

# Exercise 8 - Remote Software Install - Script 7

## **Goal:**

The goal of this exercise is to create a shell script that installs apache on multiple servers.

## **Scenario:**

Your company is growing in popularity with website visitor growth exploding. Every week you find yourself installing Apache on a system, so you decide to write a script to do this for you.

## **Shell Script Requirements:**

You think about what the shell script must do and how you would like it operate. You come up with the following list.

The script:

- Is named "install-apache-on.sh".
- Requires that at least one host is provided as an argument. If a host is not provided, then the script will display an error message and exit with a status of 1.
- Loops through each host provided as an argument and installs apache as well as enabling the web server process to start on boot.
- Creates an index.html file in the web server DocumentRoot that contains the server's hostname.
- Tests to see that the web server responds. If it does not, display an error message and ensure the script exits with a non-zero exit status.
- Pings a host and if a response is not received, assumes the host is down and continues to the next host. If a host is down, display an error message ensure script exits with a non-zero exit status.
- Enforces that it be executed without superuser (root) privileges.
- Provides a usage statement much like you would find in a man page if the user does not supply a command to run on the command line and returns an exit status of 1. All messages associated with this event will be displayed on standard error.
- Exits with an exit status of 0 if none of the error conditions above were encountered.

## **Start the Virtual Machines and Log into admin01**

In a previous exercise you created a vagrant project called multinet. Use the VMs created in that project for this exercise.

First, start a command line session on your local machine. Next, move into the working folder you created for this course.

```
cd shellclass
```

Change into the multinet directory, start the virtual machines with "vagrant up", and then connect to the admon01 VM with "vagrant ssh admin01".

```
vagrant up  
vagrant ssh admin01
```

## Navigate to the /vagrant Directory

```
cd /vagrant
```

## Write the Shell Script

At this point, you can either create the script inside the virtual machine using the vim, nano, or emacs text editors or you can create the file using your favorite text editor on your local operating system. (Atom from <https://atom.io/> is a good choice.)

When creating your script, refer back to the [shell script requirements](#). If you want or need more detailed steps to help you write your script, refer to the [pseudocode](#) at the end of this document. It was intentionally placed at the end of the document because I want to encourage you to write the script on your own. It's fine if you need the pseudocode. As you get more scripting practice, you'll be able to script without any additional aids.

## Test Your Script

Once you've finished writing the script, test it by:

- Executing it with super user privileges.
- Executing it without any options or arguments.
- Executing it and providing two hosts as command line arguments.
- Taking a server off the network and then executing the script.

Remember that the first time you execute the script you'll need to make sure it has executable permissions.

```
chmod 755 install-apache-on.sh
```

Here is an example run of the script. (Portions typed are in bold.)

```
sudo ./install-apache-on.sh  
Do not execute this script as root.  
Usage: ./install-apache-on.sh HOST [HOSTN...]  
echo ${?}  
1
```

Make sure the script displays a usage message if we don't supply a host to install apache on.

```
./install-apache-on.sh  
Usage: ./install-apache-on.sh HOST [HOSTN...]  
echo ${?}  
1
```

Make sure the script installs Apache on all the hosts listed on the command line.

```
./install-apache-on.sh server01 server02  
Starting installation process on: server01  
Loaded plugins: fastestmirror  
Repodata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast  
Determining fastest mirrors  
* base: repos.lax.quadranet.com  
* extras: centos.host-engine.com  
* updates: centos.mbni.med.umich.edu  
Resolving Dependencies  
--> Running transaction check  
---> Package httpd.x86_64 0:2.4.6-67.el7.centos.6 will be installed  
--> Processing Dependency: httpd-tools = 2.4.6-67.el7.centos.6 for package:  
httpd-2.4.6-67.el7.centos.6.x86_64  
--> Processing Dependency: /etc/mime.types for package:  
httpd-2.4.6-67.el7.centos.6.x86_64  
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package:  
httpd-2.4.6-67.el7.centos.6.x86_64  
--> Processing Dependency: libapr-1.so.0()(64bit) for package:  
httpd-2.4.6-67.el7.centos.6.x86_64  
--> Running transaction check  
---> Package apr.x86_64 0:1.4.8-3.el7_4.1 will be installed  
---> Package apr-util.x86_64 0:1.5.2-6.el7 will be installed  
---> Package httpd-tools.x86_64 0:2.4.6-67.el7.centos.6 will be installed  
---> Package mailcap.noarch 0:2.1.41-2.el7 will be installed  
--> Finished Dependency Resolution  
  
Dependencies Resolved
```

```

=====
Package           Arch           Version           Repository        Size
=====
Installing:
httpd             x86_64        2.4.6-67.el7.centos.6   updates          2.7 M
Installing for dependencies:
apr              x86_64        1.4.8-3.el7_4.1       updates          103 k
apr-util         x86_64        1.5.2-6.el7           base              92 k
httpd-tools     x86_64        2.4.6-67.el7.centos.6   updates          88 k
mailcap         noarch        2.1.41-2.el7           base              31 k

```

#### Transaction Summary

```

=====
Install 1 Package (+4 Dependent packages)

```

```

Total download size: 3.0 M
Installed size: 10 M
Downloading packages:

```

```

-----
Total                               842 kB/s | 3.0 MB  00:03

```

```

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction

```

```

Installing : apr-1.4.8-3.el7_4.1.x86_64          1/5
Installing : apr-util-1.5.2-6.el7.x86_64        2/5
Installing : httpd-tools-2.4.6-67.el7.centos.6.x86_64 3/5
Installing : mailcap-2.1.41-2.el7.noarch         4/5
Installing : httpd-2.4.6-67.el7.centos.6.x86_64 5/5
Verifying  : mailcap-2.1.41-2.el7.noarch         1/5
Verifying  : httpd-2.4.6-67.el7.centos.6.x86_64 2/5
Verifying  : apr-util-1.5.2-6.el7.x86_64        3/5
Verifying  : apr-1.4.8-3.el7_4.1.x86_64        4/5
Verifying  : httpd-tools-2.4.6-67.el7.centos.6.x86_64 5/5

```

```

Installed:
httpd.x86_64 0:2.4.6-67.el7.centos.6

```

```

Dependency Installed:
apr.x86_64 0:1.4.8-3.el7_4.1          apr-util.x86_64 0:1.5.2-6.el7
httpd-tools.x86_64 0:2.4.6-67.el7.centos.6 mailcap.noarch 0:2.1.41-2.el7

```

Complete!

```

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service
to /usr/lib/systemd/system/httpd.service.

```

server01

```

Finished installation process on: server01

```

```

Starting installation process on: server02

```

```

Loaded plugins: fastestmirror

```

```

Repodata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast

```

```

Determining fastest mirrors

```

```

* base: repos.lax.quadranet.com
* extras: centos.host-engine.com
* updates: centos.mbni.med.umich.edu
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.6-67.el7.centos.6 will be installed
--> Processing Dependency: httpd-tools = 2.4.6-67.el7.centos.6 for package:
httpd-2.4.6-67.el7.centos.6.x86_64
--> Processing Dependency: /etc/mime.types for package:
httpd-2.4.6-67.el7.centos.6.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package:
httpd-2.4.6-67.el7.centos.6.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package:
httpd-2.4.6-67.el7.centos.6.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.4.8-3.el7_4.1 will be installed
---> Package apr-util.x86_64 0:1.5.2-6.el7 will be installed
---> Package httpd-tools.x86_64 0:2.4.6-67.el7.centos.6 will be installed
---> Package mailcap.noarch 0:2.1.41-2.el7 will be installed
--> Finished Dependency Resolution

```

#### Dependencies Resolved

```

=====
Package                Arch          Version                               Repository      Size
=====
Installing:
httpd                   x86_64        2.4.6-67.el7.centos.6                updates         2.7 M
Installing for dependencies:
apr                     x86_64        1.4.8-3.el7_4.1                       updates         103 k
apr-util                x86_64        1.5.2-6.el7                            base            92 k
httpd-tools             x86_64        2.4.6-67.el7.centos.6                updates         88 k
mailcap                 noarch        2.1.41-2.el7                          base            31 k

```

#### Transaction Summary

```

=====
Install 1 Package (+4 Dependent packages)

```

```
Total download size: 3.0 M
```

```
Installed size: 10 M
```

```
Downloading packages:
```

```
-----
Total                                     842 kB/s | 3.0 MB  00:03
```

```
Running transaction check
```

```
Running transaction test
```

```
Transaction test succeeded
```

```
Running transaction
```

```

Installing : apr-1.4.8-3.el7_4.1.x86_64                1/5
Installing : apr-util-1.5.2-6.el7.x86_64                2/5
Installing : httpd-tools-2.4.6-67.el7.centos.6.x86_64   3/5
Installing : mailcap-2.1.41-2.el7.noarch                 4/5
Installing : httpd-2.4.6-67.el7.centos.6.x86_64         5/5

```

```
Verifying : mailcap-2.1.41-2.el7.noarch 1/5
Verifying : httpd-2.4.6-67.el7.centos.6.x86_64 2/5
Verifying : apr-util-1.5.2-6.el7.x86_64 3/5
Verifying : apr-1.4.8-3.el7_4.1.x86_64 4/5
Verifying : httpd-tools-2.4.6-67.el7.centos.6.x86_64 5/5
```

Installed:

```
httpd.x86_64 0:2.4.6-67.el7.centos.6
```

Dependency Installed:

```
apr.x86_64 0:1.4.8-3.el7_4.1 apr-util.x86_64 0:1.5.2-6.el7
httpd-tools.x86_64 0:2.4.6-67.el7.centos.6 mailcap.noarch 0:2.1.41-2.el7
```

Complete!

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

server02

Finished installation process on: server02

**echo \${?}**

0

**curl http://server01**

server01

**curl http://server02**

server02

Power down one of the servers and try to install apache on it. Make sure the exit status is non-zero.

**exit**

**vagrant halt server02**

==> server02: Attempting graceful shutdown of VM...

**vagrant ssh admin01**

Last login: Mon Jan 29 12:08:50 2018 from 10.0.2.2

**cd /vagrant**

**./install-apache-on.sh server02**

Starting installation process on: server02

server02 down.

**echo \$?**

1

## **Reference Material:**

### **Vagrantfile for multinet**

Here are the contents of the shellclass/multinet/Vagrantfile file with all the comments removed.

```
Vagrant.configure(2) do |config|
  config.vm.box = "jasonc/centos7"

  config.vm.define "admin01" do |admin01|
    admin01.vm.hostname = "admin01"
    admin01.vm.network "private_network", ip: "10.9.8.10"
  end

  config.vm.define "server01" do |server01|
    server01.vm.hostname = "server01"
    server01.vm.network "private_network", ip: "10.9.8.11"
  end

  config.vm.define "server02" do |server02|
    server02.vm.hostname = "server02"
    server02.vm.network "private_network", ip: "10.9.8.12"
  end

end
```

[This space intentionally left blank. Instructions continue on the following page.]

## Pseudocode

You can use the following pseudocode to help you with the logic and flow of your script.

```
# Display the usage and exit.
# Make sure the script is not being executed with superuser privileges.
# If the user doesn't supply at least one argument, give them help.
# Loop through the provided servers.
# Ping the server to make sure it's up.
# Install the httpd package
# Create an index.html file.
# Start httpd
# Enable httpd
# Test that the web server is accessible.
```