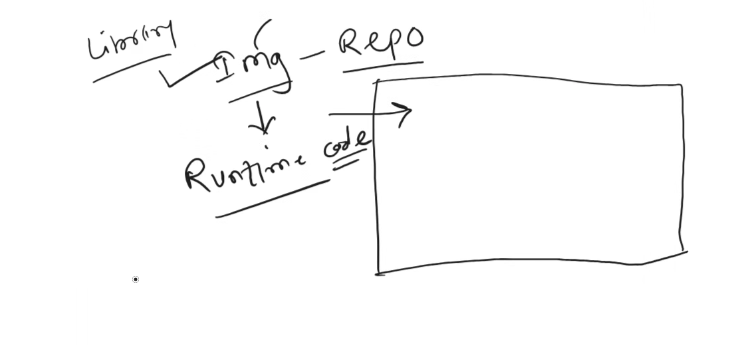
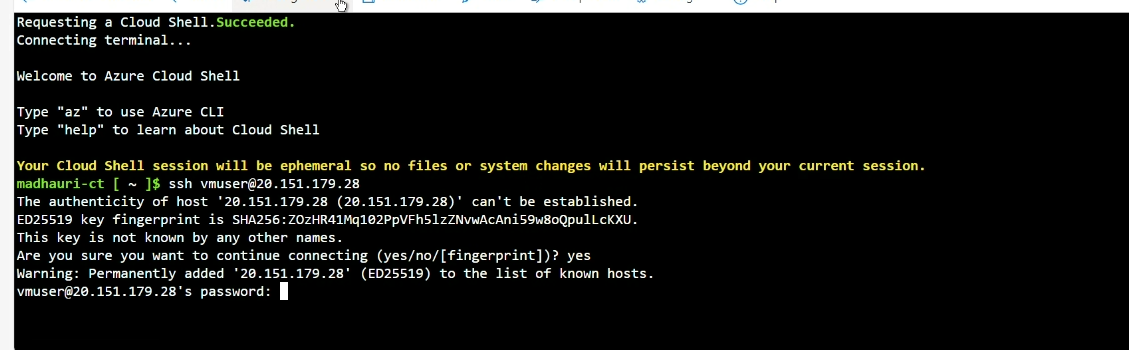
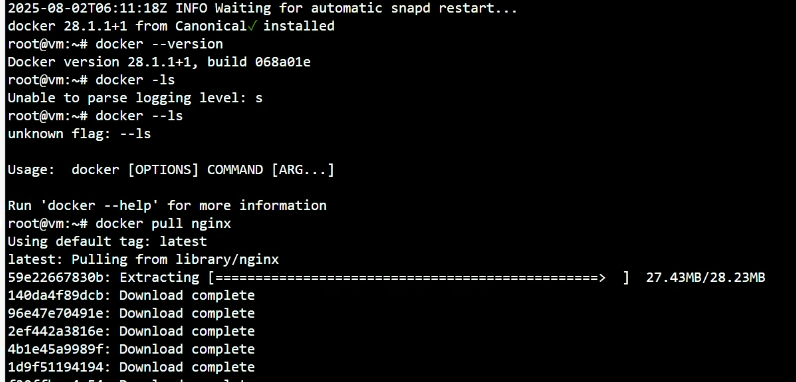
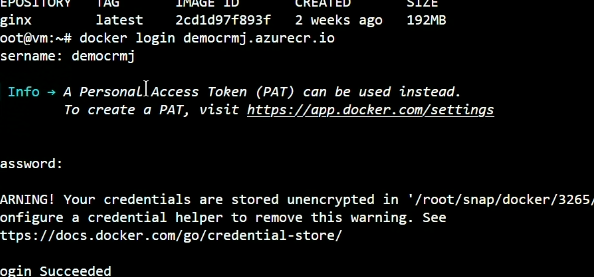


<https://learn.microsoft.com/en-us/azure/devops/pipelines/tasks/reference/publish-build-artifacts-v1?view=azure-pipelines>



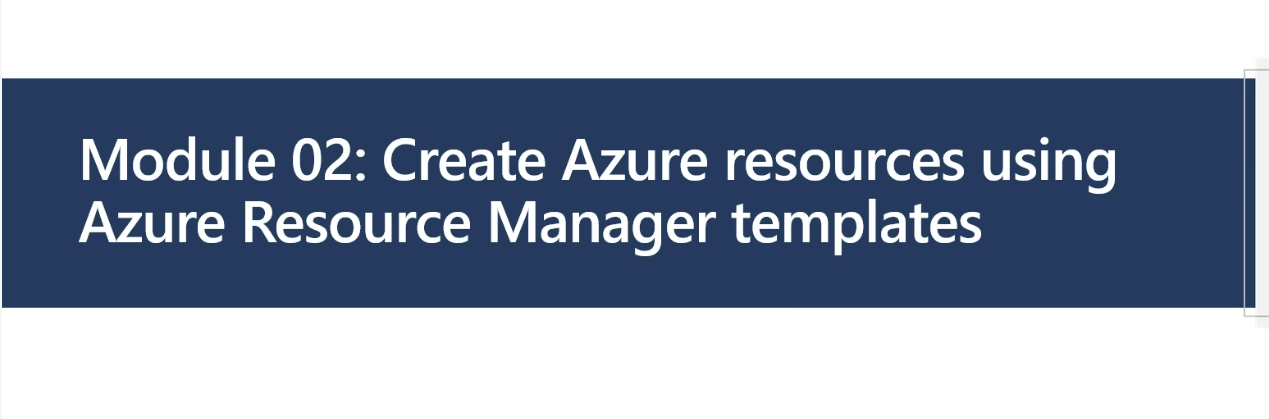
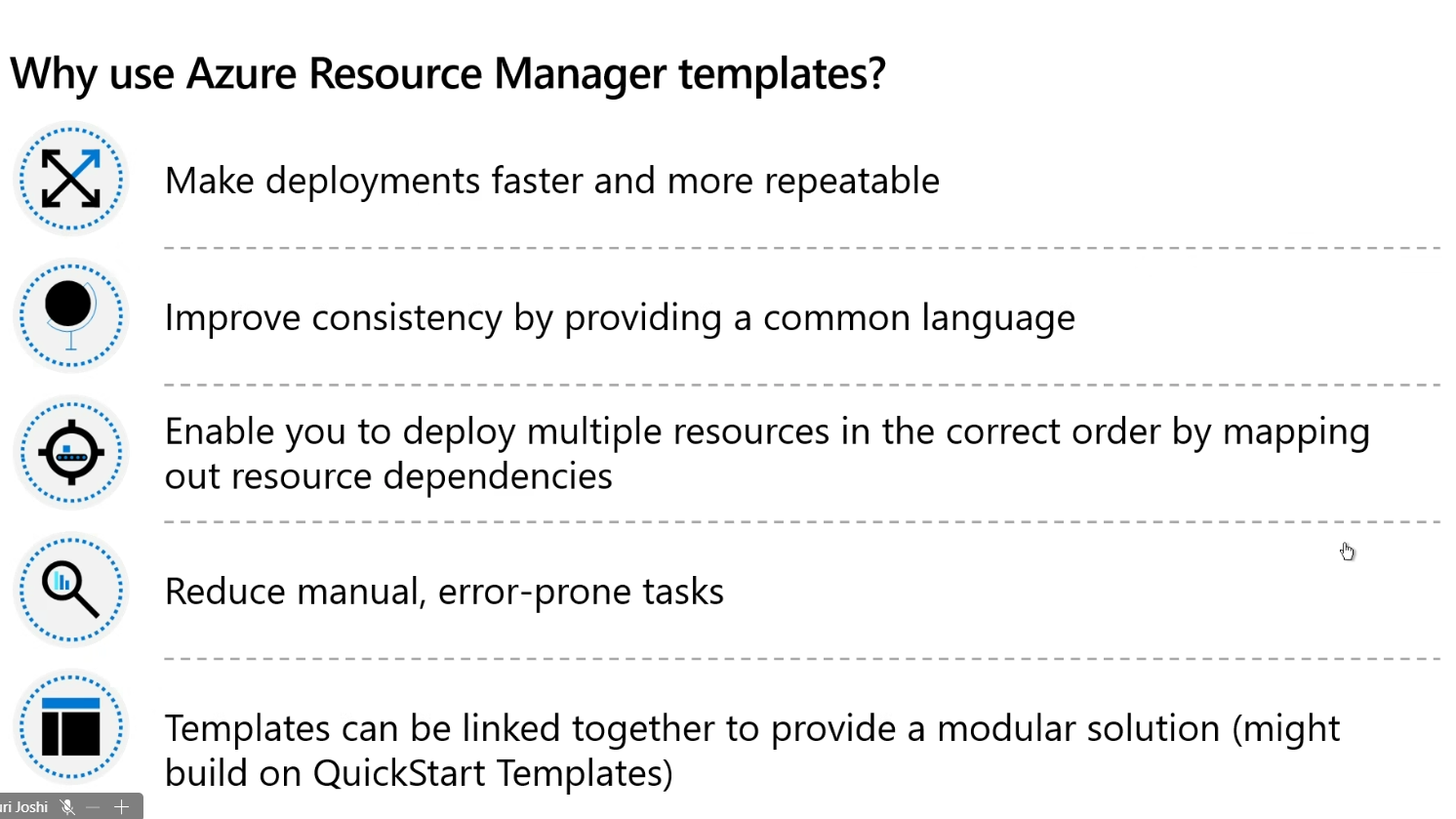


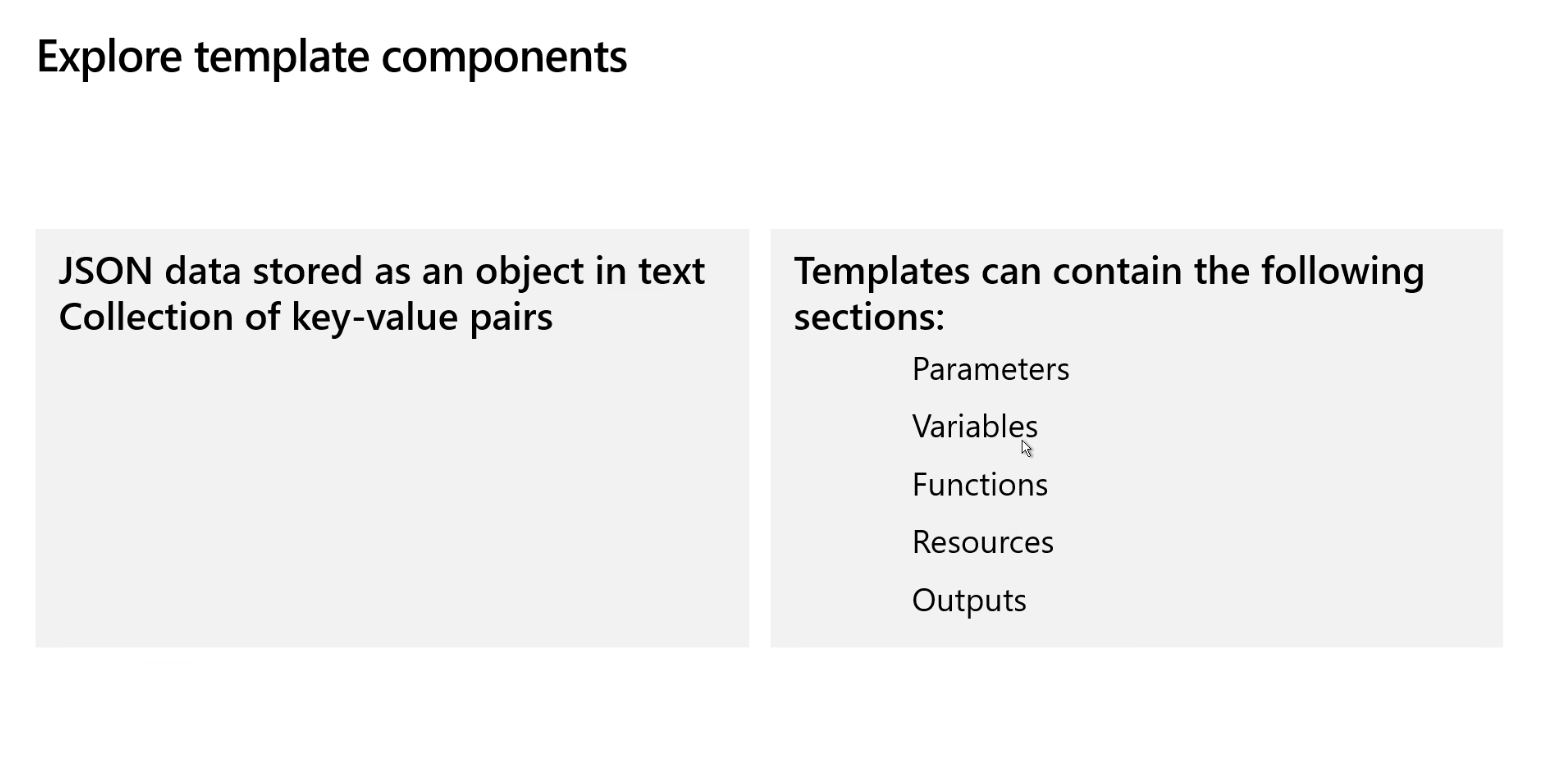
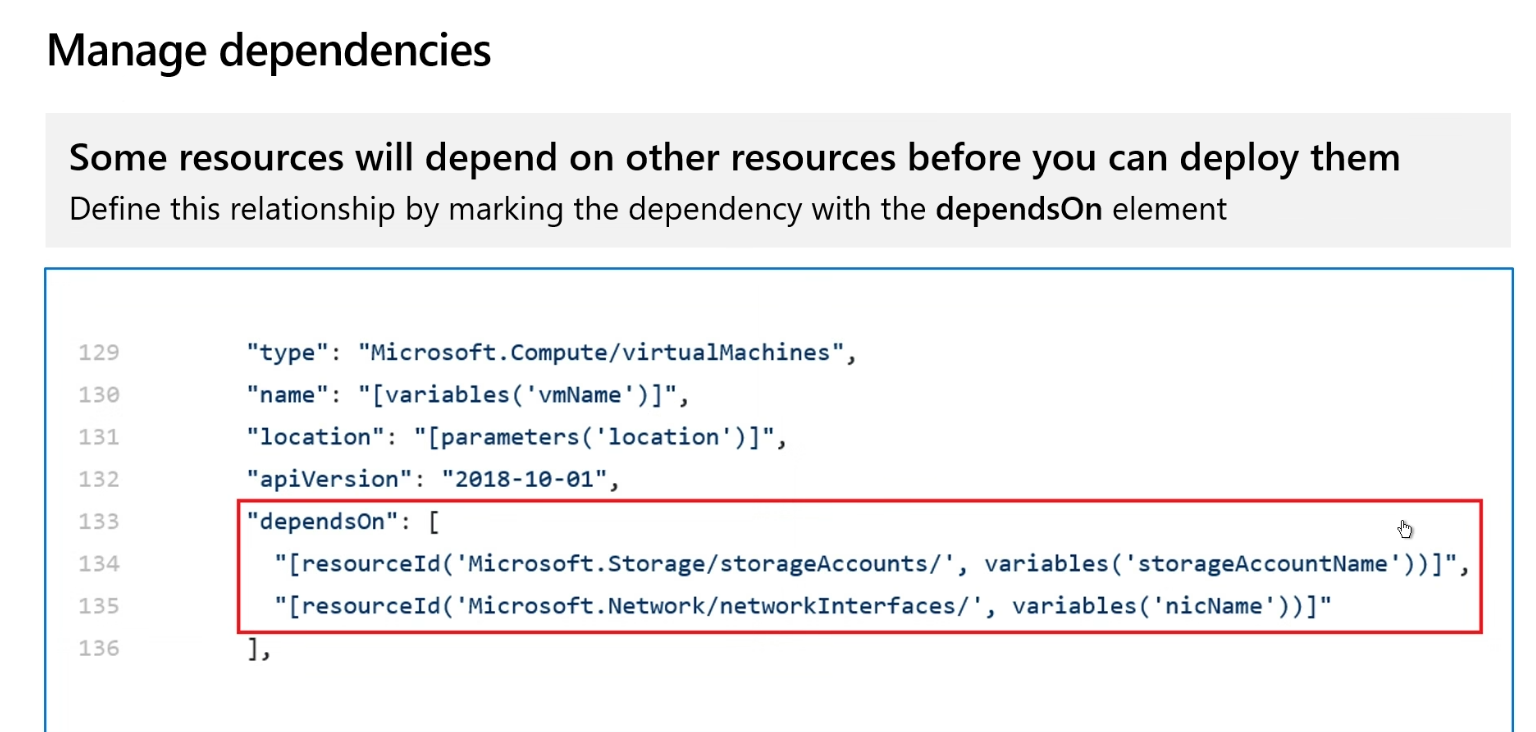


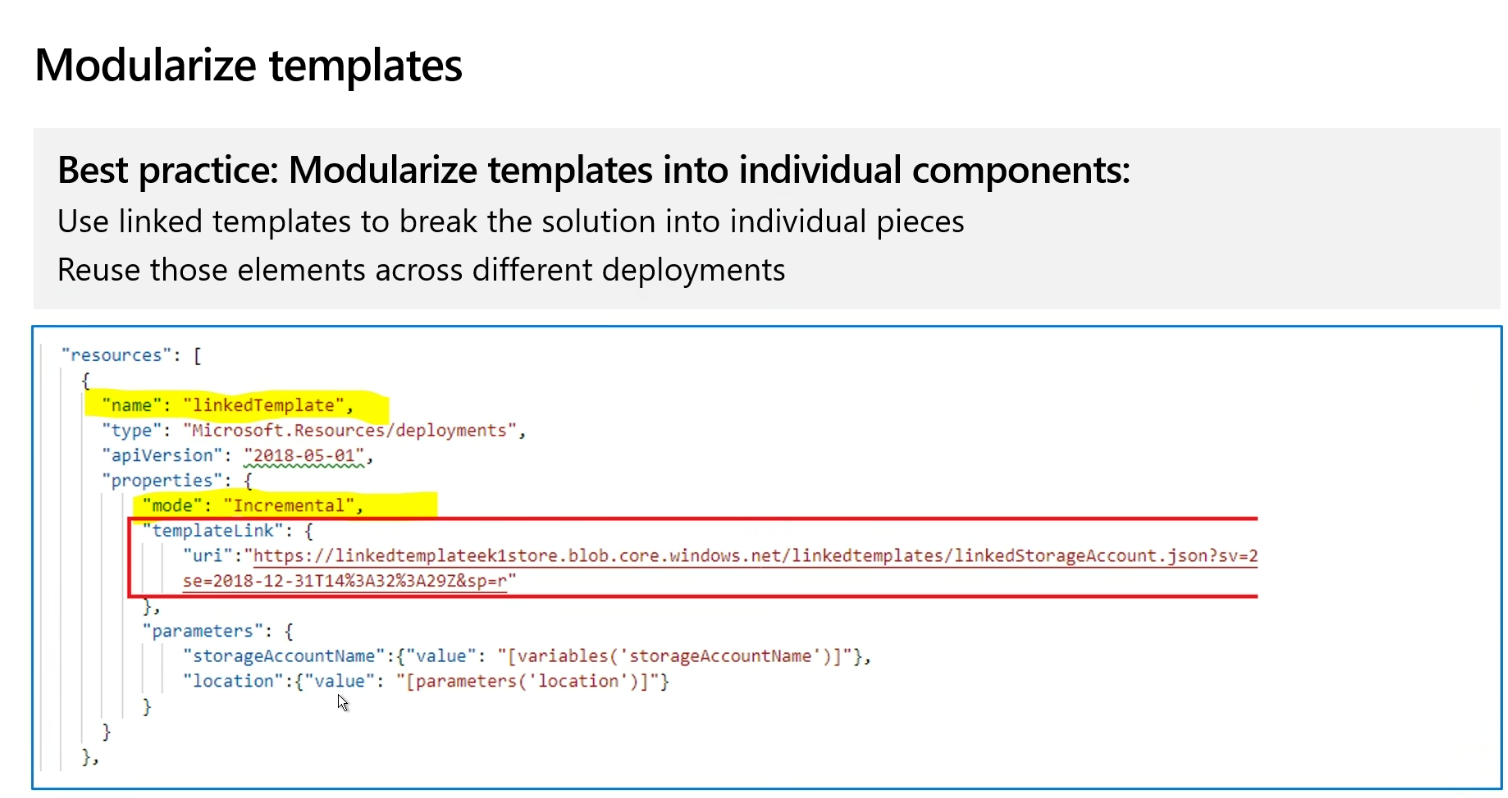


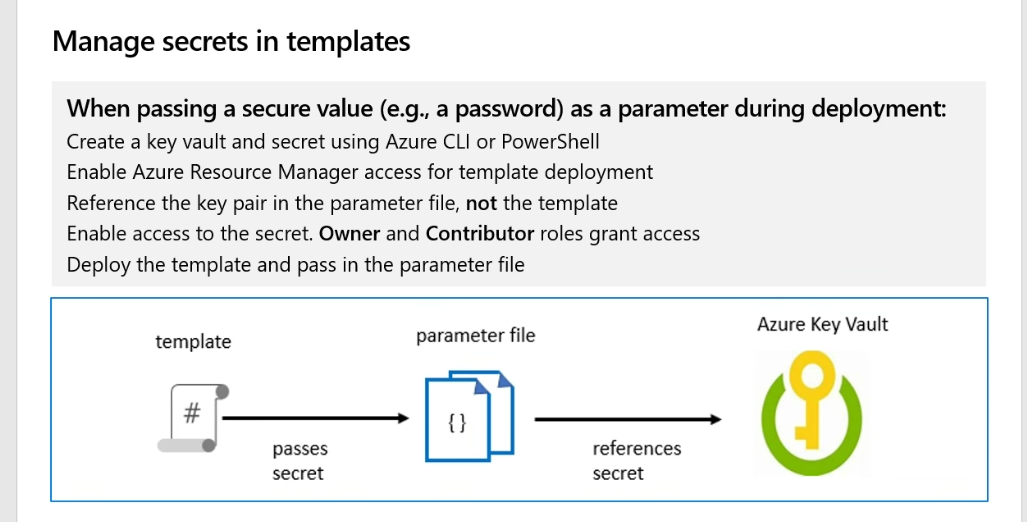


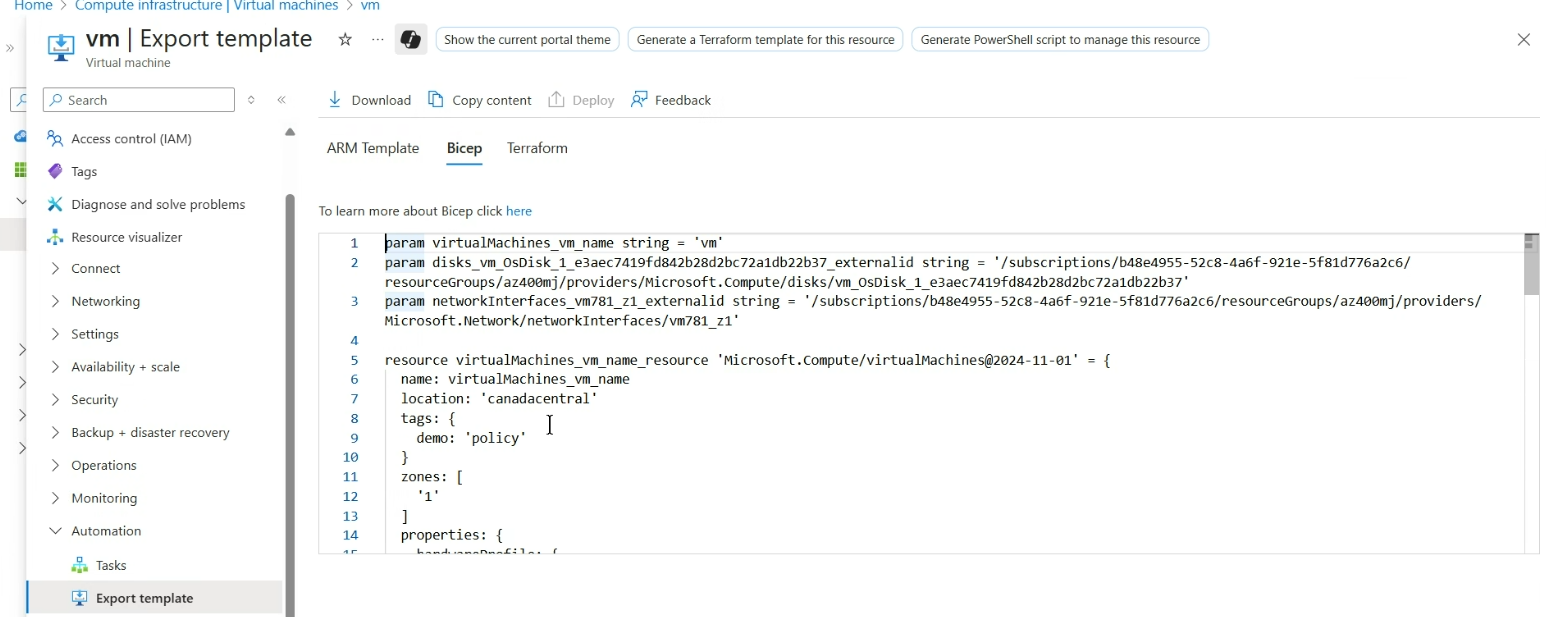
ADOCourseOrg04-eShopOnWeb-53490178-53f777fc-bede-47df-888e-f0e53b7ba51b

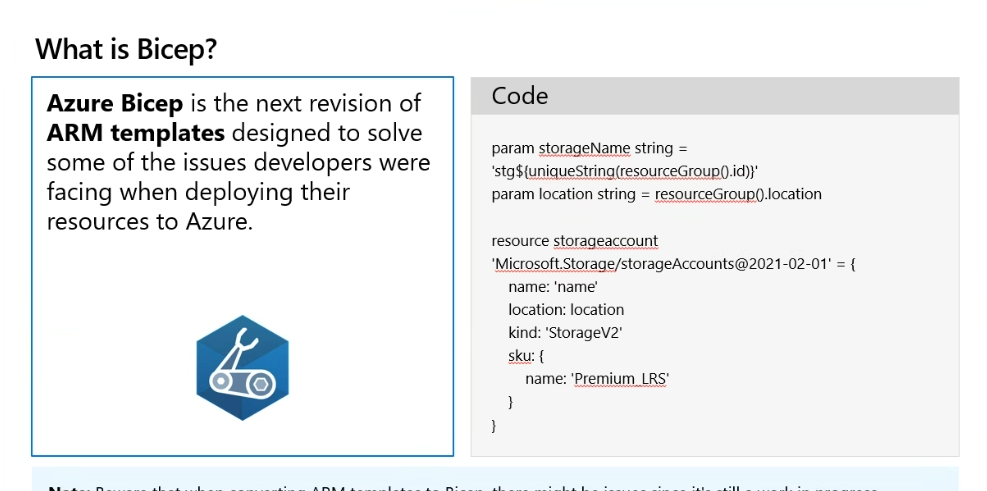
 

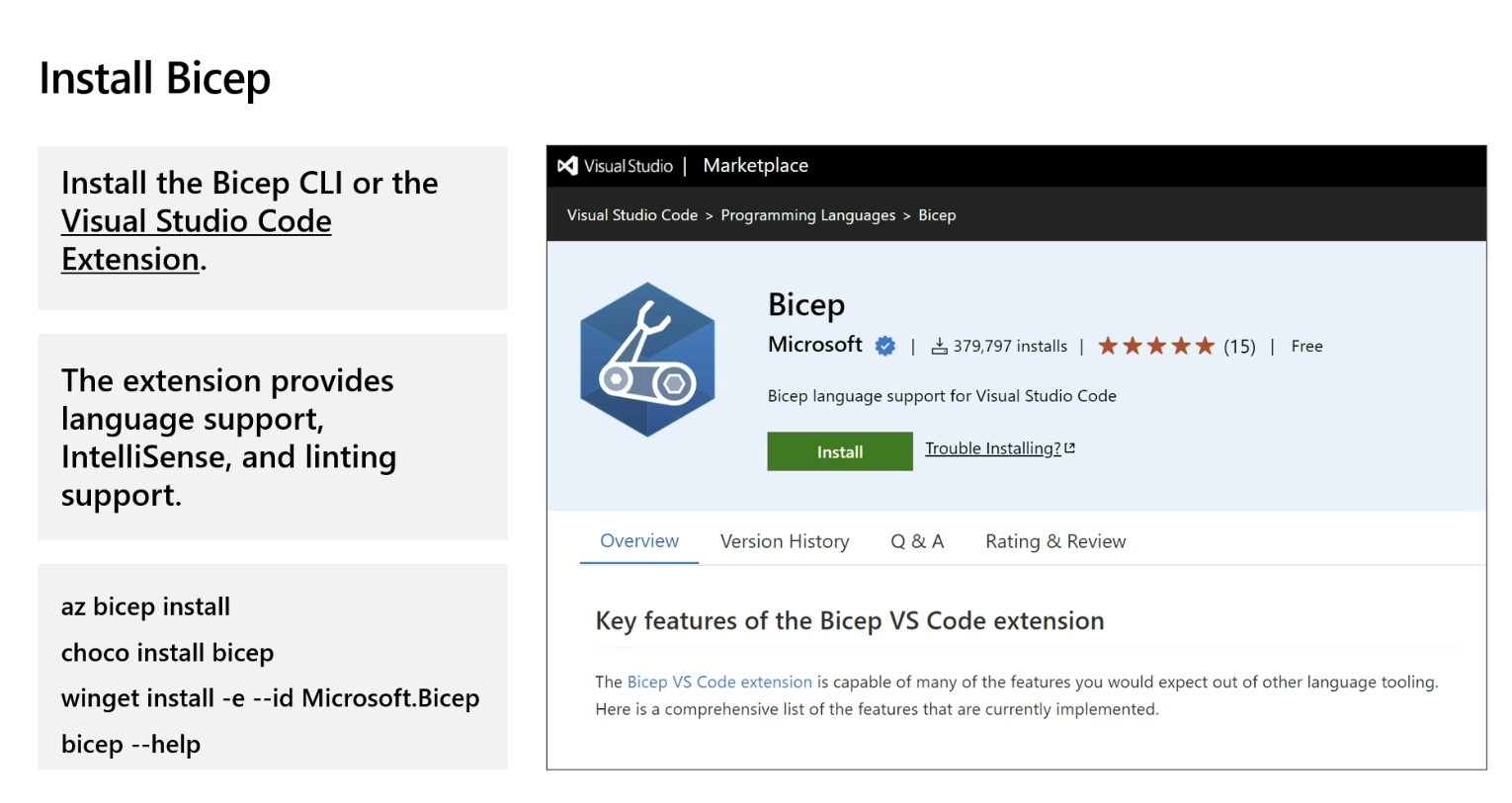
 











LOCATION='westus2'

RESOURCEGROUPNAME='az400m03l07-RG'

az group create --name $RESOURCEGROUPNAME --location $LOCATION

az group create --name $RESOURCEGROUPNAME --location $LOCATION

{

"id": "/subscriptions/6cf6ad88-2d1a-45b7-b287-d92c31bdf7de/resourceGroups/az400m03l07-RG",

"location": "westus2",

"managedBy": null,

"name": "az400m03l07-RG",

"properties": {

"provisioningState": "Succeeded"

},

"tags": null,

"type": "Microsoft.Resources/resourceGroups"

}

SERVICEPLANNAME='az400m03l07-sp1'

az appservice plan create --resource-group $RESOURCEGROUPNAME --name $SERVICEPLANNAME --sku B3

az provider register --namespace Microsoft.Web

az appservice plan create --resource-group $RESOURCEGROUPNAME --name $SERVICEPLANNAME --sku B3

{

"elasticScaleEnabled": false,

"extendedLocation": null,

"freeOfferExpirationTime": null,

"geoRegion": "West US 2",

"hostingEnvironmentProfile": null,

"hyperV": false,

"id": "/subscriptions/6cf6ad88-2d1a-45b7-b287-d92c31bdf7de/resourceGroups/az400m03l07-RG/providers/Microsoft.Web/serverfarms/az400m03l07-sp1",

"isSpot": false,

"isXenon": false,

"kind": "app",

"kubeEnvironmentProfile": null,

"location": "westus2",

"maximumElasticWorkerCount": 1,

"maximumNumberOfWorkers": 0,

"name": "az400m03l07-sp1",

"numberOfSites": 0,

"numberOfWorkers": 1,

"perSiteScaling": false,

"provisioningState": "Succeeded",

"reserved": false,

"resourceGroup": "az400m03l07-RG",

"sku": {

"capabilities": null,

"capacity": 1,

"family": "B",

"locations": null,

"name": "B3",

"size": "B3",

"skuCapacity": null,

"tier": "Basic"

},

"spotExpirationTime": null,

"status": "Ready",

"subscription": "6cf6ad88-2d1a-45b7-b287-d92c31bdf7de",

"tags": null,

"targetWorkerCount": 0,

"targetWorkerSizeId": 0,

"type": "Microsoft.Web/serverfarms",

"workerTierName": null,

"zoneRedundant": false

}

WEBAPPNAME=eshoponWebYAML53491266

az webapp create --resource-group $RESOURCEGROUPNAME --plan $SERVICEPLANNAME --name $WEBAPPNAME

WEBAPPNAME=eshoponWebYAML53491266

az webapp create --resource-group $RESOURCEGROUPNAME --plan $SERVICEPLANNAME --name $WEBAPPNAME

{

"availabilityState": "Normal",

"clientAffinityEnabled": true,

"clientCertEnabled": false,

"clientCertExclusionPaths": null,

"clientCertMode": "Required",

"cloningInfo": null,

"containerSize": 0,

"customDomainVerificationId": "1486B15F7393C688B13E3D4705E7AB78E953721B2BCAE09F6925F8BAEC67E76B",

"dailyMemoryTimeQuota": 0,

"daprConfig": null,

"defaultHostName": "eshoponwebyaml53491266.azurewebsites.net",

"enabled": true,

"enabledHostNames": [

"eshoponwebyaml53491266.azurewebsites.net",

"eshoponwebyaml53491266.scm.azurewebsites.net"

],

"endToEndEncryptionEnabled": false,

"extendedLocation": null,

"ftpPublishingUrl": "ftps://waws-prod-mwh-123.ftp.azurewebsites.windows.net/site/wwwroot",

"hostNameSslStates": [

{

"certificateResourceId": null,

"hostType": "Standard",

"ipBasedSslResult": null,

"ipBasedSslState": "NotConfigured",

"name": "eshoponwebyaml53491266.azurewebsites.net",

"sslState": "Disabled",

"thumbprint": null,

"toUpdate": null,

"toUpdateIpBasedSsl": null,

"virtualIPv6": null,

"virtualIp": null

},

{

"certificateResourceId": null,

"hostType": "Repository",

"ipBasedSslResult": null,

"ipBasedSslState": "NotConfigured",

"name": "eshoponwebyaml53491266.scm.azurewebsites.net",

"sslState": "Disabled",

"thumbprint": null,

"toUpdate": null,

"toUpdateIpBasedSsl": null,

"virtualIPv6": null,

"virtualIp": null

}

],

"hostNames": [

"eshoponwebyaml53491266.azurewebsites.net"

],

"hostNamesDisabled": false,

"hostingEnvironmentProfile": null,

"httpsOnly": false,

"hyperV": false,

"id": "/subscriptions/6cf6ad88-2d1a-45b7-b287-d92c31bdf7de/resourceGroups/az400m03l07-RG/providers/Microsoft.Web/sites/eshoponWebYAML53491266",

"identity": null,

"inProgressOperationId": null,

"isDefaultContainer": null,

"isXenon": false,

"keyVaultReferenceIdentity": "SystemAssigned",

"kind": "app",

"lastModifiedTimeUtc": "2025-08-02T09:18:44.466666",

"location": "West US 2",

"managedEnvironmentId": null,

"maxNumberOfWorkers": null,

"name": "eshoponWebYAML53491266",

"outboundIpAddresses": "4.246.19.107,4.246.16.247,4.246.18.166,4.246.18.167,4.246.19.114,4.246.19.115,4.246.19.97,4.246.19.98,4.246.19.99,4.246.19.108,4.246.19.109,4.246.19.110,4.246.19.120,4.246.19.121,4.246.19.123,4.246.18.168,4.246.18.169,4.246.18.171,20.115.232.9",

"possibleOutboundIpAddresses": "4.246.19.107,4.246.16.247,4.246.18.166,4.246.18.167,4.246.19.114,4.246.19.115,4.246.19.97,4.246.19.98,4.246.19.99,4.246.19.108,4.246.19.109,4.246.19.110,4.246.19.120,4.246.19.121,4.246.19.123,4.246.18.168,4.246.18.169,4.246.18.171,52.156.147.159,4.246.19.124,4.246.16.76,4.246.16.77,4.246.16.78,4.246.16.79,4.246.18.149,4.246.18.150,4.246.18.151,4.246.17.51,4.246.18.176,4.246.18.177,4.246.18.178,4.246.18.179,4.246.18.184,4.246.18.185,4.246.18.186,4.246.18.187,4.246.19.96,4.246.18.172,4.246.18.173,4.246.18.174,4.246.17.117,4.246.18.175,4.246.19.176,20.115.232.9",

"publicNetworkAccess": null,

"redundancyMode": "None",

"repositorySiteName": "eshoponWebYAML53491266",

"reserved": false,

"resourceConfig": null,

"resourceGroup": "az400m03l07-RG",

"scmSiteAlsoStopped": false,

"serverFarmId": "/subscriptions/6cf6ad88-2d1a-45b7-b287-d92c31bdf7de/resourceGroups/az400m03l07-RG/providers/Microsoft.Web/serverfarms/az400m03l07-sp1",

"siteConfig": {

"acrUseManagedIdentityCreds": false,

"acrUserManagedIdentityId": null,

"alwaysOn": false,

"antivirusScanEnabled": null,

"apiDefinition": null,

"apiManagementConfig": null,

"appCommandLine": null,

"appSettings": null,

"autoHealEnabled": null,

"autoHealRules": null,

"autoSwapSlotName": null,

"azureMonitorLogCategories": null,

"azureStorageAccounts": null,

"clusteringEnabled": false,

"connectionStrings": null,

"cors": null,

"customAppPoolIdentityAdminState": null,

"customAppPoolIdentityTenantState": null,

"defaultDocuments": null,

"detailedErrorLoggingEnabled": null,

"documentRoot": null,

"elasticWebAppScaleLimit": 0,

"experiments": null,

"fileChangeAuditEnabled": null,

"ftpsState": null,

"functionAppScaleLimit": null,

"functionsRuntimeScaleMonitoringEnabled": null,

"handlerMappings": null,

"healthCheckPath": null,

"http20Enabled": false,

"http20ProxyFlag": null,

"httpLoggingEnabled": null,

"ipSecurityRestrictions": [

{

"action": "Allow",

"description": "Allow all access",

"headers": null,

"ipAddress": "Any",

"name": "Allow all",

"priority": 2147483647,

"subnetMask": null,

"subnetTrafficTag": null,

"tag": null,

"vnetSubnetResourceId": null,

"vnetTrafficTag": null

}

],

"ipSecurityRestrictionsDefaultAction": null,

"javaContainer": null,

"javaContainerVersion": null,

"javaVersion": null,

"keyVaultReferenceIdentity": null,

"limits": null,

"linuxFxVersion": "",

"loadBalancing": null,

"localMySqlEnabled": null,

"logsDirectorySizeLimit": null,

"machineKey": null,

"managedPipelineMode": null,

"managedServiceIdentityId": null,

"metadata": null,

"minTlsCipherSuite": null,

"minTlsVersion": null,

"minimumElasticInstanceCount": 0,

"netFrameworkVersion": null,

"nodeVersion": null,

"numberOfWorkers": 1,

"phpVersion": null,

"powerShellVersion": null,

"preWarmedInstanceCount": null,

"publicNetworkAccess": null,

"publishingPassword": null,

"publishingUsername": null,

"push": null,

"pythonVersion": null,

"remoteDebuggingEnabled": null,

"remoteDebuggingVersion": null,

"requestTracingEnabled": null,

"requestTracingExpirationTime": null,

"routingRules": null,

"runtimeADUser": null,

"runtimeADUserPassword": null,

"sandboxType": null,

"scmIpSecurityRestrictions": [

{

"action": "Allow",

"description": "Allow all access",

"headers": null,

"ipAddress": "Any",

"name": "Allow all",

"priority": 2147483647,

"subnetMask": null,

"subnetTrafficTag": null,

"tag": null,

"vnetSubnetResourceId": null,

"vnetTrafficTag": null

}

],

"scmIpSecurityRestrictionsDefaultAction": null,

"scmIpSecurityRestrictionsUseMain": null,

"scmMinTlsCipherSuite": null,

"scmMinTlsVersion": null,

"scmSupportedTlsCipherSuites": null,

"scmType": null,

"sitePort": null,

"sitePrivateLinkHostEnabled": null,

"storageType": null,

"supportedTlsCipherSuites": null,

"tracingOptions": null,

"use32BitWorkerProcess": null,

"virtualApplications": null,

"vnetName": null,

"vnetPrivatePortsCount": null,

"vnetRouteAllEnabled": null,

"webSocketsEnabled": null,

"websiteTimeZone": null,

"winAuthAdminState": null,

"winAuthTenantState": null,

"windowsConfiguredStacks": null,

"windowsFxVersion": null,

"xManagedServiceIdentityId": null

},

"slotSwapStatus": null,

"state": "Running",

"storageAccountRequired": false,

"suspendedTill": null,

"tags": null,

"targetSwapSlot": null,

"trafficManagerHostNames": null,

"type": "Microsoft.Web/sites",

"usageState": "Normal",

"virtualNetworkSubnetId": null,

"vnetContentShareEnabled": false,

"vnetImagePullEnabled": false,

"vnetRouteAllEnabled": false,

"workloadProfileName": null

}

#NAME THE PIPELINE SAME AS FILE (WITHOUT ".yml")

# trigger:

# - main

resources:

repositories:

- repository: self

trigger: none

stages:

- stage: Build

displayName: Build .Net Core Solution

jobs:

- job: Build

pool:

vmImage: ubuntu-latest

steps:

- task: DotNetCoreCLI@2

displayName: Restore

inputs:

command: 'restore'

projects: '\*\*/\*.sln'

feedsToUse: 'select'

- task: DotNetCoreCLI@2

displayName: Build

inputs:

command: 'build'

projects: '\*\*/\*.sln'

- task: DotNetCoreCLI@2

displayName: Test

inputs:

command: 'test'

projects: 'tests/UnitTests/\*.csproj'

- task: DotNetCoreCLI@2

displayName: Publish

inputs:

command: 'publish'

publishWebProjects: true

arguments: '-o $(Build.ArtifactStagingDirectory)'

- task: PublishBuildArtifacts@1

displayName: Publish Artifacts ADO - Website

inputs:

pathToPublish: '$(Build.ArtifactStagingDirectory)'

artifactName: Website

- task: PublishBuildArtifacts@1

displayName: Publish Artifacts ADO - Bicep

inputs:

PathtoPublish: '$(Build.SourcesDirectory)/infra/webapp.bicep'

ArtifactName: 'Bicep'

publishLocation: 'Container'

- stage: Deploy

displayName: Deploy to an Azure Web App

jobs:

- job: Deploy

pool:

vmImage: 'windows-latest'

steps:

- task: DownloadBuildArtifacts@0

inputs:

buildType: 'current'

downloadType: 'single'

artifactName: 'Website'

downloadPath: '$(Build.ArtifactStagingDirectory)'

- task: AzureRmWebAppDeployment@4

inputs:

ConnectionType: 'AzureRM'

azureSubscription: 'AZURE SUBSCRIPTION HERE (b999999abc-1234-987a-a1e0-27fb2ea7f9f4)'

appType: 'webApp'

WebAppName: 'eshoponWebYAML369825031'

packageForLinux: '$(Build.ArtifactStagingDirectory)/\*\*/Web.zip'

AppSettings: '-UseOnlyInMemoryDatabase true -ASPNETCORE\_ENVIRONMENT Development'

<https://certyiq.com/>

<https://learn.microsoft.com/en-us/users/milandas-7636/?username=MilanDas-7636&section=activity>

REGION='westus2'

RESOURCEGROUPNAME='az400m03l08-RG'

az group create -n $RESOURCEGROUPNAME -l $REGION

SERVICEPLANNAME='az400m03l08-sp1'

az appservice plan create -g $RESOURCEGROUPNAME -n $SERVICEPLANNAME --sku S1

SUFFIX=53492225

az webapp create -g $RESOURCEGROUPNAME -p $SERVICEPLANNAME -n RGATES$SUFFIX-DevTest

az webapp create -g $RESOURCEGROUPNAME -p $SERVICEPLANNAME -n RGATES$SUFFIX-Prod

{

"availabilityState": "Normal",

"clientAffinityEnabled": true,

"clientCertEnabled": false,

"clientCertExclusionPaths": null,

"clientCertMode": "Required",

"cloningInfo": null,

"containerSize": 0,

"customDomainVerificationId": "7157CE0C3926C28D682DE04F517F46C6A6AA970BFA027E5BBDF94B2AF906D686",

"dailyMemoryTimeQuota": 0,

"daprConfig": null,

"defaultHostName": "rgates53492225-devtest.azurewebsites.net",

"enabled": true,

"enabledHostNames": [

"rgates53492225-devtest.azurewebsites.net",

"rgates53492225-devtest.scm.azurewebsites.net"

],

"endToEndEncryptionEnabled": false,

"extendedLocation": null,

"ftpPublishingUrl": "ftps://waws-prod-mwh-051.ftp.azurewebsites.windows.net/site/wwwroot",

"hostNameSslStates": [

{

"certificateResourceId": null,

"hostType": "Standard",

"ipBasedSslResult": null,

"ipBasedSslState": "NotConfigured",

"name": "rgates53492225-devtest.azurewebsites.net",

"sslState": "Disabled",

"thumbprint": null,

"toUpdate": null,

"toUpdateIpBasedSsl": null,

"virtualIPv6": null,

"virtualIp": null

},

{

"certificateResourceId": null,

"hostType": "Repository",

"ipBasedSslResult": null,

"ipBasedSslState": "NotConfigured",

"name": "rgates53492225-devtest.scm.azurewebsites.net",

"sslState": "Disabled",

"thumbprint": null,

"toUpdate": null,

"toUpdateIpBasedSsl": null,

"virtualIPv6": null,

"virtualIp": null

}

],

"hostNames": [

"rgates53492225-devtest.azurewebsites.net"

],

"hostNamesDisabled": false,

"hostingEnvironmentProfile": null,

"httpsOnly": false,

"hyperV": false,

"id": "/subscriptions/6dac8799-969e-4de4-9758-d6e409f49e71/resourceGroups/az400m03l08-RG/providers/Microsoft.Web/sites/RGATES53492225-DevTest",

"identity": null,

"inProgressOperationId": null,

"isDefaultContainer": null,

"isXenon": false,

"keyVaultReferenceIdentity": "SystemAssigned",

"kind": "app",

"lastModifiedTimeUtc": "2025-08-02T11:49:17.410000",

"location": "West US 2",

"managedEnvironmentId": null,

"maxNumberOfWorkers": null,

"name": "RGATES53492225-DevTest",

"outboundIpAddresses": "20.190.0.233,20.190.1.8,20.190.1.21,20.190.1.32,20.190.1.42,20.190.1.61,40.64.128.225",

"possibleOutboundIpAddresses": "20.190.0.233,20.190.1.8,20.190.1.21,20.190.1.32,20.190.1.42,20.190.1.61,4.154.249.253,20.190.1.72,20.190.0.33,20.190.1.162,20.190.1.177,20.190.1.187,20.190.1.191,52.149.26.109,52.149.26.223,52.149.26.248,52.149.27.21,52.149.27.137,52.149.28.241,51.143.61.29,52.137.93.170,52.149.29.95,52.149.29.238,52.149.30.76,52.149.30.96,20.190.1.217,20.190.2.43,20.190.2.66,52.143.80.96,52.156.145.74,52.156.145.106,40.64.128.225",

"publicNetworkAccess": null,

"redundancyMode": "None",

"repositorySiteName": "RGATES53492225-DevTest",

"reserved": false,

"resourceConfig": null,

"resourceGroup": "az400m03l08-RG",

"scmSiteAlsoStopped": false,

"serverFarmId": "/subscriptions/6dac8799-969e-4de4-9758-d6e409f49e71/resourceGroups/az400m03l08-RG/providers/Microsoft.Web/serverfarms/az400m03l08-sp1",

"siteConfig": {

"acrUseManagedIdentityCreds": false,

"acrUserManagedIdentityId": null,

"alwaysOn": false,

"antivirusScanEnabled": null,

"apiDefinition": null,

"apiManagementConfig": null,

"appCommandLine": null,

"appSettings": null,

"autoHealEnabled": null,

"autoHealRules": null,

"autoSwapSlotName": null,

"azureMonitorLogCategories": null,

"azureStorageAccounts": null,

"clusteringEnabled": false,

"connectionStrings": null,

"cors": null,

"customAppPoolIdentityAdminState": null,

"customAppPoolIdentityTenantState": null,

"defaultDocuments": null,

"detailedErrorLoggingEnabled": null,

"documentRoot": null,

"elasticWebAppScaleLimit": 0,

"experiments": null,

"fileChangeAuditEnabled": null,

"ftpsState": null,

"functionAppScaleLimit": null,

"functionsRuntimeScaleMonitoringEnabled": null,

"handlerMappings": null,

"healthCheckPath": null,

"http20Enabled": false,

"http20ProxyFlag": null,

"httpLoggingEnabled": null,

"ipSecurityRestrictions": [

{

"action": "Allow",

"description": "Allow all access",

"headers": null,

"ipAddress": "Any",

"name": "Allow all",

"priority": 2147483647,

"subnetMask": null,

"subnetTrafficTag": null,

"tag": null,

"vnetSubnetResourceId": null,

"vnetTrafficTag": null

}

],

"ipSecurityRestrictionsDefaultAction": null,

"javaContainer": null,

"javaContainerVersion": null,

"javaVersion": null,

"keyVaultReferenceIdentity": null,

"limits": null,

"linuxFxVersion": "",

"loadBalancing": null,

"localMySqlEnabled": null,

"logsDirectorySizeLimit": null,

"machineKey": null,

"managedPipelineMode": null,

"managedServiceIdentityId": null,

"metadata": null,

"minTlsCipherSuite": null,

"minTlsVersion": null,

"minimumElasticInstanceCount": 0,

"netFrameworkVersion": null,

"nodeVersion": null,

"numberOfWorkers": 1,

"phpVersion": null,

"powerShellVersion": null,

"preWarmedInstanceCount": null,

"publicNetworkAccess": null,

"publishingPassword": null,

"publishingUsername": null,

"push": null,

"pythonVersion": null,

"remoteDebuggingEnabled": null,

"remoteDebuggingVersion": null,

"requestTracingEnabled": null,

"requestTracingExpirationTime": null,

"routingRules": null,

"runtimeADUser": null,

"runtimeADUserPassword": null,

"sandboxType": null,

"scmIpSecurityRestrictions": [

{

"action": "Allow",

"description": "Allow all access",

"headers": null,

"ipAddress": "Any",

"name": "Allow all",

"priority": 2147483647,

"subnetMask": null,

"subnetTrafficTag": null,

"tag": null,

"vnetSubnetResourceId": null,

"vnetTrafficTag": null

}

],

"scmIpSecurityRestrictionsDefaultAction": null,

"scmIpSecurityRestrictionsUseMain": null,

"scmMinTlsCipherSuite": null,

"scmMinTlsVersion": null,

"scmSupportedTlsCipherSuites": null,

"scmType": null,

"sitePort": null,

"sitePrivateLinkHostEnabled": null,

"storageType": null,

"supportedTlsCipherSuites": null,

"tracingOptions": null,

"use32BitWorkerProcess": null,

"virtualApplications": null,

"vnetName": null,

"vnetPrivatePortsCount": null,

"vnetRouteAllEnabled": null,

"webSocketsEnabled": null,

"websiteTimeZone": null,

"winAuthAdminState": null,

"winAuthTenantState": null,

"windowsConfiguredStacks": null,

"windowsFxVersion": null,

"xManagedServiceIdentityId": null

},

"slotSwapStatus": null,

"state": "Running",

"storageAccountRequired": false,

"suspendedTill": null,

"tags": null,

"targetSwapSlot": null,

"trafficManagerHostNames": null,

"type": "Microsoft.Web/sites",

"usageState": "Normal",

"virtualNetworkSubnetId": null,

"vnetContentShareEnabled": false,

"vnetImagePullEnabled": false,

"vnetRouteAllEnabled": false,

"workloadProfileName": null

}

* **Deployment Strategies:**
  + **Run Once:** The simplest strategy where all lifecycle hooks (pre-deploy, deploy, post-deploy) are executed once. This is the default for many deployments.
  + **Rolling Deployment:** Gradually replaces instances of the old version with new ones on a fixed set of virtual machines, waiting for each set to complete before moving to the next.
  + **Canary Deployment:** An advanced strategy for risk mitigation, where changes are initially rolled out to a small subset of servers or users, and then gradually expanded as confidence in the new version grows.
  + **Blue-Green Deployment:** Involves running two identical production environments ("blue" for the current version and "green" for the new version). Traffic is switched to the "green" environment once it's validated, allowing for quick rollback if needed.

| **Implementation** | **Description** |
| --- | --- |
| [strategy: runOnce](https://learn.microsoft.com/en-us/azure/devops/pipelines/yaml-schema/jobs-deployment-strategy?view=azure-pipelines#strategyrunonce) | Run once deployment strategy. |
| [strategy: rolling](https://learn.microsoft.com/en-us/azure/devops/pipelines/yaml-schema/jobs-deployment-strategy?view=azure-pipelines#strategyrolling) | Rolling deployment strategy. |
| [strategy: canary](https://learn.microsoft.com/en-us/azure/devops/pipelines/yaml-schema/jobs-deployment-strategy?view=azure-pipelines#strategycanary) | Canary deployment strategy. |

Blue-green deployment is a software release strategy in Azure DevOps that aims to minimize downtime and risk during application updates by running two identical production environments, often referred to as "Blue" and "Green."

How it works in Azure DevOps:

* **Maintain two identical environments:**
  + **Blue environment:** This is the currently live production environment serving user traffic with the existing version of the application.
  + **Green environment:** This is an identical, but initially idle, environment where the new version of the application is deployed and thoroughly tested.
* **Deployment of the new version:**

Azure DevOps pipelines are used to automate the deployment of the new application version to the Green environment. This includes building the application, running automated tests, and deploying it to the designated resources (e.g., Azure App Service Deployment Slots, Azure Container Apps revisions, or separate virtual machines/clusters).

* **Testing and Validation:**

Once deployed to the Green environment, extensive testing is performed to ensure the new version functions as expected, without impacting the live Blue environment. This can involve automated tests within the pipeline, manual testing, or even A/B testing with a small subset of users.

* **Traffic Switch:**

When the new version in the Green environment is deemed stable and ready, the traffic is switched from the Blue environment to the Green environment. This can be achieved using various Azure services like:

* + **Azure App Service Deployment Slots:** Swapping the production slot with the staging slot where the new version resides.
  + **Azure Front Door or Azure Traffic Manager:** Configuring routing rules to direct traffic to the Green environment.
  + **Azure Container Apps:** Updating the traffic weight to direct more traffic to the new revision.
* **Monitoring and Rollback:**

After the switch, the Green environment becomes the live production environment. The old Blue environment is kept as a fallback. If any issues arise with the new version, traffic can be quickly redirected back to the stable Blue environment, minimizing disruption.

* **Cleanup (Optional):**

Once the new version is confirmed stable and the rollback window has passed, the old Blue environment can be decommissioned or prepared for future deployments as a new "Green" environment.

Benefits in Azure DevOps:

* **Zero Downtime:**

Users experience continuous availability during updates.

* **Reduced Risk:**

Issues in the new version are isolated to the Green environment, preventing impact on live users.

* **Fast Rollback:**

Quick recovery from problematic deployments by switching back to the old version.

* **Automated with Pipelines:**

Azure DevOps pipelines facilitate the automation of deployment, testing, and traffic switching, streamlining the process.

<https://bogdanbujdea.dev/blue-green-deployments-in-azure-devops-yaml-pipelines> -->Blue green deployment

<https://dev.azure.com/bujdea/_git/AzureDevopsYamlPipeline>

**Overview**

As you may be aware, a release pipeline specifies the end-to-end release process for an application to be deployed across a range of environments. Deployments to each environment are fully automated by using jobs and tasks. Ideally, you do not want new updates to the applications to be exposed to all the users at the same time. It is a best practice to expose updates in a phased manner i.e. expose to a subset of users, monitor their usage and expose to other users based on the experience the initial set of users had.

Approvals and gates enable you to take control over the start and completion of the deployments in a release. With approvals, you can wait for users to manually approve or reject deployments. Using release gates, you can specify application health criteria that must be met before release is promoted to the next environment. Prior to or after any environment deployment, all the specified gates are automatically evaluated until they all pass or until they reach your defined timeout period and fail.

Gates can be added to an environment in the release definition from the pre-deployment conditions or the post-deployment conditions panel. Multiple gates can be added to the environment conditions to ensure all the inputs are successful for the release.

As an example:

**Pre-deployment gates** ensures there are no active issues in the work item or problem management system before deploying a build to an environment.

**Post-deployment gates** ensures there are no incidents from the monitoring or incident management system for the app after it’s been deployed, before promoting the release to the next environment.

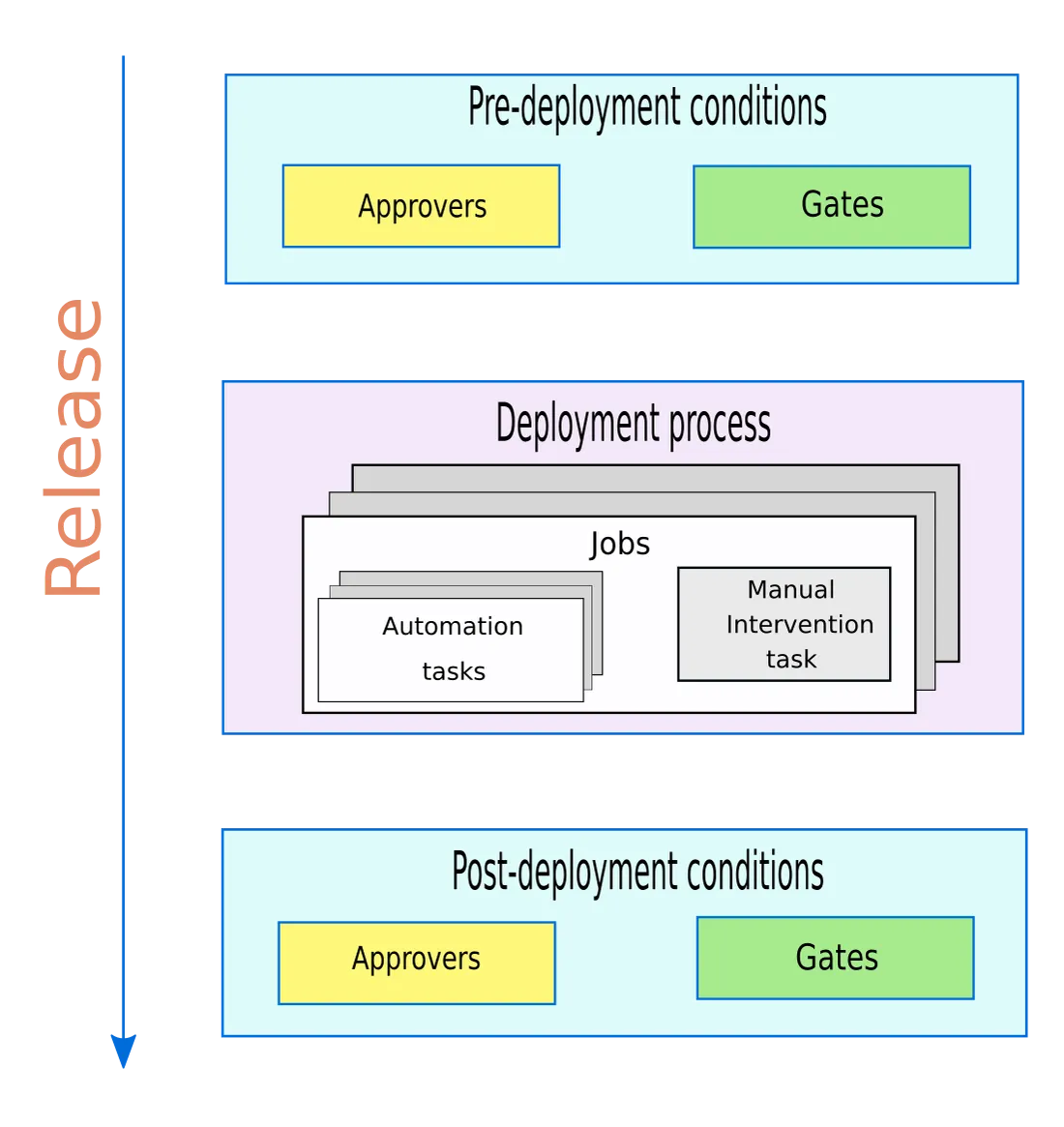
4 types of gates are included by default for every account.

1. **Invoke Azure function:** Trigger execution of an Azure function and ensures a successful completion. For more details, see [Azure function task](https://docs.microsoft.com/en-us/vsts/build-release/tasks/utility/azure-function)
2. **Query Azure monitor alerts:** Observe the configured Azure monitor alert rules for active alerts. For more details, see [Azure monitor task](https://docs.microsoft.com/en-us/vsts/build-release/tasks/utility/azure-monitor)
3. **Invoke REST API:** Make a call to a REST API and continue if it returns a successful response. For more details, see [HTTP REST API task](https://docs.microsoft.com/en-us/vsts/build-release/tasks/utility/http-rest-api)
4. **Query Workitems:** Ensures the number of matching work items returned from a query is within a threshold. For more details, see [Query Workitems task](https://docs.microsoft.com/en-us/vsts/build-release/tasks/utility/work-item-query)

<https://www.azuredevopslabs.com/labs/vstsextend/releasegates/#whats-covered-in-this-lab>

What are gates?

Gates are like the bouncers at a club — they check to make sure you meet certain criteria before you’re allowed in. In Azure Pipelines, gates are a type of pre-deployment approval check that allows you to define criteria that must be met before a deployment can proceed.

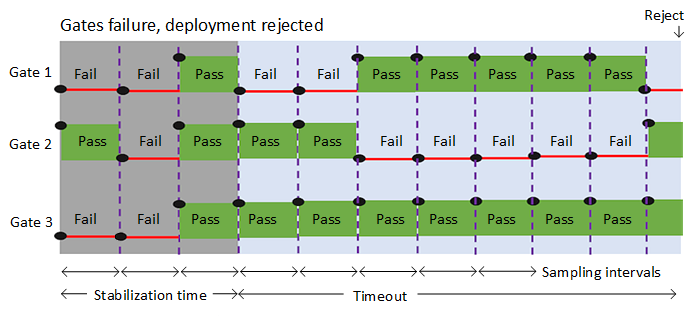


How do gates work?

Gates work by checking whether the specified criteria have been met before a deployment can proceed. For example, you might configure a gate to check that the application’s response time is within a certain range or that the application can handle a certain number of requests per second. If the criteria have been met, the deployment proceeds. If not, the deployment is halted and an error message is displayed.

Why use gates?

Gates are a powerful tool for ensuring that your applications meet the required performance standards before they are released to production. By configuring gates to check for performance issues, you can avoid downtime and other issues caused by poorly performing applications. Additionally, gates can be used to prevent unauthorized or untested code from being released to production, which can help to ensure the security and stability of your applications.



Scenario 1: Response Time

Let’s say you’re deploying a new release of your application, and you want to make sure that the response time stays within a certain range. You can configure a gate to check the response time during a load test, and if it exceeds the specified range, the deployment is halted.

Scenario 2: Error Rate

Another scenario where gates can be useful is when you want to ensure that the error rate stays within a certain threshold. You can configure a gate to check the error rate during a load test, and if it exceeds the specified threshold, the deployment is halted.

How to Create Gates in Azure Pipelines

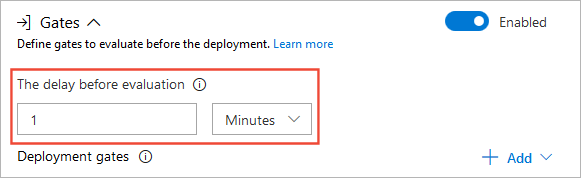
Now that we’ve covered the what and why of gates, let’s dive into how to create them in Azure Pipelines. Here are the steps you need to follow:

Step 1: Define the Criteria

First, you need to define the criteria that you want to check. This could be anything from response time to error rate to a specific test passing.

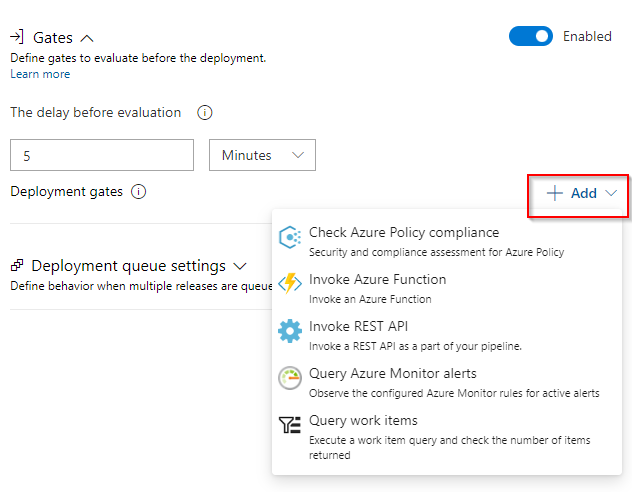
Step 2: Configure the Gate

Next, you need to configure the gate to check the criteria that you defined in step 1. This involves setting up a load test or other test that will check the criteria, and configuring the gate to check the results of that test.



Step 3: Add the Gate to the Pipeline

Finally, you need to add the gate to your pipeline. This involves adding a pre-deployment approval step to your pipeline, and configuring it to use the gate you created in step 2.



<https://medium.com/@walissonscd/gates-in-azure-pipelines-ensuring-high-performance-releases-with-pre-deployment-approval-checks-6ca15b1b4755>

<https://learn.microsoft.com/en-us/azure/devops/pipelines/release/approvals/?view=azure-devops&tabs=yaml>