3. Build and release pipelines
└ **3.7 Maintain and optimize pipelines**
└ **3.7.2 Optimize: Cost, Time, Reliability, Performance**

1. What strategies reduce pipeline execution costs in Azure DevOps and GitHub Actions?
2. How do you minimize pipeline execution time?
3. What approaches improve pipeline reliability?
4. How can parallelism and agent management be used to optimize performance?
5. What best practices reduce resource consumption in pipelines?
6. How do you cache dependencies to accelerate builds?
7. How do you monitor and control agent utilization?
8. What methods reduce wait times for pipeline approvals and manual interventions?
9. How do you ensure pipelines are robust against transient infrastructure failures?
10. How can pipeline task reuse and modularization improve efficiency?

1. What strategies reduce pipeline execution costs in Azure DevOps and GitHub Actions?
- Use self-hosted agents for long-running jobs,
- choose lower-cost hosted agents,
- minimize unnecessary runs,
- and clean up unused artifacts and resources after builds.

2. How do you minimize pipeline execution time?
- Enable parallel jobs,
- break pipelines into smaller stages,
- cache dependencies, run only required tests,
- and avoid redundant steps.
- Trigger pipelines only on relevant changes.

3. What approaches improve pipeline reliability?
- Implement retries for transient failures,
- use stable base images and tools,
- version dependencies,
- and separate critical and optional tasks.
- Regularly review and update pipeline tasks.

4. How can parallelism and agent management be used to optimize performance?
- Increase parallel jobs,
- use job dependencies to optimize order,
- distribute workloads across multiple agents,
- and use agent pools to avoid bottlenecks.

5. What best practices reduce resource consumption in pipelines?
- Remove unused tasks, artifacts, and images.
- Use selective checkout,
- optimize scripts,
- and limit scope of triggered builds and tests to affected components.

6. How do you cache dependencies to accelerate builds?
Implement build and task caching (e.g., npm, pip, Maven cache tasks in Azure Pipelines or actions/cache in GitHub Actions) to avoid downloading/building unchanged dependencies.

---

7. How do you monitor and control agent utilization?

- Track agent pool usage via *Azure DevOps* Analytics or *GitHub Actions* Insights.
- Scale agents up or down as needed, and prioritize high-value jobs.

---

8. What methods reduce wait times for pipeline approvals and manual interventions?

- Automate approvals where possible,
- use automatic triggers,
- set up environment checks,
- and streamline required manual steps to minimize human delays.

---

9. How do you ensure pipelines are robust against transient infrastructure failures?

- Configure retry logic on failed steps,
- use resilient cloud infrastructure,
- and implement fallbacks or safe defaults for network- or resource-related errors.

---

11. **How can pipeline task reuse and modularization improve efficiency?**

- Use YAML templates, task groups, and reusable workflows to standardize common tasks,
- reduce duplication,
- and simplify maintenance across multiple pipelines.