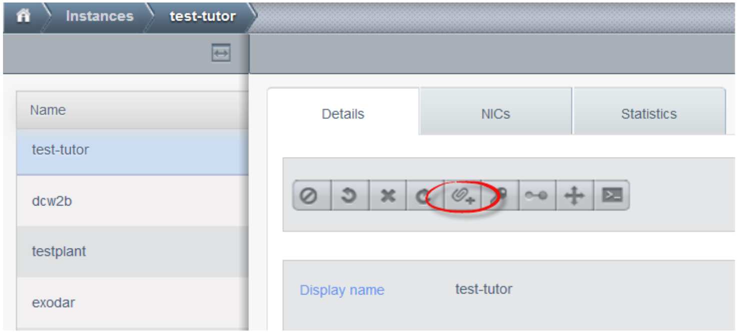
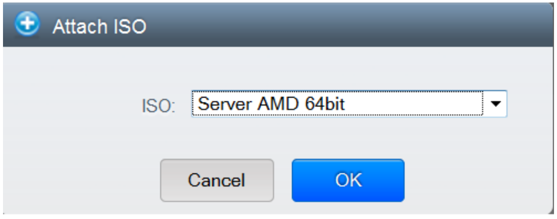
Changing user login password on templates

1. Attach an ISO via the cloudstack interface and boot the VM to rescue mode. Click on “attach iso” icon highlighted below:



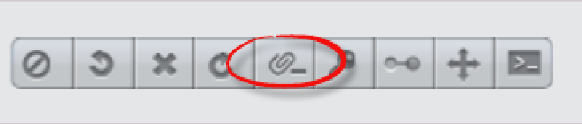
Display name: test-tutor

A popup window appears from which select the ISO from the descriptions available:

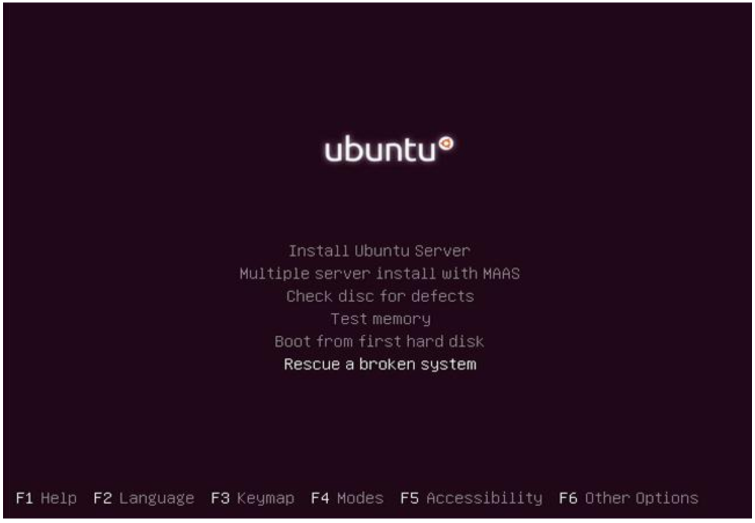


ISO: Server AMD 64bit

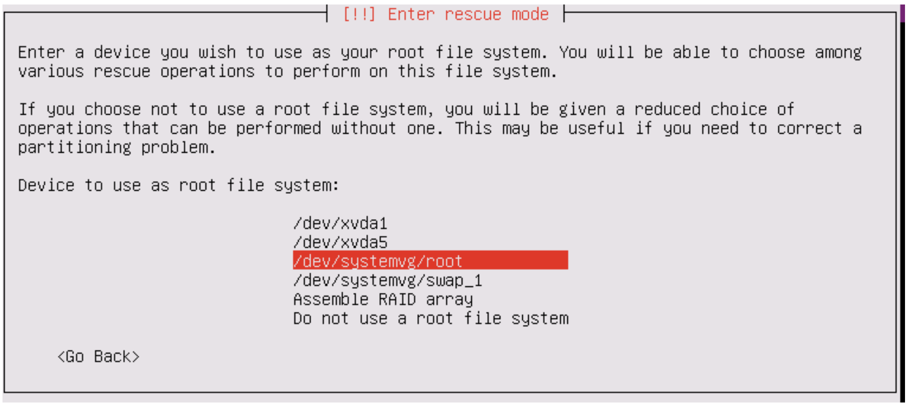
As soon as the ISO image is successfully attached the “attach iso” icon changes:



Reboot the VM and use the VM console to handle the rest of the tasks.
Select Language
Select – Rescue a Broken System

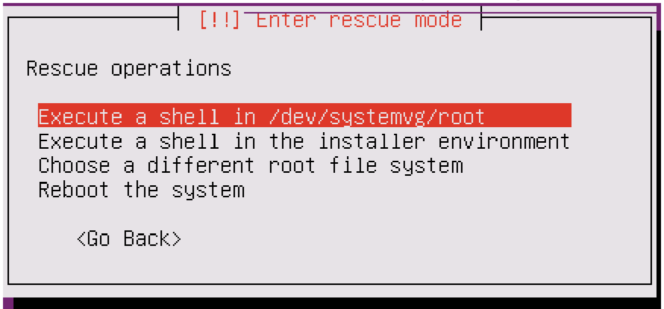

ubuntu
Install Ubuntu Server
Multiple server install with MAAS
Check disc for defects
Test memory
Boot from first hard disk
Rescue a broken system
F1 Help F2 Language F3 Keymap F4 Modes F5 Accessibility F6 Other Options

Select Language
Select Location
Select Keyboard
Wait for the OS to load additional components,
Configure the network
Configure the clock
Select the device to use as root file system (as shown below):

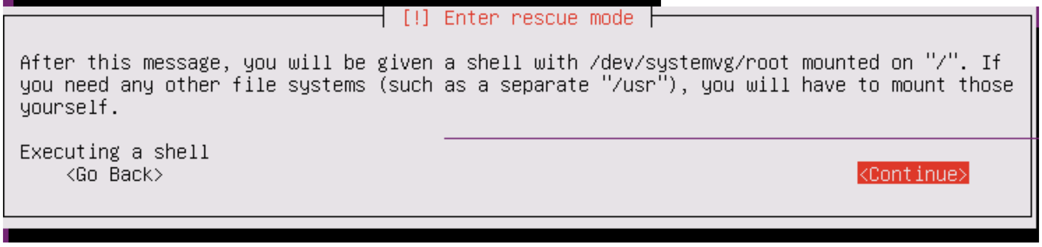


```
[!!] Enter rescue mode

Enter a device you wish to use as your root file system. You will be able to choose among
various rescue operations to perform on this file system.

If you choose not to use a root file system, you will be given a reduced choice of
operations that can be performed without one. This may be useful if you need to correct a
partitioning problem.

Device to use as root file system:

                              /dev/xvda1
                              /dev/xvda5
                              /dev/systemvg/root
                              /dev/systemvg/swap_1
                              Assemble RAID array
                              Do not use a root file system

    <Go Back>
```

It's your choice if you want to mount the /boot partition.
Select "execute a shell in /dev/systemvg/root"

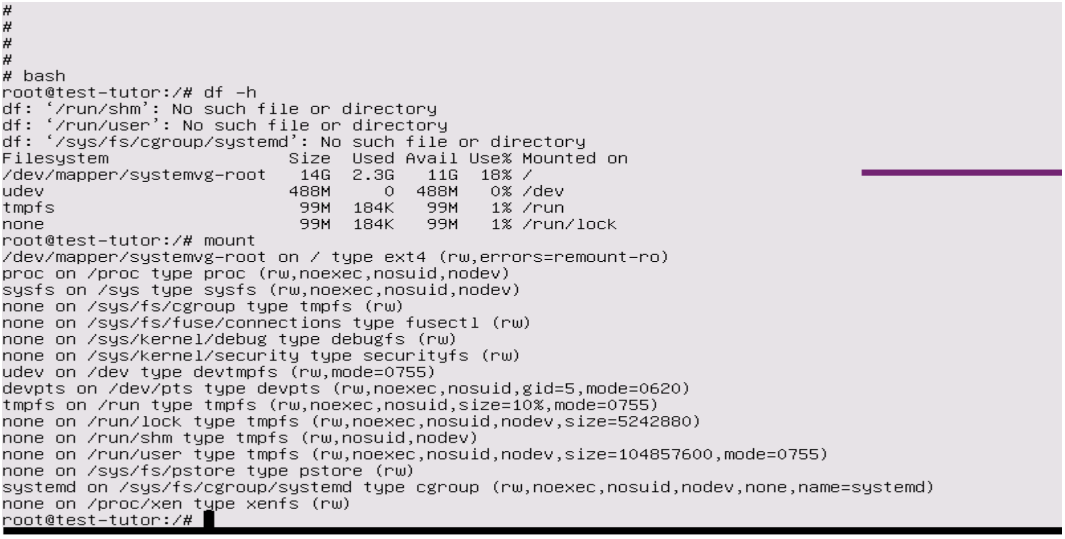


```
[!!] Enter rescue mode

Rescue operations

 Execute a shell in /dev/systemvg/root
 Execute a shell in the installer environment
 Choose a different root file system
 Reboot the system

     <Go Back>
```



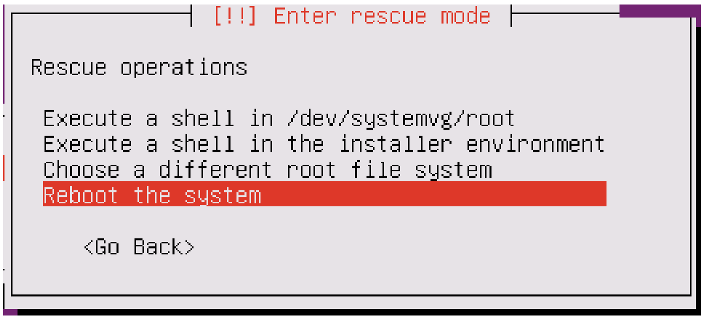
```
[!] Enter rescue mode

After this message, you will be given a shell with /dev/systemvg/root mounted on "/". If
you need any other file systems (such as a separate "/usr"), you will have to mount those
yourself.

Executing a shell
    <Go Back>                                                             <Continue>
```

You are now provided with a shell prompt, as root user.



```
#
#
#
#
# bash
root@test-tutor:/# df -h
df: '/run/shm': No such file or directory
df: '/run/user': No such file or directory
df: '/sys/fs/cgroup/systemd': No such file or directory
Filesystem                   Size  Used Avail Use% Mounted on
/dev/mapper/systemvg-root     14G  2.3G   11G  18% /
udev                         488M     0  488M   0% /dev
tmpfs                         99M  184K   99M   1% /run
none                          99M  184K   99M   1% /run/lock
root@test-tutor:/# mount
/dev/mapper/systemvg-root on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/cgroup type tmpfs (rw)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
none on /run/user type tmpfs (rw,noexec,nosuid,nodev,size=104857600,mode=0755)
none on /sys/fs/pstore type pstore (rw)
systemd on /sys/fs/cgroup/systemd type cgroup (rw,noexec,nosuid,nodev,none,name=systemd)
none on /proc/xen type xenfs (rw)
root@test-tutor:/#
```

2. Run the command "passwd <user>" and enter the new password. E.g.
   If your user login is guest: # passwd guest
3. Sync the filesystem and shutdown the VM.
   # sync ; exit; exit


[!!] Enter rescue mode

Rescue operations

Execute a shell in /dev/systemvg/root
Execute a shell in the installer environment
Choose a different root file system
Reboot the system
<Go Back>

4. Stop the VM, Detach the ISO and then start the VM.

VM Root file in read only state and fsck doesn’t resolve the issue.

Resolution: Migrate the VM to another host and restart it.

Utilising the extra disk allocated to the VM

If a 10Gb additional disk was added to the VM it would be displayed as a raw device, so if you want to extend the root volume the steps to be followed are as follows:

To display the disks use the command “fdisk”. By default the naming convention for the boot disk would be “xvda” and the additional disk would start with “xvdb”.

```
root@test-tutor:~# fdisk -l

Disk /dev/xvda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00098c7c

     Device Boot      Start         End      Blocks   Id  System
/dev/xvda1             2048      499711      248832   83  Linux
/dev/xvda2           501758    41940991    20719617    5  Extended
/dev/xvda5           501760    41940991    20719616   8e  Linux LVM

Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/xvdb doesn't contain a valid partition table
....
....
```

From the above command output we find a raw device /dev/xvdb of 10Gb, without any filesystems on it. To create a partition on it follow the commands provided below:

```
root@test-tutor:~# fdisk /dev/xvdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
```

```
Building a new DOS disklabel with disk identifier 0x1d342ae5.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): p

Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x1d342ae5

     Device Boot      Start         End      Blocks   Id  System

Command (m for help): n
Partition type:
    p   primary (0 primary, 0 extended, 4 free)
    e   extended
Select (default p): p
Partition number (1-4, default 1): <Press Return for default value:1>
Using default value 1
First sector (2048-20971519, default 2048): <Press Return for default value:2048>
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519): <Press Return
for default value:20971519>
Using default value 20971519

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

I’ve now allocated 10Gb to a single partition which can be verified with the following command.

```
root@test-tutor:~# fdisk -l /dev/xvdb

Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
107 heads, 17 sectors/track, 11529 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x1d342ae5

     Device Boot      Start         End      Blocks   Id  System
/dev/xvdb1            2048    20971519    10484736   83 Linux
```

We now need to create a physical volume (PV) on the new partition which can then be added to the volume group.
To display the configured physical volumes use the command “pvs” or “pvdisplay”.

```
root@test-tutor:~# pvs
   PV         VG       Fmt  Attr PSize  PFree
   /dev/xvda5 systemvg lvm2 a--  19.76g 20.00m
```

Create the physical volume on /dev/xvdb1:

```
root@test-tutor:~# pvcreate /dev/xvdb1
   Physical volume "/dev/xvdb1" successfully created
```

To display the available volume groups, the commands available are “vgs” and “vgdisplay”.

```
root@test-tutor:~# vgs
   VG       #PV #LV #SN Attr   VSize  VFree
   systemvg   1   2   0 wz--n- 19.76g 20.00m
```

We now add the newly created partition to the existing Volume Group (VG) which as seen above is named “systemvg”.

```
root@test-tutor:~# vgextend systemvg /dev/xvdb1
   Volume group "systemvg" successfully extended
```

If we now display the VG, we notice that the size has increased by 10G (VSize) and 10G of non allocated space (VFree)

```
root@test-tutor:~# vgs
   VG       #PV #LV #SN Attr   VSize  VFree
   systemvg   2   2   0 wz--n- 29.75g 10.02g
```

To display the available logical volumes (LV) run the command “lvs” or “lvdisplay systemvg”. We notice the existence of two LVs root and swap_1. Running the “df -h /” command we can confirm that the root vol is mounted from /dev/systemvg/root or the “root” volume available on VG “systemvg”.

```
root@test-tutor:~# lvs
   LV     VG       Attr      LSize  Pool Origin Data%  Move Log Copy%  Convert
   root   systemvg -wi-ao--- 13.75g
   swap_1 systemvg -wi-ao---  5.99g
```

To extend the “root” LV, use the command “lvextend” and the size to increment to, by adding a “+” sign. Note: If the “+” sign is not added the size of the LV would be reduced to 10G, the OS would show an alert requesting you to confirm that you want to reduce the volume.

```
root@test-tutor:~# lvextend -L +10G systemvg/root
   Extending logical volume root to 23.75 GiB
   Logical volume root successfully resized
```

We can confirm that the LV size has now increased by 10Gb.

```
root@test-tutor:~# lvs
   LV     VG       Attr      LSize  Pool Origin Data%  Move Log Copy%  Convert
   root   systemvg -wi-ao--- 23.75g
   swap_1 systemvg -wi-ao---  5.99g
```

If now have run the command “resize2fs” to increase the root volume online.

```
root@test-tutor:~# resize2fs /dev/systemvg/root
resize2fs 1.42.9 (4-Feb-2014)
Filesystem at /dev/systemvg/root is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 2
The filesystem on /dev/systemvg/root is now 6224896 blocks long.
```

To confirm that root partition has been extended use the command “df”

```
root@test-tutor:~# df -h /
```

```
Filesystem                 Size  Used Avail Use% Mounted on
/dev/mapper/systemvg-root   24G  2.3G   20G  11% /
```

A short method of doing all the above steps (as root):

```
pvcreate /dev/xvdb ; vgextend systemvg /dev/xvdb; lvextend -L +10G systemvg/root ;
resize2fs /dev/systemvg/root
```

To reduce the swap space from the current 8Gb to 2Gb for CSG provided templates and increase the root filesystem by 4Gb (as root):

```
swapoff –a                         # Switch of swap volume/device
lvchange -a n systemvg/swap_1      #Deactivate Logical Volume (LV)
lvreduce -L 2G systemvg/swap_1     #Reduce size of LV to 2Gb
lvchange -a y systemvg/swap_1      #Activate LV
mkswap  /dev/systemvg/swap_1       #Create swap device
swapon /dev/systemvg/swap_1        #Switch on swap volume/device
lvextend -L +4G systemvg/root      #Increase root LV by 4G (note the + sign)
resize2fs /dev/systemvg/root       #Resize filesystem to reflect the change
```

To create a separate Logical Volume named “data” using an additional disk.

```
pvcreate /dev/xvdb    # create a physical volume (PV)
vgextend systemvg /dev/xvdb    # add new PV to existing Volume Group (VG)
lvcreate -L +10G -n data systemvg # create a new Logical Volume (LV) named data
mkfs.ext4 /dev/systemvg/data # format the LV
mkdir /data    # create a directory to which LV is to be mounted
mount /dev/systemvg/data /data   # mount LV
df -h /data    # verify mount point
# to create a permanent mount on system boot update the file /etc/fstab
echo "/dev/mapper/systemvg-data /data ext4 defaults 0 0" >> /etc/fstab
umount /data   # unmount the filesystem
mount –a   # the mount command will read data from /etc/fstab
df –h /data   # confirm the mount has occurred, ensures there are no errors in fstab file.
```