Service Status Check Project1 Video Walkthrough

Reference:
This walkthrough is based on the visuals observed in the Project1.mp4 video. It outlines the purpose, composition, and execution of a Bash script used to monitor service uptime.

---

1. Script Initialization

I launched a terminal session and navigated to the project workspace. A Bash script (likely named status-check.sh) was already present in the working directory. This script was designed to verify whether a specific service was running.

---

2. Script Composition

I opened the script in a terminal-based text editor and began with the standard shebang line to define the environment:

```
#!/bin/bash
```

I added a simple echo command to indicate that the check had started:

```
echo "Checking service status..."
```

---

3. Service Status Logic

To check the running status of the nginx web server, I added the following command:

```
systemctl status nginx | grep Active
```

This filtered the output to show only the line indicating whether the service was active or inactive. Alternatively, on older systems, the equivalent would be:

```
service nginx status | grep Active
```

---

4. Script Execution

After saving the script, I made it executable using:

```
chmod +x status-check.sh
```

Then I executed it from the terminal:

```
./status-check.sh
```

The output displayed the active or inactive state of the nginx service, providing immediate diagnostic insight.

Conclusion

This walkthrough demonstrates how I used a Bash script to monitor the operational state of a system service. Its a useful technique for quick diagnostics and can be extended with alerting or log aggregation features in a production environment.