

## Red Hat Lab – Chapter 10

Use Red Hat Lab Environment to complete the lab. Issue the following commands immediately before step 1:

```
history -c  
history -w
```

These commands should be repeated for each user@machine prompt. See boxes below.

Paste a screenshot in the box below of the command output from the command below. Include the command itself in the screenshot:

**lab grade ssh-review**

Issue the command `history` after the last step for each user@machine prompt. Copy the command itself and the full history of commands. Paste in the boxes below:

**production1@servera**

**root@serverb**

## Red Hat Lab – Chapter 11

Use **Red Hat Lab Environment** to complete the lab. Issue the following commands immediately before step 1:

```
history -c  
history -w
```

These commands should be repeated for each user@machine prompt. See boxes below.

Paste a screenshot in the box below of the command output from the command below. Include the command itself in the screenshot:

```
lab grade log-review
```

Issue the command `history` after the last step for student@serverb. Paste a screenshot of the history in the proper box below. Include the command itself and the full history of commands.

**student@serverb**

## Lab Manual

Use the VirtualBox RHELv9 virtual machine for this lab. Do not use the Red Hat Lab Environment. Issue the following commands in the Terminal window before starting the lab on the next page:

```
history -c  
history -w
```

Repeat these commands for root@RHELv8 if necessary.

Paste the results of the history command in the box at the end of the lab.

# Lab 15: Using SSH

---

In this lab, the RHELv9 VM will be considered as a powerful machine to act as the SSH server with IP address 10.0.0.1. The Ubuntu VM will be a thin client with limited resources and will act as the SSH client with IP address 10.0.0.2.

The Ubuntu client wishes to connect to the powerful RHELv9 server. The Ubuntu client will access programs that will run on the RHELv9 server but will see them displayed on its Ubuntu display. The RHELv9 server will provide the CPU and RAM resources to run the programs.

In this scenario, the RHELv9 VM is the server or remote machine. The Ubuntu VM is the client or local machine (where the user is actually sitting).

1. Check to see if the ssh client is installed on the Ubuntu machine. Just type **ssh** at a terminal. If you see a help screen, then ssh is installed.
2. Check to see if the ssh server is installed on the RHELv9 machine. Type **service sshd status** to see if the ssh server is installed and active. To activate the service, use **service sshd start**. If the service is not installed, run **yum install openssh-server**.
3. For convenience, rename both VMs as follows:
  - a. RHELv9 VM: **sudo hostname RHELv9-remote-server-10.0.0.1**
  - b. Ubuntu VM: **sudo hostname Ubuntu-local-client-10.0.0.2**
4. Before continuing, change the IP addresses of both VMs to match what is indicated above.
5. Now try to establish an ssh session between the Ubuntu client and the RHELv9 server: **ssh 10.0.0.1**
  - a. Since we don't have the server's public key, we can't verify we are truly talking to the server. We will take a risk and say yes to continue connecting. The server's public key will be added our list of known hosts and will now be trusted. The server's public key will be stored in `~/.ssh/known_hosts`. You may verify this by using **cat ~/.ssh/known\_hosts**.
6. Type **hostname** to see that you are truly logged into the RHELv9 server. Then type **exit** to close the connection.
7. Try to connect again using **ssh 10.0.0.1**. This time the server will identify us by our password, but we authenticate the server with its public key from step 5.
8. The SSH client configuration file is `/etc/ssh/ssh_config`. The server configuration file is `/etc/ssh/sshd_config`. Edit the server configuration file to allow `X11Forwarding`. This activates SSH's tunneling features which will allow us to run X programs (GUI) on the server from our client VM. Remove the # sign at the beginning of the line to uncomment it.
9. We authenticate the server using its public key. The server may authenticate the client using the client's public key, but first we need to create one. Create both an RSA and DSA pair of keys (private/public): **ssh-keygen -t rsa** and **ssh-keygen -t dsa**.
10. Verify the client keys have been created: **ls ~/.ssh**

11. Check the permissions of the keys using **ls -l ~/.ssh**. Note that only the owner may read/write to the private keys, everyone is able to read the public keys.
12. Now let's send the client's public key (Ubuntu) to the server:  
**sudo ssh-copy-id -i ~/.ssh/id\_rsa.pub [student@10.0.0.1](#)**
13. Display the client's RSA public key: **cat ~/.ssh/id\_rsa.pub**
14. Log into the server: **ssh 10.0.0.1** (you'll need to type the passphrase if you used one the first time)
15. Now that you are logged into the server, display the RSA public key from Ubuntu and compare to the one displayed in step 13: **cat ~/.ssh/authorized\_keys**
16. Now let's try to run a GUI program on the server while seeing it on our client. Do the following:
  - a. Ubuntu client: enable remote access by adding **enable=yes** under the **[xdmcp]** section of **/etc/gdm3/custom.conf**
  - b. Ubuntu: Tell Ubuntu to accept for display in its X server data that originates on the RHELv9 server:  
**xhost +10.0.0.1**
  - c. Ubuntu: Determine the name of your display: **xdpyinfo** followed by **echo \$DISPLAY**
  - d. Ubuntu: **ssh student@10.0.0.1 -X**
  - e. RHELv9: **export DISPLAY=Ubuntu:0** (use the name of your display from step c)
  - f. RHELv9: check to make sure the edit in step 8 was completed within the RHELv9 server's **sshd\_config** file
  - g. Ubuntu: Type **hostname** to verify you are logged into the RHELv9 server from the Ubuntu client machine.
  - h. Run **/usr/bin/baobab** which is the *Disk Usage Analyzer* to see a GUI program on your Ubuntu display that is really running on the RHELv9 server.
  - i. When finished, type **exit** to return to your Ubuntu local display.
  - j. Run **xhost -10.0.0.1** to remove from the access control list.
17. Submit your command history for grading.



# Lab 16: Logs and NTP

---

1. Display the last ten lines add to the messages log file.
2. View journal data.
3. Display log entries for the past week.
4. Display log entries for the current system boot.
5. Display an overview of current time settings.
6. Set the timezone to the time in California, USA.
7. Reset the timexone to the time in Missouri, USA.