1 **Reset root password**

This is critical - if you don't know how to reset the root password you will not be able to complete the rest of the test.  
  
a. Enter Emergency Mode  
    - Reboot and enter grub edit menu by pressing the 'e' button  
    - Find the linux16 or linuxefi line and add **rd.break** to the end of the line  
    - Start the boot process by pressing **CTRL+x**  
  
b. Reset the password  
    - Remount the sysroot file system in r/w mode and then use chroot at /sysroot  
      ─────────────────────────────  
   mount -o remount,rw  /sysroot  
   chroot /sysroot  
   ─────────────────────────────  
  
    - Change the password  
      ──────  
   passwd  
   ──────  
  
    - Ensure all unlabeled files get relabeled during boot  
      ───────────────────  
   touch /.autorelabel  
   ───────────────────  
  
    - Resume boot by issuing the exit command twice

**2 Set network settings and hostname**

Spoiler

This step has to be completed before anything else can be done on the test. You might not have access to a GUI yet.  
  
a. Configure Network  
   The easiest/fastest way to do this is to use the nmtui command.  
  
   CRITICAL:  
   When configuring an IP address in nmtui don't forget to set the netmask via <IP-ADDRESS>/<SUBNET>.  
   If the subnet is not defined manually then nmtui defaults to <IP-ADDRESS>/32  
   This will cause problems when trying to access the yum repositories later.  
  
   b. Set hostname  
      ───────────────────────────────────────  
   hostnamectl set-hostname <NEW-HOSTNAME>  
   ───────────────────────────────────────

**3 Set SELinux into enforcing mode**

Spoiler

a. Set current SELinux to Enforcing  
    ────────────────────  
  setenforce Enforcing  
  ────────────────────  
  
b. Change the default mode to enforcing  
    ──────────────────────  
  vi /etc/selinux/config

**4 Enable yum repository**

Spoiler

Configure the machine to access a yum repo at server.example.com/update  
  
a. Create repo file /etc/yum.repos.d/update.repo  
   Contents:  
    ─────────────────────────────────────────────  
  [update]  
  name=update  
  baseurl=http://server.example.com/update  
  gpgcheck=0  
  enabled=1  
  ─────────────────────────────────────────────  
  
   Note:  
   I'm using the gpgcheck=0 option to make things faster.  
   If you're instructed to actually use gpgcheck then you'll need to import the GPG key from the online repository.  
  
   You can do this by downloading the file (wget or through browser) and then installing it using rpm.  
   Ex.  
   wget <http://server.example.com/update/RPM-GPG-KEY-redhat-release> /tmp/RPM-GPG-KEY-redhat-releas  
   rpm --import RPM-GPG-KEY-redhat-release  
  
b. Verify that your new repo is available  
   yum repolist  
  
c. Verify access to the repo by installing something  
   In this case I'll install the user GUI tools (they'll be handy later)  
   yum install system-config-\*

5  **User creation**

Spoiler

Create 3 users - larry, curly, and moe.  
All 3 users belong to supplemental group stooges.  
User moe should have a shell of nologin  
  
You'll need to pay attention to the wording here. The instructions say the users have to belong to a supplemental group stooges - that means they need to also belong to a primary group. The most logical assumption is that they each belong to their own private group (ex. larry:larry, curly:curly, etc…)  
  
You can install the system-config-users package and use the GUI to create the users.

6 **Create a shared folder**

Spoiler

Create a directory that members of the stooges group can access.  
Ensure that files created in the directory are owned by the stooges group.  
  
The solution is to use the SGID bit on the directory  
────────────────────────────────────────────────  
mkdir /tmp/shared  
chown :stooges /tmp/shared  
chmod g+s /tmp/shared OR chmod 2775 /tmp/shared  
────────────────────────────────────────────────

7 **Activate and start services via systemctl**

Spoiler

Fairly simple question. You'll need to use the systemctl enable and the systemctl start commands.  
  
I would also recommend enabling the GUI interface while you're at it.  
Ex.  
─────────────────────────────────  
systemctl enable graphical.target  
systemctl start graphical.target  
─────────────────────────────────  
  
Note:  
I've had the above command hang during the start of the GUI. The workaround was to enable graphical target and then reboot.  
This command would enable the GUI and the reboot would cause the system to use it.

8 **Find all files owned by user bob and copy them to a newly created directory /tmp/bobs\_files**

Spoiler

You can do this any way you want. The easiest is to pipe the output of find to xargs  
──────────────────────────────────────────────────────  
mkdir /tmp/bobs\_files  
find / -user bob | xargs -I {} cp {} /tmp/bobs\_files/  
──────────────────────────────────────────────────────

9  **Find strings matching xyz in the file /tmp/test-log-file and save them in /tmp/matching-log-entries**

Spoiler

Just use grep and redirect the output  
────────────────────────────────────────────────────────  
grep xyz /tmp/test-log-file > /tmp/matching-log-entries  
────────────────────────────────────────────────────────

10  **Create a gzipped tar archive (/tmp/log\_archive.tgz) containing all the files in /var/log**

Spoiler

Just use the tar command  
────────────────────────────────────────  
tar -zcvf /tmp/log\_archive.tgz /var/log  
────────────────────────────────────────

11  **Use ACLs on files**

Spoiler

Create file /tmp/acl\_file  
The file is owned by bob:bob  
Allow larry and curly to rwx the file  
Don’t allow moe to access the file (rwx)  
All member of stooges (except moe) should be able to access the file (rw)  
  
a. Create the file  
    ───────────────────  
  touch /tmp/acl\_file  
  ───────────────────  
  
b. Set the ownership  
    ───────────────────────────  
  chown bob:bob /tmp/acl\_file  
  ───────────────────────────  
  
c. Allow larry/curly access  
    ────────────────────────────────────  
  setafcl -m u:larry:rwx /tmp/acl\_file  
  setafcl -m u:curly:rwx /tmp/acl\_file  
  ────────────────────────────────────  
  
d. Deny moe access  
    ────────────────────────────────  
  setfacl -m u:moe:- /tmp/acl\_file  
  ────────────────────────────────  
  
e. Allow members of stooges access  
    ──────────────────────────────────────  
  setfacl -m g:stooges:rw  /tmp/acl\_file  
  ──────────────────────────────────────

12  **Update the kernel via new YUM-repo**

Spoiler

A kernel update is available in a different location (not the first yum repo)  
  
a. Create a yum repo for the new location (refer to question #4)  
  
b. Issue the command to update the kernel  
    ─────────────────  
  yum update kernel  
  ─────────────────  
  
c. Reboot to enable new kernel (visually verify that a new grub entry was created)  
    ──────  
  reboot  
  ──────

13   **Create a Cronjob for bob that runs /bin/date daily at 3:18pm and redirects the output to /home/bob/stamp**

Spoiler

a. Modify bob crontab entry  
    ───────────────  
  crontab -eu bob  
  ───────────────  
  
b. Add the command  
    ────────────────────────────────────────  
  18 15 \* \* \* /bin/date >> /home/bob/stamp  
  ────────────────────────────────────────

14 **Create a new 800MiB swap partition**

Spoiler

The test VM has a single disk with 3 partitions  
P1 - Boot  
P2 - LVM  
P3 - LVM  
  
The question didn't specify that the new swap partition had to be part of the LVM so the easiest way to do this is to create another partition to use as a swap partition  
  
a. Use fdisk to create the partition (use +800M for Last Sector)  
b. Issue the command **blkid** to get the UUID for the newly created disk  
c. Add an entry to fstab (ex. UUID=<blkid-output> swap swap 0 0)  
d. Issue the command **swapon -a** to enable the newly added partition  
  
Note:  
I had to redo this step with an extended partition instead of a primary one to create extra space for extending LVM.  
This is probably by design - the assumption while answering this question is that I can just create an extra partition,  
that assumption is later changed since we'll also need another partition to use when extending the LVM.  
Ensure that you know how to create extended partitions in fdisk.

15  **Create a new logical volume cattle with a size of 25 extents belonging to volume group farming with a PE size of 32M**

Spoiler

This is fairly easy once you break down the question into its parts  
  
a. Create a new PV  
    This is where we would create a new partition, running into the extended vs primary issue mentioned in the swap creation answer.  
  
b. Create a new VG  
    Use the vgcreate command with the flags:  
    -s or --physicalextentsize set to 32M  
  
c. Create new LV  
    Use lvcreate with the flags:  
    -n or --name to provide the name  
    -l or --extents to provide the size  
  
d. Create filesystem on the new LV  
    Use mkfs with the -t flag  
  
e. Mount LV and add an entry to fstab.

16  **Create a connection to LDAP**

Spoiler

LDAP is used for users and authentication  
A certificate is required (available for download)  
  
I just used the GUI system-config-auth - it's far easier than trying to fiddle with all the options manually.

17 **Use the NFS-automounter for specific folder**

Spoiler

At first glance this seemed like an easy question but it really isn't. The hard part here is that the directory used for mounting is defined in the root path (ex. /my\_auto\_mount) which is not defined in the SELinux fcontext for autofs.  
  
You'll need to:  
a. Configure automount (look it up)  
b. Add a new context to SELinux (semanage fcontext)  
c. Refresh the context of the new directory (restorecon)  
d. Start autofs and set to enable (systemctl enable & systemctl start)

18  **Configure NTP to synchronize with time.example.com**

Spoiler

You'll need to:  
a. Add the entry to /etc/chrony.conf  
b. Restart the service and ensure it's enabled (systemctl)  
c. Query chrony to verify (chronyc sources -v)