Here are some **Azure DevOps interview questions and answers** tailored to your 2 years of experience, focusing on core concepts, pipelines, CI/CD, and related tools.

**1. General Azure DevOps Questions**

**Q1: What is Azure DevOps?** **A:**  
Azure DevOps is a set of development tools provided by Microsoft for planning, developing, testing, deploying, and managing software projects. It offers services like:

* **Azure Repos**: Git repositories for version control.
* **Azure Pipelines**: CI/CD pipelines for build and release automation.
* **Azure Boards**: Agile project management and tracking tools.
* **Azure Test Plans**: Manual and exploratory testing.
* **Azure Artifacts**: Package management.

**Q2: What are the main benefits of using Azure DevOps?** **A:**

1. Integrated platform for the entire development lifecycle.
2. Supports multiple languages and platforms.
3. Seamless integration with third-party tools (e.g., Jenkins, Kubernetes, Docker).
4. Scalable for teams of any size.
5. Strong security with Azure Active Directory.

**Q3: What is the difference between a build pipeline and a release pipeline?** **A:**

* **Build Pipeline**:
  + Automates tasks like code compilation, running tests, and creating build artifacts.
  + Focused on **continuous integration**.
* **Release Pipeline**:
  + Automates deployment of artifacts to different environments (e.g., QA, staging, production).
  + Focused on **continuous delivery**.

**2. Azure Pipelines Questions**

**Q4: What are the different triggers in Azure Pipelines?** **A:**  
Triggers define when a pipeline should run:

1. **Continuous Integration (CI) Trigger**:
   * Automatically starts the pipeline when changes are pushed to the repository.
   * Configured using trigger: in YAML.
2. **Scheduled Trigger**:
   * Runs the pipeline at specific times.
   * Example:
   * schedules:
   * - cron: "0 0 \* \* \*"
3. **Pull Request Trigger**:
   * Runs when a pull request is created or updated.
4. **Pipeline Trigger**:
   * Starts a pipeline based on the completion of another pipeline.

**Q5: How do you secure sensitive information like passwords or API keys in Azure Pipelines?** **A:**  
Use **Azure Pipeline Secrets**:

1. Store secrets in **Azure Key Vault**.
2. Link the Key Vault to the pipeline using the AzureKeyVault@2 task.
3. Use variable groups or pipeline variables marked as secret.

**Q6: Explain the structure of a basic YAML pipeline.** **A:**  
A minimal YAML pipeline example:

trigger:

branches:

include:

- main

pool:

vmImage: 'ubuntu-latest'

steps:

- task: UsePythonVersion@0

inputs:

versionSpec: '3.x'

- script: |

echo "Installing dependencies"

pip install -r requirements.txt

displayName: 'Install Dependencies'

- script: |

echo "Running tests"

pytest

displayName: 'Run Tests'

**3. CI/CD and Automation Questions**

**Q7: What is the purpose of an artifact in Azure Pipelines?** **A:**  
Artifacts are the outputs of a pipeline (e.g., compiled binaries, Docker images). They:

* Are stored for downstream stages.
* Facilitate deployment and testing in release pipelines.
* Example: Publishing an artifact:
* steps:
* - task: PublishBuildArtifacts@1
* inputs:
* PathtoPublish: '$(Build.ArtifactStagingDirectory)'
* ArtifactName: 'drop'

**Q8: How do you handle multi-environment deployments in Azure Pipelines?** **A:**

* Use stages to define separate environments (e.g., QA, staging, production).
* Example YAML:
* stages:
* - stage: QA
* jobs:
* - job: DeployToQA
* steps:
* - script: echo "Deploying to QA"
* - stage: Production
* dependsOn: QA
* jobs:
* - job: DeployToProd
* steps:
* - script: echo "Deploying to Production"

**Q9: What is the difference between continuous integration and continuous deployment?** **A:**

* **Continuous Integration (CI)**:
  + Automates code integration and testing.
  + Ensures changes do not break the codebase.
* **Continuous Deployment (CD)**:
  + Automates the deployment of code to production after passing CI stages.
  + Requires robust automated testing.

**4. Docker and Kubernetes Questions**

**Q10: How do you integrate Docker with Azure Pipelines?** **A:**  
Use the Docker@2 task to build, push, and run Docker images. Example:

steps:

- task: Docker@2

inputs:

containerRegistry: 'MyDockerRegistry'

repository: 'my-app'

command: 'buildAndPush'

Dockerfile: 'Dockerfile'

tags: '$(Build.BuildId)'

**Q11: What is the role of Kubernetes in DevOps?** **A:**  
Kubernetes is an orchestration tool for managing containerized applications. It:

* Automates deployment, scaling, and maintenance.
* Provides load balancing and self-healing.
* Works seamlessly with CI/CD tools for automated deployments.

**5. Miscellaneous Questions**

**Q12: What are Azure Boards used for?** **A:**  
Azure Boards is a project management tool for:

* Tracking work items, bugs, and user stories.
* Planning sprints and managing tasks in Agile projects.
* Integrating with Git to link commits and pull requests to work items.

**Q13: How do you implement rollback in Azure DevOps?** **A:**

* Use pipeline stages to define a rollback step.
* Example rollback command for Kubernetes:
* kubectl rollout undo deployment/my-app

**6. Scenario-Based Questions**

**Q14: A pipeline fails during deployment. How do you debug it?** **A:**

1. Check pipeline logs for errors.
2. Use System.Debug variable to enable detailed logs:
3. variables:
4. System.Debug: true
5. Review build artifacts and deployment configurations.
6. Test the failing stage in isolation.

Would you like more advanced questions or specific scenarios? Let me know!