

# Automatisches Importieren und Exportieren der Alias-Konfiguration

## Inhalt

---

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Exportieren und Importieren der Aliastabelle](#)

[Aliastabelle mit Bash-Skript exportieren](#)

[Erläuterung](#)

[1.- Konfiguration von SSH-Schlüsseln und Pfaden](#)

[2.- Stellen Sie eine Verbindung zum Proxy her, und richten Sie den SSH-Tunnel ein.](#)

[3.- Halten Sie 5 Sekunden lang an, bevor Sie fortfahren](#)

[4.- Exportieren der aliasconfig-Datei vom Remote-System](#)

[5.- Laden Sie die Datei in das lokale Verzeichnis](#)

[Skript abschließen](#)

[SSH-Schlüssel](#)

[Überprüfen der exportierten Datei](#)

[Aliastabelle mit Bash-Skript importieren](#)

[Erläuterung](#)

[1.- SSH-Pfad- und Schlüsselkonfiguration](#)

[2.-Aktuelles Datum und aktuelle Uhrzeit abrufen](#)

[3.- Laden Sie die neue aliasconfig-Datei auf den Remote-ESA-Server hoch](#)

[4.- Importieren Sie die neue aliasconfig-Datei, und bestätigen Sie sie mit einem Kommentar.](#)

[5.-Drucken Sie die aktuelle aliasconfig, und speichern Sie sie in einer neuen lokalen Datei](#)

[Überprüfen von Änderungen](#)

[Überprüfen neuer aliasconfig-Tabelleneinträge](#)

[Änderungen bestätigen](#)

[Skriptflexibilität](#)

[Abschließende Überlegungen](#)

[Referenzlinks](#)

---

## Einleitung

In diesem Dokument werden die Schritte zur Automatisierung der Aufgaben zum Importieren und Exportieren der Alias-Konfiguration in der E-Mail-Security-Appliance beschrieben.

# Voraussetzungen

## Anforderungen

Cisco empfiehlt, sich mit folgenden Themen vertraut zu machen:

- Cisco Secure Email Gateway (SEG/ESA) AsyncOS 16.0.2
- Zugriff auf die Befehlszeilenschnittstelle der Cloud-Appliance
- Linux-Kommandozeile
- Shell-Scripting

## Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf folgender Software:

- Cloud E-Mail Security Appliance (CESA)
- Bash

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

## Hintergrundinformationen

Das Ziel besteht darin, bestimmte Aufgaben zu automatisieren, aber einige Prozesse erfordern in der Regel manuelle Eingriffe. Beim Importieren und Exportieren der aktuellen Alias-Konfiguration können diese Aufgaben jedoch vollständig automatisiert werden, sodass keine manuelle Eingabe erforderlich ist.

## Exportieren und Importieren der Aliastabelle

Um eine Aliastabelle zu importieren, überprüfen Sie zuerst den SSH- und SCP-Zugriff, um sicherzustellen, dass Sie eine Verbindung zum E-Mail-Gateway herstellen können.

Bevor Sie fortfahren, muss eine Aliastabelle in der Appliance vorhanden sein:

```
(Machine esa1.xyz.iphmx.com) (SERVICE)> clustermode cluster; aliasconfig print
```

```
test: test@example.com, test@example2.com, test@example3.com  
test2: test@domain.com, test@domain2.com, test@domain3.com  
(Cluster Hosted_Cluster) (SERVICE)>
```

Wenn Sie den Export-Unterbefehl des Befehls aliasconfig verwenden, um eine vorhandene

Aliastabelle zu sichern, wird eine Datei (mit einem von Ihnen angegebenen Namen) generiert und im Verzeichnis /configuration für den Listener gespeichert.

## Aliastabelle mit Bash-Skript exportieren

In diesem Fall stellt ein Bash-Skript eine Verbindung mit der CES-Appliance her und exportiert die Aliasdatei.

Das Bash-Skript ist wie folgt aufgebaut:

```
#!/bin/bash

# Configuration of SSH keys and paths
PROXY_KEY="/full/path/folder/.ssh/id_rsa"
SECOND_KEY="/full/path/folder/.ssh/id_rsa"
LOCAL_PORT="2200"
PROXY_USER="dh-user"
PROXY_HOST="f4-ssh.iphmx.com"
TARGET_HOST="esa1.xyz.iphmx.com"
REMOTE_USER="local_server_user"
REMOTE_FILE="/configuration/filename.csv"
LOCAL_DIR="/full/path/folder/Downloads"
LOCAL_FILE_PATH="${LOCAL_DIR}/aliasconfig-file.csv"

# 1. Connect to the proxy and set up the SSH tunnel
echo "Establishing connection to the proxy..."
ssh -i "$PROXY_KEY" -l "$PROXY_USER" -N -f "$PROXY_HOST" -L "$LOCAL_PORT:${TARGET_HOST}:22"
if [ $? -ne 0 ]; then
    echo "Error: Failed to establish connection to the proxy."
    exit 1
fi
echo "Proxy connection established."

# Pause for 5 seconds before proceeding
sleep 5

# 2. Export the aliasconfig file from the remote system
echo "Exporting aliasconfig file from the remote system..."
ssh -i "$SECOND_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" 'clustermode cluster; aliasconfig export'
if [ $? -ne 0 ]; then
    echo "Error: Failed to export the aliasconfig file."
    exit 1
fi
echo "Aliasconfig file successfully exported."

# Pause for 5 seconds before proceeding
sleep 5

# 3. Download the file to the local directory
echo "Downloading file to the local directory..."
scp -i "$SECOND_KEY" -P "$LOCAL_PORT" -O "$REMOTE_USER"@127.0.0.1:"$REMOTE_FILE" "$LOCAL_DIR" 2>/dev/nul
if [ $? -ne 0 ]; then
    echo "Error: Failed to download the file to the local directory."
    exit 1
fi
echo "File successfully downloaded to: $LOCAL_FILE_PATH"
```

```
# Pause for 5 seconds before finalizing
sleep 5

# Finalizing the script
echo "Process completed successfully."
exit 0
```

## Erläuterung

### 1.- Konfiguration von SSH-Schlüsseln und Pfaden

- PROXY\_KEY und SECOND\_KEY: Der vollständige Pfad zur privaten SSH-Schlüsseldatei, die für die Authentifizierung verwendet wird. Beide Tasten sind in diesem Fall auf den gleichen Pfad eingestellt.
- Beispiel: /full/path/folder/.ssh/id\_rsa
- LOKALER\_PORT: Gibt den lokalen Port (2200) für den SSH-Tunnel an.
- PROXY\_USER: Der Benutzername, der für die Verbindung mit dem Proxyserver verwendet wird.
- PROXY\_HOST: Der Hostname des Proxyservers.
- ZIEL\_HOST: Der vollqualifizierte Domänenname (Fully Qualified Domain Name, FQDN) des Zielhosts, aktualisiert auf esa1.xyz.iphmx.com.
- REMOTE-BENUTZER: Der Benutzername, der für die Verbindung mit der Remote-Appliance über den SSH-Tunnel verwendet wird.
- REMOTE-DATEI: Der Pfad auf dem Remote-System, auf dem die exportierte Datei gespeichert ist (/configuration/filename.csv).
- LOKALER\_DIR: Das lokale Verzeichnis zum Speichern der Datei ist auf /full/path/folder/Downloads festgelegt.
- LOKALER DATEIPFAD: Der vollständige lokale Pfad für die heruntergeladene Datei, aktualisiert auf \${LOCAL\_DIR}/aliasconfig-file.csv.

### 2.- Stellen Sie eine Verbindung zum Proxy her, und richten Sie den SSH-Tunnel ein.

```
echo "Establishing connection to the proxy..."
ssh -i "$PROXY_KEY" -l "$PROXY_USER" -N -f "$PROXY_HOST" -L "$LOCAL_PORT:${TARGET_HOST}:22"
```

- Zweck:
  - Richtet einen SSH-Tunnel zum Proxyserver für die sichere Kommunikation mit dem Zielhost ein.
- Aktualisierungen:
  - Der private SSH-Schlüssel befindet sich jetzt unter /full/path/folder/.ssh/id\_rsa.
  - Der Ziel-Hostname wurde auf esa1.xyz.iphmx.com aktualisiert.

- Fehlerbehandlung:
  - Wenn die Verbindung fehlschlägt, wird eine Fehlermeldung angezeigt, und das Skript wird mit einem Fehlercode beendet (exit 1).

### 3.- Halten Sie 5 Sekunden lang an, bevor Sie fortfahren

- Zweck:
  - Es wird eine Verzögerung eingeführt, um sicherzustellen, dass der SSH-Tunnel vollständig eingerichtet ist, bevor mit dem nächsten Schritt fortgefahren wird.

### 4.- Exportieren der aliasconfig-Datei vom Remote-System

```
echo "Exporting aliasconfig file from the remote system..."
ssh -i "$SECOND_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" 'clustermode cluster; aliasconfig export'
```

- Zweck:
  - Stellt über den SSH-Tunnel eine Verbindung zum Ziel-Host her und exportiert die Aliaskonfiguration in eine Datei mit dem Namen aliasconfig-file.csv.
- Aktualisierungen:
  - Der Dateiname für die exportierte Datei wurde auf aliasconfig-file.csv aktualisiert.
- Ausgabenumleitung:
  - 2>/dev/null unterdrückt alle Fehlermeldungen des SSH-Befehls.
- Fehlerbehandlung:
  - Wenn der Export fehlschlägt, wird eine Fehlermeldung angezeigt, und das Skript wird beendet.

### 5.- Laden Sie die Datei in das lokale Verzeichnis

```
echo "Downloading file to the local directory..."
scp -i "$SECOND_KEY" -P "$LOCAL_PORT" -O "$REMOTE_USER"@127.0.0.1:"$REMOTE_FILE" "$LOCAL_DIR" 2>/dev/null
```

- Zweck:
  - Verwendet scp, um die exportierte Datei sicher vom Remote-System in das lokale Verzeichnis zu kopieren.
- Aktualisierungen:
  - Die lokale Datei wird als aliasconfig-file.csv im Verzeichnis /full/path/folder/Downloads gespeichert.
- Fehlerbehandlung:
  - Wenn das Herunterladen der Datei fehlschlägt, wird eine Fehlermeldung angezeigt, und das Skript wird beendet.

## Skript abschließen

```
echo "Process completed successfully."  
exit 0
```

- Zweck:
  - Gibt eine Erfolgsmeldung aus und beendet das Skript mit einem Erfolgscode (Beenden 0), der angibt, dass alle Vorgänge erfolgreich abgeschlossen wurden.

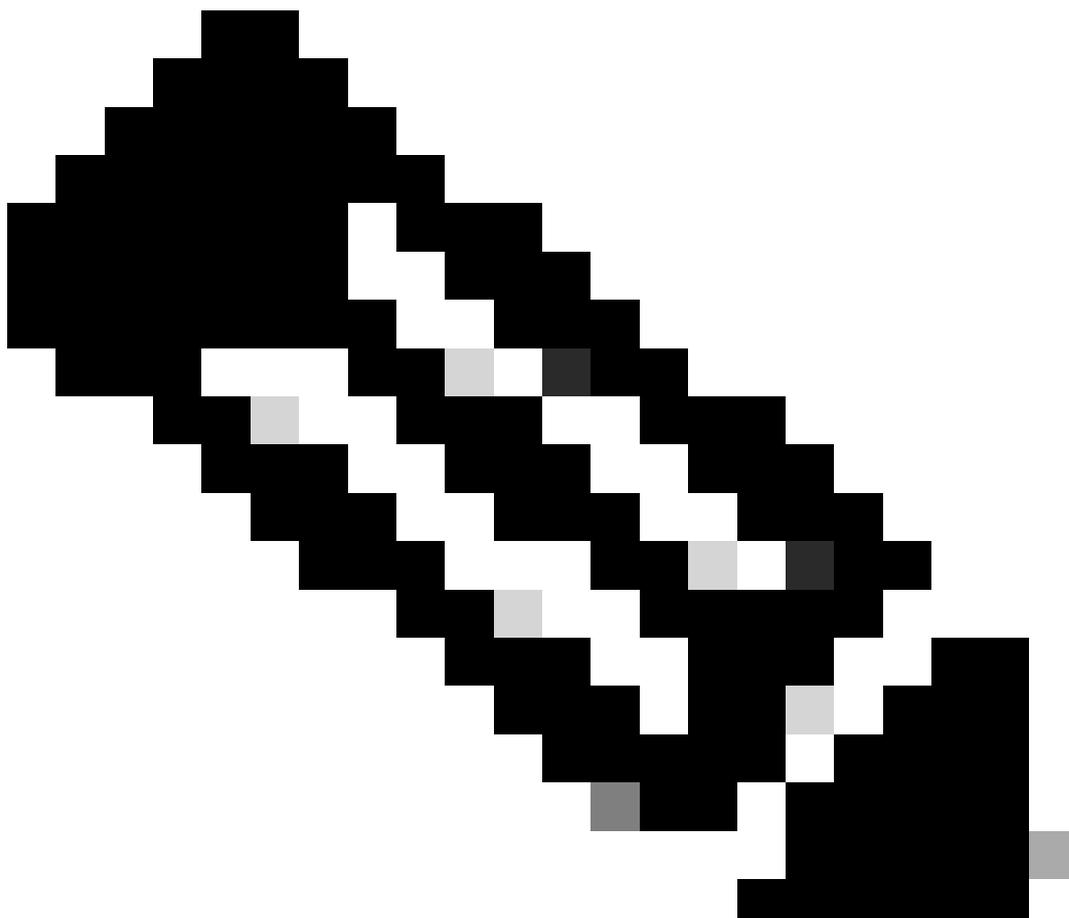
## SSH-Schlüssel

Sie können im Skript zwei Variablen PROXY\_KEY und HOST\_KEY feststellen. Diese Schlüssel können gleich oder verschieden sein.

Der PROXY\_KEY wird für die Verbindung mit der Proxy-Cloud verwendet, die erforderlich ist, um zu Ihren CESA-Servern zu springen.

HOST\_KEY ist der Schlüssel für die Anmeldung als lokaler Benutzer, sodass kein Passwort mehr benötigt wird.

---



Anmerkung: Weitere Informationen zum Konfigurieren des SSH-Zugriffs auf den CES-Proxy und zum Einrichten eines SSH-Schlüssels für einen lokalen Benutzer auf der Appliance finden Sie im Konfigurationshandbuch.

---

## Überprüfen der exportierten Datei

Nachdem das Exportskript ausgeführt wurde, können Sie den Inhalt überprüfen und feststellen, dass es dieselben Informationen enthält wie das Original, das im CLI-Befehl `aliasconfig` der Appliance dargestellt wird.

```
$ pwd
/full/path/folder/Downloads
$ ls
filename.csv
$ cat filename.csv
# File exported by the CLI at 20250702T125347
test: test@example.com, test@example2.com, test@example3.com
```

test2: test@domain.com, test@domain2.com, test@domain3.com ↗

## Aliastabelle mit Bash-Skript importieren

Nachdem Sie die aktuelle Aliasdatei exportiert haben, können Sie sie ändern, die erforderlichen Einträge hinzufügen und dann in die ESA-Aliaskonfiguration importieren.

Das Bash-Skript für den Import ist wie folgt aufgebaut:

```
#!/bin/bash

# Configuration of SSH keys and paths
SSH_KEY="/full/path/folder/.ssh/id_rsa"
LOCAL_PORT="2200"
REMOTE_USER="local_server_user"
LOCAL_FILE="/full/path/folder/Downloads/new-filename.csv"
OUTPUT_DIR="/full/path/folder/Downloads"

# Get the current local date in the desired format
CURRENT_DATE=$(date +"%Y-%m-%d_%H-%M-%S")

# 1. Upload the new aliasconfig file
echo "Uploading new aliasconfig file to the remote system..."
scp -i "$SSH_KEY" -P "$LOCAL_PORT" -O "$LOCAL_FILE" "$REMOTE_USER"@127.0.0.1:/configuration 2>/dev/null
if [ $? -ne 0 ]; then
    echo "Error: Failed to upload the aliasconfig file."
    exit 1
fi
echo "Aliasconfig file successfully uploaded."

# Pause for 5 seconds before proceeding
sleep 5

# 2. Import the new aliasconfig file and commit with a comment
COMMIT_COMMENT="Importing new entries to aliasconfig - $CURRENT_DATE"
echo "Importing the new aliasconfig file and committing changes..."
ssh -i "$SSH_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" "clustermode cluster; aliasconfig import new"
if [ $? -ne 0 ]; then
    echo "Error: Failed to import the aliasconfig file or commit changes."
    exit 1
fi
echo "Aliasconfig file successfully imported and committed with comment: '$COMMIT_COMMENT'."

# Pause for 5 seconds before proceeding
sleep 5

# 3. Print the current aliasconfig and save it to a new file
OUTPUT_FILE="${OUTPUT_DIR}/current-aliasconfig-${CURRENT_DATE}.txt"
echo "Printing current aliasconfig and saving it to: $OUTPUT_FILE..."
ssh -i "$SSH_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" 'clustermode cluster; aliasconfig print' >
if [ $? -ne 0 ]; then
    echo "Error: Failed to print the current aliasconfig."
    exit 1
fi
echo "Current aliasconfig successfully saved to: $OUTPUT_FILE"
```

```
# Finalizing the script
echo "Process completed successfully."
exit 0
```

## Erläuterung

### 1.- SSH-Pfad- und Schlüsselkonfiguration

- SSH\_KEY: Pfad zur privaten SSH-Schlüsseldatei, die für die sichere Authentifizierung gegenüber dem Remote-Server verwendet wird.
- LOKALER\_PORT: Lokaler Port, der für den SSH-Tunnel festgelegt ist.
- REMOTE-BENUTZER: Benutzerkonto für die Authentifizierung auf dem Remote-Server.
- LOKALE DATEI: Lokaler Pfad zur zu importierenden aliasconfig-CSV-Datei.
- OUTPUT\_DIR: Lokaler Ordner, in dem eine Kopie der aktuellen Konfiguration nach dem Importvorgang gespeichert wird.

### 2.- Aktuelles Datum und aktuelle Uhrzeit

```
CURRENT_DATE=$(date +"%Y-%m-%d_%H-%M-%S")
```

- Zweck:
  - Speichern Sie das aktuelle Datum und die aktuelle Uhrzeit in einem bestimmten Format für die Verwendung in Dateinamen und Kommentaren.
- Aktualisierungen:
  - Ermöglicht die einfache Nachverfolgung und Organisation von Protokollen und Backups nach Zeitstempel.

### 3.- Laden Sie die neue aliasconfig-Datei auf den Remote-ESA-Server hoch

Dies ist der neue Inhalt der aliasconfig-Datei:

```
# File exported by the CLI at 20250709T112719
test: new-data@example.com, new-date@example2.com, new-date@example3.com
test2: new-date@domain.com, new-data@domain2.com, new-data@domain3.com␣
```

Fahren Sie mit der Anweisung zum Hochladen der Datei fort:

```
echo "Uploading new aliasconfig file to the remote system..."
scp -i "$SSH_KEY" -P "$LOCAL_PORT" -O "$LOCAL_FILE" "$REMOTE_USER"@127.0.0.1:/configuration 2>/dev/nu11
```

```

if [ $? -ne 0 ]; then
    echo "Error: Failed to upload the aliasconfig file."
    exit 1
fi
echo "Aliasconfig file successfully uploaded."

```

- Zweck:
  - Übertragen Sie die CSV-Datei aliasconfig vom lokalen Computer in das Verzeichnis /configuration auf dem Remote-Server mit SCP über SSH.
- Aktualisierungen:
  - Wenn der Upload fehlschlägt, zeigt das Skript einen Fehler an und stoppt, um unvollständige Importe zu verhindern.

4.- Importieren Sie die neue aliasconfig-Datei, und bestätigen Sie sie mit einem Kommentar.

```

COMMIT_COMMENT="Importing new entries to aliasconfig - $CURRENT_DATE"
echo "Importing the new aliasconfig file and committing changes..."
ssh -i "$SSH_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" "clustermode cluster; aliasconfig import new"
if [ $? -ne 0 ]; then
    echo "Error: Failed to import the aliasconfig file or commit changes."
    exit 1
fi
echo "Aliasconfig file successfully imported and committed with comment: '$COMMIT_COMMENT'."

```

- Zweck:
  - Stellen Sie eine Verbindung über SSH her, importieren Sie die hochgeladene CSV-Datei in die Alias-Konfiguration, und bestätigen Sie die Änderungen mit einem Kommentar mit Zeitstempel für die Nachverfolgung.
- Aktualisierungen:
  - Wenn der Import oder Commit fehlschlägt, wird eine Fehlermeldung angezeigt, und das Skript wird beendet, um die Konfiguration konsistent zu halten.

5.- Drucken Sie die aktuelle aliasconfig, und speichern Sie sie in einer neuen lokalen Datei

```

OUTPUT_FILE="${OUTPUT_DIR}/current-aliasconfig-${CURRENT_DATE}.txt"
echo "Printing current aliasconfig and saving it to: $OUTPUT_FILE..."
ssh -i "$SSH_KEY" "$REMOTE_USER"@127.0.0.1 -p "$LOCAL_PORT" 'clustermode cluster; aliasconfig print' > $OUTPUT_FILE
if [ $? -ne 0 ]; then
    echo "Error: Failed to print the current aliasconfig."
    exit 1
fi
echo "Current aliasconfig successfully saved to: $OUTPUT_FILE"

```

- **Zweck:**
  - Stellt eine Verbindung über SSH her, druckt die aktuelle Alias-Konfiguration aus und speichert die Ausgabe in einer lokalen Datei mit Zeitstempel für Backup und Auditing.
- **Aktualisierungen:**
  - Wenn der Vorgang fehlschlägt, zeigt das Skript einen Fehler an und stoppt, um unvollständige Ergebnisse zu vermeiden.

## Überprüfen von Änderungen

Nach der Ausführung des Skripts können Sie die Änderungen in der Alias-Konfigurationstabelle überprüfen und die Commits in den Systemprotokollen überprüfen.

### Überprüfen neuer aliasconfig-Tabelleneinträge

Neue Änderungen wurden auf die Tabelle angewendet.

```
(Machine esa1.xyz.iphmx.com) (SERVICE)> clustermode cluster; aliasconfig print
```

```
test: new-data@example.com, new-data@example2.com, new-data@example3.com  
test2: new-data@domain.com, new-data@domain2.com, new-data@domain3.com
```

### Änderungen bestätigen

Mit diesem Befehl können Sie die an der ESA vorgenommenen Änderungen nachverfolgen und überprüfen, einschließlich des Benutzers, der die Änderung vorgenommen hat, und des Datums, an dem sie vorgenommen wurde.

```
(Machine esa1.xyz-66.iphmx.com) (SERVICE)> grep "commit" system_logs  
Wed Jul 9 11:29:42 2025 Info: PID 95790: User local_server_user commit changes: Importing new entries
```

## Skriptflexibilität

Während dieses Skript derzeit in Bash geschrieben ist, kann es problemlos in anderen Skript- oder Programmiersprachen wie Python, PowerShell oder Perl angepasst oder umgeschrieben werden, um besser auf die Präferenzen oder Anforderungen der verschiedenen Administratoren und Umgebungen abgestimmt zu sein. Diese Flexibilität gewährleistet, dass die grundlegende Logik und der Workflow beibehalten und gleichzeitig die Sprache oder Tools genutzt werden können, die am besten zu Ihren betrieblichen Anforderungen passen.

## Abschließende Überlegungen

Dieses Import-/Exportskript bietet eine praktische und effiziente Lösung für die Verwaltung von Alias-Konfigurationen direkt auf der Appliance. Durch die Automatisierung von Upload, Import und Backup von Konfigurationsdateien können Administratoren Änderungen sicher und zuverlässig ohne manuellen Eingriff vornehmen. Das Skript optimiert nicht nur den Prozess, sondern gewährleistet auch die Rückverfolgbarkeit durch Backups mit Zeitstempeln und Commit-Kommentare.

Ein solches Skript trägt außerdem dazu bei, Konsistenz und Compliance in Ihrer Umgebung zu gewährleisten, insbesondere wenn mehrere Änderungen oder Massenaktualisierungen erforderlich sind. Regelmäßige Backups der aktuellen Konfiguration bieten eine zusätzliche Sicherheitsebene, sodass bei Bedarf eine schnelle Wiederherstellung oder ein Rollback möglich ist.

Insgesamt versetzt dieser Ansatz die Teams in die Lage, Konfigurations-Updates zuverlässiger, kontrollierbarer und effizienter zu verwalten. Für zukünftige Anpassungen kann das Skript einfach angepasst werden, um andere Arten von Konfigurationsdateien zu verarbeiten oder weitere zusätzliche Wartungsaufgaben nach Bedarf zu automatisieren.

## Referenzlinks

- [Zugriff auf die Kommandozeile Ihrer Cloud Email Security \(CES\)-Lösung](#)
- [CLI-Anweisungen: PuTTY \[Windows-/PC-Benutzer\]](#)
- [So konfigurieren Sie die SSH Public Key-Authentifizierung für die Anmeldung bei der ESA ohne Kennwort](#)

## Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.