Q2 - SCENARIO Macro Life, a healthcare company has recently setup the entire Network and Infrastructure on Azure. The infrastructure has different components such as Virtual N/W, Subnets, NIC, IPs, NSG etc. The IT team currently has developed PowerShell scripts to deploy each component where all the properties of each resource is set using PowerShell commands. The business has realized that the PowerShell scripts are growing over period of time and difficult to handover when new admin onboards in the IT. The IT team has now decided to move to ARM based deployment of all resources to Azure. All the passwords are stored in a Azure Service known as key Vault. The deployments need to be automated using Azure DevOps using IaC (Infrastructure as Code).

1) What are different artifacts you need to create - name of the artifacts and its purpose

2) List the tools you will to create and store the ARM templates.

3) Explain the process and steps to create automated deployment pipeline.

4) Create a sample ARM template you will use to deploy a Windows VM of any size

5) Explain how will you access the password stored in Key Vault and use it as Admin Password in the VM ARM template.

Solution2:

1) To create ARM templates deployment pipelines, we require basically two artifacts.
   a. Template.json file
   b. Parameters.json file

   Template.json file that contain details like, information related to the resource to be created. It contains different sections like

   parameters – to declare parameters,

   variables – define resource related variables – in our case VM related variables like nsgId, vnetId, subnetRef etc.

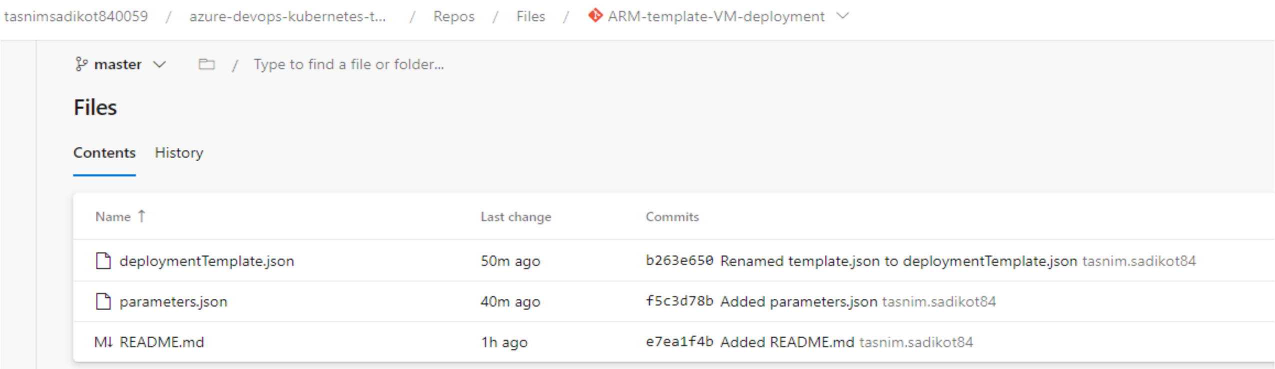
   resources – contains information for different type of resources to be created, location, dependencies related information, admin related info in creation of resources and other required information's.

   Parameters.jason file that contain details and values of each parameters required by Template.json file to create resource. This file is use to customize your resource creation based on parameters passed, like based on resource group, size of VM, NSG, subnet etc.

2) To create ARM template, **Visual Studio Code** can be used as code editor. ARM templates can be stored and accessed from any version control tool as per organization SDLC standards like **Git** or (in our case) **Azure Git** to be used by ARM deployment pipeline.

3) Steps to create automated pipeline

1. Create and check-in ARM template files described in (1) to Azure Git



tasnimsadikot840059 / azure-devops-kubernetes-t... / Repos / Files / ARM-template-VM-deployment

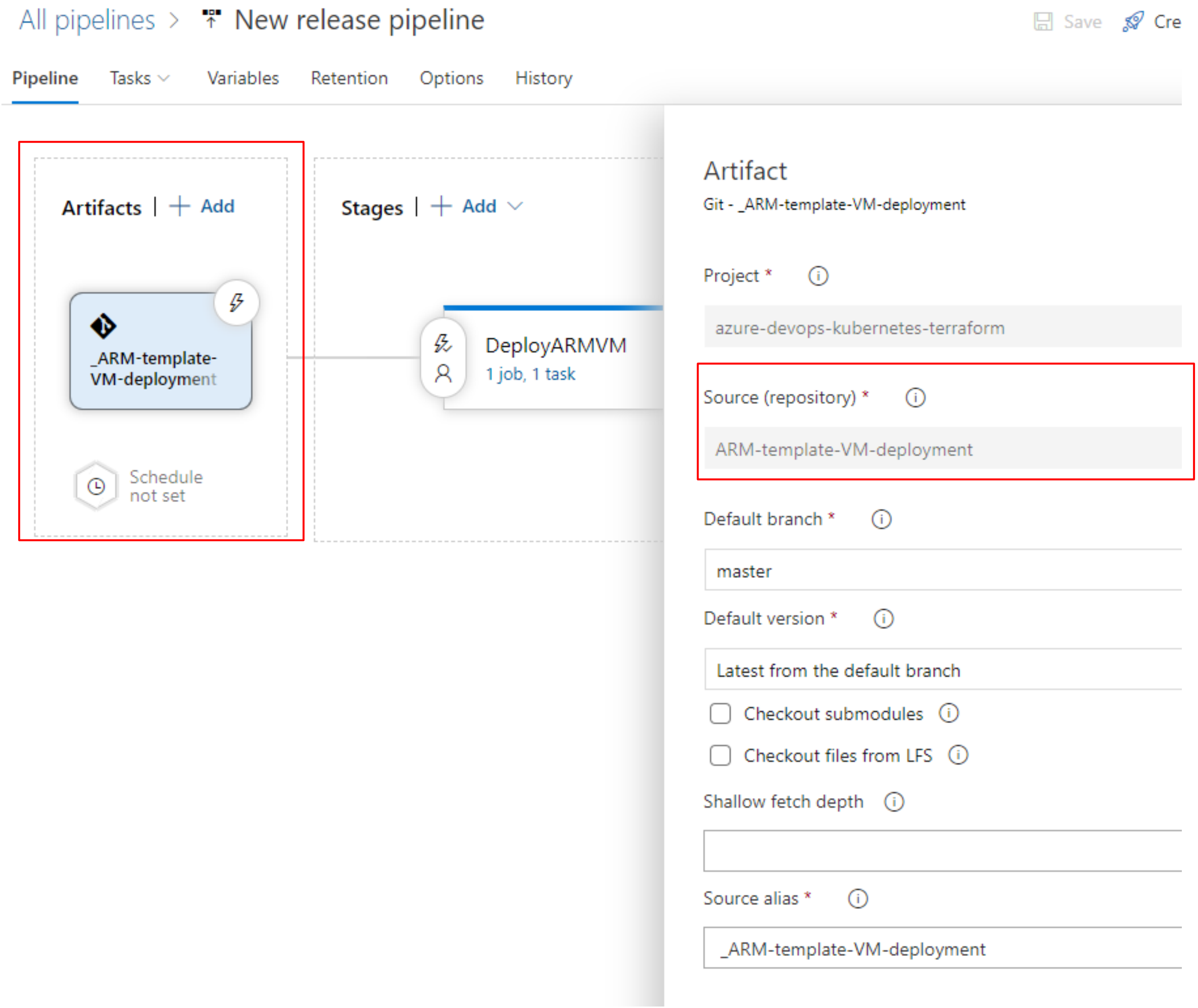
master / Type to find a file or folder...

Files

Contents History

Name ↑	Last change	Commits
deploymentTemplate.json	50m ago	b263e650 Renamed template.json to deploymentTemplate.json tasnim.sadikot84
parameters.json	40m ago	f5c3d78b Added parameters.json tasnim.sadikot84
README.md	1h ago	e7ea1f4b Added README.md tasnim.sadikot84

2. Create a blank release pipeline and add the artifacts stage configured to collect artifacts from Azure Git repo for ARM templates.



All pipelines > New release pipeline

Save Cre

Pipeline Tasks Variables Retention Options History

Artifacts | + Add

_ARM-template-VM-deployment

Schedule not set

Stages | + Add

DeployARMVM
1 job, 1 task

Artifact
Git - _ARM-template-VM-deployment

Project *: azure-devops-kubernetes-terraform

Source (repository) *: ARM-template-VM-deployment

Default branch *: master

Default version *: Latest from the default branch

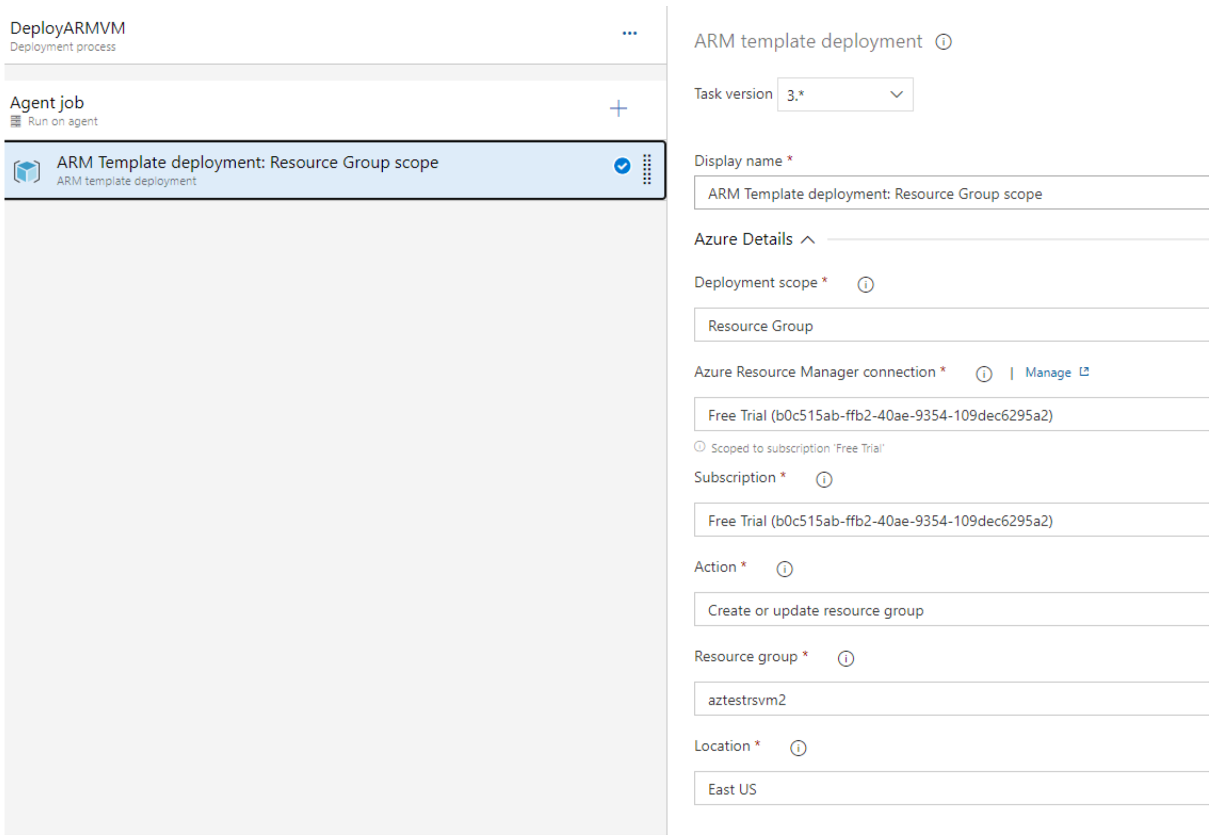
□ Checkout submodules

□ Checkout files from LFS

Shallow fetch depth:

Source alias *: _ARM-template-VM-deployment

3. Create a deployment stage to include task to deploy ARM template and provide information related to subscription of azure account, resource group details (to use available RG or to create new RG), region, locations for ARM template file and parameters file and any additional information like override parameters and save the pipeline.



DeployARMVM
Deployment process

Agent job
Run on agent

ARM Template deployment: Resource Group scope
ARM template deployment

ARM template deployment

Task version: 3.*

Display name *: ARM Template deployment: Resource Group scope

Azure Details

Deployment scope *: Resource Group

Azure Resource Manager connection * | Manage: Free Trial (b0c515ab-ffb2-40ae-9354-109dec6295a2)

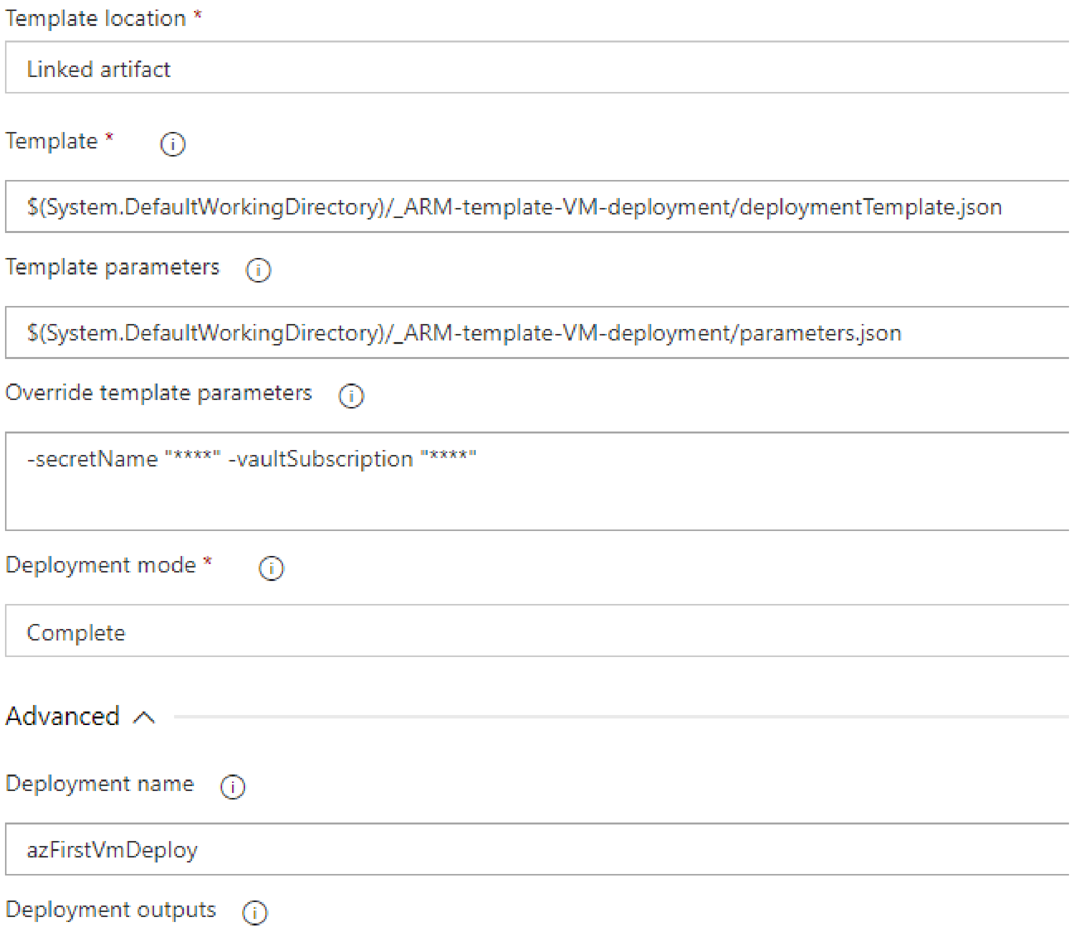
Scoped to subscription 'Free Trial'

Subscription *: Free Trial (b0c515ab-ffb2-40ae-9354-109dec6295a2)

Action *: Create or update resource group

Resource group *: aztestrsvm2

Location *: East US



Template location *: Linked artifact

Template *: $(System.DefaultWorkingDirectory)/_ARM-template-VM-deployment/deploymentTemplate.json

Template parameters: $(System.DefaultWorkingDirectory)/_ARM-template-VM-deployment/parameters.json

Override template parameters: -secretName "****" -vaultSubscription "****"

Deployment mode *: Complete

Advanced

Deployment name: azFirstVmDeploy

Deployment outputs

4) Sample ARM template is created and checked-in in github repo for reference.
Github repo url: https://github.com/tasnimbharmal/tasnim-maersk-devops-scenario

5) Password stored in Azure Key Vault can be accessed via defining keyVault details in parameters.json file with below block.

```
        "adminPassword": {
            "reference": {
                "keyVault": {
                "id": "[resourceId(parameters('vaultSubscription'), parameters('vaultResourceGroupName'), 'Microsoft.KeyVault/vaults', parameters('vaultName'))]"
                },
                "secretName": "[parameters('secretName')]"
            }
        },
```

Details of VaultResourceGroupName, VaultSubscription and VaultName needs to be entered in parameter file. And for security purpose secretName parameter can be set to null in parameter.json and can be override at the time of execution of pipeline.