**Microsoft Interview Questions**

1. **Any knowledge on microservices. How do you rate yourself?**

Ans: A microservice is a small, loosely coupled distributed service.

It is an approach to application development in which a large application is built as a suite of modular components or services.

For example, Docker is a light weight container service used on-prem or cloud to deploy applications.

Kubernetes is an orchestration tool help in automating scaling, deployment and container management.

1. **Explain the Devops flow by taking any project.**

Ans: For example take ASP Net web application project.

To start import a repo into Azure repos using the clone URL.

After you have the sample code in your repo create a build pipeline via classic editor to add the tasks required to build code. Save and queue to see build in action. Once build is succeeded an artifact is generated and stored in the path specified like in azure Artifacts or in file share storage.

After you have the Artifact in the specified location create a release pipeline by choosing a Template or Empty Job. Choose the task where to deploy the application to this stage. Define the trigger that will start deployment to this stage. Enable Artifact filters to include or exclude build Branch. Select the users who can approve or reject deployments to this stage. Define gates to evaluate before the deployment. Define behaviour when multiple releases are queued for deployment.

Continuous deployment trigger is enabled when a new build is available or when a selected artifact is available as part of a pull request workflow.

1. **How you push your code to azure repos ?**

**Ans: -** Create a new repository in Azure Repos.

* Create a local repo in computer
* Clone Azure repo to your local computer with git credentials
* Push the code

1. **What kind of branching strategy you used in your project?**

Ans: Release merge strategy. Branching & Merging - Strategies

**Git Hub Flow**

You create a new branch for every feature or bugfix and merge it to main as soon as you are done. Branches might live for days or weeks and might have to spent some time merging.

**GitFlow**

A more elaborate scheme that allows for parallel feature development, an integration branch and hotfixes.

**Trunk Based development**

No formal source control branching strategy.

Every changes goes into master; every change is deployed.

1. **Which branch code you check out? Master or feature and to which environment?**

Release: Dev and Test

1. **Have you used any branch policy? where and how will you assign?**

Ans: Azure Repos ---- > branches --- > Branch policy

**Branch Policies:**

* Require a minimum number of reviewers
* Check for linked work items
* Check for comment resolution
* Limit merge types

Setting a Required policy will enforce the use of pull requests when updating the branch

Setting a Required policy will prevent branch deletion

Manage permissions for a branch on the Security page

1. **How you created build pipeline?**

Ans: A CI pipeline or Build pipeline is the process of taking code from version control and making it readily available to users of your application in an automated fashion via compile, build, test.

Build succeeded implies an artifact generated.

Build pipeline can be created using YAML file or Classical editor.

1. **Any experience in using YAML file build pipeline creation?**

Ans: Connect to the source code ---- > select the Repo of project ---- > configure YAML file (i.e by adding steps that publish symbols, save build artifacts, deploy, and more --- > Review yml file

The yml file consists of

trigger: continuous integration triggers from specified branches

pool: pool where this job runs

variables: variables for this pipeline

steps: a list of steps to run this job

tasks:

to include pre-release versions of SKDs as an extra task to install latest versions

to include restore package from .net core/Nuget/npm/maven/Pypi

to include build

to include tests

to include publish test results

to include collect code coverage metrics/ coverlet(on LINUX)

to include publish artifact to azure pipelines or to a NuGet feed

to include to deploy a web app

to include to build an image and push to container registry

1. **What is the difference between Microsoft hosted and self-hosted agents?**

An **agent** is an installable software that runs one job at a time.

**Microsoft-hosted agent:** run jobs directly on the VM or in a container which is maintained and managed by Microsoft. And the virtual machine is discarded after one use. Thereby cleans the repo and perform clean build every time which is not suitable for incremental builds. sometimes it takes several minutes to allocate an agent depending on the load on system

**Self-hosted agent**: run jobs on VM maintained and managed by own. And more control to install dependent software needed for your builds and deployments. Machine-level caches and configuration persist from run to run, which can boost speed. Help to avoid downloading the source code to a fresh machine for every build

1. **How to run a task always even if the preceding task failed?**

**Ans: Build pipeline --- >Agent Job --- > Task ---- > Control options ----- > Run this task**

* Only when all previous tasks have succeeded
* Even if a previous task has failed, unless the build was canceled
* Even if a previous task has failed, even if the build was canceled
* Custom conditions

1. **Where you deploy ur applications? App services**

Ans: Azure App Service is an HTTP-based service that enables you to build and host many types of web-based solutions without managing infrastructure.

1. **How you give services connections? 2 ways? Any idea on principle account way of giving service connection?**

Ans: To deploy an app to an Azure resource (to an app service or to a virtual machine), you need an Azure Resource Manager service connection

Service Principal AuthenticationManaged Identity Authentication

**Steps to Create an Azure Resource Manager service connection with an automatic generated service principal**

* **Connection name:** The name used to refer to this service connection
* **Scope level:** Select Subscription or Management Group
  + - Subscription is to select an existing Azure subscription
    - Management groups are containers that help you to manage access, policy, and compliance across multiple subscriptions.
* **Resource Group**: Leave empty to allow users to access all resources defined within the subscription, or select a resource group to which you want to restrict users' access
* **Click OK** to create a service connection
* Give **connection name** and **App service name** in the settings of your pipeline

**Steps to Create an Azure Resource Manager service connection with an existing service principal**

To use a pre-defined set of access permissions, create an Azure AD application and service principal

Create an Azure Active Directory application

1. Sign in to your Azure Account through the Azure portal.
2. Select **Azure Active Directory**.
3. **App registrations > New registration**.
4. **Register an application Name**
5. **Supported account types**
   * 1. **Accounts in this organization**
     2. **Accounts in any organization**
     3. **Accounts in any organization and personal Microsoft accounts**
6. Enter the redirect URL for application and click register

Assign the application to a role

* + - 1. Portal **All services** and **Subscriptions**
      2. Select the particular subscription
      3. Select **Access control (IAM) and Add role assignment**.
      4. select the **Contributor** role to your AD application and SAVE

1. **What are ARM templates?**

**Ans:** ARM templates are used to implement infrastructure as code for Azure

1. **What are the different elements in ARM template?**

**Ans:**

“$schema": Location of the JSON schema file

"contentVersion": Version of the template (such as 1.0.0.0)

"apiProfile": An API version that serves as a collection of API versions for resource types

"parameters": Defines inputs to deployment

"variables": Reused values to simplify repeating complicated expressions

"functions": User-defined functions that are available within the template

"resources": Resource types that are deployed or updated in a resource group or subscription.

"outputs": Values that are returned after deployment

1. **What are parameters and variables? Parameters alone are enough what is the need of variables?**

**Parameters:** Defines inputs to deployment

**Variables:** Reused values to simplify repeating complicated expressions

1. **What are storage accounts? And explain what are the available features in storage accounts?**

Azure storage accounts are durable & Highly Available, secure, scalable, managed and accessible from anywhere.

Azure storage comes in 4 varities

**Blob:** object storage for text and binary

**File:** file storage for cloud or on-prem deployments

**Queue:** messaging store

**Tables:** nosql store of structured data

1. **What is the difference between service bus queues and storage queues?**

Storage Queues: part of azure storage infra

a simple REST-based GET/PUT/PEEK interface

Service Bus Queues: a part of Azure messaging infra

designed to integrate applications or application components that may span multiple communication protocols, data contracts, trust domains, and/or network environments.

1. **What are SAS keys?**

Shared Access Signature keys --- > gives limited access to the objects in the storage account to the third party. It is a signed URI with a token

1. **Explain how you configure site-to-site and point-to-site vpn connections?**

**Ans: site-2-site:** connects on-prem network with azure networks

* Create a virtual network
* Create the VPN gate way in azure
* Create the local network gateway
* Configure vpn device
* Create the vpn connection
* verify the vpn connection

**Point-2-site:** connects from remote location with azure network

* Create a virtual network. ...
* Create a virtual network gateway. ...
* Generate certificates. ...
* Add the client address pool. ...
* Configure tunnel type. ...
* Configure authentication type
* Upload the root certificate public certificate data
* Install an exported client certificate
* Generate and install the VPN client configuration package
* Connect to Azure

1. **What is the need of network gateways?**

**Ans:** a stopping point for data on its way to or from other networks.

1. **What is the purpose of NSGs? Have you configured any rules?**

NSGs are nothing but firewalls. They protect subnets or VMs using rules. Security rules are evaluated in sequence based on priority.

1. **When a Vm created what are the other default resources created? Will the Vnet be reusable?**

**Ans:** Vnet,NSGs,NIC,Public IP. Yes Vnet can be reused with other resources

1. **How do you link storage account to your APP services?**

**Ans:** To mount a storage account to a directory in your App Service app, you use the **az webapp config storage-account add** command. Storage Type can be AzureBlob or AzureFiles.

az webapp config storage-account add --resource-group <group\_name> --name <app\_name> --custom-id <custom\_id> --storage-type AzureBlob --share-name <share\_name> --account-name <storage\_account\_name> --access-key "<access\_key>" --mount-path <mount\_path\_directory>

Linking an existing directory in a web app to a storage account will delete the directory contents. If you are migrating files for an existing app, make a backup of your app and its content before you begin.

1. **How do you troubleshoot your web application deployed in APP services?**

**Ans:** App Service provides a dedicated, interactive diagnostics tool, Genie, to troubleshoot your application.

Open App service diagnostics and choose a category

* **Availability and performance**
* **Configuration and management**
* **SSL and Domains**
* **Best Practices**
* **Diagnostic Tools**

Once a category is selected like Availability and performance, choose a title

* **Application Logs**
* **Container issues**
* **cpu usage, memory usage,port usage**
* **process list**
* **web app down, web app restarted**

If application logs selected more topics can be investigated further with graphs and markdowns using diagnostic reports.

If you don't know what’s wrong with your app or don’t know where to start troubleshooting your issues, the health checkup is a good place to start.

Investigate application code issues (only for Windows app) with Apllication insights.

Also Diagnostic tools help to investigate application code issues, slowness, connection strings, and more. and proactive tools help to mitigate issues with CPU usage, requests, and memory.

In addition, we can use Azure Monitor to improve the logging and monitoring capabilities of app.

1. **Explain the configuration settings in App services**?

**Application settings:**

Frame work version: .net,php,java,python

Platform: 32-bit or 64-bit

Web socket: set on to enable web socket protocol

Always on: by default, web apps are unloaded. Always on to keep the app loaded all the time

Auto-swap: automatically swaps slot into production when you push code to it.

**Connection Strings:**

Allow to specify database servers that can be utilized per slot

It is a variable instead of a configuration file

It is secure because it doesn’t store information as a file

**Handler Mappings:**

Allow you to choose specific file extensions and handled by specific script process you define

**Virtual Applications and Directories:**

Virtual directory: path users will take to access the application

Physical path: path to the physical directory or application

Application: check this to mark a virtual directory as an application

PublishBuildArtifact takes the files in the path you specify, uploads them to Azure DevOps, and attaches them to the build so that they can be downloaded and consumed in other pipelines.

1.       A branch policy

a.       Which enforces atleast approval from 2 reviewers before completing it

b.       Don’t allow the individual user who has checked in the code to be part of that 2 reviewers. Can you do that, if yes how?

c.       How do you make sure that pull request cannot be complete if there is no work item associated

d.       Can you mention something in comment to close the work item automatically?

e.       How to make sure that comments are resolved before PR completes?

f.        How do you make sure that the branches to be created in folders

g.       How do you make sure that a code which might not result in compilation stops the PR automatically?

h.       When do you lock a branch? And what happens when you do that? How do you lock a branch?

i.         Can someone override a branch policy? If yes how?

j.         What are the different branch permissions?

2.       What is a .gitignore file?

3.       What is a lightweight and annotated tag?

4.       How are tags useful?

5.       Can you create branch from a tag?

6.       What are git limits? Repository and Push size limit?

<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-azure-devops/>

<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/v2-windows?view=azure-devops>

1.       Understand container and uses

2.       Containerization Vs Virtualization – when to use what

3.       Create a docker image

4.       Push a docker image to ACR

5.       Run an application on Azure container instance

6.       Run an application on Azure Kubernetes Services

7.       Enabling diagnostics logs

8.       Docker Life Cycle

9.       Helm Charts

10.   Docker Compose

11.   Dockerfile

12.   Docker Swarm

13.   Image life cycle in dev Prod test

14.   Components of Docker

15.   Components of Kubernetes master and slave

16.   Docker swarm VS Kubernetes

17.   Deployment of Applications branches

1.       Virtualization Vs Containerization. Through a block diagram.

2.       What is relationship between Kubernetes and Docker?

3.       What is container orchestration mean to you?

4.       What are the features of Kubernetes?

a.       Same answer can be used to tell why you prefer kubernetes over docker swarm lets say.

5.       What is kube-controller-manager, kube-apiserver, kube-scheduler, etcd, kubelet, kube-proxy?

6.       Which component of the Kubernetes Master is responsible for issuing work to Nodes? (A: scheduler)

7.       What is a pod, node, and replica terminology?

8.       How do two containers in the same Pod communicate? (A: Via the shared localhost adapter inside the Pod)

9.       How is load balanced in Kubernetes cluster? How do you troubleshoot if load is not reaching a particular node?

10.   Where do you specify the node size in the cluster like which series (what cpu, memory etc) and can you change it after the cluster is created?

11.   Where do you specify number of nodes in the cluster? How do you scale your kubernetes cluster manually and autoscale?

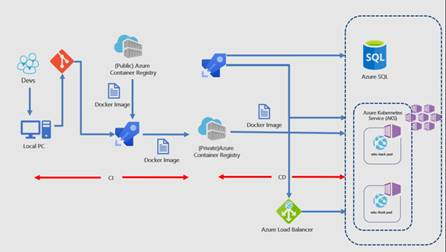
12.   Docker

a.       Docker commands for pull, build, images, ps, rm, stop, run

b.       What is the docker development environment you used?

c.       Sample yaml file for docker image creation.

13.   What is the project architecture with kubernetes and devops – be able to explain by drawing similar to below [diagram](https://www.azuredevopslabs.com/labs/vstsextend/kubernetes/):



14.   What is docker, Azure Container Registry, Azure Container Instance, AKS, nodes, pod, master node

15.   How do you authorize AKS to connect to ACR

a.       Hint: using service principle generated from AKS.

b.       Follow up question will be how will you generate service principle?

16.   What are the general steps carried out in the build pipeline and what are carried out in the release pipeline? W.r.t deploying an application to AKS?

17.   What are the monitoring tools that you are aware of? Azure monitor how do you enable as part of creating AKS steps?

18.   What is difference between ACS and AKS? (A: ACS is deprecated now… )

**Some scenario based Q&A:**

 You wish to check on your Pods, however, you are not on the terminal that they were created on, so you need to connect to them from an external source. Unfortunately you are unable to. Why?

A: You did not set a nodePort on your service.

You have a running pod that has two Containers. Due to an unknown failure, a container shuts down. How can you restart the Container while keeping the pod phase running?

A: Keeping restartPolicy as Always.

 You are working on an Azure Kubernetes Service (AKS) and while trying to upgrade or scale you get a message Changing property 'imageReference' is not allowed. What is a possible source of this problem?

A: Modifying the tags in the agent nodes inside the AKS cluster.

 You begin to notice strange behavior in a Pod that is running a node. You believe that it's an issue with Kubelet. How could you check if a node is down?

A: Run the get nodes command and check if the status says NotReady.

 You notice that one of the pending pods you have is not running, so you check its events. You see FailedScheduling in the Reason column, meaning that there are not enough resources for the Pod on any of the nodes. What can you do to fix this situation?

A: Use the kubectl scale command to update your deployment and specify number of replicas.