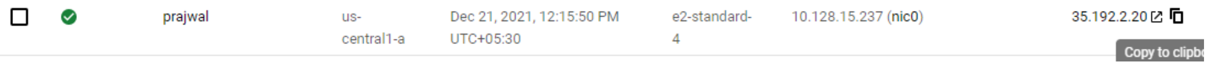
Test report - Deployment of Arktos Cluster with Mizar CNI on GCE

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-21.12.2021

Create an instance on GCE

Created instance on GCE

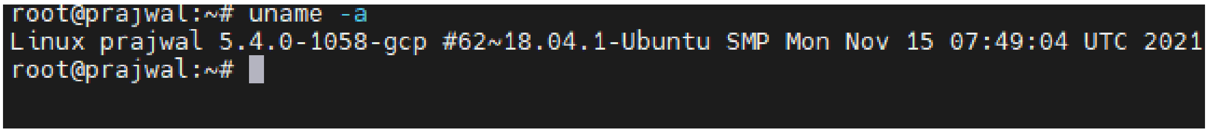


□ ✓ prajwal us-central1-a Dec 21, 2021, 12:15:50 PM UTC+05:30 e2-standard-4 10.128.15.237 (nic0) 35.192.2.20

Step-1: Update kernel (If required)

To check kernel, run following command

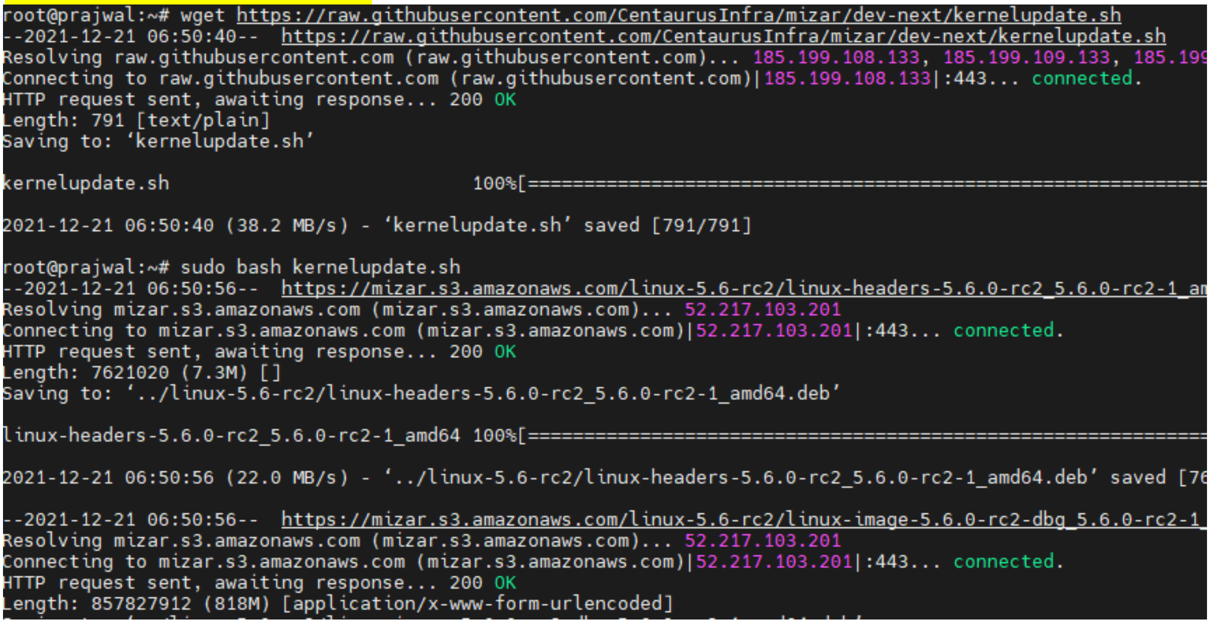
```
uname -a
```



```
root@prajwal:~# uname -a
Linux prajwal 5.4.0-1058-gcp #62~18.04.1-Ubuntu SMP Mon Nov 15 07:49:04 UTC 2021
root@prajwal:~#
```

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
```



```
root@prajwal:~# wget https://raw.githubusercontent.com/CentaurusInfra/mizar/dev-next/kernelupdate.sh
--2021-12-21 06:50:40--  https://raw.githubusercontent.com/CentaurusInfra/mizar/dev-next/kernelupdate.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 791 [text/plain]
Saving to: ‘kernelupdate.sh’

kernelupdate.sh                     100%[==============================================================

2021-12-21 06:50:40 (38.2 MB/s) - ‘kernelupdate.sh’ saved [791/791]

root@prajwal:~# sudo bash kernelupdate.sh
--2021-12-21 06:50:56--  https://mizar.s3.amazonaws.com/linux-5.6-rc2/linux-headers-5.6.0-rc2_5.6.0-rc2-1_am
Resolving mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)... 52.217.103.201
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.217.103.201|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7621020 (7.3M) []
Saving to: ‘../linux-5.6-rc2/linux-headers-5.6.0-rc2_5.6.0-rc2-1_amd64.deb’

linux-headers-5.6.0-rc2_5.6.0-rc2-1_amd64 100%[==============================================================

2021-12-21 06:50:56 (22.0 MB/s) - ‘../linux-5.6-rc2/linux-headers-5.6.0-rc2_5.6.0-rc2-1_amd64.deb’ saved [76

--2021-12-21 06:50:56--  https://mizar.s3.amazonaws.com/linux-5.6-rc2/linux-image-5.6.0-rc2-dbg_5.6.0-rc2-1_
Resolving mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)... 52.217.103.201
Connecting to mizar.s3.amazonaws.com (mizar.s3.amazonaws.com)|52.217.103.201|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 857827912 (818M) [application/x-www-form-urlencoded]
```

Step-2: Install dependencies

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

mkdir -p $GOPATH/src/github.com

cd $GOPATH/src/github.com

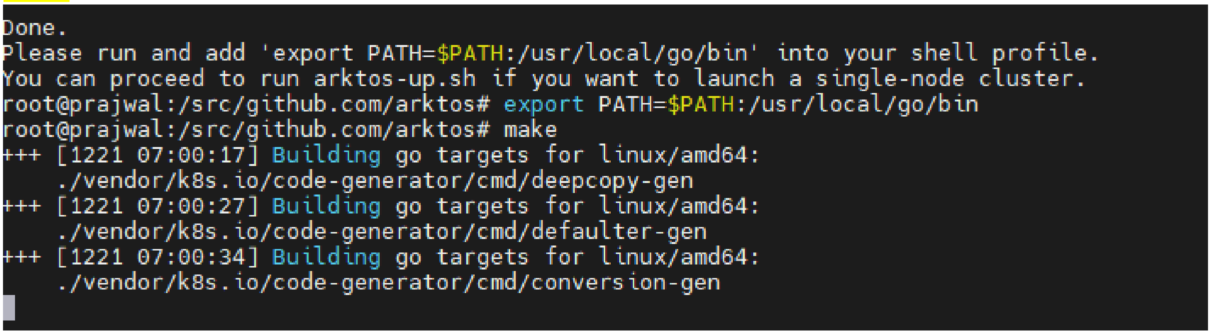
git clone https://github.com/CentaurusInfra/arktos

cd arktos

sudo bash hack/setup-dev-node.sh

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

make

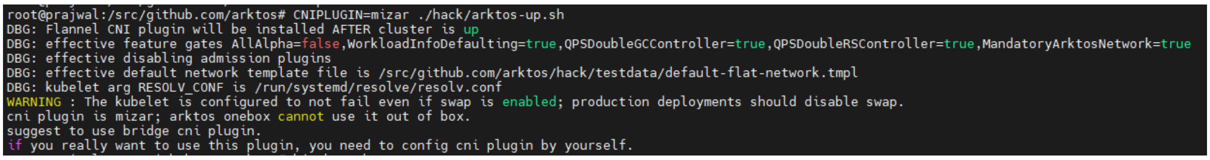


```
Done.
Please run and add 'export PATH=$PATH:/usr/local/go/bin' into your shell profile.
You can proceed to run arktos-up.sh if you want to launch a single-node cluster.
root@prajwal:/src/github.com/arktos# export PATH=$PATH:/usr/local/go/bin
root@prajwal:/src/github.com/arktos# make
+++ [1221 07:00:17] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/deepcopy-gen
+++ [1221 07:00:27] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/defaulter-gen
+++ [1221 07:00:34] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/conversion-gen
```

Step-3: Start Arktos cluster

Run following step to deploy arktos cluster with Mizar as CNI:

```
CNIPLUGIN=mizar ./hack/arktos-up.sh
```



```
root@prajwal:/src/github.com/arktos# CNIPLUGIN=mizar ./hack/arktos-up.sh
DBG: Flannel CNI plugin will be installed AFTER cluster is up
DBG: effective feature gates AllAlpha=false,WorkloadInfoDefaulting=true,QPSDoubleGCController=true,QPSDoubleRSController=true,MandatoryArktosNetwork=true
DBG: effective disabling admission plugins
DBG: effective default network template file is /src/github.com/arktos/hack/testdata/default-flat-network.tmpl
DBG: kubelet arg RESOLV_CONF is /run/systemd/resolve/resolv.conf
WARNING : The kubelet is configured to not fail even if swap is enabled; production deployments should disable swap.
cni plugin is mizar; arktos onebox cannot use it out of box.
suggest to use bridge cni plugin.
if you really want to use this plugin, you need to config cni plugin by yourself.
```

ERROR: **Arktos onebox cannot use it out of box.**