**End-to-End Web App Deployment on Azure Using Terraform and Azure DevOps**

**Overview:**

This project demonstrates an automated end-to-end deployment pipeline for a .NET web application hosted on a Windows Virtual Machine (VM) in Azure. It utilizes Terraform for infrastructure provisioning, PowerShell for VM configuration, and Azure DevOps for CI/CD pipeline management. The goal is to deploy a secure, scalable, and maintainable production-like environment with minimal manual effort.

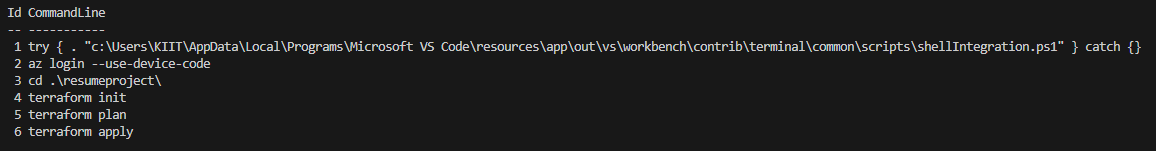
**Tech Stack**

* **Terraform** – Infrastructure as Code (IaC) to automate Azure resource provisioning
* **Azure Virtual Machines** – Windows-based hosting environment
* **.NET Hosting Bundle** – Runtime for ASP.NET Core apps on IIS
* **PowerShell** – VM and service configuration scripts
* **Azure DevOps** – Source control, pipeline, and deployment automation
* **GitHub** – Source code repository (imported into Azure Repos)
* **IIS (Internet Information Services)** – Windows web server for hosting the application

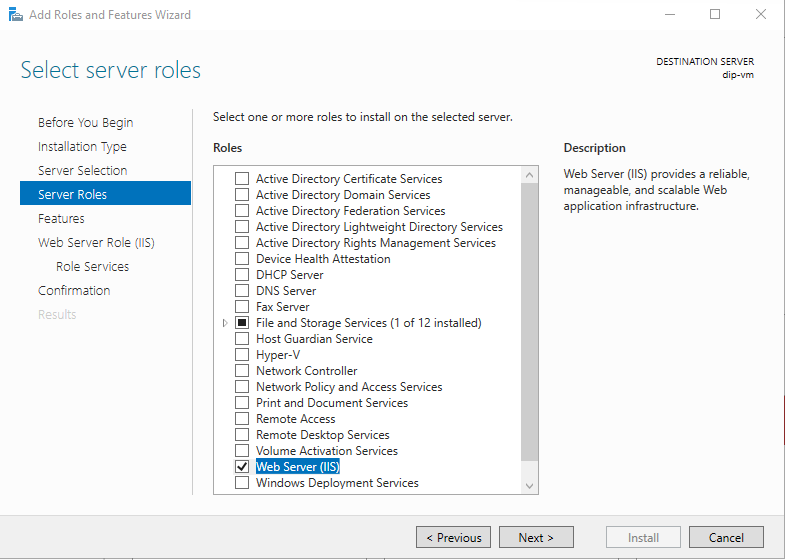
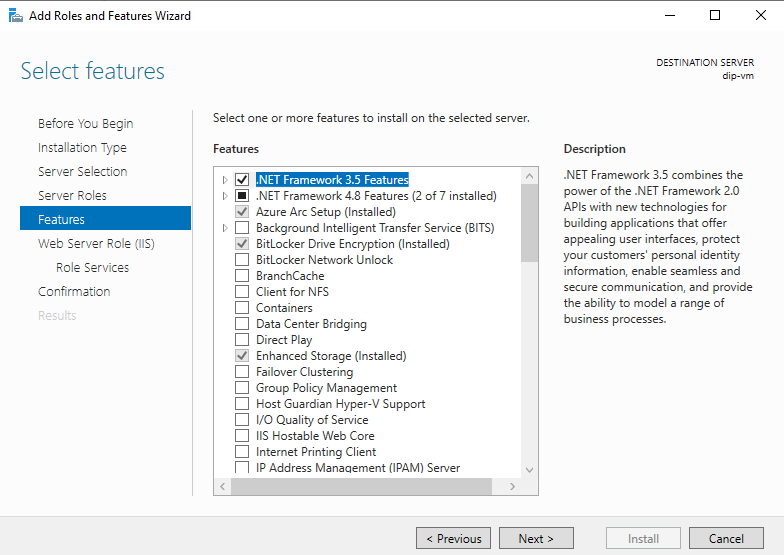
**Steps:**

1. **Infrastructure Setup with Terraform**

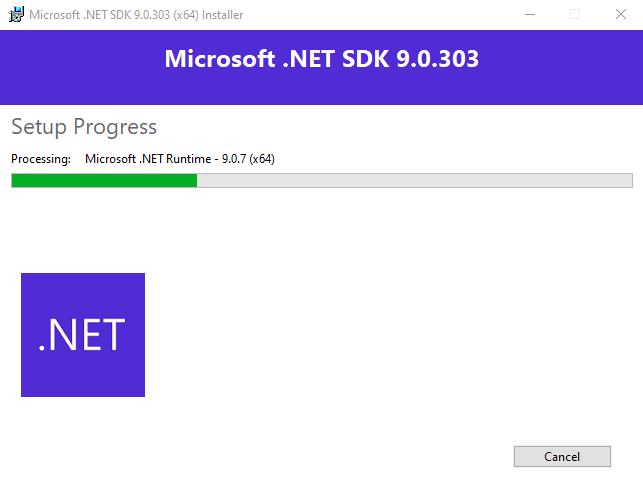
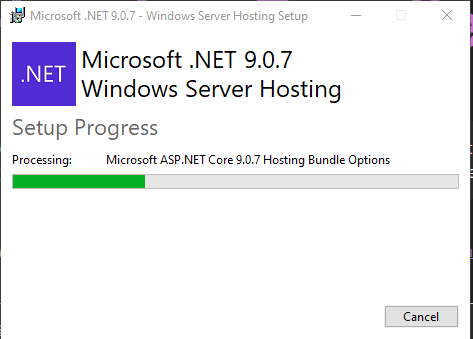
* Create Resource Group
* Create Virtual Network and Subnet
* Create Windows VM with public IP
* Attach Network Security Group allowing RDP and HTTP
* Run the Terraform script in terminal



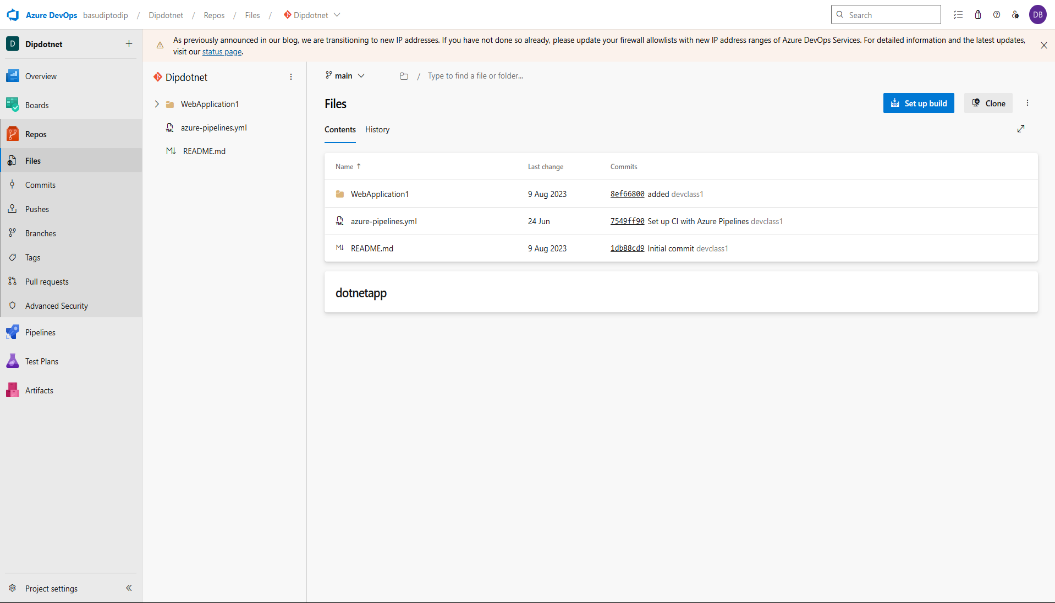
1. **Set up the VM with RDP**

* Add Web server(IIS) under Server Roles
* Add .Net Framework 3.5 Features

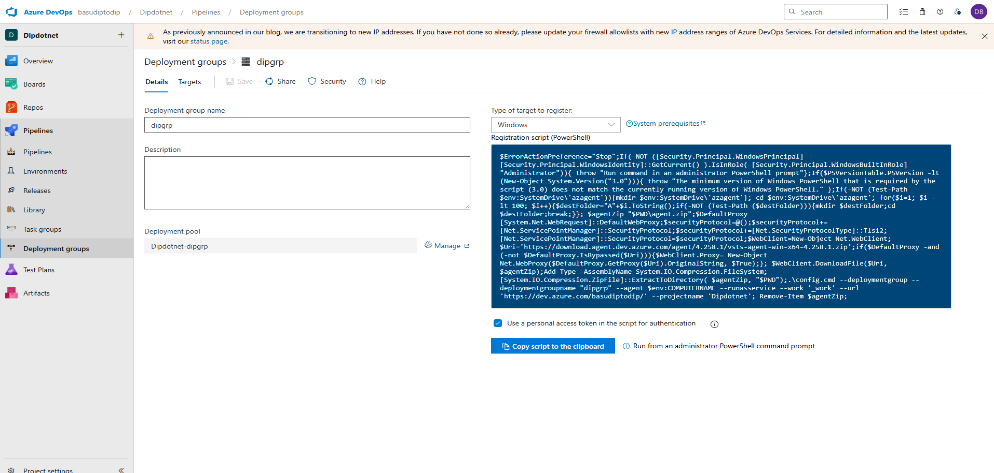
1. **Installing .NET and Hosting Bundle**

* 
* ****

1. **Importing GitHub Repo into Azure Repos**

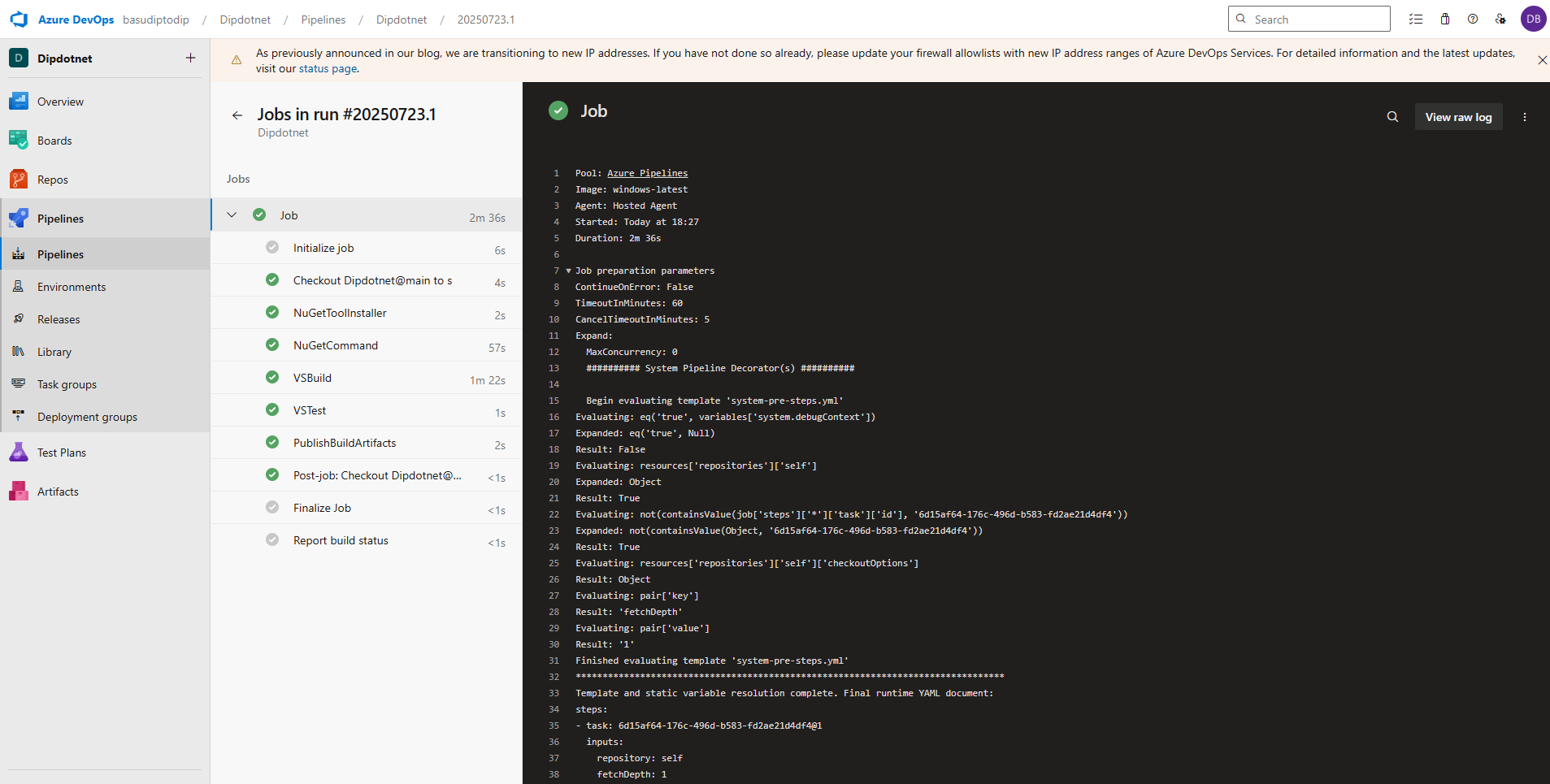
* Go to Azure Devops Dashboard
* Go to Azure repos
* Click Import a repository
* Paste your GitHub repo URL

1. **Create Deployment Group**

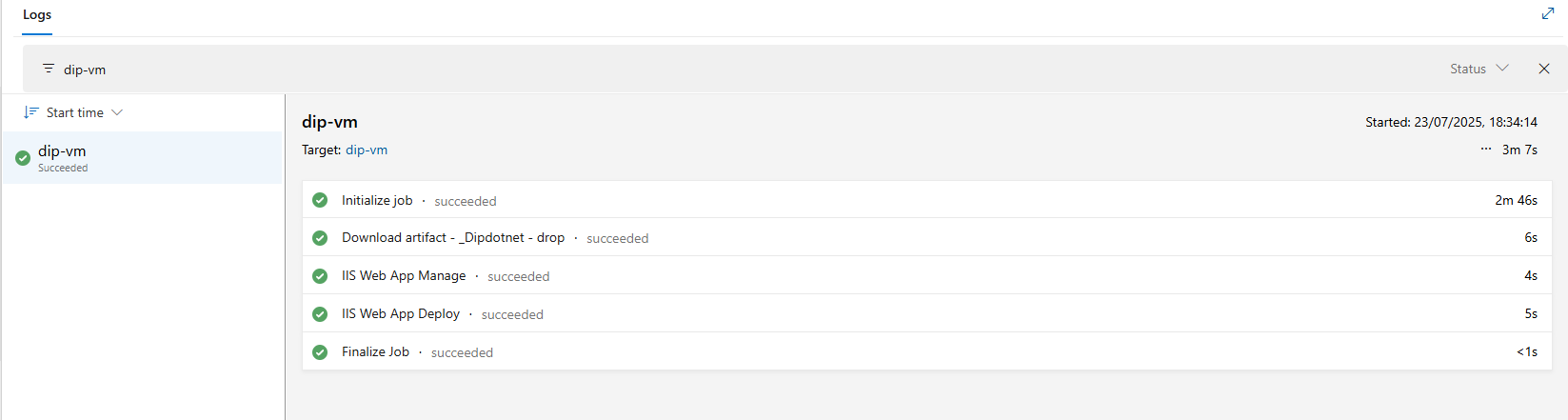
* Go to **Pipelines > Deployment groups**
* Create new group and copy the registration PowerShell script
* Run it on your Azure VM to register the agent

1. **Create a New Pipeline**

* Go to Pipelines > Add Pipeline
* Select source as azure repos

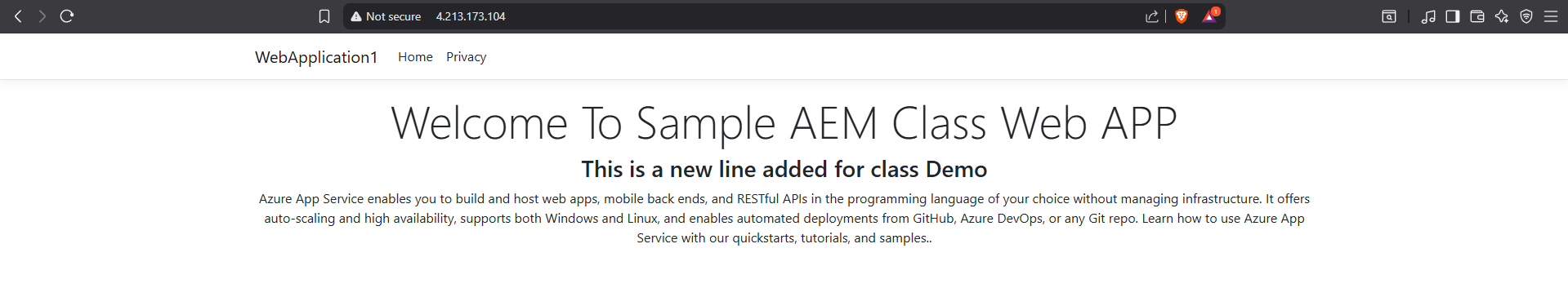


1. **Release Pipeline Configuration**

* Go to **Pipelines > Releases** and create a new release pipeline
* Add an artifact from the build pipeline
* Create Release

**Outcome**

* A Windows VM is created and configured
* .NET Hosting Bundle is installed
* Code is deployed from GitHub via Azure DevOps release pipeline
* IIS is hosting the deployed web app

****