Terraform III

If state file is present, can we create the resource again? No

Terraform project setup with modules:

1. Create a project directory
   1. mkdir 05-terraform-modules-project
2. Create a provider.tf file

provider "aws" {

region = "ca-central-1"

}

1. Create a ‘modules’ directory inside main project directory
2. Inside ‘modules’ directory we need to create ec2, s3 directories
   1. mkdir 05-terraform-modules-project/modules/ec2
   2. mkdir 05-terraform-modules-project/modules/s3
3. Need to add access key, secret key, to the project

export AWS\_ACCESS\_KEY\_ID = “”

export AWS\_SECRET\_ACCESS\_KEY=””

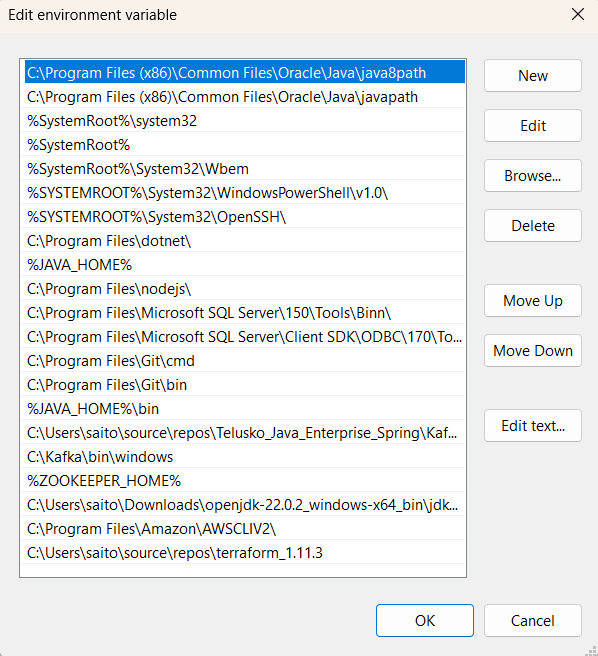
1. Create files in ec2 directory with names: input.tf, main.tf, output.tf
   1. touch input.tf
   2. main.tf
   3. output.tf

1. Create files in s3 directory with names: input.tf, main.tf, output.tf
   1. touch input.tf
   2. main.tf
   3. output.tf
2. Create main.tf, output.tf under main directory
3. Edit ec2 dir ---> main.tf, input.tf, output.tf
4. Edit s3 dir ---> main.tf

Working with Terraform in Windows machine, Lock file, Statefile, Taint and Untaint in Terraform

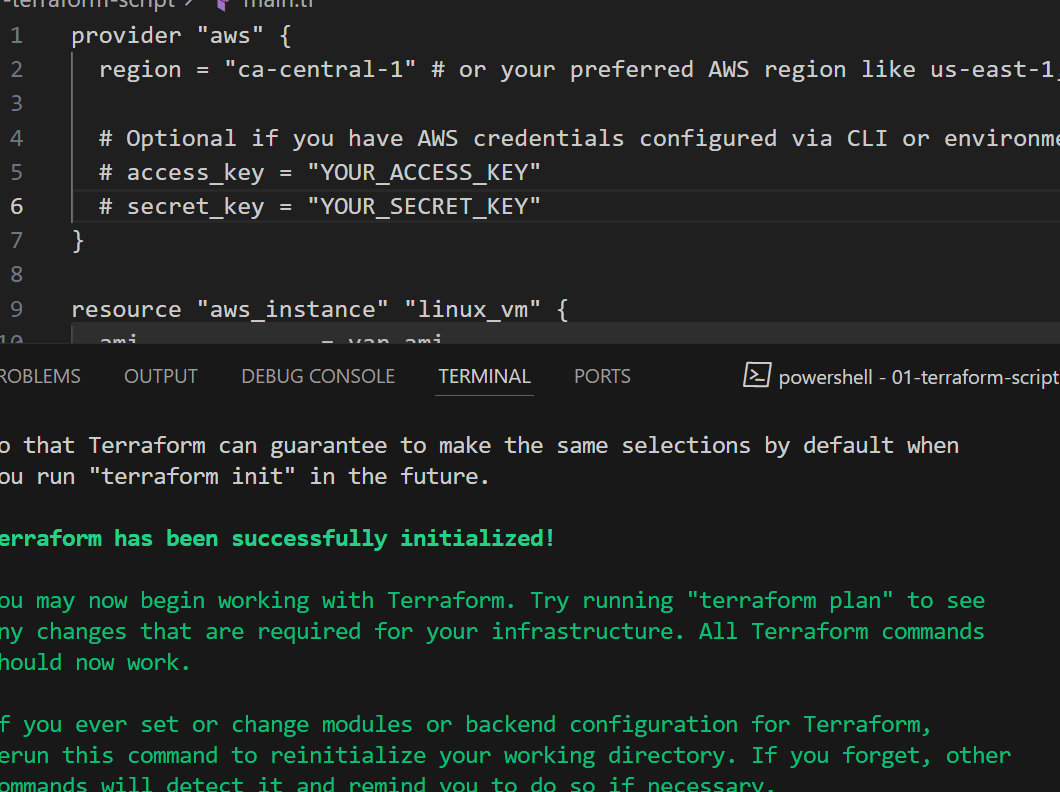
Working with Terraform in Windows machine

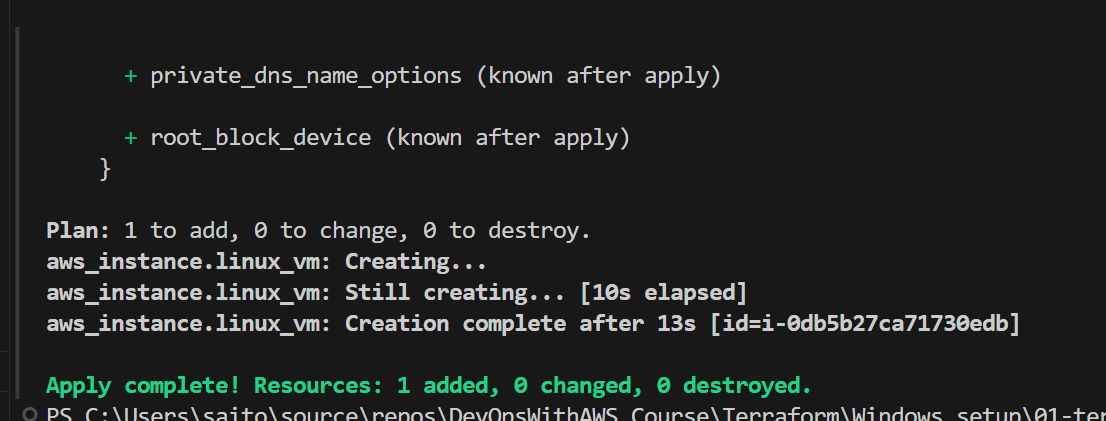
1. Download Terraform for Windows and extract the ZIP file
2. After extracting we can see terraform.exe file
3. Set path for terraform in System variables --> Environmental variables

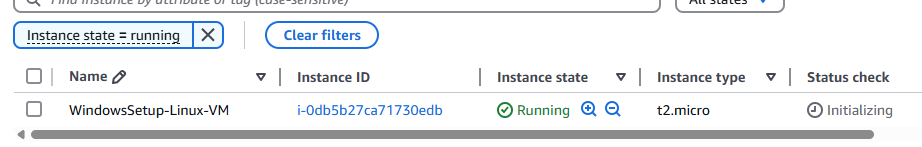


1. Configure AWS credentials in system environmental variables for local setup
2. Download and install VSCode

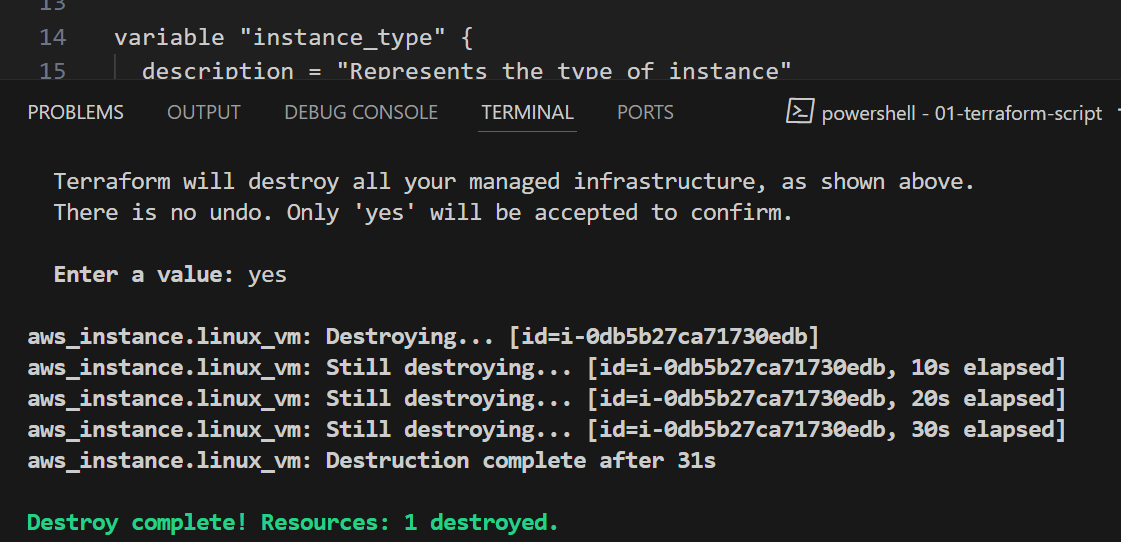
In VScode I add the code in main.tf and write terraform init



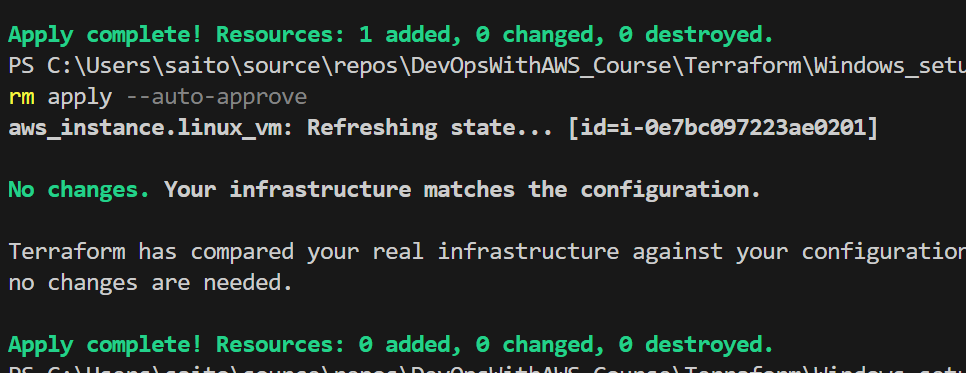
It has created EC2 from VS code: 



Destroyed from VSCode

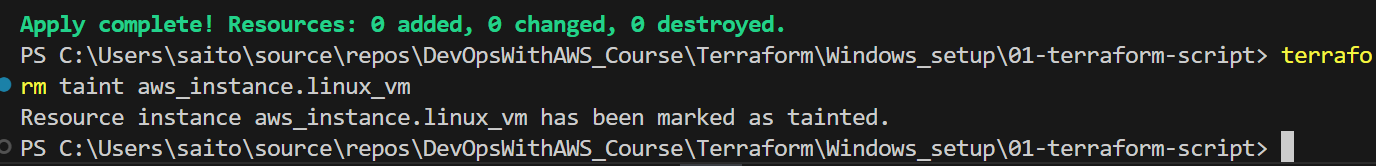


Without applying taint



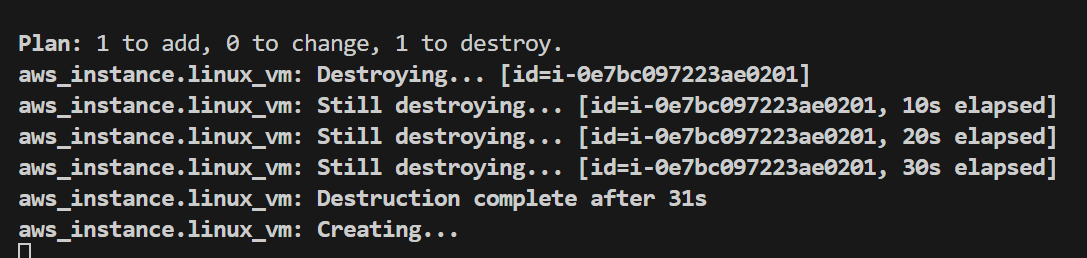
If any of the resources is tainted or added as a tainted resource, it will destroy the existing resource and add the new resource, instead of just refreshing state

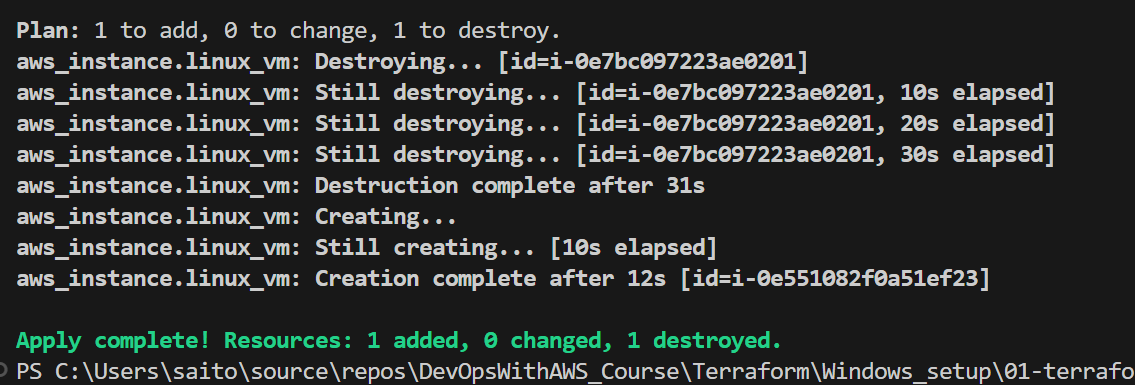
terraform taint aws\_instance.linux\_vm



After taint when we apply again

It will destroy then create the resource





The default behavior is once State file is created, we cannot re-apply

terraform taint and terraform untaint are commands used to **force Terraform to recreate or preserve specific resources** during the next apply. They’re useful when you want to manually intervene in Terraform’s lifecycle management of infrastructure.

Once we taint the resource, it is possible to re-create the new resource

You can “taint” a resource using terraform taint to mark it as needing to be re-created during the next terraform apply. It’s like telling Terraform that a resource is “bad” or needs re-deployment

terraform taint aws\_instance.linux\_vm

Using terraform untaint, you can remove the taint from a resource , so it will not be re-created and will stay as is during the next apply

To get back default behavior, we run untaint

terraform untaint aws\_instance.linux\_vm

Before untaint, taint once again then untaint

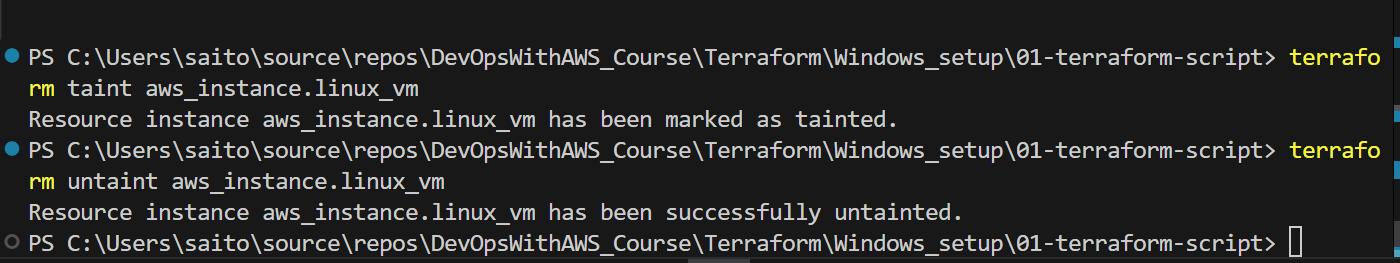
PS C:\Users\saito\source\repos\DevOpsWithAWS\_Course\Terraform\Windows\_setup\01-terraform-script> terraform taint aws\_instance.linux\_vm

Resource instance aws\_instance.linux\_vm has been marked as tainted.

PS C:\Users\saito\source\repos\DevOpsWithAWS\_Course\Terraform\Windows\_setup\01-terraform-script> terraform untaint aws\_instance.linux\_vm

Resource instance aws\_instance.linux\_vm has been successfully untainted.

PS C:\Users\saito\source\repos\DevOpsWithAWS\_Course\Terraform\Windows\_setup\01-terraform-script>



What’s a State file in Terraform?

The state file (terraform.tfstate) is where Terrform stores the current state of your infrastructure. It keeps track of all the resources that Terraform manages, so it knows what to create, update, or delete. This file is essential for Terraform to understand what is already deployed.

What’s Lock file in Terraform?

Lock file is a set of files. For example, here version=”~>5.0” is locked

If different team members within the same team are using different versions that might create a problem. We lock the version and it will maintain the consistency. All pipelines will have the same versions. Lock the versions of providers to maintain consistency across all environments

Lock file in Terraform:

A lock file (.terraform.lock.hcl) is used to lock the versions of provider plugins that Terraform uses. This ensures that your team or CI/CD pipelines are using the same versions of providers across all environments. It’s automatically generated by Terraform to avoid unexpected changes due to provider updates.

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "~> 5.0"

}

}

}

# Configure the AWS Provider

provider "aws" {

region = "us-east-1"

}

# Create a VPC

resource "aws\_vpc" "example" {

cidr\_block = "10.0.0.0/16"

}

We don’t have to create the lockfile automatically in Terraform. The moment we initialize terraform, lockfile will automatically be created.