**# Boot Migration to Larger Disk on Azure RHEL VM v.003**

**## Objective**

Migrate the root and boot partitions of an AlmaLinux 9.5 VM on Azure to a larger, newly attached disk, ensuring full kernel patching support and functional EFI boot.

---

**## Initial Conditions**

\* Azure VM deployed

\* Larger disk (150GB) attached to the VM.

\* Secure Boot was initially enabled and caused issues with `grub2-install`.

---

***## Step-by-Step Procedure***

**### 1. \*\*Disable Secure Boot via Redeployment\*\***

Terraform was used to recreate the VM with the following parameter:

```hcl

secure\_boot\_enabled = false

```

This ensured compatibility with `grub2-install` and EFI bootloader management.

* Attach new disk to VM

**### 2. \*\*Partition and Format the New Disk (/dev/sdc)\*\***

```bash

fdisk /dev/sdc

g

*EFI partition creation*

n ← create new partition

1 ← partition number

(enter) ← accept default first sector

+512M ← size

t ← change partition type

1 ← select partition 1

1 ← EFI System  
  
*Create Root partition*

n ← create new partition

2 ← partition number

(enter) ← accept default first sector

(enter) ← assign a static amount that it not the total if you want a swap disk (sdc3)

t ← change partition type

2 ← select partition 2

20 ← Linux filesystem

*(Optional) Create third partition*

n ← new partition

3 ← partition number

(enter) ← accept default start

(enter) ← accept default end or type `+XG` for size

t ← change type (if needed)

3

20 ← Linux filesystem or `19` for Linux swap

**# Created: /dev/sdc1 (vfat), /dev/sdc2 (ext4), /dev/sdc3 (ext4)**

mkfs.vfat /dev/sdc1

mkfs.ext4 /dev/sdc2

mkfs.ext4 /dev/sdc3

```

**### 3. \*\*Mount the New Root Filesystem\*\***

```bash

mount /dev/sdc3 /mnt

```

**### 4. \*\*Copy Original System to New Root\*\***

```bash

rsync -aAXv /\* /mnt --exclude={/mnt,/proc,/sys,/tmp,/run,/dev,/boot,/boot/efi}

```

**### 5. \*\*Mount System Bind Mounts\*\***

**Step 5**: Mount system bind mounts — this is **preparation** for chroot. It gives your new root environment access to critical virtual filesystems like /dev, /proc, etc., mimicking a running system.

**Validation of mnt path**  
  
df -h / # Current root

df -h /mnt # New root

```bash

**Making the directory**  
  
 mkdir -p /mnt/{dev,proc,sys,run,boot,boot/efi}

**Mounting / Binding**

mount --bind /dev /mnt/dev

mount --bind /proc /mnt/proc

mount --bind /sys /mnt/sys

mount --bind /run /mnt/run

mount /dev/sdd2 /mnt/boot

mount /dev/sdd1 /mnt/boot/efi

**rsync Boot and EFI Contents**

rsync -aAXHv --delete --info=progress2 --ignore-errors --inplace /boot/ /mnt/boot/

rsync -aAXHv --delete --info=progress2 --ignore-errors --inplace /boot/efi/ /mnt/boot/efi/

**### 6. \*\*Update/backup fstab in New Root\*\***

cp /etc/fstab /etc/fstab.backup

diff /etc/fstab /etc/fstab.backup

```bash

1. **Checking the new partitions**

blkid /dev/sdc\*

1. **# Edit /mnt/etc/fstab to reflect:**

nano /mnt/etc/fstab

UUID=<sdc3> **/**  ext4 defaults 0 1

UUID=<sdc2> **/boot** ext4 defaults 0 2

UUID=<sdc1**> /boot/efi** vfat umask=0077 0 2

```

**### 7. \*\*Chroot Into New System\*\***

```bash

chroot /mnt

```

**### 8. \*\*Install Required EFI Boot Packages\*\***

```bash

dnf install -y grub2-efi-x64 grub2-efi-x64-modules shim efibootmgr grub2-tools-efi grub2-tools-extra --verbose

```

**### 9. \*\*Install EFI Bootloader to /dev/sdc\*\***

```bash

grub2-install --target=x86\_64-efi --efi-directory=/boot/efi --bootloader-id=XXXXXX\_grub --recheck --no-nvram --force

```

**### 10. \*\*Generate GRUB Configuration\*\***

```bash

mkdir /tmp

chmod 1777 /tmp

grub2-mkconfig -o /boot/grub2/grub.cfg

```

**### 11. \*\*Ensure Boot Entry Priority\*\* (optional if not already set)**

Exit chroot (type exit)

```bash

efibootmgr -o <boot\_num\_for\_\_grub>,<others>

```

**### 12. \*\*Shutdown and Verify Boot\*\***

```bashroot

reboot

```

Post-boot verification:

```bash

uname -r

lsblk -f

mount | grep /boot

```

---

**## Issues & Fixes Summary**

| Issue | Resolution |

| --------------------------------------------- | --------------------------------------------------------- |

| `grub2-install` failed due to Secure Boot | VM redeployed with Secure Boot disabled via Terraform |

| `/usr/lib/grub/x86\_64-efi/modinfo.sh` missing | Installed `grub2-efi-x64-modules` package |

| Booted into wrong disk | Used `efibootmgr` to reprioritize EFI boot entries |

| `grub.cfg` not found | Confirmed location was `/boot/efi/EFI/almalinux/grub.cfg` |

---

**## Outcome**

\* Boot is now successfully migrated to the larger 150GB disk.

\* New kernel was installed and is in use.

\* EFI bootloader is correctly configured and operational.

\* The `/boot` partition has more free space to support future kernel upgrades.

---

**## Next Steps**

\* Snapshot or image the VM in its current stable state.

\* Optionally, remove the original disk after full validation.