**Deploying of Terraform code using Azure DevOps CI/CD pipeline**

**(Using Microsoft Hosted Agent machine)**

Deploying infrastructure as code (IaC) using Terraform is a crucial practice for modern cloud management. Integrating this with Azure DevOps CI/CD pipelines brings automation, consistency, faster deploying and reliability to your infrastructure building.

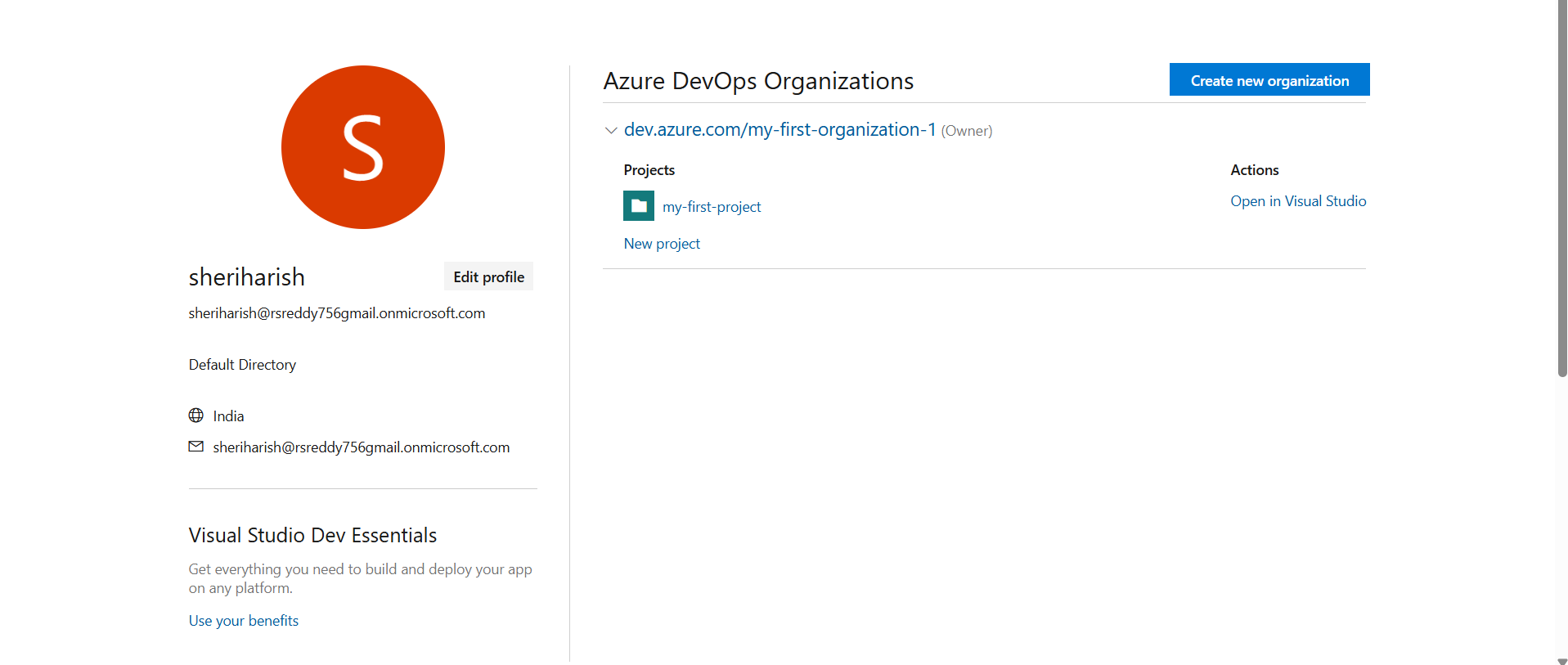
By using Azure pipeline we can easily and fastly deploy our application or infrastructure in the cloud.

Here we firstly push the terraform code (VM creation code) to the Azure Repository (Azure repo) and then we perform the CI and CD.

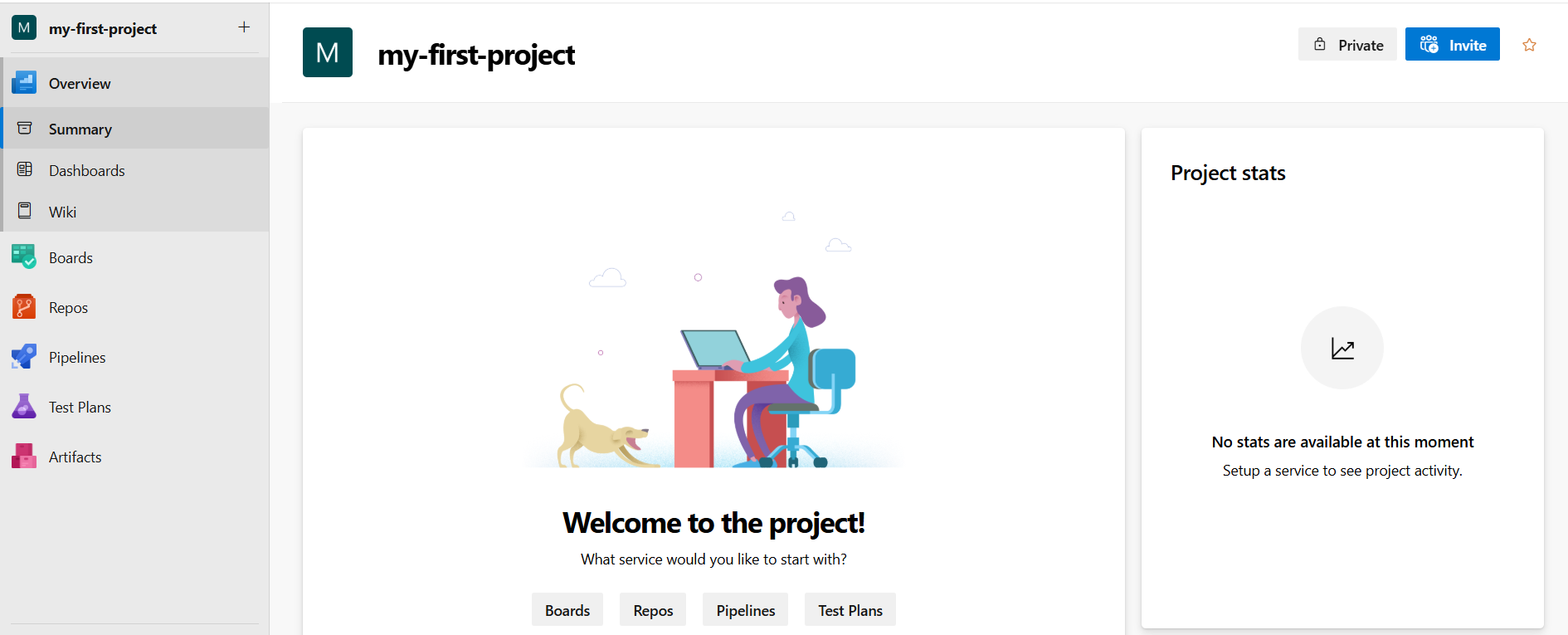
**Note:** We know that terraform code (IAC) does not required any build engine tools like Apache maven, MSBuild, Gradle, So we it does not need CI (continuous integration), it directly build the infrastructure in azure.

**Let’s do it step by step in practically:**

**Step1:** In Azure DevOps create the Organization (my-first-organization-1) within it Create the project (my-first-project).

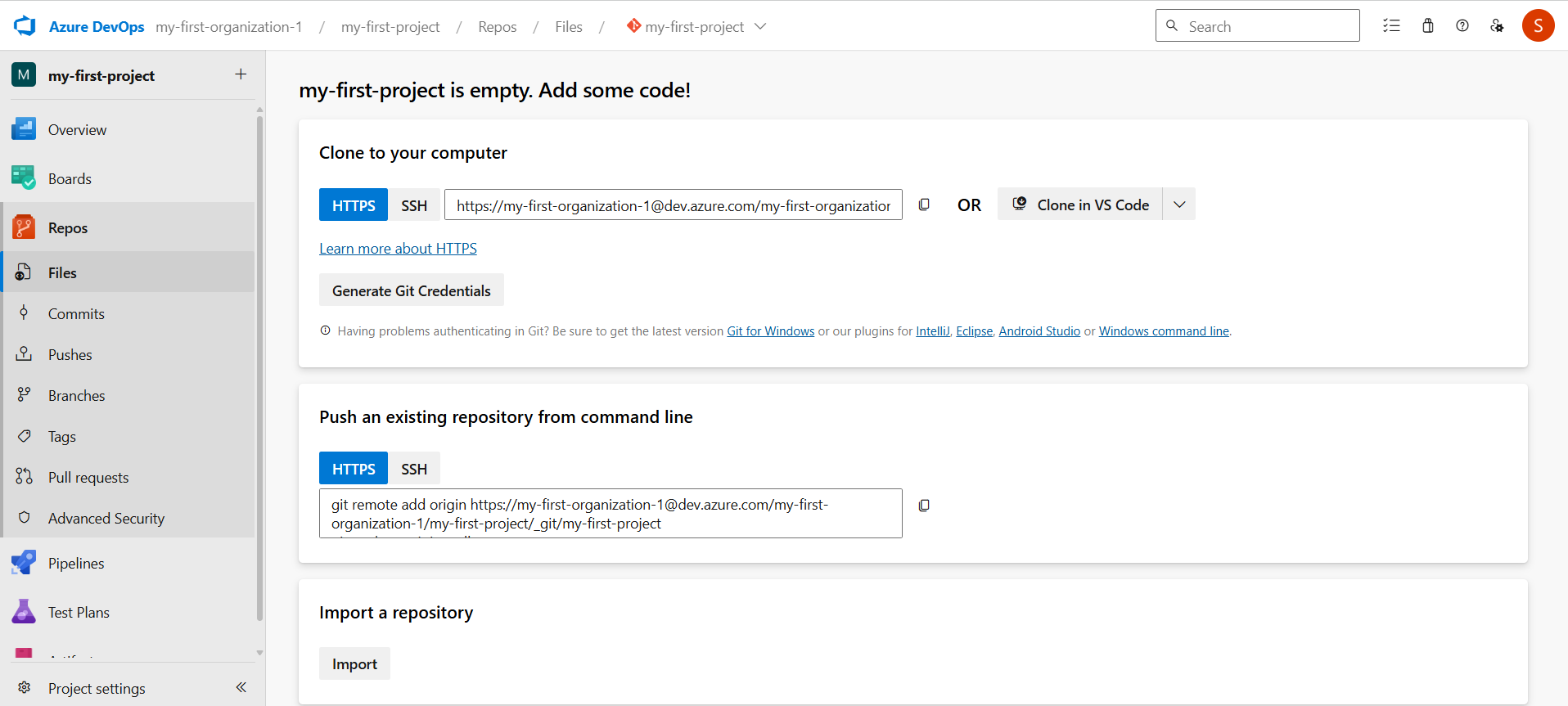
****

Within a project we can find the **Azure repository** which is a version control system provided by Microsoft azure.

****

**Fig:** Repos (Azure Repository).

Name of the repository is same as the name of the project by default as shown in below figure. We can also change it manually.



**Step2**: Push the Terraform code to the Azure Repository (my-first-project).

**Note:** In this process we no need provide authentication in main.tf file. We only passes the provider information like whether it is Azure or AWS or GCP.

#provider Block

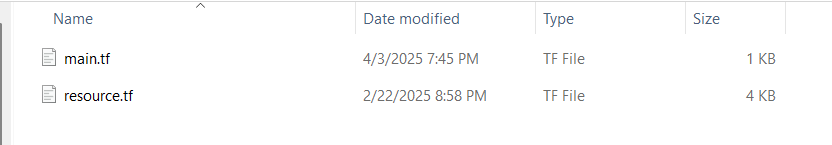
provider "azurerm" {

  features {}

}

Fig: Main.tf file

And the **resource.tf** file consist of VM creation information like resource group, network, subnet, public IP, NSG and so on.



Now Push these two terraform code file (main.tf & resource.tf) to the Azure repository (my-first-project).