**TERRAFORM**

Table of Contents

[KOPS 3](#_Toc30025630)

[Installing Go for Automation 4](#_Toc30025631)

[Go 4](#_Toc30025632)

[Dependencies 4](#_Toc30025633)

[Some theory 4](#_Toc30025634)

[Set profile 4](#_Toc30025635)

[Test 4](#_Toc30025636)

[Symlink 5](#_Toc30025637)

[Seeing in action 5](#_Toc30025638)

[GITHUB 6](#_Toc30025639)

[Seemless login 6](#_Toc30025640)

[.gitignore 6](#_Toc30025641)

[Installing environment for Terraform course 7](#_Toc30025642)

## KOPS

#!/bin/bash

install\_kops () {

kops create cluster \

--name=kops.azuka.tk \

--state=s3://kops.helm2.devopsinuse.com \

--authorization RBAC \

--zones=eu-central-1a \

--node-count=2 \

--node-size=t2.micro \

--master-size=t2.micro \

--master-count=1 \

--dns-zone=kops.azuka.tk \

--out=azuka\_helm\_terraform \

--target=terraform \

--ssh-public-key=~/.ssh/udemy\_devopsinuse.pub

}

install\_kops

**kops validate cluster --state=s3://kops.helm2.devopsinuse.com**

Using cluster from kubectl context: kops.azuka.tk

Validating cluster kops.azuka.tk

INSTANCE GROUPS

NAME ROLE MACHINETYPE MIN MAX SUBNETS

master-eu-west-1a Master t2.micro 1 1 eu-west-1a

nodes Node t2.medium 2 2 eu-west-1a

NODE STATUS

NAME ROLE READY

ip-172-20-32-69.eu-west-1.compute.internal master True

ip-172-20-61-84.eu-west-1.compute.internal node True

VALIDATION ERRORS

KIND NAME MESSAGE

Machine i-0f707dede942bf751 machine "i-0f707dede942bf751" has not yet joined cluster

**kubectl get nodes**

NAME STATUS ROLES AGE VERSION

ip-172-20-32-69.eu-west-1.compute.internal Ready master 13m v1.11.10

ip-172-20-38-121.eu-west-1.compute.internal NotReady node 19s v1.11.10

ip-172-20-61-84.eu-west-1.compute.internal Ready node 7m v1.11.10

## Installing Go for Automation

### Go

<https://www.linode.com/docs/development/go/install-go-on-ubuntu/>

### Dependencies

<https://golang.github.io/dep/docs/installation.html>

### Some theory

<https://stackoverflow.com/questions/7970390/what-should-be-the-values-of-gopath-and-goroot>

### Set profile

nano ~/.profile

***LOGOUT AND IN***

# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then

if [ -f ~/.bashrc ]; then

. ~/.bashrc

fi

fi

mesg n || true

export GOPATH=/home/michael/terraform/go

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

### Test

root@michael-VirtualBox**:/home/michael/terraform/go/src/terraform-up-and-running**# ll

total 12

drwxr-xr-x 2 root root 4096 Jan 9 18:37 ./

drwxr-xr-x 4 root root 4096 Jan 9 18:25 ../

-rw-r--r-- 1 root root 166 Jan 9 18:36 go\_sanity\_test.go

root@michael-VirtualBox:/home/michael/terraform/go/src/terraform-up-and-running# go test -v

=== RUN TestGoIsWorking

If you see this text, it's working!

--- PASS: TestGoIsWorking (0.00s)

PASS

ok terraform-up-and-running 0.003s

### Symlink

/home/michael/terraform/go/src : where code should be

/home/michael/terraform/terraform-up-and-running-code : original where code actually is

#### Create

ln –s original newlink

ln –s /home/michael/terraform/terraform-up-and-running-code /home/michael/terraform/go/src

##### results:-

lrwxrwxrwx 1 root root 53 Jan 9 22:01 terraform-up-and-running-code -> /home/michael/terraform/terraform-up-and-running-code/

#### remove

cd /home/michael/terraform/go/src

rm terraform-up-and-running-code

### Seeing in action

root@michael-VirtualBox:/home/michael/terraform/go/src/terraform-up-and-running-code/code/terraform/09-terraform-michael/small-modules/test/alb# dep init

Locking in v1.0.0 (792786c) for transitive dep github.com/pmezard/go-difflib

Locking in v1.1.1 (8991bc2) for transitive dep github.com/davecgh/go-spew

Locking in v2.2.7 (1f64d61) for transitive dep gopkg.in/yaml.v2

Using ^0.23.1 as constraint for direct dep github.com/gruntwork-io/terratest

Locking in v0.23.1 (4cf7794) for direct dep github.com/gruntwork-io/terratest

Locking in master (61a8779) for transitive dep golang.org/x/crypto

Locking in master (548cf77) for transitive dep golang.org/x/sys

Locking in master (c0dbc17) for transitive dep golang.org/x/net

Locking in v1.4.0 (221dbe5) for transitive dep github.com/stretchr/testify

root@michael-VirtualBox:/home/michael/terraform/go/src/terraform-up-and-running-code/code/terraform/09-terraform-michael/small-modules/test/alb# dep ensure

### New computer

### Symlink

/home/michael/workspace/go/src : where code should be

/home/michael/workspace/terraform-up-and-running-code : original where code actually is

#### Create

ln –s original newlink

ln –s /home/michael/workspace/terraform-up-and-running-code /home/michael/workspace/go/src

##### results:- cd /home/michael/workspace/go/src

lrwxrwxrwx 1 root root 53 Jan 9 22:01 terraform-up-and-running-code -> /home/michael/workspace/terraform-up-and-running-code/

#### remove

cd /home/michael/workspace/go/src

rm terraform-up-and-running-code

## GITHUB

### Seemless login

Ssh-keygen

Create

-rw------- 1 michael michael 1675 Jan 15 18:57 id\_rsa

-rw-r--r-- 1 michael michael 397 Jan 15 18:57 id\_rsa.pub

Copy contents of pub key to github

<https://github.com/settings/keys>

Follow url

<https://medium.com/@amanze.ogbonna/accessing-pushing-to-github-without-username-and-password-3022feb077fb>

### .gitignore

<https://labs.consol.de/development/git/2017/02/22/gitignore.html>

**git reset --hard origin/master**

## Installing environment for Terraform course

steup vm box

Create vm using image d:/ubuntu

create michael as root

1. sudo adduser michael

give passowrd for osboxes

give details

2. sudo usermod -aG sudo michael

Course : kubernetes for begineers (installing vm)

Access from gitbash

apt-get update

apt-get install openssh-server

run ifconfig : get ipaddress ep03 10.0.2.15

change the NAT port forwarding to this ip

install git

==================================================================================================================

Course Kubernetes-terraform

####################################################

# Initial setups

###################################

step 1 : install

sudo apt-get install python3-pip

sudo pip install awscli

sudo pip3 install awscli

step 2 : configure AWS

Course :See AWS Course (

create new IAM user : Nicola

create group with full admin access

generate access/key from user in AWS. associate (in downloads)

security group addmin to user and you will be given a url for user to log onto console with

#User will be asked to change password on first access

generated in download file

Access key ID,Secret access key

AKIAQXZPV4E5CGHJ3CFW,6XBHXiR2emCmb274OtIdyDJo6SlrVFbgie7AewSk

run aws configure

add access key

add secret key

eu-west-1

michael@osboxes:~$ aws iam get-user

{

"User": {

"Path": "/",

"UserName": "Nicola",

"UserId": "AIDAQXZPV4E5D3PT5JTB5",

"Arn": "arn:aws:iam::051102736698:user/Nicola",

"CreateDate": "2019-09-20T00:40:15Z",

"PasswordLastUsed": "2019-11-12T22:24:44Z",

"Tags": [

{

"Key": "Name",

"Value": "Nicola"

}

]

}

}

step 3 Install Kubectl

############################

#install kubectl

#################################

kubectl-installer.sh

KUBECTL\_BIN=kubectl

function install\_kubectl {

if [ -z $(which $KUBECTL\_BIN) ]

then

curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/$KUBECTL\_BIN

chmod +x ${KUBECTL\_BIN}

sudo mv ${KUBECTL\_BIN} /usr/local/bin/${KUBECTL\_BIN}

else

echo "Kubectl is most likely installed"

fi

}

https://kubernetes.io/docs/tasks/tools/install-kubectl/#install-kubectl-on-linux

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

chmod u+x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

kubectl

step 4 install KOPs

create script and run

function install\_kops {

if [ -z $(which kops) ]

then

curl -LO https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4)/kops-linux-amd64

chmod +x kops-linux-amd64

sudo mv kops-linux-amd64 /usr/local/bin/kops

else

echo "kops is most likely installed"

fi

}

install\_kops

step 5 install Terraform

create script and run

TERRAFORM\_ZIP\_FILE=terraform\_0.11.7\_linux\_amd64.zip

TERRAFORM=https://releases.hashicorp.com/terraform/0.11.7

TERRAFORM\_BIN=terraform

function install\_terraform {

if [ -z $(which $TERRAFORM\_BIN) ]

then

wget ${TERRAFORM}/${TERRAFORM\_ZIP\_FILE}

unzip ${TERRAFORM\_ZIP\_FILE}

sudo mv ${TERRAFORM\_BIN} /usr/local/bin/${TERRAFORM\_BIN}

rm -rf ${TERRAFORM\_ZIP\_FILE}

else

echo "Terraform is most likely installed"

fi

}

install\_terraform

step 6 Create seemless loging for kops script which uses devopsuser

######################

#generate key/pair

######################

ssh-keygen -f ~/.ssh/udemy\_devopsinuse

#Remember these configs contain your access key and secrete. We just use the udemy\_devopsinuse.pub for seemless access