

E3 Onboarding

Step 1:

From the document, I learned about the company's workspace management using VCS, specifically GitHub. I became good at basic Git commands and best practices, such as how to name commits, check out branches, and merge using specific protocols. I also finished a course that improved my understanding of how to maintain repositories, including how to commit changes, revert, and reset repositories. I can now effectively compare current and previous versions, check file statuses, create new branches, and see commit logs.

Module assessment

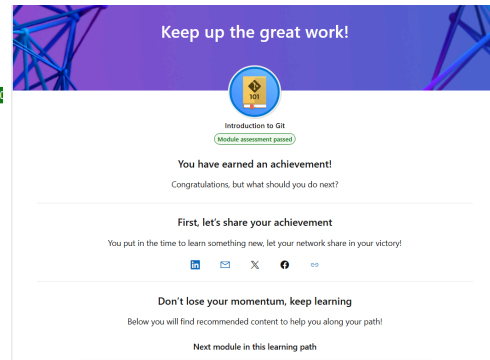
5 minutes



Module assessment passed

Great job! You passed the module assessment.

Score: 100%



Unit 3 of 6

Read through the different options available with Git and note that each command comes with its own help page, for when you start digging deeper. Not all these commands will make sense yet, but some might look familiar if you have experience using a VCS.

In the next lesson, you learn more about the commands you just tried and the basics of Git.

Next unit: Basic Git commands

< Previous Next >

Need help? See our [troubleshooting guide](#) or provide specific feedback by [reporting an issue](#).

Feedback

Was this page helpful? ☐ Yes ☐ No

English (United States) [Your Privacy Choices](#) [Theme](#)

[Previous Versions](#) [Blog](#) [Contribute](#) [Privacy](#) [Terms of Use](#) [Trademarks](#) © Microsoft 2025

```
Azure Cloud Shell

Switch to PowerShell Restart Manage files New session Editor Web preview

Requesting a Cloud Shell. Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

spartacus [ ~ ] $ git --version
git version 2.45.3
spartacus [ ~ ] $ git config --global user.name "spartacus"
spartacus [ ~ ] $ git config --global user.email "spartacus@coreedgesolution.com"
spartacus [ ~ ] $ git config --list
http.sslcapath=/etc/ssl/certs
user.name=spartacus
user.email=spartacus@coreedgesolution.com
spartacus [ ~ ] $ mkdir Cats
spartacus [ ~ ] $ cd Cats
spartacus [ ~/Cats ] $ git init --initial-branch=main
Initialized empty Git repository in /home/spartacus/Cats/.git/
spartacus [ ~/Cats ] $ git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
spartacus [ ~/Cats ] $ ls -a
.
..
.git
spartacus [ ~/Cats ] $ git --help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
[--exec-path=<path>] [--html-path] [--man-path] [--info-path]
[-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
[--config-env=<name>=<envvar>] <command> [<args>]
```

Step 2:

I learned how to handle changes at E3 using AI. I also understood that as a vibe coder AI is our most powerful resource:

- Our Thinking Pattern Should be
 - Define the Problem Through AI Collaboration
 - Plan Through AI-Guided Experimentation
 - Continuous AI-Enhanced Learning
- Learning New Tools using following steps
 - Start with Documentation
 - Build Understanding Through AI Dialogue
 - Refine Your Implementation

- Validate and Expand Your Knowledge
- Learning Technical Concepts by:
 - Gathering Initial Context
 - Building Understanding Systematically
 - Validating and Applying Knowledge
 - Maintaining Knowledge Through Continuous Review

Step 3 :

I read the article and understood the basic use cases of cursor. How it helps and gives better results because of the understanding of the whole code base. How we can use it not just to correct or complete the code but by natural language is very helpful especially in a tense development environment.

Step 4:

- **Post 1:**
<https://x.com/cjzafir/status/1899123302033215543>
 This post tells Cursor technique to remove 80% hallucinations by attaching 'coding documents' and force AI to use it as a knowledge base.
- **Post 2:**
<https://x.com/dshukertjr/status/1903795562602025076>
 We can integrate our superbase database with cursor. It would help me to manage my database with natural language too. While, working on the backend this will surely come in handy.
- **Post 3:**
<https://x.com/pyquantnews/status/1900939905431162976>
 This post explains that to obtain accurate AI responses, instruct the AI to generate multiple solutions telling it to narrow down to 1-2 best solutions and asking it to log it to find the best fit solution.
- **Post 4:**
https://x.com/PrajwalTomar_/status/1899104347532738764
 This post told me how I can improve cursor's code quality by making rules with specific scope rather than using .cursorrules. It mentions important project scope files like frontend.mdc ,etc. This post was very informative especially from a development point of view.
- **Post 5:**
<https://x.com/aaditsh/status/1902019691301986654>
 This post mentions how to create an application using AI. It says to make a new project with /init and all to have documentation. It also highlights professional practices for code optimization and conciseness, specifically recommending `/compact` for larger projects to prevent confusion in Cursor.