Initial Setup

Hello! And welcome to the Linux Academy course on creating an AWS environment using Terraform and Ansible!

In this course, we are going to create an AWS Wordpress environment. This environment will consist of a custom VPC, public and private subnets, security groups, S3 buckets, an RDS database, a dev instance that creates an AMI that is used by a Launch configuration for an autoscaling group that is served by a load balancer, and finally, the route53 records that pull it all together! And when you’re all done, it will take less time to run the command that launches it all than it took me to recite this paragraph! So let’s get to it!

First up, I’m going to go over a few buzzwords that you’ve probably heard before, so I’ll make it quick.

The first one of these terms is:

**Idempotency**: means that no matter how many times we run this script, it won’t cause duplicates or other issues with resources that are working fine. So, for instance, if you have a server that terminates or you accidentally delete an S3 bucket, you can re-run this Terraform script and it will recreate those resources without affecting the resources that are in working order. Obviously, your application may require certain resources to be recreated, but the general concept remains sound. This is a huge benefit to using Terraform and Ansible for your infrastructure.

Another devops buzzword you will hear is:

**Immutability**: This essentially means that we won’t have to fix any infrastructure if it becomes non-working. Let’s say a dev server is compromised or an elastic load balancer starts having Amazon-related issues, we can just terminate those resources and recreate them to bring our infrastructure back to the same place it was before the issue with minimal to no downtime.

And, finally, one I’m sure you’ve all heard:

**Scalability:** This means that our environment can be tailored to scale to as many or as few servers as we need to maintain the proper performance for our clients and…expenditure…for our bosses. You will see how we can configure the number of resources required with ease!

Before you proceed with this course, I strongly urge you to check out the intro courses to Ansible and Terraform.

Although, this course isn’t incredibly advanced or difficult, it will help to have more of an understanding of what I’m doing and why I do it. You may also find that there are many things you can optimize!

I encourage you to take this example and work to make it yours! If I went through every way I could possibly improve this script, this course would never end!

The more you experiment with it, the better you’ll be!

Terraform apply yourself!

Anyway, enough of the boring video-intro jargon, let’s get to coding!

First up, if you haven’t already, install Python, to check to see if you have Python already installed, you can type python --version

1. Yum –y install python

After you’ve installed python, we need to download pip and install it. I’m going to gloss over these steps as they have been covered in depth in the other terraform, ansible, and aws courses that I hope you have taken before this one.

1. Curl –O <https://boostrap.pypa.io/get-pip.py>
2. Python get-pip.py

Now we are going to install the AWS CLI:

1. Pip install awscli
   1. El Capitan: pip install awscli –ignore-installed six
2. Next up, create an IAM user, give them administrator access (power user won’t work as we will be modifying IAM roles) and download the key.

Now to add our keys to our global variables, you can do this either by using

1. Aws configure

Or by editing the credentials file directly, which will allow you to set a custom profile to manage multiple accounts.

1. Vim ~/.aws/credentials to change profile name if you want
   1. Don’t worry, this key will be long gone by the time you see this video!
2. Ssh-keygen
   1. /root/.ssh/my\_key
3. Ssh-agent bash
4. Ssh-add ~/.ssh/my\_key
5. aws route53 list-hosted-zones --query HostedZones[0].Id)
6. wget <https://releases.hashicorp.com/terraform/0.7.10/terraform_0.7.10_linux_amd64.zip>
7. Mkdir ~/terraform && unzip terraform –d ~/terraform
8. export PATH=$PATH:~/terraform
9. aws route53 create-reusable-delegation-set --caller-reference 1234
10. yum –y install ansible