Main.tf organization

Now that we’ve created our base files, let’s start investigating our infrastructure plan.

We’re going to open the main.tf file and start adding comments that will help organize our thoughts and our structure.

1. IAM: I prefer to start with IAM as I feel security should be at the top of any infrastructure plan. We only need to create one role here, luckily, and that is for S3. But I am just going to add a # hash and put IAM here.
2. Next up, we have our VPC/Networking. Networking is going to be one of the largest sections of our Terraform script. We have the
   1. #VPC
   2. #We need to create our internet gateway
   3. #We need to create our Public route table
   4. # Now our Private route table
   5. # Now for our subnets
   6. We want this to be custom and I prefer to explicitly assign subnets and availability zones for every resource that requires one. This allows me to quickly troubleshoot issues based on their IP or availability zone and know if something is out of place.
   7. #We are going to create a public subnet
   8. # Private subnet 1 for 1 group of ASG launched servers
   9. # A private subnet 2 for another group of ASG launched servers to keep them fault tolerant
   10. # We have 3 RDS subnets to keep the database fault tolerant and satisfy requirements for the DB subnet group
   11. # Now we have our security groups:
   12. #Private
   13. #Public
   14. #RDS
3. Ok great, that is our basic structure of our VPC, now we need to create our s3 bucket for our code

#S3 code bucket

If you add a cloudfront distribution (which I recommend), you will also want to create a media bucket here.

1. Now we need to create our Compute resources, there are several to create:
   1. First we need to create the Key Pair from the key we created using SSH-Keygen in a previous lesson. Again, Terraform is unable to instruct Amazon to create a key and must export one to Amazon from your machine.
   2. Next up is the Master dev server that will have the initial code on it.
      1. This will use an Ansible playbook to install the necessary software and wordpress to our instance
   3. Then we have the load balancer to forward traffic to the private instances. We create this first since the ASG needs to reference this further down in the script. Terraform is generally pretty smart about which resources to create first, but it’s best to try to be as explicit about the script order as possible. This also aids in readability.
   4. Now we need to create an AMI from our dev instance
   5. Now, the launch configuration which will use the AMI created from the previous step to deploy instances.
2. Finally, we have our Route53 records.
   1. We have our primary zone that will use the delegation set we created earlier.
   2. We will need a dub dub dub record that points to the Load balancer alias
   3. We will need a dev A record that points to the dev server Public IP address.
   4. And, finally, we have the db CNAME record that points to the RDS instance. Although this isn’t publicly accessible, this will allow our Wordpress installation to reference the RDS instance easily so you won’t have to change it if the DB instance IP changes.

Ok! That about does it! Next up, we can start building our IAM roles and the VPC!