DUMPSARENA

Red Hat Certified System Administrator - RHCSA (8.2)

RedHat EX200

Version Demo

Total Demo Questions: 21

Total Premium Questions: 210

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Understand and use essential tools	21
Topic 2, Operate running systems	35
Topic 3, Configure local storage	37
Topic 4, Create and configure file systems	10
Topic 5, Deploy, configure, and maintain systems	50
Topic 6, Manage users and groups	27
Topic 7, Manage security	28
Topic 8, Mix Questions	2
Total	210

QUESTION NO: 1 - (SIMULATION)

Upgrading the kernel as 2.6.36.7.1, and configure the system to Start the default kernel, keep the old kernel available.

ANSWER: See the explanation for the answer

Explanation:

Install the new kernel package; do **not** use an RPM upgrade operation, because `rpm -U` can remove the old kernel.

```
# cd /boot
# lftp it
lftp> get dr/dom/kernel-2.6.36.7.1*.rpm
lftp> quit
# rpm -ivh kernel-2.6.36.7.1*.rpm
```

The `-i` option installs the 2.6.36.7.1 kernel alongside the currently installed kernel. The RPM installation creates a new GRUB boot entry.

Verify that the new kernel is the first `title` entry in the legacy GRUB configuration, then set the default entry to index 0:

```
# vi /etc/grub.conf

default=0
```

For example, the first entry should be the new kernel:

```
default=0
...
title Red Hat Enterprise Linux (...2.6.36.7.1...)
    root (...)
    kernel /vmlinuz-2.6.36.7.1... ro root=...
    initrd /initramfs-2.6.36.7.1....img
```

GRUB counts entries from zero, so `default=0` boots the first (new) kernel. Do not delete the older `title` entry; it remains available from the GRUB menu as a fallback.

QUESTION NO: 2 - (SIMULATION)

Install the Kernel Upgrade.

Install suitable kernel update from:

http://server.domain11.example.com/pub/updates.

Following requirements must be met:

Updated kernel used as the default kernel of system start-up.

The original kernel is still valid and can be guided when system starts up.

ANSWER: See the explanation for the answer

Explanation:

Configure the supplied update location as a repository, install the newest `kernel` package, and explicitly select its generated boot entry as the default. Do **not** erase the currently installed kernel.

```
cat > /etc/yum.repos.d/kernel-update.repo <<'EOF'
[kernel-update]
name=Kernel update repository
baseurl=http://server.domain11.example.com/pub/updates/
enabled=1
```

```
gpgcheck=0
EOF

dnf clean all
dnf -y install kernel
```

On RHEL 7, use `yum`dnf:

```
yum clean all
yum -y install kernel
```

Find the highest-version installed kernel and make its boot image the default:

```
newkernel=$(rpm -q kernel --qf '%{VERSION}-%{RELEASE}.%{ARCH}\n' | sort -V | tail -n 1)
grubby --set-default /boot/vmlinuz-${newkernel}

grubby --default-kernel
rpm -q kernel
```

The output of `grubby --default-kernel` must be `/boot/vmlinuz-${newkernel}`. The `rpm -q kernel` output should show both the new kernel and the original kernel, so the original remains available from the GRUB boot menu.

If the URL is only a flat directory of RPM files rather than a YUM/DNF repository, download the appropriate newer `x86_64` kernel RPM(s) from that directory and install them without upgrading/removing the old package:

```
rpm -ivh kernel-*.rpm
```

Then use the same `grubby --set-default` command above to select the new kernel.

QUESTION NO: 3 - (SIMULATION)

SIMULATION

Create User Account.

Create the following user, group and group membership:

Adminuser group

User natasha, using adminuser as a sub group

User Harry, also using adminuser as a sub group

User sarah, can not access the SHELL which is interactive in the system, and is not a member of adminuser, natasha，harry，sarah password is redhat.

ANSWER: See the explanation for the answer

Explanation:

Create the `adminuser` group, then create `natasha` and `harry` with that group as a supplementary group. Create `sarah` without that supplementary group and assign her the noninteractive shell.

```
groupadd adminuser
useradd -G adminuser natasha
useradd -G adminuser harry
useradd -s /sbin/nologin sarah

echo 'natasha:redhat' | chpasswd
echo 'harry:redhat' | chpasswd
echo 'sarah:redhat' | chpasswd
```

Verify the required memberships and Sarah's shell if needed:

```
id natasha
id harry
id sarah
grep '^sarah:' /etc/passwd
```

`natasha` and `harry` must show `adminuser` in their supplementary groups.
`sarah` must not be a member of `adminuser`, and her login shell should be `/sbin/nologin`.

QUESTION NO: 4 - (SIMULATION)

Part 1 (on Node1 Server)

Task 16 [Running Containers]

Configure your host journal to store all journal across reboot

Copy all journal files from /var/log/journal/ and put them in the /home/shangrila/container-logserver

Create and mount /home/shangrila/container-logserver as a persistent storage to the container as /var/log/ when container start

ANSWER: See the explanation for the answer

Explanation:

On `node1`, configure journald for persistent storage, copy the existing persistent journal, and recreate `logserver` with the required bind mount.

```
# Make the host systemd journal persistent
sudo mkdir -p /var/log/journal
sudo mkdir -p /etc/systemd/journald.conf.d
sudo tee /etc/systemd/journald.conf.d/persistent.conf >/dev/null <<'EOF'
[Journal]
Storage=persistent
EOF
sudo systemctl restart systemd-journald

# Create the requested host directory and copy every journal file into it
mkdir -p /home/shangrila/container-logserver
sudo cp -a /var/log/journal/. /home/shangrila/container-logserver/
sudo chown -R shangrila:shangrila /home/shangrila/container-logserver
```

A container mount cannot be added to an already-created container. Stop and remove the original container, then create it again with `/home/shangrila/container-logserver` mounted at exactly `/var/log`. The `:Z` option applies a private SELinux container label to the host directory.

```
podman stop logserver
podman rm logserver

podman run -d --name logserver \
  -v /home/shangrila/container-logserver:/var/log:Z \
  registry.domain15.example.com:5000/rhel8/rsyslog:latest
```

If the container must also be started automatically after a reboot, create and enable a rootless systemd user service:

```
mkdir -p ~/.config/systemd/user
podman generate systemd --new --name logserver --files
mv container-logserver.service ~/.config/systemd/user/
systemctl --user daemon-reload
loginctl enable-linger shangrila
systemctl --user enable --now container-logserver.service
```

Verify the required mount and persistent journal setting:

```
podman inspect logserver --format '{{range .Mounts}}{{println .Source "->"
.Destination}}{{end}}'
grep -r '^Storage=persistent' /etc/systemd/journald.conf /etc/systemd/journald.conf.d/
```

The inspection output must show `/home/shangrila/container-logserver -> /var/log`.

QUESTION NO: 5 - (SIMULATION)

SIMULATION

Create a logical volume

Create a new logical volume as required:

Name the logical volume as database, belongs to datastore of the volume group, size is 50 PE. Expansion size of each volume in volume group datastore is 16MB.

Use ext3 to format this new logical volume, this logical volume should automatically mount to /mnt/database

ANSWER: See the explanation for the answer

Explanation:

Use the unused disk/partition assigned by the simulation (shown below as `/dev/vdb1`). The required LV is `/dev/datastore/database`, with 50 physical extents. Since the VG extent size is 16 MiB, the LV capacity will be 800 MiB.

```
# If the physical volume and VG do not already exist:
pvcreate /dev/vdb1
vgcreate -s 16M datastore /dev/vdb1

# Create the requested 50-PE logical volume
lvcreate -l 50 -n database datastore

# Create the ext3 filesystem and mount point
mkfs.ext3 /dev/datastore/database
mkdir -p /mnt/database

# Make the mount persistent
UUID=$(blkid -s UUID -o value /dev/datastore/database)
echo "UUID=$UUID /mnt/database ext3 defaults 0 0" >> /etc/fstab
mount -a

df -hT /mnt/database
```

If `datastore` already exists, do not run `vgcreate`; verify that it has the required 16 MiB extent size with:

```
vgs -o vg_name,vg_extent_size datastore
```

Then run `lvcreate`, format the LV, and add the `/etc/fstab` entry as shown. An existing VG's physical extent size cannot be changed after creation, so a pre-existing `datastore` must already use 16 MiB extents.

QUESTION NO: 6 - (SIMULATION)

SIMULATION

Add a swap partition.

Adding an extra 500M swap partition to your system, this swap partition should mount automatically when the system starts up. Don't remove and modify the existing swap partitions on your system.

ANSWER: See the explanation for the answer

Explanation:

Create a **new** 500 MiB partition only in unused disk space (do not alter any existing swap partition). First identify the disk with available space:

```
lsblk -f
sudo fdisk -l
```

For example, if `/dev/vdb` is the disk with free space, create a 500 MiB Linux swap partition:

```
sudo fdisk /dev/vdb
```

In `fdisk`, enter the following (choose the default free start sector):

```
n
p
<Enter>
<Enter>
+500M
t
82
w
```

For a GPT disk, set the new partition type to `Linux swap` instead of using MBR type `82`. Assuming the new partition is `/dev/vdb1`, initialize and enable it:

```
sudo partprobe /dev/vdb
sudo mkswap /dev/vdb1
sudo swapon /dev/vdb1
```

Obtain its UUID and add it to `/etc/fstab` so it is activated automatically at boot:

```
sudo blkid /dev/vdb1
sudo vi /etc/fstab
```

Add this line, substituting the UUID returned by `blkid`:

```
UUID=<new-swap-partition-UUID> none swap defaults 0 0
```

Verify the configuration:

```
sudo swapon --show
free -h
sudo mount -a
```

Answer: The new 500 MiB partition must be formatted with `mkswap`, enabled with `swapon`, and recorded in `/etc/fstab` using its UUID. Replace `/dev/vdb`/`/dev/vdb1` above with the actual disk and newly created partition shown in the simulation.

QUESTION NO: 7 - (SIMULATION)

SIMULATION

Your System is going use as a router for 172.24.0.0/16 and 172.25.0.0/16. Enable the IP Forwarding.

1. echo "1" >/proc/sys/net/ipv4/ip_forward

2. vi /etc/sysctl.conf net.ipv4.ip_forward=1

ANSWER: See the explanation for the answer

Explanation:

Enable IPv4 forwarding immediately and make it persistent across reboots.

```
echo 1 > /proc/sys/net/ipv4/ip_forward

echo 'net.ipv4.ip_forward = 1' >> /etc/sysctl.conf
```

Alternatively, apply the persistent setting immediately with:

```
sysctl -w net.ipv4.ip_forward=1
grep -q '^net.ipv4.ip_forward' /etc/sysctl.conf \
  && sed -i 's/^net\.ipv4\.ip_forward.*/net.ipv4.ip_forward = 1/' /etc/sysctl.conf \
  || echo 'net.ipv4.ip_forward = 1' >> /etc/sysctl.conf
```

`/proc/sys/net/ipv4/ip_forward` changes the running kernel setting, while `/etc/sysctl.conf` ensures the setting is restored at boot.

QUESTION NO: 8 - (SIMULATION)

SIMULATION

SELinux must be running in the Enforcing mode.

ANSWER: See the explanation for the answer

Explanation:

Set SELinux to enforcing both immediately and persistently.

```
sudo setenforce 1
sudo sed -i 's/^SELINUX=.*/SELINUX=enforcing/' /etc/selinux/config
getenforce
```

The final command must display:

```
Enforcing
```

`setenforce 1` changes the current running mode. Setting `SELINUX=enforcing` in `/etc/selinux/config` ensures SELinux remains enforcing after reboot.

QUESTION NO: 9 - (SIMULATION)

SIMULATION

Set cronjob for user natasha to do /bin/echo hiya at 14:23.

ANSWER: See the explanation for the answer

Explanation:

Create a per-user crontab entry for `natasha` that runs every day at 14:23:

```
crontab -u natasha -e
```

Add this line, then save and exit the editor:

```
23 14 * * * /bin/echo hiya
```

Alternatively, as root, install it noninteractively:

```
echo '23 14 * * * /bin/echo hiya' | crontab -u natasha -
```

Verify the configured job:

```
crontab -u natasha -l
```

QUESTION NO: 10 - (SIMULATION)

Configure a user account.

Create a user iar，uid is 3400. Password is redhat

ANSWER: See the explanation for the answer

Explanation:

Create the account with the required UID, then set its password:

```
useradd -u 3400 iar
passwd iar
```

When prompted by `passwd`, enter `redhat` twice.

Noninteractive alternative:

```
useradd -u 3400 iar
echo 'iar:redhat' | chpasswd
```

Verify:

```
id iar
```

The output should show `uid=3400(iar)`.

QUESTION NO: 11 - (SIMULATION)

SIMULATION

A YUM repository has been provided at http://server.domain11.example.com/pub/x86_64/Server.

Configure your system to use this location as a default repository.

ANSWER: See the explanation for the answer

Explanation:

Create a YUM/DNF repository definition pointing to the provided URL. On current Red Hat systems, `yum` is a DNF-compatible command, so either command can validate the configuration.

```
sudo tee /etc/yum.repos.d/base.repo >/dev/null <<'EOF'
[base]
name=Base repository
baseurl=http://server.domain11.example.com/pub/x86_64/Server
enabled=1
gpgcheck=0
EOF

sudo yum clean all
sudo yum repolist
sudo yum list available
```

The required repository settings are:

Repository ID: `base`
Base URL: `http://server.domain11.example.com/pub/x86_64/Server`
Enabled: `1`
GPG checking: `0`

If `yum repolist` displays the `base` repository and `yum list available` returns package information, the default repository is configured correctly.

QUESTION NO: 12 - (SIMULATION)

Configure AutoFS so that accessing /shares/projects mounts server1:/srv/nfs/projects.

ANSWER: See the explanation for the answer

Explanation:

Install and configure `autofs` with an indirect map for `/shares`:

```
dnf install -y autofs
mkdir -p /shares
printf '%s\n' '/shares /etc/auto.shares' >> /etc/auto.master
printf '%s\n' 'projects -fstype=nfs,rw server1:/srv/nfs/projects' > /etc/auto.shares
systemctl enable --now autofs
```

Trigger and verify the on-demand mount by accessing the path:

```
ls /shares/projects
```

The master map line makes `/shares` the AutoFS indirect-map directory. The `projects` entry then causes access to `/shares/projects` to mount `server1:/srv/nfs/projects`.

QUESTION NO: 13 - (SIMULATION)

Configure HTTP service access on a non-standard port 82 and allow it through SELinux and firewalld.

ANSWER: See the explanation for the answer

Explanation:

Configure Apache HTTPD to listen on TCP port 82, label the port for SELinux, and open it in the active firewalld zone.

```
echo 'Listen 82' >> /etc/httpd/conf/httpd.conf
semanage port -a -t http_port_t -p tcp 82
firewall-cmd --permanent --add-port=82/tcp
firewall-cmd --reload
systemctl restart httpd
```

If the system must also retain standard HTTP (TCP/80) access, add the HTTP firewalld service as well:

```
firewall-cmd --permanent --add-service=http
firewall-cmd --reload
```

`http_port_t` authorizes processes running in the `httpd_t` SELinux domain to bind to port 82. Verify the configuration with:

```
semanage port -l | grep http_port_t
firewall-cmd --list-ports
ss -ltnp | grep ':82'
```

If port 82 already has an SELinux port entry, modify it rather than adding a duplicate:

```
semanage port -m -t http_port_t -p tcp 82
```

QUESTION NO: 14 - (SIMULATION)

SIMULATION

There is a server having 172.24.254.254 and 172.25.254.254. Your System lies on 172.24.0.0/16. Make successfully ping to 172.25.254.254 by Assigning following IP: 172.24.0.x where x is your station number.

ANSWER: See the explanation for the answer

Explanation:

Configure the active Ethernet connection with the station-specific address `172.24.0.x/16`, where `x` is the station number assigned for this system. Use `172.24.254.254` as the default gateway; it is the router that can reach `172.25.254.254`.

```
nmcli device status
nmcli connection show
nmcli connection modify <connection-name> ipv4.method manual ipv4.addresses 172.24.0.x/16
ipv4.gateway 172.24.254.254
nmcli connection down <connection-name>
nmcli connection up <connection-name>
ping -c 4 172.25.254.254
```

Replace `<connection-name>` with the connection associated with the Ethernet interface shown by `nmcli device status`, and replace `x` with the assigned station number. For example, station 12 uses `172.24.0.12/16`.

DNS is not required because the destination is specified as an IP address. Verify the resulting route if needed:

```
ip addr
ip route
```

The default route should be via `172.24.254.254`, and the ping to `172.25.254.254` should succeed.

QUESTION NO: 15 - (SIMULATION)

Copy /etc/fstab document to /var/TMP directory. According the following requirements to configure the permission of this document.

The owner of this document must be root.
This document belongs to root group.
User mary have read and write permissions for this document.
User alice have read and execute permissions for this document.
Create user named bob, set uid is 1000. Bob have read and write permissions for this document.
All users has read permission for this document in the system.

ANSWER: See the explanation for the answer

Explanation:

Run the following commands as `root`. The question specifies the case-sensitive path `/var/TMP`, so create it first if it does not already exist.

```
mkdir -p /var/TMP
cp /etc/fstab /var/TMP/fstab
chown root:root /var/TMP/fstab
chmod 0644 /var/TMP/fstab
useradd -u 1000 bob
setfacl -m u:mary:rw,u:alice:rx,u:bob:rw,m::rwx /var/TMP/fstab
```

This results in:

Owner: `root`
Group: `root`
`mary`: read/write ACL permission
`alice`: read/execute ACL permission
`bob` (UID `1000`): read/write ACL permission
All other users: read permission through mode `other::r--`

Verify the final permissions with:

```
getfacl /var/TMP/fstab
```

If the intended directory is the normal system directory `/var/tmp` rather than the literal uppercase path in the question, use `/var/tmp/fstab` in each command instead.

QUESTION NO: 16 - (SIMULATION)

Who ever creates the files/directories on a data group owner should automatically be in the same group owner as data.

ANSWER: See the explanation for the answer

Explanation:

Set the **setgid (SGID)** permission on the `/data` directory. Files and subdirectories subsequently created in it will inherit the directory's group owner.

```
chmod g+s /data
```

Verify the setting:

```
ls -ld /data
```

The group permission field should contain `s`, for example:

```
drwxrws---. root sysadmin /data
```

Here, new content created under `/data` receives group ownership `sysadmin` (the group of `/data`), regardless of the creating user's primary group.

QUESTION NO: 17 - (SIMULATION)

SIMULATION

Configure the system synchronous as 172.24.40.10.

ANSWER: See the explanation for the answer

Explanation:

Configure the host to synchronize its time with the NTP server `172.24.40.10`.

```
sudo vi /etc/chrony.conf
# Add or replace the server entry with:
server 172.24.40.10 iburst

sudo systemctl enable --now chronyd
sudo chronyc makestep
chronyc sources -v
```

Ensure any default public `server` or `pool` entries are removed or commented out if the required time source must be only `172.24.40.10`. The `chronyc sources -v` output should eventually show `172.24.40.10` as the selected source (marked with `^*`).

On older RHEL systems using the `ntpd` service instead of Chrony, add `server 172.24.40.10 iburst` to `/etc/ntp.conf`, then run:

```
sudo chkconfig ntpd on
sudo service ntpd restart
```

QUESTION NO: 18 - (SIMULATION)

SIMULATION

Configure your system so that it is an NTP client of server.domain11.example.com

ANSWER: See the explanation for the answer

Explanation:

Configure `chronyd` to use `server.domain11.example.com` as its NTP source.

```
sudo vi /etc/chrony.conf
```

Remove or comment existing `server`/`pool` source entries, then add:

```
server server.domain11.example.com iburst
```

Enable and restart the service:

```
sudo systemctl enable --now chronyd
sudo systemctl restart chronyd
```

Verify that the configured server is being used or reached:

```
chronyc sources -v
chronyc tracking
```

The source list should show `server.domain11.example.com`; after synchronization, one source is normally marked with `^*`.

QUESTION NO: 19 - (SIMULATION)

Find all files owned by user tom and copy them into /root/tomfiles.

ANSWER: See the explanation for the answer

Explanation:

Create the target directory, then locate regular files owned by `tom` and copy them into it:

```
mkdir -p /root/tomfiles
find / -type f -user tom -exec cp -p {} /root/tomfiles/ \;
```

`find /` searches from the filesystem root, `-type f` limits results to regular files, and `-user tom` selects files whose owner is `tom`. The `cp -p` command copies each matching file while preserving its metadata where possible.

QUESTION NO: 20 - (SIMULATION)

SIMULATION

Adjust the size of the Logical Volume.

Adjust the size of the vo Logical Volume, its file system size should be 290M. Make sure that the content of this system is complete.

Note: the partition size is rarely accurate to the same size as required, so in the range 270M to 320M is acceptable.

ANSWER: See the explanation for the answer

Explanation:

Set `/dev/vg0/vo` to approximately `290M`. First identify its filesystem type and mount point:

```
lsblk -f /dev/vg0/vo
findmnt -S /dev/vg0/vo
lvs /dev/vg0/vo
```

For an **ext4** filesystem that must be reduced, preserve the filesystem contents by checking and shrinking the filesystem *before* reducing the logical volume. Replace `/mountpoint` with the result of `findmnt`:

```
umount /mountpoint
e2fsck -f /dev/vg0/vo
resize2fs /dev/vg0/vo 290M
lvreduce -L 290M /dev/vg0/vo
mount /dev/vg0/vo /mountpoint
df -hT /mountpoint
```

The final `df` size may not display exactly 290M because of filesystem metadata; a size from 270M through 320M is acceptable.

If the existing logical volume is instead smaller than 290M, extend it and grow the filesystem online:

```
lvextend -L 290M -r /dev/vg0/vo
df -hT /mountpoint
```

Important: do not run `lvreduce` before `resize2fs`, since doing so can destroy filesystem data. An XFS filesystem cannot be reduced in place; only use the growth command for XFS, or back up its contents, recreate a 290M filesystem, restore the data, and remount it if a reduction is required.

QUESTION NO: 21 - (SIMULATION)

Create /share/projects, make alex the owner, and configure the sticky bit so users cannot delete each other's files.

ANSWER: See the explanation for the answer

Explanation:

Create the shared directory, set `alex` as its owner, and make it world-writable with the sticky bit enabled:

```
mkdir -p /share/projects
chown alex /share/projects
chmod 1777 /share/projects
```

Verify the result:

```
ls -ld /share/projects
```

The permissions should resemble:

```
drwxrwxrwt. alex ... /share/projects
```

`1777` gives read, write, and execute permission to all users.
The leading `1` enables the sticky bit, displayed as `t` in the other-users execute position.
With the sticky bit set, users may create files in the directory but cannot delete or rename files owned by other users; only the file owner, directory owner (`alex`), or `root` can do so.