Objective

This workshop has 2 parts

1. Create an installable Ubuntu image with Code Server and Nginx
2. Automate the deployment of the Code Server image

Setup

a. Create a directory called `workshop03` in your course repository.

Workshop

Part 1 - Create the Code Server Golden Image

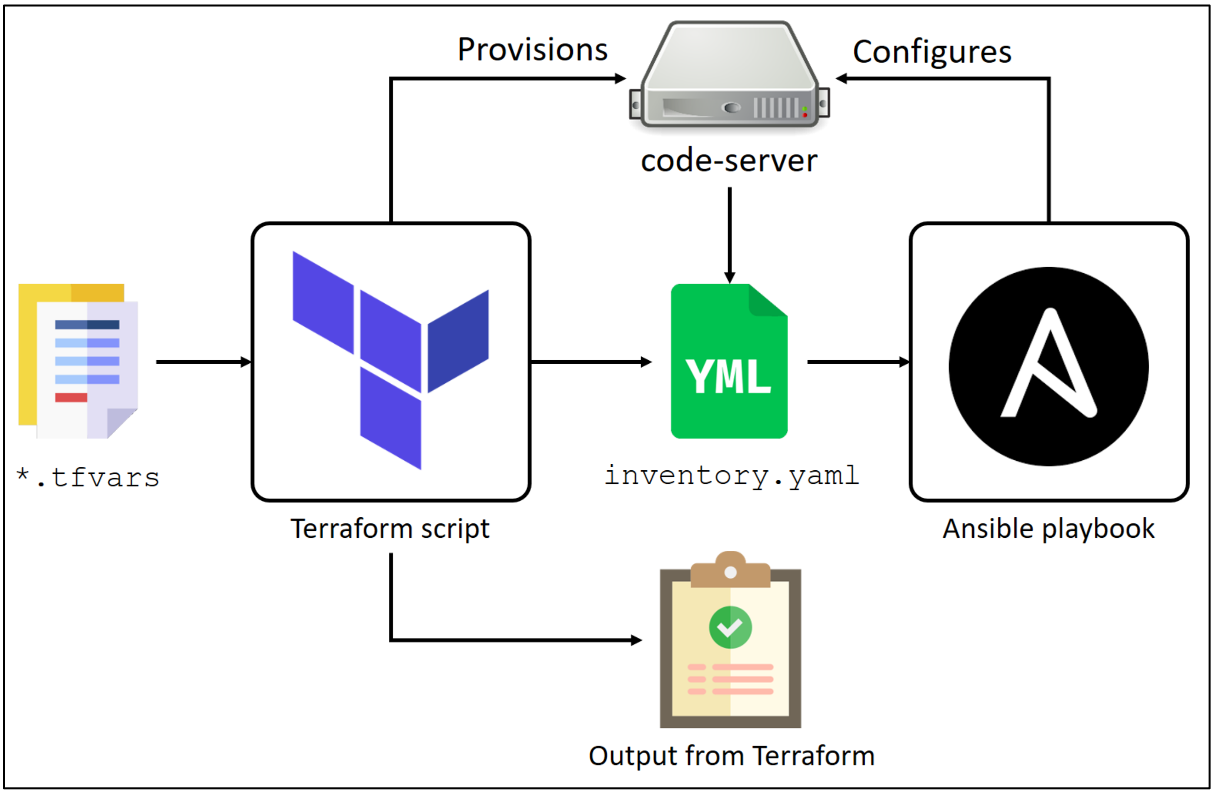
Build a golden image with the Ansible playbook that you have created in Workshop 2.

Create a directory called `build` in `workshop03` directory. Copy the playbook to `build` directory. Write a Packer script to use the playbook to configure the image.

Use Ubuntu 20.04 x64 as your base image.

Part 2 - Provisioning a Code Server Instance

The following diagram shows the stages involved in provisioning a code-server from the image created in **Part 1** above.



Your Terraform script will use the code-server image to deploy a virtual machine. When the script completes, it will produce 2 artefacts

- Domain name of the provisioned server
- An Ansible inventory file to be used by a playbook to configure and complete the configuration

Create a directory called `deploy` in `workshop03` directory. Write all your scripts for Part 2 of this workshop in the `deploy` directory.

Part 2 Stage 1 - Provisioning the Server

Write a Terraform script to provision the code-server image. The Terraform script should allow you to configure the following

- Code server version
- Code server password
- SSH key pair to be used for login in to the server. You can generate a new key pair or reuse the Docker host keys
- You should make any secret tokens, region, etc. configurable

Outputs

After the Terraform scripts completes, it should produce the following artefacts

- Output variable of the form `code-<ipv4_address>.nip.io`. This will be the domain name of our server. Note: nip.io domain allows you to map any IP address to a hostname to facilitate testing. See https://nip.io/
- An inventory file called `inventory.yaml`, to be used by Ansible. You should include SSH connection information in this inventory file

  The inventory file should also include the code server password and the IPv4 address of the server. The latter is used to configure the virtual host in Nginx

Part 2 Stage 2 - Configuring the Server

Write a playbook that will use the inventory.yaml file to configure the server. The playbook should perform the following tasks

- Update the `/lib/systemd/system/code-server.service` file with the code server password; change the following line

  ```
  Environment=PASSWORD=__PLACEHOLDER__
  ```

  with the password, assuming that the password is `mypassword`

  ```
  Environment=PASSWORD=”mypassword”
  ```

- Update the `/etc/nginx/sites-available/code-server.conf` file with the domain `code-<ipv4_address>.nip.io;` change the line with `server_name` to

  ```
  server_name code-<ipv4_address>.nip.io;
  ```

- Use systemd module to restart `nginx` and `code-server` services. You must also perform a daemon reload viz. set `daemon_reload` to yes.

Test

Test your deployment by browsing to `http://code-<ipv4_address>.nip.io`.

Submission

When you have completed this workshop, commit your work to the repository.

The instructor will clone your repository at the end