**Proof Of Concepts (POC’s) of Azure & Azure Devops:**

**Virtual Machines:**

**1)**Provision a Virtual Machine by considering different Availability options, O/S as windows, different family and series of VM’s, Vnet with subnet, auto shut down enabled, Boot diagnostics with extensions and Business tags.

**2)**Provision a Virtual Machine by considering different Availability options, O/S as Linux and perform multiple Linux commands by connecting to the VM using cloud shell, different family and series of VM’s, Vnet with subnet, auto shut down enabled, Boot diagnostics with extensions and Business tags.

**3)**Attach managed disk (Data disk) to an existing VM, Launch the disk inside the VM. Expand the existing disk size as per the business requirements.

**4)**Deploy a VM with all the basic inputs and change/increase the capacity of an existing VM

**5)**Host an IIS inside the Azure Virtual Machine and ensure the website is running on top of a VM by ensuring the VM public IP on the browser.

**6)**Create an Azure Key vault, store the secrets and keys inside the key vault, create a user in Azure AD and ensure if that user is able to access the key vault and secrets inside the key vault.

**7)**Deploy VM with multiple managed disks (data disks) in it and do perform disk encryption with Azure key vault and PowerShell scripting.

**8)**Provision a New Load Balancer and add 2 VM's in this Load Balance and ensure that the LB IP and VM IPs should be sync

**9)**Deploy 2 VM's using ARM templates and ensure that a new Vnet/LB/Subnet/IP/Disks (O/S & Data Disk)should get provisioned with this VM.

**10)**Implement and develop Azure Function Apps for serverless infrastructure provisioning of web applications and enhancements.

**PowerShell Scripting (PowerShell cmdlets) for Automations:**

**1)**Create an Azure work load with Power Shell cmdlets by considering the various components of VM(RG, Vnet, Subnet, Region, IP, NSG, Ports….etc)

**2)**Execute different Power Shell Cmdlets as demonstrated in the class.

**3)**Install Network Agent, Backup Agents or IIS Software in Azure workloads in ONE GO using Azure automation accounts, Azure Key Vault and Power Shell Cmdlets.

**4)**Apply the security/encryption on Azure VM Disks using Power Shell Cmdlets.

**Storage Accounts:**

**1)**Provision a Container Storage in Azure CCP and upload the Block blob in that container storage.

2)Create standard storage account and upload different files as Page Blob, Append Blob & Block blob to do the analysis

3)Generate SAS token for private container storage services to view and browse the data anonymous for all the users.

4)Create different container storage services with private, blob and anonymous level of access.

**5)**Provision a SA with account kind as Blob storage by working on all the tabs available in CCP and ensure the storage offering provided.

**6)**Create a File Share Storage in Azure CCP and sync/host this File share Storage in your local laptop(note: here the SMB port should be open 445) then host this on Azure Virtual Machine.

**7)**Implement SAS for Container Storage and File Share Storage and provide the Read Only Access to any user using the SAS URL

**8)**Provision a SA and ensure the data uploaded inside the SA is accessible globally with Https & Http

**9)**Deploy a Storage Account with performance as Premium with one of the storages offering and perform the analysis with Performance as standard.

**Azure Active Directory:**

**1)**Create Users and Groups in Azure Active Directory and Add Members & Owners in Azure Active Directory.

**2)**Implement the RBAC Roles (RG, VM, SA....etc) & Directory Roles(Multiple) for different users to perform the verifications after implementing the different roles for users

**3)**Finding out the authentication & Role assignments for Different users that has been implemented in Azure Active Directory.

**4)**Block the users sign-in and verify the role assignments and authentication method of the users.

**5)**Deploy an Azure Virtual Machine using PowerShell script in Azure cloud shell by considering different options like location, Vnet, Subnet, Security group, Port openings…etc. Practice all other PowerShell commands for hands-on.

**Azure Virtual Network:**

**1)**Design a Vnet consisting of at least 2 subnets.

**2)**Design multiple Vnets(vnets in different regions) and ensure to deploy 1 VM on each Vnet

**3)**Host the VM's or SA's in specific Vnet and ensure the IP's/Subnets/Vnets/NSG's....etc of that resources.

**4)**Provision the web application (App services) instance and associate this with Traffic manager profile to set up the routing preferences based up on the priority

**5)**Implement end points on TMP to setup the priority of the applications.

**6)**Develop NSG Inbound and Outbound port rules for security implementation of Azure workloads, testing of NSG ports rules with Azure workloads as per the business requirements.

**Azure SQL DB & SQL Server as IAAS & PAAS:**

**1)**Create an SQL server and SQL DB in Azure cloud platform by considering the different deployments options((i)Single DB, (ii)Elastic pool, (iii)Managed instances)

**2)**Develop an SQL DB with DTU and Vcore as per the business requirements.

**3)**Perform CRED operations in SQL DB using SQL scripts and verify the data in Azure SQL DB from private cloud to public cloud.

**4)**Deploy multiple SQL DB’s in Azure cloud on same Azure SQL instance and verify/analyse the data from Azure cloud editor.

**Azure Monitor & Log Analytics Workspace(LAW)):**

**1)**Provision an Azure Monitor, and analyse, Log analytics, Metrics, Performance counter, Alerts, Groups…etc.

**3)**Plot the different performance counters metrics for Azure workloads and save the metrics for future references in Azure Monitor

**4)**Configure the Alerts for VM's and SA's consider different threshold for servers CPU consumption and requests on SA's

**5)**Create Dashboards for different applications and try to associate the different resources to the dashboards

**6)**Provision Azure LAW, connect multiple workloads to LAW to perform the analytics using KQL queries.

**Azure Devops Organization & Project creations:**

**1)**Create an organization and configure all the settings of an Azure Devops organization.

**2)**Create an Azure Devops project with Basic process with private visibility.

**3)**Add users in Azure Devops projects and assign different level of access to Azure Devops projects.

**4)**Create teams, add members in the teams accordingly and verify the access with mail notifications and logins with user credentials.

**5)**Create Public and Private Projects in Azure Devops with version control and different work item process.

**Boards Services:**

**1)**Implement a Kanban Board with different work items process with all the field/characteristics of work items.

**2)**Design the work items in a in a hierarchy for Kanban Boards for all the work items as shown or directed in the classes.

**3)**Implement Product backlogs and Sprint backlogs and design the backlogs for Azure Boards services.

**4)**Design sprints and allocate the different work items to sprints to ensure the calendar dates for the sprints in sequential orders.

**5)**Design the details hierarchy for all the below work items for Kanban Boards

(i)Epic (ii)Feature (iii)user story (iv)task (v)bug…etc

**Repo’s Services:**

**1)**Create a Repo and using #git commands generate a .Net project then push the project from local repo to server repo

**2)**Execute all git commands to do push and pull of the .net project from local repo to server repo and server repo to local repo(vice-versa)

**3)**Add & Enhance the existing files in the project @ local Repo and push back to Server Repo

**4)**Enhance the code @ server repo by creating branches and make a pull request to merge the newly added code into master branch

**5)**Perform multiple enhancements on the code and make multiple commits in order to understand the git version control system as discussed in the classes.

**6)**Create a branch @ local repo using git commands and push the complete branch along with project in server repo using all git commands.

**7)**Perform multiple commits to save the code in local repo or server repo with pull request to code merging of different branches.

**8)**Execute different #git commands to generate .Net projects and to perform enhancements with Devops projects

**Test Plans:**

**1)**Design test cases in Kanban boards for different work items as demonstrated in the classes.

**2)**Implement the Basic+TestPlan role and provide the access to one user who is part of the project and then start designing the test cases for multiple work items designed in Kanban boards.

**3)**Design static suites, requirement-based suites, Query based suites.

**4)**Design a shared and non-shared parameters test case for multiple users with the credentials stored.

**Azure Pipelines (CI/CD):**

**1)**create a .Net project from the scratch, push the project to server repo using #git command and run the build to generate the .Yml file.

**2)**Analyse & design the .Yml file after performing the build (CI) via Azure Devops pipelines for the .net project.

**3)**Develop a website using Visual Studio .Net framework (as demonstrated in the classes) build & run the application at Visual studio to verify if the application is up and running, use #git commands to push the complete project to server repo and design a build from the scratch for Continuous integration (CI).

**4)**Design Continuous Delivery (CD) to deploy the .net web application & an IIS on an Azure workload in Azure cloud computing platform (Single Stage-IAAS)

**5)**Design Continuous Delivery (CD) to deploy the .net web application & an IIS on multiple Azure work load (2-3 as Dev, QA & Prod) in Azure cloud computing platform (Multi Stage-IAAS)

**6)**Deploy .net project web application on Azure App services using Visual studio .net framework to build and publish the project on Azure PAAS service.

**7)**Understand the complete architecture from Azure Repo’s to Azure CI/CD pipelines the complete flow of code, enhancements, code build, and code deploy (hint refer material)

**Docker containers or Docker Images:**

**1)**Install and configure Azure CLI in the cloud server to set up the Docker containers.

2)Implement a Private Docker Container Registry in Ubuntu Linux Servers in Azure cloud computing

3)Execute the docker container instance and docker container registry commands to analyse the details of docker image, docker registries, docker image versions.

4)Host custom image using node js application and deploy this image on docker container instance using private container registry.

5)Host web application on docker container instance from microsoft container registry.

6)create a private container registry in Azure cloud server for docker container instance.

**Ansible Scripting:**

**1)**Create Ansible Playbook to provision different resources in Azure cloud company Azure container instance to deploy a web application with docker container image on ACI on top of Linux O/s.

**2)**Deploy container instance, container registry and Custom image in Azure Container Registry and then host this image on Azure Container Instance (ACI) with Node JS application

**3)**Deploy container registry with custom images for various image versions.

**2)**Execute Ansible scripts to communicate with win rm, ansible control server that needs the python package for Ansible Playbooks.

**3)**Create Ansible Playbooks to provision RG, Vnet, Subnet, Public IP, NSG (for inbound security rules), NIC, Virtual Machines…. etc.

**4)**Execute Ansible scripts to open and execute the code in an Editor (to insert the Ansible playbooks code) for various Azure cloud resources.

**Terraforms:**

**1)**Design the modules for Azure Resources in Terraforms scripting.

**2)**Download, Install & Configure the terraform dll’s, and exec’s in on-prem (private cloud) to plot the terraform code for resource provisioning.

**3)**Create an application object and register the same in Azure Active Directory for authentication and authorization purposes to deploy the resources in the cloud platform.

**4)**Write the code for different blocks/patterns to provision multiple resources in CCP in one go.

**5)**Develop the terraform code to execute the script in Cloud Shell for deploying resources directly in Azure cloud computing platform(CCP).

**6)**Trigger/Exec the terraform commands in sequence and order to host the resources in the cloud platform.

**Azure Kubernetes Services (AKS):**

1)Deploy Azure Kubernetes cluster by considering node size, node count, node pools, enablement of virtual nodes, cluster Vnet, cluster subnets, Kubernetes version…etc in Azure cloud platform.

**2)**Write a Yml code to provision virtual machines, app name deployment, port numbers, Operating system, Load Balancers…etc as per the need of project requirements.

**3)**Execute Kubectl commands to get the details and insight of Kubernetes components ... .node pools, pods, containers, Kubernetes worker nodes, namespaces of Kubernetes cluster, namespaces of the pods.

**4)**Install Azure Kubernetes services AKS CLI in private cloud to do the governance and operations from local command prompt for deployment AKS containers, pods, and application deployments inside the pods.

**5)**Execute AKS commands to login to Kubernetes cluster, maintain and manage the Kubernetes cluster and its components.

**6)**Deploy an application on Kubernetes pods by triggering the various AKS commands and its services.

**7)**Test the deployed application in Kubernetes pods by considering the IP addresses of Pods which are same as LB(as this is mentioned in the Yaml file)