Azure DevOps  
Project 1

By Victor Majumder

# Part 1

## Generate Service Principle for Terraform in Azure

Azure Portal 🡪 Azure Active Directory 🡪 app registrations 🡪 new registration

Name: **Terraform**

Supported Account Types: **Accounts in this organizational directory only (single-tenant)**

Redirect URI: **you should choose "Web" for the URI type. the actual value can be left blank**

Register

Take note of the "Application (client) ID" and the "Directory (tenant) ID"

Azure Portal 🡪 Azure Active Directory 🡪 app registrations 🡪 select the new app created 🡪 Certificates & secrets 🡪 Click on New client secret button 🡪

Description: **Terraform\_Secret**

Expires: **Select the default one**

Add

Take a note of the secret ID

Subscription 🡪 Access control 🡪 Add role assignment 🡪 Specify a role 🡪 search for and select the name of the Service Principal created in Azure Active Directory to assign it this role - then press Save.

Add in env variables

ARM\_CLIENT\_ID="**f0d0b2e8-3a5a-4fd8-94e3-a69b8574c344**"

ARM\_CLIENT\_SECRET="**MXT7Q~NQ\_KtoqFrYOz4t~ya\_RnSpSGPv67yRN**"

ARM\_SUBSCRIPTION\_ID="**781ea48d-7415-4387-85a9-645ae3cecda6**"

ARM\_TENANT\_ID="**fff8ffca-f373-426f-b270-8f2af6e442d8**"

### Installation Terraform in linux

sudo wget <https://releases.hashicorp.com/terraform/1.0.3/terraform_1.0.3_linux_amd64.zip>

sudo unzip terraform\_1.0.3\_linux\_amd64.zip

sudo mv terraform /usr/bin/

terraform version

### Install Azure cli

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

## Create required Vars.tf to create two VMs in Azure

mkdir devops\_project\_1

vi vars.tf



## Create required terraform.tfvars

vi terraform.tfvars



## Create main.tf to create two Linux VMs

vi main.tf



## Init,Plan and Apply Terraform Script

terraform init

terraform validate

terraform plan

terraform apply

## Manually Start Jenkins and configure required Plug-ins and Master Slave Configuration

Login to masternode

Start jenkins

java -jar jenkins.war &

http://13.91.122.139:8080/ (http://<publicip\_of\_masternode>:8080/)

accept pluglins

create account

# Part 2 – Phase 1

## Create Maven Project with Archetype as web application in eclipse

mvn archetype:generate -DgroupId=Devops\_project\_1 -DartifactId=Devops\_project\_1 -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false

below project will be created

/home/adminuser/Devops\_project\_1

## Modify Index.jsp under src/main/webapp to display a custom message

cd src/main/webapp/

vi index.jsp

<html>

<body>

<h2>Hello This is a custom project for Devops Project 1!</h2>

</body>

</html>

## Run Maven clean install in eclipse to check the build and check for .war file in target folder

cd /home/adminuser/Devops\_project\_1

mvn clean install

cd /home/adminuser/Devops\_project\_1/target

you should see a .war file like “Devops\_project\_1.war”

# Part 2 – Phase 2

## Generate Dockerfile under project folder of your app

cd /home/adminuser/Devops\_project\_1

## Modify FORM statement to use tomcat as base image

vi Dockerfile

FROM tomcat:8.0-alpine

ADD target/Devops\_project\_1.war /usr/local/tomcat/webapps/

EXPOSE 8080

CMD [“catalina.sh”, “run”]

## Test the Dockerfile by running Docker build and create a container

sudo docker build -t devops\_project\_1\_app .

## Access the application from container and check it

sudo docker run -p 8080:8080 devops\_project\_1\_app

Run below to check your web app

<http://13.91.122.121:8080/Devops_project_1/> (http://<slave\_public\_ip>:8080/Devops\_project\_1/)

# Part 2 – Phase 3

## Create a github repository and copy repo URL

<https://github.com>

sign up or sign in

username: victormajumder999

email: [ipsita.icy@gmail.com](mailto:ipsita.icy@gmail.com)

create a new repository

repo URL: <https://github.com/victormajumder999/devops_project_1.git>

## In Eclipse convert the app in to a local repo from Team menu share Project Option

git init

git status

git add .

git commit -m my-first-commit

## Commit and Push the code to remote repo

git remote add origin <https://github.com/victormajumder999/devops_project_1.git>

git push -u origin master

it will ask for username. Enter victormajumder999

it will ask for password. Create Personal Access Token on GitHub

From your GitHub account, go to Settings => Developer Settings => Personal Access Token => Generate New Token (Give your password) => Fillup the form => click Generate token => Copy the generated Token, it will be something like ghp\_sFhFsSHhTzMDreGRLjmks4Tzuzgthdvfsrta

# Part 2 – Phase 4

## Modify Project index.jsp, Commit and Push to remote repo

cd /home/adminuser/Devops\_project\_1/src/main/webapp

vi index.jsp

<html>

<body>

<h2>Hello This is a custom project for Devops Project 1 -- update for git test!</h2>

</body>

</html>

git add .

git commit -m my-third-commit

git push -u origin master

## Check for the change in remote Repo

You can see the index.jsp updated here

<https://github.com/victormajumder999/devops_project_1/blob/master/src/main/webapp/index.jsp>

# Part 2 – Phase 5

## In build server configure Ansible manually

Login to build server(initial server from where terraform was executed)

Install python on server

sudo apt-get install -y python-pip

install ansible on server

$ sudo apt-get update

$ sudo apt-get install software-properties-common

$ sudo apt-add-repository ppa:ansible/ansible

$ sudo apt-get update

$ sudo apt-get install ansible

ansible –version

it gets installed in /etc/ansible

change ownership

sudo chown victor\_123: victor\_123/etc/ansible/ -R

## Modify ansible.cfg to use hosts file as inventory

sudo vi /etc/ansible/ansible.cfg

Enable the inventory

sudo vi /etc/ansible/hosts

[slavegroup]

slave ansible\_host=13.91.122.121

[slavegroup:vars]

ansible\_connection=ssh

ansible\_user=adminuser

generate SSH keys

ssh-keygen -m PEM -t rsa -b 4096

copy id\_rsa.pub and paste under authorized\_keys in slave node(also calling it as ansible node)

## install python-pip in ansible server

python-pip was already installed by terraform in part 1

## Using PIP install azure modules in Ansible server

Login to slave node

command to install azure modules

sudo pip install ansible[azure]

## Use the same service Principle created for terraform for ansible to get authenticated to Azure

mkdir .azure

ls -al

cd .azure/

touch credentials

vi credentials

[default]

subscription\_id=781ea48d-7415-4387-85a9-645ae3cecda6

client\_id=f0d0b2e8-3a5a-4fd8-94e3-a69b8574c344

secret=MXT7Q~NQ\_KtoqFrYOz4t~ya\_RnSpSGPv67yRN

tenant=fff8ffca-f373-426f-b270-8f2af6e442d8

## Create a playbook1 to create a vm in azure



## Update the playbook2 to install Docker engine on the VM

Add below in /etc/ansible/hosts

[dockergroup]

docker ansible\_host=40.86.174.112

[dockergroup:vars]

ansible\_connection=ssh

ansible\_user=victor\_123



## create a shell script to get VM ip and updating it in Inventory File

# Part 2 – Phase 7

## From Build server run the playbook1,playbook2 and shell script to test for the required result

ansible-playbook playbook1.yaml

ansible-playbook playbook2.yaml

# Documents and Links referred

<https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/guides/service_principal_client_secret>

<https://medium.com/@yoursshaan2212/terraform-to-provision-multiple-azure-virtual-machines-fab0020b4a6e>

<http://websystique.com/maven/create-a-maven-web-project-with-eclipse/>

<https://stackoverflow.com/questions/19793895/run-mvn-clean-install-in-eclipse>

<https://stackoverflow.com/questions/68775869/support-for-password-authentication-was-removed-please-use-a-personal-access-to>