

Network Programming im Cisco IOS

Workshop mit Praxisbeispielen

Die programmgestützte Automatisierung spielt in der Netzwerkwelt eine immer größer werdende Rolle. In den letzten Jahren hat sich bei dieser Automatisierung die interaktive, interpretierte und objektorientierte Programmiersprache Python, nicht zuletzt wegen ihrer einfach strukturierten und gut lesbaren Syntax, durchgesetzt. Somit bietet auch Cisco nunmehr die Möglichkeit der Verwendung eines Python-Interpreters auf Cisco-Netzwerkgeräten. In diesem praxisnahen Kurs werden Python Programme geschrieben, welche direkt auf Cisco-Geräten ausgeführt werden, um typische Aufgaben eines Netzwerkadministrators automatisiert zu erledigen.

Kursinhalt

- Verwendung der Guest-Shell
- Portstatistiken mit Python
- Logging
- Konfigurationen analysieren
- Konfigurations-Management
- Auslesen von Loopbackadressen
- Automatisierte Konfiguration von STP und RSTP
- Triggern von Python-Skripten mit Ciscos Event Manager
- Generieren und Analysieren von Log-Dateien
- SYSLOG und SNMP
- Automatisierte Updates

E-Book Das ausführliche deutschsprachige digitale Unterlagenpaket, bestehend aus PDF und E-Book, ist im Kurspreis enthalten.

Zielgruppe

Der Kurs richtet sich an all diejenigen, die Python zur Automatisierung auf Cisco-Geräten einsetzen wollen.

Voraussetzungen

Die Teilnehmer sollten Python-Programmierkenntnisse und Wissen bezüglich typischer Cisco Operating Systems mitbringen. Diese können in den Kursen Python für Einsteiger und CCNA erworben werden.

Kursziel

Nach diesem Kurs können Sie das Potenzial der Python-Programmierung in Cisco-Netzwerken einschätzen und erwerben praktische Erfahrungen mit den wichtigsten programmatischen Hilfsmitteln.

Dieser Kurs im Web



Alle tagesaktuellen Informationen und Möglichkeiten zur Bestellung finden Sie unter dem folgenden Link:
www.experteach.de/go/CPRN

Vormerkung

Sie können auf unserer Website einen Platz kostenlos und unverbindlich für 7 Tage reservieren. Dies geht auch telefonisch unter 06074 4868-0.

Garantierte Kurstermine

Für Ihre Planungssicherheit bieten wir stets eine große Auswahl garantierter Kurstermine an.

Ihr Kurs maßgeschneidert

Diesen Kurs können wir für Ihr Projekt exakt an Ihre Anforderungen anpassen.

Stand 18.10.2025

Training		Preise zzgl. MwSt.
Termine in Deutschland	3 Tage	€ 2.595,-
Online Training	3 Tage	€ 2.595,-
Termin/Kursort	Kurssprache Deutsch 	
11.05.-13.05.26  Frankfurt	11.05.-13.05.26  Online	

Inhaltsverzeichnis

Network Programming im Cisco IOS – Workshop mit Praxisbeispielen

1 Grundkonzepte der Programmiersprache Python	2 Die Guest Shell im IOS-XE und NX-OS	4.1 Datenmodelle und Serialization
1.1 Die Programmiersprache Python	2.1 Aktivieren der Guest Shell	4.1.1 XML
1.1.1 Download, Installation und Lieferumfang	2.1.1 Voraussetzungen	4.1.2 JSON
1.1.2 Python-Shell	2.1.2 Arbeiten in der Guest Shell im NX-OS	4.1.3 YAML
1.2 Kurze Demonstration grundlegender Kontroll- und Datenstrukturen	2.1.3 Arbeiten mit der Guest Shell im IOS-XE	4.2 YANG
1.2.1 Rechnen mit ganzen Zahlen	2.2 Erreichbarkeit der Guest Shell	4.2.1 Struktur eines Moduls
1.2.2 Verwendung von Variablen	2.2.1 Interne Anbindung der Guest Shell im NX-OS	4.2.2 Module und Submodule
1.2.3 Gleitkommazahlen und Zeichenfolgen	2.2.2 Interne Anbindung der Guest Shell im IOS-XE	4.2.3 Konfiguration eines Moduls
1.2.4 Listen und Dictionaries	2.3 Installation von Software in der Guest Shell	4.2.4 Umsetzung in XML
1.2.5 Logische Ausdrücke	2.3.1 YUM	4.3 NETCONF
1.2.6 Funktionen	2.3.2 RPM	4.3.1 Protokoll-Stapel
1.2.7 Funktionsdefinitionen und Bedingungen	2.3.3 PIP	4.3.2 Subtree Filter
1.2.8 Schleifen und Klassen		4.3.3 NETCONF-Operationen
1.3 Kontroll- und Datenstrukturen	3 On-Box Python	4.3.4 Capability Advertisement
1.3.1 Anweisungen und Blöcke in Python	3.1 Python in der Guest Shell	4.4 REST API
1.3.2 Boolesche und Arithmetische Operationen	3.1.1 Python Versionen	4.4.1 RESTful API
1.3.3 Weitere Ausdrücke in Python	3.1.2 Aufruf von Python Code	4.4.2 HTTP Methoden
1.3.4 Variablen in Python	3.2 Python Module und Packages	4.4.3 Generische Beschreibung eines REST API
1.3.5 Funktionen I: Eingebaute Funktionen	3.2.1 clip()	4.4.4 Authentifizierung
1.3.6 Die Hilfe-Funktion	3.2.2 cli()	4.5 XML in Python
1.3.7 Die Bedingung: If	3.2.3 clid()	4.5.1 Hierarchische Anordnung
1.3.8 String Details	3.2.4 Das Package cli im IOS-XE	4.5.2 Nützliche Funktionen von Element()
1.3.9 Weitere String Details	3.2.5 Erweiterungen des cli-Package im IOS-XE	4.5.3 Einlesen von XML-Strukturen
1.3.10 Iterationen I – While	3.2.6 Das Package cisco	4.5.4 XPath Support
1.3.11 While-Schleife	3.3 Anleitung für Python Skripte	4.6 ncclient
1.3.12 For – Eine Vorschau	3.3.1 Benutzer-Eingaben	4.6.1 Erzeugung eines Subtree Filters aus YANG
1.3.13 Eigene Funktionen	3.3.2 Verarbeitung langer Strings	4.6.2 Prozessierung der XML-Daten des NETCONF Servers
1.3.14 Verschachtelung	3.3.3 Auswerten von Tabellen: String-Vorlage	4.6.3 Konfigurieren mit dem ncclient
1.3.15 Global oder Lokal	3.3.4 Auswerten von Tabellen: JSON-Vorlage	4.6.4 Fehlerbehandlung im ncclient
1.3.16 Rekursion	3.3.5 Erzeugung von Konfiguration aus Python	4.6.5 Use Case: MAC Trace
1.3.17 Arten von Argumenten	3.4 Fehler-Behandlung	4.7 requests
1.3.18 Liste	3.4.1 System Exceptions	4.7.1 Programmatisches Gerüst in Python
1.3.19 Weitere Methoden von Listen	3.4.2 Cisco-spezifische Exceptions	4.7.2 Aktivierung des NXAPI
1.3.20 Tupel	3.4.3 Reaktion auf Fehler im IOS-XE	4.7.3 NXAPI CLI
1.3.21 Eine Menge - Set	3.4.4 Reaktion auf Fehler im NX-OS	4.7.4 NXAPI REST
1.3.22 Dictionaries	3.5 Dynamischer Aufruf von Python Skripten	4.7.5 NXAPI Sandbox
1.3.23 Dictionaries – Methoden	3.5.1 EEM im NX-OS	4.7.6 NXAPI REST Browser
1.4 Weitere Konzepte und Funktionen	3.5.2 EEM im IOS-XE	4.7.7 Interaktion mit einem Web Server
1.4.1 Exceptions	3.5.3 Scheduler Job im NX-OS	4.8 netmiko
1.4.2 Anonyme Funktionen	3.6 Use Cases für On-Box Python Scripting	4.8.1 Session-Aufbau
1.4.3 Comprehensions	3.6.1 Description Eraser	4.8.2 EXEC Mode Kommandos
1.5 Objektorientierte Programmierung	3.6.2 Description basierend auf CDP-Nachbarn	4.8.3 Configuration Mode Kommandos
1.5.1 Klassendefinition	3.6.3 BGP Neighbor Reachability	4.8.4 Up- und Download mit SCP
1.5.2 Verhalten hinzufügen	3.6.4 IP Address Finder	4.8.5 Behandlung von Fehlern
1.5.3 Spezialisierung durch Subklassen	3.6.5 Loopback Finder	4.9 napalm
1.6 Hinzufügen von internen und externen Modulen in Python	3.6.6 Automatische Erzeugung einer BGP-Konfiguration	4.9.1 Verbindungsaufbau
1.6.1 Import	3.6.7 Lastschwankungen erkennen	4.9.2 Configuration Management
1.6.2 Regular Expressions	3.6.8 Zero Touch Provisioning	4.9.3 Getter-Funktionen
1.6.3 Externe Module: Pip und PyPI		4.9.4 Validation
	4 Off-Box Python	

