**Projekt: Webserver mit Automatisierung, Backup und Rechteverwaltung**

**Aufgabe: Webserver-Installation und Bereitstellung einer Webseite**

Herzlichen Glückwunsch zur erfolgreichen Weiterbildung zum DevOps Engineer! Ihre erste Aufgabe bei der Firma IT-Boss-Solutions besteht darin, eine einfache Webserver-Lösung zu erstellen. Diese Aufgabe hilft Ihnen, Ihre Fähigkeiten in der Webserver-Konfiguration, Automatisierung und Rechteverwaltung zu vertiefen. Die Aufgabe ist in mehrere Schritte unterteilt, die aufeinander aufbauen.

**Ziel:**

Bereitstellung einer Webseite der Firma IT-Boss-Solutions auf einem Apache-Webserver. Zudem lernen Sie, wie man den Webserver überwacht und Fehlerbehandlungen mit if-else oder switch-case Strukturen durchführt.

**Teil 1: Webserver-Installation und Bereitstellung**

**Ziel:** Installieren Sie einen Apache-Webserver und stellen Sie eine einfache Webseite bereit.

**Aufgabenstellung:**

1. **Installation des Apache-Webservers**
   * Verwenden Sie das Paketmanagement-Tool (apt für Ubuntu/Debian oder yum für CentOS/RedHat), um den Apache-Webserver zu installieren.
2. **Starten und Aktivieren des Webservers**
   * Starten Sie den Apache-Webserver und konfigurieren Sie ihn so, dass er beim Booten des Servers automatisch startet.
3. **Bereitstellung der Webseite**
   * Kopieren Sie die bereitgestellten HTML- und CSS-Dateien in das Verzeichnis /var/www/html/. Diese Dateien sollten eine einfache Webseite der Firma "IT-Boss-Solutions" darstellen, die den Titel "IT-Boss-Solutions" anzeigt.
4. **Überprüfung des Webservers**
   * Stellen Sie sicher, dass der Webserver korrekt läuft, indem Sie die IP-Adresse des Servers in Ihrem Browser aufrufen (z. B. http://localhost)
   * Wenn der Webserver nicht läuft, soll eine Fehlerbehandlung erfolgen.

**Fehlerbehandlung mit if-else:**

Verwenden Sie eine if-else-Bedingung, um zu überprüfen, ob der Apache-Webserver erfolgreich gestartet wurde. Falls der Webserver nicht läuft, geben Sie eine Fehlermeldung aus und stoppen Sie das Skript.

**Teil 2: Automatisierung der Sicherung**

**Ziel:** Erstellen Sie ein einfaches Backup-System, das regelmäßig Sicherungen der Webseitendaten durchführt.

**Aufgabenstellung:**

1. **Erstellen eines Backup-Skripts**
   * Schreiben Sie ein Bash-Skript, das automatisch eine Sicherung des Verzeichnisses /var/www/html/ erstellt und diese an einem sicheren Ort speichert (z. B. /backups/).
2. **Automatisierung der Backups**
   * Richten Sie einen Cron-Job ein, der das Backup täglich zur gleichen Zeit (z. B. um 2 Uhr nachts) ausführt.

**Teil 3: Benutzer- und Gruppenverwaltung**

**Ziel:** Verwalten Sie Benutzer und Gruppen, um den Zugriff auf Server-Ressourcen zu kontrollieren.

**Aufgabenstellung:**

1. **Erstellen von Benutzern und Gruppen**
   * Erstellen Sie zwei neue Benutzer: webadmin (für die Verwaltung des Webservers) und webuser (für den Zugriff auf die Webseite).
   * Erstellen Sie eine Gruppe namens webgroup, der beide Benutzer angehören sollen.
2. **Ändern der Berechtigungen**
   * Ändern Sie die Berechtigungen für das Verzeichnis /var/www/html/, sodass nur webadmin Schreibzugriff hat und webuser nur Lesezugriff erhält.
3. **Überprüfen der Berechtigungen**
   * Überprüfen Sie mit dem ls -l-Befehl, ob die Berechtigungen korrekt gesetzt wurden.

**Teil 4: Schleifen und reguläre Ausdrücke (Regex)**

**Ziel:** Verwenden Sie Schleifen und reguläre Ausdrücke (Regex), um mehrere Dateien im Webserver-Verzeichnis zu überprüfen und zu benennen.

**Aufgabenstellung:**

1. **Überprüfen der Dateien im Webserver-Verzeichnis**
   * Schreiben Sie ein Bash-Skript, das alle Dateien im Verzeichnis /var/www/html/ überprüft und deren Dateinamen mit einem bestimmten Muster (z. B. alle HTML-Dateien) anzeigt.
   * Verwenden Sie dabei reguläre Ausdrücke, um nur die Dateien zu filtern, die mit .html enden.
2. **Verwenden von Schleifen**
   * Verwenden Sie eine Schleife (for oder while), um alle HTML-Dateien im Verzeichnis /var/www/html/ zu durchlaufen und deren Namen auszugeben.

**Teil 5: Log-Dateien erstellen und Fehlerprotokollierung**

**Ziel:** Erstellen Sie ein Skript, das Log-Dateien schreibt und Fehlerprotokolle mit Schleifen verwaltet.

**Aufgabenstellung:**

1. **Erstellen eines Log-Datei-Skripts**
   * Schreiben Sie ein Bash-Skript, das Log-Dateien erstellt und alle relevanten Aktivitäten des Webservers aufzeichnet. Das Skript sollte eine Datei webserver.log im Verzeichnis /var/log/ anlegen und dort Informationen über die Webserver-Aktivitäten protokollieren, wie z. B. Start und Stopp des Servers, Fehler und erfolgreiche Aktionen.
2. **Fehlerprotokollierung**
   * Falls ein Fehler bei einer der Aktionen auftritt (z. B. beim Starten des Webservers oder beim Kopieren von Dateien), soll eine Fehlermeldung in die Log-Datei geschrieben werden.
3. **Verwendung von Schleifen zur Protokollierung**
   * Erstellen Sie eine Schleife, die die letzten 5 Log-Einträge aus der Log-Datei liest und auf der Konsole anzeigt.
4. **Fehlerbehandlung in der Log-Datei**
   * Verwenden Sie eine if-else-Struktur, um zu überprüfen, ob eine Aktion erfolgreich war, und protokollieren Sie das Ergebnis. Falls ein Fehler auftritt, schreiben Sie eine entsprechende Fehlermeldung in die Log-Datei.

**Abschluss:**

Nachdem Sie alle Aufgaben erfolgreich erledigt haben, testen Sie die Funktionen Ihres Webservers und stellen Sie sicher, dass die Webseite wie gewünscht angezeigt wird. Überprüfen Sie auch, ob die Log-Dateien korrekt geschrieben und Fehler ordnungsgemäß protokolliert wurden. Dokumentieren Sie Ihre Ergebnisse und reflektieren Sie darüber, wie Sie die Aufgabe noch weiter optimieren können.

Viel Erfolg bei der Aufgabe!

Abgabe ist 30.04.2025

**Bereitgestellte Dateien:**

Die Webseite von IT-Boss-Solutions besteht aus den folgenden Dateien:

index.html (HTML-Datei der Webseite)

style.css (CSS-Datei für das Design)