



# Red Hat Training and Certification

## Instructor Demo Guide

Travis Michette

Version 1.0

# Table of Contents

Setting up a System for Demonstration for Course Delivery .....	1
1. SSH Setup .....	2
1.1. Setting up the VT Demonstration Connection .....	3
1.2. Setting up the KVM Demonstration Connection .....	4
1.3. Setting up the ROLE Demonstration Connection .....	4
2. Accessing the systems via SSH .....	5

# Setting up a System for Demonstration for Course Delivery

# Chapter 1. SSH Setup

There is a single playbook provided called **SSH\_Config\_Setup.yml**. This playbook can be run for locally hosted, nested, KVM foundations or it can be used with the ROLE/RHLS/VT environment.

*Listing 1. Variables for Playbook*

```

vars:
#   linux_username: travis   ### This variable is so that you can create the demo
#                             configuration for a user other than the Ansible User
## Course SKU Only Valid when using RHU Role courses / Not valid for RHLS
  course_sku: rh999 ①
  ROLE_IP: 150.239.52.xx ②
## Delivery Type Options VT, KVM, ROLE ##
  delivery_type: KVM ③
## IP Addresses for VT or KVM courses
  kvm_host_ip: 192.168.1.123 # IP Address for KVM Host (initial connection machine) ④
  kvm_foundation_ip: 192.168.122.71 # IP Address attached to NIC2 that is available
# on the "Externally configured NIC for F0" ⑤
  classroom_ip: "{{ ROLE_IP }}" ⑥

```

- ① Course SKU. Creates a complete SSH config file for all systems in the course.
- ② Classroom IP address provided from the ROLE/RHLS/VT interface. This is the public IP address you will be connecting to via SSH.
- ③ Type of delivery. **KVM**, **ROLE**, or **VT** and controls what files the playbook creates.
- ④ Used only for KVM deployments and is the "Physical Address" of the KVM host. This sets up the KVM host as a JumpBox to your nested Foundation0 on the KVM shared/NAT network for the secondary Internet facing NIC presented to the Foundation0 system.
- ⑤ Used only for KVM deployments and is the IP address provided by the **rht-external --status** command.
- ⑥ Recycles the **ROLE\_IP** address for use in deployed template files to keep variable names nice.

The playbook should be capable of running on all Linux distributions as well as MacOS. It has only been tested on Fedora, Red Hat, and MacOS.



### *Setting Variables and Updating **ansible.cfg***

It is important to setup variables in the playbook prior to running. It is equally important to make any changes and adjustments to the **ansible.cfg** and other adjustments to the playbook.



### *SSH Keys Required*

The playbook sets up SSH configuration files which are required for connection to remote systems. Currently these files rely on SSH keys with specific key names.

- **gls\_instructor\_id\_rsa** - The **id\_rsa** key located on Foundation0.
- **lab\_rsa** - The **lab\_rsa** key located on the workstation system.

It will be necessary to grab these files and place in `~/.ssh` with the appropriate permissions and name in order for the playbook to run.

## 1.1. Setting up the VT Demonstration Connection

This is the most common type of setup. It will also be the easiest for teaching and delivering VT courses as the playbook can be run multiple times to setup configuration files for all students and the instructor machine.

1. Modify the playbook with the **ROLE\_IP** variable.
2. Set the **delivery\_type** variable to be **VT**.
3. Execute the playbook using **ansible-playbook SSH\_Config\_Setup.yml**.

```
[travis@fedora Ansible_Playbooks]$ ansible-playbook SSH_Config_Setup.yml
BECOME password:

PLAY [Setup System for Github with Tools] *****

TASK [Gathering Facts] *****
ok: [localhost]

TASK [Create SSH Config.D Directory] *****
ok: [localhost]

TASK [Create SSH Course File for ROLE] *****
skipping: [localhost]

TASK [Create SSH Course File for KVM] *****
skipping: [localhost]

TASK [Create SSH Course File for VT] *****
ok: [localhost]

TASK [Perform Touch on File to Ensure it Exists] *****
changed: [localhost]

TASK [Update SSH Config] *****
ok: [localhost]

PLAY RECAP *****
localhost                : ok=5    changed=1    unreachable=0    failed=0
```

```
skipped=2    rescued=0    ignored=0
```

## 1.2. Setting up the KVM Demonstration Connection

1. Modify the playbook with the **kvm\_foundation\_ip** variable and the **kvm\_host\_ip** variable.
2. Set the **delivery\_type** variable to be **KVM**.
3. Execute the playbook using **ansible-playbook SSH\_Config\_Setup.yml**.

## 1.3. Setting up the ROLE Demonstration Connection

1. Modify the playbook with the **course\_sku** variable and the **ROLE\_IP** variable.
2. Set the **delivery\_type** variable to be **ROLE**.
3. Execute the playbook using **ansible-playbook SSH\_Config\_Setup.yml**.

## Chapter 2. Accessing the systems via SSH

Depending on whether you are using KVM, ROLE, or VT access, the SSH commands and capabilities for connecting will be different based on the playbook and configuration files generated.

### *KVM Hosted VMs*

1. Use SSH to connect to the workstation VM only **kvm\_workstation**.

```
[travis@fedora ~]$ ssh kvm_workstation
```

### *VT Hosted VMs*

1. Use SSH to connect to the workstation VM only **workstation-vt**.

```
[travis@fedora ~]$ ssh workstation-vt
```

### *ROLE/RHLS Hosted VMs*

This is the only deployment scenario where you can SSH to other VMs directly. For this to work, it is 100% needed that you place **ROLE** as deployment type and that all VMs have been defined in the **Templates/SSH\_Config.j2** file. Most all VMs have been created and defined, however, for new courses, it is possible to update the JINJA2 template with the newer VMs.

1. Use SSH to connect to the workstation or any other VM using **SKU\_<machine\_name>**

```
[travis@fedora ~]$ ssh rh124_workstation
```

#### *Connecting to Other Systems*

It is possible in the example above to substitute **workstation** with any other VM in the classroom environment.



- servera
- serverb
- serverc
- hub
- controller
- ... and more ...



#### *Accessing Student Systems*

It is possible to run the playbook multiple times for a course delivery to get all students setup so that those systems can be accessed remotely as an instructor.

Once the initial instructor VMs have been setup by setting the IP, SKU, and deployment type as ROLE, the playbooks can be re-run with the `-e` options to setup various students.

*Listing 2. Setting up a Student set of VMs for Instructor Access*

```
ansible-playbook SSH_Config_Setup.yml -e "ROLE_IP=150.75.250.26" -e  
"course_sku=yoda" ①
```

① In this instance, **yoda** is the name of the student user.

This will create a new config file in `~/.ssh/config.d/yoda.conf` as shown below.

```
[travis@fedora Ansible_Playbooks]$ cat ~/.ssh/config.d/yoda.conf  
Host yoda_*  
    StrictHostKeyChecking no  
    ForwardAgent yes  
    ForwardX11 no  
    PreferredAuthentications publickey  
    IdentityFile ~/.ssh/lab_rsa  
  
Host yoda_classroom  
    User instructor  
    Hostname 150.75.250.26  
    IdentityFile ~/.ssh/gls_instructor_id_rsa  
    Port 22022  
  
Host yoda_bastion  
    Hostname 172.25.252.1  
    User student  
    ProxyJump yoda_classroom  
  
Host yoda_workstation  
    Hostname workstation  
    User student  
    ProxyJump yoda_bastion
```

Now it is possible to SSH directly to student systems using the `<student_name>_<machine_name>` format as shown below.

```
[travis@fedora ~]$ ssh yoda_workstation
```



### SSH Cleanup Files

It is especially important to cleanup old SSH configuration files (especially when



creating them for various students). Running the playbooks based on the instructions above should re-write and update any configuration changes provided that for ROLE deployment, you choose to use SKU representation the same. It is recommended to use **lowercase** for the SKU.