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

This document captures the steps to deploy an Arktos cluster lab without 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.

Install golang 1.13.9

Date-13 Dec. 2021

Step-1: Update kernel (If required)

To check kernel, run following command

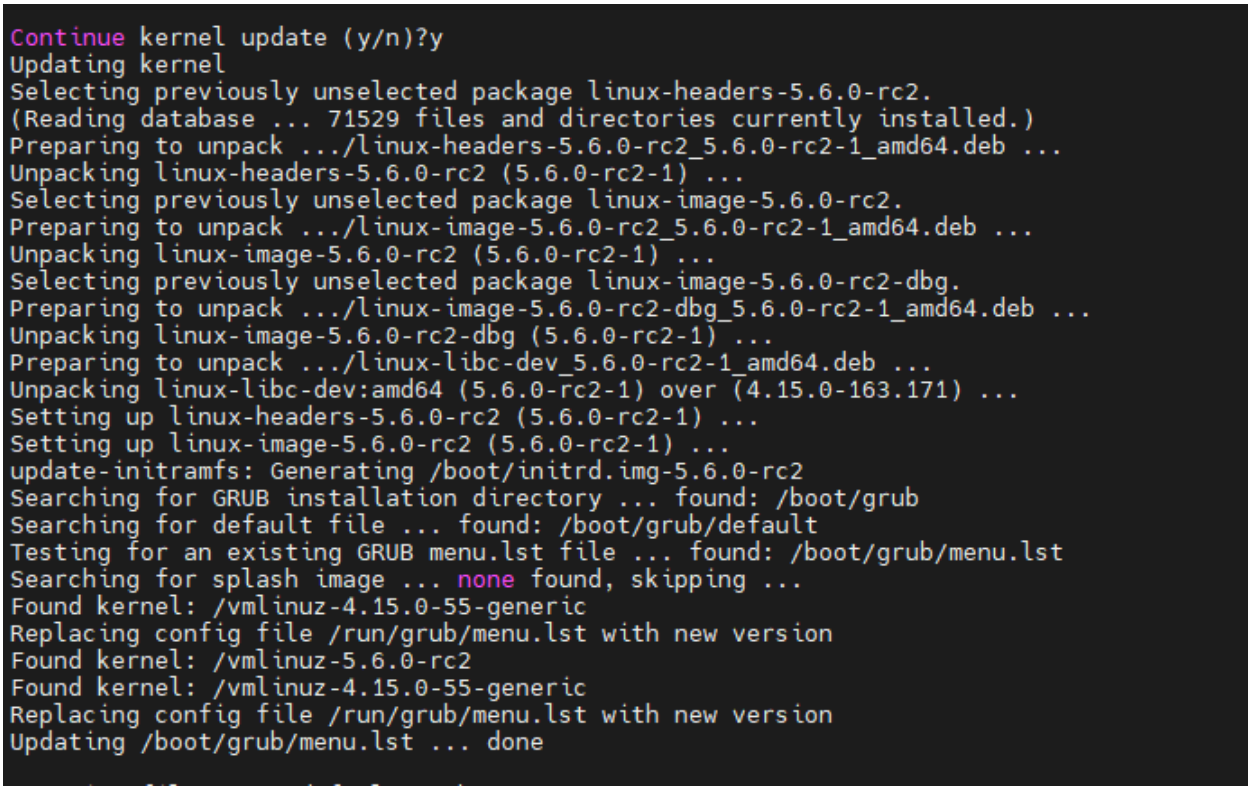
```
uname -a
```

```
root@node-b:/src/github.com/arktos# uname -a
Linux node-b 4.15.0-55-generic #60-Ubuntu SMP Tue Jul 2 18:22:20 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
root@node-b:/src/github.com/arktos#
```

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
```



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

```
root@node-b:/src/github.com/arktos# make
+++ [1216 06:53:43] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/deepcopy-gen
+++ [1216 06:54:02] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/defaulter-gen
+++ [1216 06:54:17] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/conversion-gen
+++ [1216 06:54:40] Building go targets for linux/amd64:
    ./vendor/k8s.io/kube-openapi/cmd/openapi-gen
+++ [1216 06:55:03] Building go targets for linux/amd64:
    ./vendor/github.com/go-bindata/go-bindata/go-bindata
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.
+++ [1216 06:55:27] Building go targets for linux/amd64:
    cmd/kube-proxy
    cmd/kube-apiserver
```

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@node-b:/src/github.com/arktos# ./cluster/kubectl.sh get nodes
NAME      STATUS   ROLES    AGE     VERSION
node-b    Ready    <none>   5m23s   v0.9.0
```

2) Check pods status:

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

```
root@node-b:/src/github.com/arktos# ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                              HASHKEY              READY   STATUS    RESTARTS   AGE     IP             NODE     NOMINATED NODE   READINESS GATES
kube-system   coredns-default-7df6d5588c-nfwq7  4239183630222468240  1/1     Running   0          5m57s   10.88.0.2      node-b   <none>           <none>
kube-system   kube-dns-554c5866fc-7695r         2016961547314934692  3/3     Running   0          5m57s   10.88.0.3      node-b   <none>           <none>
kube-system   virtlet-kx7l6                     4729556394073104847  3/3     Running   0          4m8s    192.168.2.51   node-b   <none>           <none>
root@node-b:/src/github.com/arktos#
```

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@node-b:/src/github.com/arktos# ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                              HASHKEY              READY   STATUS    RESTARTS   AGE   IP             NODE     NOMINATED NODE   READINESS GATES
default       netpod1                           3879396692362615395  1/1     Running   0          87s   10.88.0.4      node-b   <none>           <none>
default       netpod2                           2368972774505434648  1/1     Running   0          87s   10.88.0.5      node-b   <none>           <none>
kube-system   coredns-default-7df6d5588c-nfwq7  4239183630222468240  1/1     Running   0          12m   10.88.0.2      node-b   <none>           <none>
kube-system   kube-dns-554c5866fc-7695r         2016961547314934692  3/3     Running   0          12m   10.88.0.3      node-b   <none>           <none>
kube-system   virtlet-kx7l6                     4729556394073104847  3/3     Running   0          11m   192.168.2.51   node-b   <none>           <none>
root@node-b:/src/github.com/arktos#
```

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@node-b:/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=0.181 ms
64 bytes from 10.88.0.5: icmp_seq=2 ttl=64 time=0.159 ms
^C
--- 10.88.0.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 0.159/0.170/0.181/0.011 ms
root@node-b:/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=0.175 ms
64 bytes from 10.88.0.4: icmp_seq=2 ttl=64 time=0.148 ms
^C
--- 10.88.0.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 15ms
rtt min/avg/max/mdev = 0.148/0.161/0.175/0.018 ms
root@node-b:/src/github.com/arktos#
```