PeiDocker Terminal GUI - Simple Mode Wizard Design

Claude Code

July 23, 2025

1 Simple Mode Wizard Flow Overview

The Simple Mode provides a step-by-step wizard interface that guides users through creating a basic PeiDocker configuration. The wizard consists of 15 main steps, each focusing on a specific aspect of the container configuration.

1.1 Wizard Flow Diagram

2 Individual Screen Designs

2.1 Step 1: Project Information

2.2 Step 2: SSH Configuration

2.3 Step 3: SSH User Configuration

2.4 Step 4: SSH Root Access

2.5 Step 5: Proxy Configuration

2.6 Step 6: APT Configuration

2.7 Step 7-8: Port Mapping & Environment Variables

2.8 Step 9: Device Configuration

2.9 Steps 10-11: Mount Configuration

2.10 Steps 12-13: Entry Point Configuration

2.11 Step 14: Custom Scripts Configuration

2.12 Step 15: Configuration Summary

3 Navigation and Interaction Patterns

3.1 Common UI Elements

- **Progress Bar**: Shows current step and overall progress
- **Navigation Buttons**: Back, Next/Save, Cancel consistently placed
- **Keyboard Shortcuts**: TAB (navigate), ENTER (next/select), ESC (cancel)
- **Help Text**: Blue info boxes explaining options and providing examples
- **Warning Messages**: Orange warning boxes for important notices
- **Input Validation**: Real-time validation with error highlighting

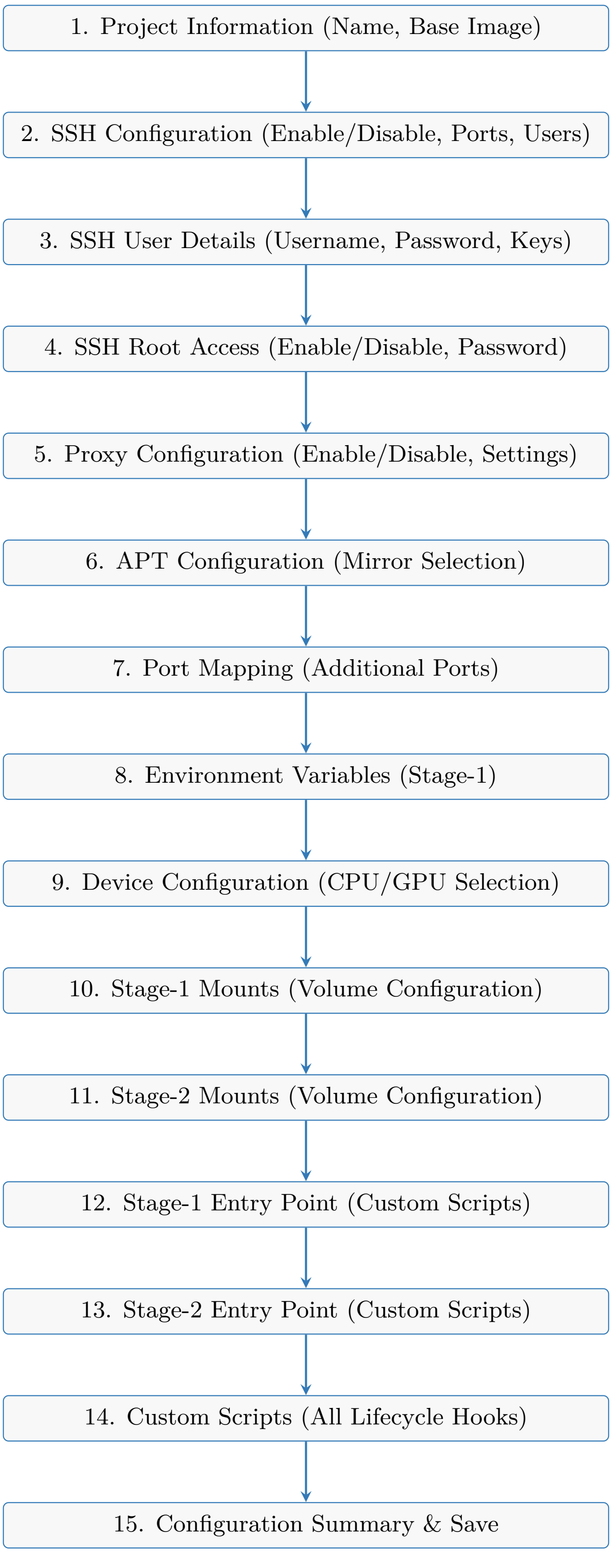
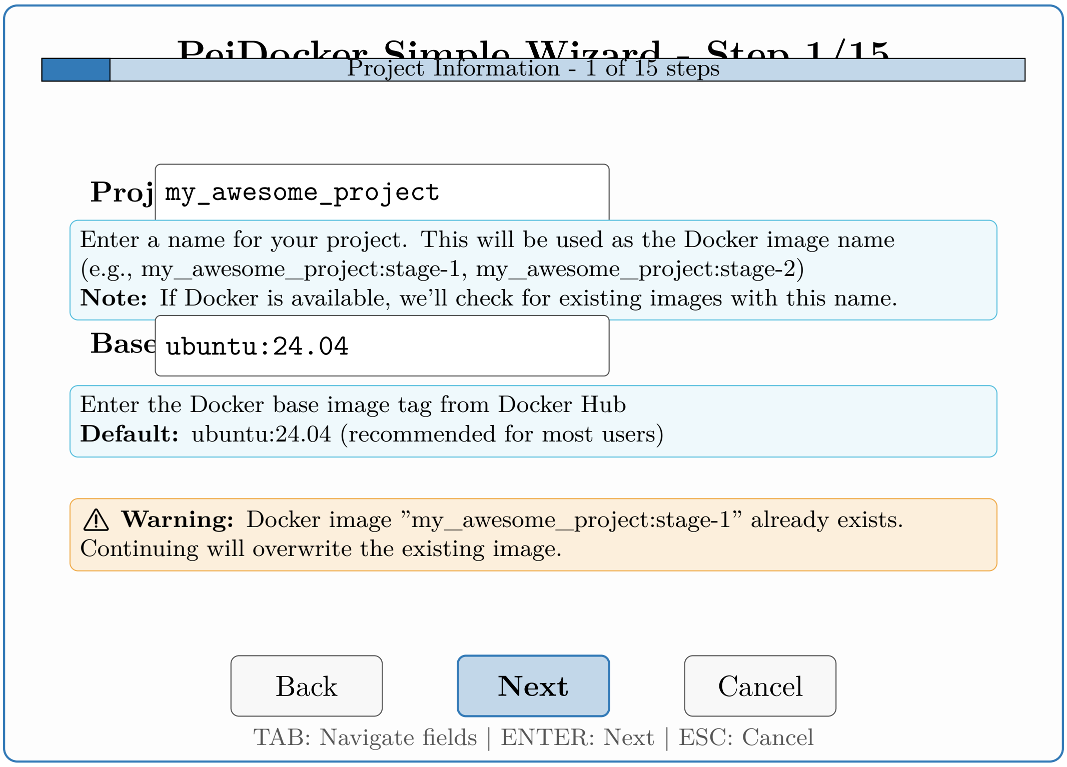

1. Project Information (Name, Base Image)
2. SSH Configuration (Enable/Disable, Ports, Users)
3. SSH User Details (Username, Password, Keys)
4. SSH Root Access (Enable/Disable, Password)
5. Proxy Configuration (Enable/Disable, Settings)
6. APT Configuration (Mirror Selection)
7. Port Mapping (Additional Ports)
8. Environment Variables (Stage-1)
9. Device Configuration (CPU/GPU Selection)
10. Stage-1 Mounts (Volume Configuration)
11. Stage-2 Mounts (Volume Configuration)
12. Stage-1 Entry Point (Custom Scripts)
13. Stage-2 Entry Point (Custom Scripts)
14. Custom Scripts (All Lifecycle Hooks)
15. Configuration Summary & Save

Figure 1: Complete Simple Mode Wizard Flow



PeiDocker Simple Wizard - Step 1/15

Project Information - 1 of 15 steps

Proj `my_awesome_project`

Enter a name for your project. This will be used as the Docker image name (e.g., my_awesome_project:stage-1, my_awesome_project:stage-2)
Note: If Docker is available, we'll check for existing images with this name.

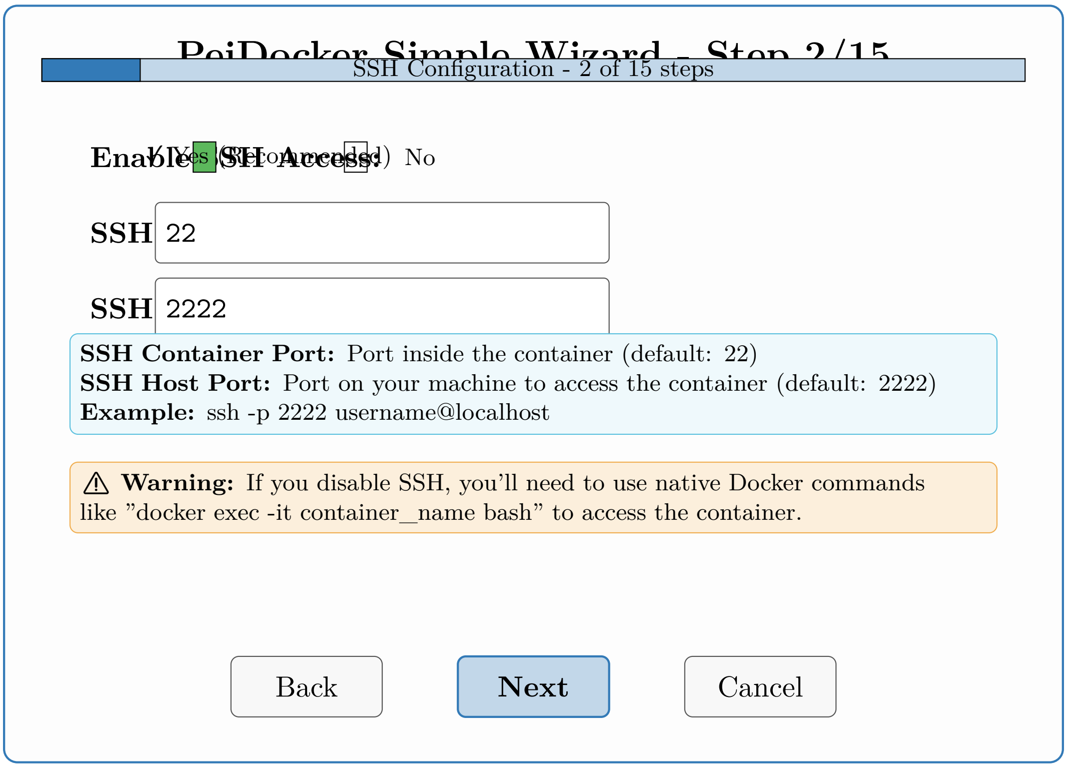
Base `ubuntu:24.04`

Enter the Docker base image tag from Docker Hub
Default: ubuntu:24.04 (recommended for most users)

⚠ **Warning:** Docker image "my_awesome_project:stage-1" already exists. Continuing will overwrite the existing image.

TAB: Navigate fields | ENTER: Next | ESC: Cancel

Figure 2: Step 1: Project Information Screen



PeiDocker Simple Wizard - Step 2/15

SSH Configuration - 2 of 15 steps

Enable SSH Access: ✓ Yes ☑ (Recommended) □ No

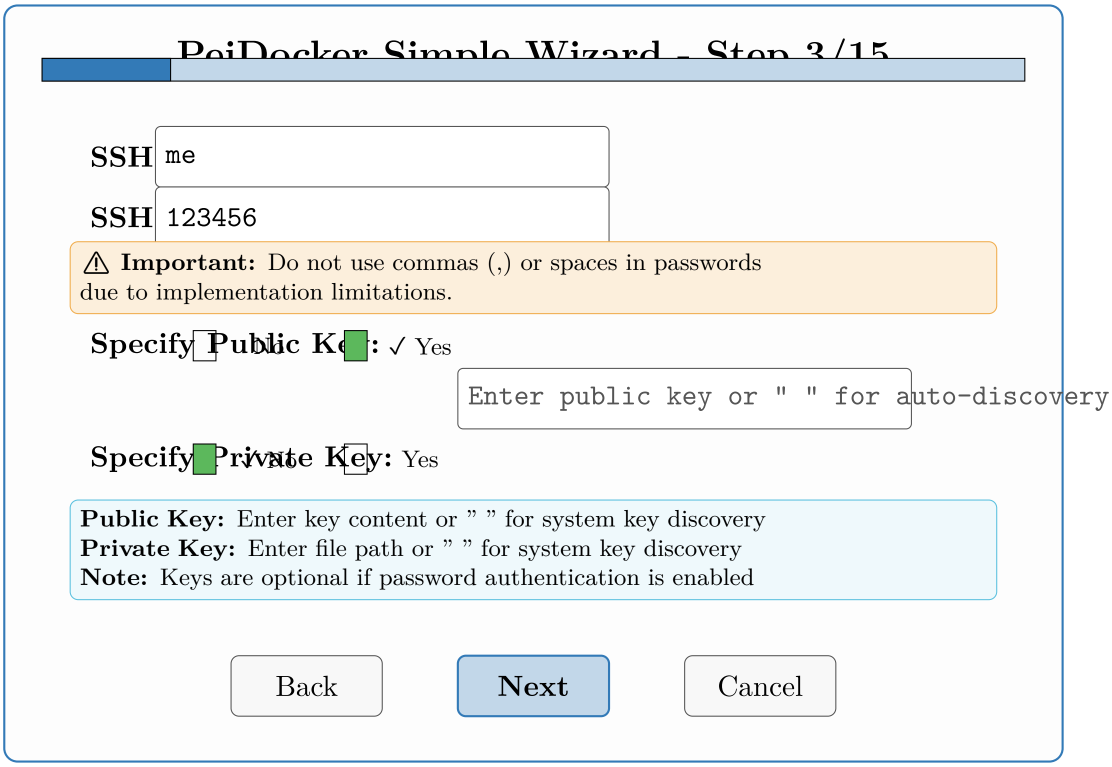
SSH 22

SSH 2222

SSH Container Port: Port inside the container (default: 22)
SSH Host Port: Port on your machine to access the container (default: 2222)
Example: ssh -p 2222 username@localhost

⚠ **Warning:** If you disable SSH, you'll need to use native Docker commands like "docker exec -it container_name bash" to access the container.

Figure 3: Step 2: SSH Configuration Screen



PeiDocker Simple Wizard - Step 3/15

SSH me

SSH 123456

⚠ **Important:** Do not use commas (,) or spaces in passwords
due to implementation limitations.

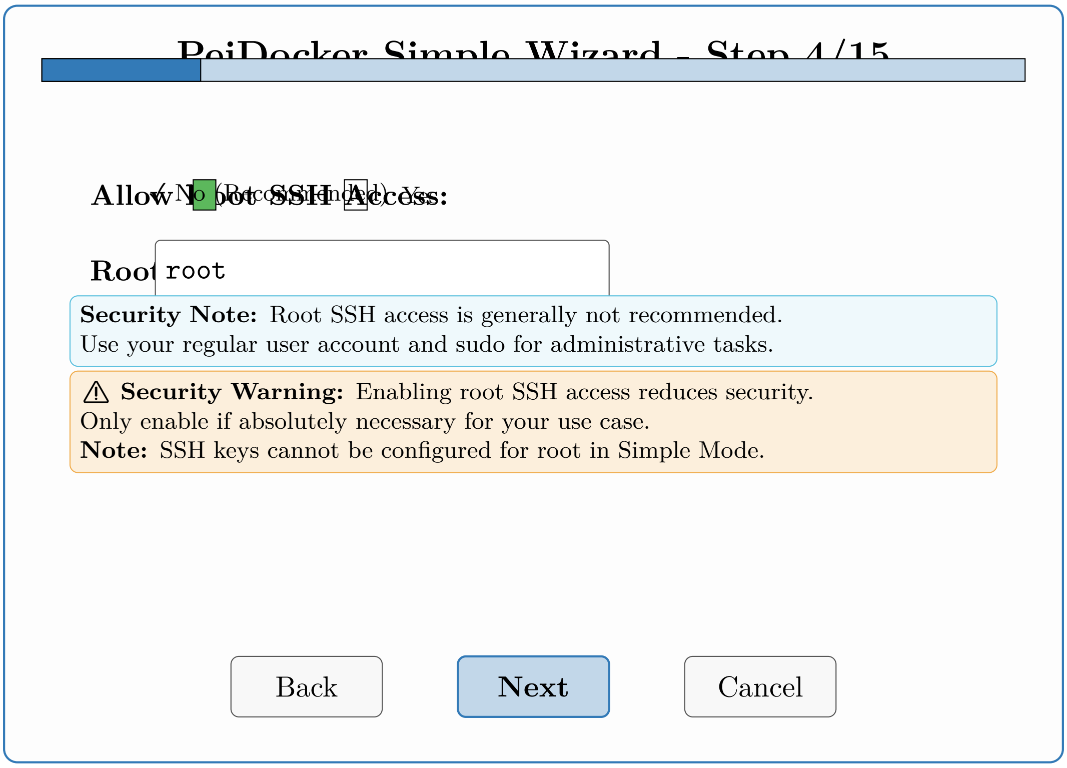
Specify Public Key: □ No 🟩 ✓ Yes

Enter public key or " " for auto-discovery

Specify Private Key: 🟩 ✓ No □ Yes

Public Key: Enter key content or " " for system key discovery
Private Key: Enter file path or " " for system key discovery
Note: Keys are optional if password authentication is enabled

Figure 4: Step 3: SSH User Configuration Screen



PeiDocker Simple Wizard - Step 4/15

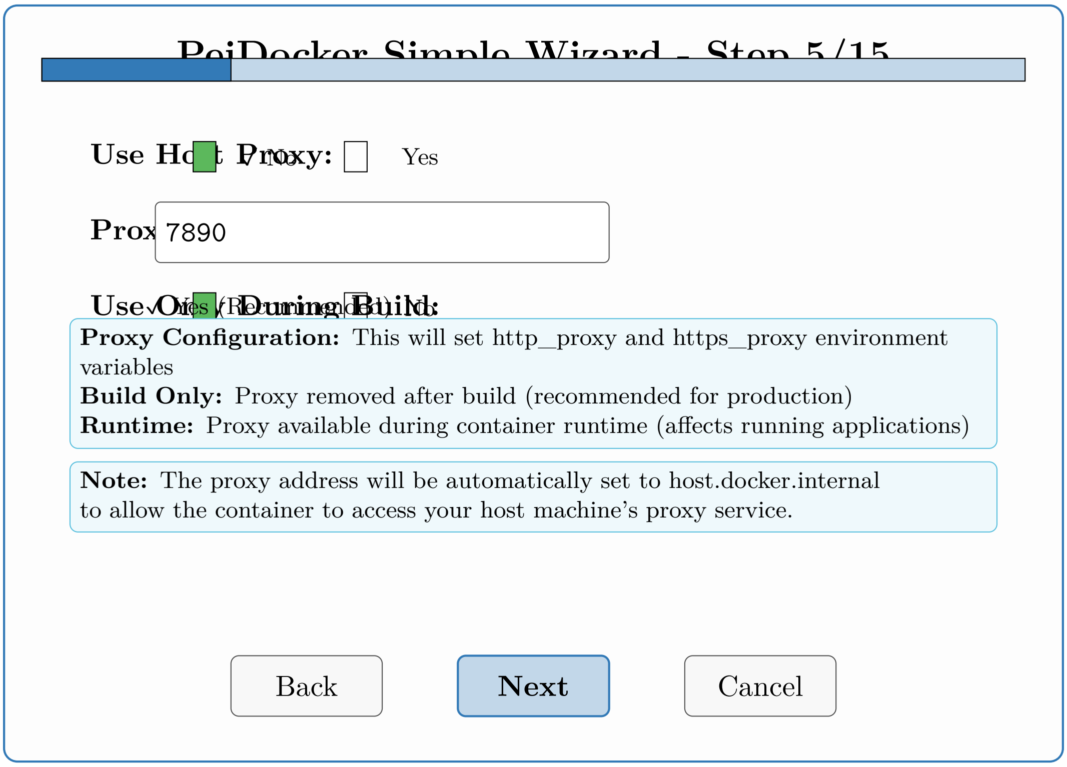
Allow Root SSH Access: 🟩 ✓ No (Recommended) □ Yes

Root root

Security Note: Root SSH access is generally not recommended.
Use your regular user account and sudo for administrative tasks.

⚠ **Security Warning:** Enabling root SSH access reduces security.
Only enable if absolutely necessary for your use case.
Note: SSH keys cannot be configured for root in Simple Mode.

Figure 5: Step 4: SSH Root Access Screen



PeiDocker Simple Wizard - Step 5/15

Use Host Proxy: 🟩 ✓ No □ Yes

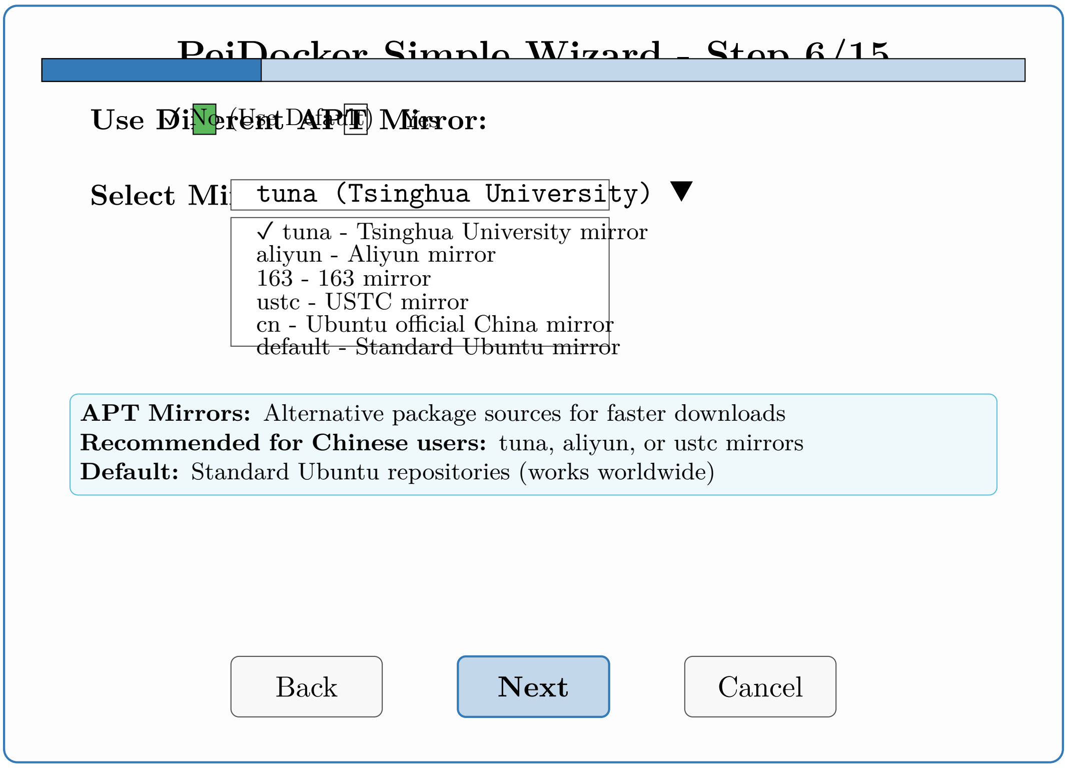
Prox 7890

Use Only During Build: 🟩 ✓ Yes (Recommended) □ No

Proxy Configuration: This will set http_proxy and https_proxy environment variables
Build Only: Proxy removed after build (recommended for production)
Runtime: Proxy available during container runtime (affects running applications)

Note: The proxy address will be automatically set to host.docker.internal
to allow the container to access your host machine's proxy service.

Figure 6: Step 5: Proxy Configuration Screen



PeiDocker Simple Wizard - Step 6/15

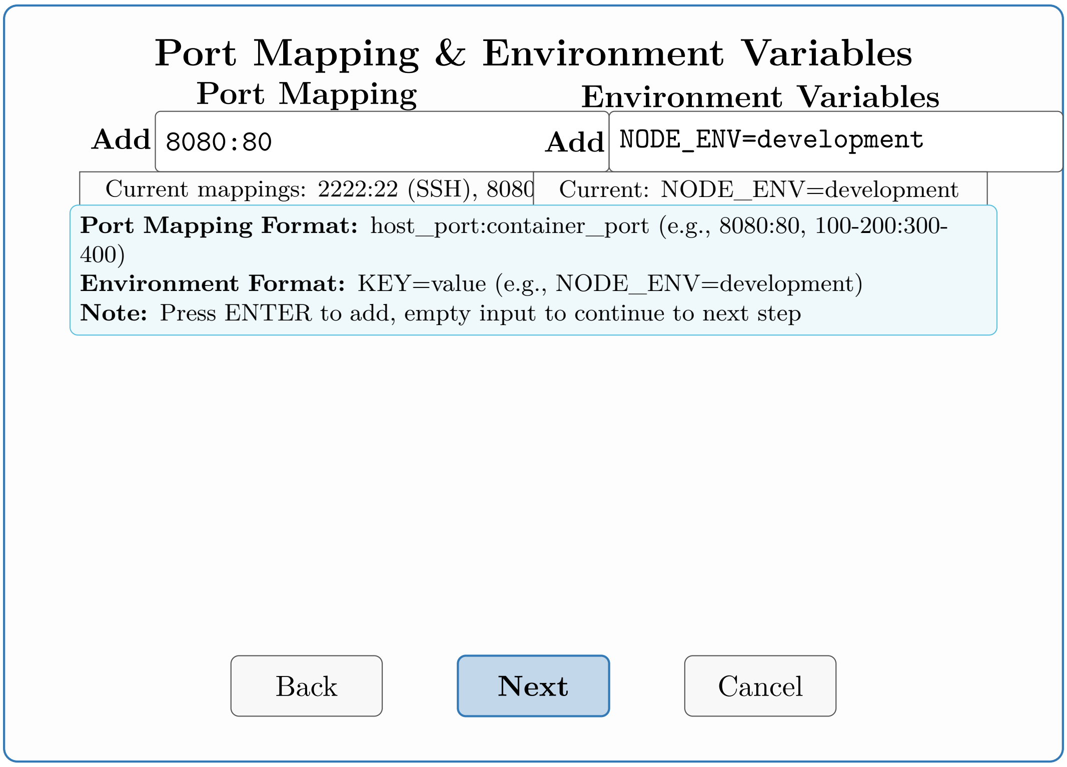
Use Different APT Mirror: 🟩 ✓ No (Use Default) □ Yes

Select Mi tuna (Tsinghua University) ▼

✓ tuna - Tsinghua University mirror
aliyun - Aliyun mirror
163 - 163 mirror
ustc - USTC mirror
cn - Ubuntu official China mirror
default - Standard Ubuntu mirror

APT Mirrors: Alternative package sources for faster downloads
Recommended for Chinese users: tuna, aliyun, or ustc mirrors
Default: Standard Ubuntu repositories (works worldwide)

Figure 7: Step 6: APT Configuration Screen



Port Mapping & Environment Variables

Port Mapping

Add `8080:80`

Current mappings: 2222:22 (SSH), 8080

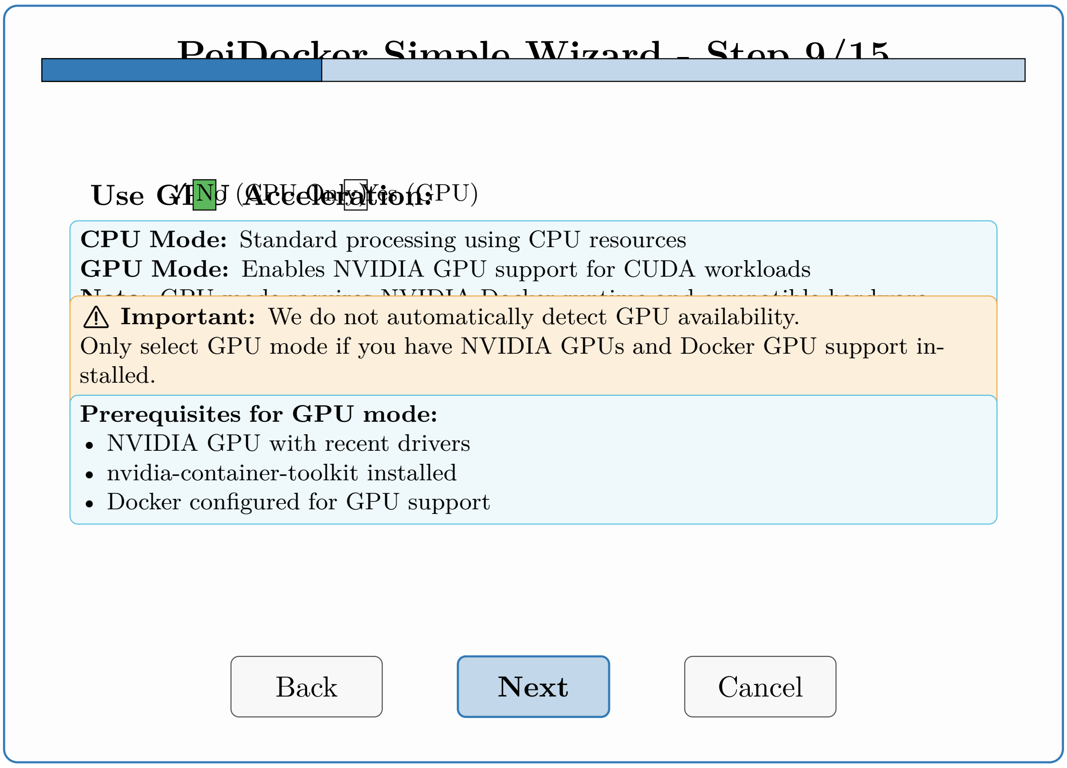
Environment Variables

Add `NODE_ENV=development`

Current: NODE_ENV=development

Port Mapping Format: host_port:container_port (e.g., 8080:80, 100-200:300-400)
Environment Format: KEY=value (e.g., NODE_ENV=development)
Note: Press ENTER to add, empty input to continue to next step

Figure 8: Step 7-8: Port Mapping & Environment Variables



PciDocker Simple Wizard - Step 9/15

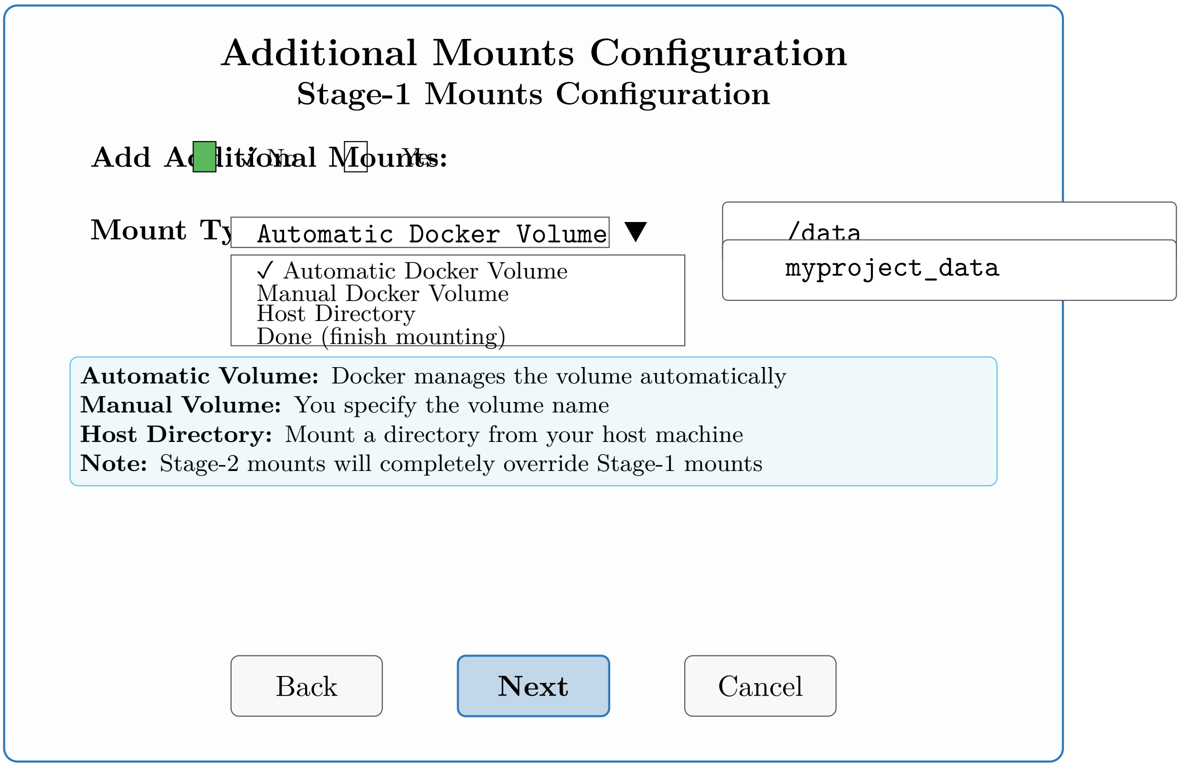
Use GPU Acceleration: ✓ No (CPU Only) □ Yes (GPU)

CPU Mode: Standard processing using CPU resources
GPU Mode: Enables NVIDIA GPU support for CUDA workloads
Note: GPU mode requires NVIDIA Docker runtime and compatible hardware

⚠ **Important:** We do not automatically detect GPU availability.
Only select GPU mode if you have NVIDIA GPUs and Docker GPU support installed.

Prerequisites for GPU mode:
- NVIDIA GPU with recent drivers
- nvidia-container-toolkit installed
- Docker configured for GPU support

Figure 9: Step 9: Device Configuration Screen



Additional Mounts Configuration

Stage-1 Mounts Configuration

Add Additional Mounts: 🟩 ✓ No □ Yes

Mount Ty Automatic Docker Volume ▼

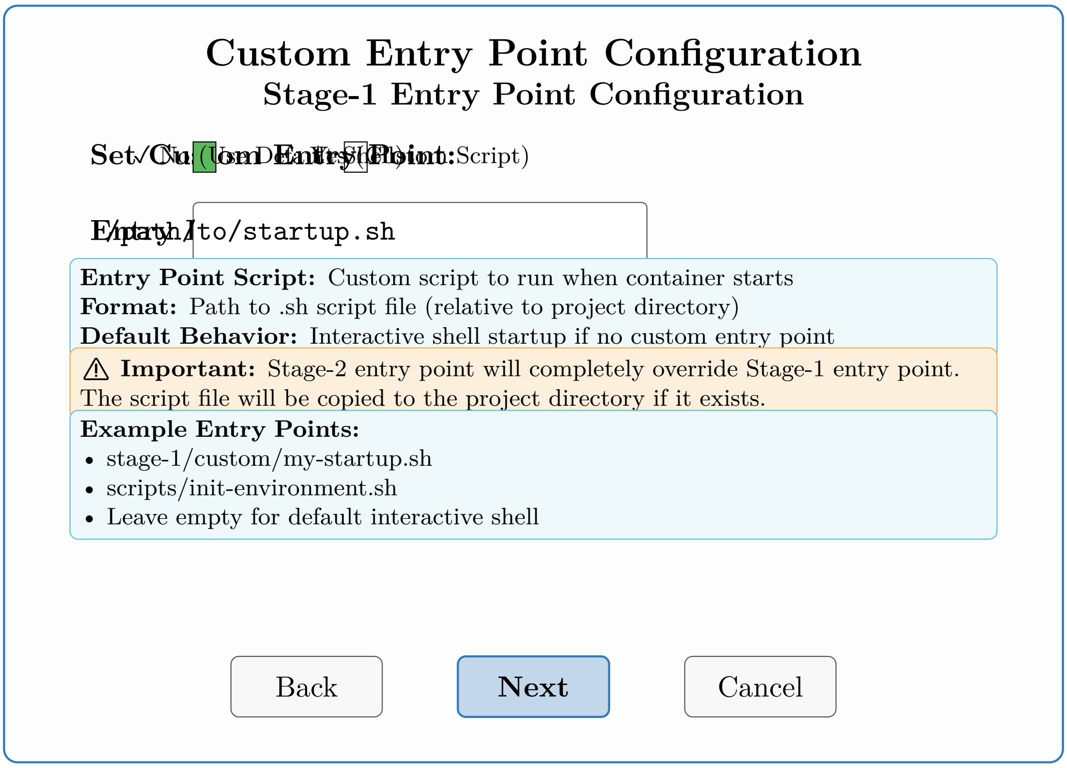
- ✓ Automatic Docker Volume
- Manual Docker Volume
- Host Directory
- Done (finish mounting)

/data

myproject_data

Automatic Volume: Docker manages the volume automatically
Manual Volume: You specify the volume name
Host Directory: Mount a directory from your host machine
Note: Stage-2 mounts will completely override Stage-1 mounts

Figure 10: Steps 10-11: Mount Configuration Screen



Custom Entry Point Configuration

Stage-1 Entry Point Configuration

Set Custom Entry Point: ✓ No (Use Default Shell) 🟩 □ Yes (Custom Script)

Entry /path/to/startup.sh

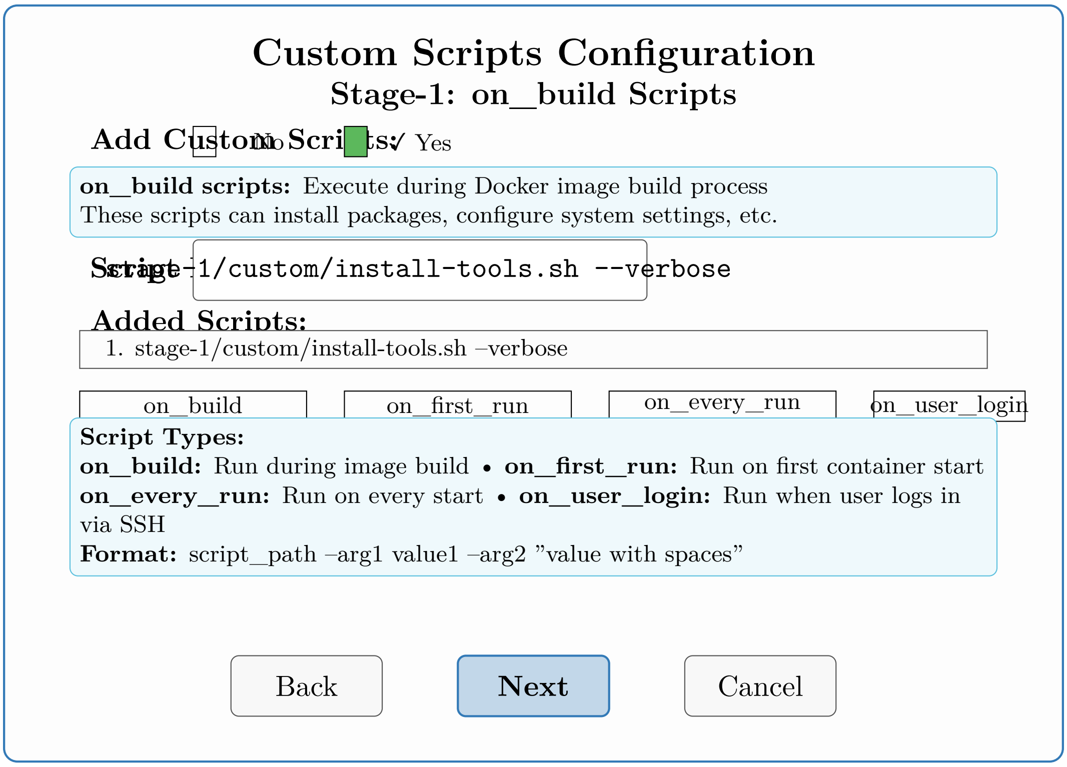
Entry Point Script: Custom script to run when container starts
Format: Path to .sh script file (relative to project directory)
Default Behavior: Interactive shell startup if no custom entry point

⚠ **Important:** Stage-2 entry point will completely override Stage-1 entry point. The script file will be copied to the project directory if it exists.

Example Entry Points:

- stage-1/custom/my-startup.sh
- scripts/init-environment.sh
- Leave empty for default interactive shell

Figure 11: Steps 12-13: Entry Point Configuration Screen



Custom Scripts Configuration

Stage-1: on_build Scripts

Add Custom Scripts: □ No 🟩 ✓ Yes

on_build scripts: Execute during Docker image build process
These scripts can install packages, configure system settings, etc.

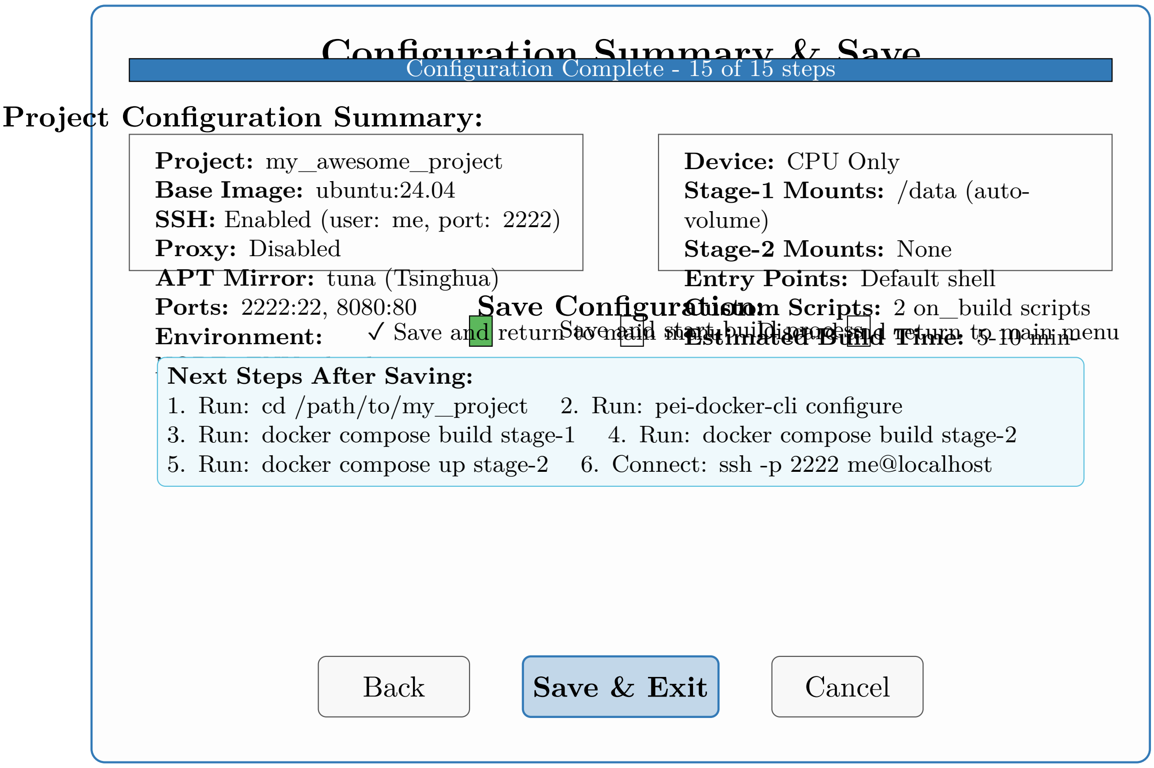
Script: `stage-1/custom/install-tools.sh --verbose`

Added Scripts:

1. stage-1/custom/install-tools.sh –verbose

Script Types:
on_build: Run during image build • **on_first_run:** Run on first container start
on_every_run: Run on every start • **on_user_login:** Run when user logs in via SSH
Format: script_path –arg1 value1 –arg2 ”value with spaces”

Figure 12: Step 14: Custom Scripts Configuration Screen



Configuration Summary & Save

Configuration Complete - 15 of 15 steps

Project Configuration Summary:

Project: my_awesome_project
Base Image: ubuntu:24.04
SSH: Enabled (user: me, port: 2222)
Proxy: Disabled
APT Mirror: tuna (Tsinghua)
Ports: 2222:22, 8080:80
Environment:

Device: CPU Only
Stage-1 Mounts: /data (auto-volume)
Stage-2 Mounts: None
Entry Points: Default shell
Custom Scripts: 2 on_build scripts
Estimated Build Time: 5-10 min

Save Configuration:
🟩 ✓ Save and return to main menu
□ Save and start build process
□ Discard and return to main menu

Next Steps After Saving:
1. Run: cd /path/to/my_project 2. Run: pei-docker-cli configure
3. Run: docker compose build stage-1 4. Run: docker compose build stage-2
5. Run: docker compose up stage-2 6. Connect: ssh -p 2222 me@localhost

Figure 13: Step 15: Configuration Summary & Save Screen

3.2 Input Patterns

- **Text Fields**: Standard text input with placeholder text
- **Radio Buttons**: Single selection with checkmarks for selected options
- **Dropdowns**: List selection with arrow indicator
- **List Management**: Add items with ENTER, show current items, remove with empty input
- **File Paths**: Support for relative paths and special syntax