3. Build and release pipelines

└ 3.7 Maintain and optimize pipelines

└ 3.7.1 Monitor pipeline health (failures,duration)

---

1. What tools in Azure DevOps and GitHub Actions are used to monitor pipeline failures and durations?
2. How do you configure pipeline run analytics and dashboards in Azure DevOps?
3. What are the key metrics to track for pipeline health?
4. How can alerts be set up for failed or long-running pipelines?
5. How do you investigate and troubleshoot failed pipeline runs?
6. How can you identify flaky tests or unreliable steps in pipelines?
7. What practices improve visibility of pipeline duration trends?
8. How do you optimize notification settings for pipeline health?
9. What integrations support advanced monitoring (e.g., Azure Monitor, Log Analytics)?
10. How do you audit historical pipeline performance and failure trends?

---

1. What tools in Azure DevOps and GitHub Actions are used to monitor pipeline failures and durations?

- *Azure DevOps* uses built-in Pipeline Run history, Analytics views, and Dashboards.
- *GitHub Actions* provides Workflow Run history and summary views. Both support integration with *Azure Monitor* and *Log Analytics*.

---

2. How do you configure pipeline run analytics and dashboards in Azure DevOps?

Use *Azure DevOps* Analytics extension and built-in widgets to display pipeline duration, success rate, and failure trends on Dashboards. Configure custom queries in Analytics views for detailed tracking.

---

3. What are the key metrics to track for pipeline health?

- Track pipeline failure rate,
- average and maximum run duration,
- time to fix failures,
- frequency of runs,
- and number of flaky tests or steps.

---

4. How can alerts be set up for failed or long-running pipelines?

- In *Azure DevOps*, configure Alerts under *Project Settings > Notifications* for build or release failures and time thresholds.
- In *GitHub Actions*, use `workflow_dispatch` or third-party integrations to trigger notifications for failures or delays.

---

5. How do you investigate and troubleshoot failed pipeline runs?

Review detailed run logs, error messages, and failed step outputs. Use build and release timeline views to identify where failures occur and correlate with recent changes or configuration updates.

---

6. How can you identify flaky tests or unreliable steps in pipelines?

Analyze test run history for intermittent failures.

Use *Azure DevOps* Test Analytics or *GitHub Actions* Test Reporter to flag non-deterministic or frequently failing tests and steps.

7. What practices improve visibility of pipeline duration trends?

- Regularly review analytics dashboards, set up charts for historical run times, and compare against baselines.
- Visualize trends using built-in or Power BI dashboards.

8. How do you optimize notification settings for pipeline health?

Configure notifications to target only relevant teams/users and filter on critical events like failures or abnormal durations. Avoid excessive notifications to reduce alert fatigue.

9. What integrations support advanced monitoring (e.g., Azure Monitor, Log Analytics)?

Connect *Azure DevOps* pipelines to *Azure Monitor* and *Log Analytics* using Diagnostic Settings or REST APIs. Export pipeline metrics for advanced querying, alerting, and dashboarding.

10. How do you audit historical pipeline performance and failure trends?

- Use *Azure DevOps* Analytics views or *Power BI* to query and visualize pipeline history, identifying long-term trends in duration, failure rate, and reliability.
- Archive logs for compliance audits if required.