

Schritt-für-Schritt Anleitung

1. Bash-Skript zur Ermittlung der Latenz erstellen

1.1 Skript-Datei erstellen:

- Öffne ein Terminalfenster.
- Erstelle eine neue Datei namens `latency_check.sh` in deinem Home-Verzeichnis:
`touch ~/latency_check.sh`.
- Öffne die Datei mit einem Texteditor deiner Wahl, beispielsweise mit `nano ~/latency_check.sh`.

1.2 Skriptinhalt strukturieren:

Das Skript wird in mehrere Funktionen unterteilt, um die Lesbarkeit und Wartbarkeit zu verbessern:

- Shebang und Variablendeklaration:
- Beginne das Skript mit dem Shebang `#!/bin/bash`.
- Definiere Variablen für den Host und den Dateinamen:

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

```
HOST="google.com" # Standard-Host, kann überschrieben werden  
LATENCY_FILE="/tmp/latency_data.txt" # Dateiname für die Latenzdaten
```

- Funktion zur Messung der Latenz:
- Erstelle eine Funktion, die den Ping durchführt und die durchschnittliche Latenz misst. ```bash
measure_latency() { local latency=$(ping -c 5 $HOST | tail -1 | awk '{print $4}' | cut -d '/' -f 2) echo
$latency }`

```
- Diese Funktion nutzt den `ping`-Befehl, sendet fünf ICMP-Echoanfragen (`-c 5`) an
```

```
- **Funktion zum Schreiben der Latenz in eine Datei:**
```

```
- Erstelle eine Funktion, um die gemessene Latenz in die definierte Datei zu schreiben
```

```
``bash  
write_latency_to_file() {  
    local latency=$(measure_latency)  
    echo $latency > $LATENCY_FILE  
}
```

- Diese Funktion ruft `measure_latency` auf und leitet das Ergebnis in die Datei `latency_data.txt` um.
- Hauptfunktion (main):

- ◆ Definiere eine `main`-Funktion, die die anderen Funktionen aufruft.

```
main() {  
    write_latency_to_file  
}
```

```
main
```

- Diese Struktur erleichtert die Verwaltung des Skriptflusses und ermöglicht eine zukünftige Erweiterung.

1.3 Skript speichern und ausführbar machen:

- Speichere die Änderungen im Skript.

- Mache das Skript ausführbar mit dem Befehl: `chmod +x ~/latency_check.sh`.

2. SystemD Unit als Benutzerdienst erstellen

2.1 Unit-Datei erstellen:

- Öffne ein Terminalfenster.
- Erstelle den folgenden Pfad, falls er noch nicht existiert `mkdir -p ~/.config/systemd/user`
- Erstelle eine neue Unit-Datei für den Benutzerdienst im entsprechenden SystemD-Verzeichnis. Benutze den Befehl: `touch ~/.config/systemd/user/latency_check.service`.
- Öffne die Datei mit einem Texteditor, zum Beispiel:
`nano ~/.config/systemd/user/latency_check.service`.

2.2 Unit-Datei konfigurieren:

- Schreibe die folgende Konfiguration in die Unit-Datei:

```
[Unit]
Description=Latenz Check Service

[Service]
ExecStartPre=/bin/bash -c 'echo "Starting latency check at $(date)" >> /tmp/latency_
ExecStart=/home/DEIN_BENUTZERNAME/latency_check.sh
Restart=always
RestartSec=20

[Install]
WantedBy=default.target
```

- Ersetze `DEIN_BENUTZERNAME` mit deinem tatsächlichen Benutzernamen.
- `ExecStartPre` fügt einen Eintrag in eine Logdatei in `/tmp` hinzu, bevor das Skript gestartet wird.
- `ExecStart` definiert den Pfad zu deinem Latenzmessung-Skript.
- `Restart=always` und `RestartSec=20` sorgen dafür, dass das Skript alle 20 Sekunden neu gestartet wird.

2.3 SystemD Unit aktivieren und starten:

- Aktiviere den Benutzerdienst mit: `systemctl --user enable latency_check.service`.
- Starte den Dienst mit: `systemctl --user start latency_check.service`.

3. Überprüfung

3.1 Überprüfen, ob der Dienst läuft:

- Überprüfe den Status des Dienstes mit: `systemctl --user status latency_check.service`.
- Dies zeigt dir den aktuellen Status des Dienstes und die letzten Log-Einträge an.

3.2 Log-Datei überprüfen:

- Sieh dir die Log-Datei in `/tmp` an, um sicherzustellen, dass der `ExecStartPre`-Befehl funktioniert:
`cat /tmp/latency_check.log`.
- Du solltest Einträge sehen, die anzeigen, dass der Dienst gestartet wurde.

4. BashRc anpassen, um die Latenzinformation im Prompt anzuzeigen

4.1 .bashrc-Datei bearbeiten:

1. Bash-Skript zur Ermittlung der Latenz erstellen

- Öffne die .bashrc-Datei in deinem Home-Verzeichnis. Du kannst dies tun, indem du nano ~/.bashrc in deinem Terminal eingibst.

Um die Latenzinformation dynamisch im Prompt anzuzeigen, unter Verwendung der PROMPT_COMMAND-Funktion, kannst du die .bashrc-Datei wie folgt anpassen:

4.2 Funktion zur dynamischen Anzeige der Latenz hinzufügen:

- Füge am Ende der .bashrc-Datei eine Funktion hinzu, die die Latenzinformation aus der Datei liest und in einer Variablen speichert.
- Beispiel:

```
update_latency() {
    if [ -f /tmp/latency_data.txt ]; then
        LATENCY=$(cat /tmp/latency_data.txt)
    else
        LATENCY="No Latency Data"
    fi
}
```

- Diese Funktion liest den Inhalt der Datei latency_data.txt und speichert ihn in der Variable LATENCY. Falls die Datei nicht existiert oder leer ist, wird eine entsprechende Meldung in LATENCY gespeichert.

4.3 Anpassen des Prompts mit PROMPT_COMMAND:**

- Anstelle der direkten Modifikation der PS1-Variablen, nutze die PROMPT_COMMAND-Variable, um update_latency aufzurufen, bevor der Prompt angezeigt wird. Dies stellt sicher, dass die Latenzdaten bei jedem neuen Prompt aktualisiert werden.
- Beispiel:

```
PROMPT_COMMAND="update_latency"
PS1="\$LATENCY ms $PS1"
```

- PROMPT_COMMAND führt update_latency vor jeder neuen Eingabeaufforderung aus, und PS1 beinhaltet nun die aktuelle Latenzinformation aus der LATENCY-Variablen.

4.4 Änderungen übernehmen:

- Speichere die Änderungen in der .bashrc-Datei und schließe den Editor.
- Um die Änderungen sofort zu übernehmen, führe source ~/.bashrc in deinem Terminal aus.

Durch diese Anpassung wird die Latenzinformation direkt am Anfang des existierenden Terminal-Prompts angezeigt. Wenn du nun ein neues Terminalfenster öffnest oder eine neue Shell-Sitzung startest, solltest du die Latenzinformation am Anfang der Prompt-Zeile sehen.

5. Testen und Debuggen der gesamten Konfiguration

Nachdem alle Komponenten eingerichtet sind, ist es wichtig, das Gesamtsystem zu testen und bei Bedarf zu debuggen.

5.1 Überprüfen der Skriptausführung:

- Führe das latency_check.sh Skript manuell aus, um sicherzustellen, dass es korrekt funktioniert: ~/latency_check.sh.

- Überprüfe, ob die `latency_data.txt` Datei im Home-Verzeichnis erstellt und aktualisiert wird. Nutze `cat ~/latency_data.txt`, um den Inhalt anzusehen.

5.2 Überprüfen der SystemD Unit:

- Überprüfe, ob der SystemD-Benutzerdienst aktiv ist und ohne Fehler läuft:
`systemctl --user status latency_check.service`.
- Sieh dir die Log-Datei in `/tmp/latency_check.log` an, um zu bestätigen, dass das Skript regelmäßig gestartet wird.

5.3 Überprüfen der Prompt-Anzeige:

- Öffne ein neues Terminalfenster oder eine neue Shell-Sitzung, um zu überprüfen, ob die Latenzinformation im Prompt korrekt angezeigt wird.
- Wenn keine Latenzinformation angezeigt wird, überprüfe die Funktion `get_latency` in der `.bashrc` auf Fehler.

5.4 Debugging:

- Wenn das Skript nicht wie erwartet funktioniert, überprüfe den Skriptcode auf Tippfehler oder Logikfehler.
- Für Probleme mit der SystemD Unit, nutze `journalctl --user -u latency_check.service` um detaillierte Logs zu sehen.
- Stelle sicher, dass die Pfade und Benutzernamen in den Skripten und Unit-Dateien korrekt sind.
- Bei Problemen mit der Prompt-Anzeige, überprüfe, ob die `.bashrc`-Änderungen korrekt sind und ob die `source`-Befehle korrekt ausgeführt wurden.

5.5 Fehlerbehebung bei der Netzwerkverbindung:

- Wenn das Skript keine Latenz misst, stelle sicher, dass dein Computer eine aktive Netzwerkverbindung hat und der angegebene Host erreichbar ist.