**Develop Azure Compute Solutions**  
 └─ **1.1 Implement Containerized Solutions**  
   └─ **1.1.2 Publish an image to Azure Container Registry**

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

**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.

**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

**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.

**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.

**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>

**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.

**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>

**What happens if the permission is 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.

**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).

**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

**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

**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)