# Setup Terraform Cloud Lab 1.13

## Expected Outcome

In this challenge, you will create your Terraform Cloud trial and your first workspace to build infrastructure in Azure.

This challenge will require that you have a github account so that you can fork repositories. Be sure to login before beginning.

## How to

### Create Service Principal

Create a Service Principal on your Azure Subscription that Terraform will use to authenticate. To do this we need to get the following:

- Tenant ID
- Subscription ID
- Client ID
- Client Secret

The tenant and subscription info are static, but we need to generate that service principal to get the Client ID and Secret. To make things easy here is a one-line command to get the job done:

```
az ad sp create-for-rbac -n RollsRoyce --role="Contributor" --scopes
/subscriptions/$(az account show -o tsv --query id)
```

Note: As mentioned above, this command might not work in the cmd shell in Windows. If you can't use PowerShell or the Git bash, you should be able to separate this into multiple commands to get around cmd shell limitations, first getting your account ID and using that in the second query, like so:

```
az account show -o tsv --query id
az ad sp create-for-rbac -n RollsRoyce--role="Contributor" --scopes
/subscriptions/<ID from above query>
```

You may see output stating "Retrying", this is normal and is just the CLI waiting for the role to be created.

When everything is complete you should see something like this:

```
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
Retrying role assignment creation: 3/36
Retrying role assignment creation: 4/36
{
  "appId": "THIS IS YOUR CLIENT ID",
"displayName": "RollsRoyce",
"name": "http://RollsRoyce",
  "password": "THIS IS YOUR CLIENT PASSWORD",
  "tenant": "THIS IS YOUR TENANT ID"
}
```

The subscription id can be seen in the Azure Portal, or by running the Azure CLI command

```
 az account show
```

Take note of all 4 of these values and keep them safe, you will need to access them throughout the workshop.

NOTE: It is a good idea to remove this Service Principal after the workshop! Using the ID of the service principal, you can run an `az ad sp delete --id <ID>`. See here for more details.

## Set Azure Credentials

Terraform Enterprise uses a Service Principal to authenticate Terraform for use with Azure.

Keep track of the following environment variables based on the Service Principal:

```
export ARM_TENANT_ID=
export ARM_SUBSCRIPTION_ID=
export ARM_CLIENT_ID=
export ARM_CLIENT_SECRET=
```

## Fork the Repository

Open up a browser and navigate to the pre-built repository – provided by instructor or this one :

https://github.com/pmwtraining/azureTFE

In the top right, click the `Fork` button.

Follow the prompts which will fork the repository into your own space you control.

Note: We need to fork this repository to allow you to connect it to Terraform Enterprise which will create webhooks to get vital information about how and when the repository changes.
You should land in your forked version of the repository.

## azureworkshop-workspaces repository

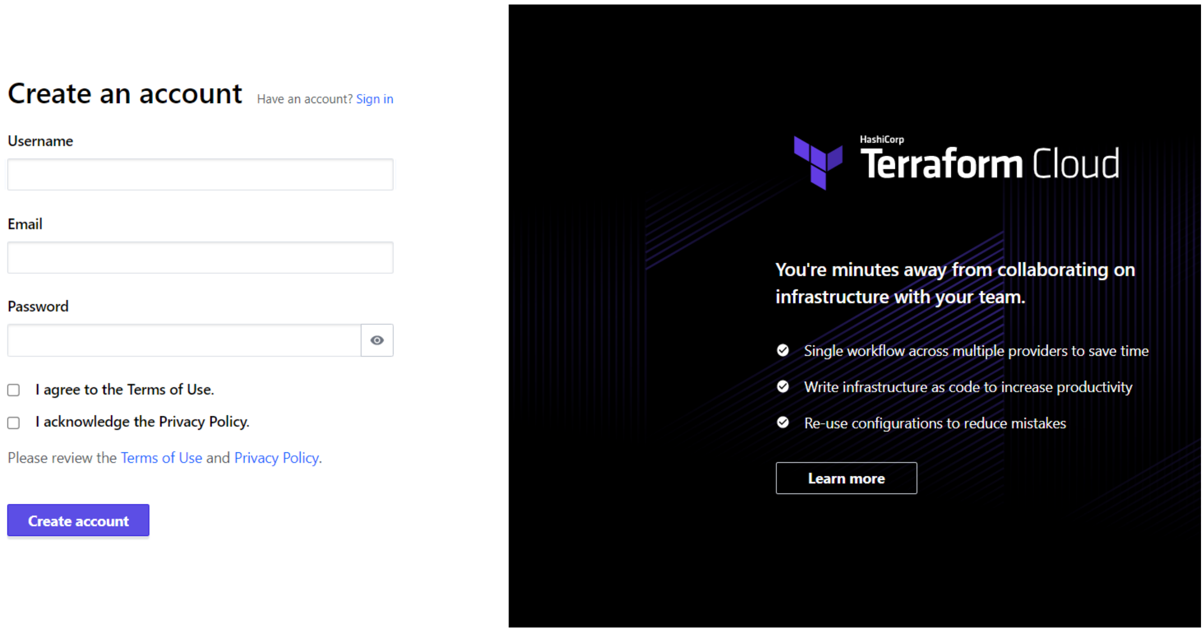
There is not much to this repository, just a couple folders that we will use in the next few challenges.

**app-dev** - Simple set of infrastructure to deploy from Terraform Enterprise.

**app-dev-modules** - More complex set infrastructure utilizing the Private Module Registry. Ignore this for now, we will use this in the next Challenge.

## Create a Trial Account for Terraform Cloud

Register for a [Terraform Cloud Trial](#).



Create an account Have an account? Sign in

Username

Email

Password

□ I agree to the Terms of Use.

□ I acknowledge the Privacy Policy.

Please review the Terms of Use and Privacy Policy.

HashiCorp Terraform Cloud

You're minutes away from collaborating on infrastructure with your team.

- Single workflow across multiple providers to save time
- Write infrastructure as code to increase productivity
- Re-use configurations to reduce mistakes

If you are working on this today with others from your organization, you can create a single trial and work together through the last few challenges.

## Create a New Organization

## Create a New Workspace

Click the "New Workspace" button.

Pick a name that indicates the intent of the infrastructure. Example: 'app-dev'

## Setup VCS

You won't have any "Source" options, so click the "+" button to connect Terraform Enterprise to your source control. You will see the following screen asking you to add a VCS root. Click the "Add VCS Provider" button to continue:

You will be brought to the Add VCS Provider page:

Follow the instructions for any of the following VCS providers (we are going to be using Github):

- Github
- Github Enterprise
- GitLab
- GitLab EE and CE
- Bitbucket Cloud
- Bitbucket Server

Note: This only has to be done once for each Version Control Provider.

Note: You will need to update your placeholder URL to successfully connect/create your GitHub VCS root. Make sure you grab your authorization callback URL from GitHub as defined in Step 3 of the instructions for GitHub above. You may not see the menus described in Step 2.

After this is done, you may have to go back and create your workspace if you didn't do so before you created the VCS provider. You can do so by navigating to the Workspaces tab at the top of the page and clicking the "New Workspace" button, choosing your GitHub VCS provider during creation.

## Connect Workspace

Connect your workspace to your VCS.

Set working directory and branch properly!

## Configure Variables

Now that you have a workspace, navigate to the variables.

Set a the Terraform Variable "name" to something unique. Example "app-dev". Click "Save".

Set Environment Variables for your Azure Service Principal (be sure check the 'sensitive' checkbox to hide these values):

- ARM_TENANT_ID
- ARM_SUBSCRIPTION_ID
- ARM_CLIENT_ID
- ARM_CLIENT_SECRET

Note: You used commands to get/set this information in Lab 1 Refer back to it if you need to refresh your memory.
Note: Remember also that some aliases/commands don't work as expected from a Windows cmd shell; if you have any issues with the commands you can try to run them either in the Git bash or PowerShell.
Click "Save".

## Run a Plan

Click the "Queue Plan" button.

## View the Plan

## Run an Apply

Enter a comment and then apply by clicking "Confirm & Apply".

### View the Apply

## Advanced areas to explore

1. Explore state versions after the apply.
2. Add another folder in the repository for 'app-prod' and create another workspace with different settings.
3. Push a change to the repository with the workspaces in it, what happens in Terraform Enterprise?

## Resources

- TFC Access