Seera Code Challenge

Task & Requirements

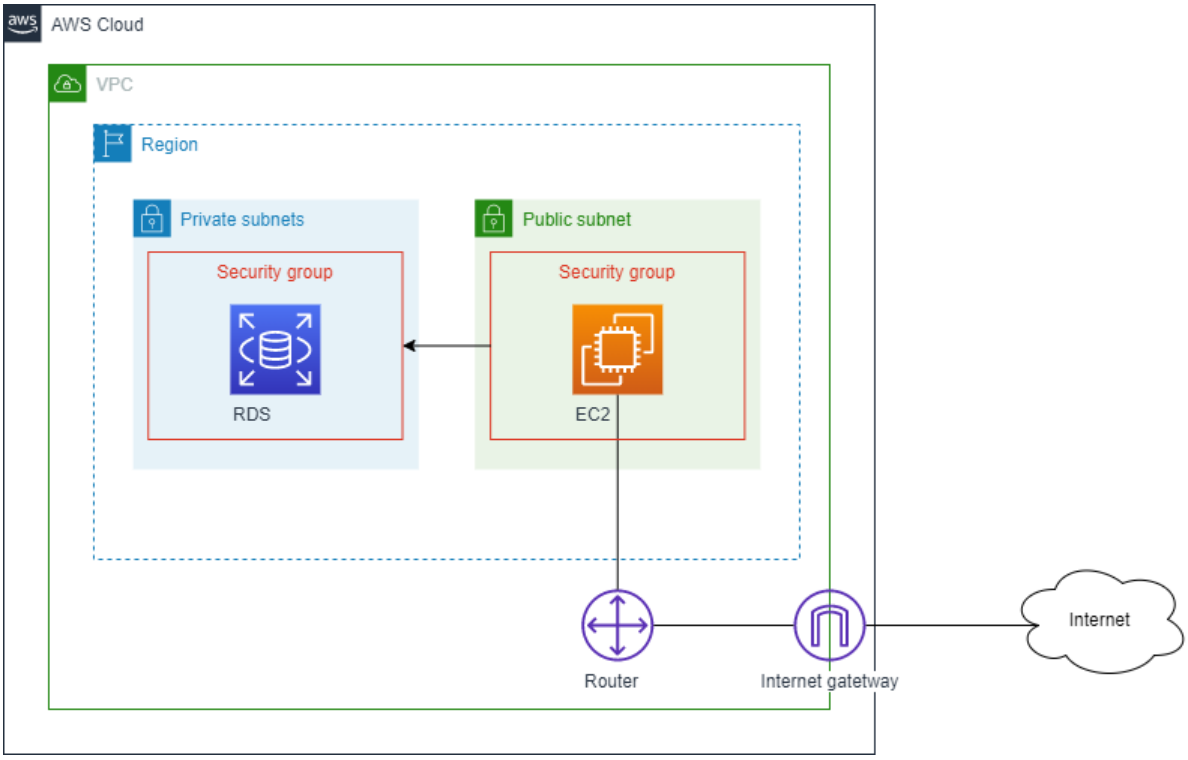
Bring up a WordPress stack using any of the infrastructures as code tools (Cloudformation / Terraform). This should create new VPC with subnets, route tables, etc. The database should use RDS.

The service should be fault tolerant (no need of HA). In case of a server failure, during termination of existing server, a new server should come up and configure everything automatically. The service should come back to existing state without any manual interventions.

For the WordPress app, you can use EC2 with standalone WordPress installation or Docker if you want to containerize it.

You can use any tools of your choice (CF, terraform, Ansible, Docker, Mesos, ECS, K8s, Shell scripts, boto, etc) to accomplish this task. Once the code is completed please push it to GitHub with instructions to bring up the stack. Please write those instructions as complete as possible, detailing any consideration you would like to explain regarding the solution, or any assumption we should take into consideration.

Network Architecture



Providers

Name	Version
terraform	>1.0.3

Usage

Variables

Even though there is no need you can change the variables in `dev.tfvars` to suite your needs. (the key names variables will affect the ssh key creation)

Steps

1. Create SSH Key

Go to the source directory:

```
cd source
```

Run:

```
ssh-keygen -t rsa -N "" -f id_seera
```

2. Create infrastructure and provision WordPress

Run the following commands one after another:

```
terraform init
```

```
terraform plan -var-file=dev.tfvars -out=tfplan
```

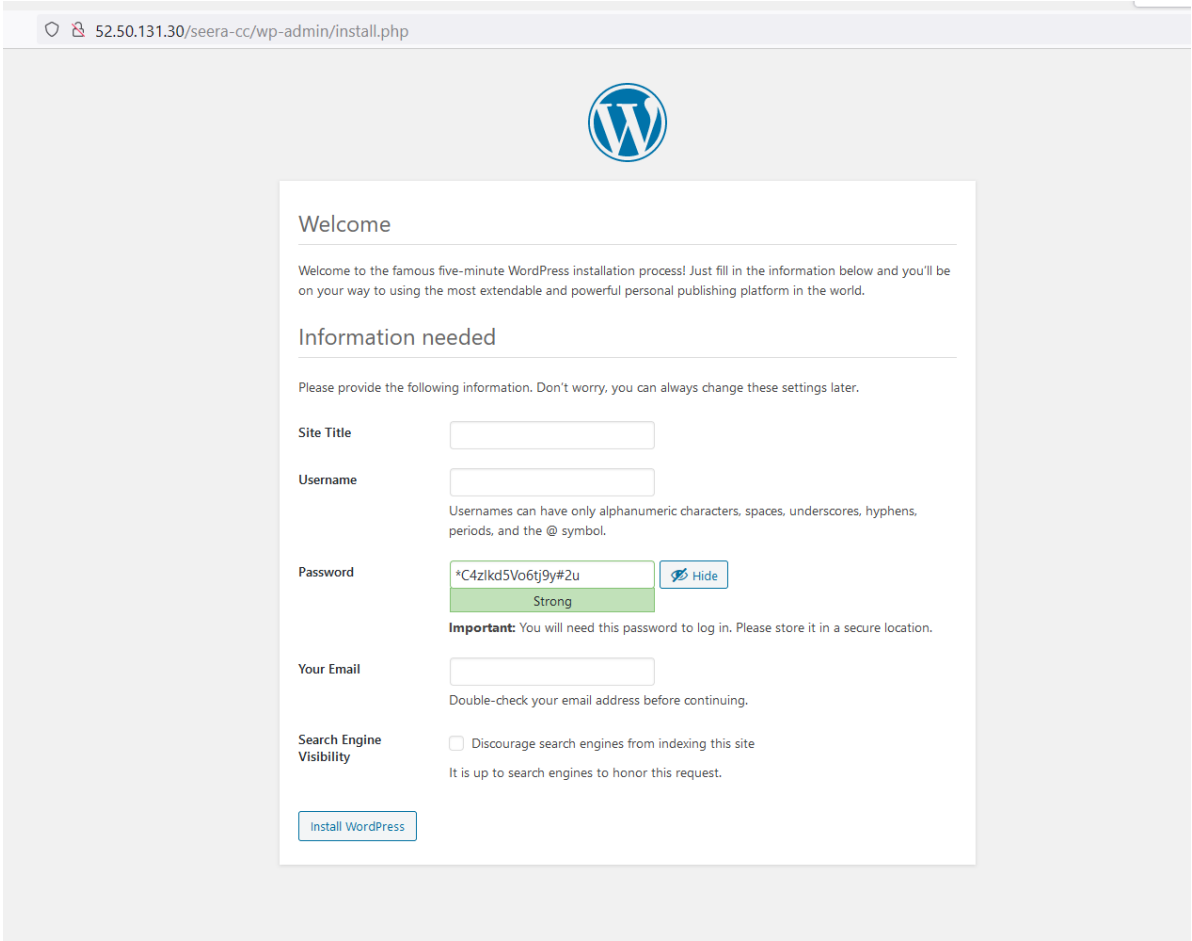
```
terraform apply -auto-approve -input=false tfplan
```

3. Look at the output from Terraform

Should look something like this:

4. Setup WordPress

Go to the link in the `web_access` output variable from the previous step and create your blog.



How to delete the resources

Run the following command to delete all resources created.

```
terraform destroy -auto-approve -var-file=dev.tfvars
```

SSH Agent Problem

In the remote connection to the ec2 instance I used `agent = false` at line 294 in `main.tf`. I did it because I run git-bash on Windows 10.

I'm unaware if this might become a problem when running the script on Linux or OSX and having an agent running. If there is a problem with the last step of `main.tf` then change the variable to `agent = true`.