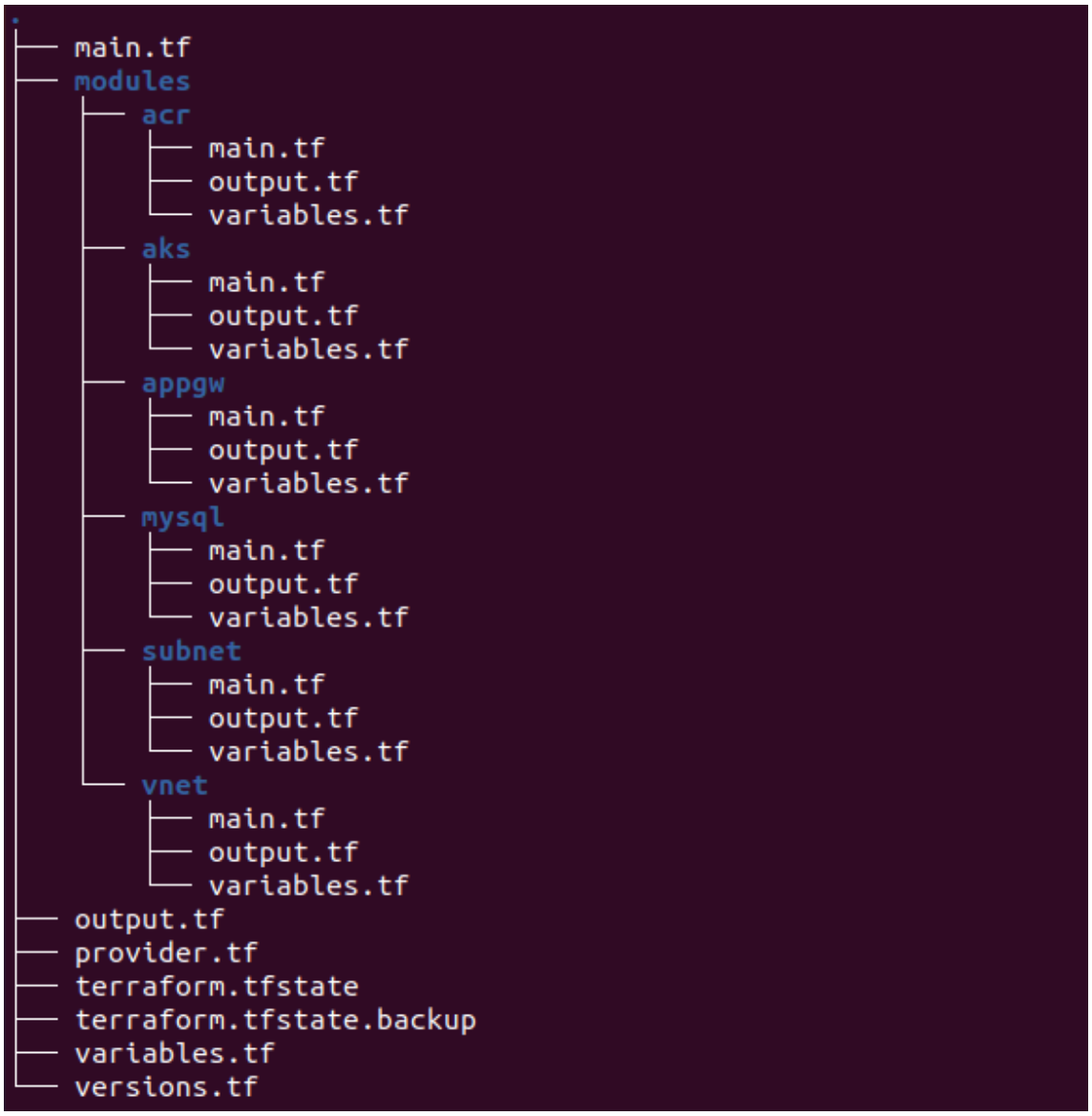
En este documento se va a explicar el código y los pasos a seguir para desplegar la infraestructura en diferentes entornos.

Este código es a nivel interno ya que no esta optimizado para entregarlo al cliente

Esta estructurado de la siguiente manera:



Como se puede observar en module definimos cada uno de los recursos necesarios para nuestra infraestructura.

Siendo en main.tf donde establecemos el orden de ejecución y dependencias de los módulos que se van a generar.

Para ejecutar nuestro código se hace mediante tres comandos

terraform init → que inicializará cargando nuestra versión y dependencias

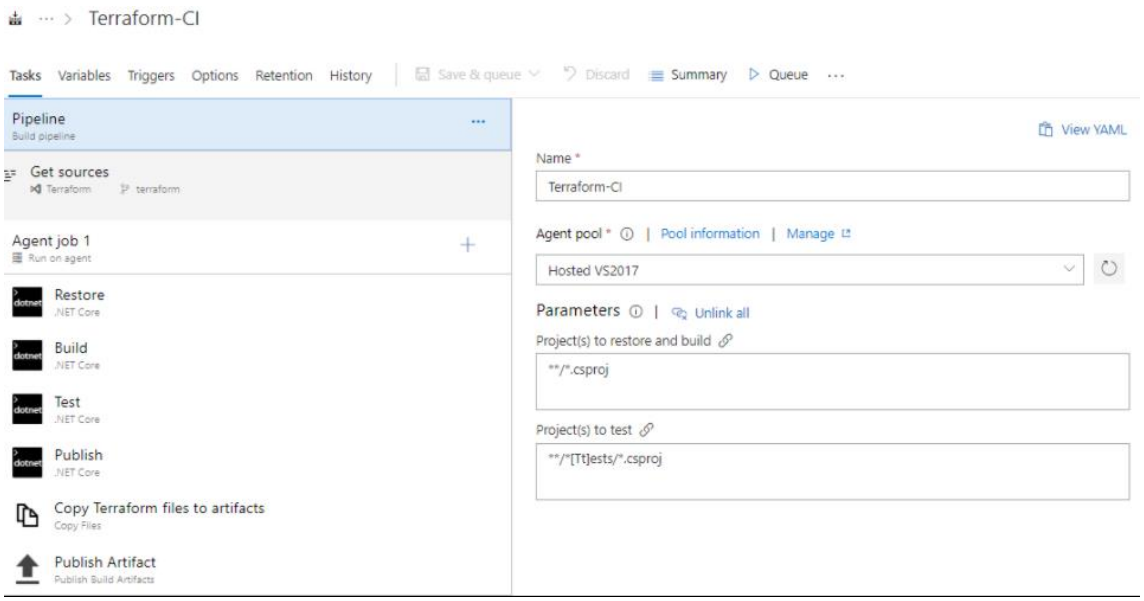
terraform plan → que hará un escaneo de nuestro código para evaluar lo que se va a generar nuevo y donde aplicará la idempotencia y nos dirá errores de configuración o de sintaxis

terraform apply → ejecuta nuestro código para generar la infraestructura

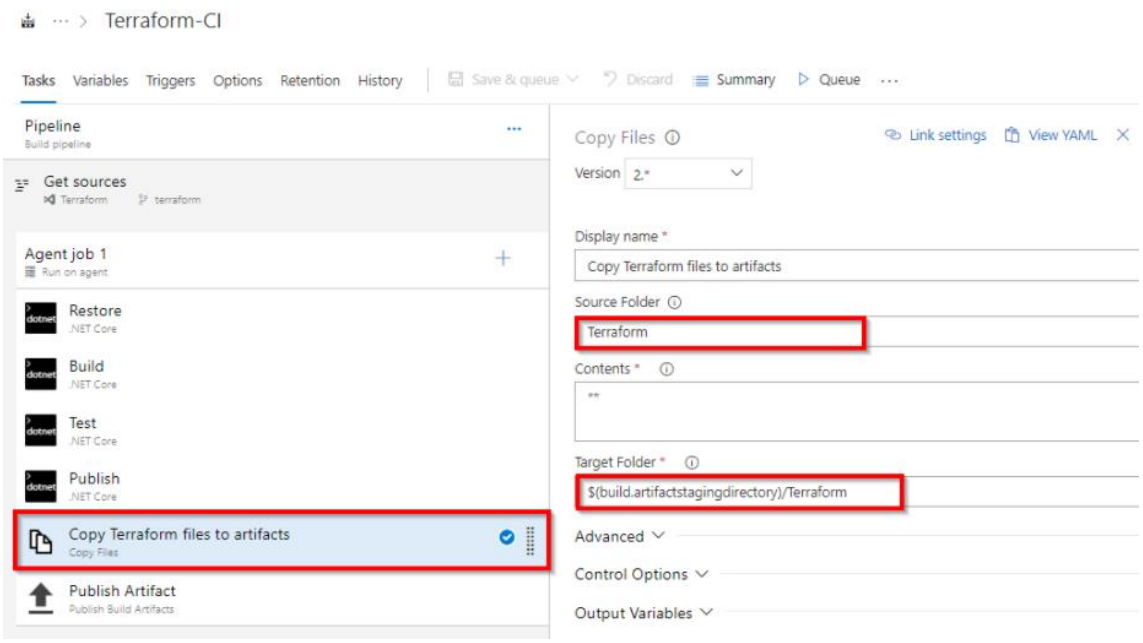
En nuestro código hay algunos valores que están fijos como por ejemplo la vnet y que para poder desplegar en diferentes entorno debes o duplicarlo(no es buena opción) o establecer un bucle que genere cuantas quieras

A la hora de establecer un pipeline en azure que lance nuestros procesos de forma automática es tan sencillo como vincular nuestro repositorio a un azure pipelines y generar su **build CI**

1º



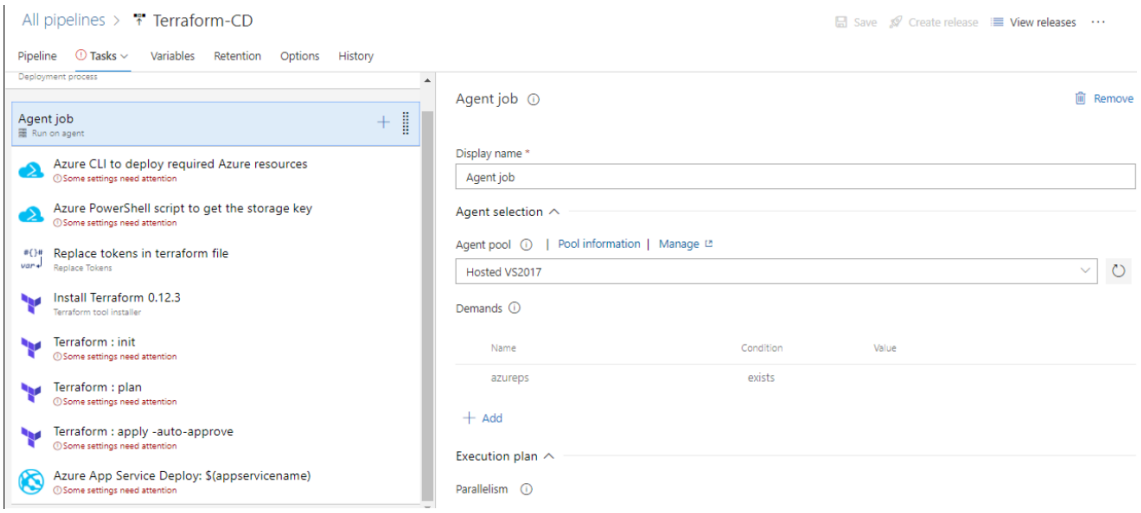
2º



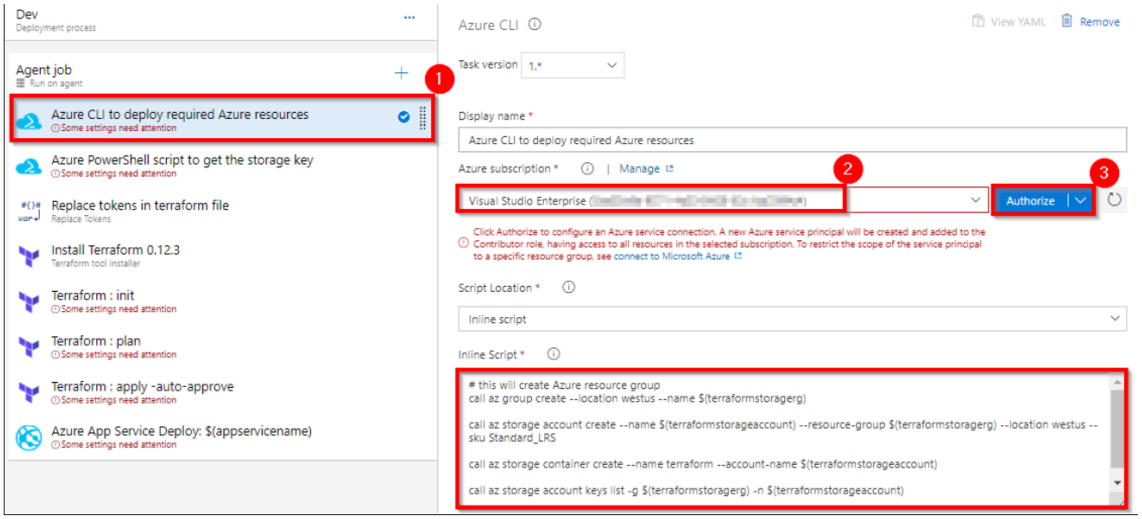
Con esto ya tendríamos nuestro Artifacts generado y listo para poder desplegarlo ¡!

Para su CD

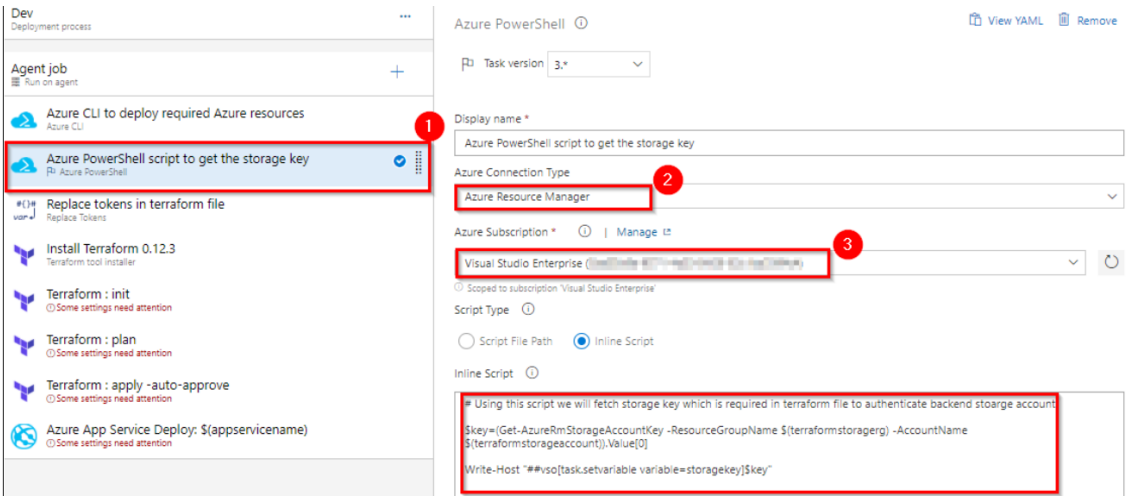
1º



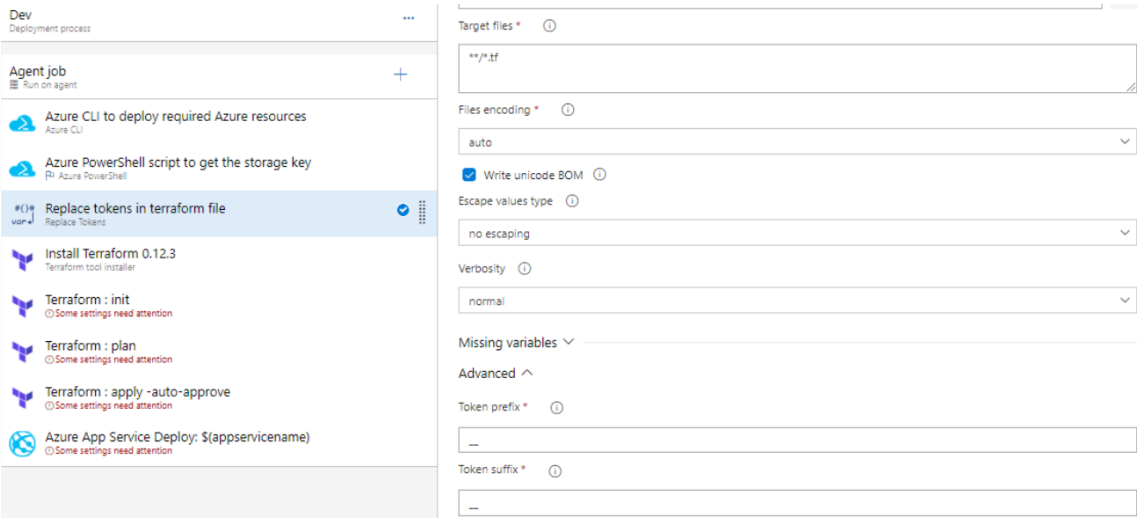
2º


Dev
Deployment process
Agent job
Run on agent
Azure CLI to deploy required Azure resources
Some settings need attention
Azure PowerShell script to get the storage key
Some settings need attention
Replace tokens in terraform file
Replace Tokens
Install Terraform 0.12.3
Terraform tool installer
Terraform : init
Some settings need attention
Terraform : plan
Some settings need attention
Terraform : apply -auto-approve
Some settings need attention
Azure App Service Deploy: $(appservicename)
Some settings need attention
Azure CLI
View YAML
Remove
Task version 1.*
1
Display name *
Azure CLI to deploy required Azure resources
Azure subscription * | Manage
2
Visual Studio Enterprise
Authorize
3
Click Authorize to configure an Azure service connection. A new Azure service principal will be created and added to the Contributor role, having access to all resources in the selected subscription. To restrict the scope of the service principal to a specific resource group, see connect to Microsoft Azure
Script Location *
Inline script
Inline Script *
this will create Azure resource group
call az group create --location westus --name $(terraformstoragerg)
call az storage account create --name $(terraformstorageaccount) --resource-group $(terraformstoragerg) --location westus --sku Standard_LRS
call az storage container create --name terraform --account-name $(terraformstorageaccount)
call az storage account keys list -g $(terraformstoragerg) -n $(terraformstorageaccount)

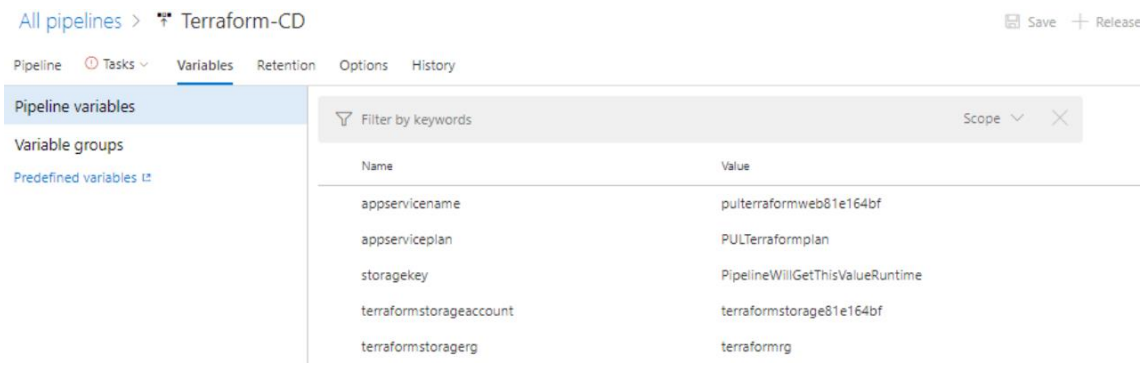
3º


Dev
Deployment process
Agent job
Run on agent
Azure CLI to deploy required Azure resources
Azure CLI
Azure PowerShell script to get the storage key
Azure PowerShell
Replace tokens in terraform file
Replace Tokens
Install Terraform 0.12.3
Terraform tool installer
Terraform : init
Some settings need attention
Terraform : plan
Some settings need attention
Terraform : apply -auto-approve
Some settings need attention
Azure App Service Deploy: $(appservicename)
Some settings need attention
Azure PowerShell
View YAML
Remove
Task version 3.*
1
Display name *
Azure PowerShell script to get the storage key
Azure Connection Type
2
Azure Resource Manager
Azure Subscription * | Manage
3
Visual Studio Enterprise
Scoped to subscription 'Visual Studio Enterprise'
Script Type
Script File Path
Inline Script
Inline Script
Using this script we will fetch storage key which is required in terraform file to authenticate backend stoarge account
$key=(Get-AzureRmStorageAccountKey -ResourceGroupName $(terraformstoragerg) -AccountName $(terraformstorageaccount)).Value[0]
Write-Host "##vso[task.setvariable variable=storagekey]$key"

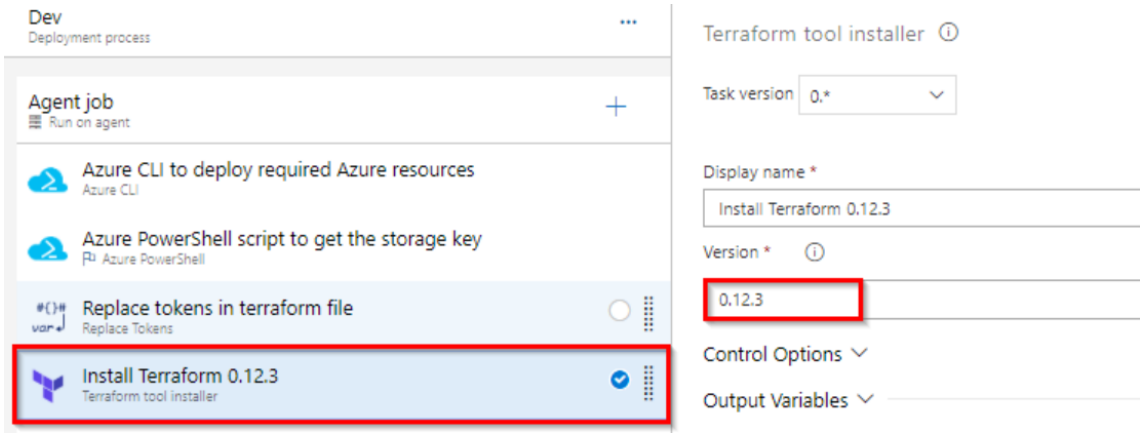
4º


Dev
Deployment process
Agent job
Run on agent
Azure CLI to deploy required Azure resources
Azure CLI
Azure PowerShell script to get the storage key
Azure PowerShell
Replace tokens in terraform file
Replace Tokens
Install Terraform 0.12.3
Terraform tool installer
Terraform : init
Some settings need attention
Terraform : plan
Some settings need attention
Terraform : apply -auto-approve
Some settings need attention
Azure App Service Deploy: $(appservicename)
Some settings need attention
Target files *
**/*.tf
Files encoding *
auto
Write unicode BOM
Escape values type
no escaping
Verbosity
normal
Missing variables
Advanced
Token prefix *
__
Token suffix *
__

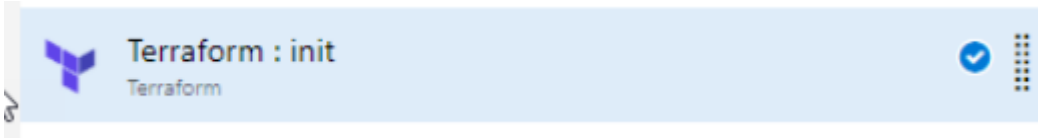
5º

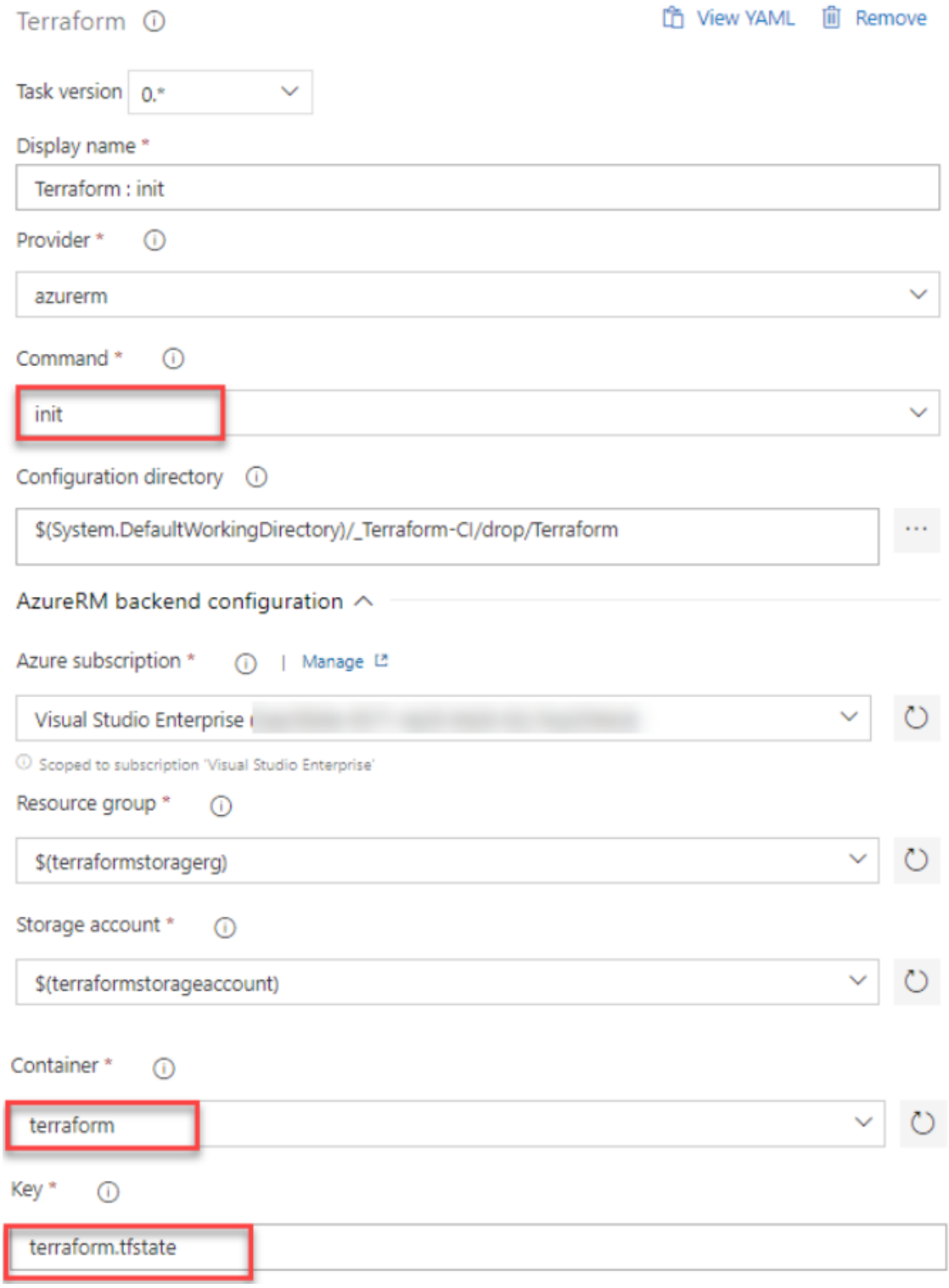

All pipelines
Terraform-CD
Save
Release
Pipeline
Tasks
Variables
Retention
Options
History
Pipeline variables
Variable groups
Predefined variables
Filter by keywords
Scope
Name
Value
appservicename
pulterraformweb81e164bf
appserviceplan
PULTerraformplan
storagekey
PipelineWillGetThisValueRuntime
terraformstorageaccount
terraformstorage81e164bf
terraformstoragerg
terraformrg

6º

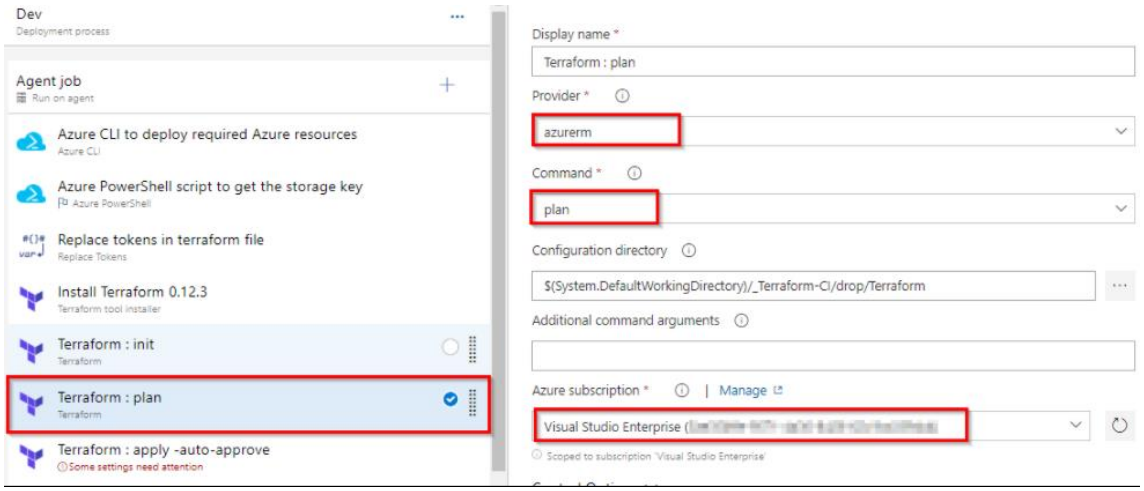

Dev
Deployment process
Agent job
Run on agent
Azure CLI to deploy required Azure resources
Azure CLI
Azure PowerShell script to get the storage key
Azure PowerShell
Replace tokens in terraform file
Replace Tokens
Install Terraform 0.12.3
Terraform tool installer
Terraform tool installer
Task version
0.*
Display name *
Install Terraform 0.12.3
Version *
0.12.3
Control Options
Output Variables

7º

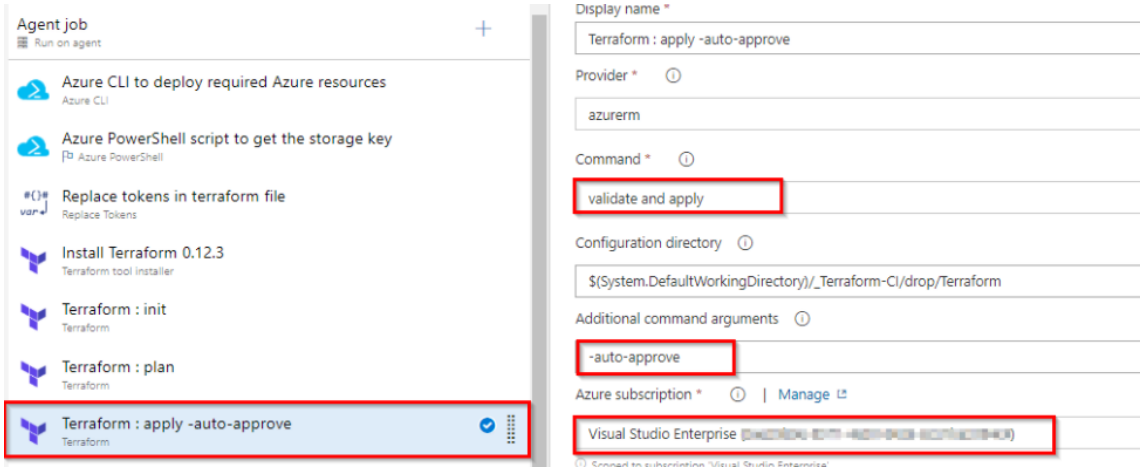

Terraform : init
Terraform


Terraform
View YAML
Remove
Task version
0.*
Display name *
Terraform : init
Provider *
azurerm
Command *
init
Configuration directory
$(System.DefaultWorkingDirectory)/_Terraform-CI/drop/Terraform
AzureRM backend configuration
Azure subscription *
Manage
Visual Studio Enterprise
Scoped to subscription 'Visual Studio Enterprise'
Resource group *
$(terraformstoragerg)
Storage account *
$(terraformstorageaccount)
Container *
terraform
Key *
terraform.tfstate

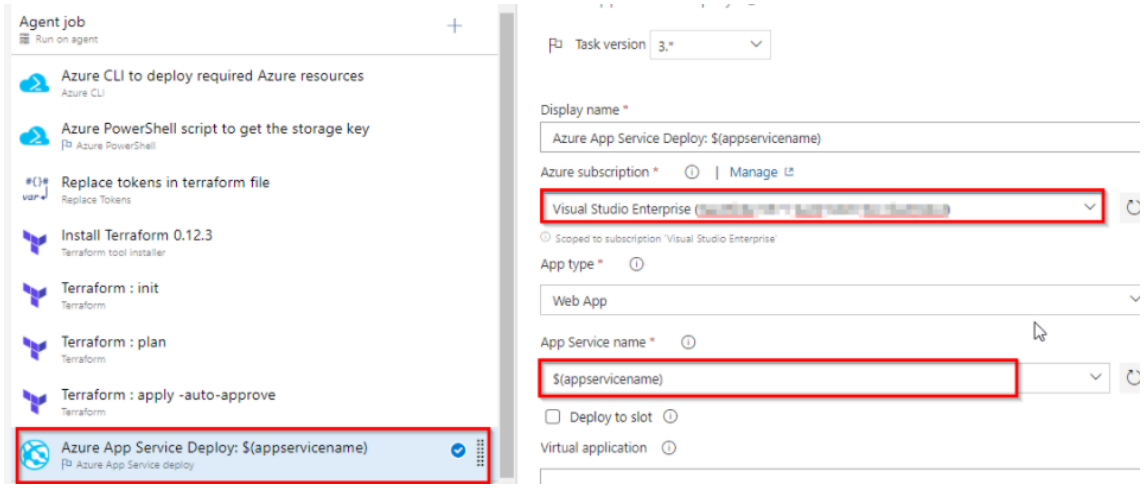
8º



9º



10º



Con estos sencillos pasos tendríamos automatizado el proceso de despliegue de nuestra infraestructura