1. Design and implement processes and communications
└ 1.3 Configure collaboration and communication
└ 1.3.4 Integrate using webhooks, GitHub/Azure Boards, Teams

---

1. What is a webhook and how is it used in DevOps integrations?
2. How do you configure a webhook in GitHub or Azure DevOps?
3. What events can trigger webhooks in GitHub and Azure DevOps?
4. How do you integrate GitHub with Azure Boards?
5. What information can be synchronized between GitHub and Azure Boards?
6. How do you integrate Azure DevOps or GitHub with Microsoft Teams?
7. What are best practices for managing webhook security?
8. How can you filter or restrict which events trigger integrations?
9. How do you monitor or troubleshoot webhook and integration delivery?
10. What are common use cases for combining these integrations in DevOps workflows?

---

1. What is a webhook and how is it used in DevOps integrations?
A webhook is an HTTP callback that notifies external services about specific events (e.g., pushes, PRs). In DevOps, webhooks automate integrations, trigger workflows, and synchronize data across systems.

---

2. How do you configure a webhook in GitHub or Azure DevOps?
In *GitHub*: Go to *repo Settings → Webhooks → Add webhook → Enter payload URL*, events, and secret.
In *Azure DevOps*: *Project Settings → Service Hooks → Create Subscription → Select service and events*.

---

3. What events can trigger webhooks in GitHub and Azure DevOps?
Common triggers: push, pull request, release, issue, build completion, work item updated, code review events.

---

4. How do you integrate GitHub with Azure Boards?
Install *Azure Boards GitHub App* in repo(s), connect organization, link commits/PRs with work items using syntax (e.g., AB#123), and enable automatic updates.

---

5. What information can be synchronized between GitHub and Azure Boards?
PRs, commits, and linked issues can update *Azure Boards* work items (state, comments, links). Work items can reflect code changes.

---

6. How do you integrate Azure DevOps or GitHub with Microsoft Teams?
Use the *Azure DevOps* or *GitHub Teams* app. Configure connectors or subscriptions to post updates on builds, PRs, or work items to Teams channels.

---

7. What are best practices for managing webhook security?
Use HTTPS URLs, validate payload signatures (e.g., X-Hub-Signature), keep secrets confidential, and limit webhook permissions to only required events.

---

8. How can you filter or restrict which events trigger integrations?
Configure only specific event types in webhook or service hook settings. Use filters (e.g., branch, file path) and conditions in automation workflows.
9. How do you monitor or troubleshoot webhook and integration delivery?
Check webhook delivery logs in *GitHub/Azure DevOps*.

Use

- Status codes
- Request/response details
- Set up retry policies for failures

What are common use cases for combining these integrations in DevOps workflows?

1. Automated notifications (*Teams*)
1. Linking code changes to work items (*Azure Boards*)
2. Triggering CI/CD on PR/merge
3. Automating release processes
4. Synchronizing incident response