Ansible Workshop - Exercises

Basics

Get to know Ansible and learn to write your first Ansible Playbooks.

1 - Check the Prerequisites

Objective

- Understand the lab topology and how to access the environment.
- Understand how to work the workshop exercises
- Understand challenge labs

These first few lab exercises will be exploring the command-line utilities of the Ansible Automation Platform. This includes

- ansible-core - the base executable that provides the framework, language and functions that underpin the Ansible Automation Platform. It also includes various cli tools like `ansible`, `ansible-playbook` and `ansible-doc`. Ansible Core acts as the bridge between the upstream community with the free and open source Ansible and connects it to the downstream enterprise automation offering from Red Hat, the Ansible Automation Platform.
- ansible-navigator - a command line utility and text-based user interface (TUI) for running and developing Ansible automation content, especially for running Execution Environments. We won't cover this tool specifically in the first day, but you can always do all exercises with the Navigator if you want to.

> **Info**
>
> We will be using especially the *ansible-core* executable and the CLI tools it provides, as currently (Q3/2025) it is still the main interface to interact with Ansible.

In the (near) future this will be replaced/supplemented by the *Ansible Navigator*, which on the one hand brings more useful additional features and in the end serves a much greater purpose than just be a drop in replacement or alias to the currently used Ansible utilities. It requires a broader introduction and explanation regarding the use of containers and collections, which we will discuss on workshop day 3.
Still, although we will be using the *ansible-core* executable in all exercises, it is shown how to also achieve everything using the `ansible-navigator` utility in a separate tab.

Ansible

Info

Although the tab is titled **Ansible**, this can be any of the *classic* utilities provided by the *ansible-core* executable. The **Navigator** tab uses the same utils but acts as an abstraction layer.

Guide

Your Lab Environment

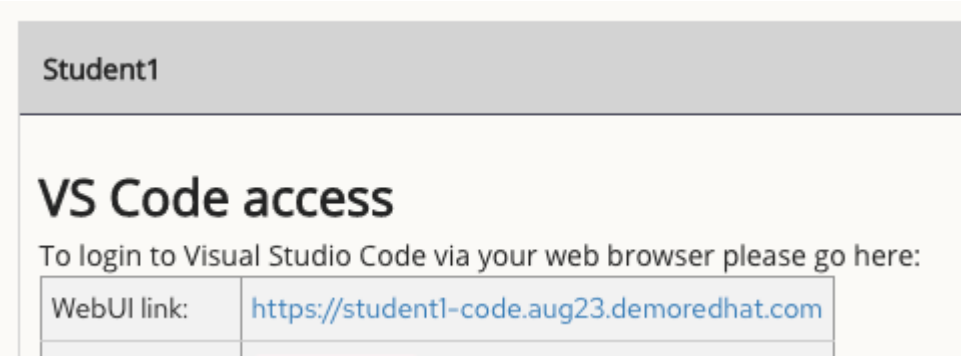
In this lab you work in a pre-configured lab environment. You will have access to the following hosts:

Role	Inventory name
Ansible Control Host	ansible-1
Managed Host 1	node1
Managed Host 2	node2
Managed Host 3	node3

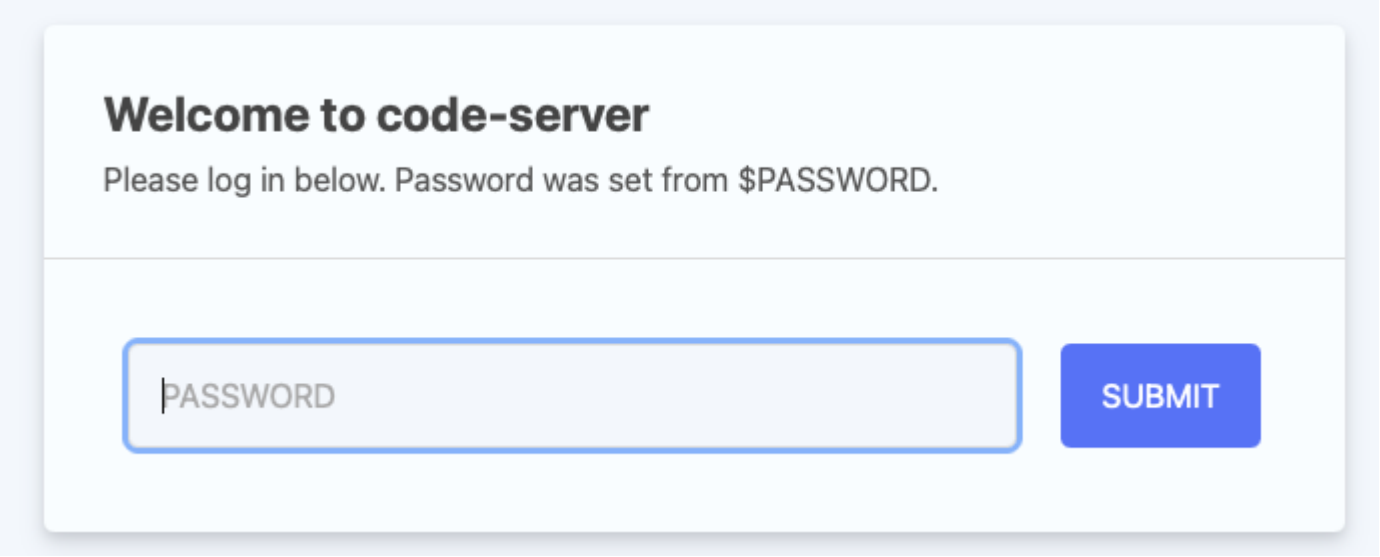
Every host is reachable via SSH.

Step 1 - Access the Environment

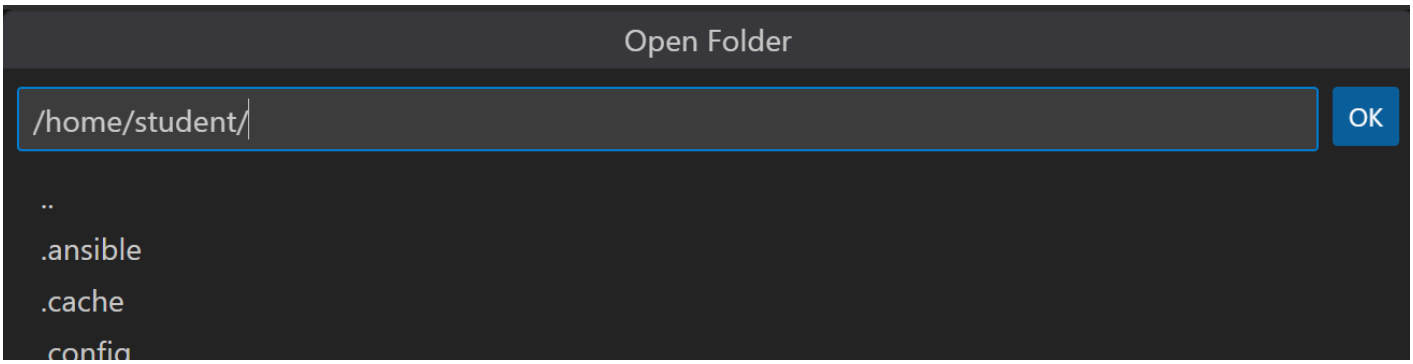
Connect to Visual Studio Code from the Workshop launch page (provided by your instructor). The password is provided below the WebUI link.



Type in the provided password to connect.

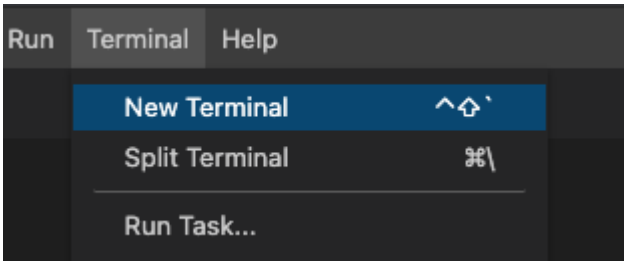


In the menu bar of VS Code, click on `File` → `Open Folder...` and open your *home directory* in Visual Studio Code (this folder should already be shown in the *Open Folder* pop-up and called `/home/student`. This will reload your browser window.



Step 2 - Using the Terminal

Open a terminal in Visual Studio Code:



Navigate to the `lab_inventory` directory on the Ansible control node terminal.

```
[student@ansible-1 ~]$ cd lab_inventory
[student@ansible-1 lab_inventory]$ pwd
/home/student/lab_inventory
[student@ansible-1 lab_inventory]$ ls
[student@ansible-1 lab_inventory]$
hosts
[student@ansible-1 lab_inventory]$ cd ..
[student@ansible-1 ~]$
```

- `~` - the tilde in this context is a shortcut for the home directory, i.e. `/home/student`
- `cd` - Linux command to change directory
- `pwd` - Linux command for print working directory. This will show the full path to the current working directory.

Step 3 - Challenge Labs

You will soon discover that many chapters in this lab guide come with a "Challenge Lab" section. These labs are meant to give you a small task to solve using what you have learned so far. The solution of the task is shown underneath a warning sign.