

Step 1:- Check the kernel version and update if version below 5.6:

`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`

```
See man sudo_root for details.

ubuntu@ip-172-31-20-46:~$ uname -a
Linux ip-172-31-20-46 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-172-31-20-46:~$ wget https://raw.githubusercontent.com/CentaurusInfra/mizar/dev-next/kernelupdate.sh
--2021-10-13 08:39:23--  https://raw.githubusercontent.com/CentaurusInfra/mizar/dev-next/kernelupdate.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|: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

2021-10-13 08:39:23 (37.0 MB/s) - 'kernelupdate.sh' saved [791/791]

ubuntu@ip-172-31-20-46:~$ sudo bash kernelupdate.sh
--2021-10-13 08:39:24--  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)... 52.216.112.35
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.216.112.35|: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_amd64.de 100%[===================================================>]   7.27M  5.17MB/s    in 1.4s

2021-10-13 08:39:26 (5.17 MB/s) - '../linux-5.6-rc2/linux-headers-5.6.0-rc2_5.6.0-rc2-1_amd64.deb' saved [7621020/7621020]

--2021-10-13 08:39:26--  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)... 52.217.196.41
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.217.196.41|: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'

linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64. 100%[===================================================>] 818.09M  4.32MB/s    in 1m 42s

2021-10-13 08:41:09 (8.01 MB/s) - '../linux-5.6-rc2/linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb' saved [857827912/857827912]

--2021-10-13 08:41:09--  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)... 52.216.30.36
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.216.30.36|: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  6.92MB/s    in 8.3s

2021-10-13 08:41:17 (6.50 MB/s) - '../linux-5.6-rc2/linux-image-5.6.0-rc2_5.6.0-rc2-1_amd64.deb' saved [56427036/56427036]

--2021-10-13 08:41:17--  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)... 52.217.200.105
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.217.200.105|: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  2.61MB/s    in 0.4s

2021-10-13 08:41:18 (2.61 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
```

```
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  2.61MB/s    in 0.4s

2021-10-13 08:41:18 (2.61 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)?y
Rebooting

Remote side unexpectedly closed network connection
```

Step 2 :- Clone the Arktos repository and install the required dependencies:

git clone https://github.com/CentaurusInfra/arktos.git ~/go/src/k8s.io/arktos.git

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

```
ubuntu@ip-172-31-20-46:~$ git clone https://github.com/CentaurusInfra/arktos.git ~/go/src/k8s.io/arktos
Cloning into '/home/ubuntu/go/src/k8s.io/arktos'...
remote: Enumerating objects: 104097, done.
remote: Counting objects: 100% (655/655), done.
remote: Compressing objects: 100% (438/438), done.
remote: Total 104097 (delta 290), reused 381 (delta 194), pack-reused 103442
Receiving objects: 100% (104097/104097), 208.18 MiB | 25.95 MiB/s, done.
Resolving deltas: 100% (62784/62784), done.
Checking out files: 100% (20761/20761), done.
ubuntu@ip-172-31-20-46:~$ sudo bash $HOME/go/src/k8s.io/arktos/hack/setup-dev-node.sh
The script is to help install prerequisites of Arktos development environment
on a fresh Linux installation.
It's been tested on Ubuntu 16.04 LTS and 18.04 LTS.
Update apt.
```

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

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

source ~/.profile

Step 3 :- Start Arktos cluster

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

```
Done.
Please run and add 'export PATH=$PATH:/usr/local/go/bin' into your shell profile.
You can proceed to run arktos-up.sh if you want to launch a single-node cluster.
ubuntu@ip-172-31-20-46:~$ echo export PATH=$PATH:/usr/local/go/bin\ >> ~/.profile
ubuntu@ip-172-31-20-46:~$ echo cd \$HOME/go/src/k8s.io/arktos >> ~/.profile
ubuntu@ip-172-31-20-46:~$ source ~/.profile
ubuntu@ip-172-31-20-46:~/go/src/k8s.io/arktos$ ./hack/arktos-up.sh
DBG: Flannel CNI plugin will be installed AFTER cluster is up
DBG: effective feature gates AllAlpha=false,WorkloadInfoDefaulting=true,QPSDoubleGCController=true,QPSDoubleRSController=true,MandatoryArktosNetwork=true
DBG: effective disabling admission plugins
DBG: effective default network template file is /home/ubuntu/go/src/k8s.io/arktos/hack/testdata/default-flat-network.tmpl
DBG: kubelet arg RESOLV_CONF is /run/systemd/resolve/resolv.conf
WARNING : The kubelet is configured to not fail even if swap is enabled; production deployments should disable swap.
WARNING : This script MAY be run as root for docker socket / iptables functionality; if failures occur, retry as root.
cni plugin is bridge; arktos will use bridge to provision pod network
Ensuring firewall to allow traffic forward by default
-P FORWARD DROP
-P FORWARD ACCEPT
Ensuring minimum cni plugin installation...
installing cni plugin binaries
./
```

Deployment successfully done.

output:

4. Leave the "arktos-up.sh" terminal and open another terminal to the master node. Verify mizar pods i.e. mizar-operator and mizar-daemon pods are in running state,

./cluster/kubectl.sh create -f https://raw.githubusercontent.com/CentaurusInfra/mizar/dev-next/etc/deploy/deploy.mizar.yaml

Step-4:- Check Cluster health

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

./cluster/kubectl.sh get nodes

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

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

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

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

./cluster/kubectl.sh get net -Ao wide
<mark>output:</mark>