### Detailed Deployment Document

#### Table of Contents

1. **Introduction**
2. **Prerequisites**
3. **Bicep Template**
   * Infrastructure Overview
   * Template Structure
4. **Setting up Azure DevOps Pipeline**
   * Azure DevOps Overview
   * Pipeline Configuration
5. **Deployment Steps**
   * Step 1: Configure Azure Service Connection
   * Step 2: Create Azure DevOps Pipeline
   * Step 3: Execute Deployment
6. **Accessing the Deployed VMs**
   * Using SSH (Linux)
7. **Post-deployment Configuration**
   * Optional: Customizations and Additional Configurations
8. **Troubleshooting**
   * Common Issues and Resolutions
9. **Conclusion**
   * Summary and Next Steps

### 1. Introduction

This document provides a comprehensive guide to deploying Azure infrastructure using Bicep, setting up automation with Azure DevOps, and accessing the deployed VMs securely through a Load Balancer's Public IP.

### 2. Prerequisites

Before you begin, ensure you have:

* An Azure subscription.
* Access to an Azure DevOps project.
* Basic familiarity with Azure resources and Azure CLI.

### 3. Bicep Template

#### Infrastructure Overview

The Bicep template (azuredeploy.bicep) provisions the following resources:

* Virtual Network (VNet) with a subnet.
* Network Security Group (NSG) for VMs.
* Azure Load Balancer with a Public IP.
* Two Virtual Machines (VMs) deployed in a private subnet.
* Network Interfaces (NICs) for each VM.

#### Template Structure : Please refer azuredeploy.bicep file.

### 4. Setting up Azure DevOps Pipeline

#### Azure DevOps Overview

Azure DevOps allows you to automate deployments using pipelines. We'll set up a pipeline to deploy the Bicep template.

#### Pipeline Configuration: Please refer azure-pipelines.yml file.

### 5. Deployment Steps

#### Step 1: Configure Azure Service Connection

1. Navigate to your Azure DevOps project.
2. Go to Project Settings -> Service connections.
3. Click on New service connection and select Azure Resource Manager.
4. Follow the prompts to authenticate and create the service connection.

#### Step 2: Create Azure DevOps Pipeline

1. Go to Pipelines in Azure DevOps.
2. Click New Pipeline and select your repository.
3. Choose Existing Azure Pipelines YAML file and select azure-pipelines.yml.
4. Save and commit the pipeline configuration.

#### Step 3: Execute Deployment

1. Trigger the pipeline manually or configure automatic triggers.
2. Monitor the pipeline execution in Azure DevOps.

### 6. Accessing the Deployed VMs

#### Using SSH (Linux)

To SSH into the deployed VMs:

1. Obtain the Load Balancer's Public IP from the Azure portal.
2. Open a terminal and use SSH with the following commands:
3. To SSH into each VM behind the Load Balancer, use the following commands:

 **Load Balancer Configuration**: The Load Balancer (myLB) is configured with:

* Frontend IP configuration (LoadBalancerFrontEnd) using a static Public IP (myLBPublicIP).
* Backend pool (BackendPool1) where VMs (myVM1 and myVM2) are members.
* Inbound NAT rules (SSHAccessVM1 and SSHAccessVM2) to map external port (e.g., 50001 and 50002) to internal SSH port (22) on each VM.

 **NSG and Security**: NSG (myNSG) includes a rule (AllowSSHFromLB) that allows SSH traffic from the Load Balancer's frontend IP (lbFrontEndIP.properties.ipAddress).

 **VM Configuration**: Two VM resources (vm1 and vm2) are defined with respective NICs (nic1 and nic2) attached to the subnet.

**For VM1:**

ssh -p 50001 azureuser@<LoadBalancerPublicIP>

**For VM2:**

ssh -p 50002 azureuser@<LoadBalancerPublicIP>

Replace <LoadBalancerPublicIP> with the Load Balancer's actual Public IP address.

Ensure you replace azureuser with your VM's admin username.

### 7. Post-deployment Configuration

Consider additional configurations such as:

* Installing software packages.
* Configuring firewall rules.
* Setting up monitoring and alerts.

### 8. Troubleshooting

#### Common Issues and Resolutions

* **Pipeline Failures**: Check pipeline logs for errors.
* **SSH Issues**: Verify NSG rules and Load Balancer configuration.
* **Deployment Errors**: Review Azure CLI output for details.

### 9. Conclusion

This document has provided a structured approach to deploying Azure infrastructure using Bicep, setting up automation with Azure DevOps, and securely accessing deployed VMs through a Load Balancer's Public IP. Follow these steps to deploy your infrastructure efficiently and securely.