2. Azure>>Full Notes

**ARM>>README.md**

**Introduction**

This folder contains two files

* template.json
* deploy.ps1

**Template.json**

* this file contains all the configuration in json format, which is also called as ARM template
* it has all the resource configuration that needs to be created

**Deploy.ps1**

* this is a powershell script, that takes few variables and when you execute it, it will create resources

**How to Execute**

* Open powershell terminal
* run below command

./deploy.ps1

**Project>>azure/ACI.md**

**To create Azure Container Instance, using AZ-CLI, follow below steps**

resourceGroup=rg-ts-200

location=eastus

storageAccount=stts200rand

containerName=aci-ts-200

image=nginx

containerPort=80

# Create a new resource group

az group create --name $resourceGroup --location $location

# Create a new storage account

az storage account create --name $storageAccount --resource-group $resourceGroup --location $location --sku Standard\_LRS

# Create a new Azure Container Instance

az container create --name $containerName --resource-group $resourceGroup --location $location --image $image --ports $containerPort

* This script creates a new resource group with the name rg-ts-200, a storage account with the name stts200rand, and an ACI with the name aci-ts-200. The ACI is created with the nginx image and exposes port 80. You can replace the image and containerPort variables with the values you need for your specific use case.

**To create Azure Container Instance, using POWERSHELL, follow below steps**

$resourceGroup = "rg-ts-201"

$location = "eastus"

$storageAccount = "stts201rand"

$containerName = "aci-ts-201"

$image = "nginx"

$containerPort = 80

# Create a new resource group

New-AzResourceGroup -Name $resourceGroup -Location $location

# Create a new storage account

New-AzStorageAccount -ResourceGroupName $resourceGroup -Name $storageAccount -SkuName Standard\_LRS -Location $location

# Create a new Azure Container Instance

New-AzContainerGroup -ResourceGroupName $resourceGroup -Name $containerName -Image $image -Port $containerPort

**Project>>azure/FunctionApp.md**

## How to use Azure CLI to create a resource group, a storage account, and a function app in the East US location

### Prerequisites

Before you begin, you will need to have the Azure CLI installed on your machine. You can install the Azure CLI by following the instructions in the [Azure CLI documentation](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli).

az group create --name rg-ts-portal-123 --location eastus

az storage account create --name sttsportal123 --resource-group rg-ts-portal-123 --location eastus --sku Standard\_LRS

az functionapp create --name fapp-ts-portal-123 --resource-group rg-ts-portal-123 --storage-account sttsportal123 --consumption-plan-location eastus --runtime python --functions-version 4 --os-type Linux

Note that we're passing the st-ts-portal-123 storage account name to the --storage-account parameter in this command, so that the function app can use it for its storage needs. We're also creating the function app with a consumption plan, in the East US location, using the --consumption-plan-location parameter.

That's it! These commands will create the resource group, storage account, and function app as per your requirements in the East US location.

## Steps to Create Resource Group, Storage Account, and Function App using POWERSHELL

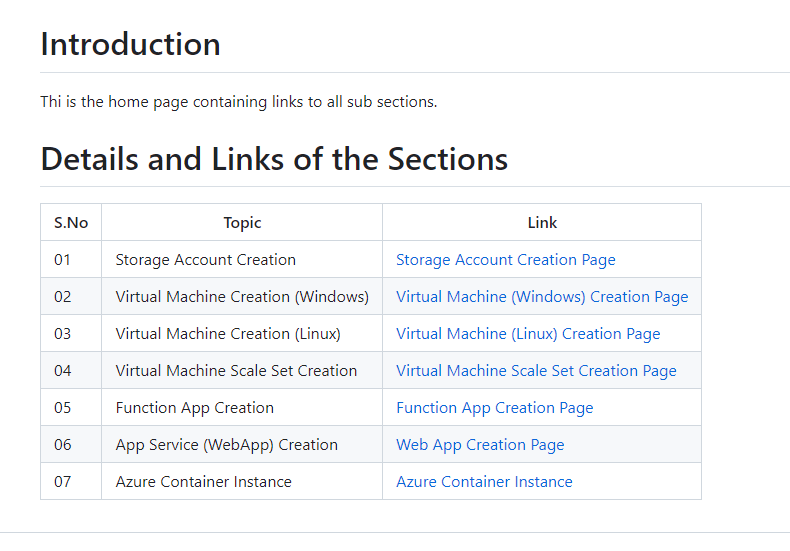
New-AzResourceGroup -Name rg-ts-001 -Location eastus

New-AzStorageAccount -ResourceGroupName rg-ts-001 -Name stts001rand -SkuName Standard\_LRS -Location eastus

New-AzFunctionApp -Name fun-ts-001 -ResourceGroupName rg-ts-001 -StorageAccount stts001rand -Runtime PowerShell -FunctionsVersion 4 -Location eastus

New-AzFunctionApp -Name fun-ts-002 -ResourceGroupName rg-ts-001 -StorageAccount stts001rand -Runtime python -FunctionsVersion 3.8 -Location eastus

**Project>>azure/README.md**



Project>>azure/StorageAccount.md

# Storage Account Creation through AZ CLI and PowerShell

## AZ CLI

### AZ Command : Example

* Make sure to install 'az cli' utility - to check run "az version" command
* "az login" - use this command to login to your az account

az group create --name <resource\_group\_name> --location <location>

az storage account create --name <storage-account-name> --resource-group <resource-group-name> --location <location> --sku <sku-name> --kind <kind-name> --encryption-services <encryption-services>

### AZ Command

az group create --name storage-az-cli --location eastus

az storage account create --name techslate1234 --resource-group storage-az-cli --location eastus --sku Standard\_LRS --kind StorageV2 --encryption-services blob --https-only true

**POWERSHELL**

### PowerShell Pre-requisites

* "Install-Module -Name Az" - To install AZ modules in PowerShell
* "Connect-AzAccount" - use this command to login to your az account

### PowerShell Command : Example

New-AzResourceGroup -Name <resource\_group\_name> -Location <location>

New-AzStorageAccount -Name <account\_name> -ResourceGroupName <resource\_group\_name> -Location <location> -SkuName <sku\_name> -Kind <kind> -EncryptionServices <services> -EnableHttpsTrafficOnly $true

### PowerShell Command

New-AzResourceGroup -Name storage-az-powershell -Location eastus

New-AzStorageAccount -Name techslate12345 -ResourceGroupName storage-az-powershell -Location eastus -SkuName Standard\_LRS -Kind StorageV2 -EncryptionServices blob -EnableHttpsTrafficOnly $true

* <account\_name>: The name of the storage account.
* <resource\_group\_name>: The name of the resource group that the storage account will be created in.
* : The location where the storage account will be created (e.g., westus).
* <sku\_name>: The name of the storage account SKU (e.g., Standard\_LRS).
* : The type of storage account (e.g., StorageV2).
* : The encryption services to enable for the storage account (e.g., blob).
* $true: A flag that indicates whether to allow only HTTPS traffic to the storage account.

**Project>>azure/VirtualMachineLinux.md**

# Virtual Machine Creation through AZ CLI and PowerShell

## AZ CLI

### AZ Command : Example

* Make sure to install 'az cli' utility - to check run "az version" command
* "az login" - use this command to login to your az account

az group create --name <resource\_group\_name> --location <location>

az vm create --resource-group <resource\_group\_name> --name myVM --image <image\_name> --admin-username <adminuser> --generate-ssh-keys

### AZ Command

az group create --name vm-linux-az-cli --location eastus

az vm create --resource-group vm-linux-az-cli --name linuxvm-using-azcli --image UbuntuLTS --admin-username azureuser --generate-ssh-keys

## POWERSHELL

### PowerShell Pre-requisites

* "Install-Module -Name Az" - To install AZ modules in PowerShell
* "Connect-AzAccount" - use this command to login to your az account

### PowerShell Command : Example

New-AzResourceGroup -Name <resource\_group\_name> -Location <location>

New-AzVm -ResourceGroupName <resource\_group\_name> -Name <vm\_name> -ImageName <image\_name> -Username <username> -SSHKeyPath "$HOME\.ssh\id\_rsa.pub"

### PowerShell Command

New-AzResourceGroup -Name storage-az-powershell -Location eastus

New-AzVm -ResourceGroupName vm-linux-ps -Name linuxvm-using-azcli -ImageName UbuntuLTS -Username azureuser -SSHKeyPath "$HOME\.ssh\id\_rsa.pub"

In this example:

* myResourceGroup is the name of the resource group.
* eastus is the location for the resource group.
* myVM is the name of the virtual machine.
* UbuntuLTS is the image used for the virtual machine.
* azureuser is the admin username for the virtual machine.
* The -SSHKeyPath option specifies the public key file for authentication, which is used to log into the virtual machine. In this case, it assumes that the public key is stored in the default ~/.ssh/id\_rsa.pub location.

**Project>>azure/WebApp.md**

# To create webapp (app-service), using AZ-CLI, follow below steps

resourceGroup=rg-ts-101

location=eastus

storageAccount=stts101rand

planName=plan-ts-101

appName=app-ts-101

az group create --name $resourceGroup --location $location

az storage account create --name $storageAccount --resource-group $resourceGroup --location $location --sku Standard\_LRS

az appservice plan create --name $planName --resource-group $resourceGroup --location $location --sku F1

az webapp create --name $appName --plan $planName --resource-group $resourceGroup

# To create webapp (app-service), using PowerShell, follow below steps

$resourceGroup = "rg-ts-100"

$location = "eastus"

$stoageAccount = "stts100rand"

$planName = "plan-ts-100"

$appName = "app-ts-100"

New-AzResourceGroup -Name $resourceGroup -Location $location

New-AzStorageAccount -ResourceGroupName $resourceGroup -Name $stoageAccount -SkuName Standard\_LRS -Location $location

New-AzAppServicePlan -ResourceGroupName $resourceGroup -Name $planName -Location $location -Tier Free

New-AzWebApp -ResourceGroupName $resourceGroup -AppServicePlan $planName -Name $appName -Location $location

**Project>>azure/azure-cli-commands.md**

Here are 50 commonly used Azure CLI commands that cover a wide range of Azure services and resource types:

* az login - logs in to Azure using a device code flow.
* az account show - shows the details of the currently authenticated subscription.
* az account set -s SUBSCRIPTION\_ID - sets the current subscription to the specified subscription ID.
* az group list - lists all resource groups in the current subscription.
* az group create -n RESOURCE\_GROUP\_NAME -l LOCATION - creates a new resource group with the specified name and location.
* az group delete -n RESOURCE\_GROUP\_NAME - deletes a resource group with the specified name.
* az vm list - lists all virtual machines in the current subscription.
* az vm create - creates a new virtual machine.
* az vm show - show details about a specific virtual machine
* az vm start - starts a stopped virtual machine.
* az vm stop - stops a running virtual machine.
* az vm deallocate - deallocates a running virtual machine.
* az vm delete - deletes a virtual machine.
* az vmss list - lists all virtual machine scale sets in the current subscription.
* az vmss create - creates a new virtual machine scale set.
* az vmss delete - deletes a virtual machine scale set.
* az network nsg list - lists all network security groups in the current subscription.
* az network nsg create - creates a new network security group.
* az network nsg rule create - creates a new security rule for a network security group.
* az storage account list - lists all storage accounts in the current subscription.
* az storage account create - creates a new storage account.
* az storage blob list - lists all blobs in a storage account.
* az storage blob upload - uploads a file to a blob storage.
* az storage blob download - downloads a blob from storage.
* az storage container create - creates a new container in a storage account.
* az cdn profile create - create a new CDN profile
* az cdn endpoint create - create a new CDN endpoint
* az appservice plan create - create a new App service plan
* az webapp create - create a new webapp
* az functionapp create - create a new function app
* az sql server create - create a new SQL server
* az sql db create - create a new SQL database
* az cosmosdb create - create a new Cosmos DB
* az monitor log-analytics workspace create - create new Log analytics Workspace
* az aks create- create a new AKS (Azure Kubernetes Service) cluster
* az aks get-credentials - get the credentials to manage a kubernetes cluster
* az functionapp deployment source config-zip - deploy function app code with zip file
* az network vnet create - create a new Virtual network
* az network public-ip create - create a new public IP
* az network nsg create - create a new network security

You can always find the most recent, up-to-date Azure CLI commands and their documentation by visiting the official Azure CLI documentation page <https://docs.microsoft.com/en-us/cli/azure/?view=azure-cli-latest>. you can also find the specific command usage by using az -h and other options for more information.