Final Project Runbook

Required Software:

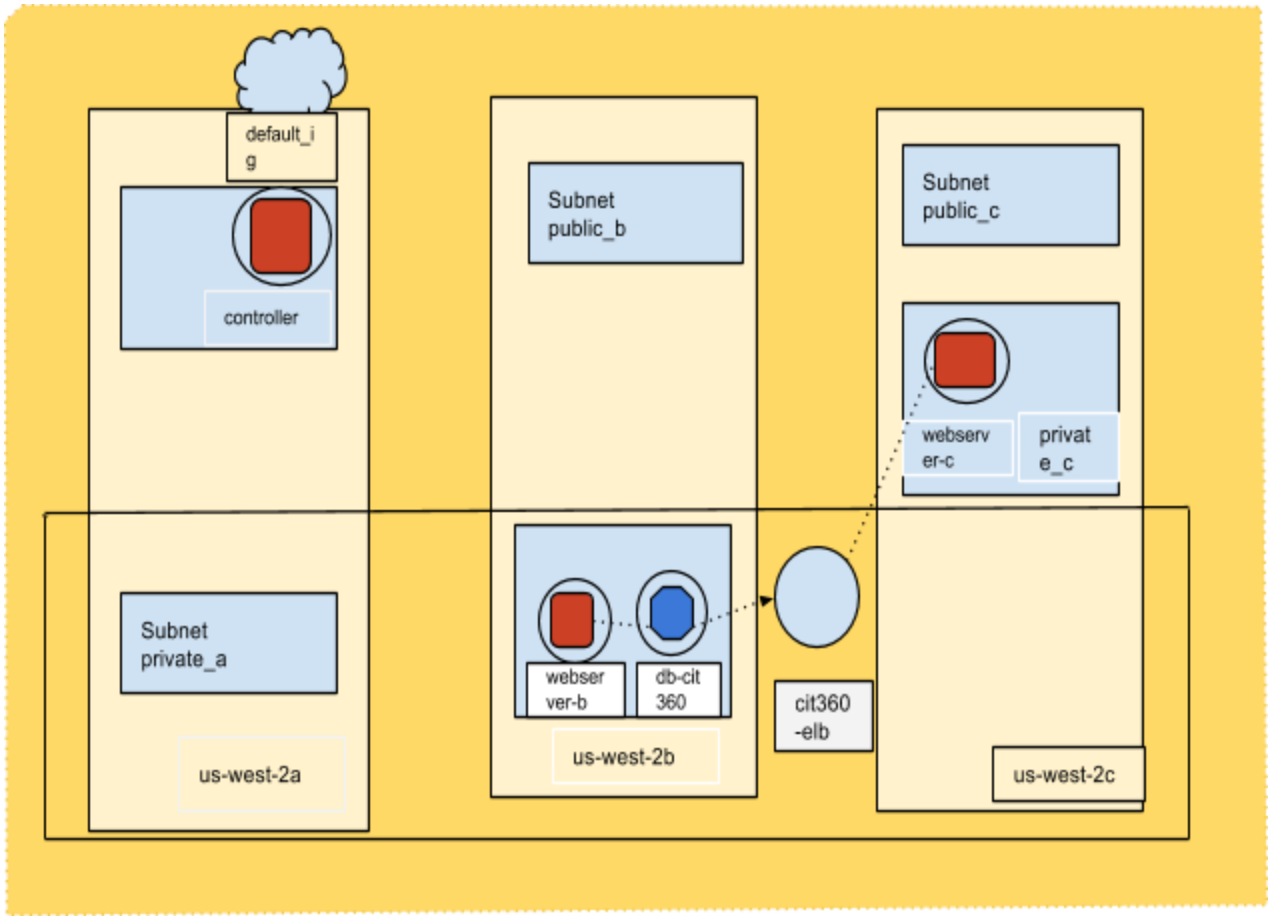
1) Amazon Web Services (AWS) account
- Accessible via provided user credentials
- Creating a new account

2) AWS Control Machine
- Ansible
- GIT
- MariaDB-client

3) Physical Machine
- AWS CLI
- GIT
- Terraform
- MobaXter,

4) AWS Web Servers
- nginx
- php
- php-ldap
- mbstring
- phy-mycrypt
- php-mysql
- composer

Architecture Diagram



Deployment

To deploy the software,one needs to install GIT, Terraform, AWS CLI, Deploy Terraform, Connect to the Bastion instance and deploy Ansible in order to connect to the web service.

Install GIT:
To install GIT go to https://git-scm.com/downloads, and select the installation which maches your computer version.

Terraform:
To install terraform, go to http://www.terraform.io/downloads.html and select the appropriate package for your system, after extracting the zipped files, and choosing where to instal terraform unto your system, now you can deploy your .tf file.

**make sure that the .tf file is in the same directory as the terraform instalation.

Amazon (AWS) Account:
To deploy the services, an Active AWS account is needed. Sign up for AWS and once installation is done, now generate a key pair. It can be found in the EC2 dashboard.

AWS CLI:
For starting to set up the AWS CLI, get the credentials from the AWS Website.
Go to the IAM users, choose users, choose the desired IAM user name, choose the security credentials tab and choose create access key. Then select show user security credential, and download the credentials.

Deploy Terraform:
First set up terraform. To run services on the wanted host, go to terminal and set up aws CLI first. This will prompt you for vpc id, password etc. Once that is set up now terraform will run on that provider.

Connect to the Bastion instance and Deploy Ansible to Configure the web service:

On windows download mobaxterm to ssh into bastion host. Then using the public dns of the EC2 instance ssh into Bastion host with the key created earlier. Once ssh connection is complete, install ansible using pip and then drag ansible files into the home directory on mobaxterm and run the files. If implemented correctly without errors using the elastic load balancer address one should be able to access the site.

Use the Web service:
Find the public DNS name of the ELB. this could be done either through the command line in our machine or from the AWS console from the EC2 dashboard.
Then copy and paste the address into a web browser, and we should see a working website.

Troubleshooting:
Error “cannot connect via ssh; public key denied” : If this error occurs there could be a configuration issue in the hosts.ini file. Go into the file and where ever the error is occuring type ‘ansible_ssh_private_key_file=~/key_name’

Error: “unprotected private key”: if this error occurs it means that the key file permissions are too open. Change file permissions to read only(400)

_