



# GETTING TO KNOW RED HAT ENTERPRISE LINUX 7

CHRISTOPH DOERBECK  
Principal Solutions Architect  
[cdoerbec@redhat.com](mailto:cdoerbec@redhat.com)

# Agenda

## Review new GUI installer

- Walk through decision points
- Discuss improved features

## Hands-on Activities

- **Installing a Webserver:**
  - Package installation
  - Storage Management with SSM
  - Web Services Configuration (systemd, httpd configuration)
  - Security with SELinux & FirewallD
  - Performance Tuning with “tuned” profiles

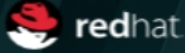
# The NEW RHEL 7 Installer

## Highlights

- Fewer Questions
- Background Installation
- Support for Complex Network Configurations
- Support for Complex Storage Configurations
- Reset partitioning without rebooting

Let's examine a few screenshots...

# Installation Summary






INSTALLATION SUMMARY



RED HAT ENTERPRISE LINUX 7.0 INSTALLATION

us



### LOCALIZATION

 <b>DATE &amp; TIME</b> <i>Americas/New York timezone</i>	 <b>KEYBOARD</b> <i>English (US)</i>
 <b>LANGUAGE SUPPORT</b> <i>English (United States)</i>	

### SOFTWARE


 <b>INSTALLATION SOURCE</b> <i>Local media</i>	 <b>SOFTWARE SELECTION</b> <i>Minimal Install</i>
---	---

### SYSTEM

 <b>INSTALLATION DESTINATION</b> <i>Automatic partitioning selected</i>	 <b>NETWORK &amp; HOSTNAME</b> <i>Not connected</i>
--	---

[Quit](#) [Begin Installation](#)

*We won't touch your disks until you click 'Begin Installation'.*

 Please complete items marked with this icon before continuing to the next step.

# Network

**NETWORK & HOSTNAME** RED HAT ENTERPRISE LINUX 7.0 INSTALLATION

[Done](#) us

**Ethernet (eth0)**  
Red Hat, Inc Virtio network device

**Ethernet (eth0)** ON  
Connected

Hardware Address 52:54:00:9A:2E:EA

Speed

IP Address 192.168.1.214

Subnet Mask 255.255.255.0

Default Route 192.168.1.1

DNS 192.168.1.1

**Add device**  
Select the type of device you wish to add


- Bond
- Team
- Vlan

[Configure...](#)

Hostname: localhost.localdomain

# Destination

INSTALLATION DESTINATION RED HAT ENTERPRISE LINUX 7.0 INSTALLATION

[Done](#) 

[Multipath Devices](#) [Other SAN Devices](#) [Firmware RAID](#)

Search By:

Search Results:

Name	WWID	Capacity	Interconnect	Model	LUN	Port	Target	Vendor
------	------	----------	--------------	-------	-----	------	--------	--------

[1 storage device selected](#)

# Partitions

**MANUAL PARTITIONING** RED HAT ENTERPRISE LINUX 7.0 INSTALLATION

[Done](#) us

### New Red Hat Enterprise Linux 7.0 Installation

You haven't created any mount points for your Red Hat Enterprise Linux 7.0 installation yet. You can:

- [Click here to create them automatically.](#)
- Create new mount points by clicking the '+'

New partition type

- Standard Partition
- BTRFS
- LVM
- LVM Thin Provisioning

When you create mount points for your Red Hat Enterprise Linux 7.0 installation, you'll be able to view their details here.

**+** **-** **✖** **↺** **⊞**

AVAILABLE SPACE	TOTAL SPACE
<b>5.11 GB</b>	<b>5.12 GB</b>

[1 storage device selected](#) Reset All

# Software Selection

**SOFTWARE SELECTION** Done RED HAT ENTERPRISE LINUX 7.0 INSTALLATION us

**Base Environment**

- Minimal Install**  
Basic functionality.
- Infrastructure Server**  
Server for operating network infrastructure services.
- File and Print Server**  
File, print, and storage server for enterprises.
- Basic Web Server**  
Server for serving static and dynamic internet content.
- Virtualization Host**  
Minimal virtualization host.
- Server with GUI**  
Server for operating network infrastructure services, with a GUI.

**Add-Ons for Selected Environment**

- Backup Server**  
Software to centralize your infrastructure's backups.
- DNS Name Server**  
This package group allows you to run a DNS name server (BIND) on the system.
- Directory Server**  
Machine and user identity servers.
- E-mail Server**  
Allows the system to act as a SMTP and/or IMAP e-mail server.
- FTP Server**  
Allows the system to act as an FTP server.
- File and Storage Server**  
CIFS, SMB, NFS, iSCSI, iSER, and iSNS network storage server.
- Hardware Monitoring Utilities**  
A set of tools to monitor server hardware.
- Identity Management Server**  
Centralized management of users, servers and authentication policies.
- Infiniband Support**  
Software designed for supporting clustering and grid connectivity using RDMA-based InfiniBand and iWARP fabrics.



# Conventions

###



```
File Edit View Search Terminal Help

[student@client-01 ~]$
[student@client-01 ~]$ sudo -i
Last login: Sat Oct 18 15:29:45 EDT 2014 on tty1
Last failed login: Fri Oct 24 13:25:18 EDT 2014 from 122.225.109.107 on ssh:notty
There were 196 failed login attempts since the last successful login.
[root@client-01 ~]#
```

3

4

1 Don't like typing? Use the lab helper  
ex: lab-helper-###

2 Consult attendee portal for URL info

SAMPLE

Host URL: CLIENT-01

Login: student

Password: n0boundaries!

\*\* CLIENT-02 provides jumphost option

3 Commands are in BOLD

4 Call-outs are highlighted in yellow

5 Just about everything we do will be as  
"root" but keep an eye on the prompt

6 Are you on the right host?

# Log In via SIAB / SSH



```
File Edit View Search Terminal Help

client-01 login: student
student@client-01.example.com's password: *****

[student@client-01 ~]$

[student@client-01 ~]$ sudo -i
Last login: Sat Oct 18 15:29:45 EDT 2014 on tty1
Last failed login: Fri Oct 24 13:25:18 EDT 2014 from 122.225.109.107 on ssh:notty
There were 196 failed login attempts since the last successful login.

root# cd

root# . helper-g2k
```

1

2

3

**1 CONNECTION INFORMATION**  
Host URL: CLIENT-01  
Login: student  
Password: n0boundaries!

**2 Don't forget "sudo -i"**

**3 Source the Lab Helpers**  
*\*\* NOTE \*\*The command starts with a period !'*

`. helper-g2k`

# Install Webserver Software

```
1 root# yum groupinstall -y "Basic Web Server"
2 Loaded plugins: langpacks, product-id, rhnplugin, subscription-manager
Resolving Dependencies
<...snip...>
Installed:
crypto-utils.x86_64 0:2.4.1-42.e17
httpd.x86_64 0:2.4.6-18.e17_0
httpd-manual.noarch 0:2.4.6-18.e17_0
mod_fcgid.x86_64 0:2.3.9-4.e17
mod_ssl.x86_64 1:2.4.6-18.e17_0

Dependency Installed:
apr.x86_64 0:1.4.8-3.e17
apr-util.x86_64 0:1.5.2-6.e17
httpd-tools.x86_64 0:2.4.6-18.e17_0
mailcap.noarch 0:2.1.41-2.e17
perl-Newt.x86_64 0:1.08-36.e17

Updated:
NetworkManager-config-server.x86_64 1:0.9.9.1-26.git20140326.4dba720.e17_0

Complete!
```

- 1 Use "yum" to install package group
- 2 On occasion we will cut out the boring messages with "<...snip...>"

# Configure HTTP

```
File Edit View Search Terminal Help
1 root# rhncfg-client get
Using server name sat5.example.com
Deploying /etc/sudoers
Deploying /etc/cron.d/ravello-nic-check
2 Deploying /etc/httpd/conf.d/example_corp.conf
Deploying /etc/systemd/system/httpd.service.d/50-httpd.conf

3 root# cat /etc/httpd/conf.d/example_corp.conf

DocumentRoot /var/www/html/example_corp

<Directory /var/www/html/example_corp/ >
    AllowOverride All
    Options Indexes FollowSymLinks
</Directory>
```

- 1 We are leveraging config file management from a local RH Satellite Server
- 2 This is the config file we are interested in
- 3 Inspect the config file

Defines DocumentRoot as:  
/var/www/html/example\_corp

# Understanding SSM

- **System Store Manager**
- **Single CLI to manage storage technologies**
  
- **Examples:**
  - Device Mapper (dm), Encryption
  - Logical Volume Manager (LVM), Snapshots
  - Multiple Devices (MD), RAID
  - More...

# Exploring SSM

```
1 root# yum install -y system-storage-manager
root# ssm list volumes
-----
Volume          Pool          Volume size  FS      FS size    Free  Type    Mount point
-----
/dev/rhel_pwob-r7/swap  rhel_pwob-r7    1.00 GB
/dev/rhel_pwob-r7/root  rhel_pwob-r7    8.51 GB  xfs     8.50 GB    6.57 GB  linear  /
/dev/vda1              500.00 MB  xfs     496.67 MB  375.54 MB  part    /boot
-----

root# mkdir -p /mnt/test

2 root# ssm -f create --fstype ext4 /dev/vdb1 /mnt/test
Physical volume "/dev/vdb1" successfully created
Volume group "lvm_pool" successfully created
<...snip...>

root# df /mnt/test
Filesystem          1K-blocks  Used Available Use% Mounted on
/dev/mapper/lvm_pool-lvol001  118867  1550   108430   2% /mnt/test

root# ssm -f remove /mnt/test lvm_pool

root# pvremove /dev/vdb1
```

- 1 Install SSM
- 2 Single Command:  
Creates VG/Pool  
Creates LV  
Formats FS  
Mounts FS

# Setup HTML Storage with SSM

```
1 root# mkdir -p /var/www/html
2 root# ssm -f create -s 50M -p webvg -n html -r 1 \
  --fstype ext4 /dev/{vdb1,vdc1} /var/www/html

Physical volume "/dev/vdb" successfully created
Physical volume "/dev/vdc" successfully created
Volume group "webvg" successfully created
Rounding up size to full physical extent 52.00 MiB
<...snip...>

root# df /var/www/html
Filesystem      1K-blocks  Used Available Use% Mounted on
/dev/mapper/webvg-html 47463 1038 42699 3% /var/www/html

3 root# restorecon -R /var/www/html

4 root# lvs webvg/html
LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert
html webvg rwi-aor--- 52.00m 100.00

5 root# grep webvg /proc/mounts >> /etc/fstab
```

- 1 Create Mount Point
- 2 Create Mirrored Filesystem
  - size 50MB
  - vg/lv name webvg/html
  - raid 1 (mirror)
  - fstype ext4
  - devices vdb1, vdc1
  - mount /var/www/html
- 3 Restore SELinux contexts
- 4 Verify RAID Syn+c status
- 5 Add Mount to fstab

# Download HTML Content

```
1 root# cd /var/www/html
2 root# wget http://sat5.example.com/pub/example_corp.tgz
--2014-10-24 22:20:56-- http://sat5.example.com/pub/example_corp.tgz
Resolving sat5.example.com (sat5.example.com)... 10.0.0.2
Connecting to sat5.example.com (sat5.example.com)|10.0.0.2|:80... connected.
<...snip...>
2014-10-24 22:20:56 (79.7 MB/s) - 'example_corp.tgz' saved [719421/719421]
3 root# tar xzf example_corp.tgz

root# ls -l
total 704
drwxr-xr-x. 3 root root    48 Oct 24 22:21 .
drwxr-xr-x. 4 root root    31 Oct 24 22:08 ..
drwxr-xr-x. 6 root root    67 Oct 16 11:55 example_corp
-rw-r--r--. 1 root root 719421 Oct 18 19:29 example_corp.tgz
```

- 1 Change directory to /var/www/html
- 2 Retrieve HTML content from http://sat5.example.com/pub
- 3 Don't forget to unpack the tarball



# Understanding Firewalld

## Still IPTables underneath

### Why Firewalld:

- Realtime rule changes without interruption
- Separate network traffic & rules by interface or zones
- GUI and CLI tools
- System configs in `/usr/lib/firewalld/*`
- Custom config in `/etc/firewalld/*`

# Exploring Firewalld

```
File Edit View Search Terminal Help
root# firewall-cmd --state
not running
root# systemctl enable firewalld
root# systemctl start firewalld
root# firewall-cmd --get-active-zones
public
  interfaces: eth0
root# firewall-cmd --list-all
public (default, active)
  interfaces: eth0
  sources:
  services: dhcpv6-client siab ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
root# firewall-cmd --remove-service dhcpv6-client
root# firewall-cmd --remove-service dhcpv6-client --permanent
```

1

2

3

4

1

Check firewalld status

2

Enable firewalld service with systemd

3

Show all firewalld services

4

Disable unwanted firewalld service

add --permanent to persist after system reboot

# Secure HTTP with Firewalld

```
1 root# firewall-cmd --add-service http --add-service https
  success

  root# firewall-cmd --add-service http --add-service https --permanent
  success

2 root# firewall-cmd --list-all
  public (default, active)
  interfaces: eth0
  sources:
  services: http https siab ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

- 1 Enable http(s) services  
add --permanent to persist after system reboot
- 2 Check enabled firewalld services

# Understanding SELinux

## Standard Security = Discretionary Access Controls

- Owner, group, world/read, write, execute
- Enhanced with Access Control Lists (ACLs)

## Security-enhanced Linux = Mandatory Access Controls

- Framework allows definition of permissions for how all processes (called subjects) interact with other parts of the system such as files, devices, sockets, ports, and other processes (called objects in SELinux).

**SELinux was introduced in RHEL 4**

# Exploring SELinux

```
File Edit View Search Terminal Help
1 root# getenforce
  Permissive
2 root# setenforce Enforcing
  root# getenforce
  Enforcing
3 root# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

- 1 Check SELinux Status
- 2 Set SELinux to Enforcing
- 3 To set persistent SELinux mode, edit config file `/etc/sysconfig/selinux`

**\*Note\***

SELinux Mode  
SELinux Type (Policy)

# Secure HTTP with SELinux

```
1 root# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:      /etc/selinux
Loaded policy name:           targeted
Current mode:                 enforcing
Mode from config file:       permissive
Policy MLS status:           enabled
Policy deny_unknown status:  allowed
Max kernel policy version:   28

root# cd /var/www/html

2 root# ls -Z
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 example_corp
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 example_corp.tgz

3 root# chcon -R -t tmp_t example_corp/images

root# ls -Z example_corp/images
-rw-r--r--. root root unconfined_u:object_r:tmp_t:s0 banner.jpg
-rw-r--r--. root root unconfined_u:object_r:tmp_t:s0 business-people.jpg
-rw-r--r--. root root unconfined_u:object_r:tmp_t:s0 pic01.jpg
-rw-r--r--. root root unconfined_u:object_r:tmp_t:s0 pic02.jpg
<.. snip ..>
```

- 1 Check SELinux Status  
  
\*Note\* how current and persistent configuration do not match
- 2 Examine SELinux security contexts
- 3 Change security context on some files  
  
changing *httpd\_sys\_content\_t* to *tmp\_t*

# Understanding Tuned

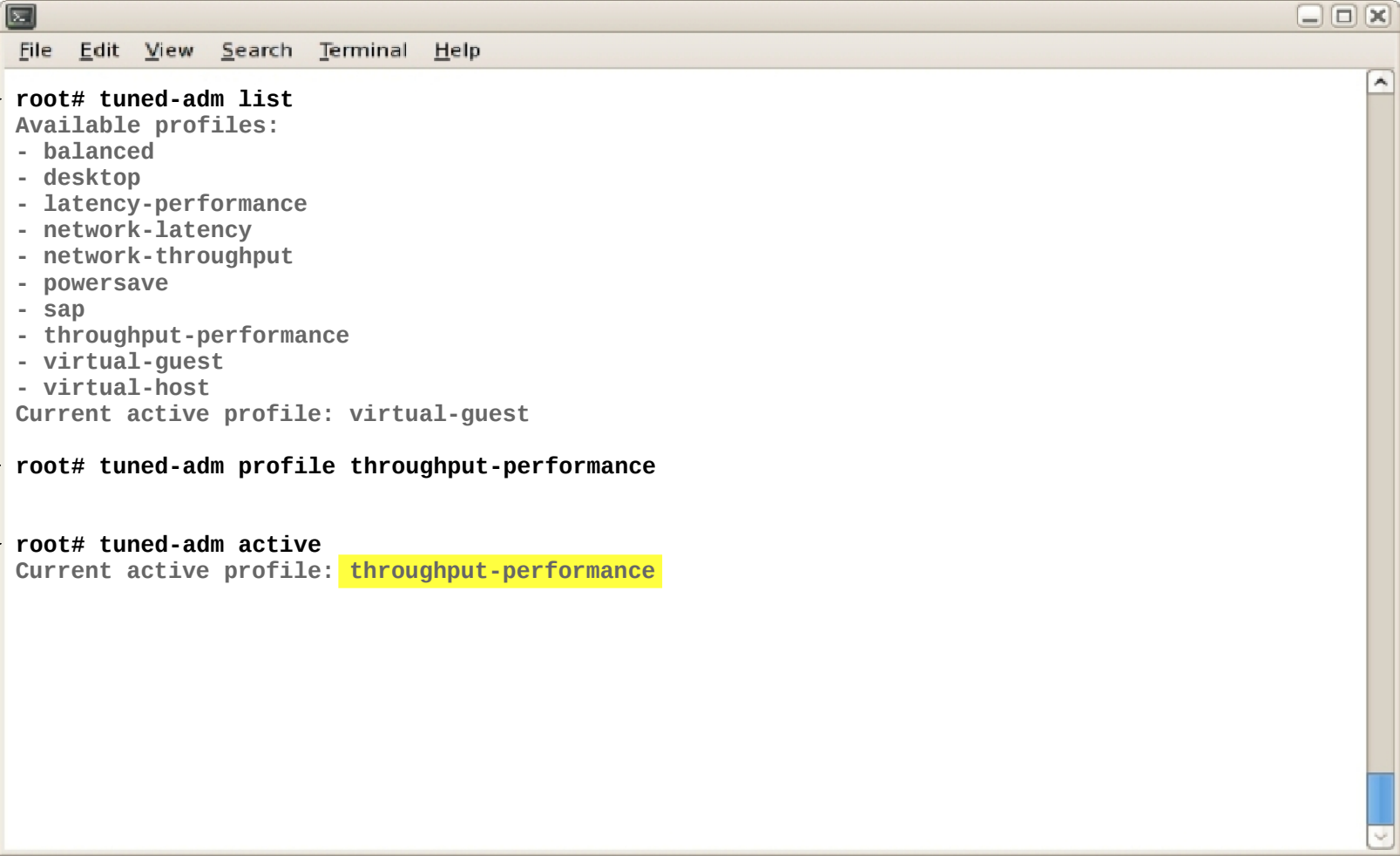
**Tuned provides recommended configurations for common workloads**

- Easy to apply
- Easy to customize
- Persists across reboots

**Installed and enabled with RHEL 7**

**Available for RHEL 6 as well**

# Exploring Tuned



```
File Edit View Search Terminal Help
root# tuned-adm list
Available profiles:
- balanced
- desktop
- latency-performance
- network-latency
- network-throughput
- powersave
- sap
- throughput-performance
- virtual-guest
- virtual-host
Current active profile: virtual-guest

root# tuned-adm profile throughput-performance

root# tuned-adm active
Current active profile: throughput-performance
```

- 1 “tuned” provides complete tuning profiles for common work-loads
- 2 Change the configured tuning profile for the system
- 3 Verify active profile



# Tune HTTP with Tuned

```
File Edit View Search Terminal Help

1 root# tuned-adm profile network-throughput

2 root# tuned-adm active
Current active profile: network-throughput
```

- 1 Change tuning profile for the system
- 2 Verify active profile

# Understanding Systemd

## Prior to RHEL 7, you had a SysV Init style “Upstart”

- Remember initd, inittab, rc scripts, run levels and who can forget rc.local?

## Systemd replaces SysV Init

- Controls “units” rather than just daemons
- Handles dependency with service information
- Tracks processes with service information
- Services are owned by a cgroup
- Simple to configure “SLAs” based on CPU memory
- Properly kills daemons (zombies are for tv & movies, not the Enterprise)
- Minimize boot times
- Debuggability – no early boot messages are lost
- Easy to learn and backwards compatible.

# Exploring Systemd

```
1 root# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Wed 2014-11-12 22:59:06 EST; 1h 5min ago
   Process: 1113 ExecStartPre=/usr/sbin/ssh-keygen (code=exited, status=0/SUCCESS)
   Main PID: 1125 (sshd)
   CGroup: /system.slice/ssh.service
           └─1125 /usr/sbin/sshd -D

<...snip...>

2 root# systemctl -t service

3 root# systemctl disable postfix
root# systemctl stop postfix
```

1 Look how much info systemd provides:

service status  
active since  
pid & ppid  
cgroup info  
more...

2 List configured services

3 Disable an unwanted service (postfix)

# Enable HTTP with Systemd

```
1 root# systemctl enable httpd
ln -s '/usr/lib/systemd/system/httpd.service'
'/etc/systemd/system/multi-user.target.wants/httpd.service'

2 root# systemctl start httpd

3 root# systemctl status httpd
httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
  Active: active (running) since Fri 2014-10-24 22:48:53 EDT; 8min ago
  Process: 2577 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=0/SUCCESS)
  Main PID: 2581 (httpd)
  Status: "Total requests: 15; Current requests/sec: 0; Current traffic:  0 B/sec"
  CGroup: /system.slice/httpd.service
          └─2581 /usr/sbin/httpd -DFOREGROUND
            └─2583 /usr/sbin/httpd -DFOREGROUND
              └─2584 /usr/sbin/httpd -DFOREGROUND

<...snip...>
```

- 1 Use systemd to enable httpd
- 2 Use systemd to start httpd
- 3 Check httpd status

cgroup slice httpd.service

# Understanding CGroups

## Cgroups provide Resource Management

- Manages CPU, Memory, Network and Block i/o consumption
- Reduce process contention, increase throughput and predictability

## Systemd replaces SysV Init

- system.slice – contains system services
- user.slice – contains user sessions
- machine.slice – contains virtual machines and containers

Services can be promoted to their own slice if needed

# Exploring CGroups

```
1 root# systemd-cgls
├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 23
├─user.slice
│   └─user-0.slice
│       └─session-5.scope
│           ├──2295 sshd: root@pts/0
│           ├──2331 -bash
│           ├──2370 systemd-cgls
│           └─2371 systemd-cgls
└─system.slice
    ├──polkit.service
    │   └─738 /usr/lib/polkit-1/polkitd --no-debug
    └─auditd.service
        └─516 /sbin/auditd -n
<... snip ...>

root# systemd-cgtop

2 root# systemctl set-property httpd.service CPUShares=2048

3 root# systemctl set-property --runtime httpd.service CPUShares=2048
```

- 1 Broad look at cgroup configuration
- 2 Apply change to subsequent httpd starts
- 3 Apply change to current running httpd

# Configure HTTP with CGroups

```
File Edit View Search Terminal Help
1 root# rhncfg-client get
Using server name sat5.example.com
Deploying /etc/sudoers
Deploying /etc/cron.d/ravello-nic-check
Deploying /usr/local/bin/ravello-nic-check.sh
Deploying /etc/httpd/conf.d/example_corp.conf
Deploying /etc/systemd/system/httpd.service.d/50-httpd.conf

2 root# cat /etc/systemd/system/httpd.service.d/50-httpd.conf
[Service]
Restart=always
CPUShares=2048
OOMScoreAdjust=-1000
```

1 Download systemd unit file for httpd

*\*\*Provided here by Satellite Configuration Management*

2 Inspect the unit file

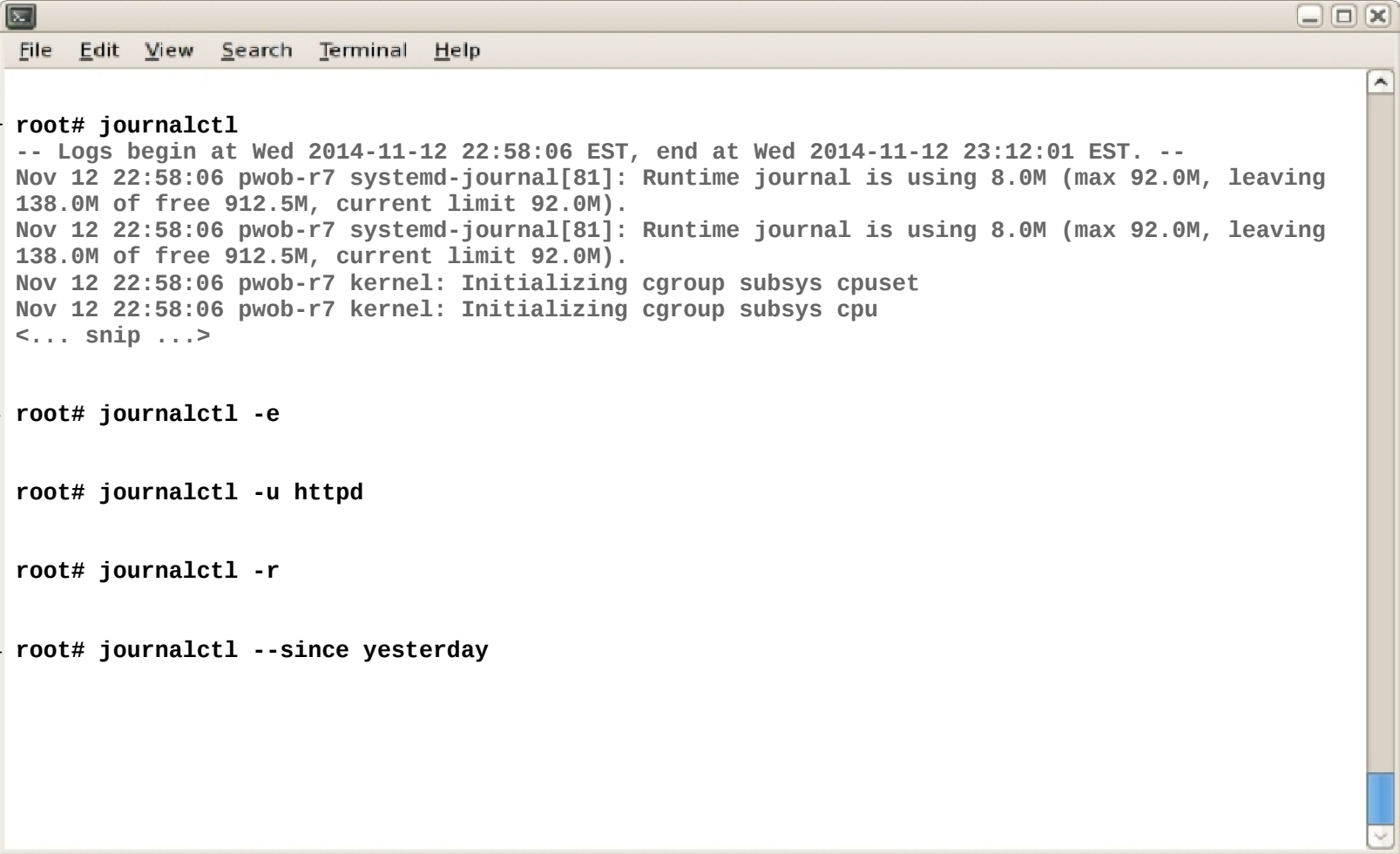
Restart=Always  
CPUShares=2048  
OOMScoreAdjust=-1000

# Understanding Journald

- **Does NOT replace rsyslog in RHEL 7**  
Continue using rsyslog for traditional logging w/ enterprise features
- **The journal is not persistent by default**
- **Collects event metadata**
- **Stored in key-value pairs**
- **Simple (or complex) filtering**
- **Indexed**
- **Formatted**
- **Message Verification – source authenticity**



# Exploring Journald



```
File Edit View Search Terminal Help

1 root# journalctl
-- Logs begin at Wed 2014-11-12 22:58:06 EST, end at Wed 2014-11-12 23:12:01 EST. --
Nov 12 22:58:06 pwob-r7 systemd-journal[81]: Runtime journal is using 8.0M (max 92.0M, leaving
138.0M of free 912.5M, current limit 92.0M).
Nov 12 22:58:06 pwob-r7 systemd-journal[81]: Runtime journal is using 8.0M (max 92.0M, leaving
138.0M of free 912.5M, current limit 92.0M).
Nov 12 22:58:06 pwob-r7 kernel: Initializing cgroup subsys cpuset
Nov 12 22:58:06 pwob-r7 kernel: Initializing cgroup subsys cpu
<... snip ...>

2 root# journalctl -e

3 root# journalctl -u httpd

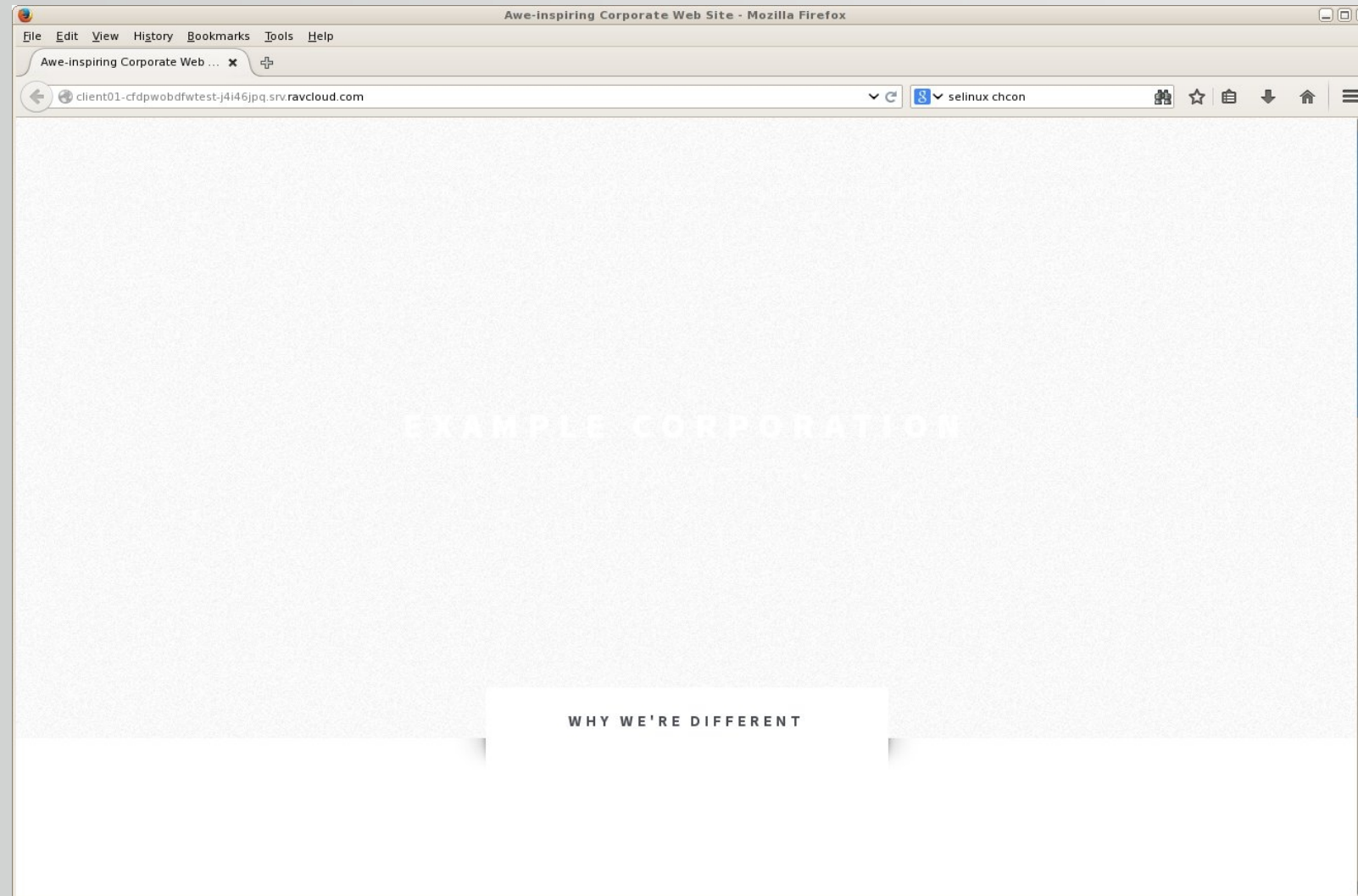
4 root# journalctl -r

5 root# journalctl --since yesterday
```

- 1 Display messages in "pager mode" starts at top of log (oldest)
- 2 Display messages in "pager mode" starts at end of log (newest)
- 3 Display messages for (unit) httpd
- 4 Reverse order
- 5 Since yesterday

# Validate Webserver Setup

## Use the URL provided



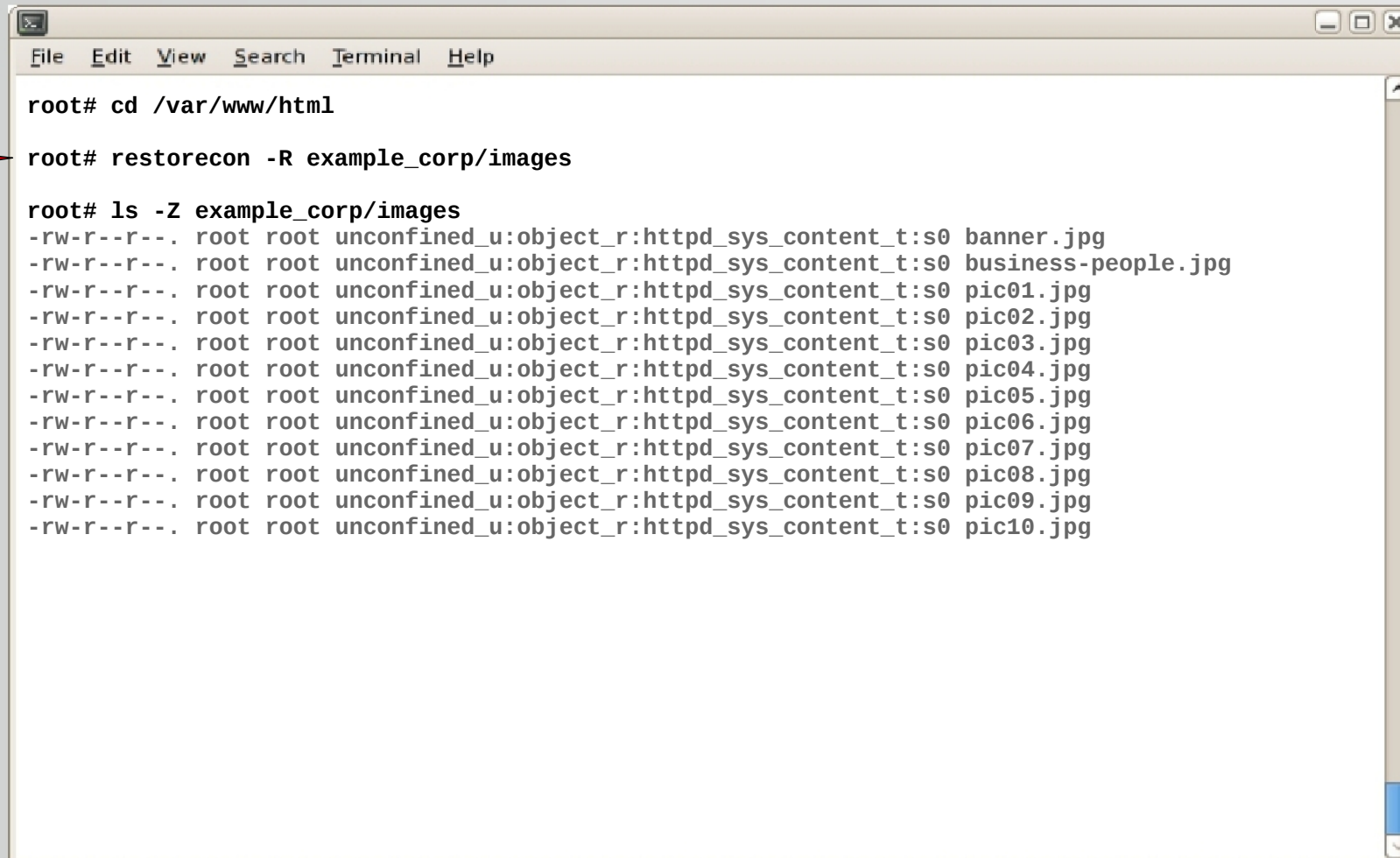
CONNECTION INFORMATION  
Host URL: CLIENT-01 AUX

### **\*Note\***

Webpage is incomplete  
All of the images are missing

Recall that we changed the  
security context of the files  
in the unpacked tarball

# Revisiting SELinux



A terminal window with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar. The terminal shows the following commands and output:

```
root# cd /var/www/html
root# restorecon -R example_corp/images
root# ls -Z example_corp/images
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 banner.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 business-people.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic01.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic02.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic03.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic04.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic05.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic06.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic07.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic08.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic09.jpg
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 pic10.jpg
```

1 Restore SELinux Security Context

# Revisiting Systemd

```
1 root# systemctl status httpd
httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
  Drop-In: /etc/systemd/system/httpd.service.d
           └─50-httpd.conf
  Active: active (running) since Mon 2014-11-10 21:40:45 EST; 54min ago
  Main PID: 826 (httpd)
  <...snip...>

2 root# pkill -9 httpd

3 root# journalctl -u httpd
<...snip...>
Nov 11 08:04:37 client-01.example.com systemd[1]: httpd.service: main process exited,
code=killed, status=9/KILL
Nov 11 08:05:10 client-01.example.com systemd[1]: Unit httpd.service entered failed state.
Nov 11 08:05:10 client-01.example.com systemd[1]: httpd.service holdoff time over, scheduling
restart.
Nov 11 08:05:10 client-01.example.com systemd[1]: Stopping The Apache HTTP Server...
Nov 11 08:05:10 client-01.example.com systemd[1]: Starting The Apache HTTP Server...
Nov 11 08:05:10 client-01.example.com systemd[1]: Started The Apache HTTP Server.
```

- 1 Check httpd status with systemd
- 2 Abruptly kill httpd processes
- 3 Examine httpd logs with journalctl

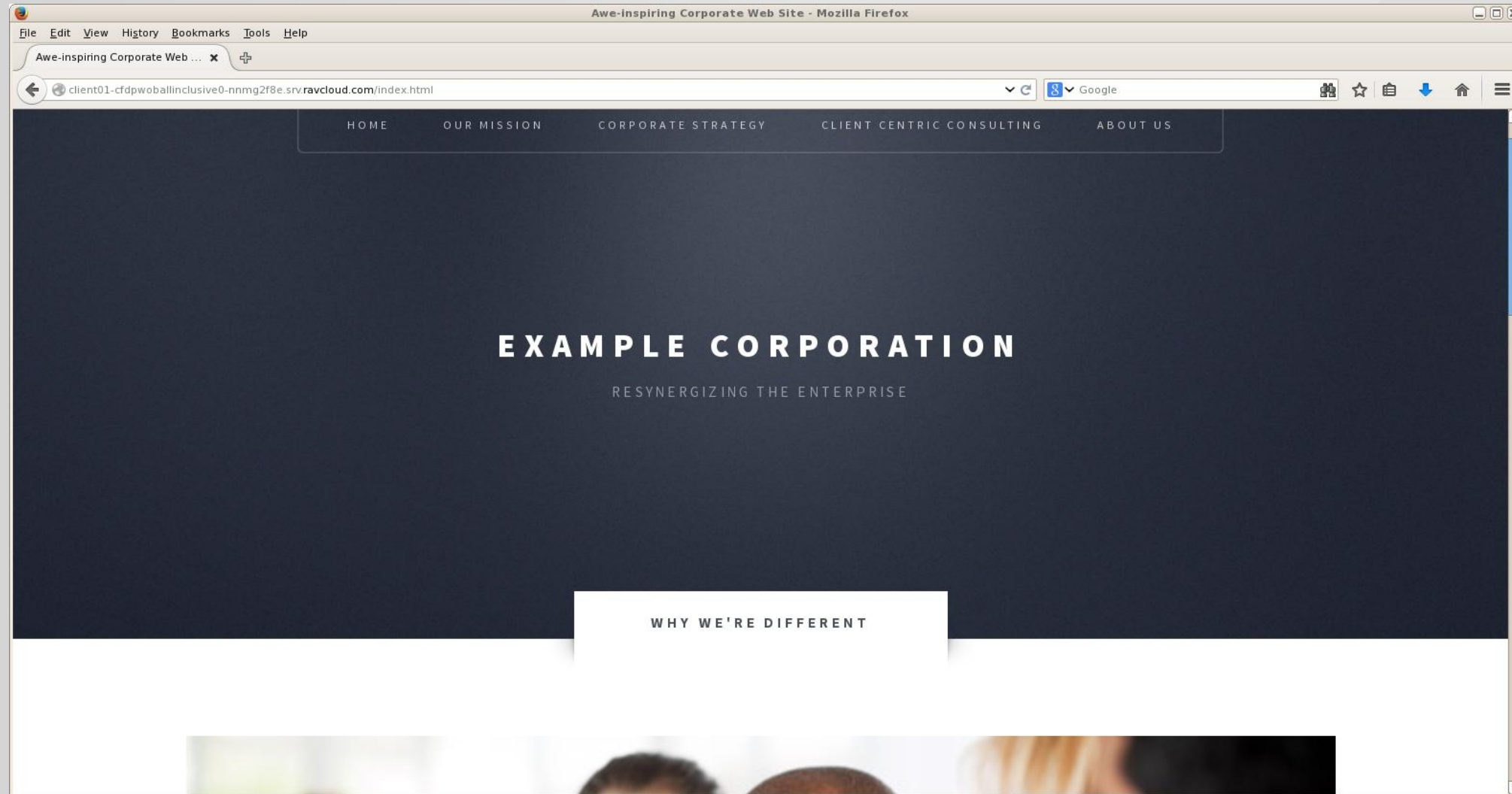
**\*Note\***

httpd exited  
httpd restarted

Now you can reload the web page

# Validate Webserver Setup

Use the URL provided



# QUESTIONS?



redhat.®