Test report - Deployment of Arktos Cluster without Mizar CNI on Premise (Community code)

This document captures the steps to deploy an Arktos cluster lab without Mizar CNI. The machine in this lab used are **16 GB RAM, 8 vCPUs, 128 GB storage and Ubuntu 18.04 LTS.**

Date-31 Dec. 2021

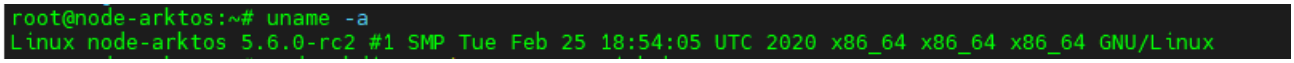
Step-1: Update kernel (If required)

To check kernel, run following command

uname -a

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:

sudo mkdir -p $GOPATH/src/github.com

cd $GOPATH/src/github.com

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

cd arktos

sudo bash hack/setup-dev-node.sh

sudo -i

cd $GOPATH/src/github.com/arktos

export PATH=$PATH:/usr/local/go/bin

make

```
root@node-arktos:~# sudo mkdir -p $GOPATH/src/github.com
root@node-arktos:~# cd $GOPATH/src/github.com
root@node-arktos:/src/github.com# sudo git clone https://github.com/CentaurusInfra/arktos
Cloning into 'arktos'...
remote: Enumerating objects: 104576, done.
remote: Counting objects: 100% (204/204), done.
remote: Compressing objects: 100% (176/176), done.
remote: Total 104576 (delta 69), reused 59 (delta 28), pack-reused 104372
Receiving objects: 100% (104576/104576), 207.99 MiB | 13.60 MiB/s, done.
Resolving deltas: 100% (63061/63061), done.
Checking out files: 100% (20766/20766), done.
root@node-arktos:/src/github.com# ls
arktos
root@node-arktos:/src/github.com# ufw status
Status: inactive
root@node-arktos:/src/github.com# cd arktos
root@node-arktos:/src/github.com/arktos# sudo bash 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.
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu bionic-security InRelease
```

```
Please run and add 'export PATH=$PATH:/usr/local/go/bin' into your shell profile.
You can proceed to run arktos-up.sh if you want to launch a single-node cluster.
root@node-arktos:/src/github.com/arktos# export PATH=$PATH:/usr/local/go/bin
root@node-arktos:/src/github.com/arktos# sudo -i
root@node-arktos:~# cd $GOPATH/src/github.com/arktos
root@node-arktos:/src/github.com/arktos# make

Can't find 'go' in PATH, please fix and retry.
See http://golang.org/doc/install for installation instructions.

!!! [1231 07:32:53] Call tree:
!!! [1231 07:32:53]  1: hack/run-in-gopath.sh:30 kube::golang::setup_env(...)

Can't find 'go' in PATH, please fix and retry.
See http://golang.org/doc/install for installation instructions.

!!! [1231 07:32:53] Call tree:
!!! [1231 07:32:53]  1: /src/github.com/arktos/hack/lib/golang.sh:775 kube::golang::setup_env(...)
!!! [1231 07:32:53]  2: hack/make-rules/build.sh:27 kube::golang::build_binaries(...)
!!! [1231 07:32:53] Call tree:
!!! [1231 07:32:53]  1: hack/make-rules/build.sh:27 kube::golang::build_binaries(...)
Makefile.generated_files:201: recipe for target '_output/bin/deepcopy-gen' failed
make[1]: *** [_output/bin/deepcopy-gen] Error 1
Makefile:560: recipe for target 'generated_files' failed
make: *** [generated_files] Error 2
root@node-arktos:/src/github.com/arktos# export PATH=$PATH:/usr/local/go/bin
root@node-arktos:/src/github.com/arktos# make
+++ [1231 07:33:52] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/deepcopy-gen
+++ [1231 07:34:22] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/defaulter-gen
+++ [1231 07:34:43] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/conversion-gen
+++ [1231 07:35:25] Building go targets for linux/amd64:
    ./vendor/k8s.io/kube-openapi/cmd/openapi-gen
+++ [1231 07:35:56] Building go targets for linux/amd64:
    ./vendor/github.com/go-bindata/go-bindata/go-bindata
```

Run Arktos

The easiest way to run Arktos is to bring up a single-node cluster in your local development box:
hack/arktos-up.sh

```
* Restart containerd
For kubernetes 1.14: RUN kubectl apply -f https://raw.githubusercontent.com/kata-containers/packaging/master/kata-deploy/k8s-1.14/kata-qemu-runtimeCl
For kubernetes 1.13: RUN kubectl apply -f https://raw.githubusercontent.com/kata-containers/packaging/master/kata-deploy/k8s-1.13/kata-qemu-runtimeCl
create 1.14 runtimeClass by default
runtimeclass.node.k8s.io/kata created
runtimeclass.node.k8s.io/kata-qemu created
A Kata example: RUN kubectl apply -f https://raw.githubusercontent.com/kata-containers/packaging/master/kata-deploy/examples/test-deploy-kata-qemu.ya
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/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-e: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-
  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-arktos:/src/github.com/arktos# ./cluster/kubectl.sh get nodes
NAME           STATUS     ROLES    AGE    VERSION
node-arktos    NotReady   <none>   53m    v0.9.0
root@node-arktos:/src/github.com/arktos#
```

2) Check pods status:

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

```
root@node-arktos:/src/github.com/arktos# ./cluster/kubectl.sh get pods -Ao wide
NAMESPACE     NAME                              HASHKEY             READY   STATUS                       RESTARTS   AGE     IP              NODE
NATED NODE   READINESS GATES
kube-system   coredns-default-7b4cbdf5cd-z9gvj   2739052452219530108   0/1     ContainerCreating            0          53m     <none>          node-arktos
e>           <none>
kube-system   kube-dns-554c5866fc-2zw94          8960008069511500356   0/3     ContainerCreating            0          53m     <none>          node-arktos
e>           <none>
kube-system   virtlet-wbdz2                      2905953027139994912   0/3     Init:CreateContainerConfigError   0          9m57s   192.168.1.211   node-arktos
e>           <none>
root@node-arktos:/src/github.com/arktos#
```

Deployment failed