# Curriculum Web Service Runbook

**Short Description**

Run a web server that will host the Curriculum Web Service website. Create instances for the Bastion Host and the MariaDB database. Configure security groups to only allow whitelisted access. Configure load balancer to use both web servers.

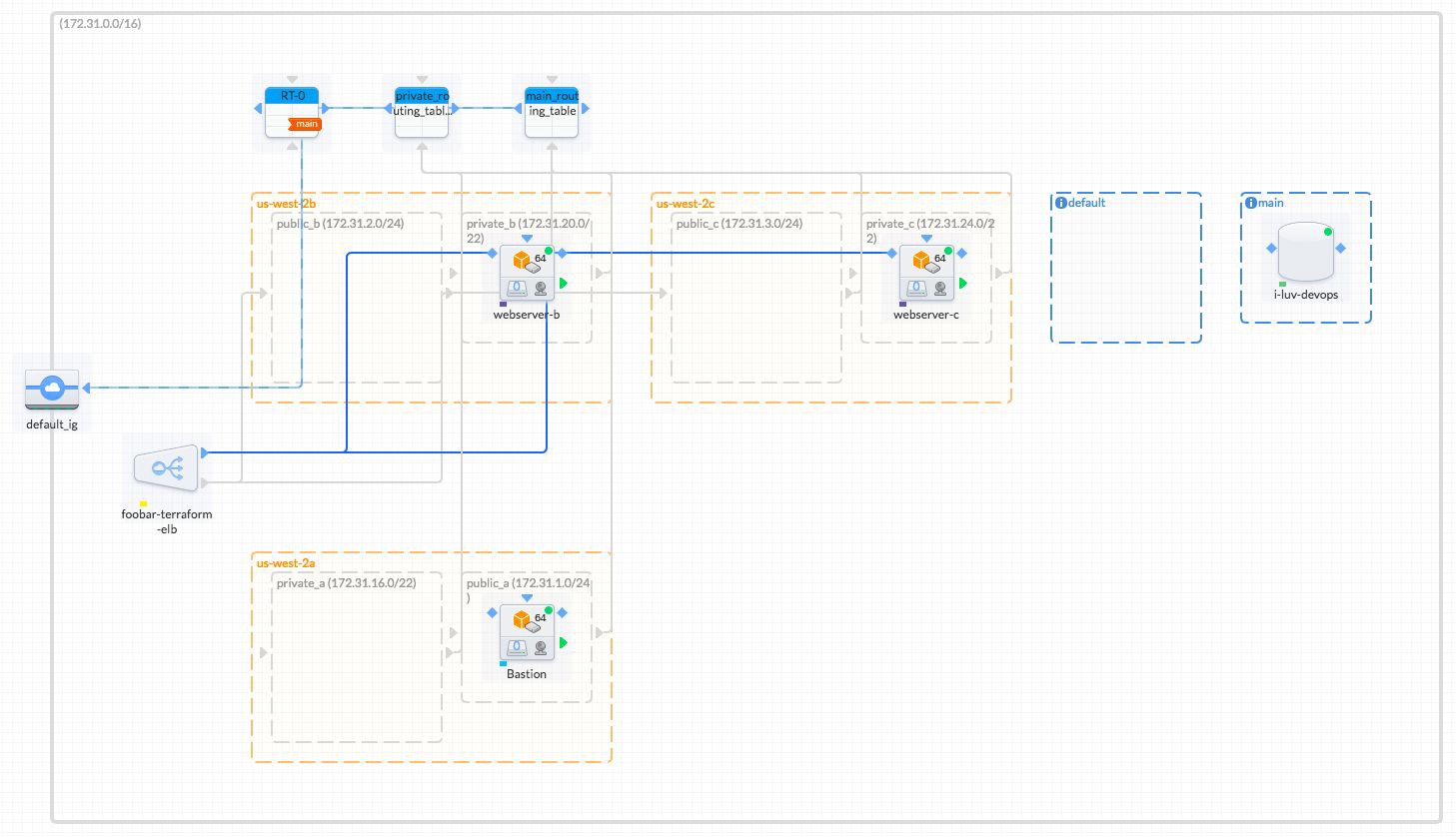
**Required Software**

Local Machine: Git, Ansible, Terraform, SSH Access, AWS CLI

Bastion Host: Ansible, MariaDB

Web servers: Nginx, PHP, Composer

**Architecture Diagram**



**Deployment**

You may download the required files from <https://github.com/bhiranso/cit-360>.

**Terraform Configuration**

In order to run these files you must install terraform. Download via <https://www.terraform.io/downloads.html> and make sure to select your appropriate operating system and version. You will be given a zipped file, extract that file to where you want terraform installed. Add terraform to your PATH by typing into terminal, “$ PATH=/usr/local/terraform/bin:/home/account-name/terraform:$PATH”. Test to see if your terraform is installed by typing into terminal, “terraform –version”.

**Amazon Web Service Command Line Interface**

Amazon offers a command line interface to control Amazon Web Services in which we will use with terraform to administer your AWS account. Refer to <http://docs.aws.amazon.com/cli/latest/userguide/installing.html> to install AWSCLI.

Once you have AWSCLI installed, type “aws configure” in terminal and enter your Access Key ID and Secret Access key which you can find using the AWS Web Console under IAM.

**Running Terraform**

To run the terraform command, you must type into terminal “terraform apply”. Make sure you are in the terraform directory where you have extracted the files.

**Configure Bastion**

In order to configure the Bastion Instance, SSH into your Bastion instance using the command “ssh –I \*PEMKEY\* ec2-user@\*YOURAWSDNS\*. Type command “sudo yum update”. To install Ansible, type command “sudo pip install ansible”. Copy the private key pair that was given by AWS to the Bastion instance (you may secure copy or FTP).

**Curriculum Website**

To access the new Curriculum website, go to the AWS console and find your public DNS of the Load Balancer and copy paste that into a web browser.

**Issues**

Title: Permission Denied SSH Into Instance

Description: Unable to SSH into instance with the permission denied error.

Remediation Steps:

Your private key file must be protected from read and write operations from any other users. If your private key can be read or written to by anyone but you, then SSH ignores your key and you see the following warning message below.

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@@@@@@@@@@@@@@@@@

@ WARNING: UNPROTECTED PRIVATE KEY FILE! @

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@@@@@@@@@@@@@@@@@

Permissions 0777 for '.ssh/my\_private\_key.pem' are too open.

It is required that your private key files are NOT accessible by others. T his private key will be ignored.

bad permissions: ignore key: .ssh/my\_private\_key.pem P ermission denied (publickey).

If you see a similar message when you try to log in to your instance, examine the first line of the error message to verify that you are using the correct public key for your instance. The above example uses the private key .ssh/my\_private\_key.pem with file permissions of 0777, which allow anyone to read or write to this file. This permission level is very insecure, and so SSH ignores this key. To fix the error, execute the following command, substituting the path for your private key file.

$chmod 0400 .ssh/my\_private\_key.pem