

RH415 - Red Hat Security

Linux in Physical, Virtual, and Cloud

Travis Michette

Version 1.0

Table of Contents

Introduction 1

 Environment Overview and Launching an Instance 1

 Accessing the System Externally 3

 Accessing FoundationX VMs from Foundation0 3

1. Gaining Remote Access with Ansible 4

 1.1. Setting Up Ansible on Foundation0 4

 1.1.1. Getting Files and Preparing Foundation0 Kiosk User 5

 1.1.2. Installing Ansible on Foundation 6

 1.2. Configuring Foundation0 8

 1.3. Configuring WorkstationX 8

2. Gaining Remote Access with BASH 11

 2.1. Configuring Foundation0 11

 2.1.1. Configuring **Foundation0** and **WorkstationX** Systems Using Bash. 11

Introduction

VMs running on FoundationX share an external 172.25.250.0/24 network, with a gateway of 172.25.250.254 (workstation.lab.example.com). DNS services for the private network are provided by 172.25.250.254 (workstation), so the Workstation VM must be started first.

Environment Overview and Launching an Instance

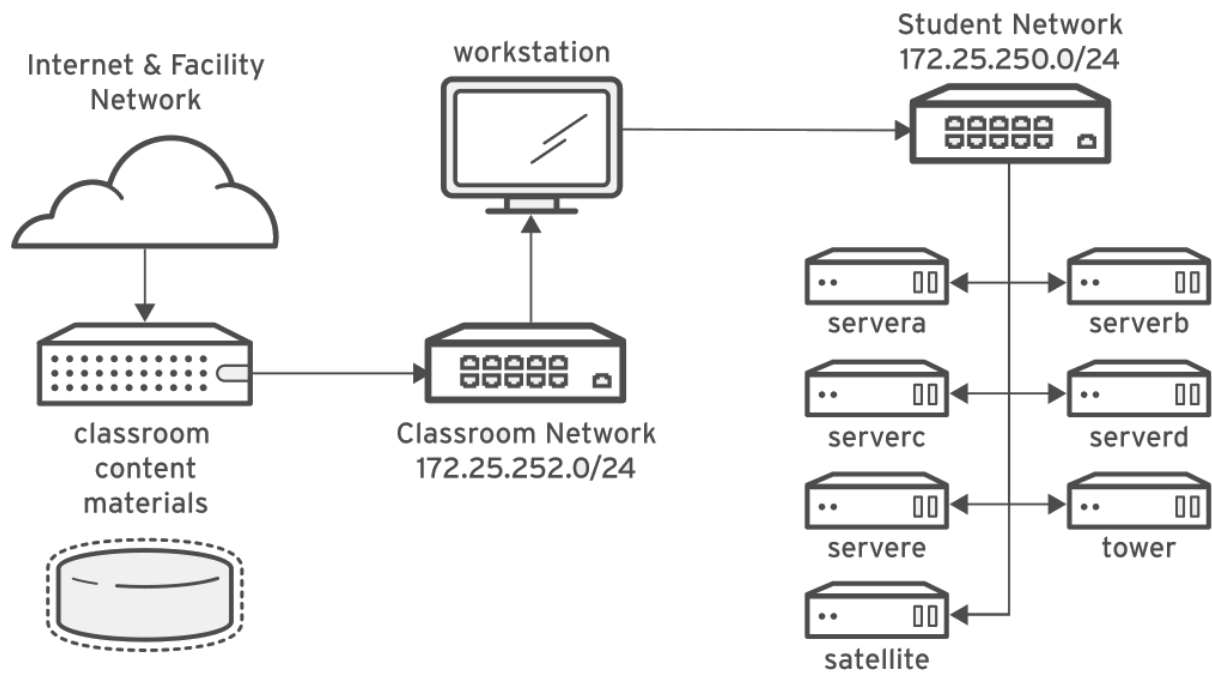


Figure 1. Classroom Environment Layout

There are eight systems used to comprise the entire classroom environment (in addition to **Workstation**). The listing of machines are:

- servera
- serverb
- serverc
- serverd
- servere
- satellite
- tower

Table 1. Security Classroom Layout and Information

Machine Name	IP Address	Role
servera.lab.example.com	172.25.250.10	Managed Server "A"
serverb.lab.example.com	172.25.250.11	Managed Server "B"
serverc.lab.example.com	172.25.250.12	Managed Server "C"
serverd.lab.example.com	172.25.250.13	Managed Server "D"
servere.lab.example.com	172.25.250.16	Managed Server "E"
satellite.lab.example.net	172.25.250.15	Red Hat Satellite 6 Server
tower.lab.example.net	172.25.250.14	Red Hat Ansible Tower server
workstation.lab.example.com / workstation0.example.com	172.25.250.254 / 172.25.252.250	Graphical Workstation as Student Desktop
classroom.example.com	172.25.254.254 / 172.25.252.254 / 172.25.253.254	Classroom utility server
foundation0.ilt.example.com / foundationX.ilt.example.com	172.25.254.250 / 172.25.253.250 / 172.25.254.X	Physical System



The **classroom** server acts as a NAT router for the classroom network. It provides DNS, DHCP, HTTP, and other services. It is also known by **content.example.com** and **materials.example.com**.

Classroom Credentials

System(s)	Username	Password
Student Systems (regular user)	student	student
Student Systems (Root user)	root	redhat
Satellite	admin	redhat
Ansible Tower	admin	redhat



The setup scripts are meant to catch up labs between chapters. It should be noted that labs are meant to be successive for this course.



The Workstation VM must be the first machine powered on. After workstation is up, the Satellite machine should be powered on before any of the other machines. After Workstation and Satellite have both been powered on and running, it is safe to start all other VMs with **rht-vmctl start all** command.



Grading/Setup scripts located <http://content/courses/rh415/rhel7.5/grading-scripts/>. The Ansible playbooks are located at <http://content/courses/rh415/rhel7.5/infrastructure/>. Overall classroom files are <http://content/courses/rh415/rhel7.5/>.

Accessing the System Externally

If using a Macbook or another system on the classroom network, it will be assigned an IP address. The way to access workstation is with the **172.25.252.X** IP address. Once on workstation, you can get to other systems. The other method is to access **FoundationX** directly, which can be done with the **172.25.254.X** IP address.



For a Mac/Linux system, you can use "sudo route -n add/delete 172.25.0.0/16" with a gateway of **172.25.254.254** to route traffic across multiple interfaces.



The **Foundation0** system IP address **172.25.254.250** is the instructor system.



Grading scripts get downloaded locally to **/usr/local/lib** and several executables for the environment also live in **/usr/local/bin/lab**



To preserve system resources, the **Satellite** and **Tower** VMs can be turned except when they are needed to be used in Chapter 8.

Accessing FoundationX VMs from Foundation0

With the exception of **WorkstationX**, VMs running on **FoundationX** cannot be accessed directly from **Foundation0**. In order to make items more efficient, a set of Ansible playbooks and resource files has been created to allow accessing VMs running on **FoundationX** remotely from Foundation0 by setting up port forwarding on **WorkstationX**.

Table 2. Accessing Systems Remotely

Machine Name	IP Port
servera	122
serverb	222
serverc	322
serverd	422
servere	522
satellite	722
tower	622

1. Gaining Remote Access with Ansible

Objectives

- Install Ansible
- Enable Remote Access

In order to access the systems remotely and gain remote access, it is necessary to setup and configure **Firewalld** to perform port forwarding, masquerading, and NAT from the **WorkstationX** machine.

A set of Ansible playbooks and resource files has been created to configure the **Foundation0** system with the Kiosk user remote access to all VMs hosted on **FoundationX** systems. These playbooks will also configure **WorkstationX** systems with port forwarding to the various **FoundationX** hosted VMs.

The Ansible playbook **DeploySSH.yml** will copy the **config** file as the **/home/kiosk/.ssh/config** file to setup the names so that it is possible to SSH to a system using **server[a-e]X**, where **X** is the number corresponding to the foundation that the system is running on. The playbook also copies the **hosts** file to **/etc/hosts** which sets up the names of the VMs running on FoundationX so that they are resolvable via DNS to work with the SSH **config** file.

The Ansible playbook **WorkstationFW.yml** will copy the **/Ansible/resource_files/workstation_external.xml** to **/etc/firewalld/zones/external.xml** file and reload the FirewallD rules. This will allow the port forwarding to be setup.

After running both playbooks, it will be possible to SSH directly to a system. The **inventory** file is setup so that **Workstation** will cover all workstationX systems and **Foundation** will cover all foundationX systems. In order to use Ansible with Foundation0, it is configured with a group called **Instructor**.

1.1. Setting Up Ansible on Foundation0

Ansible is already available from **Foundation0** and can be installed by the root user. The **RH415_Ansible.tgz** file contains the following files/Directories:

RH415_Ansible.tgz

- **/Ansible**
- **/Ansible/ansible.cfg**
- **/Ansible/inventory**
- **/Ansible/DeploySSH.yml**
- **/Ansible/WorkstationFW.yml**
- **/Ansible/resource_files**
- **/Ansible/resource_files/config**
- **/Ansible/resource_files/hosts**
- **/Ansible/resource_files/workstation_external.xml**

The **tar** file can be downloaded and extracted as the **kiosk** user and it will automatically created the **Ansible/** directory with the correct hierarchy. The **ansible.cfg** and the **inventory** file has been setup and configured so that it will run everything as root

based on the **Foundation0 RHT** training key being distributed to all systems.

1.1.1. Getting Files and Preparing Foundation0 Kiosk User

Currently, the files can be obtained from <http://people.redhat.com/~tmichett/RH415>. There is the PDF version of this guide as well as the **RH415_Ansible.tgz** file containing all files needed for Ansible. Eventually, the files will be moved to an Instructor area in RHLC, a location in MOJO, or hopefully our own GitHub/Gitlab location for projects.

1. Place the **RH415_Ansible.tgz** in the Kiosk Home Directory

Listing 1. Verifying File is in Correct Directory

```
[kiosk@foundation0 ~]$ pwd
/home/kiosk
[kiosk@foundation0 ~]$ ls *.tgz
RH415_Ansible.tgz
[kiosk@foundation0 ~]$
```

2. Extract the File(s)/Folder(s) to Kiosk Home Directory

Listing 2. Extraction and Verification of Files

```
[kiosk@foundation0 ~]$ tar -xvf RH415_Ansible.tgz
Ansible/
Ansible/resource_files/
Ansible/resource_files/hosts
Ansible/resource_files/config
Ansible/resource_files/workstation_external.xml
Ansible/DeploySSH.yml
Ansible/WorkstationFW.yml
Ansible/inventory
Ansible/ansible.cfg
Ansible/Scripts/
Ansible/Scripts/SetupRemoteAccess.sh
[kiosk@foundation0 ~]$

[root@foundation0 kiosk]# ls -alRF Ansible/
Ansible/:
total 20
drwxrwxr-x. 4 kiosk kiosk 125 Aug 10 03:19 ./
drwx----- 19 kiosk kiosk 4096 Aug 10 15:02 ../
-rw-r--r--. 1 kiosk kiosk 186 Aug 6 10:54 ansible.cfg
-rw-r--r--. 1 kiosk kiosk 259 Aug 8 09:28 DeploySSH.yml
-rw-r--r--. 1 kiosk kiosk 497 Aug 7 04:13 inventory
drwxrwxr-x. 2 kiosk kiosk 133 Aug 8 08:39 resource_files/
drwxrwxr-x. 2 kiosk kiosk 34 Aug 10 15:03 Scripts/
-rw-r--r--. 1 kiosk kiosk 265 Aug 8 09:27 WorkstationFW.yml

Ansible/resource_files:
total 20
drwxrwxr-x. 2 kiosk kiosk 133 Aug 8 08:39 ./
drwxrwxr-x. 4 kiosk kiosk 125 Aug 10 03:19 ../
-rw-r--r--. 1 kiosk kiosk 1225 Aug 8 08:39 config
-rw-r--r--. 1 kiosk kiosk 2102 Aug 8 04:11 hosts
-rw-r--r--. 1 kiosk kiosk 886 Aug 6 12:38 workstation16_external.xml
-rw-r--r--. 1 kiosk kiosk 886 Aug 6 12:38 workstation17_external.xml
-rw-r--r--. 1 kiosk kiosk 872 Aug 8 04:02 workstation_external.xml

Ansible/Scripts:
total 4
drwxrwxr-x. 2 kiosk kiosk 34 Aug 10 15:03 ./
drwxrwxr-x. 4 kiosk kiosk 125 Aug 10 03:19 ../
-rwxrwxr-x. 1 kiosk kiosk 799 Aug 10 15:03 SetupRemoteAccess.sh*
```

1.1.2. Installing Ansible on Foundation

In order to use Ansible on **Foundation0** it must first be installed. As the Kiosk user is not configured to perform **sudo** it is necessary to **su -** to get to root in order to perform the installation.

1. Switch to the **root** user using "**su -**".

Listing 3. Becoming the Root User

```
[kiosk@foundation0 ~]$ su -
Password:
Last login: Thu Nov 8 13:50:04 EST 2018 on pts/1
[root@foundation0 ~]#
```


2. Perform a "yum install ansible" to Install Ansible Packages

Listing 4. Install Ansible

```
[root@foundation0 ~]# yum install ansible

... output omitted ...

=====
Installing:
ansible                noarch      2.8.0-1.el7ae      ucf-upd      15 M
Installing for dependencies:
python-babel           noarch      0.9.6-8.el7        rhel-dvd     1.4 M
python-cffi            x86_64     1.6.0-5.el7        rhel-dvd     218 k
python-enum34          noarch      1.0.4-1.el7        rhel-dvd     52 k
python-idna            noarch      2.4-1.el7          rhel-dvd     94 k
python-jinja2          noarch      2.7.2-2.el7        rhel-dvd     516 k
python-markupsafe      x86_64     0.11-10.el7        rhel-dvd     25 k
python-paramiko        noarch      2.1.1-5.el7        rhel-dvd     268 k
python-passlib         noarch      1.6.5-1.1.el7      ucf-upd     488 k
python-pycparser       noarch      2.14-1.el7         rhel-dvd     105 k
python2-cryptography  x86_64     1.7.2-2.el7        rhel-dvd     503 k
python2-jmespath       noarch      0.9.0-4.el7ae      ucf-upd     39 k
python2-pyasn1         noarch      0.1.9-7.el7        rhel-dvd     100 k

Transaction Summary
=====
Install 1 Package (+12 Dependent packages)

Total download size: 18 M
Installed size: 101 M
Is this ok [y/d/N]: y

... output omitted ...

Installed:
  ansible.noarch 0:2.8.0-1.el7ae

Dependency Installed:
  python-babel.noarch 0:0.9.6-8.el7
  python-cffi.x86_64 0:1.6.0-5.el7
  python-enum34.noarch 0:1.0.4-1.el7
  python-idna.noarch 0:2.4-1.el7
  python-jinja2.noarch 0:2.7.2-2.el7
  python-markupsafe.x86_64 0:0.11-10.el7
  python-paramiko.noarch 0:2.1.1-5.el7
  python-passlib.noarch 0:1.6.5-1.1.el7
  python-pycparser.noarch 0:2.14-1.el7
  python2-cryptography.x86_64 0:1.7.2-2.el7
  python2-jmespath.noarch 0:0.9.0-4.el7ae
  python2-pyasn1.noarch 0:0.1.9-7.el7

Complete!
[root@foundation0 ~]#
```



This may not work depending on your specific configuration as **Ansible** may not be available. This has been tested with the latest Foundation 7.6 release.

1.2. Configuring Foundation0

After Ansible has been installed, it is time to configure Foundation0 by running the Ansible playbook **DeploySSH.yml** to configure the name resolution as well as the SSH configuration file for the names and the ports that will be used to connect to the servers.

1. As the **KIOSK** user, change directories to the **/home/kiosk/Ansible** directory.

Listing 5. Access the Ansible Directory

```
[kiosk@foundation0 ~]$ cd Ansible/
[kiosk@foundation0 Ansible]$ pwd
/home/kiosk/Ansible
[kiosk@foundation0 Ansible]$
```

2. As the **KIOSK** user, use the **ansible-playbook** command to execute the **DeploySSH.yml** playbook.

Listing 6. Run the DeploySSH.yml Playbook

```
[kiosk@foundation0 Ansible]$ ansible-playbook DeploySSH.yml

PLAY [foundation0] *****

TASK [Gathering Facts] *****
ok: [foundation0]

TASK [Copy SSH Config File] *****
changed: [foundation0]

TASK [Hosts File] *****
changed: [foundation0]

PLAY RECAP *****
foundation0      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[kiosk@foundation0 Ansible]$
```



The above steps configure the **Foundation0** system with hostnames that are good for a classroom size of 17 systems. The Ansible inventory file is also configured for Foundation1 through Foundation17. It might be necessary to modify the **inventory** file to the correct systems in order to avoid error messages.

1.3. Configuring WorkstationX

Once the **Foundation0** system has been configured with Ansible and the networking/SSH has been setup, it is necessary to configure the **WorkstationX** systems to allow remote connectivity and port forwarding for the systems residing on **FoundationX**. The **WorkstationFW.yml** needs to be run in order to configure and load FirewallD rules on the **WorkstationX** systems.

1. As the **KIOSK** user, change directories to the **/home/kiosk/Ansible** directory.

Listing 7. Access the Ansible Directory

```
[kiosk@foundation0 ~]$ cd Ansible/
[kiosk@foundation0 Ansible]$ pwd
/home/kiosk/Ansible
[kiosk@foundation0 Ansible]$
```

- As the **KIOSK** user, use the **ansible-playbook** command to execute the **WorkstationFW.yml** playbook.

Listing 8. Run the WorkstationFW.yml Playbook

```
[kiosk@foundation0 Ansible]$ ansible-playbook WorkstationFW.yml

PLAY [Workstation]
*****

TASK [Gathering Facts]
*****
fatal: [workstation2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation2 port 22: No route to host", "unreachable": true}
fatal: [workstation3]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation3 port 22: No route to host", "unreachable": true}
ok: [workstation1] ①
fatal: [workstation6]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation6 port 22: No route to host", "unreachable": true}
fatal: [workstation7]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation7 port 22: No route to host", "unreachable": true}
fatal: [workstation9]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation9 port 22: No route to host", "unreachable": true}
fatal: [workstation4]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation4 port 22: Connection timed out", "unreachable": true}
fatal: [workstation5]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation5 port 22: Connection timed out", "unreachable": true}
fatal: [workstation11]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation11 port 22: No route to host", "unreachable": true}
fatal: [workstation10]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation10 port 22: No route to host", "unreachable": true}
fatal: [workstation12]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation12 port 22: No route to host", "unreachable": true}
fatal: [workstation8]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation8 port 22: Connection timed out", "unreachable": true}
fatal: [workstation14]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation14 port 22: No route to host", "unreachable": true}
fatal: [workstation13]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation13 port 22: No route to host", "unreachable": true}
fatal: [workstation16]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation16 port 22: No route to host", "unreachable": true}
fatal: [workstation17]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation17 port 22: No route to host", "unreachable": true}
fatal: [workstation15]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host workstation15 port 22: Connection timed out", "unreachable": true}

TASK [Firewall Config File]
*****
changed: [workstation1]

TASK [Reload FirewallD]
*****
changed: [workstation1]

PLAY RECAP
*****
```

```

workstation1      : ok=3  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0  ②
workstation10     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation11     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation12     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation13     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation14     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation15     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation16     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation17     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation2      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation3      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation4      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation5      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation6      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation7      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation8      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
workstation9      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0

[kiosk@foundation0 Ansible]$

```

- ① **workstation1** system found and Ansible facts were gathered.
- ② **workstation1** system had changes applied and tasks executed successfully.



It is fine to leave the default **inventory** file. However, if the environment has less than 17 Foundation systems, it will error out contacting the workstation systems. The example above had only a single FoundationX system which meant there is only a **Workstation1** system and the other sixteen (16) systems were not available. It is important to look at the Ansible output for the systems that are present to ensure that the changes to the system were made.

2. Gaining Remote Access with BASH

Objectives

- Setup SSH **config** and **/etc/hosts** on **Foundation0**
- Copy **workstation_external.xml** to **/etc/firewalld/zones/external.xml** on **WorkstationX**
- Reload FirewallD service on **WorkstationX**

In order to access the systems remotely and gain remote access, it is necessary to setup and configure **Firewalld** to perform port forwarding, masquerading, and NAT from the **WorkstationX** machine.

2.1. Configuring Foundation0

In the previous chapter, the **RH415_Ansible.tgz** was downloaded and extracted. This contains the files needed for both **Foundation0** and **WorkstationX** systems. In order to **Foundation0** to properly access the remote VMs, both **/etc/hosts** and **/home/kiosk/.ssh/config** files need to be updated. The files that will be used on **Foundation0** are located in the **/home/kiosk/Ansible/resource_files** directory. There is also a directory **/home/kiosk/Ansible/Scripts** that contains a shell script that will complete the copying of the configuration files on **Foundation0** as well as copy the file to **WorkstationX** and restart the **Firewalld** service. The **SetupRemoteAccess.sh** must be made executable and must be run as the **root** user.

2.1.1. Configuring Foundation0 and WorkstationX Systems Using Bash

For convenience a BASH script has been created to perform all necessary configuration changes to both the **Foundation0** system and the **WorkstationX** systems. This script must be run as the **root** user.



If the system has Ansible and the items in Chapter 1 were completed, there is no need to run the BASH scripts to configure the system as Ansible accomplished that task. This chapter is meant for older classroom environments and Foundation versions where Ansible is not able to be used.



The BASH script in this section **must** be executed as the **root** user so that the **/etc/hosts** file can be updated on **Foundation0**.

1. Become the Root User

Listing 9. Becoming Root User

```
[kiosk@foundation0 ~]$ su -  
Password:  
Last login: Sat Aug 10 15:02:55 EDT 2019 on pts/1  
[root@foundation0 ~]#
```

2. Make **SetupRemoteAccess.sh** Executable

Listing 10. Becoming Root User

```
[root@foundation0 ~]# chmod +x /home/kiosk/Ansible/Scripts/SetupRemoteAccess.sh
```

3. Execute the `/home/kiosk/Ansible/Scripts/SetupRemoteAccess.sh` Script

Listing 11. Executing SetupRemoteAccess.sh

```
[root@foundation0 ~]# /home/kiosk/Ansible/Scripts/SetupRemoteAccess.sh
=====
== This Script Must be Run as Root ==
=====
Please enter the number of foundation systems for this course:
1 ①
Preparing to Setup WorkstationX Systems for the Proper Firewall Port Forwarding
workstation1
workstation_external.xml
100% 872 457.4KB/s 00:00
success
Preparing Foundation0 System
Copying the SSH config File
Copying the /etc/hosts File
If you received: cp: cannot create regular file '/home/etc/hosts': No such file or directory, then you forgot to run the script as ROOT
[root@foundation0 ~]#
```

- ① Script takes input from end-user regarding the number of connected systems.