3. Build and release pipelines

└ **3.4 Deployment strategy and execution**

└ **3.4.2 Ordered dependency deployments**

---

1. What is an ordered dependency deployment?
2. Why are ordered deployments important in multi-tier applications?
3. How can you enforce deployment order in Azure DevOps pipelines?
4. How do pipeline stages and dependencies work in YAML pipelines?
5. What are artifact dependencies and how do they affect deployment order?
6. How can environment approvals enforce ordered deployment?
7. What's the risk of ignoring dependency order in deployments?
8. How can you handle rollback if a downstream dependency fails?
9. How do you automate database and application deployment order?
10. What tools or tasks in Azure Pipelines help manage ordered deployments?

---

1. What is an ordered dependency deployment?

It is a deployment sequence that ensures components are deployed in a specific order based on their dependencies, such as deploying a database before the application that uses it.

---

2. Why are ordered deployments important in multi-tier applications?

They prevent runtime errors by ensuring lower-level components (e.g., databases, APIs) are ready before dependent components (e.g., web apps) are deployed.

---

3. How can you enforce deployment order in Azure DevOps pipelines?

Define multiple stages in YAML, using the `dependsOn` keyword to control execution order. Each stage deploys one component; later stages run only after dependencies succeed.

---

4. How do pipeline stages and dependencies work in YAML pipelines?

Stages can be explicitly ordered using `dependsOn`. A stage will only start after its dependencies have completed successfully, ensuring proper deployment sequence.

---

5. What are artifact dependencies and how do they affect deployment order?

Artifact dependencies link pipeline outputs (e.g., builds) to deployments. A deployment waits for required artifacts to be published by earlier stages, enforcing correct order.

---

6. How can environment approvals enforce ordered deployment?

You can require manual or automated approvals on environments. A later environment or stage won't proceed until prior deployments are approved, adding an extra control layer for order.

---

7. What's the risk of ignoring dependency order in deployments?

If components are deployed out of order, dependencies may be missing or incompatible, causing application errors, failed deployments, or downtime.

---

8. How can you handle rollback if a downstream dependency fails?

Automate rollback by defining pipeline tasks that revert all affected components to their previous stable versions if any dependent deployment fails.

9. How do you automate database and application deployment order?
In your pipeline YAML, create separate stages for the database and application. Set the application stage to `dependsOn` the database stage, ensuring the database is deployed and verified first.

---

11. **What tools or tasks in Azure Pipelines help manage ordered deployments?**
Azure Pipelines provides

- YAML stages,
- `dependsOn`,
- classic Release pipeline environments,
- deployment jobs,
- approvals,
- gates,
- and artifact dependencies

to control and enforce deployment order.