**Terraform**

For programmatical access to various aws resources, we need to create access key for IAM user in AWS. Steps are as below:

If you already have an IAM user, you can skip to step 5.

1. Login to AWS console >> Services >> IAM and click ‘add users’

2. Click next and select ‘Attach policies directly’

3. Under permission policies, select AdministratorAccess

4. Click ‘create user’

5. On the IAM > Users interface, click on the username you just created.

6. Select ‘security credentials’, under access keys section, you can find the ‘create access key’ option.

7. Create and download the csv file containing the access key and secret key. We will be using the same access key and secret key for terraform and s3 functions.

Download the main.tf file to your terraform server and change the ‘access\_key’ and ‘secret\_key’ values in that file with the your access key created and downloaded to local system earlier.

1. change directory to the folder main.tf file is downloaded.

2. run command:

# terraform init

3. once above command finishes, run:

# terraform plan

4. then run:

# terraform apply

Once the terraform apply command is run and finishes, we can verify the details with the below command:

# terraform show

for eg: run below command to get the public IP of the created server.

# terraform show | grep public\_ip

the main.tf file provided will create one ec2 instance, one rds instance with mysql engine, and an s3 bucket. A new key-pair with the name ‘tf-key-pair’ will be created and will be downloaded to the current working directory.

**Ansible**

Download the ansible-playbook directory to your ansible server. Download the private key ‘tf-key-pair’ created earlier and get the public IP of the server using the “terraform show” command.

On the ansible server, update the ansible hosts file with the ec2 instance IP address.

Open the file, /etc/ansible/hosts and add below two lines (replace the IP address with your ec2 instance IP)

[webservers]

wordpress-server ansible\_host=35.175.234.125

Now for ansible to communicate to its slaves, ssh public key has to be added to ~/.ssh/authorizedkeys file in the slave server.

For that we create an ssh key-pair from the ansible server with the below command:

# ssh-keygen -t rsa

After that connect to the server using the ‘tf-key-pair’ (make sure to update private key permission to 400.

# ssh -i tf-key-pair [ubuntu@IP-AD-DR-RES](mailto:ubuntu@IP-AD-DR-RES)

copy the ~/.ssh/id\_rsa.pub content from the ansible server and paste it under the existing public keys in the ~/.ssh/authorizedkeys file. Save the file.

Ansible can now communicate with the slave using the key.

Now, change directory to ansible playbook folder we have downloaded, then execute the below command:

# ansible-playbook playbook.yml -l wordpress-server -u ubuntu

The domain name specified in the ~/vars/default.yml is ‘mydomain.com’ so you may need to update the/etc/hosts file with the IP and domain to view the wodpress site.

**BASH Script**

For AWS cli in the script to work properly, we need to install and configure it on the LAMP server.

run below commands to install aws cli on your LAMP server.

# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

# unzip awscliv2.zip

# sudo ./aws/install

Once installed we need to configure it with the below command:

#aws configure

Provide the below details to configure it properly:

AWS Access Key ID --> your access key created earlier

AWS Secret Access Key --> Secret key

Default region name --> us-east-2

Default output format --> json

For mysql to work without password within the script, we need to add the username, password for the mysqldump and for mysql client.

Open the mysql conf file located at:/etc/mysql/my.cnf and add below lines. Make sure to change the host with the endpoint of your rds instance.

[mysqldump]

user = root

password = mysql\_root\_password

[client]

user = myrdsuser

password = myrdspassword

host = myrdsinstance.cj8roipihi.us-east-1.rds.amazonaws.com

Create a folder /root/backup and download the bash script(backup.sh) to the folder.

Give the executable permission to the script with the below command:

# chmod +x backup.sh

Now add a chron job for root user to run it every day at 8PM IST which is 01:30 AM in UST ( the default timezone in the server)

30 1 \* \* \* sh /root/backup.sh