

Astra Linux Troubleshooting Challenge

Project Overview

This project focuses on diagnosing and recovering an Astra Linux system that failed to boot due to a missing GRUB configuration file. The recovery process involved identifying the root cause, manually booting the operating system from the GRUB shell, restoring bootloader functionality, and verifying successful system recovery.

Objective

- Identify the cause of the Astra Linux boot failure.
- Recover the operating system from the GRUB command line.
- Restore normal boot functionality by recreating the GRUB configuration.
- Verify successful recovery by accessing the required file `/root/abc.sh`.

Prerequisites

- Astra Linux Virtual Machine
- VMware Workstation
- Basic Linux Administration Knowledge
- Access to GRUB Command Line
- Root User Credentials

Issue Description

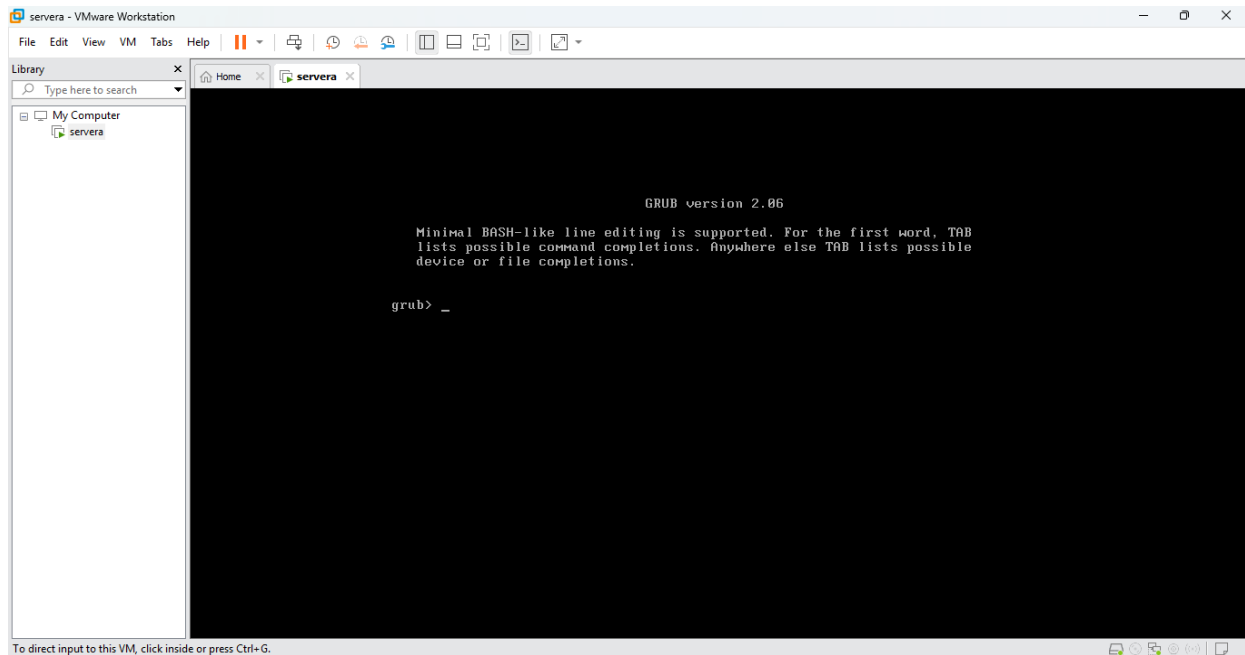
When the Astra Linux virtual machine was powered on, the operating system failed to boot normally and stopped at the GRUB command prompt instead of loading the login screen.

The system displayed: **grub>**

This indicated that the bootloader could not locate the required boot configuration needed to start the operating system.

System Failure Observation

The screenshot below shows the Astra Linux system entering the GRUB command-line interface during startup. This behavior confirmed that the normal boot process had failed and further investigation was required to identify and resolve the issue.



Understanding the Linux Boot Process

The Linux boot process follows these stages:

1. BIOS/UEFI Initialization
2. GRUB Bootloader Execution
3. Linux Kernel Loading
4. Initramfs Loading
5. Systemd Service Initialization
6. User Login Screen

Since the GRUB configuration file was missing, the system could not proceed beyond the bootloader stage and entered the GRUB command prompt

Step 1: List Available Disks



```
# ls
```

→ Displays available disks and partitions detected by GRUB.

Step 2: Inspect the Partition

```
GRUB version 2.06

Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists possible
device or file completions.

grub> ls
(hd0) (hd0,msdos1)
grub> ls (hd0,msdos1)/
dev/ proc/ run/ sys/ tmp/ etc/ root/ var/ usr/ bin sbin lib lib64 afs/ boot/ ho
me/ media/ mnt/ opt/ srv/
grub>
```

```
# ls (hd0,msdos1)/
```

→ Displays the contents of the partition and confirms that it contains the Linux filesystem.

Step 3: Verify Filesystem Information

```
grub> cat (hd0,msdos1)/etc/fstab

#
# /etc/fstab
# Created by anaconda on Sat Jan  3 14:20:00 2026
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=d1aa5bd2-a705-4dcf-b83a-307ce9263a6e /          xfs      defau
lts          0 0
grub> _
```

```
# cat (hd0,msdos1)/etc/fstab
```

→ Reads the fstab file to obtain the root filesystem UUID

Step 4: Check GRUB Files

```
grub> cat (hd0,msdos1)/boot/grub2/grub.cfg
error: ../../grub-core/fs/fshelp.c:257:file '/boot/grub2/grub.cfg' not found.
grub> _
```

cat (hd0,msdos1)/boot/grub2/grub.cfg

(Verifies whether the GRUB configuration file exists)

→ The normal `grub.cfg` file was missing.

```
grub> ls (hd0,msdos1)/boot/grub2/
device.map i386-pc/ locale/ fonts/ grubenv
grub> _
```

ls (hd0,msdos1)/boot/grub2/

→ Lists files available inside the GRUB directory

Step 5: Check Loader Entries

```
grub> ls (hd0,msdos1)/boot/loader/
entries/
grub>
```

ls (hd0,msdos1)/boot/loader/

```
grub> ls (hd0,msdos1)/boot/loader/entries/
22611a6ea130461e97f2aa80d137ec20-5.14.0-611.5.1.el9_7.x86_64.conf 22611a6ea1304
61e97f2aa80d137ec20-0-rescue.conf
grub> _
```

ls (hd0,msdos1)/boot/loader/entries/

→ Identifies installed kernel versions and boot entries. .

Root Cause Analysis

The boot failure occurred because the GRUB configuration file (`/boot/grub2/grub.cfg`) was missing from the system. During startup, GRUB relies on this file to identify installed operating systems, available kernels, and boot parameters.

Since the configuration file was unavailable, GRUB could not locate the operating system and therefore entered the GRUB command-line interface (`grub>`), preventing the normal boot process.

Recovery Procedure

Step 6 - Manually Load Linux Kernel

```
grub> linux (hd0,msdos1)/boot/vmlinuz-5.14.0-611.5.1.el9_7.x86_64 root=UUID=d1aa  
5bd2-a705-4dcf-b83a-307ce9263a6e ro  
grub> _
```

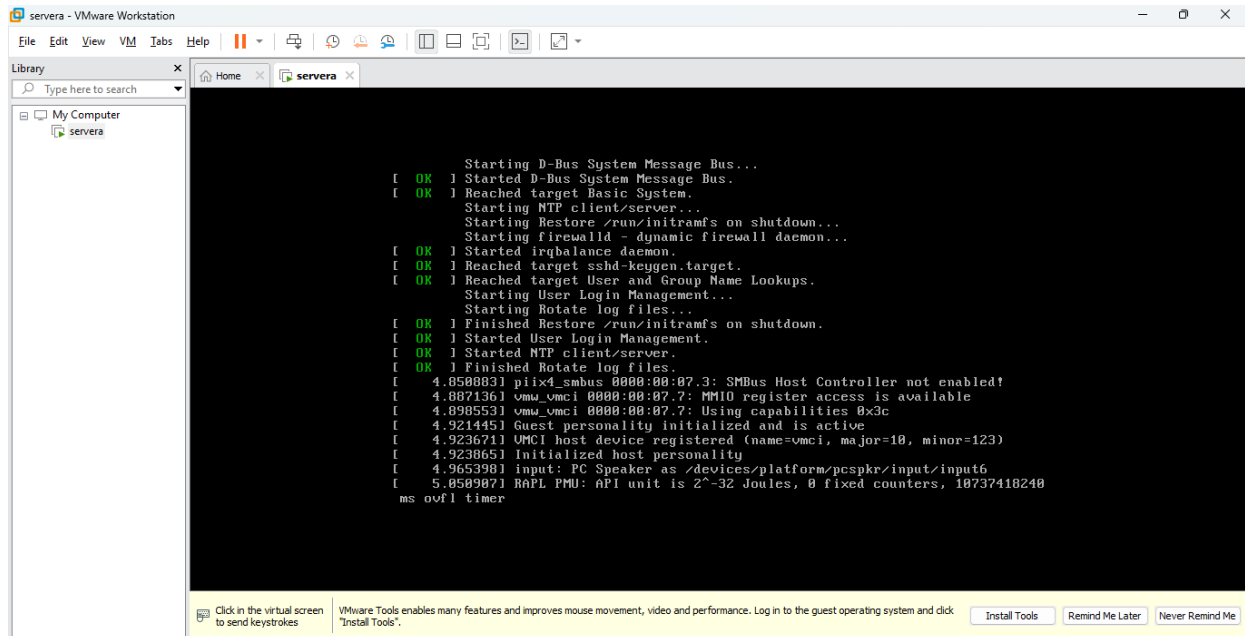
linux (hd0,msdos1)/boot/vmlinuz-5.14.0-611.5.1.el9_7.x86_64 root=UUID=<UUID> ro
→ Loads the Linux kernel manually.

Step 7 - Load Initial RAM Disk

```
grub> initrd (hd0,msdos1)/boot/initramfs-5.14.0-611.5.1.el9_7.x86_64.img  
grub>
```

initrd (hd0,msdos1)/boot/initramfs-5.14.0-611.5.1.el9_7.x86_64.img
→ Loads the initramfs image required for booting.

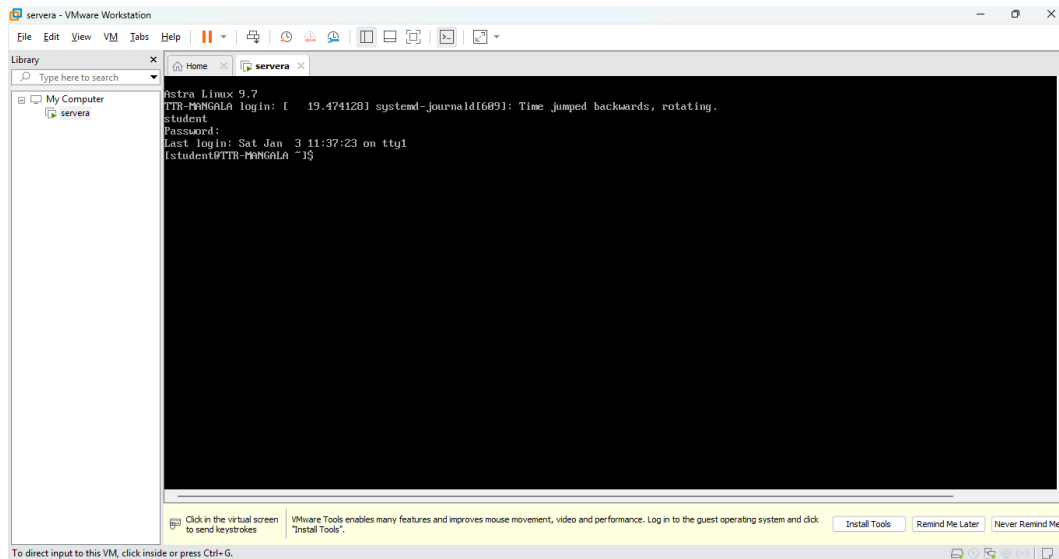
Step 8 - Boot the Operating System



boot

→ Starts the operating system using the manually loaded kernel and initramfs.

Step 9 - Login Verification

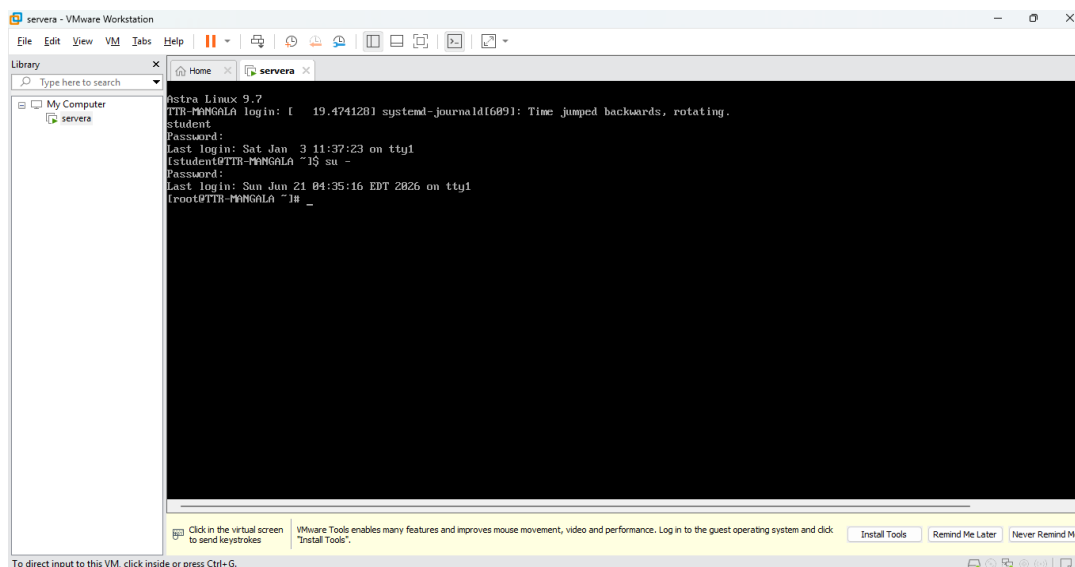


Verification

- Astra Linux started successfully.
- System services loaded correctly.
- Login screen appeared.

This confirmed that the operating system had been recovered successfully.

Step 10 - Root Access Verification



→ Obtained administrative privileges.

Step 11 - Accessing Required File

```
Astra Linux 9.7
TTR-MANGALA login: [ 19.474128] systemd-journald[609]: Time jumped backwards, rotating.
student
Password:
Last login: Sat Jan 3 11:37:23 on tty1
[student@TTR-MANGALA ~]$ su -
Password:
Last login: Sun Jun 21 04:35:16 EDT 2026 on tty1
[root@TTR-MANGALA ~]# ls -l /root/abc.sh
-rw-r--r--. 1 root root 37 Jun 21 04:36 /root/abc.sh
[root@TTR-MANGALA ~]# cat /root/abc.sh
WELCOME ASTRA LINUX

SHYAM SEBASTIAN
[root@TTR-MANGALA ~]# _
```

ls -l /root/abc.sh # cat /root/abc.sh

→ Displays the contents of the required file.

Permanent Solution

Step 12 - Verify Missing GRUB Configuration

```
[root@TTR-MANGALA ~]# ls -l /boot/grub2/grub.cfg
ls: cannot access '/boot/grub2/grub.cfg': No such file or directory
[root@TTR-MANGALA ~]#
```

ls -l /boot/grub2/grub.cfg

→ The GRUB configuration file was missing. This was the reason the system stopped at the **grub>** prompt during boot.

Step 13 - Regenerate GRUB Configuration

```
[root@TTR-MANGALA ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
[ 894.848294] device-mapper: core: CONFIG_IMA_DISABLE_HTABLE is disabled. Duplicate IMA measurements will not be recorded in the IMA log.
[ 894.848458] device-mapper: uevent: version 1.0.3
[ 894.841267] device-mapper: ioctl: 4.50.0-ioctl (2025-04-28) initialised: dm-devel@lists.linux.dev
Generating grub configuration file ...
Adding boot menu entry for UEFI Firmware Settings ...
done
[root@TTR-MANGALA ~]# _
```

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

→ This command recreates the missing GRUB configuration file and restores normal boot functionality

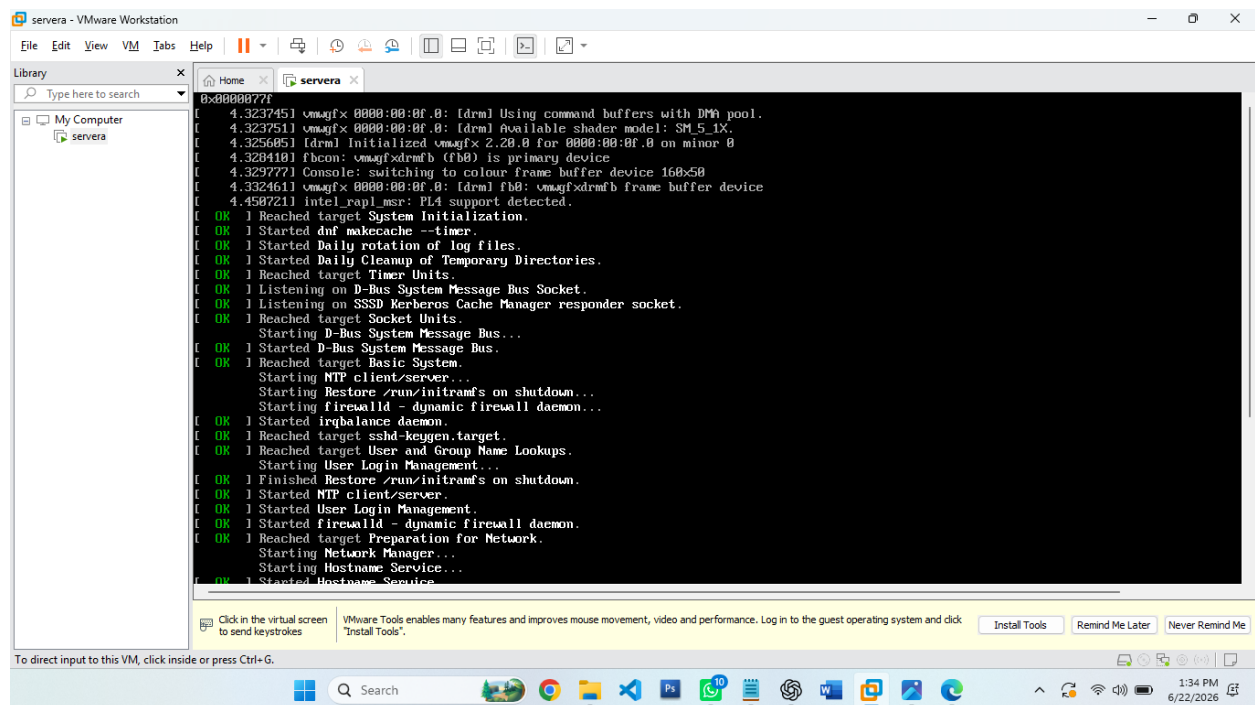
Step 14 - Verify GRUB Configuration Creation

```
[root@TTR-MANGALA ~]# ls -l /boot/grub2/grub.cfg
-rw-----. 1 root root 6880 Jun 22 04:02 /boot/grub2/grub.cfg
[root@TTR-MANGALA ~]#
```

```
# ls -l /boot/grub2/grub.cfg
```

→ The GRUB configuration file was successfully recreated.

Step 15 - Reboot the System



```
# boot
```

→ Restart the system to verify that it boots automatically without manual GRUB commands.

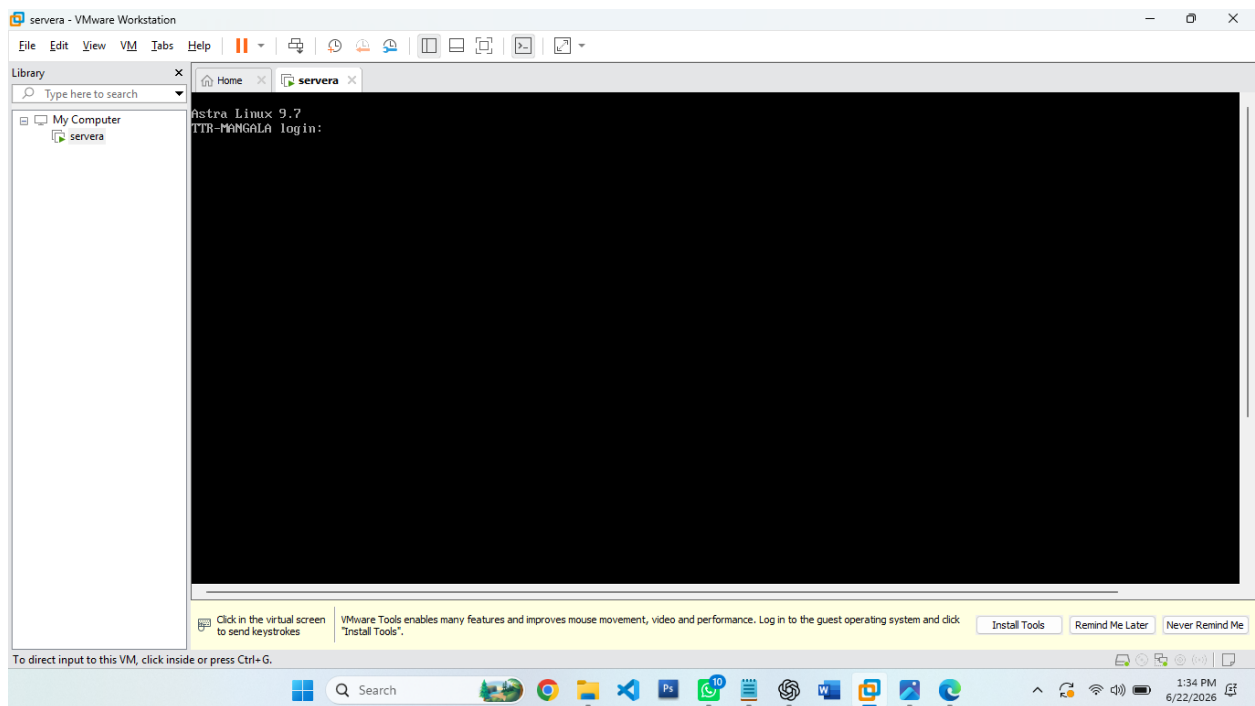
Step 16 - Boot Verification

After rebooting:

- The system loaded normally.
- Kernel boot messages appeared.
- System services started successfully.
- Login prompt appeared automatically.

This confirmed that the permanent fix was successful.

The boot issue occurred because the GRUB configuration file (`/boot/grub2/grub.cfg`) was missing. A new GRUB configuration file was generated using the `grub2-mkconfig` command. After rebooting, the system booted normally, confirming that the issue was fixed successfully.



Technical Skills Demonstrated

- Linux Troubleshooting
- GRUB Recovery
- Filesystem Analysis
- Kernel Boot Management
- System Recovery

- Root Cause Analysis
- Linux Administration
- Bootloader Configuration

Project Outcome

- ✓ Successfully identified the boot failure.
- ✓ Located the missing GRUB configuration file.
- ✓ Manually recovered the operating system.
- ✓ Regenerated the GRUB configuration file.
- ✓ Verified successful system reboot.
- ✓ Accessed the required file (/root/abc.sh).
- ✓ Restored the Astra Linux system to normal operation.

Conclusion

This project successfully demonstrated Linux troubleshooting, bootloader recovery, and system restoration skills. The Astra Linux system failed to boot due to a missing GRUB configuration file. The issue was identified by analyzing the filesystem and bootloader configuration from the GRUB shell. The system was recovered by manually loading the kernel and initramfs, and a permanent solution was implemented by regenerating the GRUB configuration file using the grub2-mkconfig command. After rebooting, the system started normally, user login was verified, and the required file /root/abc.sh was successfully accessed. All project objectives were completed successfully.

Submitted by;

Name : **Nandu Sivadas**
Course : **Cloud & Devops**
Date : **23/06/2026**
Institution : **IPSR solutions Ltd**
Kottayam