

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-24.09.2021

Create an instance on GCE

Created instance on GCE

<input type="checkbox"/>	Status	Name ↑	Zone	Creation time	Machine type	Internal IP	External IP	Connect
<input type="checkbox"/>	✓	instance-1	us-central1-a	Sep 24, 2021, 12:52:35 PM UTC+05:30	e2-standard-8	10.128.0.59 (nic0)	35.232.221.201	SSH ▾ ⋮

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
ubuntu@instance-1:~$
```

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:

```

linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_amd64.deb 100%[=====] 818.09M 56.6MB/s in 14s
2021-09-24 07:46:58 (57.6 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 07:46:58-- https://mizar.s3.amazonaws.com/linux-5.6-rc2/linux-image-5.6.0-rc2-1_amd64.deb
Resolving mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)... 52.216.10.35
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)[52.216.10.35]: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-rc2-1_amd64.deb'

linux-image-5.6.0-rc2-1_amd64.deb 100%[=====] 53.81M 57.1MB/s in 0.9s
2021-09-24 07:46:59 (57.1 MB/s) - './linux-5.6-rc2/linux-image-5.6.0-rc2-1_amd64.deb' saved [56427036/56427036]

--2021-09-24 07:46:59-- 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)... 52.216.10.35
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)[52.216.10.35]: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 5.24MB/s in 0.2s
2021-09-24 07:46:59 (5.24 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 ... 65616 files and directories currently installed.)
Preparing to unpack .../linux-headers-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-5.4.0-1051-gcp
Found initrd image: /boot/initrd.img-5.4.0-1051-gcp
Adding boot menu entry for EFI firmware configuration
done
Setting up linux-libc-dev:amd64 (5.6.0-rc2-1) ...
Reboot host (y/n)?

```

Step-2: Install dependencies

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
```

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

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

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

```
source ~/.profile
```

output:

```

go1.13.9.linux-amd64.tar.gz 100%[=====]
2021-09-24 08:38:18 (36.9 MB/s) - '/tmp/go1.13.9.linux-amd64.tar.gz' saved [120139686/120139686]

Done.
Please run and add 'export PATH=$PATH:/usr/local/go/bin' into your shell profile.
You can proceed to run arktos-up.sh if you want to launch a single-node cluster.
ubuntu@instance-1:~$ echo export PATH=$PATH:/usr/local/go/bin\ >> ~/.profile
ubuntu@instance-1:~$ echo cd \"$HOME/go/src/k8s.io/arktos >> ~/.profile
ubuntu@instance-1:~$ source ~/.profile
ubuntu@instance-1:~$ go/src/k8s.io/arktos$

```

Step-3: Start Arktos cluster

Login to instance and 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

```
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 2.20-1 from Kata Containers (katacontainers/) installed
* Checking Kata compatibility
No newer release available time="2021-09-24T08:54:06Z" level=error msg="CPU property not found" arch=amd64 description="Virtualization support" name=vmx pid=10170 source=runtime type=flag time="2021-09-24T08:54:06Z" 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=10170 source=runtime type=module time="2021-09-24T08:54:06Z" level=error msg="Host kernel accelerator for virtio" arch=amd64 module=vhost name=kata-runtime pid=10170 source=runtime type=module time="2021-09-24T08:54:06Z" 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=10170 source=runtime type=module time="2021-09-24T08:54:06Z" level=error msg="Host Support for Linux VM Sockets" arch=amd64 module=vsock name=kata-runtime pid=10170 source=runtime type=module time="2021-09-24T08:54:06Z" 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=10170 source=runtime type=module time="2021-09-24T08:54:06Z" level=error msg="Kernel property not found" arch=amd64 description="Intel KVM" name=kvm_intel pid=10170 source=runtime type=module time="2021-09-24T08:54:06Z" 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=10170 source=runtime type=module time="2021-09-24T08:54:06Z" level=error msg="Kernel property not found" arch=amd64 description="Kernel-based Virtual Machine" name=kvm pid=10170 source=runtime type=module time="2021-09-24T08:54:06Z" level=error msg="ERROR: System is not capable of running Kata Containers" arch=amd64 name=kata-runtime pid=10170 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/admin(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://instance-1:6443 --certificate-authority=/var/run/kubernetes/server-ca.crt
cluster/kubect1.sh config set-credentials myself --client-key=/var/run/kubernetes/client-admin.key --client-certificate=/var/run/kubernetes/client-admin.crt
cluster/kubect1.sh config set-context local --cluster=local --user=myself
cluster/kubect1.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/kubect1.sh get nodes -Ao wide
```

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubect1.sh get nodes -Ao wide
NAME    STATUS    ROLES    AGE    VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE    KERNEL-VERSION    CONTAINER-RUNTIME
instance-1  Ready    <none>    9m21s  v0.8.0     10.128.0.59    <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/kubect1.sh get pods -Ao wide
```

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectll.sh get pods -Ao wide
NAMESPACE NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
default mizar-daemon-lbvlr 1/1 Running 0 9m44s 10.128.0.59 instance-1 <none> <none>
default mizar-operator-6985d77546-mg6wz 1/1 Running 0 9m45s 10.128.0.59 instance-1 <none> <none>
kube-system coredns-default-5cfc5df5b4-cjrjr 0/1 ContainerCreating 0 9m45s <none> instance-1 <none> <none>
kube-system kube-dns-7f4bf79dc-fgm9t 0/3 ContainerCreating 0 9m45s <none> instance-1 <none> <none>
kube-system virtlet-pj44m 0/3 Init:0/1 0 2m34s 10.128.0.59 instance-1 <none> <none>
```

3) Check vpc status

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

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectll.sh get vpc -Ao wide
NAMESPACE NAME IP PREFIX VNI DIVIDERS STATUS CREATETIME PROVISIONDELAY
default vpc0 20.0.0.0 8 1 1 Init 2021-09-24T08:47:20.067177
```

4) Check subnets

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

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectll.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 2021-09-24T08:47:20.155373
```

5) Check net

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

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectll.sh get net -Ao wide
NAME TYPE VPC PHASE DNS
default mizar system-default-network Ready 10.0.0.110
```

6) Check dividers

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

Output

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

7) Check bouncers

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

Output

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

8) Pod deployment:

Output

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectrl.sh run nginx --image=nginx
kubectrl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version. Use kubectrl run --generator=run-pod/v1 or kubectrl create instead.
deployment.apps/nginx created
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectrl.sh get pods -Ao wide
```

NAMESPACE	NAME	HASHKEY	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
default	mizar-daemon-l8v1r	2468310505181364751	1/1	Running	0	14m	10.128.0.59	instance-1	<none>	<none>
default	mizar-operator-6985d77546-mg6wz	6018773957802313882	1/1	Running	0	14m	10.128.0.59	instance-1	<none>	<none>
default	nginx-68dcc6b55d-6xv5d	1882270975195899213	0/1	ContainerCreating	0	7s	<none>	instance-1	<none>	<none>
kube-system	coredns-default-5cfc5df5b4-cjrjr	6776417514841129512	0/1	ContainerCreating	0	14m	<none>	instance-1	<none>	<none>
kube-system	kube-dns-7f4bf79dc-fgm9t	7368901326361209507	0/3	ContainerCreating	0	14m	<none>	instance-1	<none>	<none>
kube-system	virtlet-pj44m	1799651634972541635	3/3	Running	0	7m26s	10.128.0.59	instance-1	<none>	<none>

```
ubuntu@instance-1:~/go/src/k8s.io/arktos$ ./cluster/kubectrl.sh get pods -Ao wide
```

Pod getting stuck in **ContainerCreating** state.