Test report - Deployment of Arktos Cluster without Mizar CNI on GCP

This document captures the steps to deploy an Arktos cluster lab without Mizar CNI. The machine in this lab used are **32 GB RAM, 8 vCPUs (**e2-standard-8**), 128 GB storage and Ubuntu 18.04 LTS.**

Install golang 1.13.9

Date-17 Dec. 2021

Step-1: Update kernel (If required)

To check kernel, run following command

```
uname -a
```

```
ubuntu@instance-4:~$ uname -a
Linux instance-4 5.4.0-1058-gcp #62~18.04.1-Ubuntu SMP Mon Nov 15 07:49:04 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
```

```
Continue kernel update (y/n)?y
Updating kernel
Selecting previously unselected package linux-headers-5.6.0-rc2.
(Reading database ... 71529 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.
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-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) ...
Preparing to unpack .../linux-libc-dev_5.6.0-rc2-1_amd64.deb ...
Unpacking linux-libc-dev:amd64 (5.6.0-rc2-1) over (4.15.0-163.171) ...
Setting up linux-headers-5.6.0-rc2 (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
Searching for GRUB installation directory ... found: /boot/grub
Searching for default file ... found: /boot/grub/default
Testing for an existing GRUB menu.lst file ... found: /boot/grub/menu.lst
Searching for splash image ... none found, skipping ...
Found kernel: /vmlinuz-4.15.0-55-generic
Replacing config file /run/grub/menu.lst with new version
Found kernel: /vmlinuz-5.6.0-rc2
Found kernel: /vmlinuz-4.15.0-55-generic
Replacing config file /run/grub/menu.lst with new version
Updating /boot/grub/menu.lst ... done
```

Step-2: Install dependencies

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

mkdir -p $GOPATH/src/github.com

cd $GOPATH/src/github.com

git clone https://github.com/CentaurusInfra/arktos

cd arktos

sudo bash hack/setup-dev-node.sh

make

```
Running copyright check for repo: /src/github.com/arktos, logging to _output/ArktosCopyrightTool.log
/src/github.com/arktos /src/github.com/arktos
warning: inexact rename detection was skipped due to too many files.
warning: you may want to set your diff.renameLimit variable to at least 3067 and retry the command.
/src/github.com/arktos
/src/github.com/arktos /src/github.com/arktos
warning: inexact rename detection was skipped due to too many files.
warning: you may want to set your diff.renameLimit variable to at least 3067 and retry the command.
/src/github.com/arktos
Inspecting copyright files, writing logs to _output/ArktosCopyrightTool.log
Done.
```

Run Arktos

The easiest way to run Arktos is to bring up a single-node cluster in your local development box:
cd $GOPATH/src/github.com/arktos

hack/arktos-up.sh

```
****************************************
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://node-b: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
```

1) Check nodes status:

./cluster/kubectl.sh get nodes

```
root@instance-4:/src/github.com/arktos# ./cluster/kubectl.sh get nodes
NAME         STATUS   ROLES    AGE     VERSION
instance-4   Ready    <none>   6m47s   v0.9.0
```

2) Check pods status:

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

```
root@instance-4:/src/github.com/arktos# ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                              HASHKEY              READY   STATUS    RESTARTS   AGE     IP           NODE         NOMINATED NODE
EADINESS GATES
kube-system   coredns-default-688c9c8c6-gwwdw   116986811144598106   1/1     Running   0          9m3s    10.88.0.3    instance-4   <none>
none>
kube-system   kube-dns-554c5866fc-kzpwr         215970174082571243   3/3     Running   0          9m3s    10.88.0.2    instance-4   <none>
none>
kube-system   virtlet-s58xj                     6233753631735440747   2/3     Running   0          3m54s   10.128.0.2   instance-4   <none>
none>
```

Deploy test pods:

Command:

./cluster/kubectl.sh apply -f https://github.com/Click2Cloud-Centaurus/Documentation/blob/main/test-yamls/test_pods.yaml

Check deployed pods:

Command:

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

Output

```
root@instance-4:/src/github.com/arktos# ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                              HASHKEY              READY   STATUS    RESTARTS   AGE     IP           NODE         NOMINATED NODE
EADINESS GATES
default       netpod1                           9109225830706253997   1/1     Running   0          21s     10.88.0.4    instance-4   <none>
none>
default       netpod2                           1020684756913439652   1/1     Running   0          21s     10.88.0.5    instance-4   <none>
none>
kube-system   coredns-default-688c9c8c6-gwwdw   116986811144598106   1/1     Running   0          14m     10.88.0.3    instance-4   <none>
none>
kube-system   kube-dns-554c5866fc-kzpwr         215970174082571243   3/3     Running   0          14m     10.88.0.2    instance-4   <none>
none>
kube-system   virtlet-s58xj                     6233753631735440747   3/3     Running   0          9m20s   10.128.0.2   instance-4   <none>
none>
```

Check ping deployed pods:

Command:

./cluster/kubectl.sh exec -it netpod1 ping 10.88.0.5

./cluster/kubectl.sh exec -it netpod2 ping 10.88.0.4

```
root@instance-4:/src/github.com/arktos# ./cluster/kubectl.sh exec -it netpod1 ping 10.88.0.5
PING 10.88.0.5 (10.88.0.5) 56(84) bytes of data.
64 bytes from 10.88.0.5: icmp_seq=1 ttl=64 time=0.131 ms
64 bytes from 10.88.0.5: icmp_seq=2 ttl=64 time=0.109 ms
64 bytes from 10.88.0.5: icmp_seq=3 ttl=64 time=0.083 ms
64 bytes from 10.88.0.5: icmp_seq=4 ttl=64 time=0.107 ms
64 bytes from 10.88.0.5: icmp_seq=5 ttl=64 time=0.085 ms
64 bytes from 10.88.0.5: icmp_seq=6 ttl=64 time=0.076 ms
64 bytes from 10.88.0.5: icmp_seq=7 ttl=64 time=0.089 ms
^C
--- 10.88.0.5 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 127ms
rtt min/avg/max/mdev = 0.076/0.097/0.131/0.018 ms
root@instance-4:/src/github.com/arktos# ./cluster/kubectl.sh exec -it netpod2 ping 10.88.0.4
PING 10.88.0.4 (10.88.0.4) 56(84) bytes of data.
64 bytes from 10.88.0.4: icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from 10.88.0.4: icmp_seq=2 ttl=64 time=0.077 ms
64 bytes from 10.88.0.4: icmp_seq=3 ttl=64 time=0.123 ms
^C
--- 10.88.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 38ms
rtt min/avg/max/mdev = 0.077/0.095/0.123/0.021 ms
```