

Deploying a Python Http API on Azure

This is an example solution for deploying a Python Http API on Azure using Terraform and Azure DevOps. The solution includes the following components:

- A Python Http API that exposes a `/live` endpoint
- An Azure App Service to host the Http API
- An Azure MySQL Database to store data used by the Http API
- Azure Key vault and custom domain for protection.
- Azure Monitor and Log Analytics for logging and monitoring the Http API and MySQL Database
- Azure DevOps pipeline to deploy the infrastructure and Http API to Azure

Prerequisites

Before you can deploy the solution, you need to have the following:

- An Azure subscription
- An Azure DevOps organization

Deployment

To deploy the solution:

1. Open the Azure DevOps organization in your browser and create a new pipeline.
2. Choose Azure Repos Git as the source and select the cloned repository.
3. Choose the `azure-pipelines.yml` file in the root directory as the pipeline definition.
4. Update required variables in Terraform and Pipeline like Subscription, Resource Group, Location and Storage Account Name etc.
5. Save and run the pipeline.

After the pipeline completes successfully, the Http API should be deployed and accessible through the Azure App Service.

Conclusion

In this example, we deployed a Python Http API to Azure using Terraform and Azure DevOps. This is just a starting point, and you can customize the solution to fit your needs. Like we can use Virtual Networks, Security Groups and Application Gateways to protect our App more.