

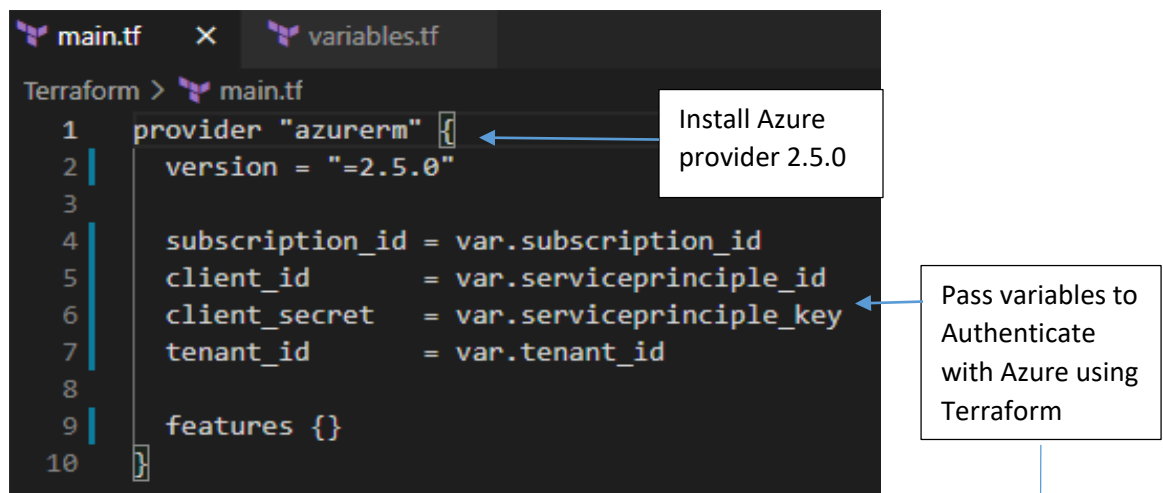
## Terraform Init and Plan

In this lesson we will learn how to create main.tf and variables.tf and how to initialize Azure provider for Terraform and how to use the plan command.

**mkdir Terraform**

**cd .\Terraform\**

### Create main.tf and initialize Azure provider

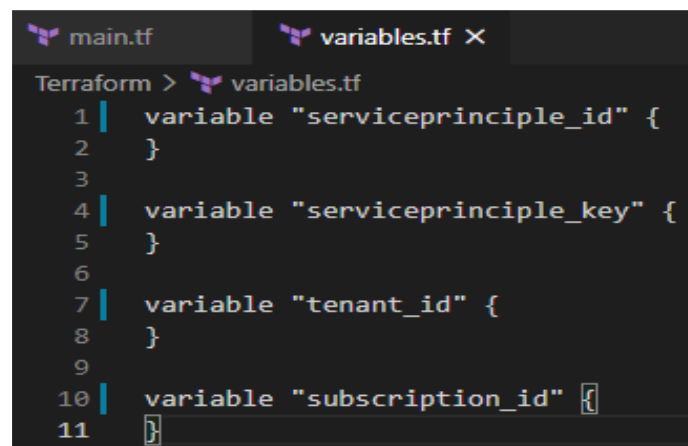


```
main.tf  X  variables.tf
Terraform > main.tf
1  provider "azurerm" {
2      version = "=2.5.0"
3
4      subscription_id = var.subscription_id
5      client_id       = var.serviceprinciple_id
6      client_secret   = var.serviceprinciple_key
7      tenant_id      = var.tenant_id
8
9      features {}
10 }
```

Install Azure provider 2.5.0

Pass variables to Authenticate with Azure using Terraform

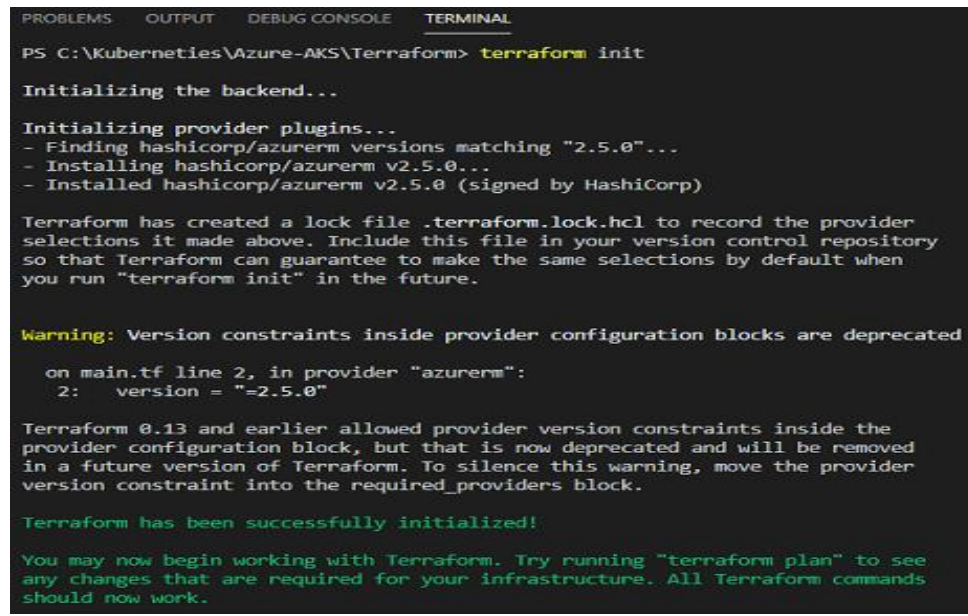
### Create variables.tf in order to pass information to main.tf



```
main.tf  variables.tf X
Terraform > variables.tf
1  variable "serviceprinciple_id" {
2  }
3
4  variable "serviceprinciple_key" {
5  }
6
7  variable "tenant_id" {
8  }
9
10 variable "subscription_id" {
11 }
```

The command below initializes terraform and installs Azure provider plug-in:

### terraform init



```
PS C:\Kubernetes\Azure-AKS\Terraform> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/azurerm versions matching "2.5.0"...
- Installing hashicorp/azurerm v2.5.0...
- Installed hashicorp/azurerm v2.5.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Warning: Version constraints inside provider configuration blocks are deprecated

  on main.tf line 2, in provider "azurerm":
   2:   version = "2.5.0"

Terraform 0.13 and earlier allowed provider version constraints inside the
provider configuration block, but that is now deprecated and will be removed
in a future version of Terraform. To silence this warning, move the provider
version constraint into the required_providers block.

Terraform has been successfully initialized!

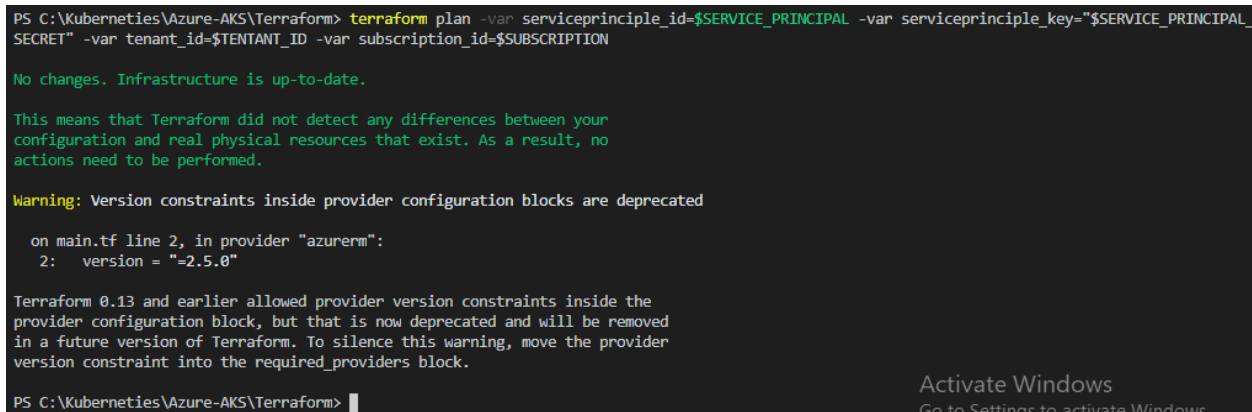
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
```

### Plan command

Plan command runs the terraform in dry-run in order to show us what terraform is about to do. We are passing to the plan command the Service Principal values that were kept in variables in the previous lesson for serviceprincipal\_id, serviceprincipal\_key, tenant\_id and subscription\_id which were defined in variables.tf for providing Terraform the Contributor role and keeping them out of the repository.

### terraform plan:

```
terraform plan -var serviceprincipal_id=$SERVICE_PRINCIPAL -var
serviceprincipal_key="$SERVICE_PRINCIPAL_SECRET" -var tenant_id=$TENTANT_ID -var
subscription_id=$SUBSCRIPTION
```



```
PS C:\Kubernetes\Azure-AKS\Terraform> terraform plan -var serviceprincipal_id=$SERVICE_PRINCIPAL -var serviceprincipal_key="$SERVICE_PRINCIPAL_SECRET" -var tenant_id=$TENTANT_ID -var subscription_id=$SUBSCRIPTION

No changes. Infrastructure is up-to-date.

This means that Terraform did not detect any differences between your
configuration and real physical resources that exist. As a result, no
actions need to be performed.

Warning: Version constraints inside provider configuration blocks are deprecated

  on main.tf line 2, in provider "azurerm":
   2:   version = "2.5.0"

Terraform 0.13 and earlier allowed provider version constraints inside the
provider configuration block, but that is now deprecated and will be removed
in a future version of Terraform. To silence this warning, move the provider
version constraint into the required_providers block.

PS C:\Kubernetes\Azure-AKS\Terraform>
```

Done!