

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: 27.09.2021

Created an instance on AWS

<input type="checkbox"/>	Centaurus-test...	i-0e3ab90cde5badc86	Running		t2.2xlarge	2/2 checks passed	No alarms	+	us-west-2b	ec2-34-212-0-179.us-w...	34.212.0.179	-
--------------------------	-------------------	---------------------	----------------------	--	------------	--------------------------------	-----------	---	------------	--------------------------	--------------	---

SSH instance using credentials

Step-1: Update kernel version

- Check kernel version:

```
uname -a
```

Output

```
ubuntu@ip-172-31-17-53:~$ uname -a
Linux ip-172-31-17-53 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-172-31-17-53:~$
```

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%[=====] 018.09M 12.8MB/s in 1m 40s
2021-09-27 08:43:39 (8.20 MB/s) - '../linux-5.6-rc2/linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb' saved [857827912/857827912]

--2021-09-27 08:43:39-- 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)... 52.216.17.0
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)[52.216.17.0]: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 7.09MB/s in 8.3s
2021-09-27 08:43:48 (6.48 MB/s) - '../linux-5.6-rc2/linux-image-5.6.0-rc2_5.6.0-rc2-1_amd64.deb' saved [56427036/56427036]

--2021-09-27 08:43:48-- 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)... 52.217.202.17
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)[52.217.202.17]: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 1.98MB/s in 0.5s
2021-09-27 08:43:49 (1.98 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-172-31-17-53:~$ 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: 104412, done.
remote: Counting objects: 100% (1075/1075), done.
remote: Compressing objects: 100% (639/639), done.
remote: Total 104412 (delta 533), reused 616 (delta 421), pack-reused 103337
Receiving objects: 100% (104412/104412), 333.08 MiB | 20.28 MiB/s, done.
Resolving deltas: 100% (63120/63120), 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 (2.30-2ubuntu1~18.04.5) ...
Setting up libcilkrts: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 (2.30-2ubuntu1~18.04.5) ...
Setting up cpp (4:7.4.0-1ubuntu2.3) ...
Setting up binutils (2.30-2ubuntu1~18.04.5) ...
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 (21.6 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.
-2021-09-27 08:55:23- https://dl.google.com/go/go1.13.9.linux-amd64.tar.gz
Resolving dl.google.com (dl.google.com)... 172.217.14.238, 2607:f8b0:4000:803::200e
Connecting to dl.google.com (dl.google.com)[172.217.14.238]: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  9.60MB/s   in 13s

2021-09-27 08:55:36 (8.94 MB/s) - '/tmp/go1.13.9.linux-amd64.tar.gz' saved [120139686/120139686]

Done.
Please run and add 'export PATH=$PATH:/usr/local/go/bin' into your shell profile.
You can proceed to run arktos-up.sh if you want to launch a single-node cluster.
ubuntu@ip-172-31-17-53:~$
```

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

```
ubuntu@ip-172-31-17-53:~$ echo export PATH=$PATH:/usr/local/go/bin\ >> ~/.profile
ubuntu@ip-172-31-17-53:~$ echo cd \"$HOME/go/src/k8s.io/arktos >> ~/.profile
ubuntu@ip-172-31-17-53:~$ source ~/.profile
ubuntu@ip-172-31-17-53:~/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

```

clusterrolebinding.rbac.authorization.k8s.io/virtlet-crd created
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 2.2.1 from Kata Containers (katacontainers.io) installed
  * checking Kata compatibility
No newer release available time="2021-09-27T09:11:13Z" level=error msg="CPU property not found" arch=amd64 description="Virtualization support" name=vmx pid=5275 source=runtime type=flag time="2021-09-27T09:11:13Z" 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=5275 source=runtime time="2021-09-27T09:11:13Z" level=error msg="kernel property not found" arch=amd64 description="Kernel-based Virtual Machine" name=kvm pid=5275 source=runtime type=module time="2021-09-27T09:11:13Z" level=error msg="Module is not loaded and it can not be inserted. Please consider running with sudo or as root" arch=amd64 module=host name=kata-runtime pid=5275 source=runtime time="2021-09-27T09:11:13Z" level=error msg="kernel property not found" arch=amd64 description="Host kernel accelerator for virtio" name=vhost pid=5275 source=runtime type=module time="2021-09-27T09:11:13Z" level=error msg="Module is not loaded and it can not be inserted. Please consider running with sudo or as root" arch=amd64 module=host_net name=kata-runtime pid=5275 source=runtime time="2021-09-27T09:11:13Z" level=error msg="kernel property not found" arch=amd64 description="Host kernel accelerator for virtio network" name=vhost_net pid=5275 source=runtime type=module time="2021-09-27T09:11:13Z" 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=5275 source=runtime time="2021-09-27T09:11:13Z" level=error msg="kernel property not found" arch=amd64 description="Host Support for Linux VM Sockets" name=vsock pid=5275 source=runtime type=module time="2021-09-27T09:11:13Z" 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_intel name=kata-runtime pid=5275 source=runtime type=module time="2021-09-27T09:11:13Z" level=error msg="kernel property not found" arch=amd64 description="Intel KVM" name=kvm_intel pid=5275 source=runtime type=module time="2021-09-27T09:11:13Z" level=error msg="ERROR: System is not capable of running Kata Containers" arch=amd64 name=kata-runtime pid=5275 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/kubernetess/admin.kubeconfig
or
export KUBECONFIG=/var/run/kubernetess/adminNN(N=0,1,...).kubeconfig
cluster/kubect1.sh

Alternatively, you can write to the default kubeconfig:

export KUBERNETES_PROVIDER=local

cluster/kubect1.sh config set-cluster local --server=https://ip-172-31-17-53:6443 --certificate-authority=/var/run/kubernetess/server-ca.crt
cluster/kubect1.sh config set-credentials myself --client-key=/var/run/kubernetess/client-admin.key --client-certificate=/var/run/kubernetess/client-admin.crt
cluster/kubect1.sh config set-context local --cluster=local --user=myself
cluster/kubect1.sh config use-context local
cluster/kubect1.sh

```

Step-4 Check Cluster health

1) Check node status

Output

```
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$ ./cluster/kubectrl.sh get nodes -Ao wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
ip-172-31-17-53	Ready	<none>	7m4s	v0.8.0	172.31.17.53	<none>	Ubuntu 18.04.5 LTS	5.6.0-rc2	containerd://1.4.0-beta.1-29-g70b0d3cf

```
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$
```

2) Check pods status

Output

```
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$ ./cluster/kubectrl.sh get pods -Ao wide
NAMESPACE    NAME                HASHKEY              RESTARTS    AGE    IP                NODE                NOMINATED NODE    READINESS GATES
default      daemon-daemon-6rst5  6342655719541791352  1/1         Running  0      8m45s            ip-172-31-17-53    <none>             <none>
default      miz-pzoperator-69n5d  8514324606060505077  1/1         Running  0      8m44s            ip-172-31-17-53    <none>             <none>
kube-system  corondn-default-68478  2396573282786857384  0/1         ContainerCreating  0      8m44s            ip-172-31-17-53    <none>             <none>
kube-system  kube-dns-7f4bf79dc-kvptm  5389920393315309850  0/3         ContainerCreating  0      8m44s            ip-172-31-17-53    <none>             <none>
kube-system  virtlet-42pdc        7909396300100950985  3/3         Running    0      3m16s            ip-172-31-17-53    ip-172-31-17-53    <none>
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$
```

3) Check vpc status

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

Output

```
ubuntu@ip-172-31-17-53:~/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    2021-09-27T09:05:55.551525
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$
```

4) Check subnets

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

Output

```
ubuntu@ip-172-31-17-53:~/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         2021-09-27T09:05:55.618266
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$
```

5) Check net

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

Output

```
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get net -Ao wide
NAME      TYPE      VPC              PHASE  DNS
default  mizar     system-default-network  Ready  10.0.0.67
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$
```

6) Check dividers

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

Output

```
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get dividers -Ao wide
No resources found.
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$
```

7) Check bouncers

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

Output

```
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh get bouncers -Ao wide
No resources found.
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$
```

8) Pod deployment:

Output

```
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$ ./cluster/kubectl.sh run ubuntu --image=ubuntu
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/ubuntu created
ubuntu@ip-172-31-17-53:~/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-6srt5	6342655719541791352	1/1	Running	0	32m	172.31.17.53	ip-172-31-17-53	<none>	<none>
default	mizar-operator-6985d77546-xkkfm	8514324608605505077	1/1	Running	0	32m	172.31.17.53	ip-172-31-17-53	<none>	<none>
default	ubuntu-58f49b45d7-xr6xp	7930265081919824615	0/1	ContainerCreating	0	3m31s	<none>	ip-172-31-17-53	<none>	<none>
kube-system	coredns-default-68478578c4-qtzzf	2396573228768573784	0/1	ContainerCreating	0	32m	<none>	ip-172-31-17-53	<none>	<none>
kube-system	kube-dns-7f4bf79dc-kvptm	5389920393315309850	0/3	ContainerCreating	0	32m	<none>	ip-172-31-17-53	<none>	<none>
kube-system	virtlet-42pdc	7909396300100950985	3/3	Running	0	27m	172.31.17.53	ip-172-31-17-53	<none>	<none>

```
ubuntu@ip-172-31-17-53:~/go/src/k8s.io/arktos$
```

Pod getting stuck in **ContainerCreating** state.