

### 3. Build and release pipelines

#### └ 3.1 Advanced pipeline design

##### └ 3.3.3 Checks and approvals with YAML environments

---

1. What are environment checks in Azure Pipelines YAML?
  2. How do you configure approvals for a deployment environment in YAML?
  3. What types of checks can be enforced in an environment?
  4. How do you reference and use environments in YAML pipelines?
  5. How do manual approvals work in multi-stage pipelines?
  6. How can you restrict deployments to specific environments?
  7. How do you require multiple approvers for an environment?
  8. What are best practices for designing approval workflows in YAML?
  9. What limitations or pitfalls exist with environment checks and approvals?
  10. How do you audit or monitor approval history for environments?
- 

#### 1. What are environment checks in Azure Pipelines YAML?

Environment checks are policies (such as approvals, service health, or business rules) that must pass before a deployment can proceed to a given environment.

---

#### 2. How do you configure approvals for a deployment environment in YAML?

Define the environment in the deployment job and set up approvals in the Azure DevOps portal under *Pipelines > Environments*. Approvals are not specified directly in YAML.

---

#### 3. What types of checks can be enforced in an environment?

Common checks include:

- Manual approvals
  - Azure Function or REST API checks
  - Business rule validations
  - Service health checks
- 

#### 4. How do you reference and use environments in YAML pipelines?

Use the `environment` keyword in a deployment job:

```
environment: 'staging'
```

This triggers checks and approvals configured for that environment.

---

#### 5. How do manual approvals work in multi-stage pipelines?

The pipeline pauses at the environment check until designated users approve. Only after approval does the deployment job execute.

---

#### 6. How can you restrict deployments to specific environments?

Limit access using environment security settings in *Azure DevOps*, specifying who can deploy and approve to each environment.

---

#### 7. How do you require multiple approvers for an environment?

In the *Azure DevOps* portal, add multiple users or groups to the approval list for the environment. You can set whether all or just one must approve.

#### **8. What are best practices for designing approval workflows in YAML?**

- Use environment approvals for production/staging
  - Limit approvers to relevant stakeholders
  - Combine automated and manual checks
  - Regularly review and update approver lists
- 

#### **9. What limitations or pitfalls exist with environment checks and approvals?**

- Approval policies must be managed in the portal, not in YAML
  - Delays if approvers are unavailable
  - Lack of fine-grained conditional logic for approvals
  - Inability to automate some business rules fully
- 

#### **10. How do you audit or monitor approval history for environments?**

View approval history and audit logs in the *Azure DevOps* portal under the environment's "History" tab. This includes who approved, when, and comments if provided.