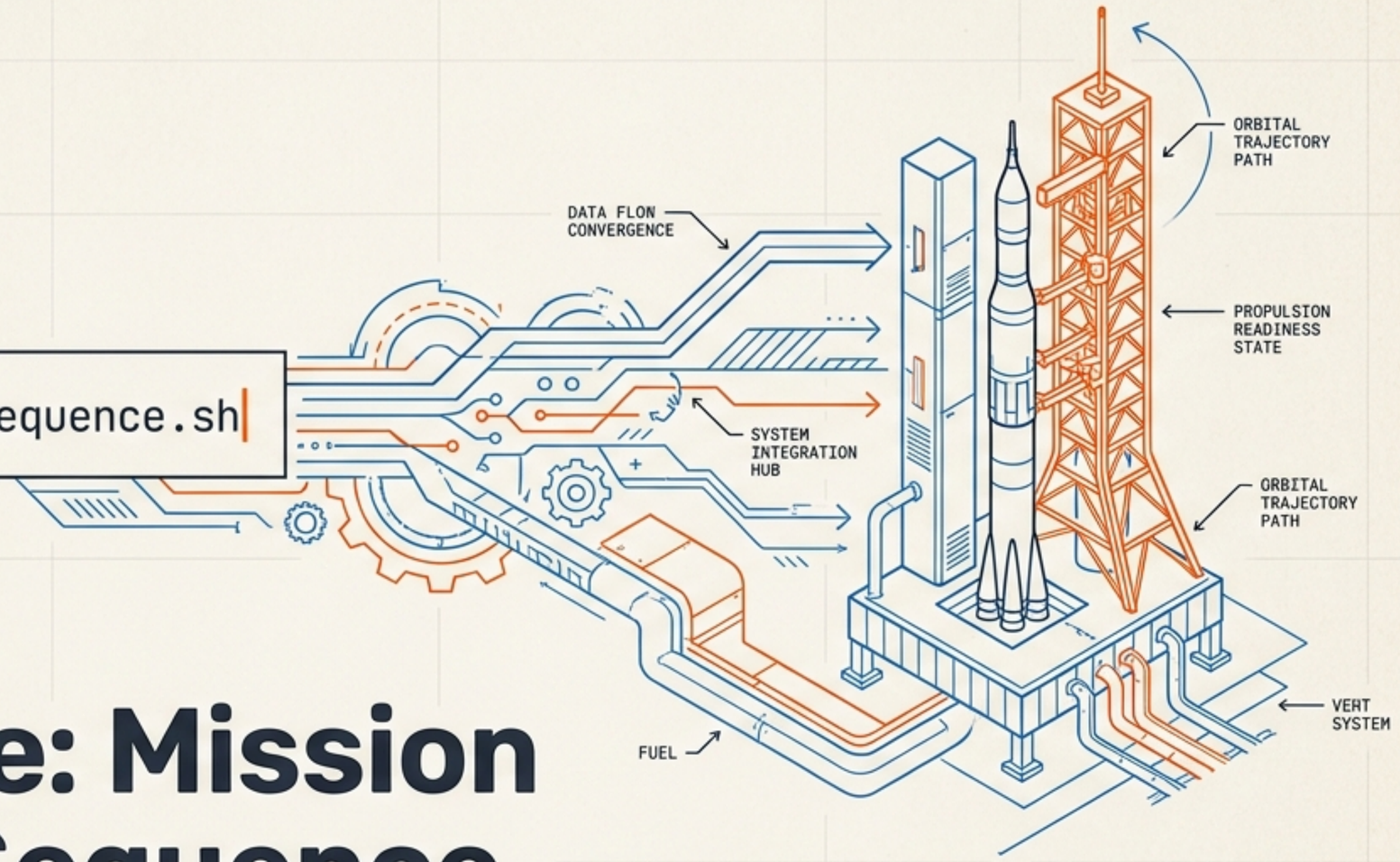
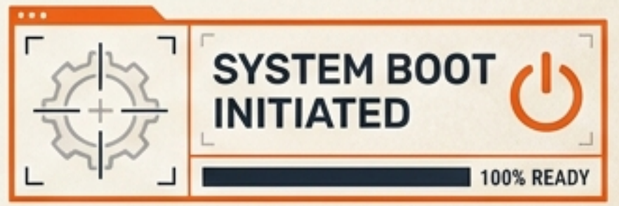
```
> initiate_launch_sequence.sh
```



The Forge: Mission Launch Sequence

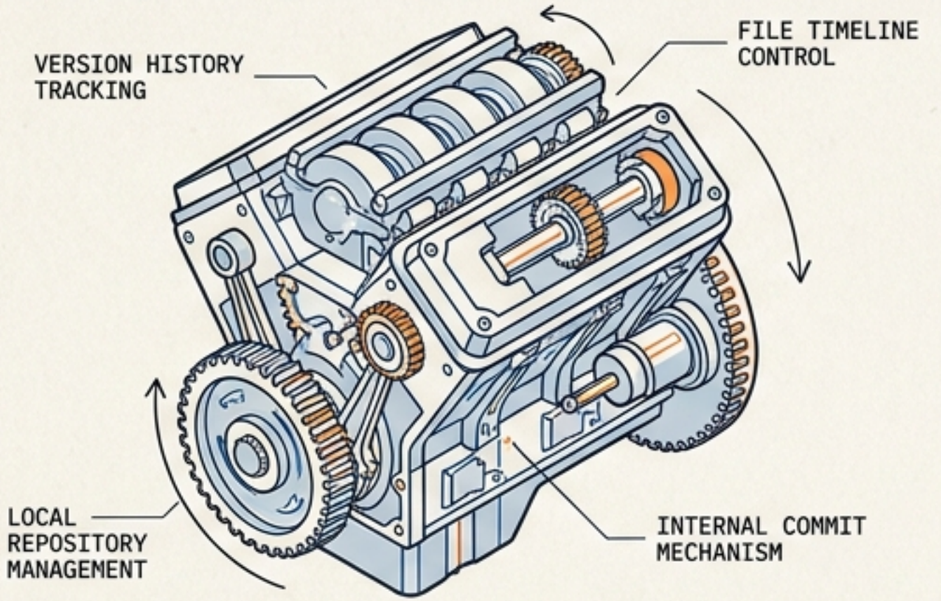
Preparing your local laboratory for professional development.



Understand your toolkit before ignition.

Git

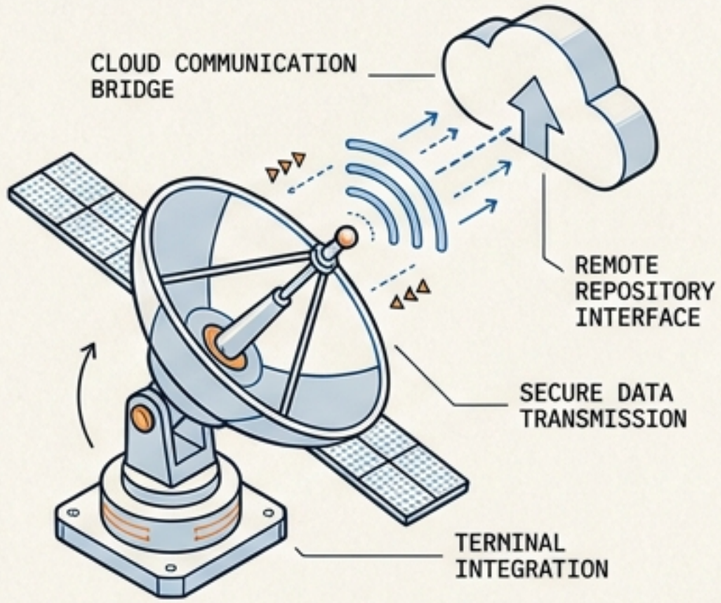
The Core Engine



Runs locally on your machine. The internal engine that tracks the history and timeline of your files.

GitHub CLI (gh)

The Pro Interface



Talks to the cloud. The professional bridge to manage remote repositories without leaving your terminal.

SYSTEM ARCHITECTURE COMPARISON V1.0
SYSTEM ARCHITECTURE COMPARISON V1.0

PHASE 1: INSTALLING THE CORE ENGINE (GIT)

Lock onto your operating system. Ignore the rest.

WINDOWS

Download from git-scm.com & run .exe.

THE NEXT RULE:

Keep clicking **Next** for all default options. They are optimized for beginners.

This installs **Git Bash**, your terminal for all future missions.

macOS

Open Terminal (Cmd + Space, type Terminal).

```
>_
> git --version
```

If not installed, a popup asks for command line developer tools. Click **Install**.

LINUX

Open terminal and run:

```
>_
> sudo apt update
> sudo apt install git-all
>
```

SYSTEM ARCHITECTURE COMPARISON V1.0

SYSTEM ARCHITECTURE COMPARISON V1.0

PHASE 2: INSTALLING THE COMMS (GITHUB CLI)

Deploy the interface to communicate with the cloud.

WINDOWS

Open CMD or PowerShell.

```
>_
> winget install --id GitHub.cli
```

macOS

Open Terminal and use Homebrew.

```
>_
> brew install gh
```

LINUX

Distro-specific.

Follow the official Linux install guide for your specific distribution.

SYSTEM ARCHITECTURE COMPARISON V1.0

SYSTEM ARCHITECTURE COMPARISON V1.0

PHASE 3a: SYSTEM VERIFICATION

Close your current terminal and open a new one (Git Bash for Windows users).

>_

```
> git --version
  git version 2.x.x

> gh --version
  gh version 2.x.x
```

If you see **'command not found'**, your system is misaligned.
Go back to Phase 1.

SYSTEM ARCHITECTURE COMPARISON V1.0
SYSTEM ARCHITECTURE COMPARISON V1.0

PHASE 3b: UPLINK TO THE MOTHERSHIP

```
>_
> gh auth login
```

What account? ➔ GitHub.com

Protocol? ➔ HTTPS

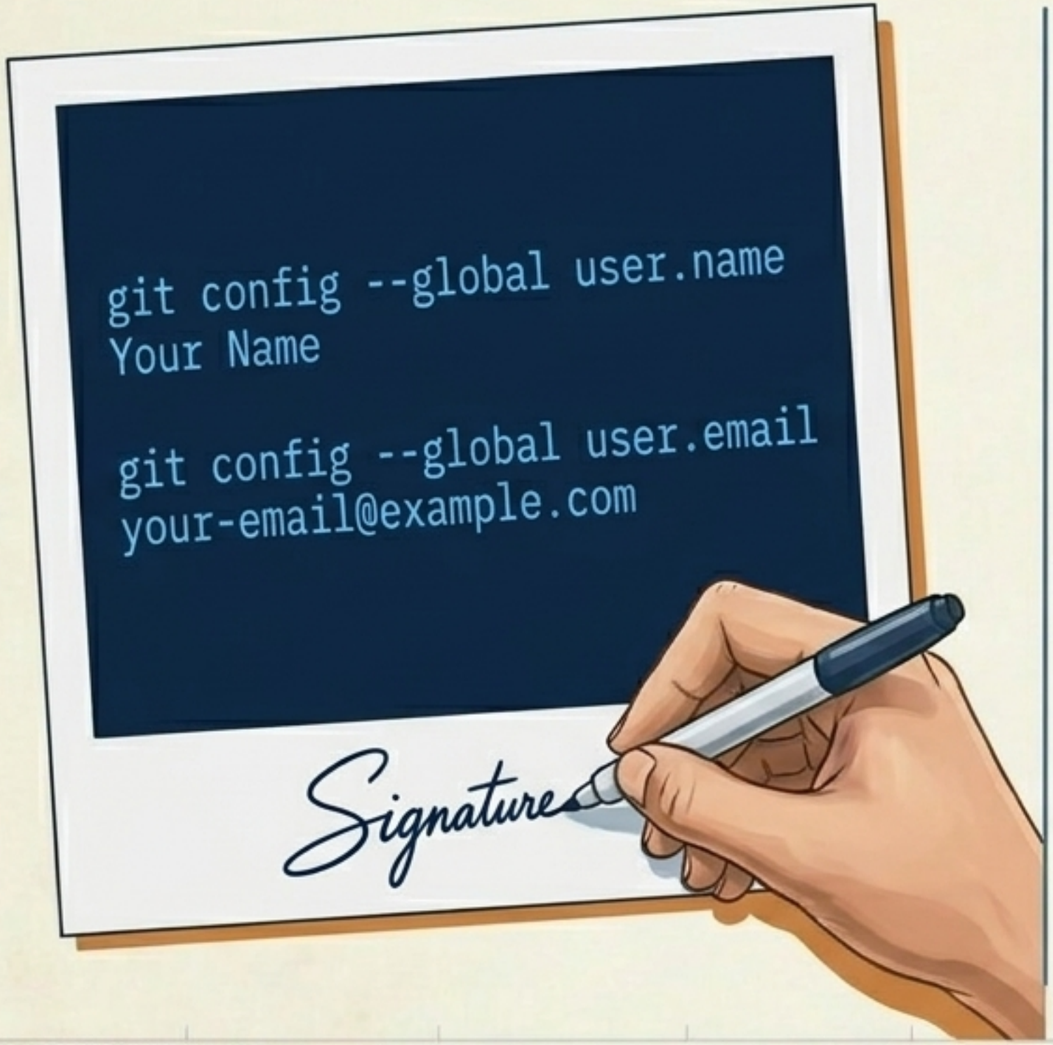
Stick to HTTPS for the easiest setup

Authenticate? ➔ Login with your browser

Click **Authorize** and enter your GitHub password if asked.

Phase 4: Pilot Registration

Tell Git who you are so your snapshots (commits) always have your name on them.




```
>_

> git config --global
 user.name "Your Name"

> git config --global
 user.email "your-email@exa-
 mple.com"
```

The Architecture of a Commit

Your Machine

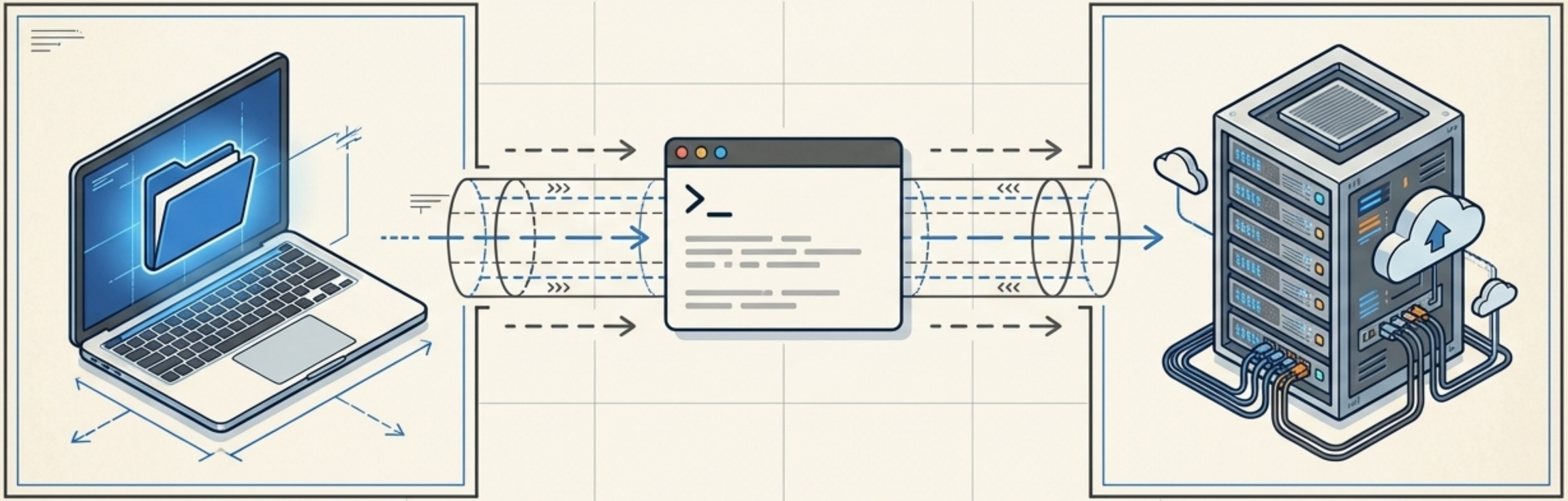
Git operates here, tracking local history and creating the snapshots.

GitHub CLI (gh)

The encrypted bridge handling authentication and commands.

GitHub.com

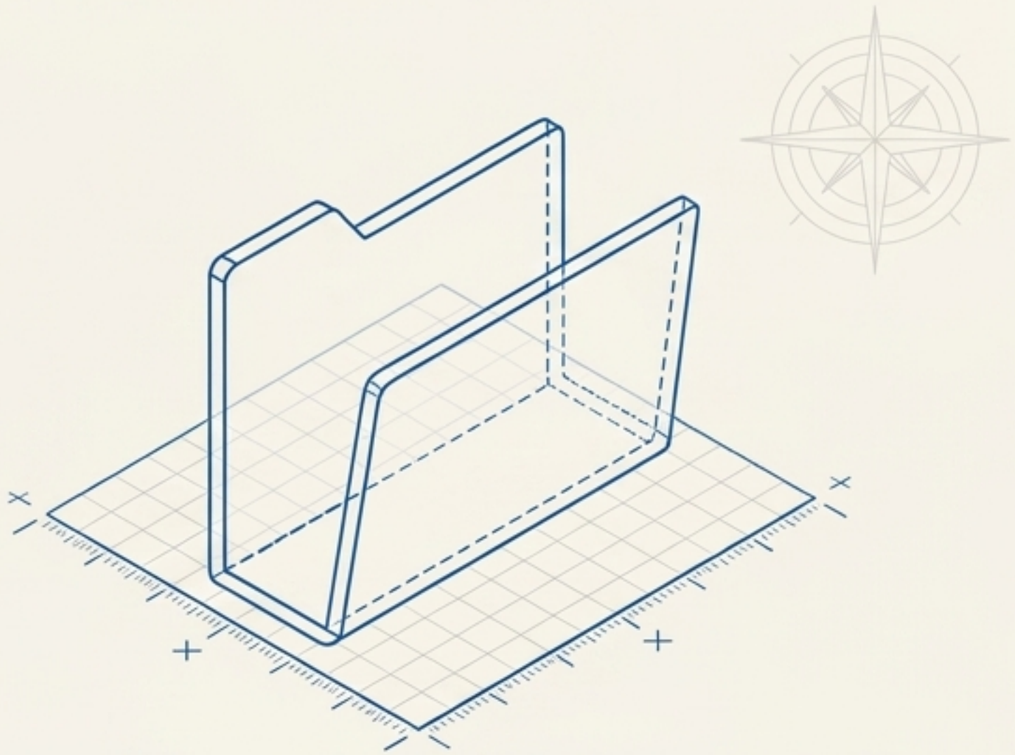
The remote destination storing your parallel universes.



The Test Flight: 1. Building the Hull

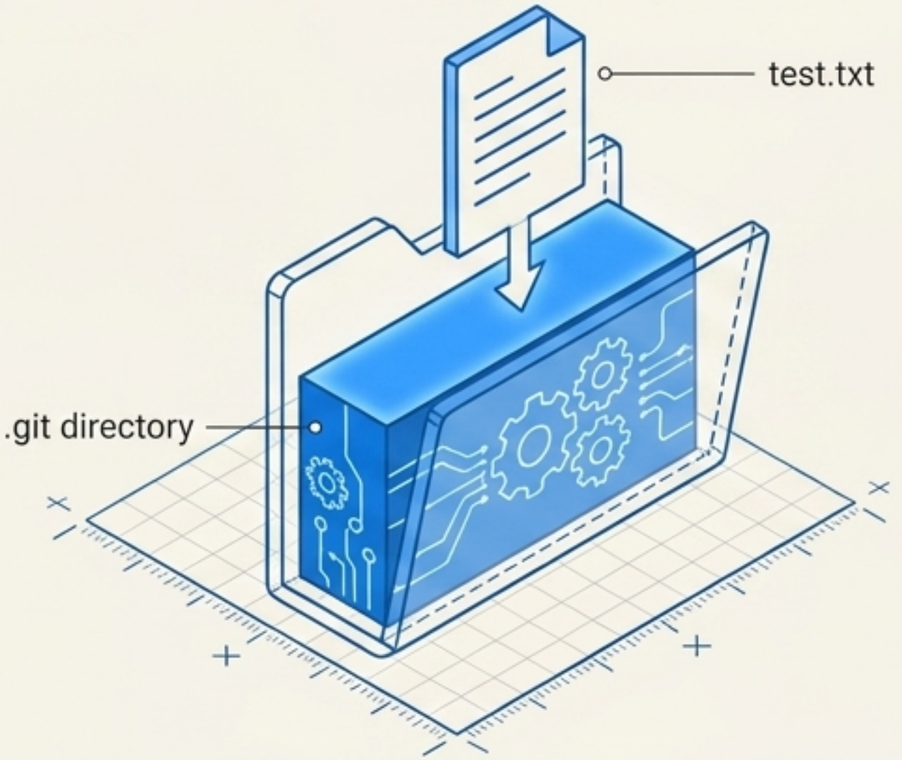
Step A: Create Workspace

```
> mkdir skill-wanderer-test
> cd skill-wanderer-test
```



Step B: Initialize Engine & Add Cargo

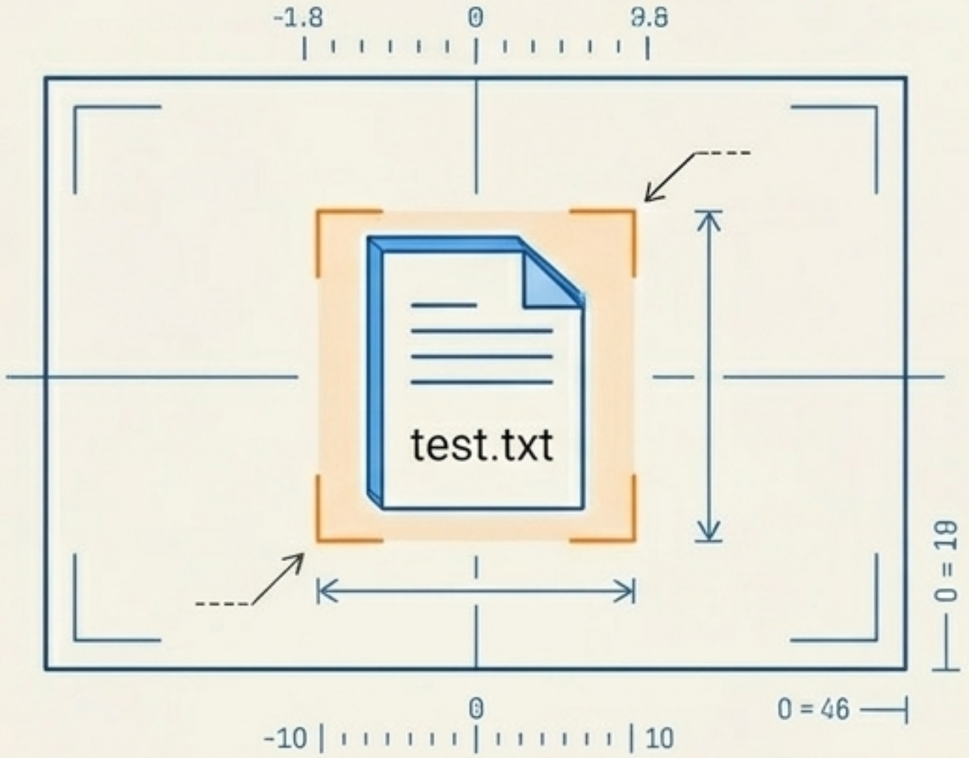
```
> git init
> echo "Hello Skill-Wanderer" > test.txt
```



The Test Flight: 2. The First Snapshot

Step 1: Frame the Shot

```
> git add .
```



Step 2: Take the Photo

```
> git commit -m "Testing my setup"
```

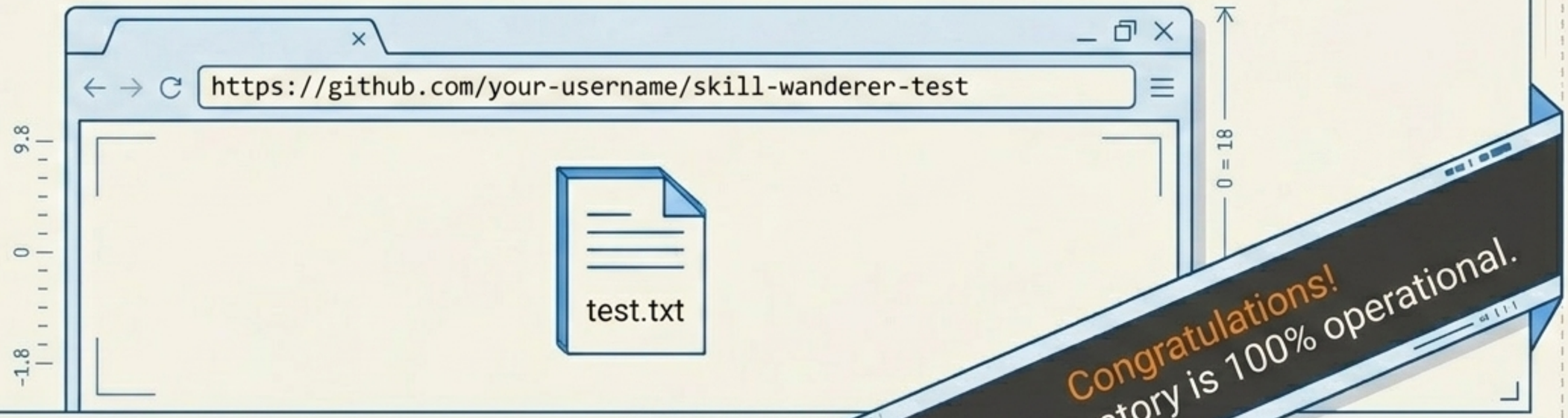
Saves the snapshot locally with your attached message.

The Test Flight: 3. Pushing to Orbit

This tests if your authentication from Phase 3b was successful.

```
> gh repo create skill-wanderer-test --public --source=. --remote=origin --push
```

If successful, your terminal will output a URL. Open it to verify.

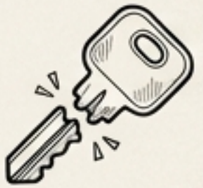


DOC. REF: AE-24-GIT-PUSH-TO-ORBIT

STATUS: OPERATIONAL

Mission Control Diagnostics

Troubleshooting common blank slate anomalies.

Symptom (Error)	Diagnosis (Meaning)	Treatment (Fix)
‘git’ is not recognized...	Git isn’t in your computer’s PATH.	Restart your Terminal. If that fails, re-install and check ‘Add to PATH’.
Permission denied (publickey)	SSH or Authentication misconfiguration.	Stick to HTTPS during the ‘gh auth login’ process. 
Author identity unknown	Git doesn’t know who is attempting to save the code.	Go back to Phase 4 and run the ‘git config’ commands.