1. Develop Azure Compute Solutions

└ 1.1 Implement Containerized Solutions

└ 1.1.1 Create and Manage Container Images for Solutions

1. What is the correct structure of a Dockerfile?
2. How do you optimize a Dockerfile for size and performance?
3. How do multi-stage builds work?
4. What commands are used to build an image locally?
5. How do you tag versions appropriately?
6. What is the correct format for image names (registry/repository:tag)?
7. How do tags work and how are they used in CI/CD pipelines?
8. How do you create and configure an ACR?
9. What are the SKU tiers and when do you use them
10. How do you push/pull images using Docker CLI?
11. How do you authenticate to ACR (admin account, service principal, managed identity)?
12. What is az acr login and when is it required?
13. How does Azure App Service, AKS and ACI consume images from ACR?
14. What are the permission requirements for pulling from ACR?
15. How do you scan images for vulnerabilities?
16. What tools or services are used to harden container images?
17. How do you manage image versions across environments?
18. How do you clean up unreferenced or old images in ACR?
19. How can ACR tasks automate image builds?
20. How do you use az acr task to create scheduled or event-driven builds?
21. What are the pros and cons of storing images publicly vs privately?
22. How do you configure access control for image repositories?

---

1. What is the correct structure of a Dockerfile?

```
FROM <base_image>
[ENV <key>=<value> ...]
[WORKDIR <directory>]
[COPY <src> <dest>]
[RUN <command>]
[EXPOSE <port>]
[CMD ["executable", "param1", ...]]
```

Key Points:

- FROM must be first (defines base image).
- WORKDIR sets working directory inside container.
- COPY adds files to the image from build context.
- RUN executes shell commands (e.g., install packages).
- EXPOSE is optional metadata; doesn't actually open ports.
- CMD defines default container startup command (only one allowed; last one wins).

2. How do you optimize a Dockerfile for size and performance?

1. Use slim/minimal base images:
   Prefer FROM mcr.microsoft.com/dotnet/aspnet:7.0-alpine over full images.
3. Leverage multi-stage builds:
   Build in one stage, copy only final output to runtime stage to reduce size.
4. Minimize layers:
   Group related RUN commands and clean up temp files in the same layer:
4. Avoid unnecessary files:
   Use .dockerignore to exclude files (e.g., .git, node_modules).
5. Set only needed environment variables and permissions:
   Avoid excessive ENV or USER changes unless required.

---

3. How do multi-stage builds work?

Reduce final image size by separating build and runtime stages.

- Define multiple FROM statements in one Dockerfile.
- Use an alias for the build stage (AS build).
- Copy only needed artifacts from the build stage into the final image.

---

4. What commands are used to build an image locally?

docker build -t <name>:<tag> <path>

e.g. docker build -t myapp:latest .

---

5. How do you tag versions appropriately?

Use semantic tags like:

latest, v1.0.0, dev, staging

e.g. docker build -t myapp:v1.0.0

Tag meaning should reflect version or environment for clarity and traceability.

---

6. What is the correct format for image names?

<registry>/<repository>:<tag> *:*

myregistry.azurecr.io/myapp:v1.0.0

Registry is optional for local images. Tag defaults to latest if omitted.

---

7. How do tags work and how are they used in CI/CD pipelines?

Tags identify image versions. Pipelines use tags to pull, test, and deploy specific builds:

docker push myapp:staging → used in staging environment.

latest often used in dev, versioned tags in prod.

---

8. How do you create and configure an ACR?

- Create ACR: az acr create --name <acr-name> --resource-group <rg> --sku Basic
- Enable admin access: az acr update -n <acr-name> --admin-enabled true
- Login: az acr login --name <acr-name>

---

9. What are the SKU tiers and when do you use them?

- Basic – Dev/test, low-cost, limited features
- Standard – Prod-ready, geo-replication support
- Premium – High-scale, content trust, private endpoints, more throughput

10. How do you push/pull images from/to ACR using Docker CLI?

Push:

1. docker tag myapp myacr.azurecr.io/myapp
2. docker push myacr.azurecr.io/myapp

Pull:

docker pull myacr.azurecr.io/myapp

---

11. How do you authenticate to ACR (admin, service principal, managed identity)?

- Admin account:
  Enable with az acr update --admin-enabled true, then use provided username/password.
- Service principal:
  Assign AcrPush/AcrPull role, login with docker login using SP credentials.
- Managed identity:
  Grant role to identity, Azure services (e.g., App Service) authenticate automatically.

---

12. What is az acr login and when is it required?

az acr login --name <acr-name>

Authenticates Docker CLI with ACR.
Required for manual Docker pushes/pulls. Not needed for Azure services using managed identity.

---

13. How do App Service, Kubernetes Service (AKS), and Container Instances consume ACR images?

- App Service: Configure container settings with ACR URL; use managed identity or admin acc.
- AKS: Enable ACR integration via az aks update or use imagePullSecrets.
- ACI: Reference image with full ACR path; grant access via identity or admin credentials.

---

14. What are the permission requirements for pulling from ACR?

The identity must have **AcrPull** role on the ACR.
Can be assigned to:

- User
- Service principal
- Managed identity (App Service, AKS, etc.)

---

15. How do you scan images for vulnerabilities?

Use Microsoft Defender for Cloud with ACR integration. It scans images on push and shows CVEs in the portal.
Enable under Defender plans > Container registries.

---

16. What tools or services are used to harden container images?

- Microsoft Defender for Cloud – vulnerability scanning
- Dockerfile best practices – minimize layers, use minimal base images
- Content trust – ensure image integrity
- Private ACR – restrict access
- ACR Tasks – automate secure builds

---

17. How do you manage image versions across environments?

Use consistent tagging strategy (e.g., dev, staging, v1.0.0).
Promote images by re-tagging and pushing to ACR for each stage.

18. How do you clean up unreferenced or old images in ACR?

Use ACR Tasks with retention policies or manual cleanup via:

```
az acr repository delete --name <acr> --image <repo>:<tag>
```

---

19. How can ACR tasks automate image builds?

ACR Tasks can auto-build images on source code or base image changes:

`az acr task create` with `--source` and `--cmd "docker build"`

Supports triggers (e.g., Git push) and scheduling.

---

20. How do you use az acr task to create scheduled or event-driven builds?

Event-driven:

```
az acr task create \
  --name mytask \
  --registry myacr \
  --image myapp:{{.Run.ID}} \
  --context https://github.com/org/repo.git \
  --file Dockerfile \
  --git-access-token <token>
```

Scheduled:

```
az acr task create \
  --name mytask \
  --registry myacr \
  --schedule "0 2 * * *" \
  --image myapp:nightly \
  --context https://github.com/org/repo.git \
  --file Dockerfile
```

---

21. What are the pros and cons of storing images publicly vs privately?

Public: Easy access, no auth needed — but insecure, no access control.

Private: Secure, controlled access — but needs auth, may cost more.

---

22. How do you configure access control for image repositories?

Assign `AcrPull` or `AcrPush` roles to users, service principals, or managed identities using Azure RBAC on the ACR resource.

Develop Azure Compute Solutions

└ 1.1 Implement Containerized Solutions

└ 1.1.2 Publish an image to Azure Container Registry

1. What is Azure Container Registry?
2. What are the core capabilities and use cases?
3. How does managed identity authentication work for ACR?
4. How do you authenticate ACR access in CI/CD workflows (e.g., GitHub Actions, Azure DevOps)?
5. How do you list repositories and images in ACR?
6. How do you delete/update images using Azure CLI or Portal?
7. How do you validate if a service (App Service, AKS, ACI) is authorized to pull from ACR?
8. What happens if permissions are missing?
9. How do you use ACR with GitHub Actions or Azure Pipelines?
10. What are best practices for secure build pipelines pushing to ACR?
11. How do you configure image signing and content trust in ACR?
12. What security best practices apply to private ACR use?

---

1. What is Azure Container Registry (ACR)?

A managed, private Docker registry in Azure used to store container images and artifacts like Helm charts. Supports geo-replication, image signing, ACR Tasks, and CI/CD integration.

---

2. What are the core capabilities and use cases?

Capabilities:

- Private image storage with RBAC
- ACR Tasks for build automation
- Integration with App Service, AKS, ACI
- Image scanning with Defender for Cloud

Use Cases:

- Hosting internal images
- Serving images to AKS
- Automating builds/pushes
- Enforcing image compliance

---

3. How does managed identity authentication work for ACR?

Azure services (e.g., App Service, AKS) use their managed identity to access ACR. You assign the `AcrPull` role to the identity on the ACR resource—no credentials needed.

---

4. How do you authenticate ACR access in CI/CD workflows?

- GitHub Actions: Use `azure/login` and `docker/login-action` with a service principal or OIDC.
- Azure DevOps: Use built-in service connection or Docker task with a service principal. Ensure the principal has AcrPush or AcrPull as needed.

---

5. How do you list repositories and images in ACR?

Use Azure CLI:

- List repos: `az acr repository list --name <acr-name>`
- List tags/images: `az acr repository show-tags --name <acr-name> --repository <repo>`

6. How do you delete or update images in ACR?

- Delete image: az acr repository delete --name <acr-name> --image <repo>:<tag>
- To update, re-tag and push a new version.
  Use retention policies or manual cleanup to manage old images.

---

7. How do you validate if a service is authorized to pull from ACR?

Check that the service's managed identity has the AcrPull role on the ACR.
Verify access using:

```
az role assignment list --assignee <identity> --scope <acr-resource-id>
```

---

8. How do you use ACR with GitHub Actions or Azure Pipelines?

GitHub Actions:

- Authenticate with azure/login and docker/login-action
- Push using docker/build-push-action

Azure Pipelines:

- Use Docker or container tasks
- Authenticate via service connection

Both require proper role assignment (e.g., AcrPush).

---

9. What happens if permissions are missing?

The service (e.g., App Service, AKS, ACI) will fail to start or pull the image, typically with an authentication or 403 error in logs.

---

10. What are best practices for secure build pipelines pushing to ACR?

- Use service principals or workload identity (OIDC)
- Grant least privilege (only AcrPush)
- Avoid storing secrets in plain text; use Key Vault or pipeline secrets
- Enable image signing and scanning post-push

---

11. How do you configure image signing and content trust in ACR?

Enable content trust to verify image integrity:

- Use Docker's DOCKER_CONTENT_TRUST=1 for signing and verification
- For ACR, use Microsoft Defender for Containers to enforce policies

---

12. What security best practices apply to private ACR use?

- Use private endpoints or firewall rules
- Disable anonymous pull access
- Require authentication via managed identity or service principal
- Enable Defender for vulnerability scanning
- Assign roles using RBAC (AcrPull/AcrPush only as needed)

Develop Azure Compute Solutions

└ **1.1 Implement Containerized Solutions**

└ **1.1.3 Run Containers by Using Azure Container Instances (ACI)**

1. What is Azure Container Instances (ACI) and when should you use it?
2. How do you create and deploy a container using ACI via Azure CLI?
3. What are the key configuration parameters for az container create?
4. How do you pull images from Azure Container Registry to ACI?
5. How does managed identity authentication work with ACI?
6. How do you assign and verify roles for ACI to access ACR?
7. What are the options for exposing containers to the internet or VNETs in ACI?
8. How do you mount Azure Files or secrets (Key Vault) into containers?
9. How do you monitor logs, metrics, and container status in ACI?
10. What are the restart policies and lifecycle options in ACI?
11. How do you use YAML to define ACI deployments?
12. What are common use cases and limitations of ACI?

---

1. What is Azure Container Instances (ACI) and when should you use it?

ACI is a serverless container platform allowing fast, isolated container runs without VM management.
Use cases:

- Short-lived jobs or batch processing
- Event-driven container execution
- Lightweight API hosting without orchestration overhead

---

2. How do you create and deploy a container using ACI via Azure CLI?

```
az container create \
    --resource-group <rg> \
    --name <container-name> \
    --image <image-name> \
    --cpu 1 \
    --memory 1 \
    --restart-policy OnFailure \
    --dns-name-label <unique-label> \
    --ports 80
```

This deploys a public-facing container running on port 80.

---

3. What are the key configuration parameters for az container create?

--image: Container image to run (e.g., from ACR or Docker Hub)
--cpu / --memory: Resource limits
--environment-variables: Inject app settings
--ports: Exposed ports
--dns-name-label: For public IP
--restart-policy: Options: Always, OnFailure, Never
--vnet and --subnet: Attach to virtual network
--secrets and --secrets-mount-path: Mount secrets

4. How do you pull images from Azure Container Registry to ACI?

ACI can pull private images from ACR by granting ACI's managed identity the **AcrPull** role on the ACR.

Ensure image format is:

```
<acr-name>.azurecr.io/<repository>:<tag>
```

Example ACI deployment with an image from ACR:

```
az container create \
  --name <container-name> \
  --resource-group <rg> \
  --image <acr-name>.azurecr.io/app:latest \
  --registry-login-server <acr-name>.azurecr.io \
  --assign-identity \
  --cpu 1 --memory 1
```

---

5. How does managed identity authentication work with ACI?

ACI supports user-assigned and system-assigned managed identities.

Steps:

1. Enable managed identity with --assign-identity.
2. Assign AcrPull role to the identity at ACR scope.
3. ACI uses this identity to authenticate and pull private images—no credentials needed.

---

6. How do you assign and verify roles for ACI to access ACR?

Use Azure CLI to assign roles:

```
az role assignment create \
  --assignee <principal-id> \
  --role AcrPull \
  --scope /subscriptions/<sub-id>/resourceGroups/<rg>/providers/Microsoft.ContainerRegistry/registries/<acr-name>
```

Verify with:

```
az role assignment list --assignee <principal-id> --scope <acr-resource-id>
```

---

7. What are the options for exposing containers to the internet or VNETs in ACI?

- Public IP (default): Use --dns-name-label and --ports to expose over the internet.
- Private IP (VNET): Use --vnet & --subnet to deploy into a virtual network for internal-only access.

ACI supports:

- Inbound public access
- Private IP in VNET (for secure inter-service traffic)
- No ingress (headless jobs)

---

8. How do you mount Azure Files or secrets (Key Vault) into containers?

Mount Azure Files:

```
az container create \
  --azure-file-volume-share-name <share> \
  --azure-file-volume-account-name <storage-account> \
  --azure-file-volume-account-key <key> \
  --azure-file-volume-mount-path /mnt/data
```

Mount Key Vault secrets:

```
az container create \
  --secrets key1=value1 key2=value2 \
  --secrets-mount-path /mnt/secrets
```

9. How do you monitor logs, metrics, and container status in ACI?

View logs:

```
az container logs --name <container-name> --resource-group <rg>
```

Get status:

```
az container show --name <container-name> --resource-group <rg> --query instanceView.state
```

---

10. What are the restart policies and lifecycle options in ACI?

Available values for --restart-policy:

- Always: Container restarts on exit (default).
- OnFailure: Restarts only on non-zero exit code.
- Never: One-time execution, used for jobs.

Lifecycle:

- No native job scheduling—combine with Logic Apps, Functions, or Event Grid for automation.
- ACI auto-deletes after manual az container delete or TTL implementation logic.

---

11. How do you use YAML to define ACI deployments?

Example aci.yaml:

```
apiVersion: 2018-10-01
location: eastus
name: mycontainer
properties:
 containers:
  - name: myapp
    properties:
     image: myacr.azurecr.io/myapp:latest
     resources:
       requests:
         cpu: 1
         memoryInGb: 1.5
     ports:
       - port: 80
 osType: Linux
 restartPolicy: OnFailure
 ipAddress:
  type: Public
  dnsNameLabel: mycontainerdemo
  ports:
    - port: 80
type: Microsoft.ContainerInstance/containerGroups
```

Deploy with:

```
az deployment group create \
  --resource-group <rg> \
  --template-file aci.yaml
```

---

12. What are common use cases and limitations of ACI?

Use Cases:

- Lightweight API/backend services
- Batch jobs
- Temporary compute (build/test)
- Event-driven processing

Limitations:

- No built-in autoscaling
- No service mesh or ingress controller
- Not suited for complex orchestration—use AKS instead

Develop Azure Compute Solutions

└ 1.1 Implement Containerized Solutions

└ 1.1.4 Create solutions by using Azure Container Apps

1. What is Azure Container Apps and when should it be used over AKS or App Services?
2. What components define an Azure Container App?
3. How do revisions work in Azure Container Apps?
4. What are the prerequisites for deploying a container to Azure Container Apps?
5. How do you deploy a container from Azure Container Registry using Azure CLI?
6. How is a YAML manifest used to deploy a container app?
7. How do you configure ingress and expose ports in Azure Container Apps?
8. How do you configure authentication for private container registries?
9. How are environment variables added to a container app?
10. How are secrets stored and injected into container apps?
11. What is KEDA and how does it apply to Azure Container Apps?
12. What scaling rules are supported in Azure Container Apps?
13. How do you configure HTTP-based autoscaling?
14. How do you configure scaling based on Azure Service Bus or Azure Queue Storage?
15. What are minReplicas and maxReplicas and how are they configured?
16. How does revision mode affect app behavior in Azure Container Apps?
17. How is traffic splitting configured across revisions?
18. How do you perform A/B testing using revisions?
19. How do you roll back to a previous revision?
20. What is Dapr and how is it used with Azure Container Apps?
21. What Dapr capabilities are supported in Azure Container Apps?
22. How do you enable and configure the Dapr sidecar?
23. How do container apps communicate using Dapr?
24. How is state management and pub/sub handled with Dapr?
25. What monitoring and logging features are built into Azure Container Apps?
26. How do you enable and access diagnostics logs?
27. How do you view application logs and container output?
28. What tools can be used to troubleshoot container app issues?
29. How does integration with Azure Monitor and Log Analytics work?
30. How do you integrate Azure Container Apps with Event Grid or Service Bus?
31. How do you connect a container app to Azure Storage queues?
32. How do you securely access Azure services from a container app?
33. How do you use managed identities in Azure Container Apps?
34. What role does VNET integration play in accessing private resources?

---

1. What is Azure Container Apps and when should it be used over AKS or App Services?

Azure Container Apps is a fully managed serverless container service for microservices, APIs, and background processing. Use it when:

- You need event-driven or HTTP-based workloads
- You prefer serverless scaling (including scale to zero)
- You want built-in Dapr and KEDA support without managing orchestration

Prefer AKS for orchestration or App Services for traditional web apps or minimal container needs.

---

2. What components define an Azure Container App?

- Container App: The deployed app instance
- Environment: A shared context for apps (networking, logging)
- Revision: An immutable version of the app
- Ingress: Controls public/private HTTP access
- Scaling Rules: Define autoscaling behavior (HTTP, KEDA, etc.)

---

3. How do revisions work in Azure Container Apps?

- Each deployment creates a new immutable revision
- Revisions can run concurrently
- Traffic can be split between revisions
- You can pin a revision or roll back
- Revision mode can be single (default) or multiple

---

4. What are the prerequisites for deploying a container to Azure Container Apps?

- A container image in ACR or public registry
- A Container Apps environment
- Azure CLI with the containerapp extension installed
- App image must expose the correct HTTP port.

---

5. How do you deploy a container from Azure Container Registry using Azure CLI?

```
az containerapp create \
  --name myapp \
  --resource-group myrg \
  --environment myenv \
  --image myacr.azurecr.io/myimage:tag \
  --target-port 80 \
  --ingress external \
  --registry-server myacr.azurecr.io \
  --registry-username <username> \
  --registry-password <password>
```

---

6. How is a YAML manifest used to deploy a container app?

Define app configuration in a .yaml file (image, ports, scaling, secrets, etc.)

- Deploy using:
  `az containerapp create --resource-group myrg --name myapp --yaml app.yaml`

---

7. How do you configure ingress and expose ports in Azure Container Apps?

- Use `--ingress external` or `internal` in CLI or `ingress:` block in YAML
- Set targetPort to match container's exposed port
- Public ingress automatically provisions HTTPS endpoint

---

8. How do you configure authentication for private container registries?

- Use `--registry-username` and `--registry-password` in CLI
- In YAML:

```
registryCredentials:
  - server: myacr.azurecr.io
    username: <username>
    passwordSecretRef: acr-password
```

- Store password as a secret and reference it

9. How are environment variables added to a container app?

- In CLI:
  --env-vars VAR1=value1 VAR2=value2
- In YAML:

```
env:
 - name: VAR1
   value: value1
 - name: VAR2
   value: value2
```

---

10. How are secrets stored and injected into container apps?

- Define secrets in CLI
  --secrets key1=value1 key2=value2
- or YAML:

```
secrets:W
 - name: key1
   value: value1
```

- Reference in env vars:

```
env:
 - name: SEC_VAR
   secretRef: key1
```

---

12. What is KEDA and how does it apply to Azure Container Apps?

KEDA (Kubernetes Event-driven Autoscaler) enables event-based scaling. In Azure Container Apps, it's integrated to scale apps based on metrics like:

- HTTP traffic
- Queue length (e.g., Service Bus, Storage Queues)
- Custom metrics

---

13. What scaling rules are supported in Azure Container Apps?

- HTTP request concurrency
- CPU utilization
- KEDA-based triggers (e.g., Azure Service Bus, RabbitMQ, Redis, Kafka)
- Cron-based schedules

---

14. How do you configure HTTP-based autoscaling?

In YAML:

```
scale:
 rules:
   - name: http-scaler
     http:
       concurrentRequests: 50
```

App will scale based on the number of concurrent HTTP requests.

---

15. What are minReplicas and maxReplicas and how are they configured?

- minReplicas: minimum number of app instances
- maxReplicas: cap on autoscaling
  In YAML:

```
scale:
 minReplicas: 1
 maxReplicas: 10
```

---

16. How do you configure scaling based on Azure Service Bus or Azure Queue Storage?

Define a KEDA trigger in YAML:

```
scale:
  rules:
    - name: sb-scaler
      azureServiceBus:
        queueName: myqueue
        connection: sb-connection
        messageCount: 100
```

*"connection" references a secret holding the Service Bus connection string

---

17. How does revision mode affect app behavior in Azure Container Apps?

- Single revision mode (default): only one revision is active; new deployments kill the previous
- Multiple revision mode: multiple revisions can run concurrently; useful for traffic splitting
- Set via CLI or YAML:

```
revisionMode: multiple
```

---

18. How is traffic splitting configured across revisions?

Assign percentage of traffic to each revision

- In CLI:

```
az containerapp revision set-trafficsplit \
  --name myapp \
  --resource-group myrg \
  --revision-weight revisionA=80 revisionB=20
```

- In YAML:

```
traffic:
 - revisionName: revisionA
   weight: 80
 - revisionName: revisionB
   weight: 20
```

---

19. How do you perform A/B testing using revisions?

- Deploy a new revision in multiple revision mode
- Split traffic between revisions (e.g., 90/10)
- Monitor metrics and logs for both
- Adjust traffic weights or rollback based on results

---

20. How do you roll back to a previous revision?

- Set traffic weight to 100% for the target revision
- Optionally disable the newer revision
- CLI:

```
az containerapp revision set-trafficsplit --name myapp --revision-weight oldrev=100
```

---

21. What is Dapr and how is it used with Azure Container Apps?

Dapr (Distributed Application Runtime) provides building blocks for microservices (e.g., service discovery, state management). Azure Container Apps has built-in Dapr support. Enable by setting daprEnabled: true. No additional setup is required for the Dapr sidecar.

---

22. What Dapr capabilities are supported in Azure Container Apps?

- Service invocation over HTTP/gRPC
- State management (e.g., Redis, Cosmos DB)
- Pub/sub messaging
- Secrets integration
- Middleware and observability tools
  Note: Components are defined via Dapr-compatible configuration files.

---

23. How do you enable and configure the Dapr sidecar?

In YAML:

```
dapr:
  enabled: true
  appId: myapp
  appPort: 80
```

---

24. How do container apps communicate using Dapr?

- Service A calls Service B via http://<appId>.dapr
- Dapr handles service discovery and routing
- Requires both apps to have dapr.enabled: true and unique appId

---

25. What monitoring and logging features are built into Azure Container Apps?

- Integrated Log Streaming via Azure CLI
- Application logs, revision logs, and system logs
- Container stdout/stderr collection
- Azure Monitor and Log Analytics integration for metrics and centralized logging

---

26. How do you enable and access diagnostics logs?

- Enable diagnostics when creating the Container App Environment
- Logs are sent to Azure Monitor (Log Analytics workspace)
- Use Azure CLI:
  `az containerapp logs show --name myapp --resource-group myrg`

---

27. How do you view application logs and container output?

Via Azure CLI:

```
az containerapp logs show --name myapp --follow
```

- Logs include stdout and stderr from the container
- For historical logs, query via Log Analytics using Kusto Query Language (KQL)

---

28. What tools can be used to troubleshoot container app issues?

- `az containerapp` logs show for live logs
- Log Analytics queries for historical data
- Metrics in Azure Monitor (CPU, memory, HTTP throughput)
- Azure CLI/Portal for revision status and health
- Re-deploy with `--debug flag` to get CLI diagnostics

---

29. How does integration with Azure Monitor and Log Analytics work?

- When enabled, diagnostics are sent to a specified Log Analytics workspace
- Use KQL to query logs (e.g., ContainerAppConsoleLogs_CL)
- Metrics surface in Azure Monitor for alerting and dashboarding

30. How do Azure Container Apps integrate with Event Grid or Service Bus?

- Use KEDA triggers to scale based on Event Grid or Service Bus messages
- Event Grid: typically triggers external logic that posts to app HTTP endpoint
- Service Bus: KEDA listens and scales app based on queue/topic message count
- Connection strings are passed as secrets and referenced in scaling rules

---

31. How do you connect a container app to Azure Storage queues?

- Use KEDA with azureQueue scaler
- Define queueName, connection, and queueLength threshold
- Store Storage Account connection string as a secret and reference it in scaling config
- App logic must poll the queue if not using an event trigger

---

32. How do you securely access Azure services from a container app?

- Use **Managed Identity** to authenticate to Azure services like Key Vault, Storage, or SQL
- Avoid hardcoding credentials
- Access tokens are obtained via Azure SDK or HTTP call to IMDS endpoint

---

33. How do you use managed identities in Azure Container Apps?

- Enable system-assigned or user-assigned identity at app level
- Assign proper RBAC role to the identity
- Access Azure services using Azure SDKs with default credential chain
- Example (Azure SDK):
  ```
  from azure.identity import DefaultAzureCredential
  from azure.keyvault.secrets import SecretClient
  ```

---

35. What role does VNET integration play in accessing private resources

Enables access to private endpoints, databases, or internal APIs

- Configure internal ingress and associate the Container Apps environment with a VNET
- Required for scenarios needing outbound traffic restrictions or private-only dependencies

Develop Azure Compute Solutions

└ 1.2 Implement Azure App Service Web Apps

└ 1.2.1 Create an Azure App Service Web App

1. What is Azure App Service and when should it be used?
2. How do you create an App Service Web App using the Azure CLI?
3. How do you create an App Service Plan and what are the pricing tiers?
4. What runtime stacks and OS options are supported?
5. How do you configure deployment credentials?
6. How do you deploy code to App Service (e.g., ZIP, GitHub Actions, Azure DevOps)?
7. How do you configure environment variables and app settings?
8. How do you assign a custom domain and configure HTTPS?
9. What is the role of App Service managed identity?
10. What are the basic scaling options available (manual, autoscale)?

---

1. What is Azure App Service and when should it be used?

A PaaS for hosting web apps and APIs. Supports .NET, Node.js, Python, Java, PHP.
Use when: You need quick deployment, scaling, custom domains, or managed identity.

---

2. How do you create an App Service Web App using Azure CLI?

```
az group create --name myRG --location eastus
az appservice plan create --name myPlan --resource-group myRG --sku B1 --is-linux
az webapp create --resource-group myRG --plan myPlan --name mywebapp123 --runtime "DOTNET|7.0"
```

Creates a Linux-based .NET 7 Web App in Basic tier.

---

3. What is an App Service Plan and how do tiers differ?

Defines compute resources for Web Apps.
Tiers:

- F1 (Free): Shared, limited
- B1 (Basic): Dedicated, no autoscale
- S1 (Standard): Autoscale, staging slots
- P1V3 (Premium): VNET, better scaling
- I1 (Isolated): Private VNET, high compliance

---

4. What OS and runtime stacks are supported?

OS: Linux or Windows
Runtimes: .NET, Node.js, Java, Python, PHP, Ruby, Static HTML
Selected via `--runtime` (e.g., "DOTNET|6.0" or "NODE|18-lts")

---

5. How do you deploy code to Web Apps?

- ZIP Deploy: `az webapp deployment source config-zip`
- GitHub Actions: `az webapp deployment github-actions add`
- Azure DevOps: Use App Service deploy task
- Local Git: Configure via Portal or CLI

6. How do you configure app settings and connection strings?

```
az webapp config appsettings set --name <app-name> --resource-group <rg> --settings KEY=VALUE
az webapp config connection-string set --name <app-name> --resource-group <rg> --settings connStr=... --connection-string-type SQLAzure
```

Stored securely and available as environment variables.

---

7. How do you bind a custom domain and enable HTTPS?

- Add domain via `az webapp config hostname add`
- Enable HTTPS: `az webapp update --https-only true`
- Use App Service-managed certs or upload custom certs

---

8. What is the purpose and setup of managed identity in App Service?

Grants the app access to Azure resources without secrets.

```
az webapp identity assign --name <app-name> --resource-group <rg>
```

Assign roles to this identity for access (e.g., Key Vault).

---

9. How does scaling work in App Service?

- Manual: Set instance count (Standard+)
- Autoscale: Use Azure Monitor rules (CPU, schedule)
  Configured via Portal or ARM/CLI (az monitor autoscale)

---

10. What diagnostic/logging options are available?

- App logs: `az webapp log config`
- Live logs: `az webapp log tail`
- App Insights: Enable via Portal or `az monitor app-insights component create`

Develop Azure Compute Solutions

└ 1.2 Implement Azure App Service Web Apps

└ 1.2.2 Configure and Implement Diagnostics and Logging

1. What types of logs are available in Azure App Service?
2. How do you enable diagnostics logging via CLI?
3. How do you stream logs from App Service in real-time?
4. How do you integrate Application Insights with App Service?
5. What's the difference between App Service logs and App Insights?
6. How do you query logs in Application Insights using KQL?

---

1. What types of logs are available in Azure App Service?

- Application Logs: stdout/stderr from app
- Web Server Logs: HTTP access logs
- Detailed Error Logs: 500-level errors
- Failed Request Tracing Logs (FREB): IIS-level tracing
- App Insights Logs: Telemetry (requests, exceptions, dependencies)

---

2. How do you enable diagnostics logging via CLI?

```
az webapp log config --name <app> --resource-group <rg> \
--application-logging filesystem --web-server-logging filesystem
```

Stores logs in local file system (Basic+) or blob storage.

---

3. How do you stream logs from App Service in real-time?

```
az webapp log tail --name <app> --resource-group <rg>
```

Streams stdout/stderr and App Logs live to console.

---

4. How do you integrate Application Insights with App Service?

```
az monitor app-insights component create --app <name> --location <region> --resource-group <rg>
az webapp config appsettings set --name <app> --resource-group <rg> --settings
APPINSIGHTS_INSTRUMENTATIONKEY=<key>
```

.NET apps may auto-integrate; others need SDK.

---

5. What's the difference between App Service logs and App Insights?

- App Service Logs: Platform logs (stdout, web server logs)
- App Insights: Full telemetry—requests, exceptions, dependencies, metrics, traces

---

6. How do you query logs in Application Insights using KQL?

Use Log Analytics (or Logs tab in App Insights):

Example query:

```
requests
| where timestamp > ago(1h)
| summarize count() by resultCode
```

Develop Azure Compute Solutions

└ 1.2 Implement Azure App Service Web Apps

└ 1.2.3 Deploy Code and Containerized Solutions

1. What deployment options exist for App Service (code and container)?
2. How do you deploy a ZIP package via Azure CLI?
3. How do you configure GitHub Actions deployment?
4. How do you deploy a containerized app to App Service?
5. How do you configure deployment slots and swap?
6. What are best practices for zero-downtime deployments?

---

1. What deployment options exist for App Service (code and container)?

- Code: ZIP Deploy, GitHub Actions, Azure DevOps, FTP, Local Git
- Container: Docker Hub, Azure Container Registry (ACR), custom registry

App type determines supported methods.

---

2. How do you deploy a ZIP package via Azure CLI?

```
az webapp deployment source config-zip \
  --resource-group <rg> \
  --name <app> \
  --src <path-to-zip>
```

---

3. How do you configure GitHub Actions deployment?

```
az webapp deployment github-actions add \
  --repo <user/repo> \
  --branch main \
  --name <app> \
  --resource-group <rg> \
  --login-with-github
```

Generates CI/CD workflow for App Service.

---

4. How do you deploy a containerized app to App Service?

```
az webapp create \
  --resource-group <rg> \
  --plan <plan> \
  --name <app> \
  --deployment-container-image-name <acr-name>.azurecr.io/app:tag
```

Ensure the Web App uses a Linux plan and has AcrPull access.

---

5. How do you configure deployment slots and swap?

- Create slot:

```
az webapp deployment slot create --name <app> --resource-group <rg> --slot staging
```

- Swap slots:

```
az webapp deployment slot swap --name <app> --resource-group <rg> --slot staging
```

Slots isolate staging, testing, and production deployments.

---

6. What are best practices for zero-downtime deployments?

- Use deployment slots
- Run warm-up tests before slot swap
- Enable auto-swap with health checks
- Avoid in-place updates on production slot

Develop Azure Compute Solutions

└ **1.2 Implement Azure App Service Web Apps**

└ **1.2.4 Configure settings including TLS,API, ServiceConnections**

1. How do you enforce HTTPS (TLS) for an App Service?
2. How do you configure minimum TLS version?
3. How do you set custom domains and bind SSL certificates?
4. How do you configure API settings such as CORS?
5. How do you connect to backend services using managed identity?
6. How do you restrict outbound traffic using VNET integration?

---

1. How do you enforce HTTPS (TLS) for an App Service?

```
az webapp update --name <app> --resource-group <rg> --https-only true
```

Redirects all HTTP traffic to HTTPS.

---

2. How do you configure minimum TLS version?

```
az webapp config set --name <app> --resource-group <rg> --min-tls-version 1.2
```

Options: 1.0, 1.1, 1.2. Use 1.2 for compliance.

---

3. How do you set custom domains and bind SSL certificates?

- Add domain:

```
az webapp config hostname add --webapp-name <app> --resource-group <rg> --hostname <custom-domain>
```

- Upload and bind cert:

```
az webapp config ssl upload --certificate-file cert.pfx --certificate-password <pwd> --name <app> --resource-group <rg>
az webapp config ssl bind --name <app> --resource-group <rg> --ssl-type SNI --certificate-thumb
```

---

4. How do you configure API settings such as CORS?

```
az webapp cors add --name <app> --resource-group <rg> --allowed-origins https://example.com
```

Use cors remove or cors show to manage rules.

---

5. How do you connect to backend services using managed identity?

1. Enable identity:

```
az webapp identity assign --name <app> --resource-group <rg>
```

2. Grant role to identity (e.g., Key Vault Reader):

```
az role assignment create --assignee <principal-id> --role Reader --scope <resource-id>
```

---

6. How do you restrict outbound traffic using VNET integration?

- Integrate with VNET (Linux apps, Standard+):

```
az webapp vnet-integration add --name <app> --resource-group <rg> --vnet <vnet-name> --subnet <subnet-name>
```

Restricts outbound traffic and enables access to private services.

Develop Azure Compute Solutions

└ 1.2 Implement Azure App Service Web Apps

└ 1.2.5 Implement Auto-Scaling

1. What scaling options are available in Azure App Service?
2. How do you configure autoscale rules using CLI?
3. What metrics can be used for autoscaling?
4. How do you set instance count manually?
5. What are best practices for autoscaling App Service?

---

1. What scaling options are available in Azure App Service?

- Manual scale: Fixed instance count
- Autoscale: Rule-based via CPU, memory, schedule
  Available in Standard, Premium, Isolated tiers only.

---

2. How do you configure autoscale rules using CLI?

```
az monitor autoscale create \
  --resource <app-service-plan-id> \
  --resource-group <rg> \
  --name autoscale-rule \
  --min-count 1 --max-count 5 --count 2
az monitor autoscale rule create \
  --resource-group <rg> \
  --autoscale-name autoscale-rule \
  --condition "CpuPercentage > 70 avg 5m" \
  --scale out 1
```

---

3. What metrics can be used for autoscaling?

- CPUPercentage
- MemoryPercentage (Premium+ plans)
- HttpQueueLength
- Schedule-based triggers

---

4. How do you set instance count manually?

```
az appservice plan update \
  --name <plan> --resource-group <rg> \
  --number-of-workers <count>
```

Used for fixed scaling when autoscale is not enabled.

---

5. What are best practices for autoscaling App Service?

- Use min/max limits to control scale boundaries
- Combine CPU and schedule rules for stability
- Enable App Insights to monitor autoscale behavior
- Use Premium tiers for memory-based and faster scaling

Develop Azure Compute Solutions

└ 1.2 Implement Azure App Service Web Apps

└ 1.2.6 Configure deployment slots

1. What are deployment slots and when should you use them?
2. How do you create a deployment slot via CLI?
3. How do you swap slots in App Service?
4. What settings can be cloned or configured per slot?
5. What are best practices for using deployment slots?

---

1. What are deployment slots and when should you use them?

Slots are live app environments (e.g., staging, testing) under the same App Service.
Use for: zero-downtime deployments, staged testing, A/B validation.

---

2. How do you create a deployment slot via CLI?

```
az webapp deployment slot create \
  --name <app> --resource-group <rg> --slot staging
```

Inherits settings from the production slot by default.

---

3. How do you swap slots in App Service?

```
az webapp deployment slot swap \
  --name <app> --resource-group <rg> \
  --slot staging --target-slot production
```

Promotes staging to production with no downtime.

---

4. What settings can be cloned or configured per slot?

- Cloned: App settings, connection strings (default)
- Slot-specific settings: Mark as "deployment slot setting" to isolate
  Examples: DB connection strings, API keys

---

5. What are best practices for using deployment slots?

- Use staging slot for validation
- Enable auto-swap with health checks
- Keep secrets slot-specific
- Minimize downtime by warming up the slot before swap

Develop Azure Compute Solutions

└ 1.3 Implement Azure Functions

└ 1.3.1 Create and Configure an Azure Functions App

1. What hosting plans are available for Azure Functions, and when should each be used?
2. How do you create an Azure Function App using Azure CLI?
3. What are the key configuration settings for a Function App?
4. How do you configure the runtime stack and version?
5. What are the authentication options for securing Function Apps?
6. How do you configure application settings and connection strings?
7. How do you set and manage function-level timeouts?
8. What are the deployment options for Azure Functions?
9. How do you configure CORS for a Function App?
10. How do you configure monitoring and diagnostics?

---

1. What hosting plans are available for Azure Functions, and when should each be used?

- Consumption Plan: Auto-scales and is cost-effective for sporadic workloads. Limited to 5 mins (default) or 10 mins (max) execution.
- Premium Plan: Supports VNETs, unlimited execution time, and pre-warmed instances. Use for high-load or enterprise apps.
- Dedicated (App Service) Plan: Use if consolidating services on the same App Service plan or for always-on requirements with full control.

---

2. How do you create an Azure Function App using Azure CLI?

```
az functionapp create \
  --resource-group <rg> \
  --consumption-plan-location <region> \
  --runtime <dotnet|node|python|java|powershell> \
  --functions-version <version> \
  --name <app-name> \
  --storage-account <storage-name>
```

---

3. What are the key configuration settings for a Function App?

- Runtime Stack & Version
- Platform Architecture (32-bit/64-bit)
- Always On (Premium/Dedicated only)
- Application Settings (APPINSIGHTS_INSTRUMENTATIONKEY, etc.)
- CORS
- Authentication

---

4. How do you configure the runtime stack and version?

Use CLI or Portal:

```
az functionapp config set \
  --name <app-name> \
  --resource-group <rg> \
  --linux-fx-version "DOTNET|6.0"
```

Use "linux-fx-version" or "windows-fx-version" depending on OS.

5. What are the authentication options for securing Function Apps?

- App-level authentication: Use Microsoft Entra ID, Facebook, Google, etc.
- Function key/token: For per-function access control.
- Anonymous access: Only if explicitly enabled.
- Recommended: Use Entra ID for secure, enterprise-grade auth.

---

6. How do you configure application settings and connection strings?

Use Azure CLI or Portal. Example with CLI:

```
az functionapp config appsettings set \
  --name <app-name> \
  --resource-group <rg> \
  --settings "MySetting=value" "MyConnStr=connectionstring"
```

Connection strings should be stored as app settings and referenced via `Environment.GetEnvironmentVariable()` in code.

---

7. How do you set and manage function-level timeouts?

- Consumption Plan: Max 5 min by default, extendable to 10 min via host.json:

```
{
 "functionTimeout": "00:10:00"
}
```

- Premium/Dedicated Plan: No timeout limit.

---

8. What are the deployment options for Azure Functions?

- Zip deployment: az functionapp deployment source config-zip
- Local Git or GitHub: Continuous integration via DevOps or GitHub Actions
- VS Code / Azure CLI: Direct push
- Azure DevOps Pipelines: Full CI/CD with environment config

---

9. How do you configure CORS for a Function App?

```
az functionapp cors add \
  --name <app-name> \
  --resource-group <rg> \
  --allowed-origins https://example.com
```

Use * only for public or development APIs — avoid in production.

---

10. How do you configure monitoring and diagnostics?

- Enable Application Insights during creation or via:

```
az monitor app-insights component create \
--app <app-name> --location <region> --resource-group <rg>
```

- Link to Function App:

```
az functionapp config appsettings set \
--name <app-name> --resource-group <rg> \
--settings "APPINSIGHTS_INSTRUMENTATIONKEY=<key>"
```

Logs and metrics are available in App Insights, Azure Monitor, or via az monitor.

Develop Azure Compute Solutions

└ 1.3 Implement Azure Functions

└ 1.3.2 Implement input and output bindings

1. What are input and output bindings in Azure Functions?
2. What types of bindings are most commonly used?
3. How do you define bindings in function.json?
4. How are bindings configured in C# using attributes?
5. What is the difference between trigger and input bindings?
6. How do you bind to Azure Blob Storage?
7. How do you bind to Azure Queue Storage?
8. How do you bind to Azure Cosmos DB?
9. How do you use output bindings to return data from a function?
10. How do you manage binding configuration via app settings?

---

1. What are input and output bindings in Azure Functions?

Bindings abstract external data sources into declarative configurations.

- Input bindings: Bring external data into the function.
- Output bindings: Send function output to an external service.

---

2. What types of bindings are most commonly used?

- Azure Blob Storage
- Azure Queue Storage
- Azure Cosmos DB
- HTTP (Trigger + Output)
- Azure Table Storage

---

3. How do you define bindings in function.json?

Bindings are defined as JSON objects. Example:

```
{
 "bindings": [
  {
   "name": "myBlob",
   "type": "blobTrigger",
   "direction": "in",
   "path": "samples/{name}",
   "connection": "AzureWebJobsStorage"
  }
 ]
}
```

---

4. How are bindings configured in C# using attributes?

Example (Queue input, Blob output):

```
public static void Run(
   [QueueTrigger("queue-name")] string input,
   [Blob("output-container/{rand-guid}", FileAccess.Write)] out string outputBlob)
            {
               outputBlob = input.ToUpper();
            }
```

Use specific attributes like [BlobTrigger], [Queue] , [CosmosDB] .

5. What is the difference between trigger and input bindings?

- Trigger: Initiates function execution (only one per function).
- Input: Supplies data for processing (can have multiple).
  Example: A blobTrigger starts the function; a Cosmos DB input binding provides extra data.

---

6. How do you bind to Azure Blob Storage?

Input (read):

```
[BlobTrigger("input-container/{name}", Connection = "AzureWebJobsStorage")] Stream inputBlob
```

Output (write):

```
[Blob("output-container/{name}", FileAccess.Write, Connection = "AzureWebJobsStorage")] Stream outputBlob
```

---

7. How do you bind to Azure Queue Storage?

Trigger:

```
[QueueTrigger("queue-name", Connection = "AzureWebJobsStorage")] string msg
```

Output:

```
[Queue("output-queue", Connection = "AzureWebJobsStorage")] out string outputQueueMsg
```

---

8. How do you bind to Azure Cosmos DB?

Input (read):

```
[CosmosDB(
   databaseName: "db",
   collectionName: "coll",
   Id = "id",
   PartitionKey = "key",
   Connection = "CosmosDBConnection")] dynamic item
```

Output (write):

```
[CosmosDB(
   databaseName: "db",
   collectionName: "coll",
   Connection = "CosmosDBConnection")] out dynamic outputItem
```

---

9. How do you use output bindings to return data from a function?

- Use the return statement for HTTP output or single output bindings:

```
[FunctionName("HttpExample")]
public static IActionResult Run([HttpTrigger(...)] HttpRequest req, [Queue("myqueue")] out string msg)
{
   msg = "Queue message";
   return new OkObjectResult("Success");
}
```

---

10. How do you manage binding configuration via app settings?

- Use placeholders in function.json or attribute Connection:

```
"connection": "AzureWebJobsStorage"
```

- Then define the value in Configuration > Application settings in the portal or via:

```
az functionapp config appsettings set --name <app-name> --settings AzureWebJobsStorage=<conn-string>
```

Develop Azure Compute Solutions

└ 1.3 Implement Azure Functions

└ 1.3.3 Implement function triggers by using data operations, timers, and webhooks

1. What are function triggers and how do they differ from bindings?
2. What are the main trigger types?
3. How do you implement an HTTP trigger?
4. How do you implement a Timer trigger and define its schedule?
5. How do you implement a Blob trigger?
6. How do you implement a Queue Storage trigger?
7. How do you implement a Cosmos DB trigger?
8. How do you implement a Service Bus trigger?
9. How do you handle errors and retries in triggered functions?
10. What are best practices for managing trigger configuration?

---

1. What are function triggers and how do they differ from bindings?

Triggers start the function — only one allowed per function.
Bindings are for data input/output — zero or more allowed.
Example: HttpTrigger, TimerTrigger, BlobTrigger.

---

2. What are the main trigger types tested on the exam?

- HttpTrigger
- TimerTrigger
- BlobTrigger
- QueueTrigger
- CosmosDBTrigger
- ServiceBusTrigger

---

3. How do you implement an HTTP trigger?

```
[FunctionName("HttpExample")]
public static IActionResult Run(
   [HttpTrigger(AuthorizationLevel.Function, "get", "post")] HttpRequest req)
{
   return new OkObjectResult("Hello from HTTP trigger");
}
```

Supports get, post, etc. Use AuthorizationLevel to control access.

---

4. How do you implement a Timer trigger and define its schedule?

```
[FunctionName("TimerExample")]
public static void Run([TimerTrigger("0 */5 * * * *")] TimerInfo myTimer)
{
   // Executes every 5 minutes
}
```

CRON format: "{second} {minute} {hour} {day} {month} {day-of-week}".

5. How do you implement a Blob trigger?

```
[FunctionName("BlobExample")]
public static void Run(
[BlobTrigger("container/{name}", Connection = "AzureWebJobsStorage")] Stream myBlob, string name)
{
   // Reacts to new/updated blobs
}
```

---

6. How do you implement a Queue Storage trigger?

```
[FunctionName("QueueTriggerExample")]
public static void Run([QueueTrigger("my-queue", Connection = "AzureWebJobsStorage")] string queueItem)
{
   // Processes messages from Azure Queue Storage
}
```

---

7. How do you implement a Cosmos DB trigger?

```
[FunctionName("CosmosTriggerExample")]
public static void Run(
[CosmosDBTrigger(
databaseName: "db",
collectionName: "coll",
ConnectionStringSetting = "CosmosDBConnection",
LeaseCollectionName = "leases")] IReadOnlyList<Document> input)
{
   // Reacts to inserts and updates in Cosmos DB
}
```

Leases track change feed progress.

---

8. How do you implement a Service Bus trigger?

```
[FunctionName("ServiceBusExample")]
public static void Run(
   [ServiceBusTrigger("queue-name", Connection = "ServiceBusConnection")] string message)
{
   // Handles messages from Azure Service Bus Queue
}
```

Use "queue-name" or "topic-name", "subscription-name" for topics.

---

9. How do you handle errors and retries in triggered functions?

- Retry policies can be set in host.json:

```
{
  "retry": {
    "strategy": "fixedDelay",
    "maxRetryCount": 3,
    "delayInterval": "00:00:05"
  }
}
```

- Unhandled exceptions automatically trigger retries (except for HTTP).

---

10. What are best practices for managing trigger configuration?

- Use app settings for all connection strings
- Abstract trigger values (e.g., queue name, CRON) into settings
- Use dependency injection for service management
- Limit execution time (set functionTimeout)
- Avoid long-running or blocking operations

Develop for Azure Storage

└ 2.1 Develop solutions that use Azure Cosmos DB

└ 2.1.1 Perform operations on containers and items by using the SDK

1. What SDKs are supported for Cosmos DB operations?
2. How do you create a container in Cosmos DB using the SDK?
3. How do you insert or update an item?
4. How do you query items using SQL syntax?
5. How do you delete an item by ID?
6. How do you use partition keys effectively?
7. What are common consistency levels and how do you set them?
8. How do you use the CosmosClient safely and efficiently?
9. How do you handle pagination (continuation tokens)?
10. How is exception handling and retry logic implemented?

---

1. What SDKs are supported for Cosmos DB operations?

.NET (Microsoft.Azure.Cosmos), Java, Python, Node.js

---

2. How do you create a container in Cosmos DB using the .NET SDK?

```
await database.CreateContainerIfNotExistsAsync("MyContainer", "/partitionKey");
```

- "/partitionKey" is required.
- Creates the container only if it doesn't exist.

---

3. How do you insert or update an item?

```
await container.UpsertItemAsync(item, new PartitionKey(item.partitionKey));
```

- UpsertItemAsync inserts or replaces item based on ID.
- Requires correct partition key.

---

4. How do you query items using SQL syntax?

```
var query = container.GetItemQueryIterator<MyItem>("SELECT * FROM c WHERE c.status = 'active'");
while (query.HasMoreResults)
{
  foreach (var item in await query.ReadNextAsync())
  {
    // Process item
  }
}
```

- Uses Cosmos SQL API.
- Handles pagination internally.

---

5. How do you delete an item by ID?

```
await container.DeleteItemAsync<MyItem>(id, new PartitionKey(partitionKey));
```

- Both ID and correct partition key are required.

---

6. How do you use partition keys effectively?

- Choose a key with high cardinality and even distribution (e.g., /userId).
- Required for most operations (read, update, delete).

7. What are common consistency levels and how do you set them?

- Levels: Strong, BoundedStaleness, Session (default), ConsistentPrefix, Eventual
- Set at CosmosClientOptions level:

```
new CosmosClient(endpoint, key, new CosmosClientOptions { ConsistencyLevel = ConsistencyLevel.Session });
```

---

8. How do you use the CosmosClient safely and efficiently?

- Reuse a single CosmosClient instance (thread-safe).
- Instantiate once at app startup (e.g., via dependency injection).

---

9. How do you handle pagination (continuation tokens)?

- Use `FeedIterator<T>` from `GetItemQueryIterator<T>()`
- Cosmos handles paging; iterate until HasMoreResults is false.

---

10. How is exception handling and retry logic implemented?

- Catch CosmosException for specific status codes:

```
catch (CosmosException ex) when (ex.StatusCode == HttpStatusCode.TooManyRequests)
{
   await Task.Delay(ex.RetryAfter);
}
```

- SDK includes automatic retry policies; customize via CosmosClientOptions.

Develop for Azure Storage

└─ 2.1 Develop solutions that use Azure Cosmos DB

└─ 2.1.2 Set the appropriate consistency level for operations

1. What are the consistency levels supported by Azure Cosmos DB?
2. What is the default consistency level and why is it recommended?
3. How do you configure consistency level at the account level?
4. How do you override the consistency level per request?
5. What is session consistency and when should it be used?
6. What are the trade-offs between strong and eventual consistency?
7. How does consistency affect performance and availability?
8. How do you check the current consistency level of an account?
9. Which operations are affected by the chosen consistency level?
10. What are best practices for setting consistency levels in real-world applications?

---

1. What are the consistency levels supported by Azure Cosmos DB?

- Strong
- BoundedStaleness
- Session *(default)*
- ConsistentPrefix
- Eventual

---

2. What is the default consistency level and why is it recommended?

- Session is default.
- Guarantees *read-your-own-writes* within a session.
- Balanced choice for consistency and performance.

---

3. How do you configure consistency level at the account level?

- Set during Cosmos DB account creation or via Azure Portal:
  - *Settings → Default consistency*
- Or using SDK:

```
CosmosClientOptions.ConsistencyLevel = ConsistencyLevel.Session;
```

---

4. How do you override the consistency level per request?

```
var requestOptions = new QueryRequestOptions
{
   ConsistencyLevel = ConsistencyLevel.Eventual
};
```

- Applies only to the specific request.
- Must be equal or weaker than the account-level setting.

---

5. What is session consistency and when should it be used?

- Guarantees *read-your-own-writes* for a session token.
- Ideal for user-specific data scenarios (e.g., profile updates, shopping carts).

6. What are the trade-offs between strong and eventual consistency?

- Strong: Highest data accuracy, lowest availability across regions.
- Eventual: Best performance and availability, but stale reads are possible.

---

7. How does consistency affect performance and availability?

- Weaker levels (Eventual, ConsistentPrefix) offer lower latency and higher throughput.
- Stronger levels (Strong, BoundedStaleness) increase latency and reduce write availability in multi-region setups.

---

8. How do you check the current consistency level of an account?

- Use Azure Portal → *Settings* → *Default consistency*
- Or SDK:
  ```
  var consistency = cosmosClient.ClientOptions.ConsistencyLevel;
  ```

---

9. Which operations are affected by the chosen consistency level?

- Read operations: The chosen level impacts how up-to-date the reads are.
- Write operations are always consistent.

---

10. What are best practices for setting consistency levels in real-world applications?

- Use Session for most app scenarios (low latency + strong enough).
- Use Strong only when global read consistency is critical.
- Use Eventual or ConsistentPrefix for high-throughput, read-heavy apps where data freshness is not critical.

Develop for Azure Storage

└ 2.1 Develop solutions that use Azure Cosmos DB

└ 2.1.3 Implement change feed notifications

1. What is the change feed in Azure Cosmos DB?
2. What types of changes does the change feed capture?
3. How do you read from the change feed using the SDK?
4. What is the difference between manual polling vs. Change Feed Processor?
5. How do you implement the Change Feed Processor in .NET?
6. How do you scale out a change feed listener?
7. What are common use cases for the change feed?
8. What is lease container and why is it required?
9. How do you resume reading from a specific point in the change feed?
10. What are best practices for change feed implementations?

---

1. What is the change feed in Azure Cosmos DB?

- A persistent, ordered log of item changes (inserts and updates) in a container.
- Enables event-driven processing without polling the whole dataset.

---

2. What types of changes does the change feed capture?

- Creates and updates only.
- Deletes are not included. You must implement soft delete patterns if needed.

---

3. How do you read from the change feed using the SDK?

```
var iterator = container.GetChangeFeedIterator<MyItem>(
   ChangeFeedStartFrom.Beginning(), ChangeFeedMode.Incremental);
while (iterator.HasMoreResults)
{
   var response = await iterator.ReadNextAsync();
   foreach (var item in response)
   {
      // Process item
   }
}
```

---

4. What is the difference between manual polling vs. Change Feed Processor?

- Manual polling: Directly queries the feed; full control but must manage state and scaling.
- Change Feed Processor: Auto-scales and handles lease/state tracking via a lease container.

---

5. How do you implement the Change Feed Processor in .NET?

```
var processor = container
   .GetChangeFeedProcessorBuilder<MyItem>("myProcessor", async (changes, token) =>
   {
      foreach (var item in changes) { /* process */ }
   })
   .WithInstanceName("worker1")
   .WithLeaseContainer(leaseContainer)
   .Build();
await processor.StartAsync();
```

- Requires a lease container for tracking progress.

6. How do you scale out a change feed listener?

- Use multiple instances of Change Feed Processor with the same lease container.
- The processor automatically partitions work across instances.

---

7. What are common use cases for the change feed?

- Event-driven processing (e.g., send emails, process orders)
- Real-time analytics
- Data movement to other stores (e.g., SQL, Blob Storage)
- Cache invalidation or sync

---

8. What is lease container and why is it required?

- A separate Cosmos DB container used by Change Feed Processor to track progress.
- Stores checkpoints and ownership info for scaling and fault-tolerance.

---

9. How do you resume reading from a specific point in the change feed?

- Change Feed Processor resumes automatically via lease container.
- Manual method: use `ChangeFeedStartFrom.Time()` or `ChangeFeedStartFrom.ContinuationToken()`.

---

10. What are best practices for change feed implementations?

- Use dedicated lease container in same database.
- Ensure idempotent processing logic.
- Handle throttling and retries using SDK's retry policies.
- Monitor lag and exceptions for performance tuning.

Develop for Azure Storage

└ 2.2 Develop solutions that use Azure Cosmos DB

└ 2.2.1 Set and retrieve properties and metadata

1. What are blob properties and blob metadata?
2. How do you set blob properties (like content type, cache control)?
3. How do you retrieve blob properties?
4. How do you set custom metadata on a blob?
5. How do you retrieve metadata from a blob?
6. What are best practices when using metadata in Azure Blob Storage?
7. What happens if you overwrite a blob — are properties and metadata preserved?
8. How do you set or update metadata without overwriting the blob content?
9. How do you use Azure SDK (C#, Python) to manage properties and metadata?
10. How can you search or filter blobs by metadata?

---

1. What are blob properties and blob metadata?

- Blob properties are system-defined attributes like `Content-Type`, `Content-Encoding`, `Cache-Control`, and `Content-Length`.
- Blob metadata consists of user-defined key-value pairs that describe the blob but do not affect its behavior.

---

2. How do you set blob properties (like content type, cache control)?

- When uploading or updating a blob, set properties using `BlobClient.Upload()` with `BlobHttpHeaders`.
- Example (C# Azure SDK):

```
await blobClient.UploadAsync(fileStream, new BlobHttpHeaders { ContentType = "image/png", CacheControl = "no-cache" });
```

---

3. How do you retrieve blob properties?

- Use `BlobClient.GetPropertiesAsync()` method.
- Example (C# Azure SDK):

```
BlobProperties properties = await blobClient.GetPropertiesAsync();
Console.WriteLine(properties.ContentType);
Console.WriteLine(properties.CacheControl);
```

---

4. How do you set custom metadata on a blob?

- Use `BlobClient.SetMetadataAsync()` with a dictionary of key-value pairs.
- Example (C# Azure SDK):

```
var metadata = new Dictionary<string, string> { { "author", "john_doe" }, { "category", "images" } };
await blobClient.SetMetadataAsync(metadata);
```

---

5. How do you retrieve metadata from a blob?

- Call `BlobClient.GetPropertiesAsync()` and access the Metadata property.
- Example (C# Azure SDK):

```
BlobProperties properties = await blobClient.GetPropertiesAsync();
foreach (var item in properties.Metadata)
{
   Console.WriteLine($"{item.Key}: {item.Value}");
}
```

6. What are best practices when using metadata in Azure Blob Storage?

- Keep metadata size small (max 8 KB total per blob).
- Use lowercase keys; metadata keys are case-insensitive.
- Metadata is stored separately; retrieving it requires an extra API call (costs apply).

---

7. What happens if you overwrite a blob — are properties and metadata preserved?

- No, uploading a blob without explicitly setting metadata and properties will reset them to defaults.
- Always reapply desired metadata and properties during overwrite if needed.

---

8. How do you set or update metadata without overwriting the blob content?

- Use `BlobClient.SetMetadataAsync()` — it updates metadata without affecting the blob's content.
- No need to re-upload the blob when updating only metadata.

---

9. How do you use Azure SDK (C#, Python) to manage properties and metadata?

- C# Example:

```
await blobClient.SetMetadataAsync(new Dictionary<string, string> { { "env", "prod" } });
BlobProperties props = await blobClient.GetPropertiesAsync();
Console.WriteLine(props.ContentType);
```

---

10. How can you search or filter blobs by metadata?

- Use Azure Blob Index Tags, not regular metadata.
- With tags, you can query blobs via FindBlobsByTags.
- Example (Azure CLI):

```
az storage blob query-tags --container-name mycontainer --where "tagName = 'value'"
```

Develop for Azure Storage

└ 2.2 Develop solutions that use Azure Cosmos DB

└ 2.2.2 Perform operations by using the appropriate SDK

1. What SDKs are supported for Azure Blob Storage operations?
2. How do you create and upload a blob using the Azure SDK (C#, Python)?
3. How do you download a blob using the Azure SDK?
4. How do you list blobs inside a container?
5. How do you delete a blob using the Azure SDK?
6. How do you perform conditional operations (e.g., upload if not exists)?
7. How do you use a stream to upload/download blobs?
8. How do you handle large blobs efficiently (e.g., upload in blocks)?
9. How do you set retries and timeouts in SDK operations?
10. What are best practices for SDK usage in production?

---

1. What SDKs are supported for Azure Blob Storage operations?

- Official Azure SDKs:
  - .NET (Azure.Storage.Blobs)
  - Python (azure-storage-blob)
  - Java (azure-storage-blob)
  - JavaScript/TypeScript (e.g., @azure/storage-blob)

---

2. How do you create and upload a blob using the Azure SDK (C#, Python)?

- C# Example:

```
BlobClient blobClient = containerClient.GetBlobClient("myblob.txt");
await blobClient.UploadAsync("localfile.txt", overwrite: true);
```

---

3. How do you download a blob using the Azure SDK?

```
BlobDownloadInfo download = await blobClient.DownloadAsync();
using (FileStream fs = File.OpenWrite("downloaded.txt"))
{
   await download.Content.CopyToAsync(fs);
}
```

---

4. How do you list blobs inside a container?

```
await foreach (BlobItem blobItem in containerClient.GetBlobsAsync())
{
   Console.WriteLine(blobItem.Name);
}
```

---

5. How do you delete a blob using the Azure SDK?

```
await blobClient.DeleteIfExistsAsync();
```

---

6. How do you perform conditional operations (e.g., upload if not exists)?

- Set the conditions parameter with an If-None-Match: * condition.
- C# Example:

```
var conditions = new BlobRequestConditions { IfNoneMatch = new ETag("*") };
await blobClient.UploadAsync("localfile.txt", conditions: conditions);
```

7. How do you use a stream to upload/download blobs?

- Upload Stream (C#):

```
using var stream = File.OpenRead("localfile.txt");
await blobClient.UploadAsync(stream, overwrite: true);
```

- Download Stream (C#):

```
BlobDownloadInfo download = await blobClient.DownloadAsync();
using var file = File.OpenWrite("output.txt");
await download.Content.CopyToAsync(file);
```

---

8. How do you handle large blobs efficiently (e.g., upload in blocks)?

- Use `UploadAsync` for files up to 256 MB (default).
- For larger files, use `UploadAsync` with automatic chunking or manually use `BlockBlobClient.StageBlockAsync` and `CommitBlockListAsync`.

---

9. How do you set retries and timeouts in SDK operations?

- C# Example:

```
BlobClientOptions options = new BlobClientOptions
{
   Retry =
   {
      MaxRetries = 5,
      Delay = TimeSpan.FromSeconds(2),
      MaxDelay = TimeSpan.FromSeconds(10),
      Mode = RetryMode.Exponential
   }
};
var blobServiceClient = new BlobServiceClient(connectionString, options);
```

---

10. What are best practices for SDK usage in production?

- Always set appropriate retry policies and timeouts.
- Prefer streams for large files.
- Handle exceptions explicitly (e.g., RequestFailedException in C#).
- Reuse BlobServiceClient, BlobContainerClient, and BlobClient instances (they are thread-safe).
- Secure secrets and connection strings (use Azure Managed Identity if possible).

Develop for Azure Storage

└ 2.2 Develop solutions that use Azure Cosmos DB

└ 2.2.3 Implement storage policies and data lifecycle management

1. What is Azure Blob Lifecycle Management?
2. How do you define a lifecycle management rule?
3. How do you move blobs between access tiers (Hot, Cool, Archive)?
4. How do you configure auto-delete for old blobs?
5. How do you apply rules based on blob metadata or naming patterns?
6. How do you create lifecycle management rules using the Azure Portal?
7. How do you create lifecycle policies programmatically (Azure CLI, SDK)?
8. What are best practices for setting retention and tiering policies?
9. How does Archive rehydration work and what are the implications?
10. How do you monitor and troubleshoot lifecycle policy actions?

---

1. What is Azure Blob Lifecycle Management?

- A feature that automatically moves, deletes, or archives blobs based on rules and conditions like age, last modified date, or access tier.
- Reduces storage costs and enforces data retention policies.

---

2. How do you define a lifecycle management rule?

- A rule consists of:
  - Filters (prefix match, blob type, metadata conditions).
  - Actions (move to Cool/Archive tier, delete blob).
- Rules are evaluated daily by Azure.

---

3. How do you move blobs between access tiers (Hot, Cool, Archive)?

- Define a lifecycle rule that moves blobs based on conditions:
  - Example: Move to Cool if not modified for 30 days.
  - Example: Move to Archive if not modified for 180 days.
- No manual intervention needed once the policy is active.

---

4. How do you configure auto-delete for old blobs?

- Set a Delete action in a lifecycle management rule.
- Example: Delete blobs 90 days after the last modified date.
- Can combine delete action with filters (e.g., only for blobs with a specific prefix).

---

5. How do you apply rules based on blob metadata or naming patterns?

- Use filters when defining the rule:
  - Prefix match: Target blobs under a virtual folder path.
  - Blob index tags: Target blobs with specific metadata conditions (e.g., env=prod).
- Example filter:

```
"prefixMatch": ["logs/"],
"blobTypes": ["blockBlob"]
```

6. How do you create lifecycle management rules using the Azure Portal?

- Navigate to Storage Account → Data Management → Lifecycle Management.
- Add a rule:
  - Define conditions (e.g., last modified > 30 days).
  - Specify actions (move, delete).
- Save and enable the rule — applies automatically to matching blobs.

---

7. How do you create lifecycle policies programmatically (Azure CLI, SDK)?

- Azure CLI Example:

```
az storage account management-policy create \
   --account-name <storageaccount> \
   --resource-group <resourcegroup> \
   --policy @"policy.json"
```

- C# SDK Example:

```
var managementPolicy = new ManagementPolicy
{
   Policy = JsonConvert.DeserializeObject<ManagementPolicySchema>(policyJson)
};
await storageAccount.UpdateAsync(managementPolicy: managementPolicy);
```

- Policy is defined in a JSON file describing rules and actions.

---

8. What are best practices for setting retention and tiering policies?

- Use Cool tier for infrequently accessed data (accessed > 30 days).
- Use Archive tier for rarely accessed data (accessed > 180 days).
- Avoid aggressive delete rules unless compliance requirements mandate it.
- Test lifecycle rules in non-production first to verify behavior.

---

9. How does Archive rehydration work and what are the implications?

- Rehydration moves blobs from Archive to Hot or Cool tier to make them accessible.
- It can take hours (up to 15 hours standard, faster rehydration possible).
- Rehydration is asynchronous; status must be polled using blob properties.

---

10. How do you monitor and troubleshoot lifecycle policy actions?

- Review Activity Log in Azure Portal for lifecycle policy operations.
- Check Blob properties to verify last tier change date.
- Use Azure Monitor metrics to track storage capacity changes by tier.

Implement Azure security

└ **3.1 Implement user authentication and authorization**

└ **3.1.1 Authenticate and authorize users by using the Microsoft Identity platform**

1. What is the Microsoft Identity Platform?
2. How do you register an application with Microsoft Entra ID (formerly Azure AD)?
3. What is the difference between single-tenant and multi-tenant apps?
4. What authentication flows are supported in the Microsoft Identity Platform?
5. How do you implement authentication using MSAL?
6. How do you configure permissions (scopes) and consent?
7. How do you acquire and validate access tokens?
8. How do you secure an API using the Microsoft Identity Platform?
9. How do you configure redirect URIs and reply URLs?
10. What is the difference between delegated and application permissions?

---

1. What is the Microsoft Identity Platform?

A Microsoft authentication system that provides OAuth 2.0 and OpenID Connect protocols for authenticating users and securing APIs. It integrates with Microsoft Entra ID (Azure AD).

---

2. How do you register an application with Microsoft Entra ID?

Use Azure Portal → Entra ID → App registrations → New registration.

Set a name, supported account types, and redirect URI. Save the Application (client) ID.

---

3. What is the difference between single-tenant and multi-tenant apps?

- Single-tenant: Only users in one Entra ID tenant can access the app.
- Multi-tenant: Users in any Entra ID tenant can authenticate.

---

4. What authentication flows are supported in the Microsoft Identity Platform?

- Authorization Code (interactive user login)
- Client Credentials (daemon apps)
- Device Code (devices without browser)
- ROPC (username/password; not recommended)
- On-Behalf-Of (service-to-service delegation)

---

5. How do you implement authentication using MSAL?

Use the Microsoft Authentication Library (MSAL) to acquire tokens:

```
var result = await app.AcquireTokenInteractive(scopes).ExecuteAsync();
```

MSAL handles caching, token renewal, and multiple flows.

---

6. How do you configure permissions (scopes) and consent?

Define scopes in the app registration under Expose an API.

- Admins or users must consent to scopes (e.g., user.read).
- API permissions tab controls delegated vs. application scopes.

---

7. How do you acquire and validate access tokens?

Use MSAL to acquire tokens (e.g., AcquireTokenInteractive, AcquireTokenForClient).

Validate tokens in the API using middleware (e.g., ASP.NET JwtBearerOptions) and Microsoft identity metadata endpoint.

8. How do you secure an API using the Microsoft Identity Platform?

- Register the API as an application.
- Define scopes under "Expose an API".
- Protect routes using [Authorize] and validate tokens using middleware (AddAuthentication().AddJwtBearer()).

---

9. How do you configure redirect URIs and reply URLs?

Set in Azure Portal → App registration → Authentication.

- Must match what's used in your app exactly.
- Used during OAuth flows to redirect users back after authentication.

---

10. What is the difference between delegated and application permissions?

- Delegated: Act on behalf of a user. Used with signed-in users.
- Application: Act as the app itself. Used in background services (e.g., daemons).

Implement Azure security

└ 3.1 Implement user authentication and authorization

└ 3.1.2 Authenticate and authorize users and apps by using Microsoft Entra ID

1. What is Microsoft Entra ID and how is it used in authentication and authorization?
2. What are managed identities and when should you use them?
3. How do you assign roles to users and apps in Entra ID?
4. How do you implement role-based access control (RBAC)?
5. How do you use Microsoft Graph to check user roles or group membership?
6. How do you authenticate using client credentials (app-only access)?
7. How do you configure an app to use a managed identity?
8. How do you restrict access to Azure resources using Entra ID?
9. How do you authorize apps to access APIs on behalf of a user?
10. What are best practices for securing app access via Entra ID?

---

1. What is Microsoft Entra ID and how is it used in authentication and authorization?

Microsoft Entra ID (formerly Azure AD) is Microsoft's cloud-based identity service.

- Authentication: Verifies user or app identity.
- Authorization: Controls access via roles, groups, or policies to Azure and custom resources.

---

2. What are managed identities and when should you use them?

System- or user-assigned identities created in Entra ID for Azure resources (e.g., App Service, Functions).
Use them to authenticate without secrets when calling Entra-secured resources like Key Vault or Graph.

---

3. How do you assign roles to users and apps in Entra ID?

- Go to the Azure resource → Access control (IAM) → Add role assignment.
- Assign roles (e.g., Reader, Contributor) to users, groups, or service principals.
  Use az role assignment create to script this.

---

4. How do you implement role-based access control (RBAC)?

Use Entra ID roles (built-in or custom) and assign them to identities.
Access is enforced based on assigned role scopes (e.g., resource group, subscription).

---

5. How do you use Microsoft Graph to check user roles or group membership?

Use Graph API endpoint /me/memberOf or /users/{id}/getMemberGroups.
Requires Group.Read.All or similar delegated/app permission.
Example:

```
GET https://graph.microsoft.com/v1.0/me/memberOf
```

---

6. How do you authenticate using client credentials (app-only access)?

Register the app in Entra ID → Generate a client secret or certificate → Grant API permissions.
Use MSAL or REST to request a token with client_id, client_secret, tenant_id, and scope.
Flow: OAuth 2.0 client credentials grant.

---

7. How do you configure an app to use a managed identity?

- Enable system-assigned identity in the Azure resource (App Service, Function, VM).
- Assign RBAC role to that identity (e.g., Key Vault Reader).
- Access tokens via Azure SDK's DefaultAzureCredential or IMDS endpoint.

8. How do you restrict access to Azure resources using Entra ID?
Use RBAC:

- Assign specific roles (e.g., Reader) to Entra identities.
- Scope can be subscription, resource group, or individual resource.
- Enforced via Entra token claims and role assignments.

---

9. How do you authorize apps to access APIs on behalf of a user?
Use delegated permissions via OAuth 2.0 authorization code flow.
The app receives an access token with the user's identity.
Ensure scopes like User.Read are consented to during sign-in.

---

10. What are best practices for securing app access via Entra ID?

- Use managed identity instead of storing secrets.
- Assign minimum required RBAC roles.
- Use conditional access policies where applicable.
- Validate token issuer, audience, and scopes in APIs.

Implement Azure security

└ 3.1 Implement user authentication and authorization

└ 3.1.3 Create and implement shared access signatures

1. What is a Shared Access Signature (SAS)?
2. What are the types of SAS and when should each be used?
3. How do you create a SAS using Azure Storage SDK or CLI?
4. What permissions can be specified in a SAS token?
5. How do you specify expiration, allowed IPs, and protocols in a SAS?
6. What is the difference between service SAS and account SAS?
7. How do you restrict SAS access by resource type?
8. How do you implement stored access policies?
9. What are security best practices for using SAS?
10. How do you revoke a SAS token?

---

1. What is a Shared Access Signature (SAS)?

A SAS is a signed URI that grants limited access to Azure Storage resources without exposing account keys. It defines permissions, scope, and expiry.

---

2. What are the types of SAS and when should each be used?

- User delegation SAS: Uses Azure AD credentials. Most secure.
- Service SAS: Grants access to specific resource (blob, file, etc.).
- Account SAS: Grants access to any service in the account (blob, queue, file, table). Use for broader access needs.

---

3. How do you create a SAS using Azure Storage SDK or CLI?

- CLI:

```
az storage blob generate-sas --account-name <name> --container-name <c> --name <blob> --permissions r --expiry <time>
```

- SDK: Use BlobSasBuilder in .NET or equivalent in other languages.

---

4. What permissions can be specified in a SAS token?

Depends on resource type. Examples:

- Blob: r (read), w (write), d (delete), l (list), a (add), c (create)
- Queue: r, a, u (update), p (process)

---

5. How do you specify expiration, allowed IPs, and protocols in a SAS?

In the SAS definition:

- --expiry (e.g., 2025-05-01T00:00Z)
- --ip (e.g., 168.1.5.60-168.1.5.70)
- --https-only true to restrict to HTTPS.

---

6. What is the difference between service SAS and account SAS?

- Service SAS: Grants access to a specific resource (e.g., a blob).
- Account SAS: Grants access across services (Blob, File, Queue, Table) in a storage account. Account SAS is broader and riskier if leaked.

7. How do you restrict SAS access by resource type?

Use the --resource-types parameter (for account SAS):

- s (service), c (container), o (object)
  Example:
  --resource-types sco limits access to services, containers, and objects.

---

8. How do you implement stored access policies?

Stored access policies are defined on containers and linked to SAS tokens to centrally manage expiry and permissions.

Create with:

```
az storage container policy create
```

Then reference the --policy-name in SAS generation.

---

9. What are security best practices for using SAS?

- Set short expiry times.
- Use HTTPS only.
- Restrict IP range if possible.
- Prefer user delegation SAS over account SAS.
- Avoid hardcoding SAS; store securely.

---

10. How do you revoke a SAS token?

- For account/service SAS: Rotate the storage account key.
- For stored access policy SAS: Modify or delete the policy; tokens linked to it become invalid.

Implement Azure security

└ 3.1 Implement user authentication and authorization

└ 3.1.4 Implement solutions that interact with Microsoft Graph

1. What is Microsoft Graph and what can it access?
2. How do you register an app to use Microsoft Graph?
3. What permissions are required to access Microsoft Graph?
4. How do you authenticate and call Microsoft Graph using MSAL?
5. How do you read user profile data from Microsoft Graph?
6. How do you list groups or check group membership?
7. How do you call Microsoft Graph from a background service?
8. How do you handle access token scopes and consent?
9. How do you use Graph SDK vs direct REST API?
10. What are best practices for calling Microsoft Graph securely?

---

1. What is Microsoft Graph and what can it access?

Microsoft Graph is a unified API endpoint (graph.microsoft.com) for accessing Microsoft 365 services like Entra ID (users, groups), Outlook, SharePoint, OneDrive, Teams, and more.

---

2. How do you register an app to use Microsoft Graph?

- In Azure Portal → Entra ID → App registrations → New registration
- Add API permissions for Microsoft Graph
- Optionally configure redirect URI and generate client secret or cert

---

3. What permissions are required to access Microsoft Graph?

- Delegated (signed-in user): e.g., User.Read, Mail.Read
- Application (daemon app): e.g., User.Read.All, Group.Read.All
  Some permissions require admin consent.

---

4. How do you authenticate and call Microsoft Graph using MSAL?

Acquire token via MSAL, then use HTTP or SDK.

Example (C#):

```
var result = await app.AcquireTokenForClient(scopes).ExecuteAsync();
var token = result.AccessToken;
```

---

5. How do you read user profile data from Microsoft Graph?

Use GET https://graph.microsoft.com/v1.0/me (delegated)
or GET /users/{id} (application permission).
Include access token in Authorization header:
Authorization: Bearer <token>

---

6. How do you list groups or check group membership?

- List groups: GET /groups
- Check membership: GET /me/memberOf or /users/{id}/memberOf
  Requires permissions like Group.Read.All.

7. How do you call Microsoft Graph from a background service?
Use application permissions with the client credentials flow:

- Acquire token via `AcquireTokenForClient()`
- Call Graph API using token; no user context needed.

---

8. How do you handle access token scopes and consent?
Scopes define the resources and actions an app can request.

- Delegated: Scopes like `User.Read` are granted on sign-in.
- Application: Requires admin consent via Azure Portal or admin consent URL.

---

9. How do you use Graph SDK vs direct REST API?

- SDK (e.g., Microsoft.Graph NuGet): Typed clients, fluent syntax, easier integration.
- REST: More control, immediate support for latest endpoints.
  Both use the same access tokens.

---

10. What are best practices for calling Microsoft Graph securely?

- Use least privilege scopes.
- Store secrets in Key Vault.
- Use managed identities if available.
- Validate token claims in APIs.
- Handle token caching and expiration properly.

3. Implement Azure security

└ **3.2 Implement secure Azure solutions**

└ **3.2.1 Secure app configuration data by using App Configuration or Azure Key Vault**

1. What is Azure App Configuration and when should it be used?
2. What is Azure Key Vault and when should it be used?
3. What types of secrets can be stored in Azure Key Vault?
4. How do you access secrets from Azure Key Vault in code using DefaultAzureCredential?
5. How do you integrate Azure Key Vault with App Service or Functions securely?
6. How do you use Azure App Configuration in .NET apps?
7. How do you enable Key Vault reference integration in Azure App Configuration?
8. How do you use managed identities to authenticate to Key Vault and App Configuration?
9. What are best practices for securing app settings and secrets?
10. How can you audit or monitor access to secrets in Azure Key Vault?

---

1. What is Azure App Configuration and when should it be used?

A centralized service for managing application settings and feature flags. Use it to decouple config from code across environments, especially in microservices or distributed apps.

---

2. What is Azure Key Vault and when should it be used?

A secure store for secrets, keys, and certificates. Use it for managing sensitive data (e.g., DB passwords, API keys) with RBAC and audit logging. Ideal for securing runtime secrets.

---

3. What types of secrets can be stored in Azure Key Vault?

- Secrets (e.g., passwords, connection strings)
- Keys (RSA, EC keys for encryption/signing)
- Certificates (incl. auto-renewing SSL certs)

---

4. How do you access secrets from Azure Key Vault in code using DefaultAzureCredential?

Use Azure SDK:

```
var client = new SecretClient(new Uri(kvUrl), new DefaultAzureCredential());
KeyVaultSecret secret = await client.GetSecretAsync("MySecret");
```

Requires proper RBAC role (e.g., Key Vault Secrets User) and managed identity.

---

5. How do you integrate Azure Key Vault with App Service or Functions securely?

Enable managed identity on the app, assign Key Vault Secrets User role, and reference secrets using:

```
@Microsoft.KeyVault(SecretUri=https://<vault-name>.vault.azure.net/secrets/<secret-name>/)
```

Used in app settings; no code change needed.

---

6. How do you use Azure App Configuration in .NET apps?

Install the package:

Microsoft.Extensions.Configuration.AzureAppConfiguration

Example usage:

```
builder.Configuration.AddAzureAppConfiguration(options =>
  options.Connect("<connection-string>")
      .Select("*"));
```

Use FeatureManagement for feature flags.

7. How do you enable Key Vault reference integration in Azure App Configuration?

In Azure App Configuration, add a key with a value using this format:

```
@Microsoft.KeyVault(SecretUri=https://<vault-name>.vault.azure.net/secrets/<secret-name>/)
```

Requires managed identity access to Key Vault and EnableKeyVault option in code.

---

8. How do you use managed identities to authenticate to Key Vault and App Configuration?

Enable system/user-assigned identity on the app. Assign roles:

- Key Vault: Key Vault Secrets User
- App Configuration: App Configuration Data Reader

In code, use DefaultAzureCredential to authenticate.

---

9. What are best practices for securing app settings and secrets?

- Never store secrets in code or config files
- Use managed identities with least privilege
- Reference secrets from Key Vault via environment/config
- Enable Key Vault logging and soft-delete

---

10. How can you audit or monitor access to secrets in Azure Key Vault?

Enable diagnostic settings to stream logs to Log Analytics.

Track:

- Secret access (AuditEvent)
- Failed attempts

Use Azure Monitor or Sentinel for alerting and analytics.

3. Implement Azure security

└ **3.2 Implement secure Azure solutions**

└ **3.2.2 Develop code that uses keys, secrets, and certificates stored in Azure Key Vault**

1. How do you retrieve a secret from Azure Key Vault using the Azure SDK?
2. How do you use a certificate from Key Vault in an HTTPS client or service?
3. How do you use Key Vault to perform cryptographic operations with stored keys?
4. What roles or permissions are needed to access keys, secrets, or certificates?
5. How do you handle secret rotation using Azure Key Vault?
6. What are the differences between software-protected and HSM-protected keys?
7. How do you access a certificate's private key from Azure Key Vault?
8. How do you manage access to Key Vault from an Azure Function or Web App?
9. What are best practices for using Key Vault in application code?
10. How do you configure Key Vault references in an ARM or Bicep deployment?

---

1. How do you retrieve a secret from Azure Key Vault using the Azure SDK?

```
var client = new SecretClient(new Uri(kvUrl), new DefaultAzureCredential());
KeyVaultSecret secret = await client.GetSecretAsync("MySecret");
string value = secret.Value;
```

Requires Key Vault Secrets User role and managed identity or credential.

---

2. How do you use a certificate from Key Vault in an HTTPS client or service?

Download the certificate as a PFX with private key:

```
var certClient = new CertificateClient(new Uri(kvUrl), new DefaultAzureCredential());
KeyVaultCertificateWithPolicy cert = await certClient.GetCertificateAsync("MyCert");
var x509 = new X509Certificate2(cert.Cer);
```

For private key use, export from a secret or use GetSecretAsync with content type application/x-pkcs12.

---

3. How do you use Key Vault to perform cryptographic operations with stored keys?

Use CryptographyClient:

```
var cryptoClient = new CryptographyClient(new Uri(keyId), new DefaultAzureCredential());
EncryptResult result = await cryptoClient.EncryptAsync(EncryptionAlgorithm.RsaOaep, data);
```

Key must allow crypto operations (e.g., encrypt, sign).

---

4. What roles or permissions are needed to access keys, secrets, or certificates?

- Secrets: Key Vault Secrets User
- Keys: Key Vault Crypto Service Encryption User, Key Vault Key User
- Certificates: Key Vault Certificates Officer
  Use RBAC or Key Vault access policies (legacy).

---

5. How do you handle secret rotation using Azure Key Vault?

- For manual rotation: update secret value and update app references.
- For automatic rotation (certs): configure lifetimeAction in certificate policy.
- Enable soft-delete and purge protection for rollback and audit.

6. What are the differences between software-protected and HSM-protected keys?

- *Software-protected*: Stored and processed in software; suitable for general use.
- *HSM-protected*: Backed by FIPS 140-2 Level 2+ compliant Hardware Security Modules; use for high-security needs like compliance-bound apps.

---

7. How do you access a certificate's private key from Azure Key Vault?

Download as a secret in PFX format:

```
var secret = await secretClient.GetSecretAsync("MyCert");
var certBytes = Convert.FromBase64String(secret.Value);
var cert = new X509Certificate2(certBytes, (string)null, X509KeyStorageFlags.Exportable);
```

Ensure certificate is imported with the private key.

---

8. How do you manage access to Key Vault from an Azure Function or Web App?

- Enable system-assigned identity
- Assign appropriate RBAC role (e.g., Key Vault Secrets User)
- Use DefaultAzureCredential in app code for auth
  No secrets stored in config needed.

---

9. What are best practices for using Key Vault in application code?

- Use DefaultAzureCredential
- Use caching to minimize latency and throttling
- Do not log secret values
- Handle retries and transient failures with SDK policies

---

10. How do you configure Key Vault references in an ARM or Bicep deployment?

Use @Microsoft.KeyVault reference in resource parameters:

```
"mySecret": {
  "reference": {
    "keyVault": {
      "id": "[resourceId('Microsoft.KeyVault/vaults', 'my-kv')]"
    },
    "secretName": "my-secret"
  }
}
```

Used to inject secrets at deploy time into app settings or parameters.

3. Implement Azure security

└ **3.2 Implement secure Azure solutions**

└ **3.2.3 Implement Managed Identities for Azure resources**

1. What are managed identities and what problem do they solve?
2. What is the difference between system-assigned and user-assigned managed identities?
3. How do you enable a managed identity on an Azure resource?
4. How do you assign RBAC roles to a managed identity?
5. How do you authenticate to Azure services using managed identities in code?
6. How do you use managed identity with Azure Key Vault?
7. How do you troubleshoot managed identity access issues?
8. How do managed identities behave during resource deletion or scaling?
9. What services support managed identities?
10. What are best practices when using managed identities in cloud apps?

---

1. What are managed identities and what problem do they solve?

Managed identities provide Azure-hosted identities for applications to access Azure resources securely without storing credentials in code or config.

---

2. What is the difference between system-assigned and user-assigned managed identities?

- *System-assigned*: Tied to the resource lifecycle; deleted with the resource.
- *User-assigned*: Standalone; reusable across multiple resources; managed separately.

---

3. How do you enable a managed identity on an Azure resource?

Via Azure CLI:

```
az webapp identity assign --name <app-name> --resource-group <rg>
```

Or in ARM/Bicep: identity: { type: 'SystemAssigned' }

---

4. How do you assign RBAC roles to a managed identity?

Use Azure CLI:

```
az role assignment create \
  --assignee <clientId-or-objectId> \
  --role <role-name> \
  --scope <resource-scope>
```

---

5. How do you authenticate to Azure services using managed identities in code?

Use DefaultAzureCredential from Azure SDK:

```
var client = new SecretClient(new Uri(kvUrl), new DefaultAzureCredential());
```

Automatically uses the managed identity of the running resource.

---

6. How do you use managed identity with Azure Key Vault?

1. Enable managed identity on the resource
2. Assign Key Vault Secrets User role to the identity at Key Vault scope
3. Use DefaultAzureCredential in code to access secrets

7. How do you troubleshoot managed identity access issues?

- Verify identity is enabled
- Confirm role assignment at correct scope
- Check az role assignment list and Key Vault diagnostics logs
- Ensure DefaultAzureCredential is used correctly in code

---

8. How do managed identities behave during resource deletion or scaling?

- *System-assigned*: Deleted when the resource is deleted
- *User-assigned*: Must be manually managed; survives resource deletion
  Scaling (e.g., in App Service) automatically reuses the same identity

---

9. What services support managed identities?

Supported in:

- App Service, Functions
- VMs, VMSS
- Logic Apps
- Azure Container Apps
- Azure Kubernetes Service (AKS)
- Azure Data Factory, and more

---

10. What are best practices when using managed identities in cloud apps?

- Prefer system-assigned for single-resource use
- Use user-assigned for cross-resource or lifecycle-independent needs
- Always use RBAC for access control
- Avoid storing credentials; rely on identity + DefaultAzureCredential

4. Monitor and troubleshoot Azure solutions

└ 4.1 Monitor and troubleshoot solutions by using Application Insights

└ 4.1.1 Monitor and analyze metrics, logs, and traces

1. What is Application Insights and what core telemetry does it collect?
2. What is the difference between metrics, logs, and traces in App Insights?
3. How do you instrument code to send telemetry to Application Insights?
4. How do you configure Application Insights in an Azure App Service or Function?
5. How do you analyze telemetry data using the Azure portal?
6. What is Kusto Query Language (KQL) and how is it used with App Insights?
7. How do you filter and visualize logs using Log Analytics?
8. How can you track request dependencies and failures?
9. How do you correlate telemetry across distributed services?
10. What are performance counters and how are they monitored?

---

1. What is Application Insights and what core telemetry does it collect?

Application Insights is an APM tool in Azure Monitor that automatically collects telemetry like requests, exceptions, dependencies, traces, custom events, and performance metrics from applications.

---

2. What is the difference between metrics, logs, and traces in App Insights?

- Metrics: Numeric time-series data (e.g., CPU, request count)
- Logs: Structured records from telemetry (e.g., exceptions, requests)
- Traces: Developer-written debug/log messages for tracing app behavior

---

3. How do you instrument code to send telemetry to Application Insights?

- Use SDKs like `Microsoft.ApplicationInsights.AspNetCore`
- Initialize `TelemetryClient` to send custom events
- Use auto-instrumentation via Application Insights extension in App Services or Azure Functions

---

4. How do you configure Application Insights in an Azure App Service or Function?

Enable App Insights from the Azure Portal by turning on "Application Insights" in the Monitoring section. The connection string / instrumentation key is injected into the app's environment variables.

---

5. How do you analyze telemetry data using the Azure portal?

Use the Application Insights blade to view built-in charts, failures, performance, and live metrics. Use "Logs" to query telemetry with KQL and "Failures" or "Performance" tabs for drill-down diagnostics.

---

6. What is Kusto Query Language (KQL) and how is it used with App Insights?

KQL is a read-only query language for analyzing telemetry in Azure Monitor. You use it in the "Logs" section of Application Insights to query tables like `requests`, `exceptions`, `dependencies`, `traces`.

7. How do you filter and visualize logs using Log Analytics?

Use the "Logs" tab in Application Insights to write KQL queries (e.g., `requests | where duration > 1s`). Use the Chart button to visualize results and pin them to Azure dashboards.

---

8. How can you track request dependencies and failures?

Use the "Application Map" and "Failures" tab in Application Insights. Dependency telemetry captures outbound calls (SQL, HTTP, etc.) with duration and result codes for diagnosing bottlenecks and errors.

9. How do you correlate telemetry across distributed services?
Application Insights uses `operation_Id` and `parent_Id` to link related telemetry. Use the "End-to-End Transaction Details" view or join telemetry tables in KQL using `operation_Id`.

---

10. What are performance counters and how are they monitored?
Performance counters like CPU, memory, and request duration are automatically collected in supported environments. View them under the "Performance" tab or query the `performanceCounters` table via KQL.

4. Monitor and troubleshoot Azure solutions

└ 4.1 Monitor and troubleshoot solutions by using Application Insights

└ 4.1.2 Implement Application Insights web tests and alerts

1. What are Application Insights web tests and their core types?
2. How do you create a URL ping test in Application Insights?
3. What are multi-step web tests and when should you use them?
4. What are availability test locations and why are they important?
5. How are alerts configured for availability tests?
6. How do you use metric-based alerts in Application Insights?
7. What is the difference between classic alerts and new Azure Monitor alerts?
8. How can you configure alerts for failed requests or dependencies?
9. How do action groups work with alerts in Application Insights?
10. What best practices apply to monitoring app availability with web tests?

---

1. What are Application Insights web tests and their core types?

Application Insights web tests are synthetic availability tests. The two core types are:

- URL ping tests: Periodically hit an endpoint to verify availability.
- Multi-step tests: Scripted tests that simulate user journeys (deprecated for new use).

---

2. How do you create a URL ping test in Application Insights?

Go to the “Availability” tab in App Insights, choose “Add test”, select “URL ping test”, configure the test name, URL, locations, frequency, success criteria, and alert settings.

---

3. What are multi-step web tests and when should you use them?

Multi-step tests use .webtest files uploaded to App Insights to simulate complex scenarios. They're useful for emulating real user workflows but are deprecated and replaced by Azure Load Testing or Playwright tests in GitHub Actions.

---

4. What are availability test locations and why are they important?

Availability tests run from multiple Azure datacenters (e.g., US West, Europe North). This validates global uptime and helps detect region-specific failures or latency issues.

---

5. How are alerts configured for availability tests?

When creating a web test, you can enable alerts on failure. Alternatively, use Azure Monitor to create an alert rule that triggers when the availabilityResults signal detects failed tests.

---

6. How do you use metric-based alerts in Application Insights?

Navigate to Azure Monitor > Alerts > New Alert Rule. Choose an App Insights resource, then select metrics like requests/failed, availabilityResults/availabilityPercentage, define condition, and assign an action group.

---

7. What is the difference between classic alerts and new Azure Monitor alerts?

Classic alerts are legacy and limited in flexibility. Azure Monitor alerts support advanced logic, dynamic thresholds, metric-based conditions, and integration with action groups.

8. How can you configure alerts for failed requests or dependencies?
Create an alert rule in Azure Monitor using Application Insights as the resource. Select signals like requests/failed or dependencies/failed, define a threshold (e.g., count > 5 in 5 mins), and link to an action group.

---

9. How do action groups work with alerts in Application Insights?
Action groups define how alerts notify users or systems. You can trigger emails, SMS, webhooks, Azure Functions, or Logic Apps. They are reusable across multiple alert rules.

---

10. What best practices apply to monitoring app availability with web tests?

- Test from multiple regions
- Set up alerts for sustained failures, not transient ones
- Use short intervals for mission-critical endpoints
- Use secure URLs (HTTPS) and validate content in the response

4. Monitor and troubleshoot Azure solutions

└ **4.1 Monitor and troubleshoot solutions by using Application Insights**

└ **4.1.3 Instrument an app or service to use Application Insights**

1. What are the main ways to instrument an application with Application Insights?
2. How do you install and configure the Application Insights SDK?
3. How does automatic vs. manual instrumentation differ?
4. How do you log custom events and metrics?
5. How do you enable Application Insights in Azure App Service or Azure Functions?
6. What is TelemetryClient and how is it used?
7. How do you set context (e.g., user ID, session ID) in telemetry?
8. How do you integrate App Insights with non-.NET apps (e.g., Node.js, Java)?
9. What is Application Insights Sampling and why is it used?
10. How do you verify instrumentation is working?

---

1. What are the main ways to instrument an application with Application Insights?

- SDK-based: Add App Insights SDK to the code (e.g., ASP.NET, Node.js)
- Agent-based: Enable from Azure Portal (App Service, Functions)
- Connection string: Set `APPLICATIONINSIGHTS_CONNECTION_STRING` in app settings

---

2. How do you install and configure the Application Insights SDK?

For ASP.NET Core:

Install `Microsoft.ApplicationInsights.AspNetCore` via NuGet, then add to `Program.cs` or `Startup.cs`:

```
builder.Services.AddApplicationInsightsTelemetry("<connection-string>");
```

---

3. How does automatic vs. manual instrumentation differ?

- Automatic: Captures requests, exceptions, dependencies without code changes
- Manual: Use `TelemetryClient` to track custom events, metrics, or exceptions explicitly in code

---

4. How do you log custom events and metrics?

Use `TelemetryClient.TrackEvent("eventName")` or `TrackMetric("metricName", value)`. You can add custom properties using dictionary parameters for additional context.

---

5. How do you enable Application Insights in Azure App Service or Azure Functions?

Go to the resource in the Azure Portal, select "Application Insights", enable monitoring, and link or create an instance. The instrumentation key or connection string is auto-injected.

---

6. What is TelemetryClient and how is it used?

`TelemetryClient` is the core class for sending custom telemetry. Instantiate it via DI or manually, then call methods like `TrackTrace`, `TrackException`, or `TrackEvent`.

---

7. How do you set context (e.g., user ID, session ID) in telemetry?

Use `TelemetryClient.Context` to set properties like `User.Id`, `Session.Id`, or `Operation.Id`. This enables correlating logs by user or session.

8. How do you integrate App Insights with non-.NET apps (e.g., Node.js, Java)?
Install the respective SDK (e.g., `applicationinsights` for Node.js, `applicationinsights-agent` for Java), then initialize with the connection string and enable auto-collection.

---

9. What is Application Insights Sampling and why is it used?
Sampling reduces telemetry volume by sending a subset of data. It helps control cost and data ingestion while preserving statistical accuracy. Enabled via `AddApplicationInsightsTelemetry` or configuration.

---

10. How do you verify instrumentation is working?
Use the Live Metrics Stream and “Logs” tab in the App Insights blade. You should see traces, requests, and dependencies appear within seconds of activity.

5. Connect to and consume Azure services and third-party services

└ **5.1 Implement API Management**

└ **5.1.1 Create an Azure API Management instance**

1. What is Azure API Management (APIM) and when should it be used?
2. What are the components of an APIM instance?
3. What are the pricing tiers of APIM and their core differences?
4. How do you provision an APIM instance using Azure CLI?
5. What are the networking options when creating an APIM instance (public vs. internal)?
6. How do you configure a custom domain during APIM creation?
7. What is the default management and developer portal behavior post-deployment?
8. What is the purpose of the publisher email and name fields?
9. What RBAC roles are relevant for managing APIM instances?
10. What are the prerequisites for deploying APIM in a VNET?

---

1. What is Azure API Management (APIM) and when should it be used?

A fully managed service to publish, secure, transform, and monitor APIs. Use APIM to expose internal or third-party services via a unified gateway with rate limiting, authentication, logging, and analytics.

---

2. What are the components of an APIM instance?

- API Gateway – Handles API calls, policies, and traffic.
- Developer Portal – Auto-generated site for API consumers.
- Management Plane – For configuring APIs and policies.
- Publisher Portal – Admin UI in Azure Portal.

---

3. What are the pricing tiers of APIM and their core differences?

- Developer – Non-production use, low-cost, no SLA.
- Basic – Entry-level prod tier, no VNET support.
- Standard – VNET support, SLA-backed, scalable.
- Premium – Multi-region, zone redundancy, higher throughput.
- Consumption – Serverless, pay-per-call, no custom domain/VNET.

---

4. How do you provision an APIM instance using Azure CLI?

```
az apim create \
  --name myapim \
  --resource-group myrg \
  --publisher-email admin@contoso.com \
  --publisher-name Contoso \
  --location eastus \
  --sku-name Developer
```

---

5. What are the networking options when creating an APIM instance (public vs. internal)?

- External (default): Public endpoint for API gateway and portals.
- Internal: Requires Premium tier, deploys into a VNET with private endpoints for secure access.

6. How do you configure a custom domain during APIM creation?
Use Azure CLI or Portal after deployment. Requires a valid certificate (Key Vault or PFX). Configure for gateway, developer portal, and management endpoints separately.

---

7. What is the default management and developer portal behavior post-deployment?
Both portals are enabled with default subdomains. Developer portal is public and customizable. Management portal is accessed via Azure Portal and not intended for external users.

---

8. What is the purpose of the publisher email and name fields?
Used in developer portal metadata and system-generated emails (e.g., confirmation, invitations). Must be valid to ensure email-based features work correctly.

---

9. What RBAC roles are relevant for managing APIM instances?

- API Management Service Contributor – Full management access.
- Reader – View-only access.
- Developer Portal Administrator – Limited to developer portal customization.

---

10. What are the prerequisites for deploying APIM in a VNET?

- Must use Premium tier.
- Requires subnet with enough IPs and correct NSG/UdR settings.
- DNS resolution must be in place for Azure services used by APIM.

5. Connect to and consume Azure services and third-party services

└ **5.1 Implement API Management**

└ **5.1.2 Create and document APIs**

1. How do you import an API into Azure API Management?
2. What API definition formats are supported for import?
3. What are the common ways to create an API manually in APIM?
4. What is the purpose of API operations and how are they defined?
5. How do you add or modify request/response parameters in APIM?
6. How does versioning work in APIM for APIs?
7. How do you document APIs using OpenAPI (Swagger) in APIM?
8. What is the role of the developer portal in API documentation?
9. How do you secure developer portal access?
10. What tools support automated API documentation publishing to APIM?

---

1. How do you import an API into Azure API Management?

Use the Azure Portal, CLI, or ARM/Bicep to import via OpenAPI, WSDL, GraphQL, or direct URL.

```
az apim api import \
  --resource-group myrg \
  --service-name myapim \
  --path myapi \
  --specification-format OpenApi \
  --specification-path ./openapi.json
```

---

2. What API definition formats are supported for import?

- OpenAPI 2.0/3.0 (JSON/YAML)
- WSDL (SOAP)
- GraphQL schemas
- Azure Functions or App Services (via App Insights tracing)

---

3. What are the common ways to create an API manually in APIM?

- Define operations manually in the Azure Portal
- Use Azure CLI or ARM templates
- Add operations under a base path, define HTTP verbs, request parameters, and response types

---

4. What is the purpose of API operations and how are they defined?

Operations define how the API is consumed (e.g., GET /users). Each includes:

- HTTP method
- URL template
- Request/response schema
- Optional policies (rate limits, auth)

---

5. How do you add or modify request/response parameters in APIM?

Via Portal or CLI:

- Define path/query/header parameters for requests
- Define expected status codes and response body schemas
- Example: Add header param in operation settings under Inbound Processing

6. How does versioning work in APIM for APIs?
Supported methods:

- Path-based (e.g., /v1/orders)
- Query string (e.g., ?api-version=1.0)
- Header-based (e.g., X-Version: 1.0)
  APIM supports version sets to group versions under a single API entity.

---

7. How do you document APIs using OpenAPI (Swagger) in APIM?
Import an OpenAPI file to auto-generate operations and documentation.
OpenAPI description appears in both the Azure Portal and developer portal.
Supports Swagger UI rendering in developer portal.

---

8. What is the role of the developer portal in API documentation?
Provides a public or private interface for users to:

- Discover and test APIs
- View request/response details
- Access API keys
- Read autogenerated documentation

---

9. How do you secure developer portal access?

- Enable require sign-in in portal settings
- Configure identity providers (AAD, Facebook, etc.)
- Assign users to products with subscription-based access

---

10. What tools support automated API documentation publishing to APIM?

- Swagger/OpenAPI CLI
- Postman → export to OpenAPI
- Azure DevOps/GitHub Actions → automate import via `az apim api import`
- Supports CI/CD for keeping API definitions in sync

5. Connect to and consume Azure services and third-party services

└ **5.1 Implement API Management**

└ **5.1.3 Configure access to APIs**

1. What authentication mechanisms are supported by Azure API Management (APIM)?
2. How do you secure APIs using subscription keys?
3. How do you configure OAuth 2.0 authentication with APIM?
4. How to configure a client application to call an APIM-secured API using a bearer token?
5. How do you restrict API access using IP filtering in APIM?
6. What is the role of policies in controlling access to APIs?
7. How can you enforce rate limits and quotas per subscription in APIM?
8. How do you enable CORS in API Management?
9. What is the difference between product-level and API-level access control?
10. How do you use managed identities to call APIs behind APIM securely?

---

1. What authentication mechanisms are supported by Azure API Management (APIM)?

- Subscription key
- OAuth 2.0 / OpenID Connect
- JWT validation
- Client certificates
- Managed identities

---

2. How do you secure APIs using subscription keys?

- Add APIs to a product.
- Require subscription on the product.
- Each caller must pass `Ocp-Apim-Subscription-Key` in header or query.

---

3. How do you configure OAuth 2.0 authentication with APIM?

- Register APIM as a client app in Microsoft Entra ID (or other provider).
- Configure OAuth 2.0 settings in APIM (under security tab).
- Set `validate-jwt` policy in inbound section of the API to enforce token validation.

---

4. What are the steps to configure a client application to call an APIM-secured API using a bearer token?

1. Register the client app in Entra ID.
2. Acquire token using MSAL or ADAL libraries.
3. Call the API with `Authorization: Bearer <token>` header.
4. Ensure APIM has a `validate-jwt` policy matching token settings.

---

5. How do you restrict API access using IP filtering in APIM?

- Use the `check-header` or `check-ip` policy in the inbound policy section.
- Example:

```
<check-header name="X-Forwarded-For" failed-check-httpcode="403" failed-check-error-message="Access denied">
  <value>203.0.113.1</value>
</check-header>
```

6. What is the role of policies in controlling access to APIs?

- Policies define request/response behavior at runtime.
- Used to enforce security (e.g., validate-jwt, check-header), rate limits, IP restrictions, CORS, etc.
- Applied at inbound, backend, outbound, or on error sections.

---

7. How can you enforce rate limits and quotas per subscription in APIM?

- Use built-in rate-limit and quota policies.
- Define policies in product or API scope.
- Example:

```
<rate-limit calls="10" renewal-period="60" />
<quota calls="1000" renewal-period="604800" />
```

---

8. How do you enable CORS in API Management?

- Add the cors policy in the inbound section.
- Example:

```
<cors allow-credentials="true">
  <allowed-origins><origin>*</origin></allowed-origins>
  <allowed-methods><method>GET</method></allowed-methods>
</cors>
```

---

9. What is the difference between product-level and API-level access control?

- Product-level: Controls who can access any API within the product using subscriptions.
- API-level: Policies or restrictions applied to individual APIs regardless of product membership.

---

10. How do you use managed identities to call APIs behind APIM securely?

- Enable system-assigned or user-assigned identity on APIM.
- Grant API backend (e.g., Azure Function) the necessary role (e.g., Function App Contributor).
- Use authentication-managed-identity policy in outbound call:

```
<authentication-managed-identity resource="https://<resource>" />
```

5. Connect to and consume Azure services and third-party services

└ **5.1 Implement API Management**

└ **5.1.4 Implement policies for APIs**

1. What are API Management policies and where can they be applied?
2. How do you use the set-header policy to manipulate request/response headers?
3. How does the validate-jwt policy work for token validation?
4. How do you apply rate limiting using the rate-limit policy?
5. What's the difference between quota and rate-limit policies?
6. How do you rewrite URLs using rewrite-uri or set-backend-service?
7. How do you handle conditional logic in policies using choose and when?
8. How can you transform request or response bodies using set-body?
9. What tools are used to author, test, and debug APIM policies?
10. What are best practices for policy organization and reuse?

---

1. What are API Management policies and where can they be applied?

- XML-based logic executed at runtime.
- Applied at four scopes: inbound, backend, outbound, and on-error.
- Can be configured at global, product, API, or operation levels.

---

2. How do you use the `set-header` policy to manipulate request/response headers?

- Adds or updates headers.
- Example:

```
<set-header name="X-Custom-Header" exists-action="override">
  <value>my-value</value>
</set-header>
```

---

3. How does the `validate-jwt` policy work for token validation?

- Validates JWT tokens in Authorization header.
- Requires configuration of issuer, audience, and signing keys.
- Example:

```
<validate-jwt header-name="Authorization" require-scheme="Bearer">
  <openid-config url="https://login.microsoftonline.com/<tenant>/v2.0/.well-known/openid-configuration" />
  <required-claims><claim name="aud"><value>api-client-id</value></claim></required-claims>
</validate-jwt>
```

---

4. How do you apply rate limiting using the `rate-limit` policy?

- Restricts number of calls in a time window.
- Example:

```
<rate-limit calls="100" renewal-period="60" />
```

  Limits client to 100 calls per minute.

---

5. What's the difference between `quota` and `rate-limit` policies?

- `rate-limit`: short-term throttle (e.g., per minute).
- `quota`: long-term usage limit (e.g., daily/weekly).
- Both can be used together for layered control.

6. How do you rewrite URLs using `rewrite-uri` or `set-backend-service`?

- `rewrite-uri`: Changes request path.
  Example:
  ```
  <rewrite-uri template="/v2/resource" />
  ```
- `set-backend-service`: Redirects to a different backend.
  Example:
  ```
  <set-backend-service base-url="https://api.contoso.com/v2" />
  ```

---

7. How do you handle conditional logic in policies using `choose` and `when`?

- Enables branching logic based on conditions.
  Example:
  ```
  <choose>
    <when condition="@(context.Request.Headers["x-version"] == "v2")">
      <set-backend-service base-url="https://api-v2.contoso.com" />
    </when>
    <otherwise>
      <set-backend-service base-url="https://api-v1.contoso.com" />
    </otherwise>
  </choose>
  ```

---

8. How can you transform request or response bodies using `set-body`?

- Overwrites the payload.
  Example (JSON):
  ```
  <set-body>@("{\"message\": \"Access denied\"}")</set-body>
  ```
- Can also use Liquid templates for dynamic content.

---

9. What tools are used to author, test, and debug APIM policies?

- Azure Portal policy editor with IntelliSense.
- Test console in Azure Portal.
- Developer portal (limited testing).
- Trace/debug by enabling "trace" and reviewing trace logs in the test console.

---

10. What are best practices for policy organization and reuse?

- Keep logic centralized at product/API level when possible.
- Use named values for reusable values.
- Document policy use with `<comment>` blocks.
- Avoid deep nesting for readability.

5. Connect to and consume Azure services and third-party services

└ 5.2 Develop event-based solutions

└ 5.2.1 Implement solutions that use Azure Event Grid

1. What is Azure Event Grid and what are its primary use cases?
2. What are the key components of Event Grid (events, topics, event subscriptions)?
3. How do you create a custom topic and event subscription via Azure CLI?
4. What event sources can trigger Event Grid?
5. What are the supported event handlers (destinations)?
6. How do you secure Event Grid with authentication and authorization?
7. How do you filter events in Event Grid subscriptions?
8. How do you enable dead-lettering for undeliverable events?
9. How do you validate Event Grid event schema in consumers?
10. How do you troubleshoot failed event deliveries?
11. Is Event Grid push or pull-based?
12. What delivery guarantees does Event Grid provide?
13. What is the default schema of an Event Grid event?
14. How do you choose between Event Grid, Event Hubs, and Service Bus?
15. How are retries and error handling managed in Event Grid?

---

1. What is Azure Event Grid and what are its primary use cases?

Azure Event Grid is a fully managed event routing service that enables reactive programming using events from Azure services or custom sources.
Use cases: Event-driven architecture, serverless automation, resource provisioning notifications, microservices decoupling.

---

2. What are the key components of Event Grid?

- Event Sources: Where events originate (e.g., Blob Storage, Resource Groups).
- Topics: Channels where events are sent.
- Event Subscriptions: Define how to handle events (e.g., send to Function, Webhook).
- Event Handlers: Endpoints that process events (e.g., Azure Function, Logic Apps).

---

3. How do you create a custom topic and event subscription via Azure CLI?

```
az eventgrid topic create --name mytopic --resource-group myrg --location eastus
az eventgrid event-subscription create --name mysub --source-resource-id
/subscriptions/<id>/resourceGroups/myrg/providers/Microsoft.EventGrid/topics/mytopic --endpoint <url>
```

---

4. What event sources can trigger Event Grid?

- Native: Blob Storage, Resource Groups, Event Hubs, IoT Hub, Media Services, etc.
- Custom sources: Via custom topics.
- Third-party: via webhook-compatible services.

---

5. What are the supported event handlers (destinations)?

- Azure Function
- Logic Apps
- Event Hubs
- Webhooks (HTTP/S endpoint)
- Service Bus Queue/Topic

6. How do you secure Event Grid with authentication and authorization?

- Event delivery: Signed with validationCode or Azure Active Directory token.
- Inbound auth: Use Azure RBAC for publishing to topics.
- Outbound auth: Use Webhook validation and shared access keys for endpoints.

---

7. How do you filter events in Event Grid subscriptions?

Use `--included-event-types` or advanced filters:

```
--advanced-filter data.subject StringBeginsWith "/blobServices/default/containers/images"
```

Filters reduce noise and only forward relevant events.

---

8. How do you enable dead-lettering for undeliverable events?

Set `--deadletter-destination` when creating a subscription, usually to a Blob Storage container:

```
--deadletter-destination blobcontainer:<storage-account>/<container>
```

This stores failed events for later review.

---

9. How do you validate Event Grid event schema in consumers?

Consumers must handle a **validationEvent** initially.

Respond to the `eventType = Microsoft.EventGrid.SubscriptionValidationEvent` by echoing back `data.validationCode`.

---

10. How do you troubleshoot failed event deliveries?

- Enable dead-lettering.
- Check metrics in Azure Monitor.
- Use `az eventgrid event-subscription show` to inspect subscription state.
- Inspect logs at the handler side (e.g., Azure Function failures).

---

11. Is Event Grid push or pull-based?

Event Grid is push-only — it pushes events to subscribers via HTTP POST; consumers must expose endpoints to receive events.

---

12. What delivery guarantees does Event Grid provide?

At-least-once delivery.

Events are retried with exponential backoff for up to 24 hours if the destination is unavailable.

---

13. What is the default schema of an Event Grid event?

Default schema fields include:

- `id`, `eventType`, `subject`, `data`, `eventTime`, `dataVersion`, `metadataVersion`.
  Custom topics use this schema unless configured for CloudEvents v1.0.

---

14. How do you choose between Event Grid, Event Hubs, and Service Bus?

- Event Grid: Discrete events, fan-out, serverless, push-only.
- Event Hubs: Telemetry, streaming data, high-throughput ingestion.
- Service Bus: Reliable message delivery with ordering and sessions.

---

15. How are retries and error handling managed in Event Grid?

- Retries: Exponential backoff for transient failures.
- Permanent failure: After 24 hours or persistent errors, events are dead-lettered if configured.
- Subscriber should return HTTP 2xx for success, otherwise it triggers retries.

5. Connect to and consume Azure services and third-party services
└ 5.2 Develop event-based solutions
└ 5.2.2 Implement solutions that use Azure Event Hub

1. What is Azure Event Hubs and what is it used for?
2. How does Event Hubs differ from Event Grid and Service Bus?
3. What are the key components of Event Hubs (namespace, event hub, partition, consumer group)?
4. How is data transmitted and consumed in Event Hubs (push/pull)?
5. What are partitions and why are they important in Event Hubs?
6. What is a consumer group and how is it used?
7. What are the differences between Standard, Basic, and Dedicated tiers?
8. How do you send events to Event Hubs using Azure SDK or Azure CLI?
9. How do you consume events from Event Hubs using EventProcessorClient?
10. What is checkpointing and why is it needed?
11. How do you implement checkpointing with Azure Blob Storage?
12. How do you authenticate and authorize access to Event Hubs (Shared Access Policies vs Azure AD)?
13. What are throughput units and how do they impact performance?
14. What are the retention and capture capabilities of Event Hubs?
15. How do Event Hubs integrate with Azure Stream Analytics and Azure Functions?

---

1. What is Azure Event Hubs and what is it used for?
Event Hubs is a data streaming platform and event ingestion service for high-throughput data pipelines.
Use cases: telemetry ingestion, app/event logging, IoT data streams.

---

2. How does Event Hubs differ from Event Grid and Service Bus?

- Event Hubs: Streaming, pull-based, low-latency, high-volume.
- Event Grid: Push-based, lightweight eventing.
- Service Bus: Reliable messaging with ordering and sessions.

---

3. What are the key components of Event Hubs?

- Namespace: Container for event hubs.
- Event Hub: Stream that holds data.
- Partition: Log stream shard for parallelism.
- Consumer Group: Independent view for multiple consumers.

---

4. How is data transmitted and consumed in Event Hubs?
Data is pushed by producers and pulled by consumers. Consumers use SDKs to poll events per partition.

---

5. What are partitions and why are they important?
Partitions enable horizontal scaling and parallel processing.
Events with the same partition key always go to the same partition to preserve order.

---

6. What is a consumer group and how is it used?
A consumer group is a view of an event hub.
Each consumer group has its own state, allowing multiple apps to read independently.

7. What are the differences between Standard, Basic, and Dedicated tiers?

- Basic: Single consumer group, limited features.
- Standard: Multiple consumer groups, capture, 1–20 throughput units.
- Dedicated: Reserved capacity, higher scale, 90+ MB/s ingress.

---

8. How do you send events to Event Hubs using Azure SDK or CLI?

CLI:

```
az eventhubs eventhub send --name <hub> --namespace-name <ns> --resource-group <rg> --message "event data"
```

.NET SDK:

```
await producer.SendAsync(new EventData(Encoding.UTF8.GetBytes("event data")));
```

---

9. How do you consume events from Event Hubs using EventProcessorClient?

Use Azure.Messaging.EventHubs.Processor:

```
var processor = new EventProcessorClient(blobContainerClient, "$Default", connStr, hubName);
processor.ProcessEventAsync += async args => { /* handle event */ };
await processor.StartProcessingAsync();
```

---

10. What is checkpointing and why is it needed?

Checkpointing stores the last successfully processed event offset.
It ensures events aren't reprocessed after restarts and supports fault tolerance.

---

11. How do you implement checkpointing with Azure Blob Storage?

Use EventProcessorClient with a Blob container as storage:

```
new EventProcessorClient(blobContainerClient, "$Default", eventHubConnectionString, eventHubName)
```

---

12. How do you authenticate and authorize access to Event Hubs?

- Shared Access Policies: Use connection string with Send, Listen, or Manage claims.
- Azure AD: Use Azure Identity SDK + RBAC on Event Hub namespace or resource.

---

13. What are throughput units and how do they impact performance?

Each Throughput Unit (TU) allows:

- 1 MB/s ingress or 1000 events/s
- 2 MB/s egress
  You can purchase 1–20 TUs in Standard tier to scale performance.

---

14. What are the retention and capture capabilities of Event Hubs?

- Retention: Configurable up to 7 days (Standard) or 90 days (Dedicated).
- Capture: Automatically stores data to Blob Storage or Data Lake in AVRO format.

---

15. How do Event Hubs integrate with Azure Stream Analytics and Azure Functions?

- Stream Analytics: Direct input via Event Hub; use for real-time queries and dashboards.
- Azure Functions: Use Event Hub trigger to react to events:

```
[EventHubTrigger("hubname", Connection = "EventHubConnection")] EventData[] events
```

5. Connect to and consume Azure services and third-party services
└ **5.3 Develop message-based solutions**
└ **5.3.1 Implement solutions that use Azure Service Bus**

1. What is Azure Service Bus and what are its use cases?
2. What is the difference between Service Bus, Event Hubs, and Event Grid?
3. What messaging models does Service Bus support?
4. What is the difference between standard and premium tiers in Service Bus?
5. What are message sessions and when are they needed?
6. What is dead-lettering and how is it configured?
7. How do you send messages using Azure SDK or CLI?
8. How do you receive messages from a queue?
9. What is peek-lock vs receive-and-delete mode?
10. How do you implement message deferral and why?
11. What is auto-forwarding in Service Bus?
12. How do you configure filters in topic subscriptions?
13. How do you authenticate and authorize access to Service Bus?
14. What are delivery and retry behaviors in Service Bus?
15. How do Service Bus and Azure Functions integrate?

---

1. What is Azure Service Bus and what are its use cases?
A fully managed message broker for enterprise apps.
Use cases: decoupled microservices, order processing, transactions, retries, delayed delivery.

---

2. What is the difference between Service Bus, Event Hubs, and Event Grid?
- **Service Bus**: Reliable messaging, ordering, sessions, dead-lettering.
- **Event Hubs**: High-throughput telemetry streaming.
- **Event Grid**: Lightweight, push-based eventing.

---

3. What messaging models does Service Bus support?
- **Queues**: Point-to-point (1 sender → 1 receiver).
- **Topics/Subscriptions**: Publish-subscribe (1 sender → multiple filtered subscribers).

---

4. What is the difference between standard and premium tiers in Service Bus?
- **Standard**: Basic features, shared resources, limited performance.
- **Premium**: Dedicated compute, faster, supports VNET, encryption, higher scale.

---

5. What are message sessions and when are they needed?
Sessions group related messages for ordered processing.
Required when strict message ordering per entity is needed (e.g., per user/cart).

---

6. What is dead-lettering and how is it configured?
Dead-letter queue (DLQ) stores undeliverable messages (e.g., max delivery attempts reached).
Enable via `EnableDeadLetteringOnMessageExpiration` or use `Abandon`/`DeadLetter` in SDK.

7. How do you send messages using Azure SDK or CLI?

.NET SDK:

```
await sender.SendMessageAsync(new ServiceBusMessage("Hello"));
```

CLI:

```
az servicebus message send --queue-name myqueue --namespace-name myns --resource-group myrg --body "msg"
```

---

8. How do you receive messages from a queue?

.NET SDK:

```
var msg = await receiver.ReceiveMessageAsync();
await receiver.CompleteMessageAsync(msg);
```

---

9. What is peek-lock vs receive-and-delete mode?

- **Peek-lock** (default): Two-phase – lock then explicitly complete.
- **Receive-and-delete**: One-shot; message is deleted on receive.

---

10. How do you implement message deferral and why?

Deferred messages are postponed for later retrieval by sequence number.
Useful when processing must be delayed:

```
var deferred = await receiver.ReceiveDeferredMessageAsync(sequenceNumber);
```

---

11. What is auto-forwarding in Service Bus?

Automatically forwards messages from one queue/topic to another.
Used for message routing, chaining processing steps.

---

12. How do you configure filters in topic subscriptions?

Use SQL-based filters:

```
az servicebus topic subscription rule create \
  --name highPriority \
  --subscription-name mysub \
  --topic-name mytopic \
  --filter-sql-expression "priority = 'high'"
```

---

13. How do you authenticate and authorize access to Service Bus?

- **Shared Access Signature (SAS)**: via policy keys and connection string.
- **Azure AD**: via RBAC roles like `Azure Service Bus Data Sender`.

---

14. What are delivery and retry behaviors in Service Bus?

- Default: 10 delivery attempts, exponential backoff.
- Messages that fail retry go to DLQ if enabled.
- Customize via `MaxDeliveryCount` and `LockDuration`.

---

15. How do Service Bus and Azure Functions integrate?

Use Service Bus trigger:

```
[ServiceBusTrigger("myqueue", Connection = "ServiceBusConnection")] string msg
```

5. Connect to and consume Azure services and third-party services
└ 5.3 Develop message-based solutions
└ 5.3.2 – Implement solutions that use Azure Queue Storage queues

1. What is Azure Queue Storage and what are its typical use cases?
2. How does Azure Queue Storage compare to Azure Service Bus queues?
3. What are the key components of an Azure Storage queue?
4. How are messages added to and retrieved from a queue?
5. What is the default behavior when retrieving messages from a queue?
6. What is message visibility timeout and how does it affect processing?
7. How do you handle message retries and poison messages in Azure Queue Storage?
8. How do you create and manage queues using Azure CLI or SDK?
9. How do you secure access to Azure Queue Storage (SAS tokens, RBAC, shared keys)?
10. What are the message size limits and encoding requirements?
11. How do you configure and use Base64 encoding with queue messages?
12. What are best practices for processing at scale with Queue Storage?
13. How do you integrate Azure Queue Storage with Azure Functions?
14. How do you monitor and troubleshoot Azure Queue Storage usage?
15. What are common limitations of Azure Queue Storage compared to other messaging services?

---

1. What is Azure Queue Storage and what are its typical use cases?
A simple, cloud-based message queueing service for decoupling components.
Use cases: task queues, background jobs, retry buffers.

---

2. How does Azure Queue Storage compare to Azure Service Bus queues?

- **Queue Storage**: Basic FIFO queues, lightweight, no ordering guarantees, no DLQ, no sessions.
- **Service Bus**: Enterprise-grade, supports DLQ, ordering, transactions, filtering.

---

3. What are the key components of an Azure Storage queue?

- **Queue**: A named message container.
- **Message**: Max 64 KB of Base64-encoded text.
- **Visibility timeout**: Temporary hide period after retrieval.

---

4. How are messages added to and retrieved from a queue?
Add: `PutMessageAsync()`
Get: `GetMessagesAsync()`
Delete: `DeleteMessageAsync()` after processing.

---

5. What is the default behavior when retrieving messages from a queue?
Messages are hidden for the visibility timeout duration.
If not deleted within that time, they become visible again for reprocessing.

---

6. What is message visibility timeout and how does it affect processing?
Defines how long a retrieved message stays hidden.
If not deleted in time, it is returned to the queue for another attempt.

7. How do you handle message retries and poison messages?

Track dequeue count using `DequeueCount`.

Manually delete or move messages to a custom "poison" queue if `DequeueCount` exceeds threshold.

---

8. How do you create and manage queues using Azure CLI or SDK?

CLI:

```
az storage queue create --name myqueue --account-name mystorage
```

.NET SDK:

```
await queueClient.CreateIfNotExistsAsync();
await queueClient.SendMessageAsync("my message");
```

---

9. How do you secure access to Azure Queue Storage?

- **Shared keys**: Full access via storage account key.
- **SAS tokens**: Scoped, time-limited access.
- **RBAC**: Role-based access via Azure AD (Blob/Queue Contributor roles).

---

10. What are the message size limits and encoding requirements?

- Max message size: **64 KB** (48 KB when Base64 encoded by default).
- Messages must be **text-based** (UTF-8 or Base64).

---

11. How do you configure and use Base64 encoding with queue messages?

By default, SDK encodes messages as Base64.

Can opt out using:

```
new QueueClient(..., new QueueClientOptions { MessageEncoding = QueueMessageEncoding.None })
```

---

12. What are best practices for processing at scale with Queue Storage?

- Use multiple consumer instances.
- Tune visibility timeout per job duration.
- Monitor queue length for autoscaling triggers.

---

13. How do you integrate Azure Queue Storage with Azure Functions?

Use the `QueueTrigger` binding:

```
[FunctionName("ProcessQueue")]
public void Run([QueueTrigger("myqueue", Connection = "StorageConnection")] string message)
```

---

14. How do you monitor and troubleshoot Azure Queue Storage usage?

- Enable diagnostics in Azure Monitor.
- Use metrics: `QueueLength`, `DequeueCount`, `Ingress`, `Egress`.
- View activity logs and errors in Log Analytics if enabled.

---

15. What are common limitations of Azure Queue Storage compared to other messaging services?

- No message ordering guarantee.
- No native dead-letter queues.
- No message filtering or pub/sub.
- Limited throughput and features compared to Service Bus.