3. Build and release pipelines
└─ **3.2 Design and implement pipelines**
└─ **3.2.4 Pipeline triggers, YAML usage, job execution**

1. What types of pipeline triggers are supported in Azure Pipelines?
2. How do you define continuous integration (CI) and pull request (PR) triggers in YAML?
3. How do you disable or override triggers for specific branches or paths?
4. What is the purpose of paths and paths-ignore in triggers?
5. How do you set up scheduled (cron) triggers in a YAML pipeline?
6. How do you structure multi-stage YAML pipelines?
7. What are jobs and how do you define them in YAML?
8. How do you configure job dependencies and execution order?
9. How do you run jobs in parallel or sequentially?
10. What are key best practices for reliable job execution in Azure Pipelines YAML?

1. What types of pipeline triggers are supported in Azure Pipelines?

- Continuous integration (CI) triggers
- Pull request (PR) triggers
- Scheduled triggers (cron)
- Pipeline triggers (triggered by another pipeline)

2. How do you define continuous integration (CI) and pull request (PR) triggers in YAML?

Use the trigger and pr keys with properly indented branches:

```
trigger:
  branches:
    include:
      - main
      - develop
pr:
  branches:
    include:
      - main
      - develop
```

3. How do you disable or override triggers for specific branches or paths?

Set trigger: none or use exclude: under branches.

```
trigger: none

trigger:
  branches:
    exclude:
      - experimental
```

4. What is the purpose of paths and paths-ignore in triggers?

- paths: Run pipeline only when files matching the patterns are changed
- paths-ignore: Ignore runs if only files matching these patterns are changed

```
trigger:
  branches:
    include:
      - main
  paths:
    include:
      - src/**
  paths-ignore:
    - docs/**
```

5. How do you set up scheduled (cron) triggers in a YAML pipeline?

Use the schedules key:

```
schedules:
  - cron: "0 0 * * 1"
   displayName: "Weekly build"
   branches:
     include:
       - main
   always: true
```

6. How do you structure multi-stage YAML pipelines?

Use the stages key to define multiple stages, each with its own jobs.

```
stages:
  - stage: Build
   jobs:
     - job: BuildJob
     steps:
       - script: echo Building
  - stage: Deploy
   jobs:
     - job: DeployJob
     steps:
       - script: echo Deploying
```

7. What are jobs and how do you define them in YAML?

Jobs are collections of steps run on agents. Define under jobs.

```
jobs:
  - job: TestJob
   steps:
     - script: echo Running tests
```

8. How do you configure job dependencies and execution order?

Use the dependsOn keyword. By default, jobs run in parallel unless dependencies are set.

```
jobs:
  - job: Build
   steps:
     - script: echo Build
  - job: Test
   dependsOn: Build
   steps:
     - script: echo Test
```

9. How do you run jobs in parallel or sequentially?

- Jobs run in parallel by default
- Use `dependsOn` to run jobs sequentially
- No `dependsOn` means jobs start together

---

10. What are key best practices for reliable job execution in Azure Pipelines YAML?

- Use clear job/stage names
- Define explicit dependencies with `dependsOn`
- Isolate steps to minimize side effects
- Use variables and templates for reusability
- Monitor pipeline runs for failures and flaky jobs