# Terraform 301

Given your familiarity of Terraform and AWS by now, do expect additional changes beyond the provided steps.

## Pre-requisites

1. Completed [Coaching: Terraform 201 | CE4](https://docs.google.com/document/d/1_eBXma14nQqQhss7CX8NWp3FS8hY2Z-bvSDQLu6cfLA/edit?usp=sharing) for understanding
2. Clone a copy of the completed source code from [here](https://github.com/seanlim1/tf-201), our activity continues from here

## Activities

### Activity G: Warm-up

#### Steps

1. Update the SecurityGroup resource block to have egress to all
2. Improve on the name and descriptions
   1. EC2 instance
   2. Security group
3. Plan and apply
4. Validate

#### Hints

* [Terraform AWS Provider: Security Group (resource)](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/security_group)

### Activity H: Refactor Module

#### Steps

1. Copy variable.tf file out of the module into the project root directory
2. Update the module variables to be mandatory arguments
3. Update the module block to use arguments from your variables
4. Plan and apply
5. Validate

#### Prompts

* Are the module outputs still showing?

#### Hints

* [Terraform Docs: References to Values](https://developer.hashicorp.com/terraform/language/expressions/references)

### 

### Activity I: Count

#### Steps

1. Use count to have multiple instances of the ec2 resource
2. Update the tag to include the index to its name
3. Plan and apply
4. Validate

#### Prompts

* Can we make the count an argument to the module?

#### Hints

* [Terraform Docs: Count](https://developer.hashicorp.com/terraform/language/meta-arguments/count)
* [Terraform Docs: References to Values](https://developer.hashicorp.com/terraform/language/expressions/references)

### Activity J: Conditional Expressions

#### Steps (Project Root)

1. Create a new ec2 instance resource for a webapp
2. Create a new security group resource to allow inbound on port 80
3. Use a variable block to enable/disable the creation of this webapp
4. Plan and apply
5. Validate

#### Hints

* [Terraform Docs: Conditional Expressions](https://developer.hashicorp.com/terraform/language/expressions/conditionals)

### 

### Activity K: EC2 with User Data

#### Steps

1. Create a file `init.sh` in your project root

| #!/bin/bash  yum install -y httpd  systemctl enable httpd  systemctl start httpd |
| --- |

1. Add this into the webapp instance resource block

| user\_data = file("${path.module}/init.sh") |
| --- |

1. Use a output block to show the public ips of the webapp instance(s)
2. Plan and apply
3. Validate

#### Prompt

* After providing the user data, did it get applied to the ec2 instance?
* Does the existing security group permit connection to the package manager repositories?

#### Hints

* [Terraform AWS Provider: EC2 Instance (resource)](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance)

### 

### Activity J: S3 Backend

#### Pre-requisites

* S3 bucket to store the state

#### Steps

1. Use a backend block to store terraform state in S3
2. Plan and apply
3. Validate

#### Prompt

* Is the state file in your local project directory still being used?
* What is the name of the state file in the remote backend?
* Should you lose your local project directory (e.g. hard disk failure), can your teammates still work on your terraform project?
* What happens if two individuals run the `terraform apply` at the same time? Can the race condition be avoided?

#### Hints

* [Terraform Docs: S3 Backend](https://developer.hashicorp.com/terraform/language/settings/backends/s3)