**Unit Code - AI-PE - AI for Platform Engineers Scope and Sequence**
**AI-PE-01 (a) - AI Tools Overview for Platform Engineers - Async** (60 min, Asynchronous)

| | |
|---|---|
| **Session Objectives** | • Identify common AI tools for DevOps tasks.<br>• Explain how AI reduces manual workload in platform engineering.<br>• Assess risks associated with AI reliance. |
| **Key Points** | • **GitHub Copilot:** An AI-powered code completion tool that assists in writing infrastructure scripts, CI/CD configurations, and automation tasks.<br>• **GitHub Actions:** A CI/CD platform built into GitHub. It lets you automate workflows like testing, building, and deploying your code.<br>• **ChatGPT:** An AI-based assistant that helps debug, optimise, and explain DevOps scripts and configurations.<br>• **CodeGPT:** Visual Studio Code extension that integrates OpenAI's ChatGPT to assist developers with code generation, explanation, refactoring, and debugging. It enables users to interact with AI directly within the VS Code editor, helping to streamline coding tasks and enhance productivity.<br>• **AI for DevOps:** The use of artificial intelligence tools to automate, optimise, and enhance DevOps workflows, including infrastructure provisioning, monitoring, and CI/CD automation.<br>• **AI-Assisted Debugging:** Process of using AI tools such as ChatGPT to analyse and fix script errors, reducing manual troubleshooting.<br>• **Automation in DevOps:** Using AI and scripting to streamline repetitive tasks such as deployment, monitoring, and infrastructure management.<br>• **Security Risks in AI-Generated Code:** Potential vulnerabilities or misconfigurations introduced by AI-generated scripts that require human review for compliance and safety.<br>**Potential Risks and Mitigations:**<br>• **Hardcoded secrets**<br>→ Risk: API keys directly in code<br>→ Mitigation: Use environment variables or secret managers<br>• **Overly permissive permissions**<br>→ Risk: Broad access in workflows<br>→ Mitigation: Follow the least privilege principle |

| | |
|---|---|
| | • **Missing input validation**<br>→ Risk: Unvalidated external inputs<br>→ Mitigation: Add input checks and sanitise inputs<br>• **Insecure defaults**<br>→ Risk: Using 'latest' tags, open ports, etc.<br>→ Mitigation: Use secure configurations intentionally<br>• **Lack of logging/auditing**<br>→ Risk: No traceability for actions<br>→ Mitigation: Add logging and monitoring<br>• **Outdated dependencies**<br>→ Risk: Vulnerabilities in libraries<br>→ Mitigation: Use tools like *npm audit* or Dependabot |
| **Rubrics** | |

**Unit Code - AI-PE - AI for Platform Engineers Scope and Sequence**
**AI-PE-01 (a) - AI Tools Overview for Platform Engineers - Async** (60 min, Asynchronous)

## MODULE OVERVIEW

| | |
|---|---|
| **Module Name** | AI-PE-01 (a) - AI Tools Overview for Platform Engineers (a) (60 min) |
| **Module Settings** | Lock until: indicate date: _____<br>Prerequisites (other modules): indicate module(s): _____<br><br>Requirements:<br>**Students must complete all of these requirements**<br>Students must move through requirements in sequential order<br>Students must complete one of these requirements |

| Item Title (include text headers) | Item Type (include text headers) | Module Requirements | Indentation (0/ 1/ 2/ 3) |
|---|---|---|---|
| AI-PE-01 (a) - AI Tools Overview for Platform Engineers (a) (60 min) | Text Header | NA | 0 |
| AI-PE-01 (a) - AI-Powered Automation (12 min) | Page | **view the item** | 1 |
| AI-PE-01 (a) - AI Reflection (3 min) | Quiz | **submit the assignment** | 1 |
| AI-PE-01 (a) - Objectives (1 min) | Page | **view the item** | 1 |
| AI-PE-01 (a) - Overview of AI Tools (6 min) | Page | **view the item** | 1 |
| AI-PE-01 (a) - Streamlining Manual Tasks in DevOps with AI (6 min) | Page | **view the item** | 1 |

| | | | |
|---|---|---|---|
| AI-PE-FM-TUC-GQ1/AI-PE-01 (a): AI Tools Overview for Platform Engineers - Check for Understanding (5 min) | Quiz | **submit the assignment** | 1 |
| AI-PE-01 (a) - Using AI to Build and Improve a CI/CD Pipeline for an Apache Web Server (20 min) | Page | **view the item** | 1 |
| AI-PE-FM-TUC-GQ2/AI-PE-01 (a) - AI Tools Overview for Platform Engineers Independent Practice | Quiz | **submit the assignment** | 1 |
| AI-PE-01 (a) - Final Reflection (6 min) | Discussion forum | **contribute to the page** | 1 |
| AI-PE-01 (a) - Closing (1 min) | Page | **view the item** | 1 |

| **TEXT HEADER** | |
|---|---|
| **Item Title** | AI-PE-01 (a) - AI Tools Overview for Platform Engineers (a) (60 min) |
| **Indent Level** | 0 |

## ITEMS DETAILS

| 1. Hook - Page | |
|---|---|
| **Item Title** | AI-PE-01 (a) - AI-Powered Automation (10 min) |
| **Content (format as seen)** | <Insert DP Banner - General><br><br>**Try It Yourself: Predict What AI Will Code**<br>This short, hands-on activity helps you explore how AI tools can support DevOps tasks you've already worked with, like Kubernetes, Linux, and WordPress. You'll try **GitHub Copilot** for the first time, make a simple prediction, and reflect on the results.<br><br>**What is GitHub Copilot?**<br>**GitHub Copilot** is an AI-powered coding assistant integrated into Visual Studio Code (VS Code). As you type, it suggests lines or blocks of code based on the context of your work, acting like a helpful AI pair programmer.<br><br>It can help when:<br>● You want to get started quickly<br>● You're not sure how to write something from scratch<br>● You want to see how AI might approach a task<br><br>Example (in a YAML file):<br><Insert Canvas Paragraph - Preformatted><br># Generate a Kubernetes deployment for a WordPress server<br><br>Copilot then tries to write the code for you. |

**Optional**: Watch a quick walkthrough to see what GitHub Copilot can do

| **Embedded Video** | |
|---|---|
| **Video URL** | https://www.youtube.com/watch?v=n0NlxUyA7FI |
| **Accessible?** | Yes<br>No |
| **Credit Source**<br>**(in font size 10, black text)** | Credit: GitHub. (2025, March 3). *Getting started with GitHub Copilot | Tutorial* [Video]. YouTube. https://www.youtube.com/watch?v=n0NlxUyA7FI |

**Set Up GitHub Copilot (First-Time Use)**

Follow these steps to install and activate GitHub Copilot for the first time:

1. Open **VS Code**.
2. Go to **View > Extensions** or press **Ctrl + Shift + X**.
3. Search for **GitHub Copilot** and click **Install**.
    - This will install two extensions: **GitHub Copilot** (which offers inline code suggestions) and **GitHub Copilot Chat**, a conversational AI assistant that helps with specific coding tasks.
    - GitHub Copilot offers a free tier known as Copilot Free, which provides limited access to its AI-powered coding assistance features. With Copilot Free, you receive up to 2,000 code completions and 50 chat requests per month. This plan is suitable for occasional users and small projects.
4. Once installed, you'll need to sign in with your GitHub account that has access to GitHub Copilot.
    - If you haven't previously authorised VS Code in your GitHub account, you'll be prompted to sign in directly within VS Code.

- Alternatively, click the account icon in the bottom-left corner of VS Code and choose **Sign in with GitHub**.

5. You'll be redirected to your browser. Click **Continue** to grant GitHub the permissions needed for GitHub Copilot.
6. Return to VS Code. You should now see:
   - A **Copilot icon at the top** next to the command centre
   - A **Copilot icon at the bottom** of the editor
7. Click the bottom icon. If prompted, choose your GitHub account. When set up correctly, the status should say **"Ready"**.

✅ *If you see the Copilot icons in your editor, you've successfully installed GitHub Copilot.*

💡 *You don't need to use a special command — Copilot starts working as soon as you type a comment or a line of code.*

---

**What You'll Do**

| **Accordion** |
|---|
| <Vertical arrangement: To allow learners to open or close dropdown information when they click on a heading<br>To use for step-by-step process, when you have long headers> |
| ACCORDION NAME 1<br>ACCORDION NAME 2<br>ACCORDION NAME 3<br>ACCORDION NAME 4<br>ACCORDION NAME 5 |
| **1. Choose a Familiar Task to Automate**<br>Pick one of the following tasks:<br>● Automating a **Kubernetes deployment** |

- Writing a **Linux shell script**
- Creating a **WordPress deployment script**

**2. Prompt GitHub Copilot or Ask GitHub AI Assistant**
Create a new file based on the task you've chosen (for example, a **.yaml** file for Kubernetes, a **.sh** script for Linux, or a **.conf** file for WordPress).

There are two ways to get started:

- **Option A: Use GitHub Copilot inline**
  Start typing a comment at the top of your file (e.g. **# Create a Kubernetes deployment…**). Copilot will suggest code. Press **Tab** to accept the suggestion.
- **Option B: Use GitHub Copilot Chat (AI Assistant)**
  Press **Ctrl + I** (or click the Copilot icon in the sidebar) to open the chat assistant. Type a request like:
  "**Generate a shell script that creates a new user and starts Apache**"

Then, write a prompt (e.g. a comment or a message to the AI assistant) describing what you want Copilot to generate. Use the file type that matches your task:

- **Kubernetes** → **.yaml** file
  Comment example:

  <Insert Canvas Paragraph - Preformatted>
  # Create a Kubernetes deployment for a MySQL server running on Apache

- **Linux** → **.sh** or **.txt** file
  Comment example:

  <Insert Canvas Paragraph - Preformatted>
  # Shell script to create a new user and start Apache

- **WordPress** → **.sh**, **.yaml**, or **.conf** file

Comment example:

<Insert Canvas Paragraph - Preformatted>
# Script to deploy WordPress on CentOS with a database

You'll see a grey suggestion appear. Accept it by pressing **Tab**.

**Note**

You don't need to begin your prompt with #, but it's helpful — especially in shell scripts or YAML — because it lets Copilot know you're describing a task in plain language. In other cases, Copilot can respond to partial code or natural language too.

**Alternative using GitHub AI Assistant:** You can also try opening the GitHub Copilot Chat panel and typing your request instead of using a comment. For example:

"**Generate a shell script that adds a new user and starts Apache**"

Compare which version is easier to understand or more complete.

**3. Pause and Predict (Just a Little!)**
Before you accept the full suggestion, ask yourself:

- What might be included?
- Are there one or two parts I expect to see?

**Use this quick prediction template:**
**Task chosen: [e.g. Kubernetes deployment]. My prediction: I think Copilot will include...[Write 1–2 sentences about what you expect – e.g. an image name, port, or command.]**

Example:

**Task chosen: WordPress deployment. My prediction: I think Copilot will include the WordPress image and expose port 80. Maybe it will use volumes, too.**

*You don't need to be exact — it's just to help you think ahead.*

**4. Reveal the Output**
Let Copilot generate the full script or configuration. Read it through and see what stands out.

**5. Reflect (Gently)**
Use this short template to help you reflect:
**What I expected: [Write what you predicted]**
**What Copilot gave me: [Mention any match or difference]**
**One thing that surprised me was: [Optional – something unexpected or interesting]**

This is not about right or wrong — just noticing how AI can support your thinking.

| **Options** | Users allowed to edit this page: Only teachers<br>Add to student to-do |
|---|---|

| **2. Activate Prior Knowledge - New Quizzes (NON-Graded Survey)** | |
|---|---|
| **Assignment Name** | AI-PE-01 (a) - AI Reflection (3 min) |

| | | |
|---|---|---|
| **Settings options for Intro Quiz Page** | **Points** | **1 pt** |
| | **Assignment Group** | Active Participation<br>If select "Create Group", indicate group and bold: _____ |
| | **Display Grade As** | Complete or Incomplete<br>**Do not count this assignment toward the final grade**<br>Do not display in gradebook or the student's grade page |

| | |
|---|---|
| **Title in "Build" page** | AI-PE-01 (a) - AI Reflection (4 min) |
| **Instructions in "Build" page** | Think back to a manual DevOps task you've worked on — for example:<br>● Deploying and configuring WordPress<br>● Deploying WordPress to a Kubernetes cluster<br>● Writing a Linux bash script for tasks like adding a user, starting a service, checking disk usage, backing up a directory, or updating system packages |

| | |
|---|---|
| | Now, imagine using AI to support that task.<br><br>_____<br><br>To start the activity:<br>● If using a Web browser, click "Begin"<br>● If using the Canvas Student App, tap “Launch External Tool” and then tap “Begin”<br><br>After you have completed the activity,<br>● Click "Submit" to submit your responses.<br>● Then, click "Return" on the top right corner to come back to this page<br>● Lastly, click "Next" on the bottom of this page, in order to proceed to the next activity. |

| | |
|---|---|
| **Settings options for the Setting page** | Shuffle questions |
| | Shuffle answers |
| | One question at a time<br>Allow backtracking |
| | Require a student access code |
| | Time limit<br>____Hours ___ Minutes |

| | |
|---|---|
| | Filter IP addresses |
| | Allow Calculator |
| | Allow clearing selection (Multiple Choice) |
| | Show custom feedback with results |
| | Allow multiple attempts |
| | **Hide results from students**<br>**Show questions**<br>**Show student responses**<br>**For all attempts**<br>Only once after each attempt<br>Only after their 1st attempt<br>Only once after their last attempt<br>Indicate response as correct or incorrect<br>Show correct answer<br>Only after their last attempt<br>**Show feedback**<br>Show points possible (Display both points overall and per question)<br>Show points awarded (Display both points overall and per question) |

<table>
<tr><th colspan="2">Multiple Answers</th></tr>
<tr><td>Question Title</td><td>AI-PE-01 (a) - Survey 1 - Q1</td></tr>
<tr><td>Points</td><td>N/A</td></tr>
<tr><td>Add Question Stem</td><td>Which of these tasks felt most challenging when you completed them manually? Select all that apply</td></tr>
<tr><td>Answer Options</td><td>
<table>
<tr><th>Tick the Correct Answers</th><th>Answer Options</th></tr>
<tr><td></td><td>Writing a Kubernetes deployment YAML file</td></tr>
<tr><td></td><td>Configuring WordPress by editing its configuration files</td></tr>
<tr><td></td><td>Writing or modifying a Linux bash script</td></tr>
<tr><td></td><td>None of these felt particularly challenging for me</td></tr>
</table>
</td></tr>
<tr><td rowspan="3">Feedback</td><td>For a correct answer:</td></tr>
<tr><td>For an incorrect answer:</td></tr>
<tr><td>Provide general feedback (regardless of answer):<br>Thanks for your response! This question helps you reflect on which manual DevOps tasks felt most demanding and identify where AI tools might be most useful.</td></tr>
</table>

| | |
|---|---|
| **Options** | Show on-screen calculator<br>Shuffle Choices<br><br>Grading:<br>Partial credit with Penalty (*Students are awarded points for every correct answer selected and deducted points for every incorrect answer selected)*<br>**Exact match (*Learners are awarded full credit if all correct answers are selected and no incorrect answers are selected.)*** |
| **Align to Outcomes** | |
| **Item Banking** | Add to Bank<br>Existing item bank<br>Select the destination bank:<br>New item bank<br>New Bank Title: |

| **Multiple Answers** | |
|---|---|
| **Question Title** | AI-PE-01 (a) - Survey 1 - Q2 |
| **Points** | N/A |
| **Add Question Stem** | What was the most challenging part of that manual task? |
| **Answer Options** | |

<table>
<tr><td></td><td>
<table>
<tr><th>Tick the Correct Answers</th><th>Answer Options</th></tr>
<tr><td></td><td>Remembering the right syntax or structure</td></tr>
<tr><td></td><td>Troubleshooting errors when it didn’t work</td></tr>
<tr><td></td><td>Making it work in your specific environment</td></tr>
<tr><td></td><td>It wasn’t difficult — I felt confident doing it</td></tr>
</table>
</td></tr>
<tr><td rowspan="3">Feedback</td><td>For a correct answer:</td></tr>
<tr><td>For an incorrect answer:</td></tr>
<tr><td>Provide general feedback (regardless of answer):<br>Thank you for sharing! Reflecting on the toughest parts helps you see where AI can support tasks like setup, troubleshooting, or script customisation.</td></tr>
<tr><td>Options</td><td>Show on-screen calculator<br>Shuffle Choices<br><br>Grading:<br>Partial credit with Penalty (Students are awarded points for every correct answer selected and deducted points for every incorrect answer selected)<br>Exact credit (Learners are awarded full credit if all correct answers are selected and no incorrect answers are selected.)</td></tr>
<tr><td>Align to Outcomes</td><td></td></tr>
</table>

| | |
|---|---|
| **Item Banking** | Add to Bank<br>Existing item bank<br>Select the destination bank:<br>New item bank<br>New Bank Title: |

| **Multiple Answers** | |
|---|---|
| **Question Title** | AI-PE-01 (a) - Survey 1 - Q3 |
| **Points** | N/A |
| **Add Question Stem** | Based on your most recent DevOps experience, which type of AI support would have been most helpful to you? *(Select the one that best matches your needs.)* |
| **Answer Options** | |

| Tick the Correct Answers | Answer Options |
|---|---|
| | Getting started can take time — I'd use AI to give me a working example |
| | Fixing errors can be frustrating — AI could help me troubleshoot |
| | Adjusting scripts to my environment can be tricky — AI could help with customisation |
| | Some parts of the scripts aren't always clear — I'd use AI to explain what each part does |

<table>
<tr><td></td><td><table><tr><td></td><td>I feel comfortable doing these tasks manually — I might not need AI support right now</td></tr></table></td></tr>
<tr><td rowspan="3">Feedback</td><td>For a correct answer:</td></tr>
<tr><td>For an incorrect answer:</td></tr>
<tr><td>Provide general feedback (regardless of answer):<br>Your answer shows how AI can support different learning stages—whether you're building confidence, solving problems, or validating your approach.</td></tr>
<tr><td>Options</td><td>Show on-screen calculator<br>Shuffle Choices<br><br>Grading:<br>Partial credit with Penalty (Students are awarded points for every correct answer selected and deducted points for every incorrect answer selected)<br>Exact credit (Learners are awarded full credit if all correct answers are selected and no incorrect answers are selected.)</td></tr>
<tr><td>Align to Outcomes</td><td></td></tr>
<tr><td>Item Banking</td><td>Add to Bank<br>Existing item bank<br>Select the destination bank:<br>New item bank<br>New Bank Title:</td></tr>
</table>

<table>
<tr><th colspan="2">Multiple Choice</th></tr>
<tr><td>Question Title</td><td>AI-PE-01 (a) - Survey 1 - Q4</td></tr>
<tr><td>Points</td><td>1</td></tr>
<tr><td>Add Question Stem</td><td>Are you ready to start?</td></tr>
<tr><td>Answer Options</td><td>
<table>
<tr><th>Tick the Correct Answers</th><th>Answer Options</th></tr>
<tr><td></td><td>Yes! Let’s begin!</td></tr>
</table>
</td></tr>
<tr><td rowspan="3">Feedback</td><td>For a correct answer:</td></tr>
<tr><td>For an incorrect answer:</td></tr>
<tr><td>Provide general feedback (regardless of answer):<br>Let’s start!</td></tr>
<tr><td>Options</td><td>Show on-screen calculator<br>Shuffle Choices</td></tr>
<tr><td>Align to Outcomes</td><td></td></tr>
<tr><td>Item Banking</td><td>Add to Bank</td></tr>
</table>

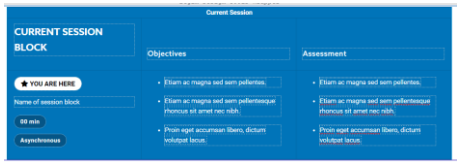
| | |
|---|---|
| | Existing item bank<br>Select the destination bank:<br>New item bank<br>New Bank Title: |

| **3. Objectives - Page** | |
|---|---|
| **Item Title** | AI-PE-01 (a) - Objectives (1 min) |
| **Content (format as seen)** | <Insert DP Banner - Objectives & Agenda><br><Insert DP Element - Checklist><br><br>Welcome to the first session of the **AI-PE-01: AI Tools Overview for Platform Engineers** module!<br>By the end of this session, you will be able to:<br>● Identify common AI tools for DevOps tasks.<br>● Explain how AI reduces manual workload in platform engineering.<br>● Assess risks associated with AI reliance.<br><br>The module consists of the following sessions. Please see the current session's objectives in dark blue, marked with a star. |

**Current Session**

| **SESSION 1** | **Objectives** | **Assessment** |
|---|---|---|
| AI-PE-01 (a): AI Tools Overview for Platform Engineers | By the end of this session, you will be able to: | A reflection on how AI supported your DevOps workflow, including one |

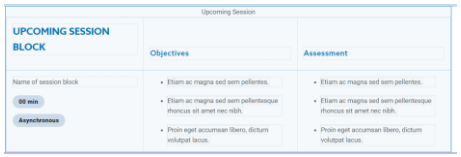
| | | |
|---|---|---|
| 60 minutes<br>Asynchronous | • Identify common AI tools for DevOps tasks.<br>• Explain how AI reduces manual workload in platform engineering.<br>• Assess risks associated with AI reliance. | improvement made with GitHub Copilot or Copilot Chat and how it helped streamline or enhance your CI/CD pipeline. |




**Upcoming Session**

| SESSION 2 | Objectives | Assessment |
|---|---|---|
| AI - PE - 01 (b): AI Tools Overview for Platform Engineers (b)<br>60 minutes<br>Synchronous | By the end of this session, you will be able to:<br>• Discuss the practical benefits and limitations experienced when using various AI tools.<br>• Compare experiences across different AI tools, identifying best-fit scenarios.<br>• Reflect on effective prompting | A reflection that includes specific insights or takeaways from the examples explored, highlighting how each AI tool supported your DevOps workflow. |

| | |
|---|---|
| | strategies learned. | |




| | |
|---|---|
| **Options** | Users allowed to edit this page: Only teachers<br>Add to student to-do |

| **4. Concept Introduction - Page** | |
|---|---|
| **Item Title** | AI-PE-01 (a) - Overview of AI Tools (4 min) |
| **Content (format as seen)** | <Insert DP Banner - Generic><br><br>As a platform engineer, you manage a wide range of responsibilities — from writing code to monitoring systems and responding to incidents. What if a few AI tools could help ease some of that workload?<br><br>**Get to Know Your AI Tools** |

Here is a brief overview of four AI tools commonly used in platform engineering, along with their primary functions:

| **Table** | | | | |
|---|---|---|---|---|
| **Build Out the Table in the Content block** | **Tool** | **What It Does** | **Best Use Case** | **Think of It As...** |
| | **GitHub Copilot** | Autocompletes code and suggests full functions. | Writing or debugging code in IDEs | Your AI pair programmer. |
| | **Aider** | Natural language code edits via terminal. | CLI refactoring, debugging. | Git-integrated ChatGPT in CLI. |
| | **Cline** | Pull requests from diffs, commit messages, and code aid. | PR workflow, code documentation. | IDE AI coding partner. |
| | **New Relic AI** | Detects anomalies and automates alerts. | Monitoring, incident response. | Your AI-powered system watchdog. |
| **Table Caption** | **AI Tools: Purpose and Use Cases** | | | |

| **Hyperlink** | |
|---|---|
| **Hyperlink text** | *GitHub Copilot* |

| | |
|---|---|
| **URL** | https://github.com/features/copilot |
| **External or Course link?** | External link<br>Course link |

| **Hyperlink** | |
|---|---|
| **Hyperlink text** | *Aider* |
| **URL** | https://aider.chat/ |
| **External or Course link?** | External link<br>Course link |

| **Hyperlink** | |
|---|---|
| **Hyperlink text** | *Cline* |
| **URL** | https://cline.bot/ |
| **External or Course link?** | External link<br>Course link |

| **Hyperlink** | |
|---|---|
| **Hyperlink text** | *New Relic AI* |
| **URL** | https://newrelic.com/ |
| **External or Course link?** | External link<br>Course link |

The following brief descriptions outline what each tool does and how it enhances your workflow:

- "Hi, I'm **Copilot**! I love turning your TODO comments into working code."
- "I'm **Aider**, your terminal buddy. Just tell me what you want to fix—I'll handle the edits."
- "**Cline** here—I turn your git diffs into polished pull requests."
- "I'm **New Relic AI**, watching over your systems so you can sleep."

---

**Optional: Watch the Tools in Action**

To get a clearer idea of how each tool functions, you can watch the short video overviews below. These are optional but provide helpful insights into the tools' features and how they support common development tasks.

| **Embedded Video** | |
|---|---|
| **Video URL** | https://www.youtube.com/watch?v=IqXNhakuwVc |

| | |
|---|---|
| **Accessible?** | Yes<br>No |
| **Credit Source**<br>**(in font size 10, black text)** | Credit: GitHub. (2022, November 15). *What is GitHub Copilot?* [Video]. YouTube. https://www.youtube.com/watch?v=IqXNhakuwVc |

| **Embedded Video** | |
|---|---|
| **Video URL** | https://www.youtube.com/watch?v=ooEQm0dyCjU |
| **Accessible?** | Yes<br>No |
| **Credit Source**<br>**(in font size 10, black text)** | Credit: Praison, M. (2025, January 14). *Is Aider AI REALLY the Future of Programming? (Ollama, Groq)* [Video]. YouTube. https://www.youtube.com/watch?v=ooEQm0dyCjU |

| **Embedded Video** | |
|---|---|
| **Video URL** | https://www.youtube.com/watch?v=KjqQC4AnJ1I |

| | |
|---|---|
| **Accessible?** | Yes<br>No |
| **Credit Source**<br>**(in font size 10, black text)** | Credit: Praison, M. (2025, January 13). *Cline + VS Code Changed How I Code Forever* [Video]. YouTube. https://www.youtube.com/watch?v=KjqQC4AnJ1I |

| **Embedded Video** | |
|---|---|
| **Video URL** | https://www.youtube.com/watch?v=3YjpWM0JWMo |
| **Accessible?** | Yes<br>No |
| **Credit Source**<br>**(in font size 10, black text)** | Credit: New Relic. (2024, February 28). *Introducing New Relic AI Monitoring*. YouTube. https://www.youtube.com/watch?v=3YjpWM0JWMo |

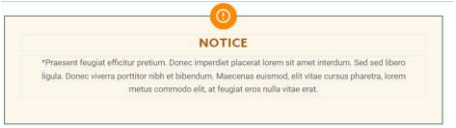
These AI tools aren't just features — they're here to support the way you work. As you learn about them, consider how they could help you work smarter, respond faster, and focus on what matters most.

| | |
|---|---|
| **Options** | Users allowed to edit this page: Only teachers<br>Add to student to-do |

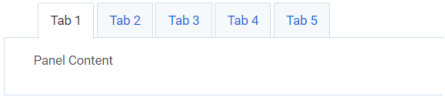
| **5. Modelling - Page** | |
|---|---|
| **Item Title** | AI-PE-01 (a) - Streamlining Manual Tasks in DevOps with AI (6 min) |
| **Content (format as seen)** | <Insert DP Banner - Generic><br><Insert DP Element - Plain TextBox><br><br>Even experienced DevOps professionals know that manual work can be tedious, repetitive, and error-prone. Whether writing complex deployment files or responding to a flood of alerts in production, these tasks can consume time and attention that could be better spent on high-value work — like optimising infrastructure, improving reliability, or enabling faster delivery for development teams.<br><br>In this section, you'll explore two practical ways AI can reduce the manual workload in everyday DevOps tasks:<br><br>1. **Aider** helps reduce the friction of directly working with scripts and configuration files in your terminal. It understands the context of your code, applies changes, and explains them — all within your CLI workflow. |

2. **New Relic AI** transforms incident triage by automatically correlating alerts, filtering out noise, and identifying likely root causes — cutting resolution time from minutes to seconds.

These tools don't replace your expertise — they amplify it, reducing manual effort so you can focus on what matters most.

| **Notice** |
| --- |
| To alert learners to pay attention to certain important content.<br>Icon can be removed to just have the call-out box to contain session key points or key takeaways in closing. |
|  |
| Before using any AI-powered tool shown in this module, make sure to check your organisation's policies and confirm with your supervisor that the tool is approved for use in your environment. |

| **Tabs** |
| --- |
| Horizontal arrangement: To present content separated by multiple headings<br>Need to bold the tabs<br>To use when you have short headers and relatively even amount of content for each tab<br>Note: Tab 1 is always opened for view |




**Tab 1: Improving a Bash Script with Aider**

**Watch the tutorial**
This short video walks you through the activity step by step, showing how Aider helps improve a Bash script directly from your terminal using natural language prompts.

| **Embedded Video** | |
|---|---|
| **Video URL** | https://youtu.be/zboEchOfdss |
| **Accessible?** | Yes<br>No |
| **Credit Source**<br>**(in font size 10, black text)** | Credit: Pacheco, D. (2025, April 7). *AI PE 01 (a): Improving a Bash Script with Aider* [Video]. YouTube. https://youtu.be/zboEchOfdss |

Then, review the step-by-step instructions below to see how it works in more detail.

---

🔷 **Step 1: Install Aider**
Install Aider by following the official installation guide

| Hyperlink | |
|---|---|
| **Hyperlink text** | official installation guide |
| **URL** | https://aider.chat/docs/install.html |
| **External or Course link?** | External link<br>Course link |

◆ **Step 2: Create a Simple Bash Script**

You'll start with a basic script that attempts to start the Apache service.

In your terminal or preferred editor, create a new file called **start-apache.sh** with the following content:

<Insert Canvas Paragraph- Preformatted>

```
#!/bin/bash
systemctl start apache2
```

At this point, the script does not check whether Apache is already running — that's what you'll improve with Aider.

◆ **Step 3: Launch Aider and Update the Script**

Now, let's improve the script using Aider.
Open your terminal and navigate to the folder where start-apache.sh is saved.
Launch Aider with the script by running:

<Insert Canvas Paragraph- Preformatted>

```
aider --model <model> --api-key <provider>=<key> start-apache.sh
```

Where:

- **--model:** AI model to use
- **--api-key:** Your key for that model provider
- **start-apache.sh:** File to edit with AI help

When prompted, type the following instruction:
"*Update this script to check if Apache is already running before starting it, and print a message accordingly.*"

This tells Aider to load the file, understand its context, and help you modify it
Aider will read your script, apply the update, and explain the changes it made.

✅ **Result: A More Robust Script**
Here's the updated version Aider generates:

<Insert Canvas Paragraph- Preformatted>

```
#!/bin/bash

# Check if Apache is already running
if systemctl is-active --quiet apache2; then
    echo "Apache is already running."
else
    systemctl start apache2
    echo "Apache started."
fi
```

With Aider, you can edit and improve scripts using AI — all from the terminal.
No need to jump between your editor, browser, and terminal.
Just type what you want to change, review the suggestion, and apply it — all in one place.

This keeps you focused and makes learning more efficient.

**Tab 2: Incident Triage with New Relic AI**

**Start by Watching This Video**

To get a first look at how New Relic uses AI to support incident detection and response, watch the video Detecting Anomalies Automatically with New Relic.

This video helps you understand the challenges it's designed to address.

| **Embedded Video** | |
|---|---|
| **Video URL** | https://www.youtube.com/watch?v=ImmpojfvKoc |
| **Accessible?** | Yes<br>No |
| **Credit Source**<br>**(in font size 10, black text)** | Credit: New Relic. (2021, March 17). *Detecting Anomalies Automatically with New Relic* [Video]. YouTube. https://www.youtube.com/watch?v=ImmpojfvKoc |

**Now, imagine this in your environment**

Your team manages a production e-commerce application. One evening, the system experiences a sudden spike in response times. Multiple alerts are triggered across services: database latency, CPU usage, failed transactions, and frontend timeouts:

- Jumping between dashboards (New Relic, AWS CloudWatch, logs)
- Manually filtering out irrelevant alerts
- Paging the whole team to investigate different parts of the stack
- Spending 30+ minutes just identifying the source of the problem

DevOps teams can spend **20 to 40 minutes just figuring out what's wrong** before they can even start fixing the issue — sifting through alerts, checking logs, and ruling out false positives. That's valuable time lost, especially during critical incidents where every minute matters.

**With New Relic AI: The Automated Way**

1. It automatically correlates all these alerts into one single incident.
2. It detects that the spike started with the **payment service API** timing out.
3. This is flagged as the probable root cause based on past incidents and system context.
4. It suppresses unrelated alerts and routes the incident directly to the payments team — with a clear summary of what happened and why.

✅ **Result**:

- The right engineer is notified
- Only one actionable alert is raised
- **Root cause suggested** — no need to dig manually

| | |
|---|---|
| | With New Relic AI, triage time is reduced from around **30 minutes to under 5** — by automatically linking related alerts, identifying the payment service API as the likely root cause, and routing the issue straight to the right team. Engineers can skip the detective work and focus immediately on resolving the actual problem. |

| | |
|---|---|
| **Options** | Users allowed to edit this page: Only teachers<br>Add to student to-do |

| **6. Check For Understanding - New Quizzes (Graded Quiz)** | |
|---|---|
| **Assignment Name** | AI-PE-FM-TUC-GQ1/AI-PE-01 (a): AI Tools Overview for Platform Engineers - Check for Understanding (5 min) |

| | | |
|---|---|---|
| **Settings options for Intro Quiz Page** | **Points** | **3 pt** |
| | **Assignment Group** | Formative Technical |
| | **Display Grade As** | Points |

| | | |
|---|---|---|
| | | Do not count this assignment toward the final grade<br>Do not display in gradebook or the student's grade page |

| | |
|---|---|
| **Title in "Build" page** | AI-PE-FM-TUC-GQ1/AI-PE-01 (a): AI Tools Overview for Platform Engineers - Check for Understanding (5 min) |
| **Instructions in "Build" page** | This quiz checks your understanding of how AI tools can support and challenge DevOps workflows. You'll identify key benefits and consider potential risks.<br><br>---<br><br>To start the activity:<br>● If using a Web browser, click "Begin"<br>● If using the Canvas Student App, tap "Launch External Tool" and then tap "Begin"<br><br>After you have completed the activity,<br>● Click "Submit" to submit your responses.<br>● Then, click "Return" on the top right corner to come back to this page<br>● Lastly, click "Next" on the bottom of this page, in order to proceed to the next activity. |

| | |
|---|---|
| **Settings options for the Setting page** | Shuffle Questions |
| | Shuffle Answers |

| | |
|---|---|
| *For Assignment Groups: Diagnostic, Formative, or Summative* | **One question at a time**<br>**Allow backtracking** |
| | Require a student access code |
| | **Time limit**<br>**____Hours _5_ Minutes** |
| | Filter IP addresses |
| | Allow Calculator |
| | Allow clearing selection (Multiple Choice) |
| | Show custom feedback with results |
| | **Allow multiple attempts**<br>Score to Keep: Latest<br>Allowed attempts: Limited<br>**Attempts: 2**<br>Waiting period: Require time between attempts<br>Build on last attempt: Enable build on last attempt |
| | **Hide results from students**<br>**Show questions**<br>**Show student responses**<br>**For all attempts**<br>Only once after each attempt<br>Only after their 1st attempt |

| | |
|---|---|
| | Only once after their last attempt<br>**Indicate response as correct or incorrect**<br>**Show correct answer**<br>**Only after their last attempt**<br>**Show feedback**<br>**Show points possible (Display both points overall and per question)**<br>**Show points awarded (Display both points overall and per question)** |

| **Multiple Choice** | |
|---|---|
| **Question Title** | AI-PE-FM-TUC-GQ1/AI-PE-01 (a): AI Tools Overview for Platform Engineers - Check for Understanding - Q1 |
| **Points** | 1 |
| **Add Question Stem** | Which of the following is a benefit of using AI tools like GitHub Copilot in DevOps workflows? |
| **Answer Options** | |

| **Tick the Correct Answers** | **Answer Options** |
|---|---|
| | Automates repetitive scripting tasks |
| | Suggests code snippets but may need human review |
| | Eliminates the need for version control |

<table>
<tr><td></td><td><table><tr><td></td><td>Writes perfect code without errors</td></tr></table></td></tr>
<tr><td rowspan="3">Feedback</td><td>For a correct answer:</td></tr>
<tr><td>For an incorrect answer:</td></tr>
<tr><td>Provide general feedback (regardless of answer):<br>"Automates repetitive scripting tasks" is correct because it captures one of the main strengths of AI coding tools like GitHub Copilot. These tools are designed to save developers time and effort by generating code for routine, repetitive tasks — for example, writing boilerplate code, setting up common functions, or handling standard input/output operations.<br><br>"Suggests code snippets but may need human review" is close, but it doesn't quite capture the main advantage of AI tools. While it's true that AI can suggest code snippets and that these suggestions usually require human oversight to ensure they’re correct, secure, and appropriate for the specific context, this statement misses the key point.<br>The primary benefit of AI tools lies in automation — they can speed up repetitive or routine coding tasks, generate boilerplate code, assist with documentation, and help identify potential issues. The automation aspect significantly improves productivity and efficiency for developers<br><br>"Eliminates the need for version control" is incorrect because AI tools cannot replace version control systems such as Git. Version control is essential for keeping track of changes, collaborating with others, and managing different versions of code. Even if an AI assists in writing or improving code, developers still require version control to organise and manage their work over time.<br><br>"Writes perfect code without errors" is also incorrect because AI tools are not infallible. They can generate code quickly, but that code may still contain bugs, security vulnerabilities, or may not function as intended. Developers must still review, test, and sometimes correct the code produced by AI.</td></tr>
<tr><td>Options</td><td>Show on-screen calculator</td></tr>
</table>

| | |
|---|---|
| | Shuffle Choices<br><br>Grading:<br>Partial credit with Penalty (*Students are awarded points for every correct answer selected and deducted points for every incorrect answer selected)*<br>**Exact credit (*Learners are awarded full credit if all correct answers are selected and no incorrect answers are selected.)*** |
| **Align to Outcomes** | TUC-TIFC-209.1 |
| **Item Banking** | Add to Bank<br>Existing item bank<br>Select the destination bank:<br>New item bank<br>New Bank Title: |

| **Multiple Choice** | |
|---|---|
| **Question Title** | AI-PE-FM-TUC-GQ1/AI-PE-01 (a): AI Tools Overview for Platform Engineers - Check for Understanding - Q2 |
| **Points** | 1 |
| **Add Question Stem** | How can an AI-based tool assist during a CI/CD pipeline configuration task? |
| **Answer Options** | |

<table>
<tr><td></td><td>
<table>
<tr><th>Tick the Correct Answers</th><th>Answer Options</th></tr>
<tr><td></td><td>By executing deployments directly</td></tr>
<tr><td></td><td>By writing front-end interface elements</td></tr>
<tr><td></td><td>By reviewing and suggesting improvements in your configuration scripts</td></tr>
<tr><td></td><td>By automatically publishing your pipeline to a remote repository</td></tr>
</table>
</td></tr>
<tr><td rowspan="3">Feedback</td><td>For a correct answer:</td></tr>
<tr><td>For an incorrect answer:</td></tr>
<tr><td>Provide general feedback (regardless of answer):<br>“By reviewing and suggesting improvements in your configuration scripts” is the best answer because AI tools are effective at reviewing and refining configuration scripts such as YAML. They provide suggestions to improve clarity, structure, and consistency.<br><br>“By automatically publishing your pipeline to a remote repository” is close but misleading—AI tools can't perform direct actions like publishing to repositories. AI tools don’t typically perform actions like pushing commits or publishing code to remote repositories directly. Instead, what they do is generate or suggest the necessary changes (like YAML pipeline configurations or CI/CD steps), which you can then review and manually commit and push yourself.<br><br>“By executing deployments directly” and “By writing front-end interface elements” don’t accurately represent the role of AI in infrastructure configuration, as AI tools are typically used to assist with generating or suggesting configuration code—not to perform live deployments or build user interfaces..</td></tr>
</table>

| | |
|---|---|
| **Options** | Show on-screen calculator<br>Shuffle Choices<br><br>Grading:<br>Partial credit with Penalty (*Students are awarded points for every correct answer selected and deducted points for every incorrect answer selected)*<br>**Exact credit (*Learners are awarded full credit if all correct answers are selected and no incorrect answers are selected.)*** |
| **Align to Outcomes** | TUC-TIFC-209.1 |
| **Item Banking** | Add to Bank<br>Existing item bank<br>Select the destination bank:<br>New item bank<br>New Bank Title: |

| **Multiple Choice** | |
|---|---|
| **Question Title** | AI-PE-FM-TUC-GQ1/AI-PE-01 (a): AI Tools Overview for Platform Engineers - Check for Understanding - Q3 |
| **Points** | 1 |
| **Add Question Stem** | Which of the following is a potential limitation of relying on AI-generated code? |
| **Answer Options** | |

<table>
<tr><td></td><td>
<table>
<tr><th>Tick the Correct Answers</th><th>Answer Options</th></tr>
<tr><td></td><td>May improve some test automation steps but misses human judgment</td></tr>
<tr><td></td><td>Lack of understanding of context and potential security risks</td></tr>
<tr><td></td><td>Risk of over-relying on AI-generated suggestions without understanding the code</td></tr>
<tr><td></td><td>Possibility of missing edge-case scenarios in deployment configurations</td></tr>
</table>
</td></tr>
<tr><td rowspan="3">Feedback</td><td>For a correct answer:</td></tr>
<tr><td>For an incorrect answer:</td></tr>
<tr><td>Provide general feedback (regardless of answer):<br>"Lack of understanding of context and potential security risks" is correct because AI doesn’t fully grasp your project’s purpose or structure. This can lead to code that doesn’t fit properly or overlooks key details.<br><br>It may also introduce security issues, like unsafe defaults or poor handling of user input, as it can’t reliably judge what’s secure or appropriate in every situation.<br><br>“May improve some test automation steps but misses human judgment” sounds reasonable but does not directly highlight a risk — which is that over-relying on AI-generated tests could lead to gaps in coverage, missed edge cases, or false confidence in test quality without proper human oversight.<br><br>“Risk of over-relying on AI-generated suggestions without understanding the code” is close but less precise, as the issue is not just over-reliance but misunderstanding what AI generates — the core issue isn’t just dependence on AI, but the risk that developers may misunderstand or misinterpret what the AI generates, leading to incorrect implementations or unnoticed errors in critical systems.</td></tr>
</table>

| | |
|---|---|
| | "**Possibility of missing edge-case scenarios in deployment configurations**" is incorrect because while missing edge-case scenarios can be a limitation, it's not as central or frequent as the broader risk of missing context, such as team-specific conventions, environment-specific dependencies, or organisational policies, which are critical for correct and safe deployment configurations. |
| **Options** | Show on-screen calculator<br>Shuffle Choices<br><br>Grading:<br>Partial credit with Penalty (*Students are awarded points for every correct answer selected and deducted points for every incorrect answer selected)*<br>**Exact credit (*Learners are awarded full credit if all correct answers are selected and no incorrect answers are selected.)*** |
| **Align to Outcomes** | TUC-TIFC-209.3<br>TUC-TIFC-209.5<br>BSM-OD-01 |
| **Item Banking** | Add to Bank<br>Existing item bank<br>Select the destination bank:<br>New item bank<br>New Bank Title: |

| **7. Guided Practice - Page** | |
|---|---|
| **Item Title** | AI-PE-01 (a) - Using AI to Build and Improve a CI/CD Pipeline for an Apache Web Server (20 min) |

**Content (format as seen)**

<Insert DP Banner - Generic>
<Insert DP Element - Plain Text Box>

Modern DevOps rely on automation to streamline deployment, testing, and configuration. In this short activity, you'll use **GitHub Copilot and Copilot Chat** to:

- Build a basic CI/CD pipeline to deploy an Apache-based website using **GitHub Actions**
- Review and improve the pipeline using **AI-powered suggestions**
- Reflect on how AI contributes to real-world DevOps practices

**Watch this short tutorial video**

To get an overview of the activity, watch the video below. It walks you through the process step by step and demonstrates how GitHub Copilot and Copilot Chat can support each stage of the pipeline.

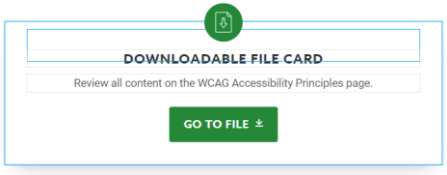
| **Embedded Video** | |
|---|---|
| **Video URL** | https://www.youtube.com/watch?v=ueFPWqRBGlE |
| **Accessible?** | Yes<br>No |
| **Credit Source**<br>**(in font size 10, black text)** | Credit: Pacheco, D. (2025, April 6). *AI PE 01 (a): Using AI to Build and Improve a CI CD Pipeline for an Apache Web Server* [Video]. YouTube. https://www.youtube.com/watch?v=ueFPWqRBGlE |

Then continue with the detailed step-by-step instructions below to try it yourself.

<Insert DP Element - Plain Text Box>

**Before You Begin**
**Using the Provided ZIP Folder**

You will begin by downloading a **.zip** file that contains a sample Apache project. This lets you get started quickly by automatically setting up the folder structure and content—no manual setup is required.

| **Card - Downloadable File** |
| --- |
| To provide a link for learners to download files |
|  |
| *Simple Apache Site*<br><br>*Click to download AI-PE-01 (a): AI Tools Overview for Platform Engineers-simple-apache-site.zip.*<br><br>*GO TO FILE* ***(hyperlink file)*** |

**Instructions to Set Up the Project in GitHub**

Set up your project repository on GitHub and upload the required files before using GitHub Actions and AI tools in this lesson. The steps below will guide you through the process:

- Creating a new GitHub repository

- Uploading the provided project files
- Preparing the repository so you can begin building and improving your CI/CD pipeline

| **Accordion** |
|---|
| Vertical arrangement: To allow learners to open or close dropdown information when they click on a heading<br>To use for step-by-step process, when you have long headers |
|  |
| Accordion 1: Step 1: Create a GitHub Repository<br><br>1. Go to GitHub and log in.<br>2. Click the ➕ icon in the top right → **New repository**<br>3. Name it: simple-apache-site (or another name)<br>4. ✅ Tick **Add a README file**<br>5. ❌ Do **not** add a .gitignore or license<br>6. Click **Create repository** |

| **Hyperlink** | |
|---|---|
| **Hyperlink text** | GitHub |

| | |
|---|---|
| **URL** | https://github.com |
| **External or Course link?** | External link<br>Course link |

🔐 **Use SSH for Secure and Seamless GitHub Authentication**

If you regularly use GitHub from the command line, consider setting up SSH authentication to streamline access.. It's a secure, password-free way to push and pull code once set up — ideal for long-term use.
🔎 **GitHub no longer allows password authentication for Git operations**

To learn more, consult GitHub's official guide:

Connecting to GitHub with SSH – GitHub Docs

| **Hyperlink** | |
|---|---|
| **Hyperlink text** | *Connecting to GitHub with SSH – GitHub Docs* |
| **URL** | https://docs.github.com/en/authentication/connecting-to-github-with-ssh |
| **External or Course link?** | External link<br>Course link |

Accordion 2: Step 2: Upload the Project Files to GitHub

1. Extract the **AI-PE-01 (a): AI Tools Overview for Platform Engineers-simple-apache-site.zip** file.
2. Open a terminal in the unzipped folder.
3. Push the files to your new repo using the following commands:

<Insert Canvas Paragraph- Preformatted>

```
git init
git remote add origin https://github.com/YOUR-USERNAME/simple-apache-site.git
git add .
git commit -m "Initial commit"
git branch -M main
git push -u origin main
```

Replace YOUR-USERNAME with your GitHub username.

Accordion 3: Step 3: Continue the Lesson

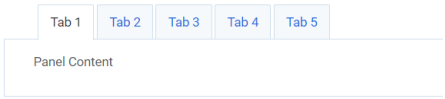
Now, your repo is ready! You can:

- Add the GitHub Actions workflow
- Use GitHub Copilot and Copilot Chat to scaffold and improve it

**Quick Reminder:**

Before proceeding with the activity, review the key concepts behind **GitHub Actions** and **workflows**. These tools form the foundation of the automation you'll be implementing.

| **Hyperlink** | |
|---|---|
| **Hyperlink text** | *GitHub Actions and workflows* |
| **URL** | https://docs.github.com/en/actions/about-github-actions/understanding-github-actions |
| **External or Course link?** | External link<br>Course link |

| **Tabs** |
|---|
| Horizontal arrangement: To present content separated by multiple headings<br>Need to bold the tabs<br>To use when you have short headers and a relatively even amount of content for each tab<br>Note: Tab 1 is always opened for view |
|  |
| **Tab 1: What Is GitHub Actions?**<br><br>**GitHub Actions** is a tool built into GitHub that automates tasks like building, testing, or deploying code — a key part of CI/CD (Continuous Integration / Continuous Deployment). Think of it as setting up little robots that automatically handle repetitive DevOps tasks for you.<br><br>These automations are defined using something called a **workflow**. |

**Tab 2: What's a GitHub Workflow?**

A **GitHub workflow** is a file written in **YAML format** (usually saved in **.github/workflows/**), which tells GitHub **when** to run specific steps and **what** to do.

Each workflow contains:

- **Triggers** – when the workflow should run (e.g., on every push to main)
- **Jobs** – tasks that GitHub will run (e.g., install Apache, deploy files)
- **Steps** – the specific commands to execute in each job

**Example**:
When you push to the main branch, your workflow can:

- Install Apache on a virtual machine
- Copy your website files into **/var/www/html/**
- Restart Apache
  ...all automatically.

In this activity, you'll write a workflow that installs Apache and deploys your website automatically whenever you push changes to your GitHub repository.

**Build & Improve a CI/CD Pipeline with AI**

Now that you've reviewed the key concepts and set up your GitHub repository, you can begin the core activity.

At this stage, you'll use **GitHub Copilot** to create a CI/CD workflow for deploying an Apache web server and then apply suggestions from **GitHub Copilot Chat** to improve it.

Follow the steps below to complete the process.

| **Accordion** |
| --- |
| Vertical arrangement: To allow learners to open or close dropdown information when they click on a heading<br>To use for step-by-step process, when you have long headers |
|  |
| **Accordion 1: Step 1 – Generate the Initial Workflow**<br><br>**Create the workflow file:**<br>Since you've already pushed the ZIP contents to GitHub, you can continue working directly in the local project folder. There's no need to clone the repo again unless you're switching to a new device or starting fresh.<br><br>1. Open your project folder in VS Code:<br><Insert Canvas Paragraph - Preformatted><br>code .<br><br>2. In the **Explorer** view in VS Code (from the root of your project folder), create the following folder structure if it doesn't exist: **.github/workflows/**<br><br>3. **In the workflows folder,** click **Add File > Create New File** and name it:<br>apache-deploy.yaml |

4. At the top of the file, **Press Control + I** and type this comment:

   <Insert Canvas Paragraph - Preformatted>
   # Create GitHub Actions workflow to deploy Apache web server

5. Let **GitHub Copilot** suggest a script—press tab to accept the suggestion.
   If needed, begin with:

   <Insert Canvas Paragraph - Preformatted>

```
name: Deploy Apache Site

on:
  push:
    branches: [ main ]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Install Apache
        run: |
          sudo apt update
          sudo apt install apache2 -y
      - name: Copy site files
        run: |
          sudo cp -r ./public-html/* /var/www/html/
```

   ✅ Save the file.

**Accordion 2: Step 2 – Request Feedback from Copilot Chat**

1. Highlight the YAML script or place your cursor inside the file.
2. **Open Chat view:** In the Visual Studio menu bar, click on the GitHub Icon, then click Open Chat.
3. **Ask**: "Review the following GitHub Actions workflow and suggest improvements and best practices. Provide explanations and examples where appropriate."
    a. Note the AI's suggestions (e.g., using caching, improving file copying logic, hardening for security).
    b. You can tailor your review based on a specific focus. For example: "Review this GitHub Actions workflow and suggest improvements related to performance, security, and maintainability. For each suggestion, explain your reasoning and include examples where helpful."

**Accordion 3: Step 3 – Implement Improvements**

Choose **at least two** suggestions and apply them to your workflow.

Examples might include:

- Adding caching to speed up **apt** installations.
- Securing the use of GitHub Secrets for configuration.
- Adding post-deployment checks.

**Accordion 4: Step 4 – Final Review**

**Ask** Copilot Chat:

> "Does this workflow follow best practices for CI/CD performance and security?"

Make any further refinements based on its response.

**Accordion 5: Step 5 – Push Your Workflow File to GitHub**

After applying improvements to your GitHub Actions workflow file, you need to commit and push it to GitHub so the workflow can run.

Before committing, make sure to **save your apache-deploy.yaml** file in VS Code.

Then, open your terminal in the project folder and run the following commands to commit and push your changes:

Open your terminal in the project folder and run the following commands:

<Insert Canvas Paragraph - Preformatted>

```
git add .github/workflows/apache-deploy.yaml

git commit -m "Add and improve Apache deployment workflow"

git push
```

✅ This step ensures your saved and improved workflow is pushed to GitHub and triggers the GitHub Actions pipeline.

---

**Check That Your Workflow Ran Successfully**

Once you've pushed your updated workflow file to GitHub, GitHub Actions will automatically run it—if it's configured correctly.

**Why It Runs Automatically**

In your **apache-deploy.yaml** file, you likely included this trigger:

<Insert Canvas Paragraph - Preformatted>

```
on:
  push:
    branches: [ main ]
```

This means GitHub will run the workflow **any time a change is pushed to the *main* branch**—including the creation or update of the workflow file itself.

When you ran:

<Insert Canvas Paragraph - Preformatted>

```
git push
```

GitHub automatically detected the change and launched the workflow defined in **.github/workflows/**.

### ◈ A. Go to the Actions Tab

1. In your GitHub repository, click the **Actions** tab (next to Code, Issues, and Pull Requests).
2. You should see your workflow listed under **"All workflows"**.

### ◈ B. Check the Status Icon

Each workflow run will have an icon next to it:

- ✅ **Green check** means the workflow ran successfully.

- ❌ **Red X** means the workflow ran but encountered an error.

If you see a red ❌, don't worry—it just means something in the workflow needs adjusting.

**Accordion 6: Step 6 – Reflect**

In a sentence or two, write your response to:

"One improvement AI helped me make was…"

This promotes reflection and critical thinking about the AI's contribution to your DevOps work.

| **Tip** |
| --- |
| To highlight select information for learners to pay attention to<br>Icon can be removed to just have the call-out box to contain quotes or short key content to highlight or emphasize, etc. |
|  |
| **There Are Other Ways to Ask Copilot Chat** |

You've used one example prompt in this activity, but there are many other ways to interact with GitHub Copilot Chat depending on what you need — from explaining code to debugging errors or generating test cases.

👉 Explore more examples and guidance in the official documentation:
How to Use GitHub Copilot Chat – Visual Studio Code Docs

| **Hyperlink** | |
|---|---|
| **Hyperlink text** | *How to Use GitHub Copilot Chat – Visual Studio Code Docs* |
| **URL** | https://code.visualstudio.com/docs/copilot/getting-started-chat |
| **External or Course link?** | External link<br>Course link |

Feel free to experiment with different questions as you work — Copilot Chat is designed to help you learn, improve, and problem-solve as you go.

## Summary

By completing this activity, you have:

- **Built a CI/CD pipeline** to deploy a simple Apache web server using GitHub Actions automatically.

| | |
|---|---|
| | • **Used GitHub Copilot and Copilot Chat** to review and enhance your workflow with AI-driven suggestions.<br>• **Explored how AI tools can support real-world DevOps practices**, helping to automate tasks, improve code quality, and boost efficiency. |

| | |
|---|---|
| **Options** | Users allowed to edit this page: Only teachers<br>Add to student to-do |

| **8. Independent Practice - New Quizzes (Graded Quiz)** | |
|---|---|
| **Assignment Name** | AI-PE-FM-TUC-GQ2/AI-PE-01 (a) - AI Tools Overview for Platform Engineers Independent Practice |

| | | |
|---|---|---|
| **Settings options for Intro Quiz Page** | **Points** | **1 pt** |
| | **Assignment Group** | Formative Technical |
| | **Display Grade As** | Points<br>Do not count this assignment toward the final grade<br>Do not display in gradebook or the student's grade page |

| | |
|---|---|
| **Title in "Build" page** | AI-PE-FM-TUC-GQ2/AI-PE-01 (a) - Independent Practice: Self-Assessment Checklist (1 min) |
| **Instructions in "Build" page** | Use the following checklist to confirm you've completed each part of the "Using AI to Build and Improve a CI/CD Pipeline for an Apache Web Server".<br><br>---<br><br>To start the activity:<br>● If using a Web browser, click "Begin"<br>● If using the Canvas Student App, tap "Launch External Tool" and then tap "Begin" |

| | |
|---|---|
| | After you have completed the activity,<br>● Click "Submit" to submit your responses.<br>● Then, click "Return" on the top right corner to come back to this page<br>● Lastly, click "Next" on the bottom of this page, in order to proceed to the next activity. |

| | |
|---|---|
| **Settings options for the Setting page** | Shuffle Questions |
| | Shuffle Answers |
| | **One question at a time**<br>**Allow backtracking** |
| | Require a student access code |
| | **Time limit**<br>**____Hours _5_ Minutes** |
| | Filter IP addresses |
| | Allow Calculator |
| | Allow clearing selection (Multiple Choice) |
| | Show custom feedback with results |

| | |
|---|---|
| | **Allow multiple attempts**<br>Score to Keep: Latest<br>Allowed attempts: Limited<br>**Attempts: 2**<br>Waiting period: Require time between attempts<br>Build on last attempt: Enable build on last attempt |
| | **Hide results from students**<br>**Show questions**<br>**Show student responses**<br>**For all attempts**<br>Only once after each attempt<br>Only after their 1st attempt<br>Only once after their last attempt<br>**Indicate response as correct or incorrect**<br>**Show correct answer**<br>**Only after their last attempt**<br>**Show feedback**<br>**Show points possible (Display both points overall and per question)**<br>**Show points awarded (Display both points overall and per question)** |

| **Multiple Answer** | |
|---|---|
| **Question Title** | AI-PE-FM-TUC-GQ2/AI-PE-01 (a) - Independent Practice: Self-Assessment Checklist - Q1 |
| **Points** | 1 |

<table>
<tr><td>Add Question Stem</td><td>Have I successfully used AI tools to build and improve my CI/CD pipeline? (Select all that apply.)</td></tr>
<tr><td>Answer Options</td><td>
<table>
<tr><th>Tick the Correct Answers</th><th>Answer Options</th></tr>
<tr><td></td><td>I created or opened a YAML file under .github/workflows/ in my Apache project.</td></tr>
<tr><td></td><td>I used the comment # GitHub Actions workflow to deploy Apache web server to prompt GitHub Copilot</td></tr>
<tr><td></td><td>I allowed Copilot to generate a basic workflow and accepted or modified the suggestion.</td></tr>
<tr><td></td><td>I used GitHub Copilot Chat to review the workflow and asked, “What improvements could be made to this GitHub Actions workflow?”</td></tr>
<tr><td></td><td>I selected and implemented at least two improvements suggested by the AI.</td></tr>
<tr><td></td><td>I asked Copilot Chat whether the final version followed best practices for performance and security.</td></tr>
<tr><td></td><td>I made final adjustments based on the AI’s feedback.</td></tr>
<tr><td></td><td>I wrote a reflection explaining how AI helped improve my workflow.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">Feedback</td><td>For a correct answer:</td></tr>
<tr><td>For an incorrect answer:</td></tr>
</table>

| | |
|---|---|
| | **Provide general feedback (regardless of answer):** If you selected most or all items, you've successfully completed the activity and demonstrated how multiple AI tools can support DevOps automation. If not, revisit any step(s) to improve your workflow and AI-prompting skills. |
| **Options** | Show on-screen calculator<br>Shuffle Choices<br><br>Grading:<br>Partial credit with Penalty (*Students are awarded points for every correct answer selected and deducted points for every incorrect answer selected)*<br>**Exact credit (*Learners are awarded full credit if all correct answers are selected and no incorrect answers are selected.)*** |
| **Align to Outcomes** | TUC-TIFC-209.1<br>BSM-OD-01<br>BSM-PA-01 |
| **Item Banking** | Add to Bank<br>Existing item bank<br>Select the destination bank:<br>New item bank<br>New Bank Title: |

| 9. Feedback and Application - Discussion Forum (NON-graded) | |
|---|---|
| **Item Title** | AI-PE-01 (a) - Final Reflection (5 min) |
| **Content (format as seen)** | <Insert DP Banner - Discussion><br><br>Reflect on how AI tools like GitHub Copilot and ChatGPT can support your own DevOps work moving forward.<br><br>**Steps**<br><br>1. Think about a real or typical DevOps workflow you've worked on (e.g. provisioning infrastructure, managing deployments, or configuring CI/CD).<br>2. Reflect on how AI could assist or improve that workflow.<br>3. Submit a short written response (2–3 sentences) in the forum by answering:<br>*"Where in your DevOps workflow could AI help the most, and how?"*<br><br>Then, read at least one peer's post and leave a brief reply (1–2 sentences). Your comment can:<br><br>● Add an idea or perspective based on your own experience<br>● Ask a thoughtful follow-up question<br>● Share how your workflow might benefit similarly |

| | |
|---|---|
| **Options** | Anonymous Discussion: Off<br>**Users must post before seeing replies**<br>Enable podcast feed<br>Graded<br>**Allow 'liking'**<br>Only graders can like<br>Group discussion<br>Group category: _____<br>Add to student to-do |

| **10. Closing - Page** | |
|---|---|
| **Item Title** | AI-PE-01 (a) - Closing (1 min) |
| **Content (format as seen)** | <Insert DP Banner - Closing><br><Insert DP Element - Checklist><br><br>Thank you for your participation! By now, you should be able to:<br>• Identify common AI tools for DevOps tasks.<br>• Explain how AI reduces manual workload in platform engineering.<br>• Assess risks associated with AI reliance.<br><br>In the next module, you'll reflect on the practical use, limitations, and prompting strategies of different AI tools to identify which work best in specific DevOps scenarios. |

| | |
|---|---|
| **Options** | Users allowed to edit this page: Only teachers<br>Add to student to-do |