AnsibleLabs

Ansible Lab Exercises for learning and testing Ansible automation with the Fedora Remix Lab environment.

Overview

This project provides Ansible configuration files, inventory, and example playbooks designed to work with the Fedora Remix Lab virtual lab environment. It's perfect for learning Ansible automation, testing playbooks, and practicing infrastructure-as-code concepts.

Prerequisites

Before using AnsibleLabs, you need to have the Fedora Remix Lab environment set up and running.

Using Fedora Lab Live Remix (Recommended)

The easiest way to get started is to use the **Fedora Lab Live Remix** image, which comes pre-configured with:

- Ansible and ansible-navigator
- Virtualization tools (KVM/libvirt)
- Fedora Remix Lab management scripts
- All required packages and dependencies

To use the Fedora Lab Live Remix:

1. **Download the Fedora Lab Live Remix ISO** from the Fedora Remix project
    - The ISO is built using the kickstart file: `Fedora_Remix/Setup/Kickstarts/FedoraRemixLab.ks`
2. **Boot from the Live ISO** or install it to disk
3. **Create and start the lab VMs**:

```
cd ~/Github/Fedora_Remix_Lab
sudo ./create-lab-vms.sh
sudo ./start-lab-vms.sh
sudo ./manage-hosts.sh add
```

4. **Verify VMs are running**:

```
sudo ./lab-status.sh
```

Once the lab VMs are running, you're ready to use AnsibleLabs!

> **Note:** For manual installation on other systems, see the Appendix: Manual Installation section.

Project Structure

```
AnsibleLabs/
├── README.md                     # This file
├── ansible.cfg                  # Ansible configuration
├── ansible-navigator.yml        # ansible-navigator configuration
├── inventory                    # Ansible inventory file
├── ping-and-info.yml            # Example playbook (ping + system info)
└── test-ping.sh                 # Script to test connectivity
```

Getting Started

1. Verify Lab Environment

First, ensure your Fedora Remix Lab VMs are running and accessible:

```
# From Fedora_Remix_Lab directory
cd ../Fedora_Remix_Lab
sudo ./lab-status.sh
```

You should see both `FedoraLab1` and `FedoraLab2` in a running state.

2. Test Connectivity

Test Ansible connectivity to all hosts using the ping module:

Option A: Using the test script

```
cd /Users/travis/Github/AnsibleLabs
./test-ping.sh
```

Option B: Using ansible command directly

```
ansible all -m ping
```

Option C: Using ansible-navigator

```
ansible-navigator run test-ping.sh
```

Expected output should show `pong` responses from both nodes:

```
fedoralab1.example.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
fedoralab2.example.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

3. Run the Example Playbook

Run the included playbook to test connectivity and gather system information:

Using ansible-playbook:

```
ansible-playbook ping-and-info.yml
```

Using ansible-navigator:

```
ansible-navigator run ping-and-info.yml
```

This playbook will:

- Test connectivity with ping
- Display hostname
- Display IP address
- Display memory information (total, free, used)

Configuration Details

ansible.cfg

The `ansible.cfg` file provides default settings:

- Sets inventory file location
- Disables host key checking (useful for lab environments)
- Configures privilege escalation (sudo)
- Enables SSH connection pipelining for better performance
- Sets color output for better readability
- Configures Python interpreter discovery (`interpreter_python = auto`)
- Suppresses deprecation warnings (`deprecation_warnings = False`)

Note: The `interpreter_python = auto` setting allows Ansible to automatically discover the correct Python interpreter on each host. The `deprecation_warnings = False` setting suppresses deprecation warnings (such as warnings about deprecated `auto_legacy_silent` option) to provide cleaner output in lab environments.

Deprecation Note: The `auto_legacy_silent` option is deprecated as of Ansible Core 2.19 and will be removed in version 2.21. It now has the same effect as `auto`, so we use `auto` instead for future compatibility.

inventory

The inventory file defines your lab hosts:

- **Control node**: `localhost` (local connection)
- **Lab nodes**:
  - `fedoralab1.example.com` (192.168.100.10)
  - `fedoralab2.example.com` (192.168.100.11)

Connection settings:

- User: `ansibleuser`
- Password: `Automation!`
- Sudo: Enabled (passwordless)

ansible-navigator.yml

Configuration for ansible-navigator:

- **Execution Environment**: `quay.io/ansible/awx-ee:latest`
- **Output**: Standard output (stdout) enabled
- **Artifacts**: Disabled (no playbook artifacts created)

Common Tasks

List all hosts in inventory

```
ansible all --list-hosts
```

Run ad-hoc commands

```
# Check uptime on all hosts
ansible all -a "uptime"

# Check disk usage
ansible all -a "df -h"

# Install a package
ansible all -m dnf -a "name=vim state=present" -b
```

Run playbooks

```
# Using ansible-playbook
ansible-playbook ping-and-info.yml

# Using ansible-navigator
ansible-navigator run ping-and-info.yml

# Run on specific hosts
ansible-playbook ping-and-info.yml --limit nodes
```

Test specific hosts

```
# Test single host
ansible fedoralab1.example.com -m ping

# Test specific group
ansible nodes -m ping
```

Lab Environment Details

The Fedora Remix Lab provides:

VM Name	IP Address	FQDN	Hostname
FedoraLab1	192.168.100.10	fedoralab1.example.com	fedoralab1
FedoraLab2	192.168.100.11	fedoralab2.example.com	fedoralab2

VM Specifications:

- Memory: 1 GB each
- vCPUs: 2 each
- Network: `labnet` (192.168.100.0/24)

User Account:

- Username: `ansibleuser`
- Password: `Automation!`

- Sudo: Passwordless (`NOPASSWD: ALL`)

Troubleshooting

Connection refused or timeout

1. **Verify VMs are running:**

   ```
   cd ../Fedora_Remix_Lab
   sudo ./lab-status.sh
   ```

2. **Start VMs if needed:**

   ```
   sudo ./start-lab-vms.sh
   ```

3. **Check network connectivity:**

   ```
   ping fedoralab1.example.com
   ping fedoralab2.example.com
   ```

SSH authentication issues

1. **Verify /etc/hosts entries:**

   ```
   cd ../Fedora_Remix_Lab
   ./manage-hosts.sh status
   ```

2. **Add hosts entries if missing:**

   ```
   sudo ./manage-hosts.sh add
   ```

Permission denied errors

The inventory is configured with password authentication. If you encounter issues:

- Verify the password in the inventory matches the VM password (`Automation!`)
- Check that the user `ansibleuser` exists on the VMs
- Ensure sudo is configured for passwordless access

ansible-navigator issues

If ansible-navigator fails to pull the image:

```
# Pull the execution environment image manually
podman pull quay.io/ansible/awx-ee:latest
```

Next Steps

Now that you have the basic setup working, you can:

1. **Create your own playbooks** - Start with simple tasks and build up
2. **Explore Ansible modules** - Try different modules like `copy`, `template`, `service`, etc.
3. **Learn about roles** - Organize your playbooks into reusable roles
4. **Practice with collections** - Use Ansible collections for specific technologies
5. **Experiment with variables** - Use group_vars and host_vars for configuration

Resources

- Ansible Documentation
- Ansible Galaxy
- Fedora Remix Lab - Lab environment documentation
- Fedora Remix Project - Source for the Fedora Lab Live Remix ISO

Appendix: Manual Installation

If you're not using the Fedora Lab Live Remix and need to set up the environment manually:

Installing Ansible

On Fedora/RHEL:

```
sudo dnf install ansible
```

On macOS:

```
brew install ansible
```

Using pip:

```
pip install ansible
```

Installing ansible-navigator

On Fedora/RHEL:

```
sudo dnf install ansible-navigator
```

On macOS:

```
pip install ansible-navigator
```

Setting up Fedora Remix Lab Environment

If you’re not using the Live Remix, you’ll need to set up the lab environment manually:

1. **Install virtualization packages:**

```
sudo dnf install qemu-kvm libvirt virt-manager libguestfs-tools
virt-viewer
```

2. **Enable libvirtd:**

```
sudo systemctl enable --now libvirtd
```

3. **Clone and set up Fedora Remix Lab:**

```
git clone https://github.com/tmichett/Fedora_Remix_Lab.git
cd Fedora_Remix_Lab
sudo ./download-image.sh
sudo ./create-lab-vms.sh
sudo ./start-lab-vms.sh
sudo ./manage-hosts.sh add
```

Kickstart File Reference

The Fedora Lab Live Remix is built using the kickstart file located at:

```
Fedora_Remix/Setup/Kickstarts/FedoraRemixLab.ks
```

This kickstart file includes:

- Base Fedora Live image configuration
- Ansible and ansible-navigator installation
- Virtualization packages (@Virtualization)
- libguestfs tools for VM management
- Fedora Remix Lab scripts and tools
- All required dependencies

For more details about the kickstart configuration, see the Fedora Remix project.

License

This project is part of the Fedora Remix project.