Ubuntu Docker VM

Overview

This project sets up an isolated Ubuntu environment in a Docker container for development work. The setup provides complete isolation from the host filesystem while maintaining internet access.

Features

- Ubuntu 22.04 LTS base image
- Complete isolation from host filesystem
- Internet access enabled
- Persistent storage in /workspace
- Development tools pre-installed:
  - **Languages**: Python 3.10.12, Node.js 12.22.9
  - **Editors**: Neovim, Vim, Nano
  - **Build Tools**: gcc, g++, make, cmake
  - **Python Tools**: pip, ipython, black, flake8, pytest, poetry
  - **Version Control**: Git
- Non-root user (aiuser) with limited sudo access
- Ready for AI coding assistants (Claude Code, GitHub Copilot, etc.)

Prerequisites

- Docker installed and running
- Docker Compose installed
- Sufficient disk space (at least 20GB free)

Quick Start

1. **Initial Setup** (first time only):

   ```
   ./scripts/start.sh
   ./scripts/setup-dev-env.sh  # Installs dev tools
   ```

2. **Daily Usage:**

   ```
   ./scripts/start.sh          # Start container
   ./scripts/connect.sh        # Connect to container
   # ... do your work ...
   ./scripts/stop.sh           # Stop when done (optional)
   ```

3. **Check Installed Tools:**

```
./scripts/check-tools.sh
```

File Structure

```
ubuntu-docker-vm/
├── README.md
├── PLAN.md
├── Dockerfile
├── docker-compose.yml
├── rollback.sh
├── test-setup.sh
├── backups/
├── storage/
├── scripts/
│   ├── start.sh
│   ├── stop.sh
│   ├── connect.sh
│   ├── mount-storage.sh
│   └── resize-volume.sh
└── config/
    └── security.json
```

Management Scripts

start.sh

Builds and starts the Ubuntu VM container. If the container already exists, it will just start it without rebuilding.

stop.sh

Stops and removes the running container.

connect.sh

Opens an interactive bash session in the container.

setup-dev-env.sh

Installs additional development tools in the container (Python packages, editors, etc.)

check-tools.sh

Verifies all installed development tools and their versions.

resize-volume.sh

Resizes the storage volume. Usage:

```
./scripts/resize-volume.sh 30  # Resize to 30GB
```

rollback.sh

Removes all created Docker resources and restores backed up resources if available.

Testing

Run the automated test script to verify the setup:

```
./test-setup.sh
```

Security

- Container runs as non-root user (aiuser)
- Limited sudo permissions (only apt-get and apt)
- Network isolation with bridge network
- Resource limits enforced (2 CPUs, 4GB RAM)
- No access to host filesystem

Troubleshooting

Docker not running

Ensure Docker Desktop or Docker daemon is running before executing scripts.

Permission denied

Make sure all scripts are executable:

```
chmod +x ./scripts/*.sh ./test-setup.sh ./rollback.sh
```

Storage issues

The storage uses a sparse file system. If you need more space:

```
./scripts/resize-volume.sh 40  # Resize to 40GB
```

Notes

- The sparse file system means the 20GB is not immediately allocated
- Storage only uses disk space as data is written
- Internet connectivity is enabled by default

- Development tools (Python, Node.js, Git) are pre-installed
- You can install additional tools including AI assistants as needed