RED HAT®
CONSULTING

RH354 - RHEL 8 New Features for Experienced Linux Administrators

Travis Michette

Version 1.0

Table of Contents

Before You Begin

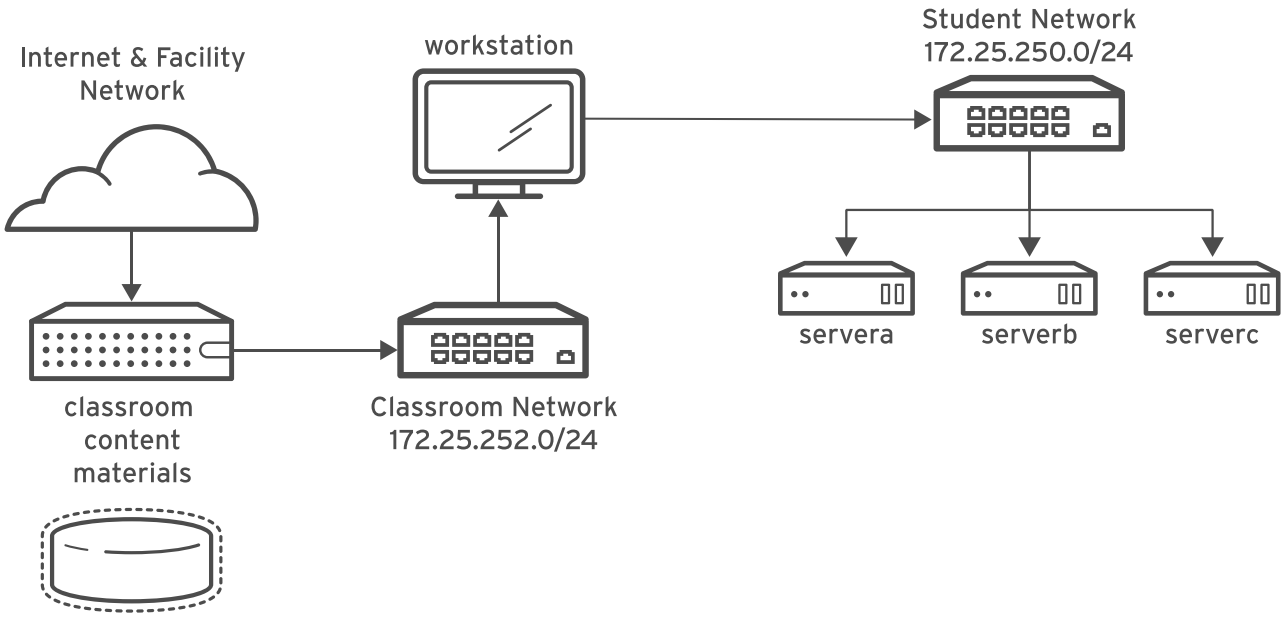

Figure 1. RH354 Classroom Layout

Table 1. Classroom Machines

Machine name	IP addresses	Role
workstation.lab.example.com	172.25.250.254	Gateway system to connect student private network to **classroom server** (must always be running). Graphical workstation used for system administration
servera.lab.example.com	172.25.250.10	First server
serverb.lab.example.com	172.25.250.11	Second server
serverc.lab.example.com	172.25.250.11	Third server (set as not booted)

1. Previewing Red Hat Enterprise Linux 8

1.1. Red Hat Enterprise Linux 8 Overview

New Features of Interest

- Restructure of channels **BaseOS** and **Appstream**
- Install and boot from NVDIMM devices
- **BOOM** Boot Manager
  - https://github.com/bmr-cymru/boom
  - https://www.redhat.com/en/blog/boom-booting-rhel-lvm-snapshots
  - https://www.youtube.com/watch?v=guM_jkA6Xfg
- Secure-boot Linux guests
- **Kernel Changes**
  - **kernel-core** provides core kernel module
  - **kernel-modules** and **kernel-modules-extra** contains kernel modules
  - **kernel** is now a *meta* which installs both **kernel-core** and **kernel-modules**
- **kdump** supported earlier in the boot process
- Process scheduler enhancements with new class **SCHED_DEADLINE**
- **Firewall changes**
  - **firewalld** still daemon with **firewall-cmd** to manage firewall
  - **iptables** removed in favor of **nft** directly
  - **iptables** compatibility tools available for interacting with **nft**
- **NetworkManager**
  - **nmcli** preferred way to manage the network
  - **ifup/ifdown** requires NetworkManager
- **NTP** time is now using **chronyd** as the default NTP implementation as **ntpd** is no longer available
- **yum** us now **YUM4** which is **dnf**
  - Modules are now supported
    - Modules allow installation of packages independent of the Operating System version
    - Modules replace the Red Hat Software Collections Suite of packages for RHEL8
    - Modules are tied to application stream
  - **dnf** is re-write of Yum but allows Yum4 to facilitate all commands using Yum
- **Storage Changes**

    - Stratis Storage Manager is introduced
        - https://stratis-storage.github.io/
    - **VDO** Virtual Data Optimizer is introduced
        - https://www.redhat.com/en/blog/look-vdo-new-linux-compression-layer
        - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/vdo
    - XFS Copy-on-Write is enabled by default when filesystem created
    - LUKS2 encryption is the new encryption mechanism replacing LUKS1
        - https://fedoraproject.org/wiki/Disk_Encryption_User_Guide
- **Security Features**
    - New **crypto** policies
    - New **sosreport** tool which can support profiles
    - Updated **audit** and **rsyslog** subsystems with new features
    - **tcp_wrapper** support removed
- **Graphical Environment**
    - Wayland is default display server **Xorg** still currently available
    - Gnome has been updated and **KDE** has been removed
- **Virtualization**
    - **e1000** network driver removed
    - KVM defaults to Q35 hardware model
    - **virt-manager** deprecated and **cockpit** preferred management solution
- **Linux Containers**
    - **podman** runs daemonless containers
    - **buildah** is used to build container images from scratch or a Dockerfile

Learning RHEL8 Online with Labs

Red Hat Learning Community (RHLC) RHEL8 Labs Post: https://learn.redhat.com/t5/Platform-Linux/Start-with-RHEL8-without-having-RHEL8/td-p/5813

Red Hat Labs: http://lab.redhat.com/

1.2. TMUX Usage

As part of the upgrade process and package replacement process in RHEL8, several packages have not only been deprecated, they've been completely removed. The **screen** package is one package that is no longer available for installation. Instead, a new terminal program **tmux** has been introduced to provide the **screen** functionality as well as other enhancements.

TMUX Usage

TMUX References

Red Hat Learning Community: https://learn.redhat.com/t5/Platform-Linux/Using-tmux-to-execute-commands-on-servers-in-parallel/m-p/2200

Tactical TMUX: https://danielmiessler.com/study/tmux/

In the example below, we will explore the **screen** functionalities of TMUX with respect to attaching and detaching of sessions.

Installing TMUX

1. Install **tmux** with YUM

Listing 1. Installation of TMUX

```
[root@workstation ~]# yum install tmux
Red Hat Enterprise Linux 8.0 AppStream (dvd)     21 MB/s | 5.3 MB     00:00
Red Hat Enterprise Linux 8.0 BaseOS (dvd)        23 MB/s | 2.2 MB     00:00
Last metadata expiration check: 0:00:01 ago on Mon 13 Apr 2020 10:55:47 AM EDT.
Package tmux-2.7-1.el8.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

2. Launch **tmux**

Listing 2. Using **tmux**

```
[student@workstation ~]$ tmux
```

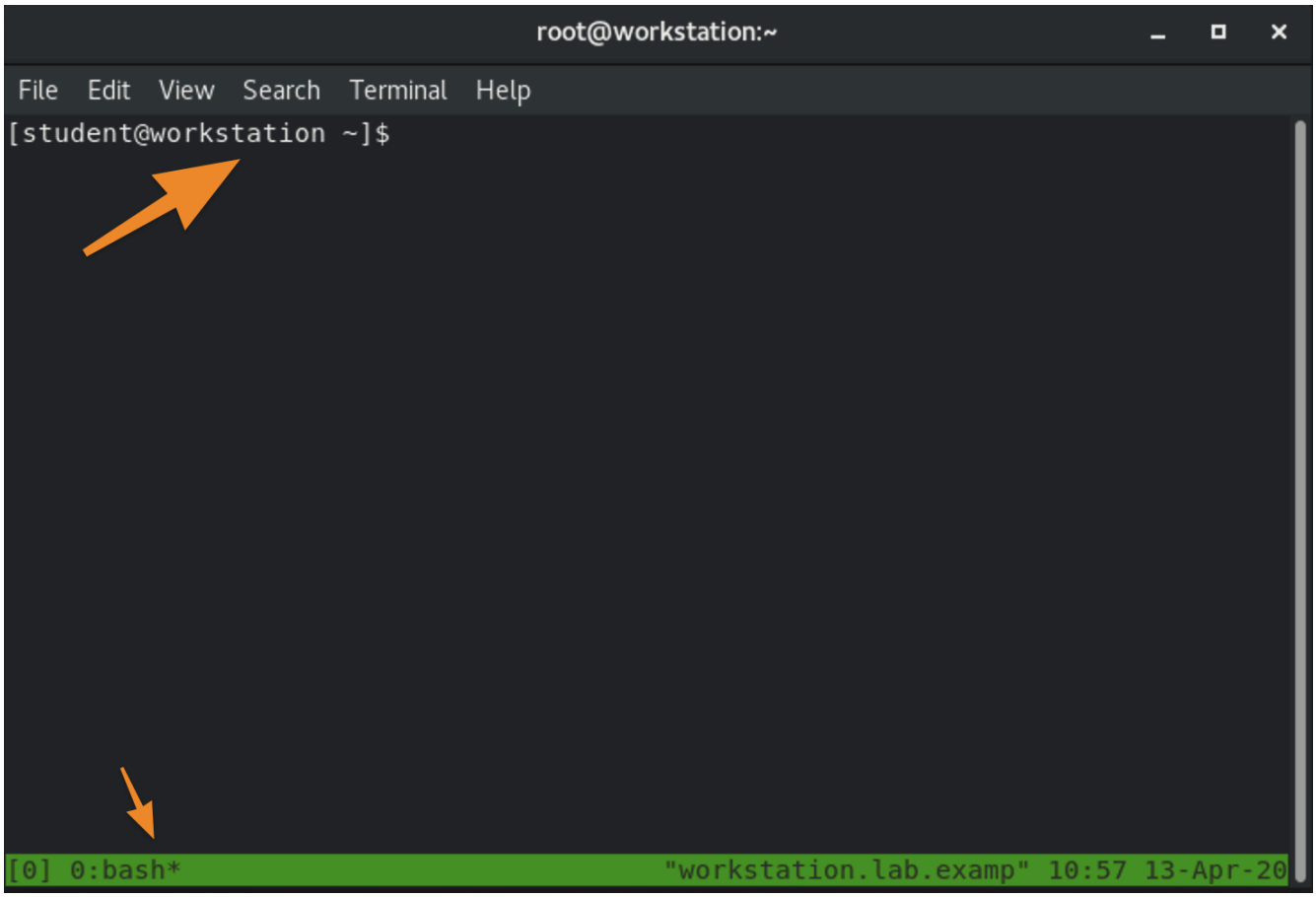

Figure 2. **tmux** *Terminal Window*

3. Placing **tmux** Window in Background (*CTRL+B+D*)

Listing 3. Detaching a **tmux** *Session*

```
[student@workstation ~]$ tmux
[detached (from session 0)]
```

4. Re-attaching to **tmux**

Listing 4. Source Description

```
[student@workstation ~]$ tmux list-sessions
0: 1 windows (created Mon Apr 13 11:01:53 2020) [80x23]

[student@workstation ~]$ tmux attach-session -t 0
```

Naming or Renaming Sessions

It is possible that once you are in a **tmux** session to rename the session.

Listing 5. Renaming a TMUX Session

```
[student@workstation ~]$ tmux rename-session --help
usage: rename-session [-t target-session] new-name

[student@workstation ~]$ tmux rename-session -t 0 travis-demo
```

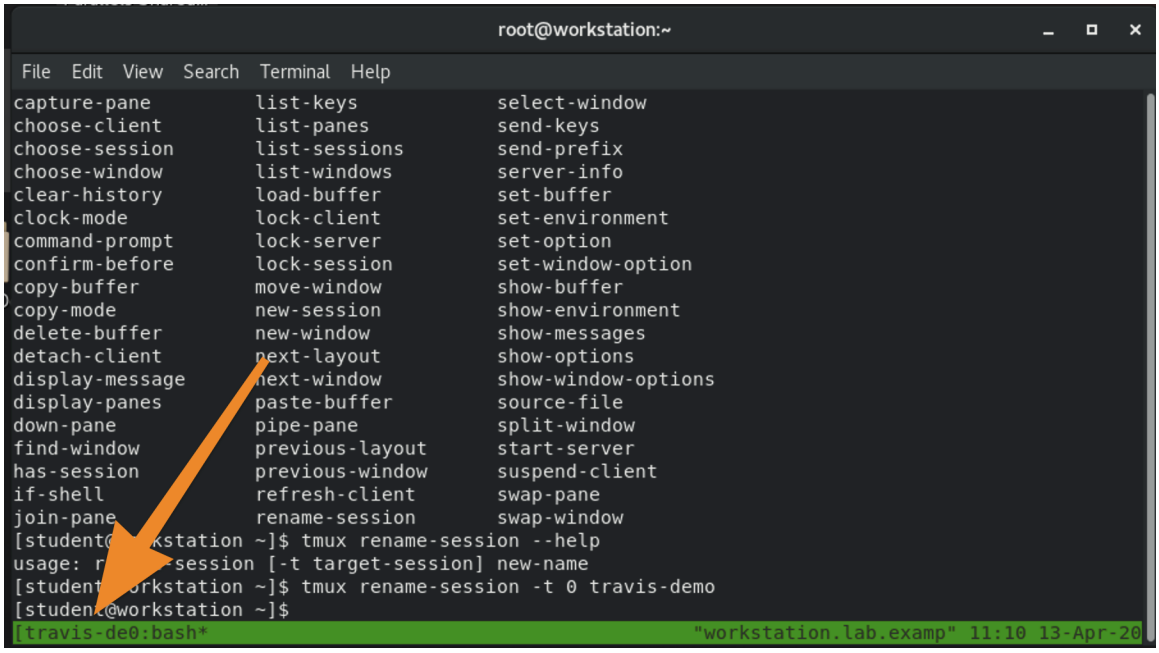

Figure 3. Renamed **tmux** *Session*

```
[student@workstation ~]$ tmux new-session
[detached (from session travis-demo)]

[student@workstation ~]$ tmux list-sessions
travis-demo: 1 windows (created Mon Apr 13 11:08:28 2020) [98x23]

[student@workstation ~]$ tmux attach-session -t travis-demo
```

1.3. NoMachine

NoMachine can be used to remote control graphical workstations. There is a client/server model that is setup and NoMachine makes the initial connection on port 4000 using SSL. NoMachine can be found on multiple platforms using the following URL: https://www.nomachine.com/.

Note Header

Encryption for NoMachine: https://www.nomachine.com/AR10K00705

SSL Client Authentication: https://www.nomachine.com/AR10M00866

NoMachine KB Articles for SSL: https://www.nomachine.com/articles?keys=ssl&an=ssl&form_build_id=form-111ef1c9e4bf56fd314eadead0c410b2&field_appliesto_value_many_to_one=All&custom_filter=any

2. Installing or Upgrading to Red Hat Enterprise Linux 8

2.1. Installing Red Hat Enterprise Linux 8

Notes on Installation

- **BaseOS** and **Appstream** Repositories are both required for installation of a RHEL8 system
  - Binary DVD contains both **BaseOS** and **Appstream** channels (*approx. 7.5GB*)
- RHEL8 Supports package Modularity
  - Modules are a set of RPMs
    - Application Streams: https://developers.redhat.com/blog/2018/11/15/rhel8-introducing-appstreams/
- System Purpose, System Roles, and Usage are all part of the new installation process in Anaconda

Red Hat Service Level Agreement

It is extremely important to note that the Service Level Agreement can cause issues with Red Hat Entitlements if not properly selected.

2.2. Upgrading Servers to Red Hat Enterprise Linux 8

2.3. FirewallD Service Definitions

FirewallD was introduced in RHEL7 as part of the SystemD transition and a new way to manage firewalls without using the underlying firewall implementation (**iptables**). FirewallD with **firewall-cmd** continues to be used in RHEL8 as the preferred method of managing and maintaining firewall rules.

FirewallD Resources

https://firewalld.org/

https://www.liquidweb.com/kb/an-introduction-to-firewalld/

https://cheatography.com/mikael-leberre/cheat-sheets/firewall-cmd/

2.3.1. The firewall-cmd Utility

The **firewall-cmd** utility is the primary method to manage and interact with firewall rules on RHEL7/8 systems. The **firewall-cmd** utility supports BASH completion and allows firewall rules to be added based on defined services or by specifying ports/protocols.

Listing 6. Allowing HTTP through the Firewall by Port/Protocol

```
[root@servera ~]# firewall-cmd --add-port=80/tcp
success

[root@servera ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp1s0
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 80/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@servera ~]# firewall-cmd --remove-port=80/tcp
success
```

Listing 7. Allowing HTTP through the Firewall by Service

```
[root@servera ~]# firewall-cmd --add-service=
Display all 154 possibilities? (y or n)

[root@servera ~]# firewall-cmd --add-service=http
success

[root@servera ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp1s0
  sources:
  services: cockpit dhcpv6-client http ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

firewall-cmd *Usage Warning*

The **firewall-cmd** utility can be used to make changes to the running firewall as shown in the examples above. This does not make changes to the firewall config file. In order to make the changes to the configuration file, it is necessary to use the **--permanent** options to have the changes written to a file.

When using **--permanent**, and you are not making changes to the current firewall runtime, it is also necessary to use: **firewall-cmd --reload** to reload or load new firewall rules from the firewall configuration file.

Important Header

It is important to note that presently the Red Hat Web Console (cockpit) only supports management of **firewalld** using defined services.

2.3.2. FirewallD Files and Locations

FirewallD has a few locations for files both for configuration and usage. As with most configuration files on a Linux system, those rely in **/etc/**. The user configurable files for FirewallD also reside in **/etc/**. Default configuration files for services, zones, and other FirewallD functionality resides in **/usr/lib/firewalld**. This location contains all defined services files and default configuration files for FirewallD and used by the **firewall-cmd** utility.

Listing 8. FirewallD Configuration Files

```
[root@servera ~]# tree /etc/firewalld/
/etc/firewalld/
├── firewalld.conf
├── helpers
├── icmptypes
├── ipsets
├── lockdown-whitelist.xml
├── services
└── zones
    ├── public.xml
    └── public.xml.old

5 directories, 4 files
```

Listing 9. FirewallD Default Configuration Files

```
[root@servera ~]# tree /usr/lib/firewalld/
/usr/lib/firewalld/
├── helpers
│   ├── amanda.xml
│   ├── ftp.xml
│   ├── h323.xml
│   ├── irc.xml
│   ├── netbios-ns.xml
│   ├── pptp.xml
│   ├── proto-gre.xml
... output omitted ...
└── zones
    ├── block.xml
    ├── dmz.xml
    ├── drop.xml
    ├── external.xml
    ├── home.xml
    ├── internal.xml
    ├── public.xml
    ├── trusted.xml
    └── work.xml

5 directories, 222 files
```

2.3.3. Defining a Custom Service File

It is possible to define custom **firewalld** service files. These files can be used for your environments for custom applications or custom firewall rules. These service files can also be checked into a version control system such as **git**.

Creating a Custom Service File using Existing File as Base

It is easiest to take an existing service file, copy it to the **/etc/firewalld/services** directory, rename and edit the file.

1. Copy existing FirewallD service file to **/etc/firewalld/services**

Listing 10. Using SSH Service File as a Starting Point

```
[root@servera ~]# cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/custom_ssh.xml
```

2. Edit the file and specify the new options

Listing 11. Edit the Custom SSH Service Definition

```
[root@servera ~]# vim /etc/firewalld/services/custom_ssh.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH_Custom</short>
  <description>This is a custom SSH service definition for paranoid people that don't want to run SSH on the default port of 22. This will
allow SSH to run on the port 8022 to meet our defined security guidance.</description>
  <port protocol="tcp" port="8022"/>
</service>
```

3. Listing FirewallD Services to Verify New Service

Listing 12. Getting Listing of FirewallD Services

```
[root@servera ~]# firewall-cmd --get-services | grep custom_ssh
```

FirewallD Delays in Discovering a Service

Depending on system speed and refreshing of FirewallD daemon, the new service might not be picked up immediately. It will be discovered or if you are in a hurry, you can run **firewall-cmd --reload** command to immediately have the service discovered and available.

2.3.4. FirewallD Configuration Files

As stated above, the main FirewallD configuration files are located in **/etc/firewalld**. To specifically change or view the configuration file on the system, you generally want to look at the firewalls in the **Firewall Zone**. This is found by opening the corresponding zone file in **/etc/firewalld/zones** directory.

1. Viewing Firewall configuration for the Public Zone.

Listing 13. FirewallD Configuration for Public

```
[root@servera ~]# cat /etc/firewalld/zones/public.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Public</short>
  <description>For use in public areas. You do not trust the other computers on networks to not harm your computer. Only selected incoming
connections are accepted.</description>
  <service name="ssh"/>
  <service name="dhcpv6-client"/>
  <service name="cockpit"/>
</zone>
```

2. Adding a custom service

Listing 14. Adding our new Service

```
[root@servera ~]# firewall-cmd --add-service=custom_ssh --permanent
success
```

3. Verifying Firewall configuration for the Public Zone.

Listing 15. FirewallD Configuration for Public with Custom Service

```
[root@servera ~]# cat /etc/firewalld/zones/public.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Public</short>
  <description>For use in public areas. You do not trust the other computers on networks to not harm your computer. Only selected incoming
connections are accepted.</description>
  <service name="ssh"/>
  <service name="dhcpv6-client"/>
  <service name="cockpit"/>
  <service name="custom_ssh"/>
</zone>
```

3. Provisioning and Configuring Servers

3.1. Performing Package Management using Yum

3.2. Administering Servers with Cockpit

3.2.1. Extending Cockpit

3.2.2. Installing Additional Cockpit Packages

Listing 16. Install all Base Cockpit Packages

```
[root@servera ~]# yum install cockpit*
```

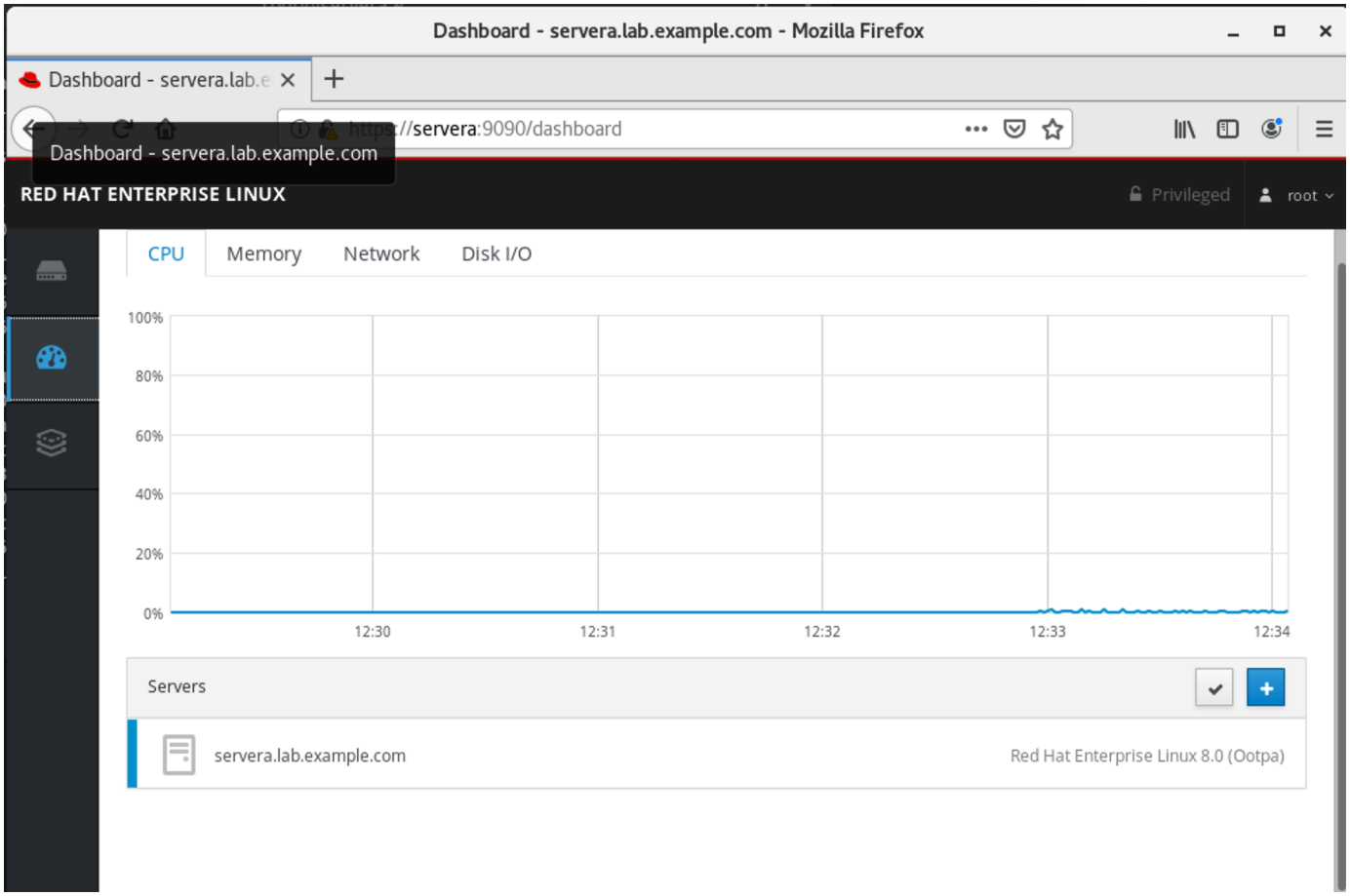

Figure 4. Cockpit Dashboard

Important Header

Using the installation method above will install all Cockpit packages and plugins that are available in currently subscribed channels. However, not all components will work until additional back-end components are installed. For example, the **Composer** cockpit plugins need composer and other items installed in order to be able to be fully utilized. This installation gives the **Cockpit Dashboard Plugin** which allows connecting to multiple Web Consoles.

3.2.2.1. Cockpit to Manage Multiple Systems

It is possible to use a single **cockpit** Interface to manage multiple servers.

1. Ensure cockpit socket/service is running and configured on all systems.

Listing 17. Test and Enable Cockpit

```
[student@workstation ~]$ ssh root@servera
Activate the web console with: systemctl enable --now cockpit.socket

[root@servera ~]# systemctl enable --now cockpit.socket
Created symlink /etc/systemd/system/sockets.target.wants/cockpit.socket → /usr/lib/systemd/system/cockpit.socket.
```

2. Connect to the Cockpit Web Console

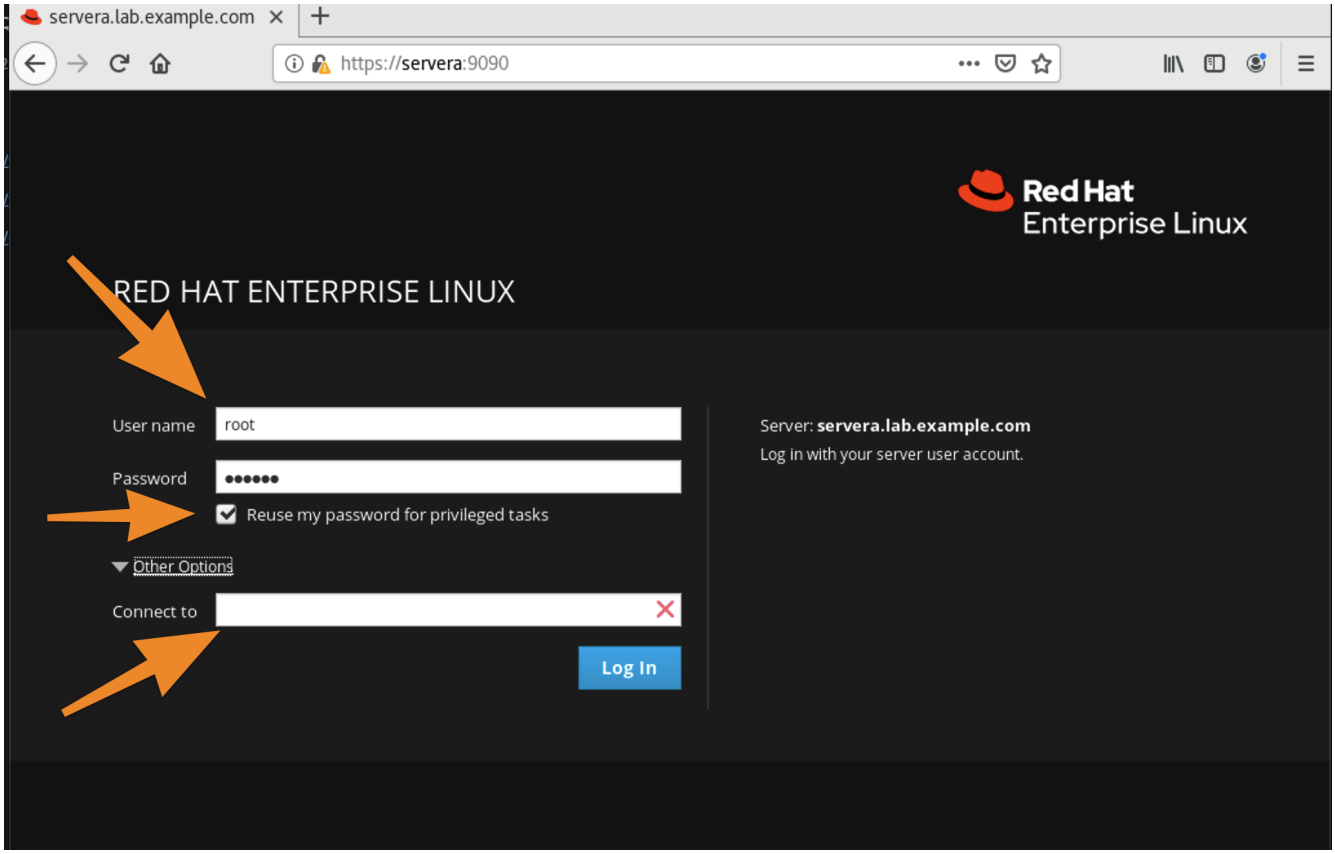

Figure 5. Red Hat Web Console (cockpit)

3. Add another Web Console from the Cockpit Dashboard
   a. Navigate to the Dashboard
   b. Click the "+" and complete the information

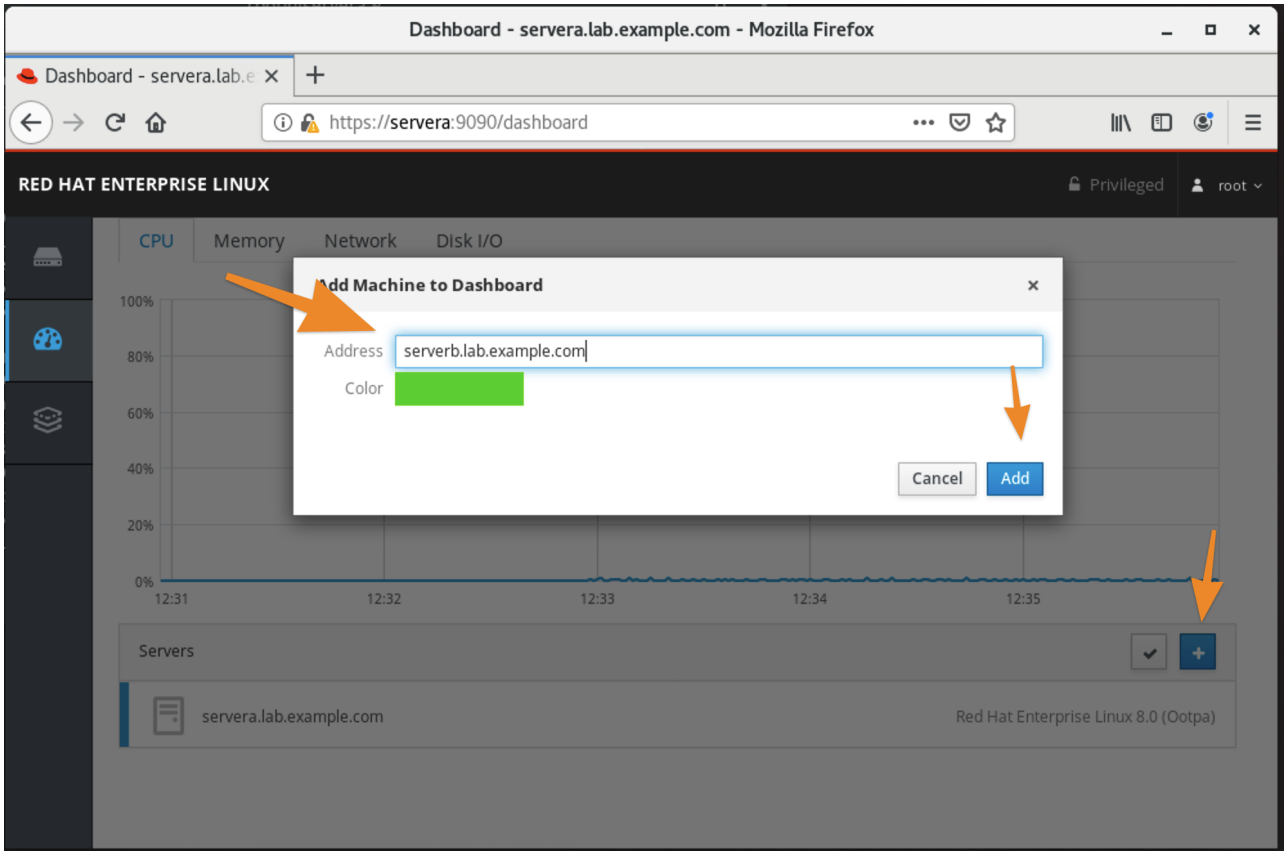

Figure 6. Adding Additional machines

4. Verify the System Fingerprint and click **Connect** image::Chapter3-87075.png[title="Fingerprint Verification", align="center"]
5. It is now possible to switch systems from the System Drop-down menu

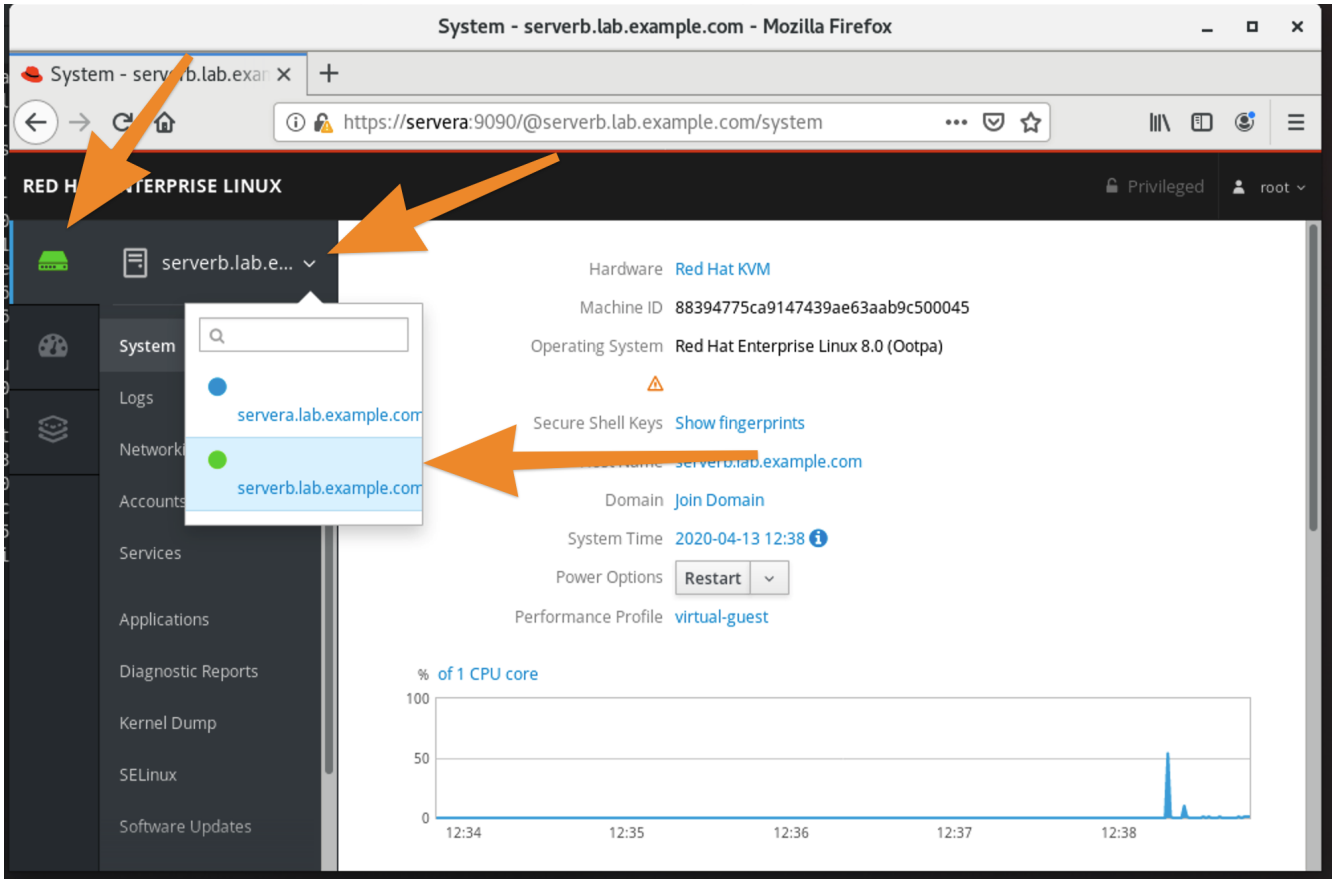

Figure 7. Switching Systems

3.3. Building System Images with Composer

3.3.1. Composer with Cockpit

In order to use the Lorax composer with **Cockpit** the cockpit packages and the composer packages must be installed.

1. Install the Lorax composer packages

Listing 18. Installing Composer Packages

```
[root@servera ~]# yum install lorax-composer composer-cli cockpit-composer
```

2. Enable the Composer Services

Listing 19. Enable Lorax Composer Socket

```
[root@servera ~]# systemctl enable --now lorax-composer.socket
```

3. Open the Red Hat Web Console and Navigate to Image Builder

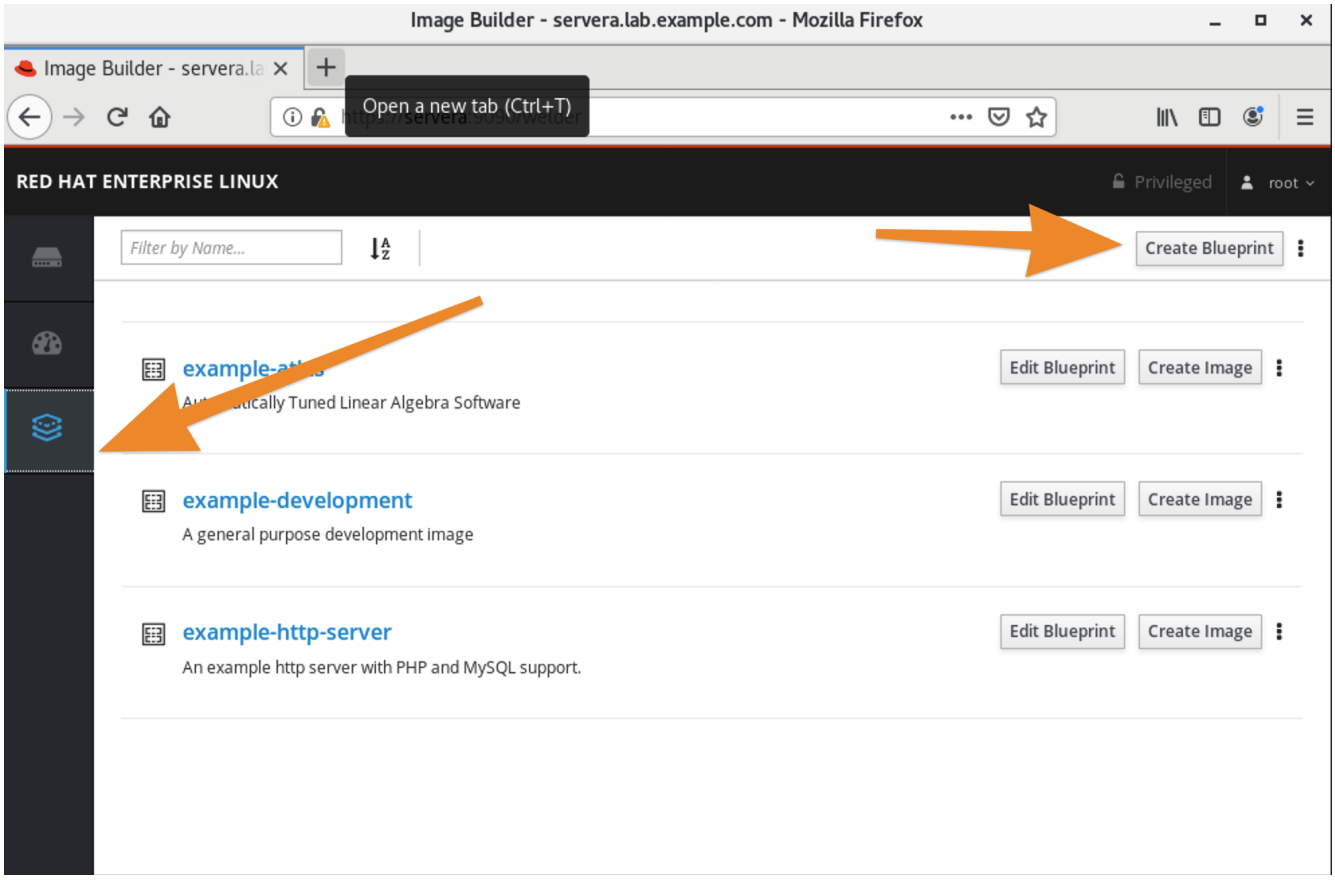

Figure 8. Image Builder - Web Console

3.4. Automating with RHEL System Roles

3.4.1. Ansible Basics

In order to use Ansible, the following items are required:

- **ansible** is installed on the Control Node
- An **ansible.cfg** file exists and points to a working inventory
- An **inventory** file exists containing managed nodes

3.4.1.1. Ansible Concepts and Architecture

Ansible Machine Types

- **Control Node**: Location where Ansible is installed and used to run playbooks and execute Ansible ad-hoc commands
- **Managed Host**: Network device that is managed by an Ansible control node

The required Ansible components are the following:

- **ansible.cfg**: Ansible configuration file with directives on how Ansible should work
- **inventory**: Listing and organization of ansible managed nodes/hosts
- **module**: Small piece of code to perform a variety of tasks (generally written in Python or Powershell)
- **plugins**: Code that can be used to extend and adapt Ansible to new uses. Capable of manipulating data and extracting information from output. (*Used more in the next course DO447*)
- **Ansible Tower/API**: Framework to control and manage Ansible at scale. Ansible Tower is not a core part of Ansible, but is an add-on product to more effectively utilize Ansible with teams.

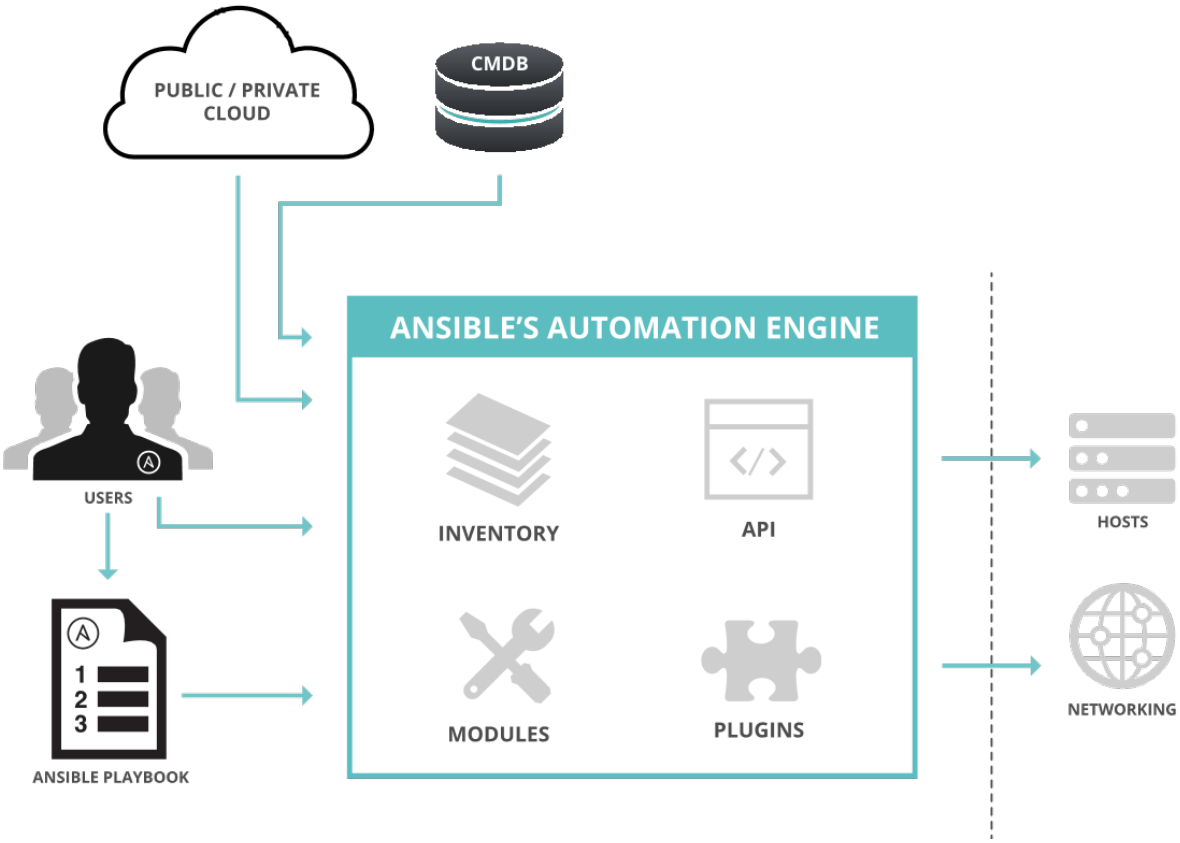

Figure 9. Ansible Architecture

3.4.1.2. Building an Ansible Inventory

An **inventory** file is the minimum needed item for Ansible to run. Inventory files provide a listing of hosts for Ansible to manage. There are multiple ways to define an inventory file.

3.4.1.2.1. Static Inventory

Static inventory files are generally defined in the INI format.

Listing 20. Simple Inventory

```
web1.example.com
web2.example.com
db1.example.com
db2.example.com
192.0.2.42

[webservers]
web1.example.com
web2.example.com
192.0.2.42
```

3.4.1.3. Ansible Configuration Files

It is important to note, there can be multiple Ansible configuration files. Ansible will look for and process configuration files in order.

1. Location specified in **ANSIBLE_CONFIG**
2. The **ansible.cfg** file in the current working directory
3. The **.ansible.cfg** file in the user's home directory
4. The default **/etc/ansible/ansible.cfg** file

It is important to note that the first file in this order Ansible sees is what will be used. Therefore, if anything is setup for the **ansible.cfg** file in locations 1-3, it won't ever use the default file ***/etc/ansible/ansible.cfg***

Ansible Configuration file documentation: http://docs.ansible.com/ansible/intro_configuration.html

Table 2. **Ansible Settings**

Setting	Command Line Option
inventory	Location of the Ansible inventory file.
remote_user	The remote user account used to establish connections to managed hosts.

Setting	Command Line Option
become	Enables or disables privilege escalation for operations on managed hosts.
become_method	Defines privilege escalation method on managed hosts.
become_user	User account to escalate privileges to on managed hosts.
become_ask_pass	Defines whether privilege escalation on managed hosts should prompt for password.

4. Adapting to Core System Changes

4.1. Displaying the Desktop with Wayland and X

4.1.1. Wayland and X11

Wayland replaced Xorg and the default X11 display server. Because of how Wayland compositing works and that Xorg no longer is providing an X11 server, there is no way to forward X11 traffic over an SSH connection. As mentioned earlier, NoMachine can provide a solution if the server has a graphical interface installed and NoMachine is installed and running on both the client and the Server.

For now, it is possible to switch the display server from Wayland to Xorg allowing the X11 server to function which would the provide the ability to forward an X11 connection.

Wayland and X11 Resources

https://wayland.freedesktop.org/xserver.html

https://www.tecmint.com/configure-xorg-as-default-gnome-session/

https://www.redhat.com/en/blog/red-hat-enterprise-linux-8-gnome-and-display-server-changes

https://unix.stackexchange.com/questions/202891/how-to-know-whether-wayland-or-x11-is-being-used

https://www.secjuice.com/wayland-vs-xorg/

1. Login and determine session ID using **loginctl**

Listing 21. Determining Session ID

```
[student@workstation ~]$ loginctl
SESSION  UID USER    SEAT  TTY
      4    0 root
      6 1000 student seat0 tty2
     c1   42 gdm     seat0 tty1
```

2. Use **loginctl**

Listing 22. Determining Session Type

```
[student@workstation ~]$ loginctl show-session 6 -p Type
Type=wayland
```

3. Examine the GDM configuration

Listing 23. Examine GDM Configuration

```
[student@workstation ~]$ grep -i wayland /etc/gdm/custom.conf
#WaylandEnable=false
```

Disable Wayland and Switch to XOrg X11

You can disable Wayland and enabled Xorg X11 by modifying the line in the **/etc/gdm/custom.conf** to just uncomment so it would read as:

```
[student@workstation ~]$ grep -i wayland /etc/gdm/custom.conf
WaylandEnable=false
```

Also, you will need to add **DefaultSession=gnome-xorg.desktop** to the **[daemon]** section of the file below the WaylandEnable=False.

```
[student@workstation ~]$ cat /etc/gdm/custom.conf
# GDM configuration storage

[daemon]
# Uncoment the line below to force the login screen to use Xorg
WaylandEnable=false
DefaultSession=gnome-xorg.desktop

... output omitted ...
```

If any changes are made to the GDM configuration, the system must be rebooted in order to take effect.

Setting a Default Desktop Session

The **DefaultSession=gnome-xorg.desktop** section isn't specifically required for enabling X11 and disabling Wayland. It is one method that can be used to ensure that **xorg** is used as the default Gnome Session Manager.

Know Server Locations

It is important to know information about the server/desktop that is being used as the client system. The client system is responsible for having an XWindows server to allow X11 forwarding. It is possible to have a remote system with Wayland. As long as the client system supports X11 and has an X-Windows server, it is still possible to forward X11 connections over SSH.

Windows SSH Client with XWindows Support: https://mobaxterm.mobatek.net/

4.2. Managing User Authentication with Authselect

Objective

- Manage user authentication settings in **PAM**, **NSS**, and **dconf** using **Authselect**
- Explain the differences between **Authselect** and **Authconfig**

4.2.1. Introducing Authselect

Replaces **authconfig** making changes simpler for administrators. Used for configuration of passwords, certificates, smart cards, and fingerprints.

Features

- Adjusts **PAM**, **NSS**, and **GNOME dconf** settings.
- Ships with three ready-to-use profiles: **sssd**, **winbind**, and **nis**.
- **pam_pwquality** is enabled by default to enforce password quality restrictions on local users.

Comparing Authselect and Authconfig

- **Authselect** uses profiles instead of modifying authentication config files directly
- **Authselect** only modifies files in **/etc/nsswitch.conf**, **/etc/pam.d/**, and **/etc/dconf/db/distro.d/**

Using Authselect

- Use the **authselect list** command to list the default and custom profiles.
- The default profiles are stored in **/usr/share/authselect/** default.
- Use the **authselect create-profile** command to create new custom profiles.
- Custom profiles are stored in the **/etc/authselect/custom/** directory.

Example 1. Authselect Demo

Listing 24. Getting Help for **authselect**

```
[root@servera ~]# authselect --help
Usage:
authselect COMMAND COMMAND-ARGS

Available commands:
- select             Select profile
- apply-changes      Regenerate configuration for currently selected command
- list               List available profiles
- show               Show profile information
- requirements       Print profile requirements
- current            Get identificator of currently selected profile
- check              Check if the current configuration is valid
- test               Print changes that would be otherwise written
- enable-feature     Enable feature in currently selected profile
- disable-feature    Disable feature in currently selected profile
- create-profile     Create new authselect profile

Common options:
  --debug            Print error messages
  --trace            Print trace messages
  --warn             Print warning messages

Help options:
  -?, --help         Show this for a command
  --usage            Show brief usage message for a command
```

Listing 25. Listing **authselect** *Profiles*

```
[root@servera ~]# authselect list
- nis        Enable NIS for system authentication
- sssd       Enable SSSD for system authentication (also for local users only)
- winbind    Enable winbind for system authentication
```

Listing 26. Listing **authselect** *Current Profile*

```
[root@servera ~]# authselect current
Profile ID: sssd
Enabled features: None
```

Listing 27. Using **authselect** *to Create a Profile*

```
[root@servera ~]# authselect current
Profile ID: sssd
Enabled features: None
[root@servera ~]# authselect create-profile Travis_Custom \
>  -b sssd --symlink-meta
New profile was created at /etc/authselect/custom/Travis_Custom
[root@servera ~]#
```

Listing 28. Modifying a Custom Profile with Additional Features

```
[root@servera ~]# cd /etc/authselect/custom/Travis_Custom
[root@servera Travis_Custom]#

[root@servera Travis_Custom]# authselect select custom/Travis_Custom with-sudo
Profile "custom/Travis_Custom" was selected.
The following nsswitch maps are overwritten by the profile:
- passwd
- group
- netgroup
- automount
- services
- sudoers

Make sure that SSSD service is configured and enabled. See SSSD documentation for more information.


[root@servera Travis_Custom]# authselect current
Profile ID: custom/Travis_Custom
Enabled features:
- with-sudo
[root@servera Travis_Custom]
```

authselect keeps profiles of authentication configuration files and settings. This tool is based on profiles and allows custom profiles to be created. This should only be used for systems that are not part of a REALM with LDAP server or AD - **ipaclient** should be used or **realmd** for AD and LDAP/IPA respectively.

4.3. Configuring NTP with Chrony

Objective

- Maintain NTP time synchronization using **Chrony**, and configure the time zone with **timedatectl**.

4.3.1. Chrony Replaces ntpd

Chrony was implemented initially as part of RHEL7, but as of RHEL8, **chrony** is the only option for NTP services and **ntpd** is no longer available.

Using Chrony as NTP Replacement

- The **/usr/share/doc/chrony/ntp2chrony.py** script **/etc/ntp.conf** file to **/etc/chrony.conf**
- **timedatectl** can be used to set time aand ate and display.
- Use **timedatectl set-timezone** to define timezones.

Listing 29. Using the **timedatectl** *Command*

```
[root@servera ~]# timedatectl
list-timezones   set-ntp          set-timezone     show-timesync    timesync-status
set-local-rtc    set-time         show             status


[root@servera ~]# timedatectl set-time 12:05:17
Failed to set time: NTP unit is active

[root@servera ~]# vim /etc/chrony.conf

[root@servera ~]# chronyc sources -v
210 Number of sources = 1

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
| /   '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                               .- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.         |  xxxx = adjusted offset,
||      Log2(Polling interval) --.      |        |  yyyy = measured offset,
||                                \     |        |  zzzz = estimated error.
||                                 |    |         \
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* classroom.example.com         8   6    17    30   +506us[+1518us] +/- 2959us

[root@servera ~]# timedatectl
               Local time: Thu 2019-07-25 16:09:38 EDT
           Universal time: Thu 2019-07-25 20:09:38 UTC
                 RTC time: Thu 2019-07-25 16:13:02
                Time zone: America/New_York (EDT, -0400)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

4.4. Managing Python Versions in Red Hat Enterprise Linux 8

5. Implementing Storage Using New Features

5.1. Managing Layered Storage with Stratis

Objective: Manage multiple storage layers using Stratis local storage management.

Packages

- **stratis-cli**: Provides the *stratis* command that sends user requests to **stratisd** service.
- **stratisd**: Provides the *stratisd* service implementing a D-Bus interface used managing and monitoring elements of **Stratis**.

5.1.1. Describing the Stratis Architecture

Traditional RHEL storage solutions allowed for scalable file systems, snapshots, RAID logical devices, multipathing, thin provisioning, caching, and a few other things like support for virtualization.

RHEL Storage Solutions

- **dm**: Device Mapper
- **LVM**: Logical Volume Manager
- **XFS**: 64-bit journaling file system (originally created by SGI)

Each of the RHEL storage solutions required management using layer specific commands requiring management of physical devices, volumes, and file systems as separate storage components.

RHEL 8 Storage Solution

- **Stratis**: Storage management solution that works with existing RHEL storage components.

Prior to RHEL 8, the ability to use ***volume-managing file systems*** was only considered development and not stable enough to become primary local storage for an enterprise-class system.

RHEL 8 introduced the **Stratis** storage management solution to work with the existing enterprise-class RHEL storage components. **Stratis** runs as a service that can manage pools of physical storage devices, create and manage volumes (transparently) for file systems being created. Since it uses existing storage drivers and tools, all advanced features available in **LVM**, **XFS**, and **device mapper** are used and supported by **Stratis**.

Volume-Managed File System

Volume-managed file systems are built inside of shared pools of disks using *thin provisioning*. **Stratis** file systems are built using a hidden LVM volume and **Stratis** manages the underlying volume and can expand it when needed.

Stratis file systems don't have fixed sized an no longer preallocate unused block space. Essentially, this can yield results similar to just-in-time configuration of file systems.

Stratis *File System Facts*

- In-use size of a file system is seen as amount of blocks in use by contained files.

- Space available to a file system is the amount of ***unused*** space in the pooled devices
- Multiple file systems can reside in same pool of disk devices (sharing available space)
- File systems can reserve pool space to guarantee availability when needed.

Stratis uses stored metadata to recognize managed pools, volumes and file systems. **Stratis** volumes should never be managed manually and only be managed using **Stratis** tools and commands. Manually managing/configuring/changing a **Stratis** file system could result in data loss and prevent the file system from being recognized.

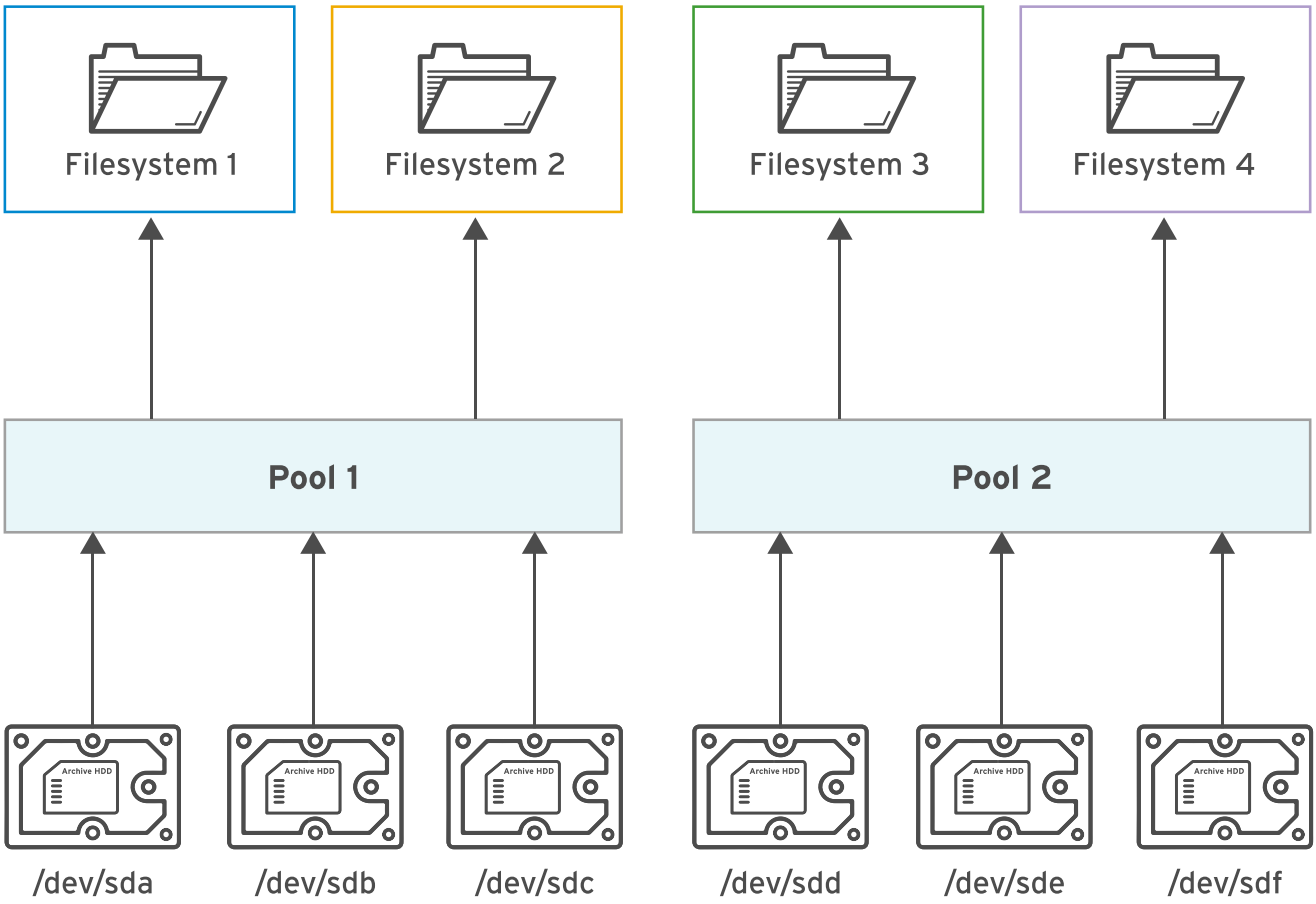

Figure 10. Stratis Elements

Stratis can break up storage into pool tiers. The **data tier** and the **cache tier**. Typically, the **cache tier** should use block devices with higher (IOPS) such as SSD.

Stratis *Pool Groups*

- **data tier**: Focuses on flexibility and integrity

- **cache tier**:Focuses on improved performance

5.1.1.1. Describing the Simplified Storage Stack

It should be noted that **Anaconda** now leverages **Stratis** to simplify disk setup and management. Other products using **Stratis** include **Cockpit**, Red Hat Virtualization (RHV), and Red Hat Enterprise Linux Atomic Host.

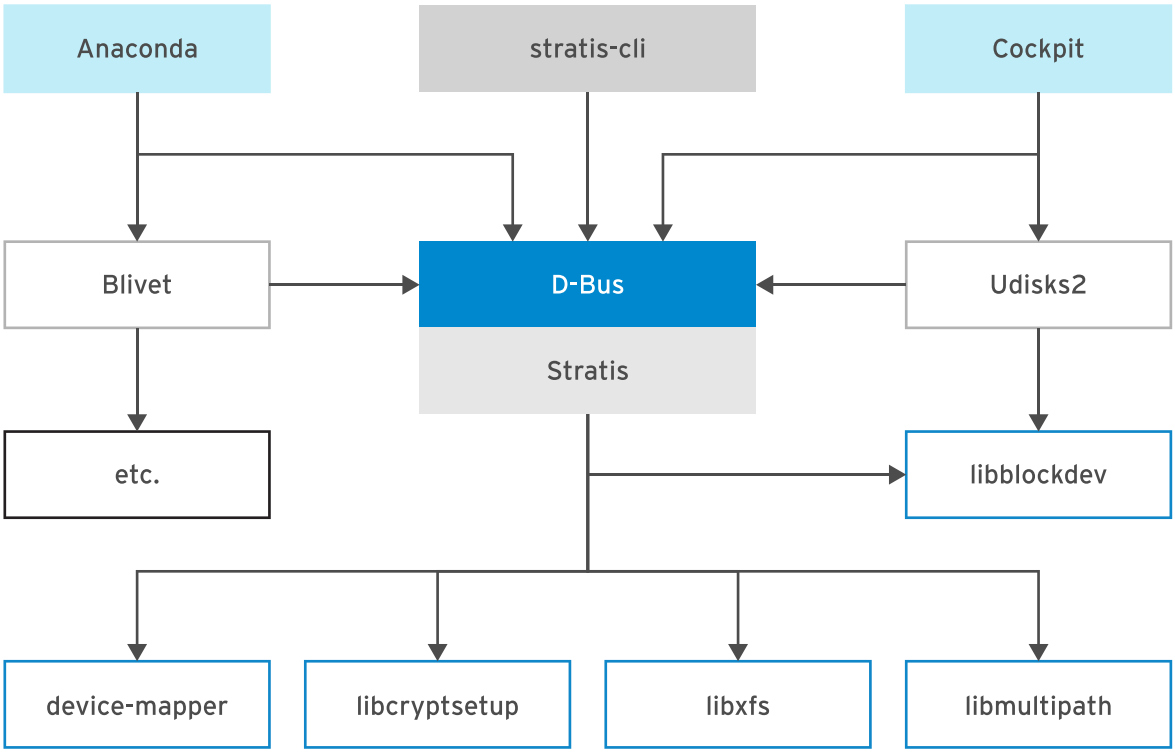

Figure 11. Stratis in the Storage Management Stack

Stratis provides management to storage space and snapshots and allows easier integration with higher-level management tools than using a CLI programmatically.

5.1.1.2. Describing Stratis Layers

Stratis has multiple layers to manage the storage system.

- **Backstore**: Subsystem managing block devices
- **Thinpool**: Subsystem that manages pools
- **dm-thin**: The **Thinpool** subsystem uses the **dm-thin** device mapper driver to replace LVM for virtual volume sizing and management.

The **Backstore** subsystem has data tier maintaining on-disk metadata for block devices and detects/corrects data corruption. The

cache tier uses high-performance block devices acting as cache on top of the data tier.

The **Thinpool** subsystem manages thin-provisioned volumes associated with Stratis file systems.

The **dm-thin** driver creates volumes with a large virtual size, formatted with XFS, but a smaller physical size. As the physical size becomes full, **Stratis** automatically enlarges the disk.

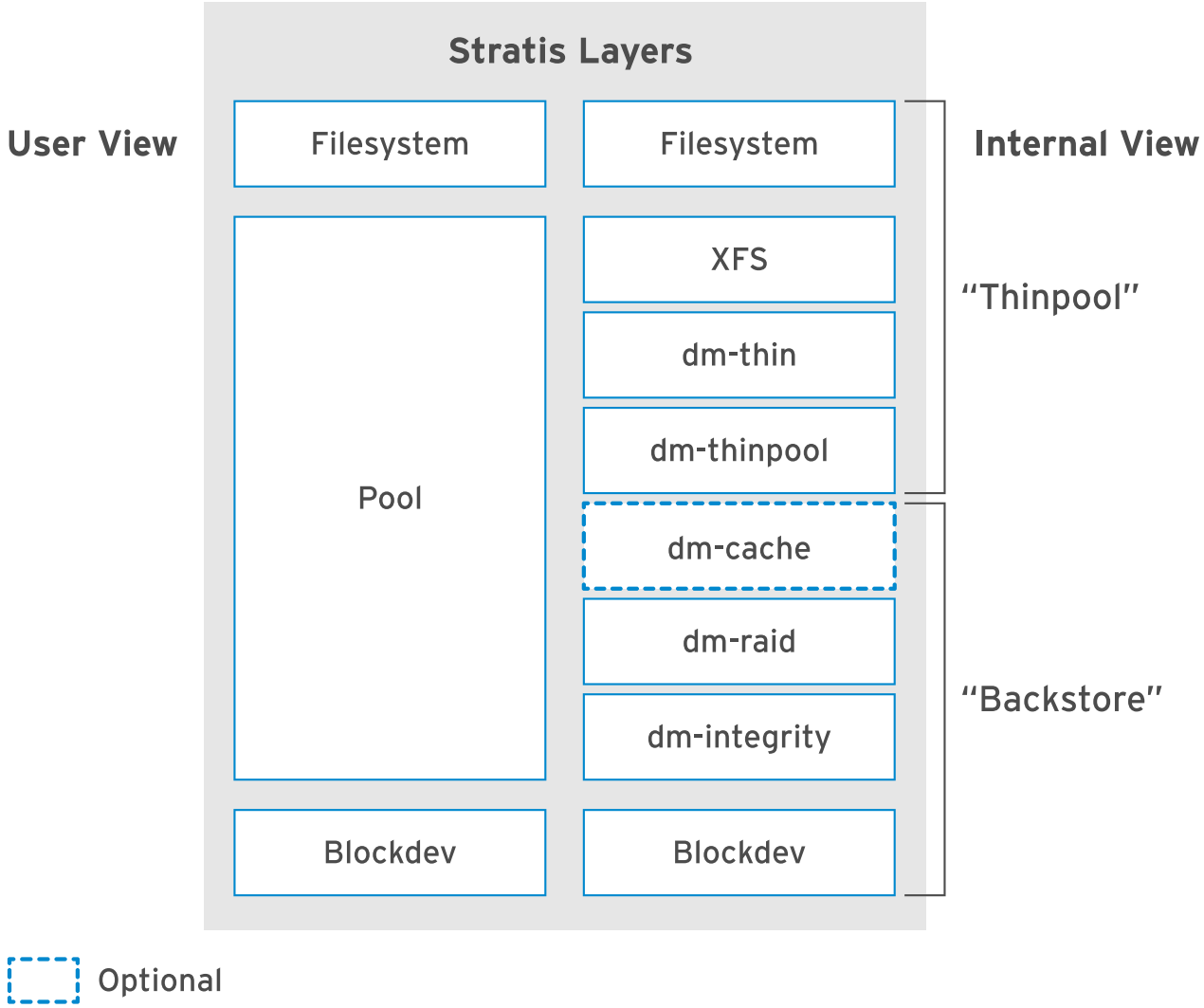

Figure 12. Stratis Layers

5.1.1.3. Managing Thin-provisioned File Systems

In order to manage file systems using **Stratis** the ***stratis-cli*** and ***stratisd*** packages must be installed.

Example 2. **Stratis Demo**

1. Install **Stratis**

```
[root@servera ~]# yum install stratis-cli stratisd
```

2. Enable and Activate the **Stratis** service

```
[root@servera ~]# systemctl enable stratisd --now

[root@servera ~]# systemctl status stratisd
● stratisd.service - A daemon that manages a pool of block devices to create fl>
   Loaded: loaded (/usr/lib/systemd/system/stratisd.service; enabled; vendor pr>
   Active: active (running) since Fri 2019-06-21 14:49:32 EDT; 16s ago
     Docs: man:stratisd(8)
 Main PID: 23003 (stratisd)
    Tasks: 1 (limit: 11405)
   Memory: 884.0K
   CGroup: /system.slice/stratisd.service
           └─23003 /usr/libexec/stratisd --debug
```

Using and Managing Stratis

- Creating Pools of one or more block devices

```
[root@servera ~]# stratis pool create DemoStorage /dev/vdb
```

- List Stratis Pools (**stratis pool list**)

```
[root@servera ~]# stratis pool list
Name           Total Physical Size  Total Physical Used
DemoStorage                  5 GiB              52 MiB
```

- Add Additional Disks/Block Devices (**stratis pool add-data**)

```
[root@servera ~]# stratis pool add-data DemoStorage /dev/vdc
```

- View block devices in a pool (**stratis blockdev list**)

```
[root@servera ~]# stratis blockdev list DemoStorage
Pool Name    Device Node    Physical Size   State  Tier
DemoStorage  /dev/vdb              5 GiB  In-use  Data
DemoStorage  /dev/vdc              5 GiB  In-use  Data
```

- Create a Filesystem (**stratis filesystem create**)

```
[root@servera ~]# stratis filesystem create DemoStorage DemoStorageFS1
```

- Create File system snapshots (**stratis filesystem snapshot**)

```
[root@servera ~]# stratis filesystem snapshot DemoStorage DemoStorageFS1 DS_FS_Snap1
```

- Listing Available Filesystems (**stratis filesystem list**)

```
[root@servera ~]# stratis filesystem list
Pool Name    Name            Used     Created            Device                              UUID
DemoStorage  DemoStorageFS1  546 MiB  Jun 21 2019 15:12  /stratis/DemoStorage/DemoStorageFS1  0155cbeda6cd4735ad540f394e74a8d3
DemoStorage  DS_FS_Snap1     546 MiB  Jun 21 2019 15:14  /stratis/DemoStorage/DS_FS_Snap1     0ad900732d8f4daba22037c0f4f1e8b0
```

Mounting Stratis Filesystems

It is best to mount the filesystem using the UUID in **/etc/fstab**. Additionally, it is extremely important to use the **x-systemd.requires=stratisd.service** mount option as it delays the mount attempt until **systemd** starts **stratisd.service**.

- Obtaining UUID can be done with **lsblk** or with the **stratis filesystem list**

```
[root@servera ~]# lsblk --output=UUID /stratis/DemoStorage/DemoStorageFS1
UUID
0155cbed-a6cd-4735-ad54-0f394e74a8d3
```

- Updating **/etc/fstab**

```
[root@servera ~]# mkdir /MyDemoStorage ①

[root@servera ~]# echo "UUID=0155cbed-a6cd-4735-ad54-0f394e74a8d3 /MyDemoStorage xfs defaults,x-systemd.requires=stratisd.service 0
0" >> /etc/fstab ②

[root@servera ~]# mount -a ③
```

① Create a mountpoint

② Update **/etc/fstab** with the newly defined storage, mountpoint and parameters

③ Test the changes to **/etc/fstab** before rebooting the system using **mount -a** option.

Stratis *Help*

The **Stratis** CLI has multiple ways of obtaining help. The traditional **man** page provides documentation and examples on using **Stratis**. In addition to the **man** pages, the **stratis** CLI can use the ***--help*** option at multiple levels of the command to obtain detailed help information.

Listing 30. **Stratis** *Man Page*

```
[root@servera stratis-cli]# man stratis

... omitted lines ...

EXAMPLES
       Example 1. Creating a Stratis pool

       stratis pool create mypool /dev/sdb /dev/sdc

       Example 2. Creating a filesystem from a pool

       stratis filesystem create mypool data1

... Output Omitted ...
```

Listing 31. **Stratis** *CLI Help*

```
[root@servera stratis-cli]# stratis --help ①
usage: stratis [-h] [--version] [--propagate]
               {pool,blockdev,filesystem,fs,daemon} ...

Stratis Storage Manager

... Output Omitted ...

subcommands:
  {pool,blockdev,filesystem,fs,daemon}
    pool                Perform General Pool Actions
    blockdev            Commands related to block devices that make up the
                        pool
    filesystem (fs)     Commands related to filesystems allocated from a pool
    daemon              Stratis daemon information

[root@servera stratis-cli]# stratis pool --help ②
usage: stratis pool [-h] {create,list,destroy,rename,add-data,add-cache} ...

optional arguments:
  -h, --help            show this help message and exit

subcommands:
  {create,list,destroy,rename,add-data,add-cache}

... Output Omitted ...


[root@servera stratis-cli]# stratis pool create --help ③
usage: stratis pool create [-h] [--redundancy {none}]
                           pool_name blockdevs [blockdevs ...]

... Output Omitted ...
```

① Top-level **stratis** CLI command help

② Pool subcommand-level CLI command help

③ Pool Create subcommand-level CLI command help

Failure to use the **x-systemd.requires=stratisd.service** mount option in **/etc/fstab** for the Stratis file system will result in the machine booting to **emergency.target** on the next reboot.

Documentation: RHEL 8 - Managing File Systems - Chapter 11

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/managing_file_systems/index#managing-layered-local-storage-with-stratis_managing-file-systems

5.2. Compressing and Deduplicating Storage with VDO

Objective: Optimize storage space by using VDO to compress and deduplicate data on storage devices.

5.2.1. Describing Virtual Data Optimizer

RHEL 8 includes the Virtual Data Optimizer (VDO) driver that optimizes data footprint on block devices.

- **VDO**: Linux device mapper driver reducing disk space on block devices and minimizes replication of data saving disk space and can also increase data throughput. The VDO driver includes two (2) kernel modules to accomplish this task.
- **kvdo**: Kernel module included as part of VDO to transparently control data compression.
- **uds**: Kernel module allowing deduplication of data.

The VDO layer is on top of existing storage devices (RAID or local disks). The block devices can also be encrypted devices. Storage layers such as LVM volumes and file systems are placed on top of a VDO device.

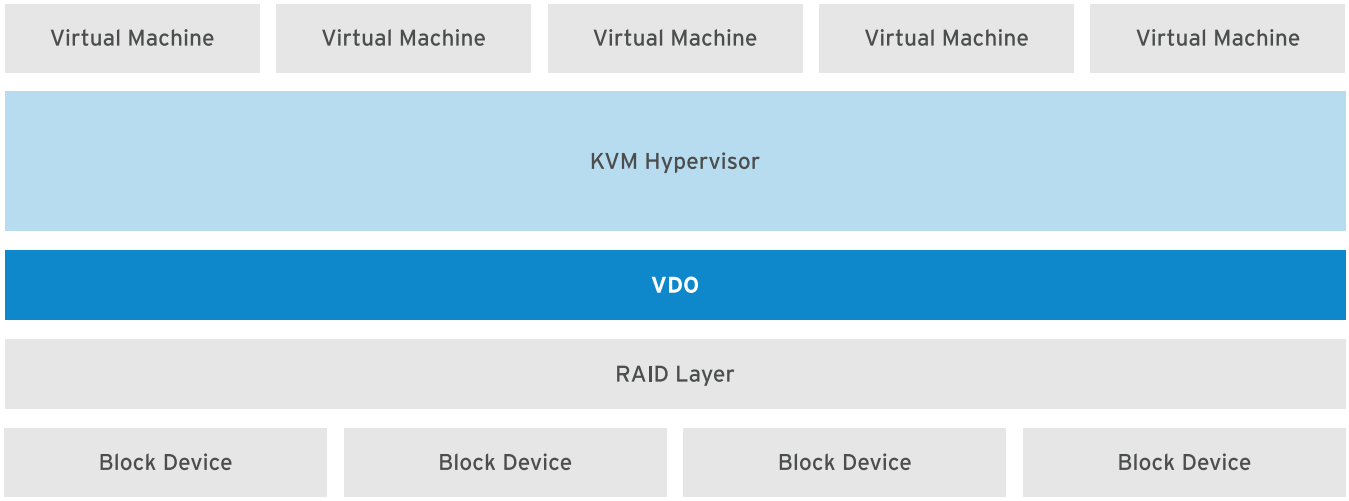

Figure 13. VDO-based Virtual Machines

VDO Three Phases to Reduce Data Footprint on Storage Devices

1. **Zero-Block Elimination**: Filters out data blocks that contain only zero values and records information about those blocks in metadata only. Non-zero blocks are passed to next phase. This phase enables thin provisioning.

2. **Deduplication**: Eliminates redundant data blocks. When multiple copies of same data are created, VDO detects duplicate data blocks and updates the metadata to use duplicate blocks as references to the original data block without creating redundant data blocks. This is accomplished using **uds** which is the **universal deduplication service** kernel module that checks redundancy of data through the metadata it maintains.
3. **Compression**: Last phase of VDO. The **kvdo** kernel module compresses data blocks using LZ4 compression and groups them on 4KB blocks.

5.2.2. Implementing Virtual Data Optimizer

Logical devices created with VDO are called **VDO Volumes**. A VDO Volume is similar to a disk partition in that it can be formatted and mounted like a regular filesystem. A VDO volume can also be used as an LVM physical volume.

In order to create a VDO volume, the block device must be specified along with the name of the logical device that VDO will present to the user. Optionally, the logical size of the VDO volume can be specified. The logical size of the VDO volume can be greater than the physical size of the actual block device.

VDO volumes are thinly provisioned allowing the user to only see the logical space in use and not the physical space available. If the VDO logical size isn't specified during creation, VDO assumes the actual physical size as the logical size of the volume.

If the VDO volume logical size is greater than the physical size, it should be proactively monitored using **vdostats --verbose**.

Example 3. **VDO** *Examples*

- Install **VDO** packages (**vdo kmod-kvdo**)

```
[root@servera ~]# yum install vdo kmod-kvdo
```

- Create a VDO Volume (**vdo create**)

```
[root@servera ~]# vdo create --name=MyVDO --device=/dev/vdd --vdoLogicalSize=75G
Creating VDO MyVDO
Starting VDO MyVDO
Starting compression on VDO MyVDO
VDO instance 0 volume is ready at /dev/mapper/MyVDO
```

- Format a VDO Volume

```
[root@servera ~]# mkfs.xfs /dev/mapper/MyVDO
meta-data=/dev/mapper/MyVDO      isize=512    agcount=4, agsize=4915200 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=19660800, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=9600, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

- Mounting a VDO Volume

Listing 32. Getting UUID for Block Device

```
[root@servera ~]#  lsblk --output=UUID /dev/mapper/MyVDO
UUID
f9fa889a-1e7f-44c3-9edf-47ee986d73f0
```

Listing 33. Creating a Mount Point

```
[root@servera ~]# mkdir /MyVDOStorage
```

*Listing 34. Modifying **/etc/fstab***

```
[root@servera ~]# echo "UUID=f9fa889a-1e7f-44c3-9edf-47ee986d73f0 /MyVDOStorage xfs defaults,x-systemd.requires=vdo.service 0 0" >>
/etc/fstab
```

*Listing 35. Testing **/etc/fstab***

```
[root@servera ~]# mount -a
```

- Analyzing a VDO Volume

```
[root@servera ~]# vdo status --name=MyVDO
VDO status:
  Date: '2019-06-21 17:11:14-04:00'
  Node: servera.lab.example.com
Kernel module:

... Output Omitted ...

    Logical size: 75G ①
    Logical threads: 1
    Max discard size: 4K
    Physical size: 5G ②
    Physical threads: 1
    Slab size: 2G
    Storage device: /dev/vdd
    VDO statistics: ③
      /dev/mapper/MyVDO:

... Output Omitted ...
```

① Shows the logical size that was defined for the VDO volume

② Shows the physical size of the disks backing the VDO volume

③ Begins the section providing VDO statistics

5.3. Administering NFS Enhancements

Objectives

- Identify NFS share information.
- Create a directory to use as a mount point.
- Mount an NFS share using the mount command or by configuring the /etc/fstab file.
- Unmount an NFS share using the umount command.
- Configure an NFS client to use NFSv4 using the new nfsconf tool.

5.3.1. Mounting and Unmounting NFS Shares

NFS (Network File System) - default in RHEL 8 is now NFS v4.2. NFS exports shares (directories) which a client can mount using

- The **mount** command manually
- Automatically at boot with **/fstab**
- On demand using **autofs** service or **systemd.automount**

Only the major versions NFSv4 and NFSv3 are supported and NFSv2 support has been dropped.

NFSv4 and greater use TCP protocol

Since NFSv4 is being used, the **showmount** command can no longer be used as it can only reliably query NFSv3 servers.

Example 4. Demo of NFS

Listing 36. Prepare NFS Export

```
[root@serverb ~]# mkdir /ExportDemo
[root@serverb ~]#  echo "/ExportDemo *(rw,no_root_squash)" > /etc/exports
[root@serverb ~]# exportfs -ra
```

Listing 37. Mount NFS Export

```
[root@servera ~]#  mkdir /NFS_Share_Demo
[root@servera ~]#  mount -t nfs serverb:/ExportDemo /NFS_Share_Demo
[root@servera ~]# touch /NFS_Share_Demo/testfile
```

5.3.2. The nfsconf Tool

RHEL 8 bring a new tool **nfsconf** to manage NFS client and server configuration files. The **nfsconf** tool manages the **/etc/nfs.conf** file. Previous versions of this configuration files resided in **/etc/sysconfig/nfs**, but are now deprecated.

The new **nfsconf** tool should be used to ***set***, ***get***, or ***unset*** NFS configuration parameters.

Example 5. Demo of **nfsconf**

/etc/nfs.conf Configuration File

```
#
# This is a general configuration for the
# NFS daemons and tools
#
... Output Omitted ...
[nfsd]
# debug=0
# threads=8
# host=
# port=0
# grace-time=90
# lease-time=90
# tcp=y
# vers2=n
# vers3=y
# vers4=y
# vers4.0=y
# vers4.1=y
# vers4.2=y
# rdma=n

... Output Omitted ...
```

nfsconf *Tool Usage*

nfsconf --set section key value nfsconf --get section key

Listing 38. Changing **/etc/nfs.conf** *Values with* **nfsconf**

```
[root@servera ~]#  nfsconf --set nfsd vers4.2 n
[root@servera ~]# grep vers4.2 /etc/nfs.conf
vers4.2 = n
# vers4.2=y
[root@servera ~]#
```

Listing 39. Retrieving **/etc/nfs.conf** *Values with* **nfsconf**

```
[root@servera ~]#  nfsconf --get nfsd vers4.2
n
```

5.4. Automounting Network-Attached Storage

Objectives

- Describe the benefits of using the automounter.
- Automount NFS shares using direct and indirect maps, including wildcards.

5.4.1. Mounting NFS Shares with the Automounter

automounter is a service (**autofs**) that allows for on-demand mounting of NFS shares.

Automounter Benfits

- Standard users can use without needing **mount/umount** privileges
- Available to all users but subject to access control restrictions
- Configured on client-side, no server-side configuration needed
- Uses same options as mount command
- Supports direct and indirect mount-point mapping
- **autofs** creates/removes indirect mount mounts automatically
- **autofs** is managed like other system services

5.4.1.1. Creating an Automount

1. Install **autofs**

```
root@servera ~]# yum install autofs
```

2. Add a *master map* file to **/etc/auto.master.d** called *Auto_Mount_Demo.autofs*

```
[root@servera ~]# vim /etc/auto.master.d/Auto_Mount_Demo.autofs

/nfs_shares  /etc/autofs.demo ①
```

① Top-level directory that will allow indirect mapped mounts

3. Create mapping Files

```
[root@servera ~]# vim /etc/autofs.demo

NFS_Work  -rw,sync  serverb:/ExportDemo ①
```

① Relative Name/Path for the Directory to be Mounted under the **/nfs_shares** directory

4. Start/Enable the **automounter** service

```
[root@servera ~]# systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/lib/systemd/system/autofs.service.
```

5. Test that it is Working

```
[root@servera ~]# ls /nfs_shares/NFS_Work
testfile
```

5.4.1.2. Direct Maps

Direct maps are used to map NFS shares to an absolute path as a mount point.

See Student Guide

5.4.1.3. Indirect Wildcard Maps

When NFS servers export multiple subdirectories within a directory, automounter can be configured to allow indirect mapping using a single mapping entry containing wildcards.

The mount point (or key) is an asterisk character (*****), and the subdirectory on the source location is an ampersand character (**&**).

6. Managing Containers with the New Runtime

6.1. Deploying Containers with the New Container Runtime

6.1.1. The Podman Container Engine

RHEL8 includes he **container-tools** package module. New engine is **podman** replaces **docker** and **moby**. It also contains new tools **buildah** to build container images and **skopeo** to manage images on registries like **runc**. The new toolset allows building/running containers without daemons.

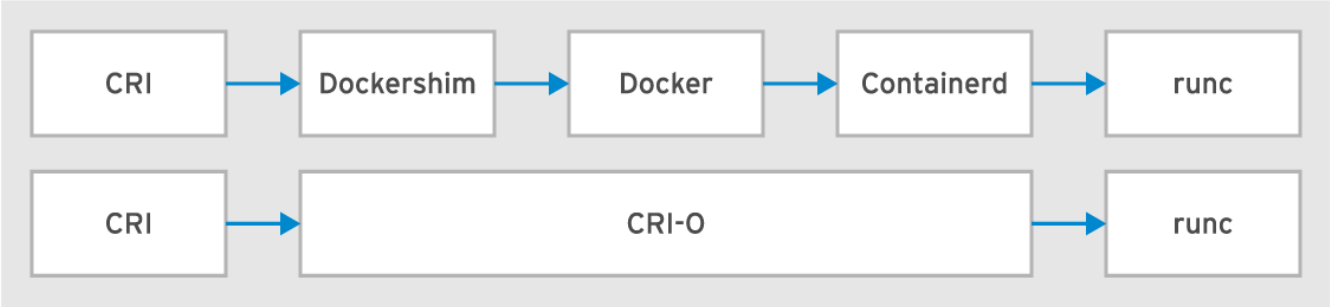

Figure 14. Docker to RHEL8 Container Runtime

Container Runtime Toolset

- Docker replaced with new container runtime
- New toolset supports OCI and reuse of third-party images
- Integrates with **audit** of Docker client-server model
- **container-tools** module provides new container runtime tools and engine.

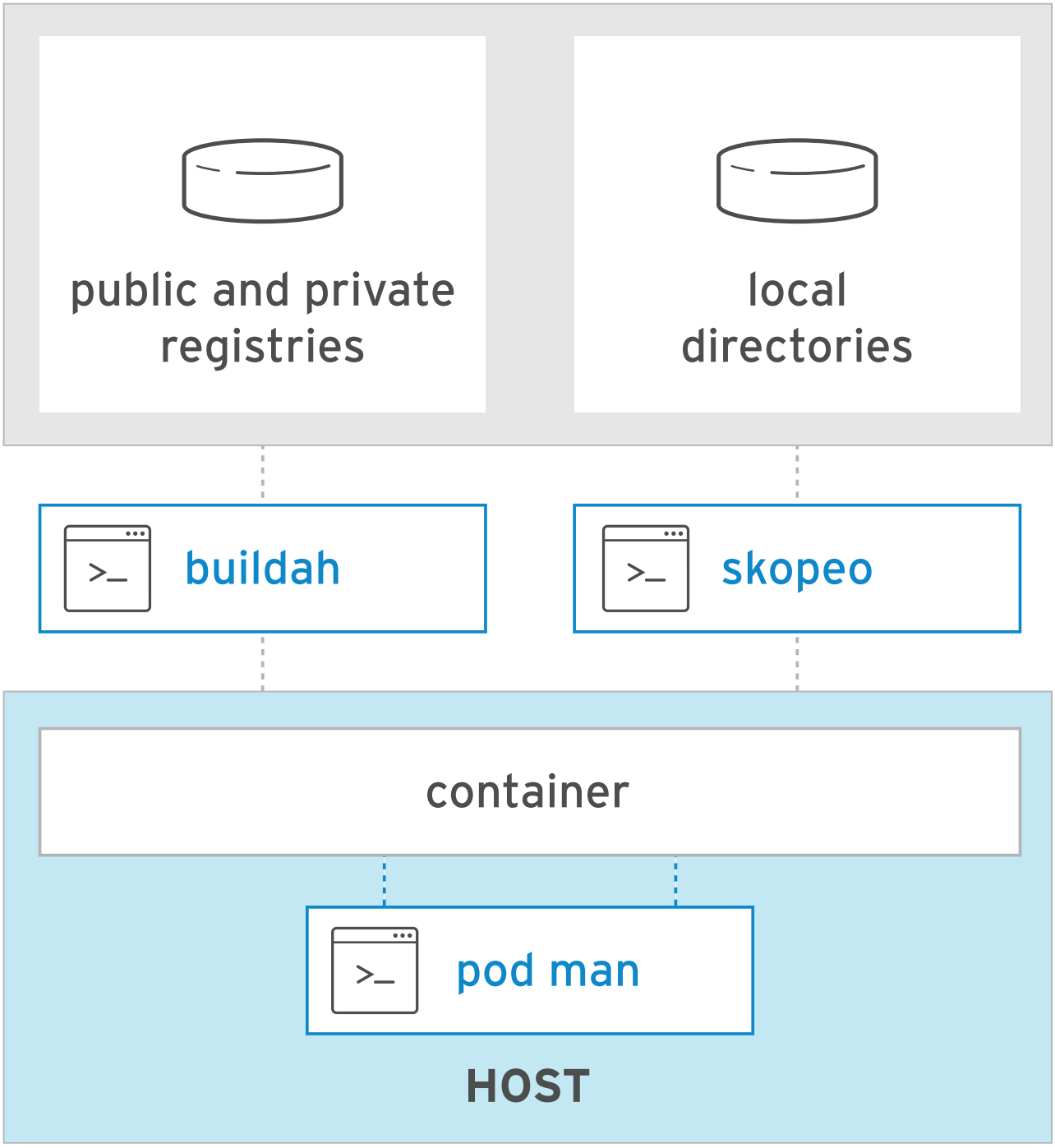

Figure 15. New Container Runtime

Describing new Container Runtime Tool

- The **podman** engine is daemonless and supporting container execution.
- **podman** syntax is similar to the docker command, supporting **Dockerfile** use
- **Buildah** builds container images, from scratch or a Dockerfile.
- Copy and inspect container images in registries with **Skopeo**
- **Skopeo** supports Docker and private registries, the Atomic registry, and local directories, including those which use OCI

RHEL8 includes **Pacemaker** containers with **podman** as a tech preview. Pacemaker supports execution of the container across multiple hosts.

Example 6. Using Container Tools

Listing 40. Installation of Container Tools

```
[root@servera ~]# yum module install container-tools
```

Listing 41. Creating a Custom Container

```
[root@servera ~]# buildah from scratch
working-container
```

Listing 42. Naming and Inspecting a Custom Container

```
[root@servera ~]# buildah config --label name=My-Container working-container
[root@servera ~]# buildah inspect working-container
```

Listing 43. Installing Packages on Working Container

```
[root@servera ~]#  buildah mount working-container ①


[root@servera ~]# yumdownloader --destdir=/tmp redhat-release-server ②

[root@servera ~]# rpm -ivh --root
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged /tmp/redhat-release-8.0-
0.39.el8.x86_64.rpm ③

[root@servera ~]# cp  /etc/yum.repos.d/rhel_dvd.repo
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged/etc/yum.repos.d/ ④

[root@servera ~]# yum install --installroot
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged httpd ⑤


[root@servera ~]# echo "This is a custom webserver container for me" >>
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged/var/www/html/index.html ⑥

[root@servera ~]#  yum install --installroot
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged httpd-manual ⑦

[root@servera ~]# buildah config --cmd "/usr/sbin/httpd -DFOREGROUND" working-container ⑧

[root@servera ~]# buildah config --port 80/tcp working-container ⑨

[root@servera ~]# yum clean all --installroot
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged ⑩

[root@servera ~]#  buildah unmount working-container ⑪

[root@servera ~]# buildah commit working-container my-container-image ⑫

[root@servera ~]# buildah images ⑬
```

① Mount container image filesystem for modification

② Download Red Hat Release RPM for installation

③ Install Red Hat Release RPM

④ Create repository for container image so files can be installed

⑤ Install the HTTP package for a webserver

⑥ Create an **index.html** file for the webserver

⑦ Install the Apache manual for reference documentation

⑧ Configure webserver to run

⑨ Configure and open port **80** for the **TCP** protocol for the container

⑩ Clean up yum data to minimize required disk space

⑪ Unmount the container image filesystem

⑫ Commit the container image

⑬ List container images

Listing 44. Testing the Container Image

```
[root@servera ~]# podman run -d -p 8080:80 localhost/my-container-image

[root@servera ~]# curl localhost:8080
This is a custom webserver container for me

[root@servera ~]# curl http://localhost:8080/manual/
```

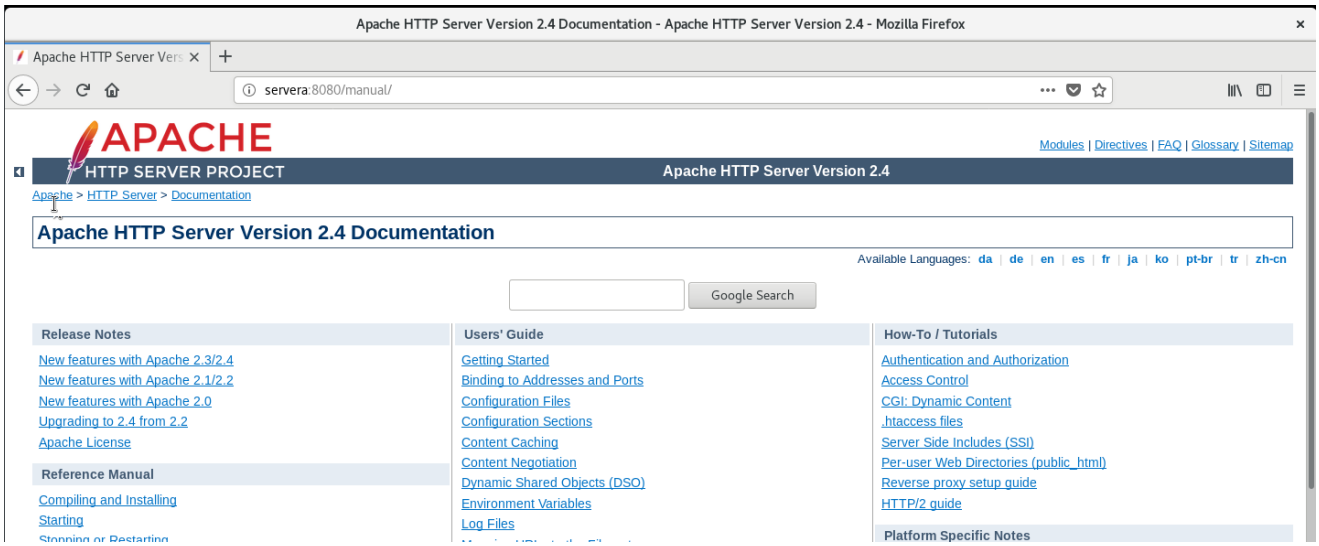

Figure 16. Testing Container

Listing 45. Stopping and Cleanup of Image

```
[root@servera ~]# podman list ①
CONTAINER ID  IMAGE                                COMMAND               CREATED        STATUS            PORTS                 NAMES
9bd572633953  localhost/my-container-image:latest  /usr/sbin/httpd -...  2 seconds ago  Up 1 second ago  0.0.0.0:8080->80/tcp
cranky_stonebraker

[root@servera ~]# podman stop 9bd572633953 ②
9bd572633953276ac75417db3ac8e70875a0f2713e8cdfd32253fe343d06153d

[root@servera ~]# podman stop -a ③

[root@servera ~]# podman rm cranky_stonebraker ④

[root@servera ~]# podman rm 9bd572633953276ac75417db3ac8e70875a0f2713e8cdfd32253fe343d06153d ⑤

[root@servera ~]# podman rmi localhost/my-container-image ⑥

[root@servera ~]# buildah delete working-container ⑦
```

① Listing Running Containers

② Stopping Single Container by ID

③ Stopping All Running Containers

④ Remove Container by Name

⑤ Remove Container by ID

⑥ Removing Container Image from Registry

⑦ Delete Working Container from System

7. Implementing Enhanced Networking Features

Objectives

- Explain the new nftables firewall back-end design, advantages and configuration.
- Explain how the NetworkManager service has become an integral and mandatory component in modern network management, able to configure complex, layered network interfaces and components.

7.1. Managing Server Firewalls in RHEL 8

Objective

- Explain **nftables** firewall

7.1.1. Introducing Nftables

nftables *with* **firewalld**

- **firewalld** uses **nftables** as its back-end.
- The **nft** command replaces the **iptables**, **ip6tables**, **arptables**, and **ebtables** commands.
- **firewalld** is the recommended way to manage the firewall
- **iptables** commands are links to the **xtables-nft-multi** command, which accepts iptables syntax but creates **nftables** rules instead.

It is recommended to continue using **firewalld** to manage firewalls as the syntax remains the same even for the Rich Rules. It should be noted that **firewalld** can also be managed via **Cockpit**.

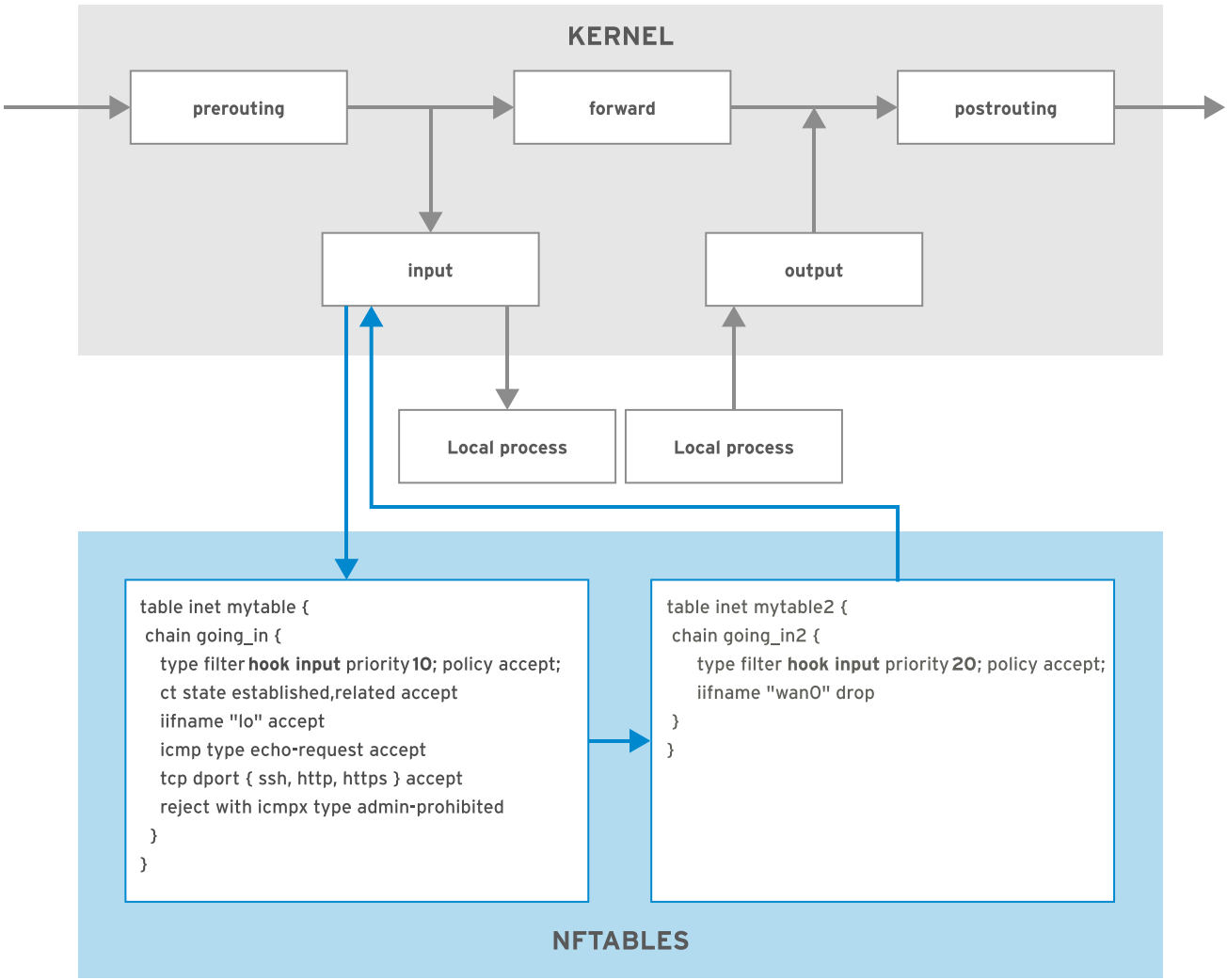

Figure 17. **nftables** *Chains Process*

Example 7. Firewall Management

Listing 46. Listing **nftables**

```
[root@servera ~]# nft list tables
```

Listing 47. **firewalld** *and* **nftables**

```
[root@servera ~]# nft list ruleset | grep http

[root@servera ~]# firewall-cmd --add-service=http

[root@servera ~]# nft list ruleset | grep http
```

It is possible to use **iptables** with RHEL8, but you must mask/disable the **firewalld** and **nftables** services as well as install the **iptables-services** package and enable it.

7.2. Configuring Server Networking with NetworkManager

8. Adapting to Virtualization Improvements

8.1. Configuring Virtual Machines

Appendix A: System Security Policy and Compliance

System security and compliance is a primary concern for people when thinking about the protection of systems and integrity of data. This set of hands-on procedures will focus on obtaining the content and necessary packages to perform a basic scan and remediate the system based on scan results. As part of the lab, you will be customizing your own SCAP content for a scan, view the results, and generate an Ansible playbook based on the failed results.

A.1. Customizing SCAP Content

Red Hat includes the SCAP Workbench application as a GUI application which allows scanning, customizing, and saving SCAP scans and results. The SCAP Workbench can be used to perform scans of remote systems over an SSH connection or it can be utilized to scan the local system. Since the SCAP Workbench is a GUI application, it must run on a system with X-Windows installed.

Servers to Configure

- workstation

Packages to Install

- scap-workbench

Since we are using an application on a VM that requires a GUI, we can use X11 forwarding with the SSH connection by specifying a **-X** on the command line.

Step 1 - SSH to Workstation

The fist step is to connect to the workstation and forward X11 traffic back to the local system.

Example 8. Connecting to the Workstation VM

Listing 48. Connecting to Workstation Using SSH

```
# ssh root@workstation -X
```

Step 2 - Install Packages

After connecting to the Workstation VM, you will need to install the SCAP Workbench and its dependencies.

Example 9. Installing SCAP Workbench

Listing 49. Using Yum to Install SCAP Workbench

```
# yum install scap-workbench

Loaded plugins: langpacks, search-disabled-repos
Resolving Dependencies
--> Running transaction check
---> Package scap-workbench.x86_64 0:1.1.6-1.el7 will be installed
--> Processing Dependency: openscap-utils >= 1.2.0 for package: scap-workbench-1.1.6-1.el7.x86_64
--> Processing Dependency: scap-security-guide for package: scap-workbench-1.1.6-1.el7.x86_64
--> Running transaction check
---> Package openscap-utils.x86_64 0:1.2.16-8.el7_5 will be installed
--> Processing Dependency: openscap-containers = 1.2.16-8.el7_5 for package: openscap-utils-1.2.16-8.el7_5.x86_64
---> Package scap-security-guide.noarch 0:0.1.36-9.el7_5 will be installed
--> Processing Dependency: openscap-scanner >= 1.2.5 for package: scap-security-guide-0.1.36-9.el7_5.noarch
--> Running transaction check
---> Package openscap-containers.noarch 0:1.2.16-8.el7_5 will be installed
---> Package openscap-scanner.x86_64 0:1.2.16-8.el7_5 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch      Version             Repository           Size
================================================================================
Installing:
 scap-workbench         x86_64    1.1.6-1.el7         rhel--server-dvd    1.8 M
Installing for dependencies:
 openscap-containers    noarch    1.2.16-8.el7_5      rhel_updates         27 k
 openscap-scanner       x86_64    1.2.16-8.el7_5      rhel_updates         61 k
 openscap-utils         x86_64    1.2.16-8.el7_5      rhel_updates         27 k
 scap-security-guide    noarch    0.1.36-9.el7_5      rhel_updates        2.6 M

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

Total download size: 4.5 M
Installed size: 64 M
Is this ok [y/d/N]:
```

Some things might already be installed for you, if SCAP Workbench is already installed, please move on to the next step. Also note the dependencies for SCAP Workbench as they are automatically installed.

Step 3 - Launching SCAP Workbench

In order to run a scan or customize SCAP content, you will need to launch the SCAP Workbench application.

You **must** use SCAP Workbench from a GUI, so it will need to run either locally or through an SSH connection with X11 forwarded.

Example 10. Launching SCAP Workbench

Listing 50. Using SCAP Workbench

```
# scap-workbench
```

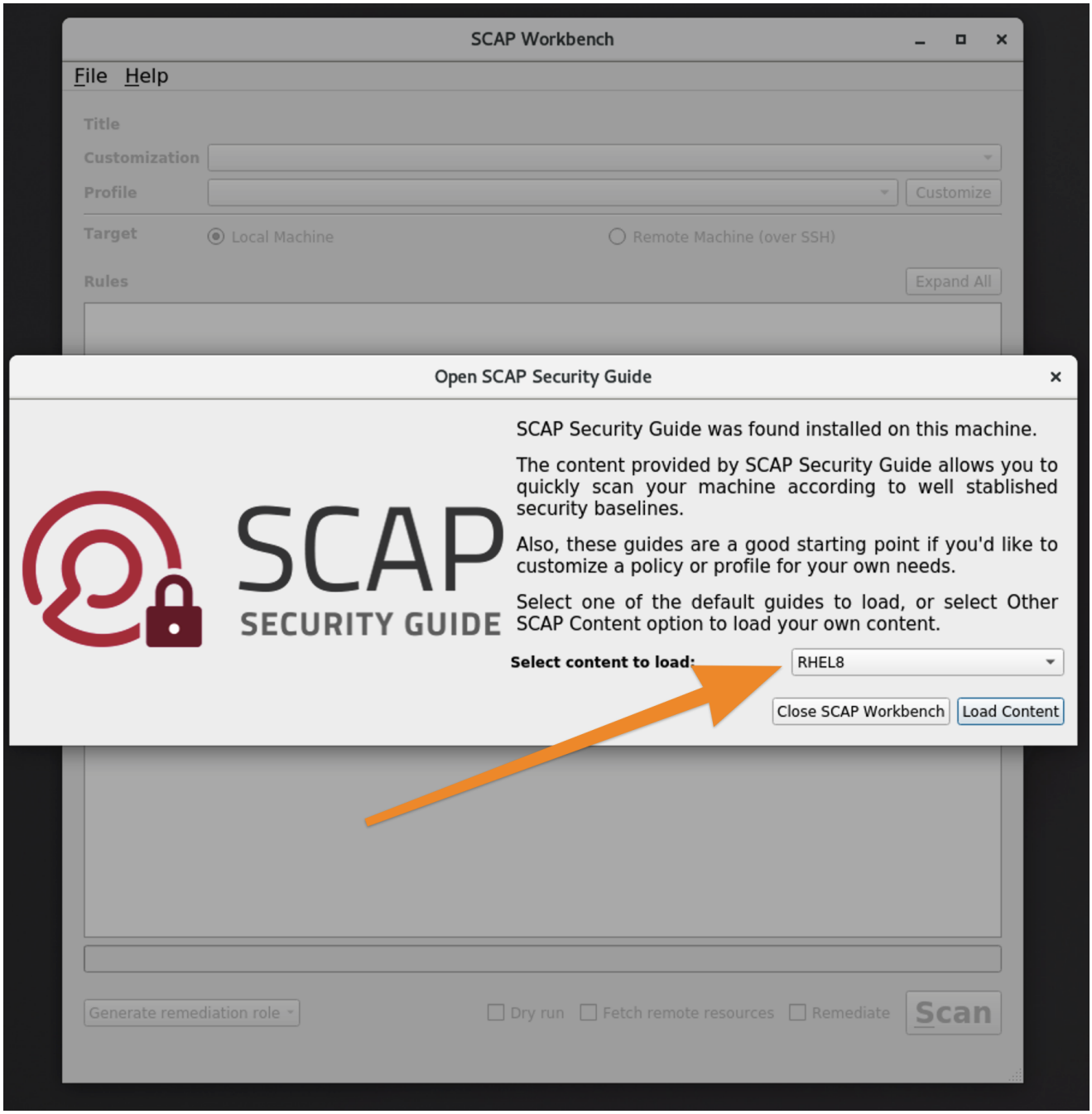

Figure 18. SCAP Workbench Startup

Step 4 - Creating Custom Content

Once SCAP Workbench has been launched, select the content to load. For this lab, we will be using the RHEL7 content.

Example 11. Creating Custom Content

1. For **Select content to load:** select "**RHEL8**", then click "**Load Content**"
2. Select the **Profile** you want to use to start customization

For this example, we will use the **OSPP - Protection Profile for General Purpose Operating Systems** Baseline

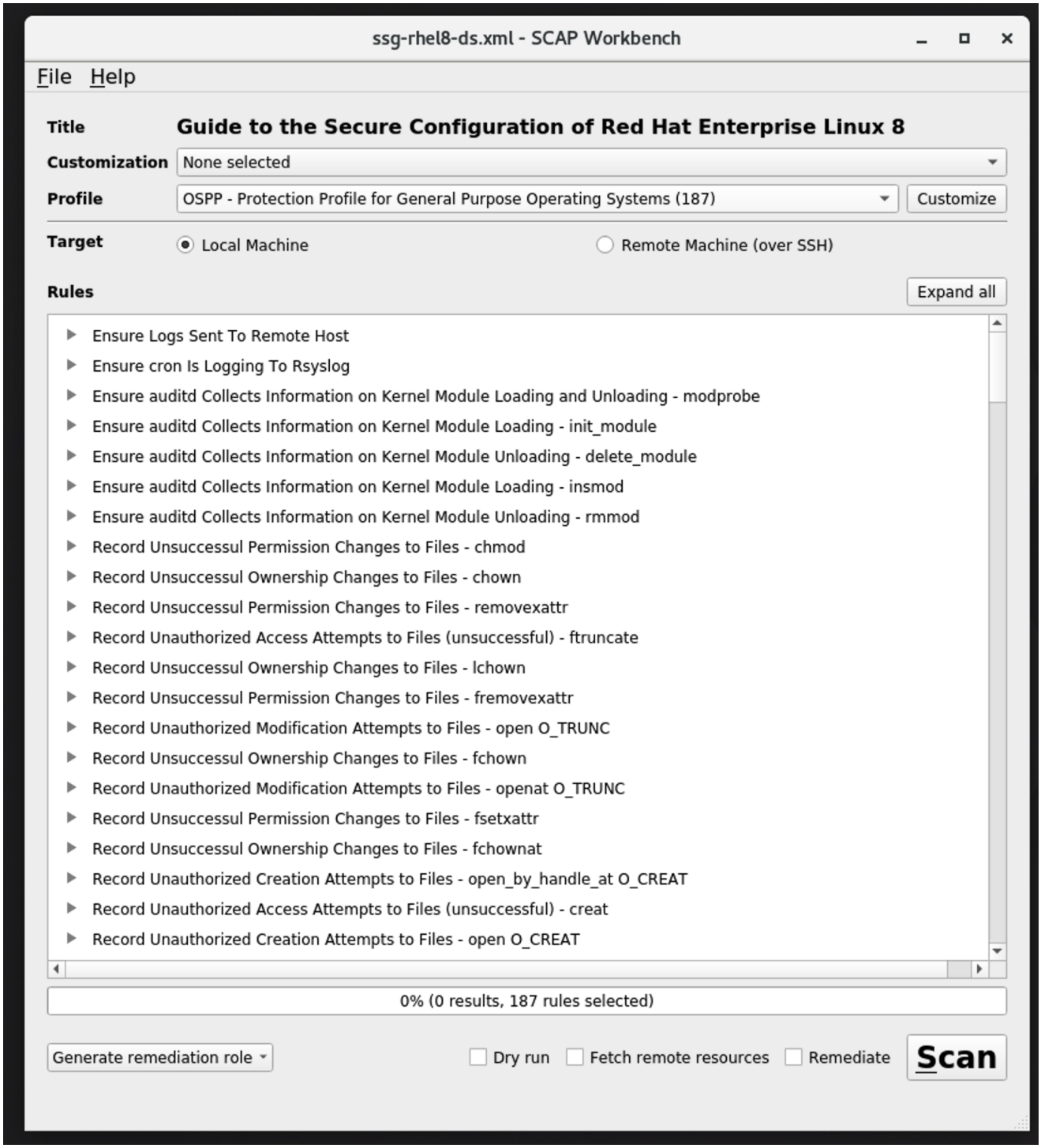

Figure 19. SCAP Workbench OSPP Profile

3. Click "**Customize**" to create custom SCAP content based on the chosen profile, and give it a name.

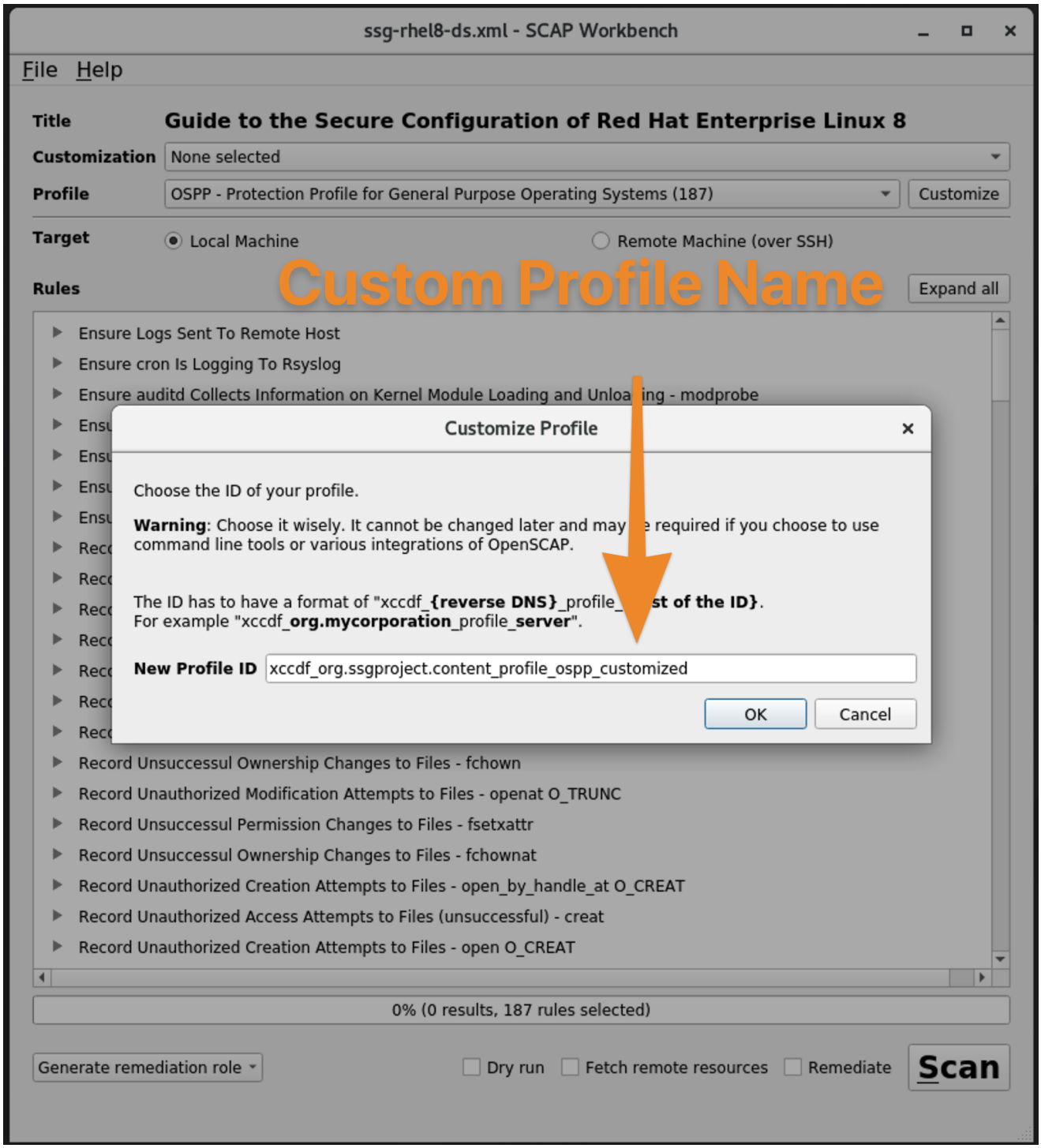

Figure 20. SCAP Custom OSPP Profile Creation

The name for this is: **xccdf_org.ssgproject.content_profile_ospp_customized**

4. Click "Deselect All" so that you can select the items you wish to include in your custom scan profile. **NOTE:** we are doing this to also limit it to a few checks for the example.

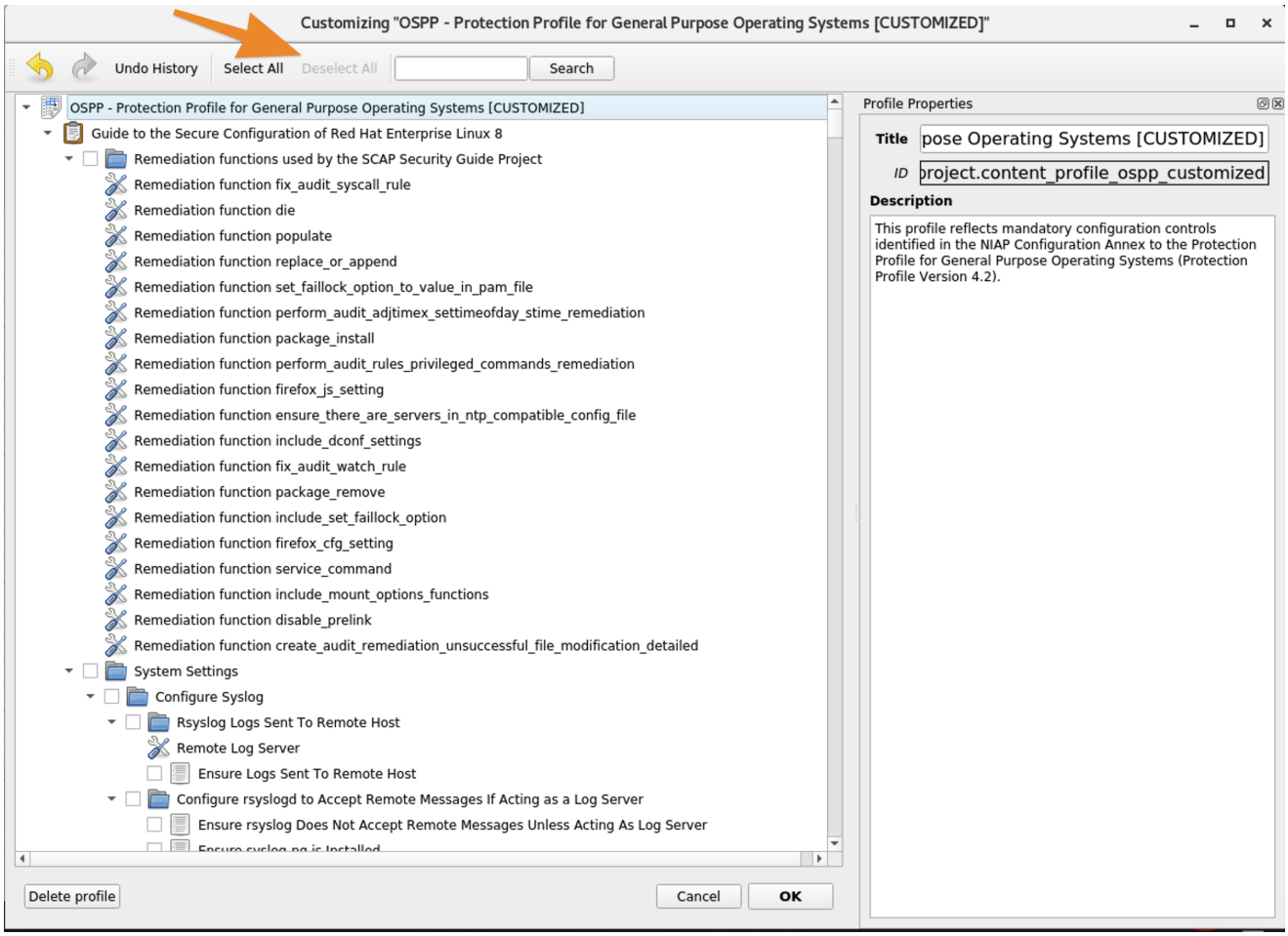

Figure 21. SCAP Custom Profile Selections

For this lab, we will be setting the minimum password length and PAM Password quality settings

5. Search for Password to set **minimum password length** and set the values in **login.defs**. Check **Set Password Minimum Length in login.defs** and click on the **minimum password legnth** and set the value to **18**

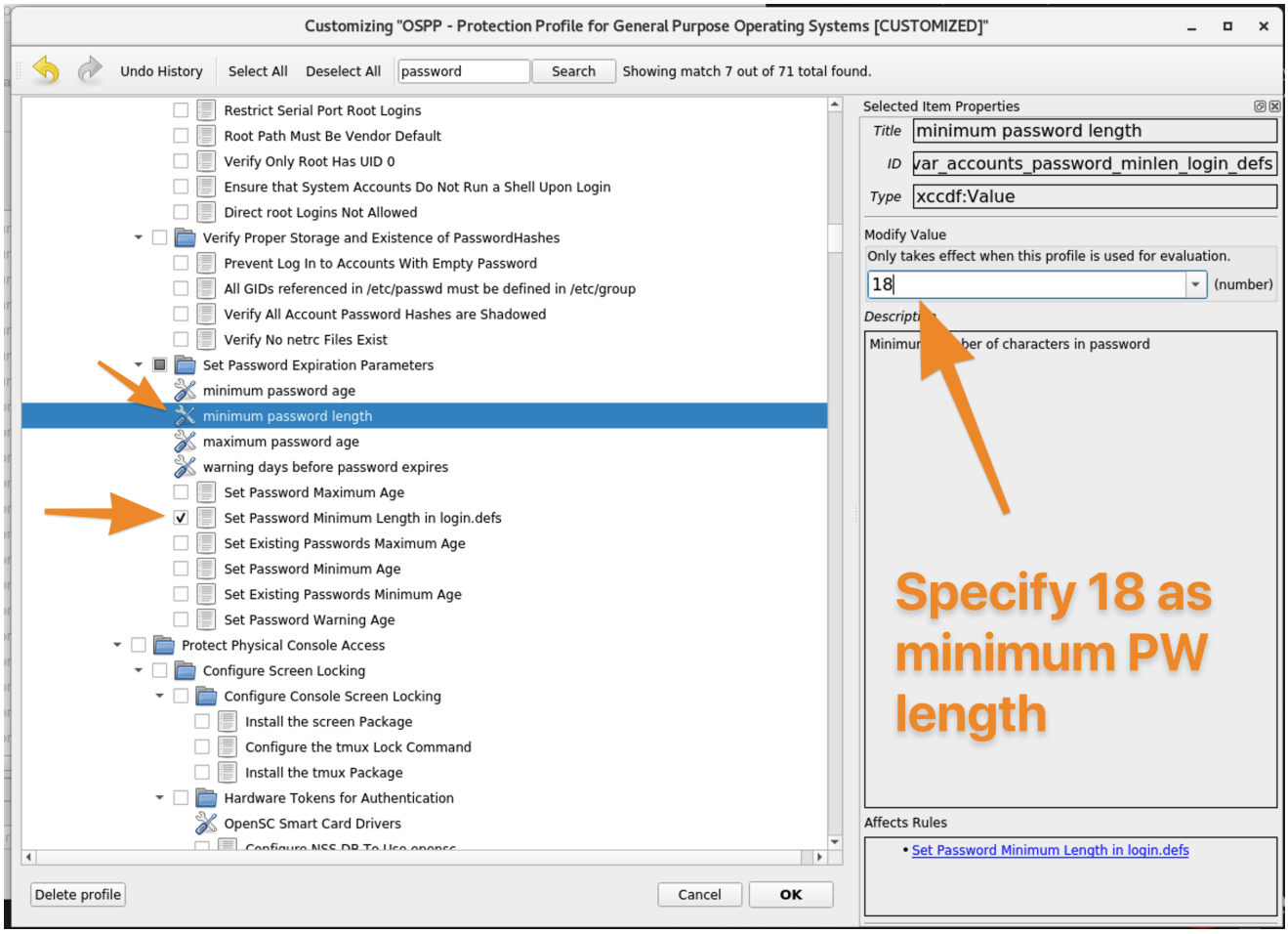

Figure 22. SCAP Custom Profile Password Settings for Login.Defs

6. Set password quality requirements with PAM. Search for the minlen and set it to **18**. Also, place a checkbox in **Set Password Quality Requirements with pam_quality**. Then click "OK"

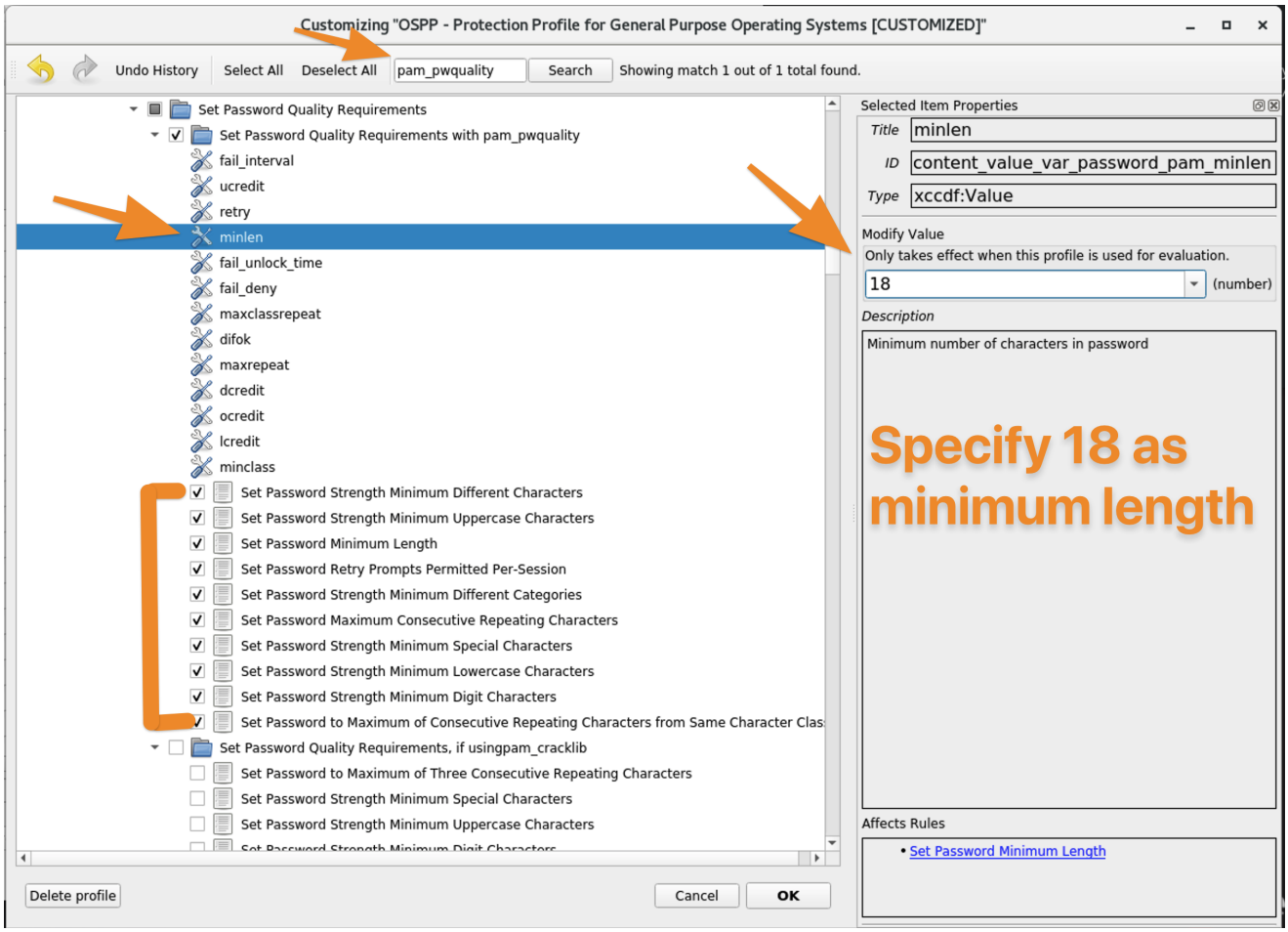

Figure 23. SCAP Custom Profile PAM Quality Requirements

7. At this point, we have taken the default settings from the OSPP profile with only the tailored pieces that we selected. The next step is to click "File ⇒ Save Customization Only" to save the custom content

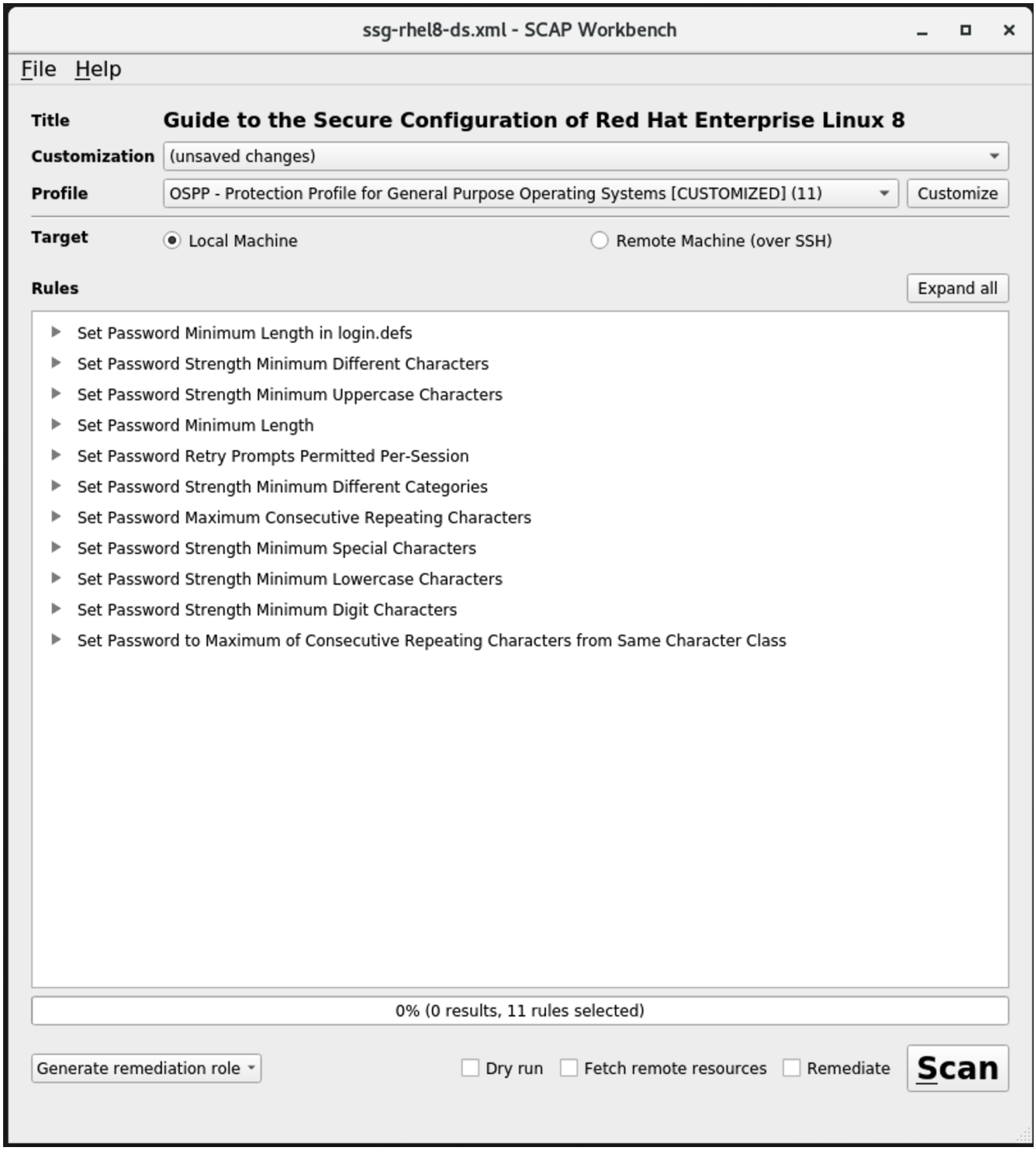

Figure 24. SCAP Custom Profile Selected Settings View

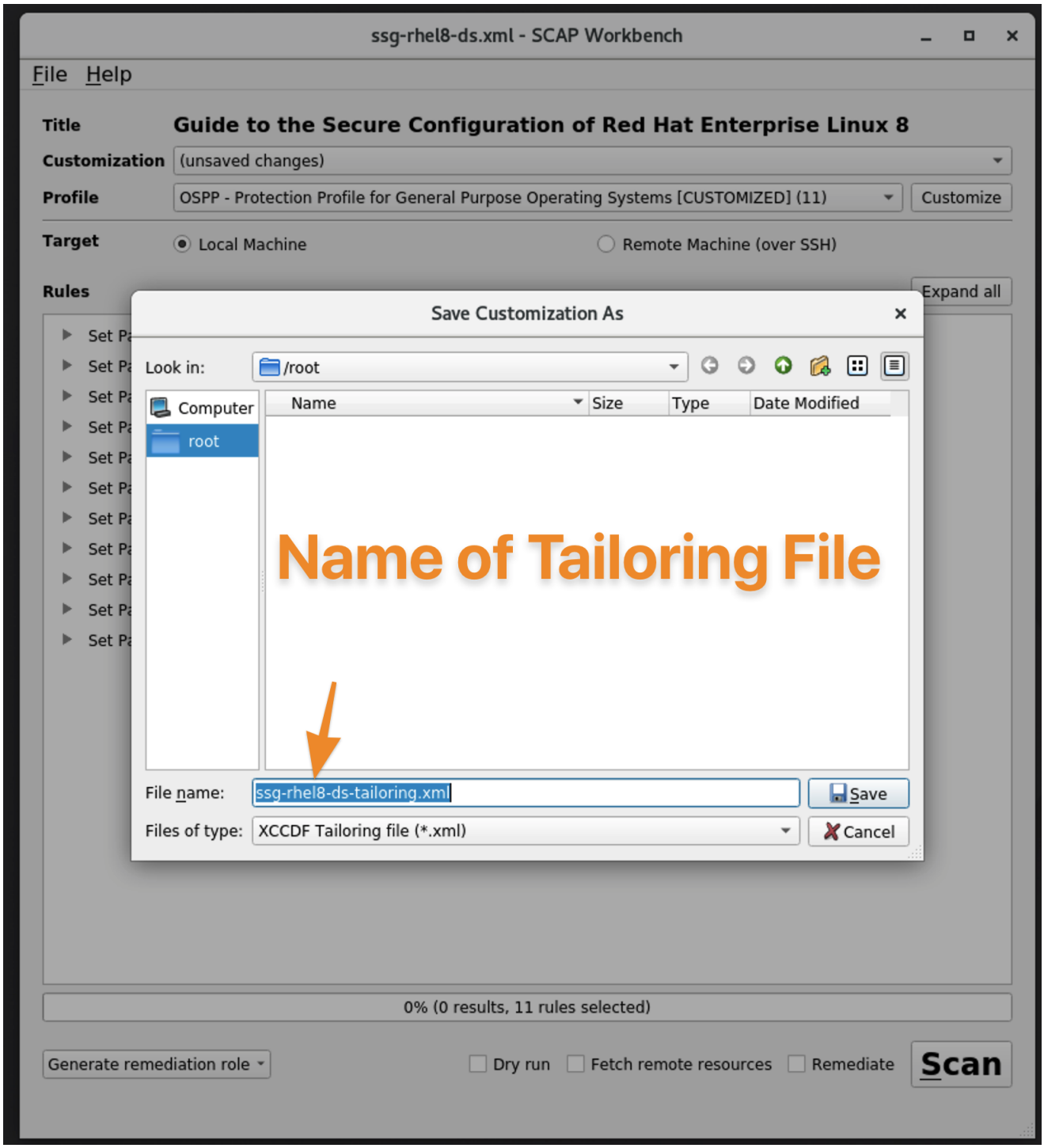

Figure 25. SCAP Custom Profile Creation Saving

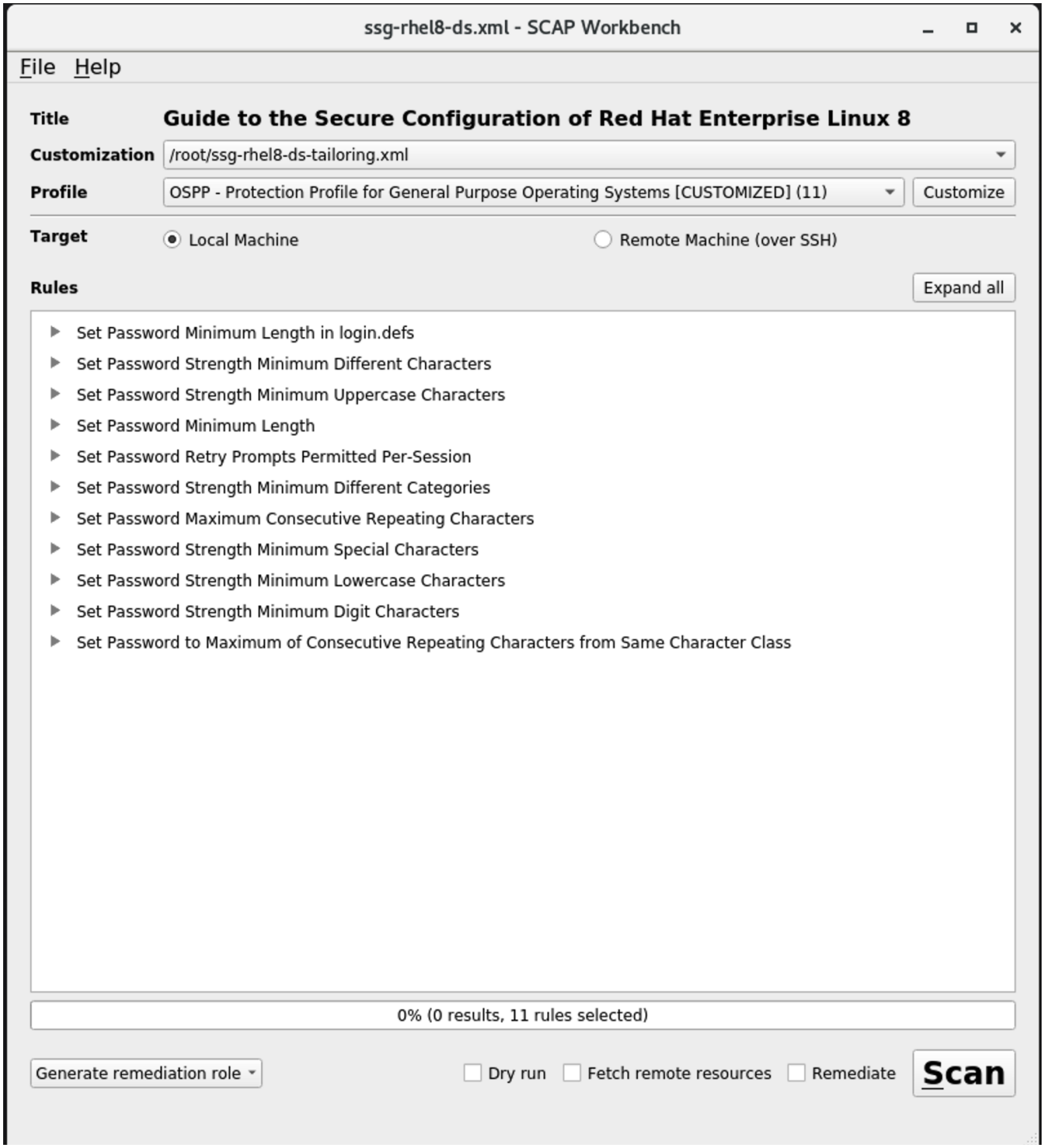

Figure 26. SCAP Custom Profile Final View

8. Copy the custom tailoring file to the server(s) being scanned. In this case, we will want to copy the file to **servera**

Listing 51. Copy custom content

```
[root@workstation ~]# scp ssg-rhel8-ds-tailoring.xml root@servera:
ssg-rhel8-ds-tailoring.xml                    100%   28KB  10.7MB/s   00:00
[root@workstation ~]#
```

A.2. Running a SCAP Scan with Custom Content

Servers to Configure

- servera

Packages to Install

- openscap-scanner
- scap-security-guide

Step 1 - SSH to serverc

The fist step is to connect to the server.

Example 12. Connecting to the **servera** *VM*

Listing 52. Connecting to **servera** *Using SSH*

```
# ssh root@servera
```

Step 2 - Install packages on servera

The second step is to install software on the server.

Example 13. Install software on **servera**

Listing 53. Installing Software on **servera**

```
[root@servera ~]# yum install scap-security-guide
Last metadata expiration check: 0:47:44 ago on Thu 16 Apr 2020 08:26:15 AM EDT.
Dependencies resolved.
================================================================================
 Package          Arch   Version       Repository                          Size
================================================================================
Installing:
 scap-security-guide
                  noarch 0.1.42-11.el8 rhel-8.0-for-x86_64-appstream-rpms 3.4 M
Installing dependencies:
 openscap         x86_64 1.3.0-7.el8   rhel-8.0-for-x86_64-appstream-rpms 3.3 M
 openscap-scanner x86_64 1.3.0-7.el8   rhel-8.0-for-x86_64-appstream-rpms  66 k
 xml-common       noarch 0.6.3-50.el8  rhel-8.0-for-x86_64-baseos-rpms     39 k

Transaction Summary
================================================================================
Install  4 Packages

Total download size: 6.9 M
Installed size: 132 M
Is this ok [y/N]: y


... output omitted ...

  Verifying        : openscap-1.3.0-7.el8.x86_64                              1/4
  Verifying        : openscap-scanner-1.3.0-7.el8.x86_64                      2/4
  Verifying        : scap-security-guide-0.1.42-11.el8.noarch                 3/4
  Verifying        : xml-common-0.6.3-50.el8.noarch                           4/4

Installed:
  scap-security-guide-0.1.42-11.el8.noarch    openscap-1.3.0-7.el8.x86_64
  openscap-scanner-1.3.0-7.el8.x86_64         xml-common-0.6.3-50.el8.noarch

Complete!
```

Learning about SCAP Commands

The SSG man page is a very good source of information for usage of the **oscap** tool as well as provides examples of how to use the SCAP SSG Guide profiles itself.

Listing 54. Looking at **SCAP Security Guide (SSG)** *Man Page*

```
# man scap-security-guide

scap-security-guide(8)      System Manager's Manual     scap-security-guide(8)

NAME
       SCAP  Security Guide - Delivers security guidance, baselines, and asso-
       ciated validation mechanisms utilizing the Security Content  Automation
       Protocol (SCAP).


... output omitted ...

EXAMPLES
       To  scan  your  system utilizing the OpenSCAP utility against the ospp-
       rhel7 profile:

       oscap xccdf eval --profile  ospp-rhel7  --results  /tmp/`hostname`-ssg-
       results.xml  --report  /tmp/`hostname`-ssg-results.html  --oval-results
       /usr/share/xml/scap/ssg/content/ssg-rhel7-xccdf.xml
```

Listing 55. Looking at ***oscap*** *Man Page*

```
# man oscap

OSCAP(8)                System Administration Utilities               OSCAP(8)

NAME
       oscap - OpenSCAP command line tool

SYNOPSIS
       oscap  [general-options]  module operation [operation-options-and-argu-
       ments]

DESCRIPTION
       oscap is Security Content Automation Protocol (SCAP) toolkit  based  on
       OpenSCAP  library.  It  provides  various  functions for different SCAP
       specifications (modules).

       OpenSCAP tool claims to provide capabilities of Authenticated  Configu-
       ration  Scanner  and  Authenticated Vulnerability Scanner as defined by
       The National Institute of Standards and Technology.

... output omitted ...

EXAMPLES
       Evaluate XCCDF content using CPE dictionary and produce html report. In
       this  case  we  use  United  States  Government  Configuration Baseline
       (USGCB) for Red Hat Enterprise Linux 5 Desktop.

               oscap xccdf eval --fetch-remote-resources --oval-results \
                       --profile united_states_government_configuration_baseline
 \
                       --report usgcb-rhel5desktop.report.html \
                       --results usgcb-rhel5desktop-xccdf.xml.result.xml \
                       --cpe usgcb-rhel5desktop-cpe-dictionary.xml \
                       usgcb-rhel5desktop-xccdf.xml
```

Step 3 - Running *oscap* scan

We will run the **oscap** utility to generate a report and a results file that can be sent back to the **workstation** system so that we can create an Ansible playbook for remediation and view the results of the report.

Be very careful about the name of the profile as this was selected during the creation of the custom profile/tailoring file portion when doing SCAP Workbench customizations.

Example 14. Scanning **servera**

Listing 56. Using **oscap** *and the tailoring profile to scan* **servera**

```
# [root@servera ~]# oscap xccdf eval \
--profile xccdf_org.ssgproject.content_profile_ospp_customized \
--tailoring-file ssg-rhel8-ds-tailoring.xml \
--results custom_scan_results.xml \
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml

Title   Set Password Minimum Length in login.defs
Rule    xccdf_org.ssgproject.content_rule_accounts_password_minlen_login_defs
Ident   CCE-80652-1
Result  fail

Title   Set Password Strength Minimum Different Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_difok
Ident   CCE-80654-7
Result  fail

Title   Set Password Strength Minimum Uppercase Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_ucredit
Ident   CCE-80665-3
Result  fail

Title   Set Password Minimum Length
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_minlen
Ident   CCE-80656-2
Result  fail

Title   Set Password Retry Prompts Permitted Per-Session
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_retry
Ident   CCE-80664-6
Result  fail

Title   Set Password Strength Minimum Different Categories
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_minclass
Result  fail

Title   Set Password Maximum Consecutive Repeating Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_maxrepeat
Result  fail

Title   Set Password Strength Minimum Special Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_ocredit
Ident   CCE-80663-8
Result  fail

Title   Set Password Strength Minimum Lowercase Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_lcredit
Ident   CCE-80655-4
Result  fail

Title   Set Password Strength Minimum Digit Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_dcredit
Ident   CCE-80653-9
Result  fail

Title   Set Password to Maximum of Consecutive Repeating Characters from Same Character Class
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_maxclassrepeat
Result  fail
```

Getting Custom Profile Name from Tailoring File

If you need to locate the profile used for the custom scanning content from the tailoring file, you can search for it with **grep**.

```
[root@servera ~]#  grep "Profile id" ssg-rhel8-ds-tailoring.xml
  <xccdf:Profile id="xccdf_org.ssgproject.content_profile_ospp_customized" extends
="xccdf_org.ssgproject.content_profile_ospp">
```

Step 4 - Creating a Results Report

You can create a results report file from the results file so you have a nice HTML file that is easy to ready with the results from the SCAP scan.

Example 15. Creating a SCAP Report from a Results File

Listing 57. Generating a Report

```
[root@servera ~]# oscap xccdf generate report \
custom_scan_results.xml > Custom_Scan_Report.html
```

Combining Steps 3 & 4

It is possible to perform a custom content scan which will generate the results file and the report for transfer back to the workstation for review.

Need to Specify

- **--results**
- **--report**

Listing 58. Creating a Results File and Report During Custom Content Scan

```
[root@servera ~]# oscap xccdf eval \
--profile xccdf_org.ssgproject.content_profile_ospp_customized \
--tailoring-file ssg-rhel8-ds-tailoring.xml \
--results custom_scan_results_2.xml \
--report Custom_Scan_Report_2.html \
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml

Title   Set Password Minimum Length in login.defs
Rule    xccdf_org.ssgproject.content_rule_accounts_password_minlen_login_defs
Ident   CCE-80652-1
Result  fail

Title   Set Password Strength Minimum Different Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_difok
Ident   CCE-80654-7
Result  fail

Title   Set Password Strength Minimum Uppercase Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_ucredit
Ident   CCE-80665-3
Result  fail

Title   Set Password Minimum Length
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_minlen
Ident   CCE-80656-2
Result  fail

Title   Set Password Retry Prompts Permitted Per-Session
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_retry
Ident   CCE-80664-6
Result  fail

Title   Set Password Strength Minimum Different Categories
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_minclass
Result  fail

Title   Set Password Maximum Consecutive Repeating Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_maxrepeat
Result  fail

Title   Set Password Strength Minimum Special Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_ocredit
Ident   CCE-80663-8
Result  fail

Title   Set Password Strength Minimum Lowercase Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_lcredit
Ident   CCE-80655-4
Result  fail

Title   Set Password Strength Minimum Digit Characters
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_dcredit
Ident   CCE-80653-9
Result  fail

Title   Set Password to Maximum of Consecutive Repeating Characters from Same Character Class
Rule    xccdf_org.ssgproject.content_rule_accounts_password_pam_maxclassrepeat
Result  fail
```

Step 5 - Transferring Results File and Report to Workstation

After you have the results files and the report, you should transfer it to your graphical workstation (**workstation**) for further analysis.

Example 16. Transferring Results

Listing 59. Transferring the Results and Report Files

```
[root@servera ~]# scp *.xml *.html root@workstation:
The authenticity of host 'workstation (<no hostip for proxy command>)' can't be established.
ECDSA key fingerprint is SHA256:pOQ1OJmyF2PFI+jxyFoOSCfi+1oWNsUruy2DZNjg+N0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'workstation' (ECDSA) to the list of known hosts.
root@workstation's password:
custom_scan_results_2.xml                     100% 4086KB  32.4MB/s   00:00
custom_scan_results.xml                       100% 4086KB  60.0MB/s   00:00
ssg-rhel8-ds-tailoring.xml                    100%   28KB  16.2MB/s   00:00
Custom_Scan_Report_2.html                     100%  332KB  44.3MB/s   00:00
Custom_Scan_Report.html                       100%  332KB  37.6MB/s   00:00
[root@servera ~]#
```

Step 6 - Viewing the SCAP scan report

After you have transferred the results file to **workstation** you can open the HTML report in a web browser. In this case we will use ***firefox*** to open the file.

Example 17. Viewing the SCAP Report

Listing 60. Opening the SCAP HTML Report with Firefox

```
[root@workstation ~]# firefox Custom_Scan_Report.html
```

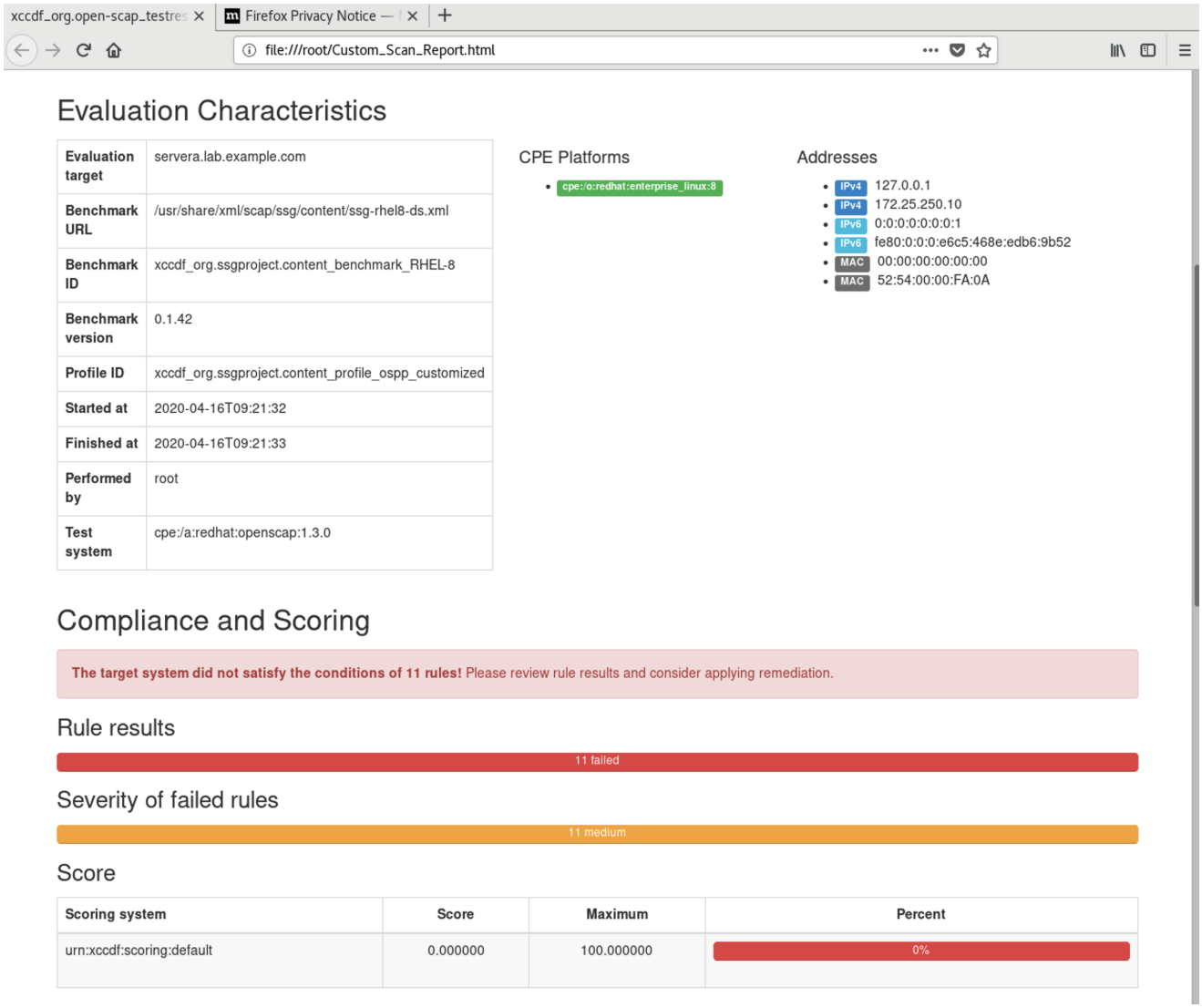

Figure 27. SCAP Scan Results Report in Firefox

Firefox may not open the file based on SELinux context triggers. In order to get around this you can use the command prompt and do **setenforce 0** to allow you to open the report.

A.3. Creating an Ansible Remediation Playbook Based on SCAP Scan Results

The OpenSCAP project and content created by Red Hat can automatically remediate findings from OpenSCAP scans. The findings can be remediated in many ways (**BASH**, **Ansible**, etc.). While things are mostly complete, there are some automated remediations that have not yet been developed.

There are multiple automatic remediation methods developed, but at this time, there isn't a script to fix everything.

Servers to Configure

- servera

We will continue to use **workstation** as our master SCAP system as it should have Ansible and SCAP Workbench installed.

Step 1 - Creating an Ansible Playbook from Results

The first step will be to generate an Ansible playbook from the SCAP scan results for system remediation.

Example 18. Generating Ansible Playbook

Listing 61. Ansible Playbook Generation

```
[root@workstation ~]# oscap xccdf generate fix \
--profile xccdf_org.ssgproject.content_profile_ospp_customized \
--tailoring-file ssg-rhel8-ds-tailoring.xml \
--fix-type ansible \
--result-id "" \
custom_scan_results.xml > Custom_Scan_Fix.yml
```

Viewing Remediation Playbook

It is also a good idea to view the created playbook for the system prior to running it.

```
[root@workstation ~]# cat Custom_Scan_Fix.yml
---
#
# Ansible remediation role for the results of evaluation of profile
xccdf_org.ssgproject.content_profile_ospp_customized
# XCCDF Version:  unknown
#
# Evaluation Start Time:  2020-04-16T09:21:32
# Evaluation End Time:  2020-04-16T09:21:33
#
# This file was generated by OpenSCAP 1.3.0 using:
#   $ oscap xccdf generate fix --result-id xccdf_org.open-
scap_testresult_xccdf_org.ssgproject.content_profile_ospp_customized --template urn:xccdf:fix:script:ansible
xccdf-results.xml
#
# This script is generated from the results of a profile evaluation.
```

```
# It attempts to remediate all issues from the selected rules that failed the test.
#
# How to apply this remediation role:
# $ ansible-playbook -i "localhost," -c local playbook.yml
# $ ansible-playbook -i "192.168.1.155," playbook.yml
# $ ansible-playbook -i inventory.ini playbook.yml
#
################################################################################


 - hosts: all
   vars:
      var_accounts_password_minlen_login_defs: !!str 18
      var_password_pam_difok: !!str 8
      var_password_pam_ucredit: !!str -1
      var_password_pam_minlen: !!str 18
      var_password_pam_retry: !!str 3
      var_password_pam_minclass: !!str 3
      var_password_pam_maxrepeat: !!str 3
      var_password_pam_ocredit: !!str -1
      var_password_pam_lcredit: !!str -1
      var_password_pam_dcredit: !!str -1
      var_password_pam_maxclassrepeat: !!str 4
   tasks:


    - name: "Set Password Minimum Length in login.defs"
      lineinfile:
        dest: /etc/login.defs
        regexp: "^PASS_MIN_LEN *[0-9]*"
        state: present
        line: "PASS_MIN_LEN        {{ var_accounts_password_minlen_login_defs }}"
      tags:
        - accounts_password_minlen_login_defs
        - medium_severity
        - restrict_strategy
        - low_complexity
        - low_disruption
        - CCE-80652-1
        - NIST-800-53-IA-5(f)
        - NIST-800-53-IA-5(1)(a)
        - NIST-800-171-3.5.7
        - CJIS-5.6.2.1


... Output Omitted ...

    - name: Ensure PAM variable maxclassrepeat is set accordingly

      lineinfile:
        create: yes
        dest: "/etc/security/pwquality.conf"
        regexp: '^#?\s*maxclassrepeat'
        line: "maxclassrepeat = {{ var_password_pam_maxclassrepeat }}"

      tags:
        - accounts_password_pam_maxclassrepeat
        - medium_severity
        - restrict_strategy
        - low_complexity
        - low_disruption
        - NIST-800-53-IA-5
        - NIST-800-53-IA-5(c)
```

Ansible is not setup for the lab

Before we can do the next steps, we will download an Ansible config file and an inventory file so we can properly run the playbook.

Listing 62. Error Output Message

```
[root@workstation ~]# ansible-playbook Custom_Scan_Fix.yml
 [WARNING]: provided hosts list is empty, only localhost is available. Note
that the implicit localhost does not match 'all'


PLAY [all] *********************************************************************
skipping: no hosts matched

PLAY RECAP *********************************************************************
[root@workstation ~]#
```

Step 2 - Downloading Ansible Config and Ansible Inventory Files

This step is needed so that our Ansible system can be configured with various configuration options and the inventory files so we can run the given playbook.

Example 19. Downloading Ansible Files

Listing 63. Downloading Ansible Files

```
[root@workstation ~]# wget http://people.redhat.com/~tmichett/rh354/inventory
--2020-04-16 09:38:50--  http://people.redhat.com/~tmichett/rh354/inventory
Resolving people.redhat.com (people.redhat.com)... 209.132.183.19
Connecting to people.redhat.com (people.redhat.com)|209.132.183.19|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24
Saving to: 'inventory'

inventory           100%[===================>]      24  --.-KB/s    in 0s

2020-04-16 09:38:50 (2.69 MB/s) - 'inventory' saved [24/24]


[root@workstation ~]# wget http://people.redhat.com/~tmichett/rh354/ansible.cfg
--2020-04-16 09:39:34--  http://people.redhat.com/~tmichett/rh354/ansible.cfg
Resolving people.redhat.com (people.redhat.com)... 209.132.183.19
Connecting to people.redhat.com (people.redhat.com)|209.132.183.19|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 159
Saving to: 'ansible.cfg'

ansible.cfg         100%[===================>]     159  --.-KB/s    in 0s

2020-04-16 09:39:35 (13.8 MB/s) - 'ansible.cfg' saved [159/159]
```

Reviewing Ansible Configurations

The **inventory** file provided only has a single host **servera** in there. On real systems, you must be very cautious of running remediation playbooks against an inventory file as it could apply to unintended systems. Additionally the **ansible.cfg** file provided was created for use in this lab environment. Both of these items should be taken into account when doing going through the process on production systems.

```
[root@workstation ~]# cat inventory
servera.lab.example.com

[root@workstation ~]# cat ansible.cfg
[defaults]
roles_path = /etc/ansible/roles:/usr/share/ansible/roles
log_path   = /tmp/ansible.log
inventory  = ./inventory

[privilege_escalation]
become=True
[root@workstation ~]#
```

Step 3 - Run the Ansible Playbook

This step will utilize the **workstation** system which is configured as your Ansible management node and will run the playbook to remediate the results on the **servera** system.

*Example 20. Remediation of **serverc** with Ansible Playbook*

Listing 64. Running the Ansible Playbook

```
[root@workstation ~]# ansible-playbook Custom_Scan_Fix.yml

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [servera.lab.example.com]

TASK [Set Password Minimum Length in login.defs] *******************************
changed: [servera.lab.example.com]

TASK [Ensure PAM variable difok is set accordingly] ****************************
changed: [servera.lab.example.com]

TASK [Ensure PAM variable ucredit is set accordingly] **************************
changed: [servera.lab.example.com]

TASK [Ensure PAM variable minlen is set accordingly] ***************************
changed: [servera.lab.example.com]

TASK [Set Password Retry Prompts Permitted Per-Session - system-auth (change)] ***
ok: [servera.lab.example.com]

TASK [Set Password Retry Prompts Permitted Per-Session - system-auth (add)] ****
changed: [servera.lab.example.com]

TASK [Ensure PAM variable minclass is set accordingly] *************************
changed: [servera.lab.example.com]

TASK [Ensure PAM variable maxrepeat is set accordingly] ************************
changed: [servera.lab.example.com]

TASK [Ensure PAM variable ocredit is set accordingly] **************************
changed: [servera.lab.example.com]

TASK [Ensure PAM variable lcredit is set accordingly] **************************
changed: [servera.lab.example.com]

TASK [Ensure PAM variable dcredit is set accordingly] **************************
changed: [servera.lab.example.com]

TASK [Ensure PAM variable maxclassrepeat is set accordingly] *******************
changed: [servera.lab.example.com]

PLAY RECAP *********************************************************************
servera.lab.example.com    : ok=13   changed=11   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

After running the playbook, you can see that there were 10 changes that were made to the system and exactly which parameters were changed. The next thing to do is perform another scan of the system to ensure that it is now fully compliant.

Step 4 - Rescan System and Review Results

Example 21. Scanning System after Fixes and Verifying Results

Listing 65. Performing SCAP Verification Scan

```
[root@servera ~]# oscap xccdf eval \
--profile xccdf_org.ssgproject.content_profile_ospp_customized \
--tailoring-file ssg-rhel8-ds-tailoring.xml \
--results custom_scan_results_fixed.xml \
--report Custom_Scan_Report_Fixed.html \
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

Listing 66. Copying Results to Workstation

```
[root@servera ~]# scp custom_scan_results_fixed.xml Custom_Scan_Report_Fixed.html workstation:
root@workstation's password:
custom_scan_results_fixed.xml                 100% 4086KB  60.3MB/s   00:00
Custom_Scan_Report_Fixed.html                 100%  282KB  48.4MB/s   00:00
```

Listing 67. Viewing Results on Workstation

```
[root@workstation ~]# firefox Custom_Scan_Report_Fixed.html
```

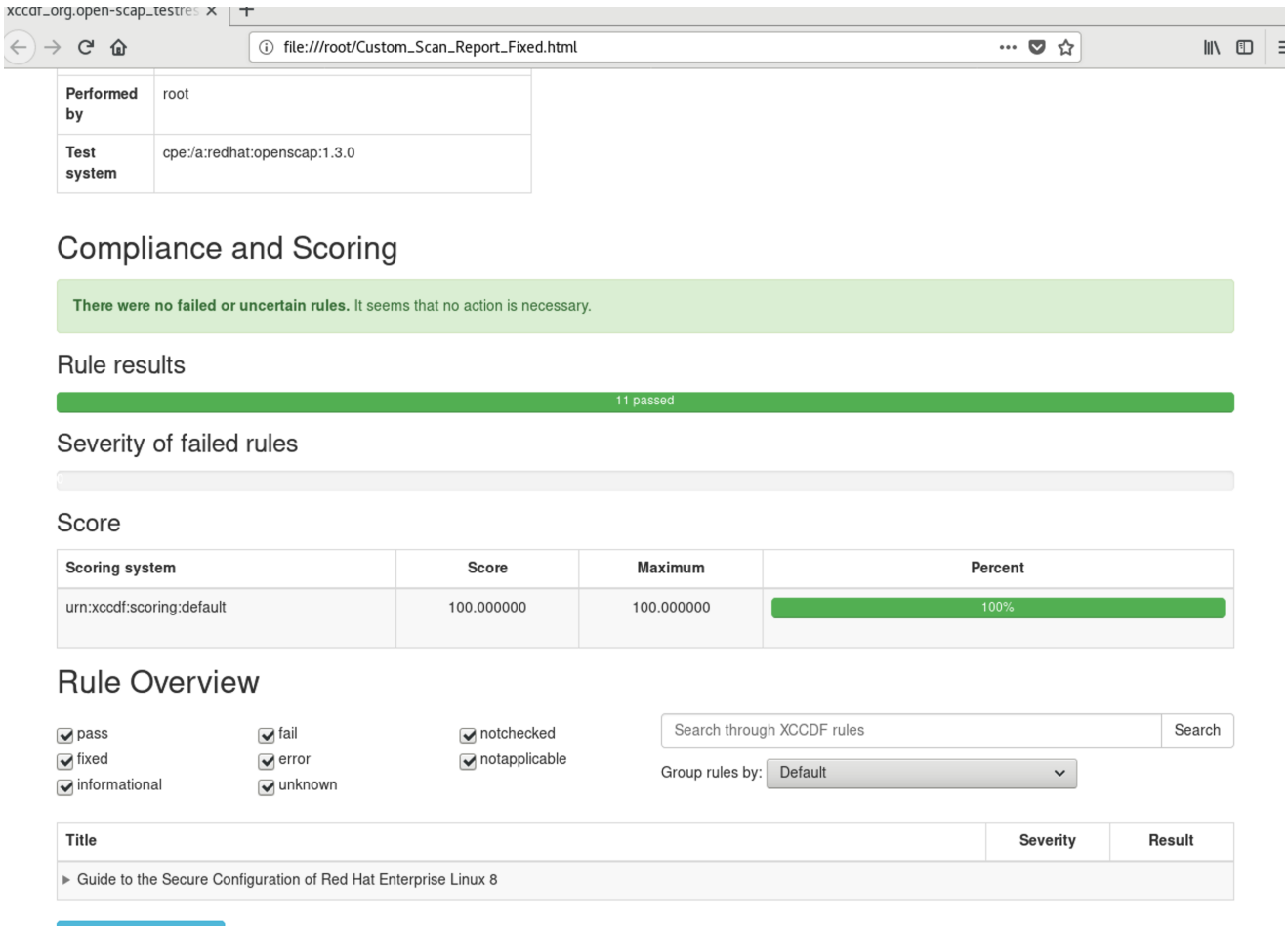

Figure 28. Fixed SCAP Scan Results Report in Firefox

Appendix B: IdM Server on RHEL8

B.1. Installing the IdM Server on RHEL8

Installing IdM Server on RHEL8

Red Hat RHEL8 IdM KB Article: https://access.redhat.com/articles/3623971

** Requires login*

Installing the IdM Server

1. List yum modules

Listing 68. Source Description

```
[root@servera ~]# sudo yum module list | grep -i idm
idm                 DL1         common [d], adtrust, client, dns, server The Red Hat Enterprise Linux Identity Management system module
idm                 client [d]  common [d]                               RHEL IdM long term support client module
```

RHEL8 Application Streams and Modules

Because RHEL8 uses DNF/YUM4 and has AppStreams and Modules, many packages may be setup as defaults while other modules aren't available. The IdM server package is one such package as the IdM server package is installed from the DL1 application stream.

2. Enable IdM Server packages by enabling the **IdM DL1** stream

Listing 69. Enabling a Module Stream

```
[root@servera ~]# yum module enable idm:DL1
Last metadata expiration check: 0:07:46 ago on Tue 14 Apr 2020 02:20:48 AM EDT.
Dependencies resolved.
================================================================================
 Package              Arch               Version              Repository           Size
================================================================================
Enabling module streams:
 389-ds                                  1.4
 httpd                                   2.4
 idm                                     DL1
 pki-core                                10.6
 pki-deps                                10.6

Transaction Summary
================================================================================

Is this ok [y/N]: y
Complete!

Switching module streams does not alter installed packages (see 'module enable' in dnf(8) for details)
```

3. Obtain information on the Module Stream

Listing 70. Obtaining Module Stream Packages

```
[root@servera ~]# yum module info --profile idm:DL1
Last metadata expiration check: 0:09:07 ago on Tue 14 Apr 2020 02:20:48 AM EDT.
Name    : idm:DL1:820190227212412:5986f621:x86_64
common  : ipa-client
adtrust : ipa-idoverride-memberof-plugin
        : ipa-server-trust-ad
client  : ipa-client
dns     : ipa-server
        : ipa-server-dns
server  : ipa-server
```

4. Installing the **ipa-server** package

Listing 71. Installing **ipa-server**

```
[root@servera ~]# yum module install idm:DL1/server
Last metadata expiration check: 0:11:29 ago on Tue 14 Apr 2020 02:20:48 AM EDT.
Dependencies resolved.
================================================================================================
 Package              Arch   Version                              Repository                Size
================================================================================================
Installing group/module packages:
 ipa-server           x86_64 4.7.1-11.module+el8+2842+7481110c   rhel-8.0-for-x86_64-appstream-rpms 502 k

... output omitted ...

  python3-sss-murmur-2.0.0-43.el8.x86_64
  redhat-logos-httpd-80.7-1.el8.noarch
  sssd-dbus-2.0.0-43.el8.x86_64
  sssd-tools-2.0.0-43.el8.x86_64

Complete!
```

IPA Server Options

It is important to note that for this demonstration, the DNS server functionality was not installed with the YUM command and it is not installed or configured as part of the **ipa-server-install** command.

5. Run the IdM Installer command **ipa-server-install** with all options required

Listing 72. Performing IPA Server Installation

```
[root@servera ~]# ipa-server-install --realm LAB.EXAMPLE.COM --ds-password redhat123 --admin-password redhat123 --no-ntp

The log file for this installation can be found in /var/log/ipaserver-install.log
==============================================================================
This program will set up the IPA Server.
Version 4.7.1

This includes:
  * Configure a stand-alone CA (dogtag) for certificate management
  * Create and configure an instance of Directory Server
  * Create and configure a Kerberos Key Distribution Center (KDC)
  * Configure Apache (httpd)
  * Configure the KDC to enable PKINIT
```

```
Excluded by options:
  * Configure the NTP client (chronyd)

To accept the default shown in brackets, press the Enter key.

Do you want to configure integrated DNS (BIND)? [no]:

Enter the fully qualified domain name of the computer
on which you're setting up server software. Using the form
<hostname>.<domainname>
Example: master.example.com.


Server host name [servera.lab.example.com]:

The domain name has been determined based on the host name.

Please confirm the domain name [lab.example.com]:


The IPA Master Server will be configured with:
Hostname:       servera.lab.example.com
IP address(es): 172.25.250.10
Domain name:    lab.example.com
Realm name:     LAB.EXAMPLE.NET

The CA will be configured with:
Subject DN:   CN=Certificate Authority,O=LAB.EXAMPLE.NET
Subject base: O=LAB.EXAMPLE.NET
Chaining:     self-signed

WARNING: Realm name does not match the domain name.
You will not be able to establish trusts with Active Directory unless
the realm name of the IPA server matches its domain name.


Continue to configure the system with these values? [no]: yes

... output omitted ...

Setup complete

Next steps:
    1. You must make sure these network ports are open:
        TCP Ports:
          * 80, 443: HTTP/HTTPS
          * 389, 636: LDAP/LDAPS
          * 88, 464: kerberos
        UDP Ports:
          * 88, 464: kerberos

    2. You can now obtain a kerberos ticket using the command: 'kinit admin'
       This ticket will allow you to use the IPA tools (e.g., ipa user-add)
       and the web user interface.

Be sure to back up the CA certificates stored in /root/cacert.p12
These files are required to create replicas. The password for these
files is the Directory Manager password
The ipa-server-install command was successful
```

6. Open Firewall

Listing 73. IPA Firewall Service Provisioning

```
[root@servera ~]#  firewall-cmd --add-service=freeipa-ldap --add-service=freeipa-ldaps --add-service=dns --permanent
success
[root@servera ~]# firewall-cmd --reload
success
```

7. Login to IdM Web Console

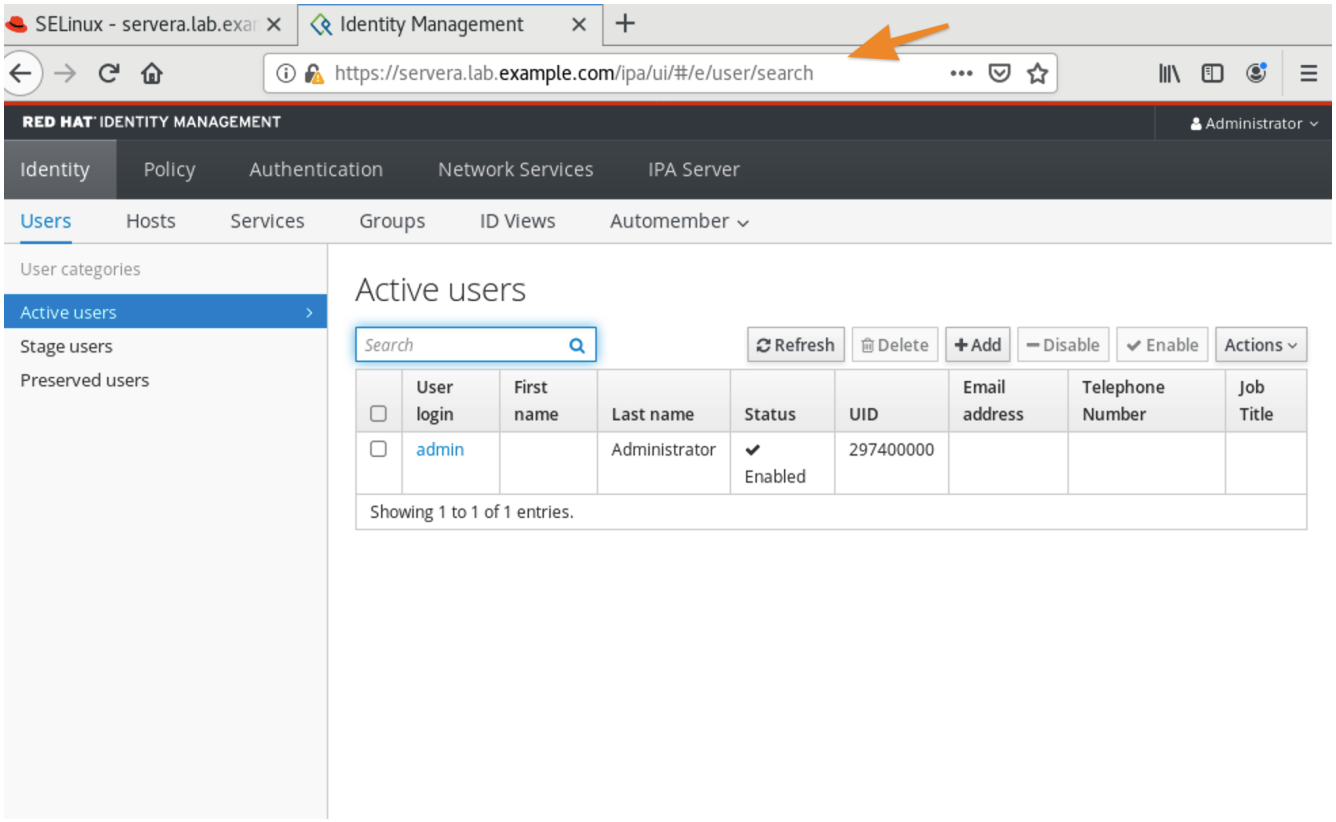

Figure 29. IdM Web Console

B.2. Installing the IdM Client on RHEL8

IdM Client Installation

The default Application Stream and Module for RHEL8 already supports and provides the **ipa-client** package so there is nothing special to perform in order to get the client installed.

1. Install the **ipa-client** package

Listing 74. YUM Installation of **ipa-client**

```
[root@serverb ~]#  yum install ipa-client
Red Hat Enterprise Linux 8.0 AppStream (dvd)                    281 kB/s | 3.2 kB     00:00
Red Hat Enterprise Linux 8.0 BaseOS (dvd)                       476 kB/s | 2.7 kB     00:00

... output omitted ...

  sssd-tools-2.0.0-43.el8.x86_64
  xmlrpc-c-1.51.0-5.el8.x86_64
  xmlrpc-c-client-1.51.0-5.el8.x86_64

Complete!
```

2. Modify DNS settings of system to point to IdM Server

Listing 75. Using Network Manager to Change DNS Settings

```
[root@serverb ~]# nmcli connection modify "Wired connection 1" ipv4.dns 172.25.250.10

[root@serverb ~]# nmcli connection up "Wired connection 1"
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/1325)

[root@serverb ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search lab.example.com example.com
nameserver 172.25.250.10
```

3. Run the client installation command

Listing 76. Running the Installer Command for the IdM Client

```
[root@serverb ~]# ipa-client-install --principal admin --password redhat123 --mkhomedir --unattended --ntp-server=172.25.254.254 --domain
lab.example.com --server servera.lab.example.com --realm LAB.EXAMPLE.NET
This program will set up IPA client.
Version 4.7.1

Client hostname: serverb.lab.example.com
Realm: LAB.EXAMPLE.NET
DNS Domain: lab.example.com
IPA Server: servera.lab.example.com
BaseDN: dc=lab,dc=example,dc=net

Synchronizing time
Configuration of chrony was changed by installer.
Attempting to sync time with chronyc.
Time synchronization was successful.
Successfully retrieved CA cert
    Subject:     CN=Certificate Authority,O=LAB.EXAMPLE.NET
    Issuer:      CN=Certificate Authority,O=LAB.EXAMPLE.NET
    Valid From:  2020-04-14 06:38:52
    Valid Until: 2040-04-14 06:38:52

... output omitted ...

SSSD enabled
Configured /etc/openldap/ldap.conf
Configured /etc/ssh/ssh_config
Configured /etc/ssh/sshd_config
Configuring lab.example.com as NIS domain.
Client configuration complete.
The ipa-client-install command was successful
```