**CICD Design**

**Branching Strategy/GitFlow**

Graphical user interface, application

Description automatically generated

|  |  |
| --- | --- |
| **1** | Master/Main branch - represents a future deployment to production. Have branch protection Rule enabled, without pull request changes cannot be merged. |
| **2** | Hotfixes can be applied to Main and merged back into develop branch. |
| **3** | Release branch will be merged to Main branch and taken from develop |
| **4** | Develop branch will be merged to Main. This branch will be deployed to development, test and staging region. Have branch protection Rule enabled, without pull request changes cannot be merged. |
| **5** | Feature branches will merge to Develop branch which is the CI branch |

**Image Pipeline**

Image will be published in dev subscription and using RBAC we can give read only permission to test, staging and production subscriptions. Admin credentials are stored in secrets in Azure keyvault.

Diagram

Description automatically generated

|  |  |
| --- | --- |
| 1 | Window server 2019 image will be picked from Azure Market place |
| 7 | New Image gallery can be created simultaneously. Can be controlled using parameters. |
| 2 | New virtual machine will be spinned. Using a powershell script/extensions, required set of software will be installed. |
| 3 | Specialized VM Image will be captured and store it in image gallery |
| 4 | Using Access IAM of image, RBAC access will be given to subscriptions for other environment (test, staging and production) |
| 8 | Replication to another region can be done. Can be controlled using parameters |
| 5 | Final copy of image is published in CI flow. Will be used in further steps |
| 6 | Delete Vm |

**References:-**

[Create shared Azure Linux VM images using the portal - Azure Virtual Machines | Microsoft Docs](https://docs.microsoft.com/en-us/azure/virtual-machines/linux/shared-images-portal#share-the-gallery)

[Image sharing across multiple subscription in same tenant - Microsoft Q&A](https://docs.microsoft.com/en-us/answers/questions/514994/image-sharing-across-multiple-subscription-in-same.html)

[Share VM images in a compute gallery - Azure Virtual Machines | Microsoft Docs](https://docs.microsoft.com/en-us/azure/virtual-machines/shared-image-galleries?tabs=azure-cli)

**Infrastructure pipeline**

**Prerequisites**

Create service principle for dev, test, staging and production subscriptions

Store service principle and subscriptions details in Github secrets.

A picture containing diagram

Description automatically generated

|  |  |
| --- | --- |
| 1 | Published image from image pipeline |
| 2 | Bicep template stored in Github will refer published image. Pipeline as a code is also stored in github workflows |
| 3 | GH action will use checkov to lint bicep code |
| 4 | GH action will create package for all bicep files |
| 5 | Deploy package to Azure Artifacts |
| 6 | Azure pipeline will download and extract the package from Azure Artifacts and deploy the infrastructure on template |
| 7 | Infrastructure is deployed on cloud. Up and running. |

**Application pipeline**

**Graphical user interface

Description automatically generated with medium confidence**

|  |  |
| --- | --- |
| 1 | Monorepo in Github |
| 2 | Image pipeline will be triggered. If parameter A is selected. |
| 3 | Infrastructure pipeline will be triggered. If parameter B is selected. |
| 4 | Code is built using GHAction in continuous integration phase. |
| 5a | Github hosted runners – Standard runners with default set of softwares |
| 5b | Github self-hosted runners are Azure VM with required set of software not available in standard edition |
| 6a | External dependencies are .net dependencies downloaded from nuget.org or similar sites required to build the code |
| 6b | Internal dependencies are the Custom .net packages developed by customer (like DLL files) required to build the code |
| 7 | Deployable artifacts stored in Azure Artifacts |
| 8 | Pipeline used to do continuous deployment |
| 9a | Microsoft hosted agents – standard agents with default set of softwares |
| 9b | Microsoft self-hosted agents are Azure VM with required set of software not available in standard edition |
| 10a | Non-production environment is supported by non-production agent pool |
| 10b | Production environment is supported by production agent pool |
| 11 | Landing zone to support blue/green deployment |
| 12 | After Production deployment, DR of VMSS will be updated. |

**Blue/Green Deployment**

Create a new VMSSBlue with new version of software and tag=new.

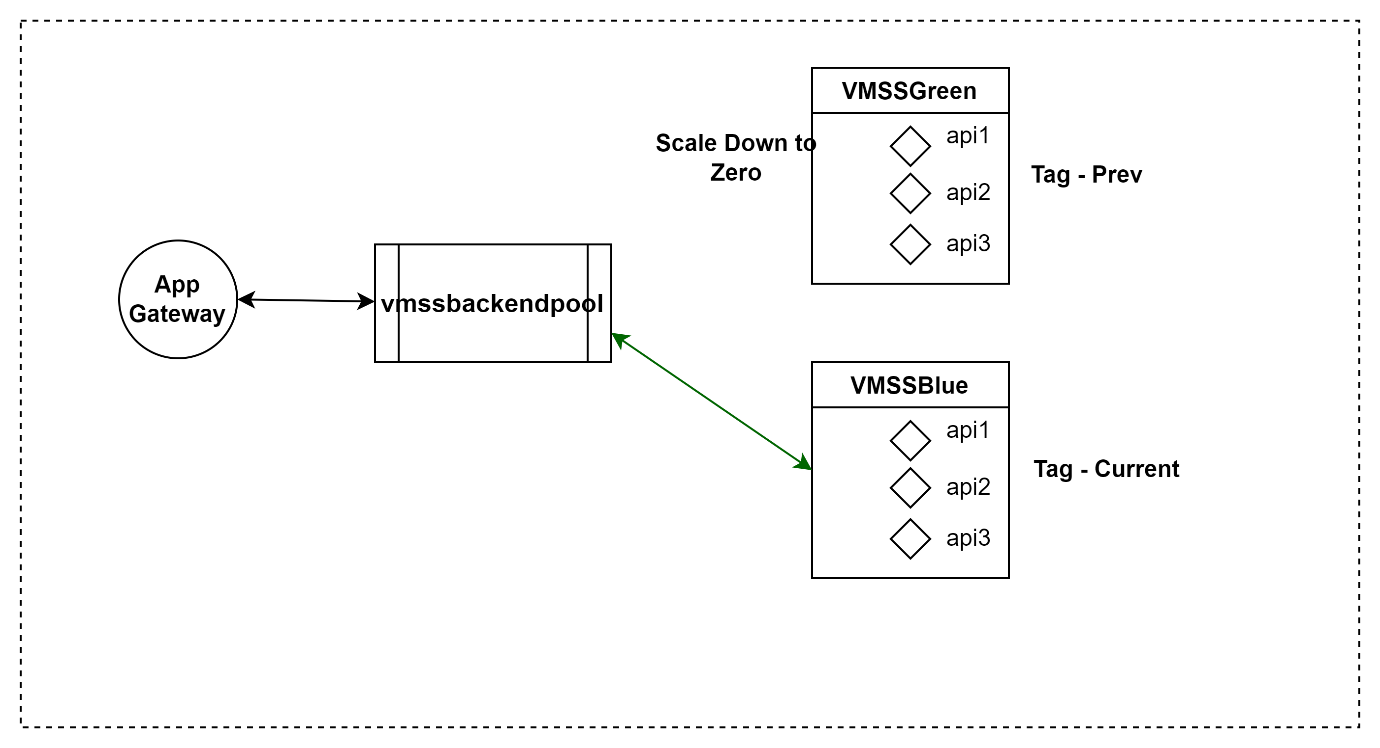
Update Backendpool and add VMSSBlue in backendpool

Once VMSSblue becomes healthy, update its tag to current.

Remove vmssgreen from backendpool and scale it down to zero and update its tag to prev

**Diagram

Description automatically generated**

****

**DR**

**Strategy1**

Whole infrastructure will always be present in Melbourne region without any instance. Whenever DR region is required vmss will be scaled and CNAME record in traffic manager will be upgraded using azure runbook to point to A record of DR region.

**Drawbacks: -**

We need to pay cost of application gateway and other infrastructure all the time.

**Strategy2**

Deploy the whole infra structure using Application pipeline in DR region and can be controlled using parameters.

**Drawbacks: -**

It will require time to get whole infrastructure ready. Downtime for application.