

**BY DEVOPS  
SHACK**

# **100 AZURE DEVOPS INTERVIEW QUESTIONS**



**Azure  
DevOps**



## 1. What is Azure DevOps?

### Answer:

Azure DevOps is a cloud-based DevOps platform providing CI/CD pipelines, version control, testing, and deployment automation.

## 2. What is CI/CD in Azure DevOps?

### Answer:

Continuous Integration (CI): Automates building and testing code changes.

Continuous Deployment (CD): Automates deploying the application to environments.

## 3. What are the core components of Azure DevOps?

### Answer:

Azure Repos – Version control

Azure Pipelines – CI/CD automation

Azure Artifacts – Package management

Azure Test Plans – Testing framework

Azure Boards – Agile project tracking

## 4. What is an Azure Pipeline?

### Answer:

Azure Pipelines is a CI/CD service that automates software build, test, and deployment.

## 5. What is the difference between and Classic Pipelines?

### Answer:

Pipelines: Code-based, version-controlled.

Classic Pipelines: GUI-based, easier setup.



## 6. How do you define a pipeline using ?

**Answer:**

trigger:

branches:

include:

- main

stages:

- stage: Build

jobs:

- job: BuildJob

steps:

- script: echo "Building the project"

## 7. What are triggers in Azure Pipelines?

**Answer:**

Manual Trigger

Commit Trigger

Scheduled Trigger

Pipeline Trigger

Pull Request (PR) Trigger

## 8. What is an Agent in Azure DevOps?

**Answer:**

An agent is a virtual machine (VM) that runs pipeline jobs.

Microsoft-hosted agents: Managed by Azure.

Self-hosted agents: Installed by users for custom configurations.



## 9. How do you store secrets securely in Azure Pipelines?

**Answer:**

Use Azure Key Vault.

Use Pipeline Variables with "secret" option.

## 10. What are Service Connections in Azure DevOps?

**Answer:**

Service Connections allow pipelines to securely connect to external services like Azure, AWS, or Kubernetes.

## 11. How do you restrict a pipeline to run on specific branches?

**Answer:**

trigger:

branches:

include:

- main

exclude:

- feature/\*

## 12. What are Artifacts in Azure Pipelines?

**Answer:**

Artifacts are build outputs (binaries, packages, Docker images) stored and shared between pipeline stages.

## 13. What is a Multi-Stage Pipeline?

**Answer:**

A multi-stage pipeline has separate stages for build, test, and deployment.



---

**14. How do you implement pipeline approvals in Azure DevOps?****Answer:**

Use pre-deployment approvals in Environments.

**15. What is a Deployment Group?****Answer:**

A Deployment Group is a collection of servers or VMs used for on-premises deployments.

**16. How do you monitor pipeline execution in Azure DevOps?****Answer:**

Use Logs, Azure Monitor, and Application Insights.

**17. How do you implement CI/CD for Docker containers?****Answer:**

Use Azure Container Registry (ACR) with Azure Kubernetes Service (AKS).

**18. How do you enforce code quality in Azure Pipelines?****Answer:**

Use SonarQube for static analysis.

Enable code coverage reports in tests.

**19. How do you implement rollback in Azure Pipelines?****Answer:**

Re-deploy the previous successful artifact.

Use App Service Slots to swap environments.



## 20. How do you cache dependencies in Azure Pipelines?

**Answer:**

```
- task: Cache@2
```

```
inputs:
```

```
key: 'npm | package-lock.json'
```

```
path: $(npm_config_cache)
```

## 21. What is a Self-Hosted Agent?

**Answer:**

A self-hosted agent is an agent you install on your own server instead of using Microsoft's cloud agents.

## 22. How do you implement branch policies in Azure Repos?

**Answer:**

Enable branch policies for PR approvals, required builds, and work item linking.

## 23. How do you deploy to Azure App Service using Azure Pipelines?

**Answer:**

```
- task: AzureWebApp@1
```

```
inputs:
```

```
azureSubscription: 'MyAzureServiceConnection'
```

```
appName: 'my-web-app'
```

```
package: '$(Build.ArtifactStagingDirectory)/myapp.zip'
```

## 24. How do you configure pipeline notifications?

**Answer:**

Enable email, Teams, or Slack notifications in Pipeline Settings.



## 25. How do you implement Blue-Green Deployment?

### Answer:

Run two environments (Blue = Old, Green = New) and switch traffic using Azure Traffic Manager.

## 26. How do you implement Infrastructure as Code (IaC) in Azure Pipelines?

### Answer:

Infrastructure as Code (IaC) automates infrastructure provisioning using ARM templates, Terraform, or Bicep.

### Example using Terraform in Azure Pipelines:

```
- task: TerraformTaskV1
```

```
inputs:
```

```
command: 'apply'
```

```
workingDirectory: '$(System.DefaultWorkingDirectory)/terraform'
```

## 27. How do you automate database deployments in Azure Pipelines?

### Answer:

Use DACPAC for SQL databases or Entity Framework Migrations.

### Example using DACPAC in :

```
- task: SqlAzureDacpacDeployment@1
```

```
inputs:
```

```
azureSubscription: 'AzureServiceConnection'
```

```
serverName: 'myserver.database.windows.net'
```

```
databaseName: 'mydb'
```

```
dacpacFile: '$(Build.ArtifactStagingDirectory)/database.dacpac'
```



## 28. What is a Canary Deployment, and how do you implement it?

### Answer:

Canary Deployment releases new updates to a subset of users before full deployment.

Example using Azure App Service Traffic Routing:

```
- task: AzureAppServiceManage@0
```

```
inputs:
```

```
  action: 'Set Traffic Routing'
```

```
  appName: 'my-app'
```

```
  trafficWeight: '20'
```

## 29. How do you integrate Terraform with Azure Pipelines?

### Answer:

Install Terraform CLI in your pipeline.

Use Terraform Task to deploy resources.

### Example:

```
- task: TerraformTaskV1
```

```
inputs:
```

```
  command: 'apply'
```

```
  workingDirectory: '$(System.DefaultWorkingDirectory)/terraform'
```

## 30. How do you deploy Kubernetes Helm Charts using Azure Pipelines?

### Answer:

Use Helm tasks in Azure Pipelines to deploy Helm Charts.

### Example:

```
- task: HelmDeploy@0
```

```
inputs:
```

```
  connectionType: 'Azure Resource Manager'
```

```
  azureSubscription: 'MyAzureSubscription'
```





```
kubernetesCluster: 'my-aks-cluster'
```

```
namespace: 'my-namespace'
```

```
command: 'upgrade'
```

```
chartType: 'FilePath'
```

```
chartPath: 'charts/mychart'
```

### 31. How do you scan Docker images for security vulnerabilities in Azure Pipelines?

**Answer:**

Use Trivy or Microsoft Defender for Containers.

**Example using Trivy:**

```
- script: |  
  trivy image myregistry.azurecr.io/myapp:latest
```

### 32. What is Zero Downtime Deployment, and how do you implement it?

**Answer:**

Deploy the new version alongside the old one and swap them when ready.

**Example using Azure App Service Slots:**

```
- task: AzureAppServiceManage@0  
  inputs:  
    action: 'Swap Slots'  
    appName: 'my-app'  
    sourceSlot: 'staging'  
    targetSlot: 'production'
```

### 33. How do you handle secret rotation in Azure Pipelines?

**Answer:**

Store secrets in Azure Key Vault.

Rotate them automatically and update pipeline references.



**Example Key Vault Task:**

```
- task: AzureKeyVault@1
```

```
inputs:
```

```
azureSubscription: 'MyAzureServiceConnection'
```

```
KeyVaultName: 'myKeyVault'
```

```
SecretsFilter: '*'
```

**34. How do you enforce compliance policies in CI/CD?****Answer:**

Use Azure Policy and Defender for Cloud to enforce security rules.

**35. What is the difference between Rolling Deployment vs Blue-Green Deployment?****Answer:**

Rolling Deployment	Blue-Green Deployment
Updates instances in batches	Runs two environments (Blue = Old, Green = New)
Partial downtime if errors occur	Zero downtime
Slower rollback	Instant rollback

**36. How do you implement automatic rollback in Azure Pipelines?****Answer:**

Use health checks to detect failures.

Re-deploy last stable version if the new release fails.

**36. How do you handle failures in Azure Pipelines?****Answer:**

Use error handling mechanisms such as set -e in shell scripts.

Enable retry policies for transient failures.

Configure alerts and notifications in Azure Monitor.



**Example:** Using set -e to Stop Execution on Failure in

```
#!/bin/  
set -e  
echo "Starting process..."  
some_command_that_might_fail  
echo "This line will not execute if the above command fails."
```

### 37. How do you implement approval gates in Azure DevOps pipelines?

**Answer:**

Approval gates ensure that manual approval is required before deploying to production.

**Steps to Implement:**

Navigate to Azure DevOps → Pipelines → Environments.

Configure pre-deployment approvals for environments.

Assign users or groups for approval.

**Example:** Adding Manual Approval in

```
- stage: Deploy  
  jobs:  
    - job: Approval  
      steps:  
        - task: ManualValidation@0  
          inputs:  
            notifyUsers: 'admin@company.com'  
            instructions: 'Approve the deployment'
```

### 38. How do you enforce security policies in Azure Pipelines?

**Answer:**

Use Azure Policy to enforce compliance.

Use Azure Key Vault for secrets management.



Enable Azure Security Center recommendations.

**Example:** Enforce Security Policies in Azure DevOps

```
- task: AzureSecurityCenterAssessment@1
```

inputs:

```
subscriptionId: 'my-subscription'
```

```
policyAssignmentName: 'EnforceSecurityPolicies'
```

### 39. What is an Artifact in Azure Pipelines?

**Answer:**

An artifact is the output of a build process, such as binaries, Docker images, or deployment packages.

**Types of Artifacts in Azure Pipelines:**

**Build Artifacts:** .zip, .jar, .dll files.

**Pipeline Artifacts:** Shared files between pipeline stages.

**Azure Artifacts:** NuGet, npm, Maven package storage.

### 40. How do you deploy an application to Azure App Service using Azure Pipelines?

**Answer:**

Use the Azure WebApp task in pipelines.

**Example:** Deploying a Web App

```
- task: AzureWebApp@1
```

inputs:

```
azureSubscription: 'MyAzureServiceConnection'
```

```
appName: 'my-web-app'
```

```
package: '$(Build.ArtifactStagingDirectory)/myapp.zip'
```



#### 41. How do you deploy to multiple environments using Azure Pipelines?

**Answer:**

Use separate stages for Dev, QA, Staging, and Production.

Define environment variables per stage.

Implement approval gates before production deployments.

**Example:** Multi-Environment Deployment in

stages:

- stage: Dev

jobs:

- job: DeployToDev

steps:

- script: echo "Deploying to Dev"

- stage: Production

dependsOn: Dev

jobs:

- job: DeployToProd

steps:

- script: echo "Deploying to Production"

#### 42. How do you integrate SonarQube with Azure Pipelines?

**Answer:**

SonarQube is used for static code analysis and security scanning.

**Steps to Integrate:**

Install SonarQube extension in Azure DevOps.

Add SonarQubePrepare, SonarQubeAnalyze, and SonarQubePublish tasks.

Set up quality gates.



**Example:** SonarQube Integration in

```
- task: SonarQubePrepare@4
```

```
inputs:
```

```
  SonarQube: 'SonarQubeServiceConnection'
```

```
  scannerMode: 'CLI'
```

```
  extraProperties: |
```

```
    sonar.projectKey=my-project
```

```
    sonar.organization=my-org
```

### 43. How do you configure release variables in Azure Pipelines?

**Answer:**

Release variables allow different configurations per environment.

**Steps to Configure:**

Navigate to Azure DevOps → Pipelines → Variables.

Define global variables or per stage variables.

Use variables in the pipeline.

**Example:** Using Variables in a Pipeline

```
variables:
```

```
  env: 'Production'
```

```
steps:
```

```
- script: echo "Deploying to $(env)"
```

### 44. How do you trigger an Azure Pipeline on a schedule?

**Answer:**

Use cron syntax in to schedule pipelines.



**Example:** Run Pipeline Every Day at 10 AM

schedules:

- cron: "0 10 \* \* \*"

displayName: "Daily 10 AM Build"

branches:

include:

- main

**45. How do you set up role-based access control (RBAC) in Azure DevOps?**

**Answer:**

Assign Azure DevOps roles (Reader, Contributor, Admin).

Define RBAC roles for specific pipelines and projects.

Use Service Connections for external access control.

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

**Answer:**

Feature	Microsoft-Hosted	Self-Hosted
Managed By	Azure	User
Speed	Fast but shared	Custom & dedicated
Custom Software	Limited	Full control
Cost	Free for limited builds	Requires own VM

**47. How do you enforce PR validation in Azure Repos?**

**Answer:**

Enable Branch Policies in Azure Repos.

Require approvals and passing builds before merging PRs.

**48. How do you configure artifact retention policies?**

**Answer:**

Navigate to Pipelines → Retention Policies.

Set maximum artifact retention period.



Use to delete old artifacts.

#### 49. How do you automate Terraform deployments in Azure Pipelines?

**Answer:**

Use Terraform tasks for infrastructure provisioning.

**Example:** Terraform Apply in

```
- task: TerraformTaskV1
```

```
inputs:
```

```
command: 'apply'
```

```
workingDirectory: '$(System.DefaultWorkingDirectory)/terraform'
```

#### 50. How do you deploy Helm charts in Azure Pipelines?

**Answer:**

Use Helm task to deploy charts to Kubernetes.

**Example:**

```
- task: HelmDeploy@0
```

```
inputs:
```

```
command: 'install'
```

```
chartPath: './charts/mychart'
```

#### 51. How do you implement blue-green deployment in Azure Pipelines?

**Answer:**

Blue-Green Deployment ensures zero downtime deployment by switching between two environments (Blue and Green).

**Steps to Implement:**

Deploy the new version (Green) alongside the old one (Blue).

Use Azure Traffic Manager or App Service slots to switch traffic.

Rollback to the old version if issues arise.





**Example:** Swapping App Service Slots in

```
- task: AzureAppServiceManage@0
```

```
inputs:
```

```
  action: 'Swap Slots'
```

```
  appName: 'my-app'
```

```
  resourceGroupName: 'my-resource-group'
```

```
  sourceSlot: 'staging'
```

```
  targetSlot: 'production'
```

**52. How do you deploy an application to Azure Kubernetes Service (AKS) using Azure Pipelines?**

**Answer:**

Deploying to AKS requires kubectl or Helm.

**Example:** Deploying with kubectl

```
- task: Kubernetes@1
```

```
inputs:
```

```
  connectionType: 'Azure Resource Manager'
```

```
  azureSubscription: 'MyAzureServiceConnection'
```

```
  kubernetesCluster: 'my-aks-cluster'
```

```
  namespace: 'my-namespace'
```

```
  command: 'apply'
```

```
  arguments: '-f deployment.'
```

**53. How do you configure rollback in Azure Pipelines if deployment fails?**

**Answer:**

Use Azure App Service Slot Swaps for rollback.

Use AKS rollback with kubectl rollout undo.

Store previously successful artifacts for rollback.



**Example:** Rollback to the Last Stable Kubernetes Deployment

```
- script: |  
  kubectl rollout undo deployment my-app
```

#### 54. How do you enforce mandatory approvals in Azure DevOps Pipelines?

**Answer:**

Use Azure Pipelines Pre-Deployment Approvals.

**Steps:**

Go to Environments → Approvals and Checks.

Add approvers (QA, Security, etc.).

Ensure pipeline stops until approval is granted.

**Example:** Manual Validation in

```
- task: ManualValidation@0  
  inputs:  
    notifyUsers: 'qa@company.com'  
    instructions: 'Please approve the deployment'
```

#### 55. How do you automate security scanning for CI/CD in Azure Pipelines?

**Answer:**

Use Microsoft Defender for Containers for container security.

Use OWASP Dependency Check for Java/.NET security.

Use SonarQube for static code analysis.

**Example:** Using Trivy to Scan Container Images

```
- script: |  
  trivy image myregistry.azurecr.io/myapp:latest
```



## 56. How do you configure auto-scaling for Azure DevOps self-hosted agents?

### Answer:

Use Azure VM Scale Sets to dynamically add/remove agents.

Use containerized agents in Kubernetes for auto-scaling.

**Example:** Scaling Self-Hosted Agents Using Kubernetes

```
- task: Kubernetes@1
```

```
inputs:
```

```
command: 'scale'
```

```
arguments: '--replicas=5 deployment/azure-agent'
```

## 57. How do you optimize pipeline execution time in Azure DevOps?

### Answer:

Enable pipeline caching for dependencies.

Use parallel jobs to speed up execution.

Use self-hosted agents for performance improvements.

**Example:** Caching npm Dependencies in

```
- task: Cache@2
```

```
inputs:
```

```
key: 'npm | package-lock.json'
```

```
path: $(npm_config_cache)
```

## 58. How do you enforce branch protection rules in Azure Repos?

### Answer:

Require code reviews before merging.

Enforce CI build success before merge.

Restrict who can push to main branches.



## 59. How do you deploy a database schema update using Azure Pipelines?

**Answer:**

Use Azure SQL DACPAC Deployment Task.

**Example:** Deploying Database Changes

```
- task: SqlAzureDacpacDeployment@1
```

inputs:

```
azureSubscription: 'MyAzureServiceConnection'
```

```
serverName: 'mydb.database.windows.net'
```

```
databaseName: 'mydb'
```

```
dacpacFile: '$(Build.ArtifactStagingDirectory)/database.dacpac'
```

## 60. How do you configure CI/CD for mobile applications using Azure DevOps?

**Answer:**

Use Azure DevOps + App Center for mobile CI/CD.

**Best Practices:**

Use Xcode & Android Gradle build tasks.

Automate APK & IPA signing.

Deploy to Google Play or App Store via App Center.

**Example:** Android Build Pipeline in

```
trigger:
```

```
- main
```

```
pool:
```

```
vmImage: 'macOS-latest'
```

```
steps:
```

```
- task: Gradle@2
```

inputs:

```
gradleWrapperFile: 'gradlew'
```

```
gradleOptions: '-Xmx3072m'
```



```
tasks: 'assembleRelease'
```

## 61. How do you handle pipeline failures and retry failed jobs automatically?

### Answer:

Use retry strategies and pipeline conditions.

### Best Practices:

Use "dependsOn" with "condition: failed()" for retries.

Implement backoff logic in scripts.

**Example:** Auto-Retry on Failure in

```
jobs:
```

```
- job: Deploy
```

```
  steps:
```

```
    - script: |
```

```
      echo "Starting deployment..."
```

```
      exit 1 # Simulating failure
```

```
      continueOnError: true
```

```
- job: RetryDeploy
```

```
  dependsOn: Deploy
```

```
  condition: failed()
```

```
  steps:
```

```
    - script: echo "Retrying deployment..."
```

## 62. How do you implement SLA (Service Level Agreement) compliance in Azure CI/CD?

### Answer:

Use Azure Monitor for pipeline uptime tracking.

Set deployment success rate thresholds.

Implement self-healing mechanisms (e.g., auto-retries).



### 63. How do you set up A/B testing deployment in Azure DevOps?

**Answer:**

Use Azure App Service Slots + Azure Front Door for traffic split.

**Best Practices:**

Route 50% of users to New Version (B) and the rest to Stable Version (A).

Gradually increase B variant exposure.

**Example:** Traffic Splitting Using Azure App Service Slots

```
- task: AzureAppServiceManage@0
```

```
inputs:
```

```
  action: 'Set Traffic Routing'
```

```
  appName: 'my-app'
```

```
  trafficWeight: '50'
```

### 64. How do you secure Azure Pipelines against insider threats?

**Answer:**

Implement Role-Based Access Control (RBAC).

Restrict Service Connections & Secret Variables.

Use Azure DevOps Audit Logs for tracking access.

### 65. How do you integrate Terraform with Azure Pipelines for Infrastructure as Code (IaC)?

**Answer:**

Use Terraform Task in Azure Pipelines.

Store Terraform state files in Azure Storage.

Implement terraform plan & apply stages.

**Example:** Terraform Apply in



```
- task: TerraformTaskV1
```

```
inputs:
```

```
command: 'apply'
```

```
workingDirectory: '$(Build.SourcesDirectory)/terraform'
```

```
environmentServiceName: 'MyAzureServiceConnection'
```

## 66. How do you implement continuous deployment for serverless applications using Azure Pipelines?

### Answer:

Use Azure Functions + Azure Pipelines.

Deploy serverless code using ZIP Deploy or ARM Templates.

**Example:** Deploy Serverless Function App

```
- task: AzureFunctionApp@1
```

```
inputs:
```

```
azureSubscription: 'MyAzureServiceConnection'
```

```
appName: 'my-function-app'
```

```
package: '$(Build.ArtifactStagingDirectory)/function.zip'
```

## 67. How do you implement Blue-Green deployments in Kubernetes using Azure DevOps?

### Answer:

Use Kubernetes Rolling Updates + Traffic Management.

**Example:** Kubernetes Rolling Update Deployment

```
- script: |
```

```
kubectl set image deployment/my-app my-app=myregistry.azurecr.io/myapp:v2
```

## 68. How do you integrate Azure Key Vault secrets into CI/CD pipelines?

### Answer:

Use AzureKeyVault task in pipelines.



**Example:** Fetch Secret from Key Vault

```
- task: AzureKeyVault@1
```

```
inputs:
```

```
azureSubscription: 'MyAzureServiceConnection'
```

```
KeyVaultName: 'myKeyVault'
```

```
SecretsFilter: '*'
```

## 69. How do you set up CI/CD for AI/ML models using Azure DevOps?

**Answer:**

Use Azure Machine Learning SDK for training & deployment.

Automate model retraining and publishing.

**Example:** Deploy ML Model Using Azure ML Task

```
- script: |
```

```
az ml model deploy -n myMLModel --model myModel.pkl --compute myAksCluster
```

## 70. How do you automate rollback in Azure DevOps Pipelines?

**Answer:**

Use previous build artifacts for rollback.

Configure deployment health checks.

**Example:** Rollback Deployment to Last Successful Version

```
- script: |
```

```
kubectl rollout undo deployment my-app
```

## 71. How do you integrate Azure DevOps with GitHub Actions for CI/CD?

**Answer:**

You can integrate GitHub Actions with Azure DevOps to trigger pipelines when a new commit or PR is made.





**Best Practices:**

Use GitHub Webhooks to trigger Azure Pipelines.

Use Service Connections to link GitHub and Azure DevOps.

Use Azure CLI in GitHub Actions to interact with Azure.

**Example:** Triggering Azure Pipelines from GitHub Actions

```
name: GitHub to Azure DevOps CI/CD
on: [push, pull_request]
jobs:
  trigger_azure_pipeline:
    runs-on: ubuntu-latest
    steps:
      - name: Trigger Azure Pipeline
        run: |
          curl -u "user:token" -X POST
            "https://dev.azure.com/org/project/_apis/pipelines/{pipelineId}/runs?api-version=6.0"
```

**72. How do you implement feature flags in Azure DevOps for CI/CD?****Answer:**

Feature flags enable controlled feature rollouts without redeploying code.

**Steps to Implement:**

Use Azure App Configuration to store feature flags.

Integrate feature flags in application code.

Use feature toggles to enable/disable features dynamically.

**Example:** Using Feature Flags in .NET Core

```
var featureFlag = config["EnableNewFeature"];
if (featureFlag == "true")
{
    Console.WriteLine("New Feature Enabled!");
}
```



### 73. How do you implement observability in Azure Pipelines?

**Answer:**

Observability helps track pipeline performance, errors, and failures using monitoring tools.

**Best Practices:**

Use Azure Monitor + Log Analytics.

Enable Application Insights for real-time telemetry.

Configure alerts for pipeline failures.

**Example:** Enable Logging for Azure Pipelines

```
- task: AzureCLI@2
```

```
  inputs:
```

```
    scriptType: "
```

```
    scriptLocation: 'inlineScript'
```

```
    inlineScript: 'az monitor log-profiles create --name "PipelineLogs" --categories "All" --locations "global"'
```

### 74. How do you handle disaster recovery (DR) for Azure CI/CD Pipelines?

**Answer:**

DR ensures business continuity in case of Azure DevOps outages.

**Best Practices:**

Use Azure DevOps Backup & Restore.

Store pipeline configurations in Git repositories.

Replicate artifacts and source code across multiple regions.

### 75. How do you configure high availability (HA) for self-hosted agents in Azure DevOps?

**Answer:**

Use multiple self-hosted agents across regions.

Deploy agents on Kubernetes for auto-scaling.

Use Azure Virtual Machine Scale Sets (VMSS).



**Example:** Scaling Self-Hosted Agents on Kubernetes

- task: Kubernetes@1

inputs:

command: 'scale'

arguments: '--replicas=5 deployment/azure-agent'

## 76. How do you configure CI/CD for Azure API Management (APIM)?

**Answer:**

Use ARM templates or Bicep to manage APIM resources.

Automate API deployments using Azure DevOps Pipelines.

Use APIM policies for security and rate-limiting.

**Example:** Deploy API to APIM using Azure CL

- task: AzureCLI@2

inputs:

scriptType: "

scriptLocation: 'inlineScript'

inlineScript: 'az apim api import --resource-group myRG --service-name myAPIM --path /myapi --specification-url https://myapi.swagger.json'

## 77. How do you automate versioning for artifacts in Azure DevOps?

**Answer:**

Use Build IDs or Git Tags in versioning.

Use Semantic Versioning (1.0.0 format).

Configure versioning in package.json, pom.xml, etc..

**Example:** Auto-Increment Build Version

name: \$(Build.DefinitionName)\_\$(Build.BuildId)



## 78. How do you configure pipeline dependency management in Azure DevOps?

### Answer:

Use Pipeline Triggers to run dependent pipelines.

Define artifacts as dependencies between stages.

**Example:** Triggering a Dependent Pipeline in

resources:

pipelines:

- pipeline: BuildPipeline

source: Build-CI

trigger:

branches:

include:

- main

## 79. How do you enforce security compliance in CI/CD pipelines?

### Answer:

Use Azure Policy to enforce compliance rules.

Run security scans (OWASP, SonarQube).

Implement pre-deployment checks.

## 80. How do you automate load testing in Azure Pipelines?

### Answer:

Use Azure Load Testing Service.

Run JMeter or k6 tests in pipelines.

**Example:** Running JMeter Load Tests in CI/CD

- task: JMeter@1

inputs:

testPlanFile: 'tests/load\_test.jmx'



## 81. How do you manage multi-cloud CI/CD deployments using Azure Pipelines?

### Answer:

Multi-cloud CI/CD deployments involve deploying applications across Azure, AWS, and GCP.

### Best Practices:

Use Azure DevOps Service Connections for AWS & GCP.

Use Terraform or Ansible for infrastructure provisioning.

Implement cross-cloud monitoring with Prometheus and Grafana.

**Example:** Deploying to AWS from Azure Pipelines

```
- task: AWSCLI@1
```

```
  inputs:
```

```
    awsCredentials: 'AWSServiceConnection'
```

```
    regionName: 'us-east-1'
```

```
    command: 's3'
```

```
    arguments: 'cp myapp.zip s3://my-bucket/'
```

## 82. How do you identify and fix performance bottlenecks in Azure CI/CD pipelines?

### Answer:

Enable pipeline caching to speed up builds.

Use parallel jobs for faster execution.

Optimize deployment steps (reduce unnecessary builds & tests).

Monitor pipeline execution time using Azure Monitor.

**Example:** Caching npm Dependencies in YAML

```
- task: Cache@2
```

```
  inputs:
```

```
    key: 'npm | package-lock.json'
```

```
    path: $(npm_config_cache)
```



### 83. How do you integrate security scanning with Azure Pipelines?

**Answer:**

Use Microsoft Defender for Cloud for security scanning.

Use OWASP ZAP or SonarQube for application security.

Scan container images with Trivy before deployment.

**Example:** Running a Container Security Scan in YAML

```
- script: |  
  trivy image myregistry.azurecr.io/myapp:latest
```

### 84. How do you enable audit logging in Azure DevOps?

**Answer:**

Enable Azure DevOps Audit Logs in Organization Settings.

Use Log Analytics to track pipeline execution history.

Configure Azure Monitor alerts for suspicious activities.

**Example:** Enable Logging for Azure DevOps Pipelines

```
- task: AzureCLI@2  
  inputs:  
    scriptType: "  
    scriptLocation: 'inlineScript'  
    inlineScript: 'az monitor diagnostic-settings create --name "PipelineLogs" --categories "All"  
--locations "global"'
```

### 85. How do you migrate CI/CD pipelines between Azure DevOps organizations?

**Answer:**

Export YAML pipeline configurations from the source organization.

Recreate Service Connections in the target organization.

Re-import repositories and artifacts.



**Example:** Export and Import Pipeline Using REST API

```
curl -X GET "https://dev.azure.com/{source_org}/_apis/build/definitions?api-version=6.0" -  
H "Authorization: Bearer {PAT}"
```

```
curl -X POST "https://dev.azure.com/{target_org}/_apis/build/definitions?api-version=6.0" -  
H "Authorization: Bearer {PAT}" -d @pipeline.json
```

## 86. How do you use Azure Blueprints for CI/CD governance?

**Answer:**

Azure Blueprints help enforce compliance and governance across multiple environments.

**Best Practices:**

Define approved infrastructure templates.

Assign Blueprints automatically during deployment.

Use Azure Policy to enforce security rules.

**Example:** Assigning an Azure Blueprint Using Azure CLI

```
az blueprint assignment create \  
--name MyBlueprintAssignment \  
--location eastus \  
--identity-type SystemAssigned \  
--blueprint-name MyBlueprint \  
--subscription {subscriptionId}
```

## 87. How do you handle pipeline secrets across multiple environments?

**Answer:**

Use Azure Key Vault to manage secrets securely.

Configure environment-specific secrets.

Restrict access using RBAC (Role-Based Access Control).



**Example:** Fetching Secrets from Key Vault in Azure Pipelines

```
- task: AzureKeyVault@1
```

```
inputs:
```

```
azureSubscription: 'MyAzureServiceConnection'
```

```
KeyVaultName: 'myKeyVault'
```

```
SecretsFilter: '*'
```

**88. How do you implement shift-left security testing in CI/CD pipelines?**

**Answer:**

Shift-left security means integrating security testing early in the CI/CD pipeline.

**Best Practices:**

Use SAST (Static Application Security Testing) for code analysis.

Use DAST (Dynamic Application Security Testing) for runtime analysis.

Implement OWASP ZAP for API security scanning.

**Example:** Running OWASP ZAP Security Scans in CI/CD

```
- script: |
```

```
zap.sh -quickurl https://myapp.com -quickout report.html
```

**89. How do you optimize pipeline costs for large-scale deployments?**

**Answer:**

Use self-hosted agents to reduce Microsoft-hosted agent costs.

Implement pipeline caching to avoid redundant computations.

Use Azure DevOps Pipeline Billing Reports to track spending.

**90. How do you configure data backup and recovery in Azure Pipelines?**

**Answer:**

Use Azure Backup for VM-based CI/CD agents.

Enable Azure Blob Storage snapshots for artifact backups.





Store pipeline configurations in Git repositories.

### 91. How do you manage regulatory compliance (HIPAA, GDPR) in CI/CD?

**Answer:**

Use Azure Policy to enforce compliance rules.

Encrypt sensitive data using Azure Key Vault.

Implement audit logging for security tracking.

### 92. How do you integrate Terraform state management in CI/CD pipelines?

**Answer:**

Store Terraform state in Azure Storage for consistency.

Use Terraform Cloud for team collaboration.

**Example:** Configuring Terraform Backend in Azure Storage

```
terraform {  
  backend "azurerm" {  
    storage_account_name = "mystorageaccount"  
    container_name       = "terraformstate"  
    key                  = "prod.tfstate"  
  }  
}
```

### 93. How do you implement rollback strategies in YAML-based Azure Pipelines?

**Answer:**

Store previous successful builds as artifacts.

Use Azure App Service Deployment Slots for rollback.

**Example:** Rollback a Deployment in YAML

```
- stage: Rollback  
  dependsOn: Deploy
```



```
condition: failed()
```

```
jobs:
```

```
- job: RollbackApp
```

```
steps:
```

```
- script: echo "Rolling back deployment"
```

#### 94. How do you automate database schema migrations in CI/CD?

##### Answer:

Use Entity Framework Migrations for .NET apps.

Automate SQL schema deployment using DACPAC in Azure Pipelines.

**Example:** Running EF Migrations in Azure Pipelines

```
- script: dotnet ef database update
```

#### 95. How do you secure containerized applications in CI/CD?

##### Answer:

Use Trivy or Microsoft Defender to scan containers.

Implement Kubernetes RBAC for security control.

Use Azure Policy to enforce container security baselines.

#### 96. How do you handle pipeline failures in air-gapped environments?

##### Answer:

Use self-hosted agents in private networks.

Store artifacts in offline package repositories.

Use Azure DevOps YAML pipelines with local dependencies.

#### 97. How do you implement canary releases in Azure Pipelines?

##### Answer:

Use Azure Traffic Manager for traffic control.



Deploy a new version to a subset of users.

Gradually shift traffic based on performance monitoring.

**Example:** Canary Deployment Using App Service Slots

```
- task: AzureAppServiceManage@0
```

```
  inputs:
```

```
    action: 'Set Traffic Routing'
```

```
    appName: 'my-app'
```

```
    trafficWeight: '10'
```

## 98. How do you configure pipeline notifications in Microsoft Teams?

**Answer:**

Use Azure DevOps Webhooks to send messages to Teams.

Integrate Teams Connector with Azure DevOps Pipelines.

## 99. How do you enforce deployment freeze windows in CI/CD pipelines?

**Answer:**

Use Pipeline Conditions to restrict deployments during specific times.

Implement Azure DevOps Deployment Gates to enforce freeze periods.

**Example:** Prevent Deployment During Weekends

```
condition: and(succeeded(), not(contains(variables['Build.SourceBranch'],  
'refs/heads/main')), not(contains(variables['System.DayOfWeek'], 'Saturday|Sunday')))
```

## 100. How do you optimize deployment strategies for serverless applications?

**Answer:**

Use Azure Functions + Zip Deploy for fast deployments.

Implement rolling updates for seamless function upgrades.

Use Feature Flags for gradual rollouts.

