

Cisco DevNET Associate

خلاصه :

سلام و خوش اومدید به دنیای جذاب و دوست داشتنی من. من سینا روزبه یکی از اعضای کوچیک Cisco DevNET هستم که در حال حاضر بر روی مبنای توسعه دهنده شبکه های Cisco کار می کنم و امروز می خوام در مورد این که چطور تبدیل به یه مهندس شبکه در دنیای امروز بشید، با شما صحبت کنم. این که چطور، Programmability تو شبکه و چگونگی ورود به Cloud را کشف کنید. ما به مهندسی شبکه در دنیای امروز و مهارت های مورد نیاز اون و چگونگی اون ها نگاه خواهیم کرد. این دوره، Cisco DevNET Associate یا اصول برنامه نویسی شبکه می تونه به شما در تسریع این امر کمک کنه.

مفهوم Network در عصر جدید دچار تغییراتی شده است. یعنی Network قدیمی که می شناسیم دائماً در حال تغییر است و از حالت Traditional به حالت Programmability در حال مهاجرت می باشد. مباحثی که جدیداً در بحث Network مطرح می شود شامل، بحث Network as a Service (NaaS)، Software Defined Networking (SDN)، Network Functions Virtualization (NFV) و بحث ترکیب Programmability یا برنامه نویسی در شبکه است.

Network as a Service شامل اجزاء مختلفی می شود. ۲ جزء اصلی NaaS، NFV یا Network Function Virtualization و SDN یا Software Defined Networking است.

NFV یا Network Functions Virtualization در واقع مجازی سازی شبکه در سازمان ها است. یعنی در یک سازمان خبری از Firewall ها، Switch ها، Router ها و ... نیست. در واقع Device های فیزیکی وجود ندارند. در عوض با یک Server، تمامی وظایف این Device ها به صورت Virtual به عنوان یک Service در حال اجرا هستند. Cisco برای راه اندازی این Service، راه کار ارائه می کند. مثل استفاده از Switch های Nexus 1000V، Router های JSRV، ASA v ها و vWAAS.

هر Network Device و از هر Vendor، که در شبکه در حالت Traditional استفاده می شود، دارای ۳ تا Plane مختلف است. Control Plane، Data Plane، Management Plane و Control Plane. Control Plane می است که به عنوان مثال، مسئولیت ساخت و ایجاد Routing Table ها را دارد. Data Plane، Data Plane می است که مسئول انتقال Data اصلی و Physically در بستر شبکه است. Management Plane، Console هایی در اختیار ما به عنوان Network Engineer قرار می دهد، مثل CLI، SSH، SNMP و ... که توسط آن ها می توان Device های شبکه را Management نمود. در شبکه هایی که به صورت SDN یا Software Defined Networking مدیریت می شود، Control Plane از Device برداشته می شود و مدیریت آن به صورت متمرکز توسط یک Controller انجام می شود. این Controller، عملیات Control Plane همه Device های شبکه را انجام می دهد. در واقع Networking Process که بر روی مغز Device ها قرار دارد را، در یک نقطه متمرکز قرار می دهیم و Device های شبکه را Dummy می بیند. این قابلیت برای Network هایی کاربرد دارد که خیلی Dynamic و بزرگ هستند و روز به روز، براساس نیاز مشتری Configuration، Network در حال تغییر است. مزیت این روش نسبت به حالت Traditional این است که، نیاز نیست Network Engineer تک تک Device های مسیر را Manage بکنند. در واقع SDN منجر به کاهش زمان، مدیریت یکپارچه و کاهش خطای انسانی در شبکه می شود. از طرف دیگر، Controller یک دید کلی نسبت به تمام Network ما خواهد داشت.

Cisco ACI (Application Centric Infrastructure) همان SDN است. این روشی است که Cisco برای راهاندازی شبکه‌ها به صورت SDN ارائه می‌دهد. از طرف دیگر، APIC (Application Policy Infrastructure Controller)، Component از Cisco ACI است که به عنوان SDN Network Controller عمل می‌کند.

API یا Application Program Interface، Interface های Application هستند که توسط آن‌ها می‌توان با Controller به عنوان Network Engineer، صحبت کرد و مدیریت SDN Network را برعهده گرفت. ۲ مدل API در بحث SDN و Controller مطرح می‌شود، Northbound Interface (NBI) و Southbound Interface (SBI).

SBI بخشی از API است که Controller استفاده می‌کند تا به واسطه آن با Network Device ها صحبت کند. در واقع استفاده از یکسری Protocol استاندارد، مثل OpenFlow می‌باشد. البته Cisco نیز بعدها از Protocol انحصاری به اسم Cisco OpFlex استفاده نمود که به عنوان استاندارد هم مطرح شد و جزء Protocol های استاندارد محسوب می‌شود. CLI هم جزء Protocol های SBI محسوب می‌شود که توسط آن می‌توان Device های قدیمی‌تر، که از استانداردهای OpenFlow و OpFlex پشتیبانی نمی‌کنند، شبکه SDN را Manage کرد.

NBI بخشی از API است که User و کاربر شبکه، توسط آن می‌تواند با Controller صحبت کند و Configuration های لازم را بر روی آن اعمال نماید. در این قسمت می‌توان از GUI یا Interface گرافیکی که Controller به ما می‌دهد استفاده کنیم، یا اینکه از JavaScript یا Python برای انتقال اطلاعات بر روی Controller و Configure کردن Device های شبکه استفاده کنیم، یا از Controller، اطلاعات را دریافت کنیم.

برای شروع سفر، می‌خوایم با یه سوال در مورد مهندسی شبکه‌های قدیمی که همه ما از اونجا اومدیم شروع کنیم. در واقع جایی که زمینه همه ما بیرون اومده و این که چطور می‌تونیم به یک مهندس شبکه امروزی تبدیل بشیم. برای انجام این کار می‌خوام شما را با شبکه و شخصیت Carl آشنا کنم. مهندس Carl یک مهندس شبکه است که مهارت‌های زیادی تو مهندسی شبکه داره که اون را تا این لحظه بسیار موفق کرده. مفاهیم لایه ۲ مته STP را بخوبی درک می‌کنه و در لایه ۳ با پروتکل‌های مسیریابی، امنیت، خدمات با طراحی جدید VPN، برنامه‌هایی VoIP و MPLS بخوبی آشناست. اون تمام مهارت‌های شبکه‌ی خودش رو تحت کنترل داره. اون همچنین در طی راهی که داشته، مجبور شده، مهارت‌های برنامه‌نویسی خودش رو کمی تقویت کند. مثل TCL، EEM و ... همچنین با استفاده از اسکریپت‌های بخصوصی به خودکارسازی برخی از پیکربندی‌های شبکه پرداخته.

Meet Carl the Network Engineer

Programming Skills

- TCL
- EEM
- Expect Scripts

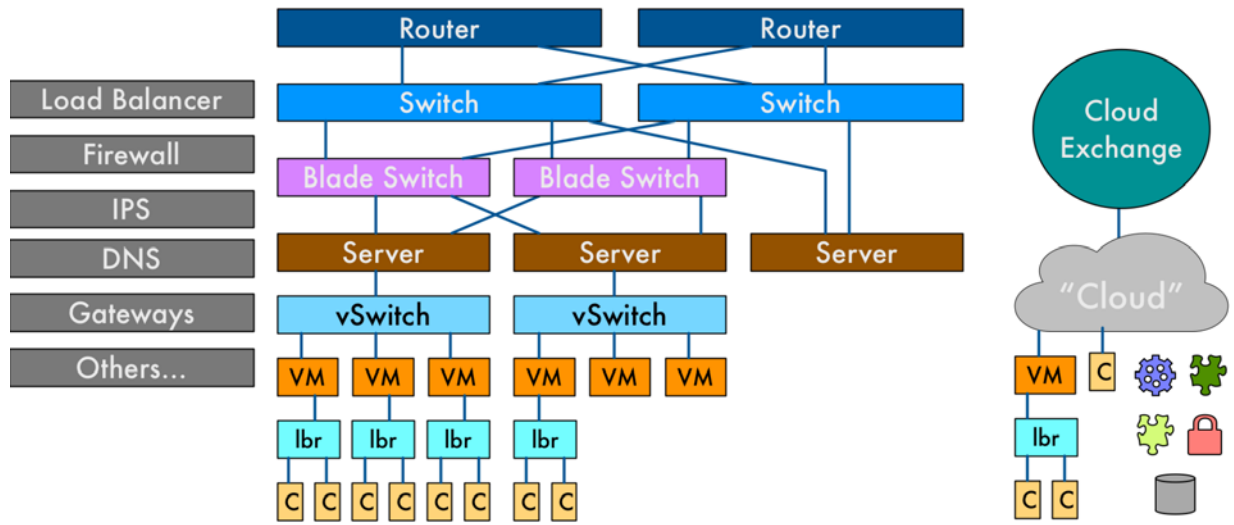


Networking Skills

- Spanning-Tree
- Routing Protocols
- QoS
- VPN Design
- Spanning-Tree
- VOIP
- Fibre Channel
- Security Policy
- MPLS
- Spanning-Tree
- Did I mention Spanning-Tree?

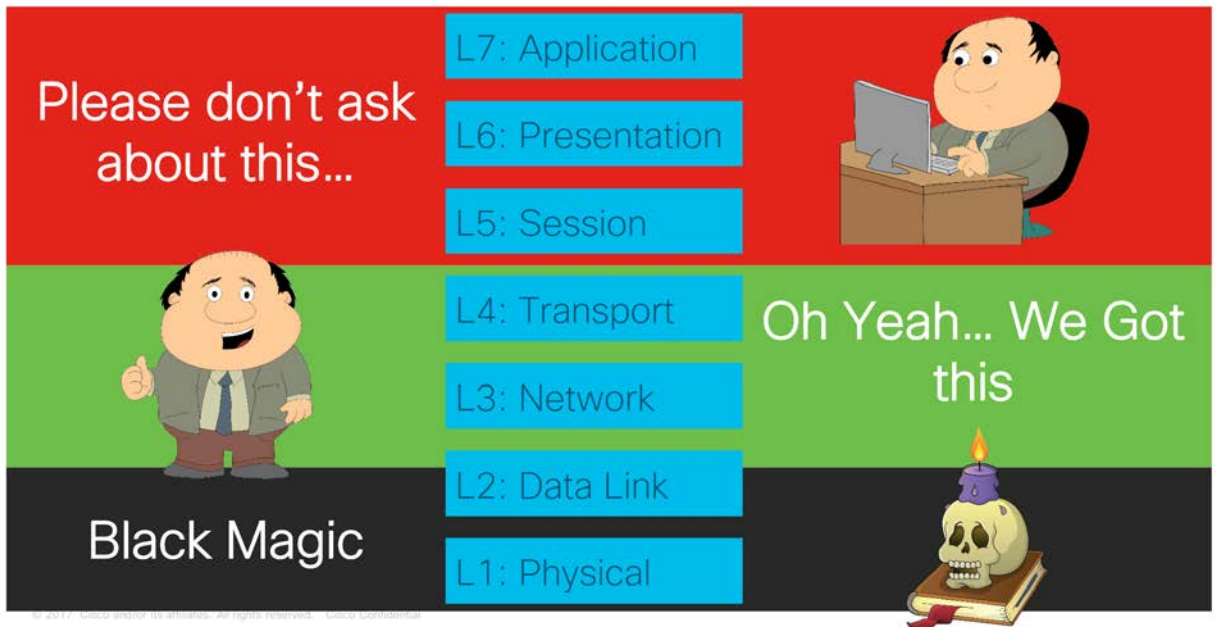
در حال حاضر این مجموعه مهارت‌ها مناسب Carl هستند البته تا زمانی که در شبکه‌ها یعنی در صنعت، Routerها و Switchها و Serverهای متصل به خودش رو داشته باشه. Switchها را که همه ما می‌تونیم به خاطر بیاریم. وقتی شبکه به این شکل به نظر می‌رسید، به سادگی می‌تونستیم به Data Centerها بریم و می‌تونستیم تجهیزات شبکه‌مون رو لمس کنیم. اما با گذشت زمان شبکه شروع به تکامل کرد. اولین تغییرات Virtualization بود. در واقع Vswitchها بودند. در این مرحله Carl و مابقی مهندسين شبکه به اون‌ها نگاه می‌کردن. تصور همه‌ی ما بر این بود Switchهای مجازی یا Vswitchها واقعیت ندارن، مگه می‌شه که نرم‌افزار به تیم Server اجازه بده Vswitchها را کنترل کنن! به هر حال ما به عنوان مهندسين شبکه به Routerها و Switchهای فیزیکی‌مون چسبیدیم. جالب اینجاست که به Vswitchها ختم نشد. بعد از مدتی در شبکه Carl، ارتباط بین Switchها با Serverها از طریق Blade Switchها انجام می‌شد و حالا Carl حتی از Serverها دورتر شد. این موضوع نگران کننده بود چرا که Blade Switchها اغلب از فروشندگان Server تامین می‌شد. بنابراین یه بار دیگه Carl و همکارانش گفتن، می‌دونید چه چیزی اجازه می‌ده تیم Server همچنان به تبادل اطلاعات با کسانی که در شبکه‌های واقعی هستند بپردازن، Routerها و Switchهای فیزیکی، و ما روی اون شبکه فیزیکی که هستیم تمرکز می‌کنیم. خب جالب بود که باز به همین جا ختم نشد! بعداً ما در شبکه‌ها Container و Containerهایی داشتیم به اسم LBR یا Linux Bridge. و در حال حاضر مفاهیم شبکه رو بر پایه Virtual Machineها داریم که خیلی دور از Routerها و Switchهایی فیزیکی که Carl داره مدیریت می‌کنه هستن. در ادامه Cloud و مباحث راینش ابری را داریم که ارتباط بین Data Centerها توی اون انجام می‌گیره. اما بدتر از همه، Virtual Deviceهایی هستن که شبکه فیزیکی تو اون‌ها وجود نداره. همه این Containerها به طور مستقیم به Cloud که به عنوان Network as a Service شناخته می‌شود متصل هستن. فراموش نمی‌کنیم کلیه خدمات و تجهیزات شبکه و امکانات مثل Load Balancerها، Firewallها، IPS و ... تو شبکه وجود دارن که باید با اون‌ها سر و کار داشته باشیم.

The Network...



تمامی این صحبت‌ها، Carl و مهندسين شبکه که خود ما هستيم را به من معرفی می‌کنه. تمامی این موارد رو باید بپذیريم. بيايد يه بار ديگه به دوست قديمی خودمون یعنی مدل OSI نگاهي بندازيم. همه ما مدل OSI رو باید به خاطر داشته باشيم چرا که تفسیر کل دنياي Network بر مبنای اون هست. باید اون رو بخاطر بسپاريم چرا که در تمامی آزمون‌های صدور گواهینامه مته MCSE ، MCSA ، CCNA, CCNP, CCIE و ... به اون نیاز داریم. اما بياين با خودمون صادق باشيم. در واقع فقط سه ناحیه قابل توجه تو این مدل وجود داره.

The OSI Model of Networking...



لایه‌های میانی تو مدل OSI رو داریم، همه ما با اون‌ها آشنایی داریم. موارد و مفاهیم مربوط به Mac Address ها، IP Address ها و Protocol های، TCP و UDP . اما در زیر این مفاهیم، Black Magic را داریم که به لایه فیزیکی معروف است. اتصال الکتریکی و فیزیکی و نحوه برخورد با فیبر، کابل‌ها و Connector ها. همه این موارد در این ناحیه انجام می‌شه. و اگه کسی در مورد لایه‌های بالایی از ما سوال بپرسه، می‌گیم لطفاً از ما درباره این لایه‌ها، سؤال نکنید. اما چالشی که امروز با اون سرو کار داریم، درک هر ۷ لایه شبکه است. امروز برنامه‌های کاربردی تو شبکه در حال اجرا هستند که به ما در درک و ساخت پروتکل‌های مطمئن برای برقرای ارتباط با Device های شبکه کمک می‌کنن، مثه HTTP . ما از عصر حجر شبکه شروع به کار کردیم. یعنی دوره‌ای که توی اون فقط باید نگران مباحث STP ، VLAN و ... می‌بودیم. تا در نهایت به عصر Programmability رسیدیم جایی که به ما تلنگر می‌زنه.



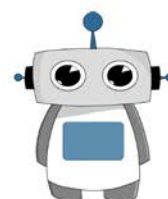
Stone Age
Spanning Tree
VLANs



Bronze Age
Routing Protocols
WAN Design
IP-magedon



The Renaissance
SDN
OpenFlow
Controllers
Overlays
MP-BGP
VXLAN
Micro-Segmentation
White Box

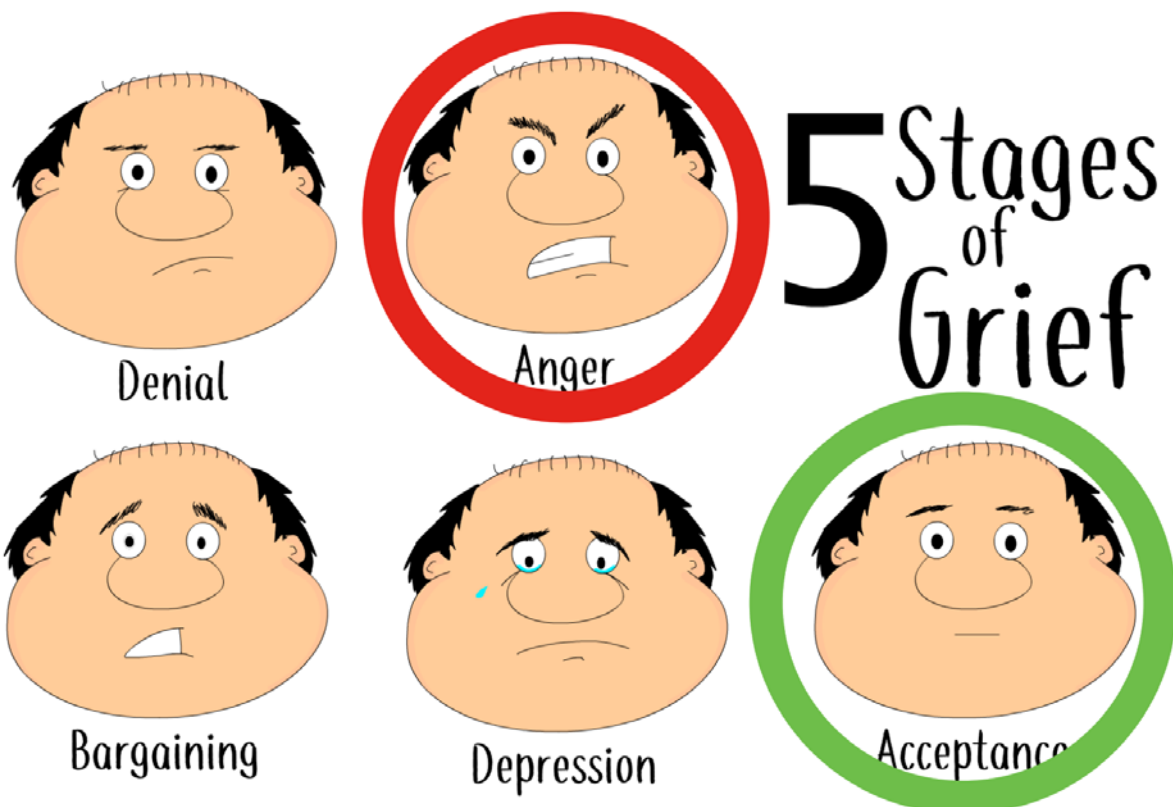


Programmable Age
Cloud
Python
REST / APIs
NETCONF / YANG
"Fabrics"
Network Function
Virtualization (NFV)
DevOps
Containers

The Four Ages of Networking.....

می‌بینیم که پایتون تبدیل به یک زبان جدید برای شبکه شده و مهندسين شبکه باید برای اتوماسیون شبکه با API ها آشنا باشن. برای اتوماسیون شبکه از استانداردهای جدید و فن‌آوری جدیدی مثه Netconf ، REST API باید استفاده بشه که Cisco و غول‌های بزرگ شبکه در حال طراحی و ساختن اون‌ها هستن. اما چرا همچنين تغيير سريع از عصر حجر به عصر Programmability داشتیم؟! برای ما خوبه چرا که در مسیر دیجیتالی شدن و مکانیزمه شدن تمامی کارها هستیم. از شرکت‌ها و تغییراتی که در همه مشاغل ما در حال اتفاق افتادن، تا انتظارات کاربران از نظر اقتصادی و رسیدن به درجه بالاتری از چابکی مطرح هستش. ممکنه این تغییرات سریع شما رو دچار احساس عصبانیت کنه و بگید چرا با چنین شتابی فناوری‌ها در حال حرکت‌اند.

همه این موارد از API ها می‌گذره و شما فقط احساس ناامیدی می‌کنید و عصبانی هستید، که این عالیه! اما ما نمی‌تونیم اونچه رو که در حال تغییر متوقف کنیم. احساس ناامیدی به این معنی که شما فراتر از انکاری که من قبلاً داشتم حرکت کردین. پس واقعا بهتون تبریک می‌گم. هدف من اینه که مهندسین و افراد شبکه به خودشون یه تکونی بدن. حرکت تو مسیر DevNET کمک می‌کنه تا مهندسین شبکه فراتر از خشم حرکت کنن و خودشون رو به سمت پذیرش اون سوق بدن.



این خیلی قابل قبوله که تو اون می‌تونیم با این چالش‌ها مقابله کنیم و واقعاً زمینه رو فراهم کنیم. من اینجا هستم تا در مورد مهندسی شبکه امروزی صحبت کنم. درباره اونچه که یک مهندس شبکه تو دنیای امروز نیاز به اون داره. تو این گفتگو وقتی که در مورد Carl بحث کردیم، Carl با همه روبرو شد. این دقیقاً در مورد من و شما هم صادق. Carl با یک رویکرد سه مرحله‌ای راهی برای رسیدن به اون پیدا کرد. تمام فازهای که اون پشت سر گذاشته رو تو مرحله اول تجزیه و تحلیل کنید. اون برای یه مدت شبکه رو کنار گذاشت و روی برخی برنامه نویسی‌های اصلی متمرکز شد. برخی مهارت‌های پایتون و API ها رو با توجه به قالب‌های داده مته XML، JSON و YAML یاد گرفت و شروع به آزمایش در Source Code های موجود در GitHub کرد. همچنین اون به مهارت‌های لینوکس و Tool های اون نگاهی انداخت و شروع به یادگیری اون‌ها کرد. مته Docker، Ansible، Puppet و Chef. با یه سری از پروتکل‌های جدید شبکه مته NETCONF/YANG و RESTCONF آشنا شد. دقیقاً این یعنی همون مسیری که شما هم مته Carl نیاز به طی کردن اون دارین و با توجه به شرایط شغلی تون باید مته Carl اون رو در آغوش بگیرید.

Carl's 3 Step Approach to Network Programmability

Phase 1

- Python
- REST APIs
- JSON/XML
- git/GitHub



Phase 2

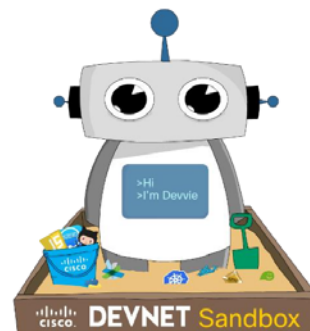
- Linux Skills
- Ansible
- Docker
- NETCONF/YANG

As Needed

- Network Controllers
- IOT Networking
- Cloud Networking
- NFV
- "DevOps"

Phase 3

- Linux Networking
- Container Networking
- NFV



مدت دوره: ۶۰ ساعت

پیش نیاز: CCNA

اهداف دوره:

دوره Cisco DevNET یا اصول برنامه نویسی شبکه به شما کمک می‌کند تا سفر خودتون رو به برنامه‌نویسی شبکه، شروع کنید. این دوره شامل مازول‌های اصلی هستش که کل مناطق مختلف شبکه رو پوشش می‌ده تا شما رو به طور پایه‌ای محکم و استوار کنه، به طوری که شما می‌تونید تو سراسر اتوماسیون و مفاهیم اون با عمق بیشتری نفوذ کنید تا در نهایت به یه مهندس شبکه تو عصر Programmability تبدیل بشین.

اهمیت Python به عنوان یک زبان برنامه‌نویسی برای Network Engineers بسیار مهم است. در صورتی که شبکه SDN داریم، می‌توان از Python استفاده کنیم و Configuration‌های مد نظر را برای Controller ارسال کنیم و در واقع با Controller صحبت کنیم.

حتی در صورتی که در شبکه، SDN استفاده نمی‌کنیم، زبان Python به عنوان Task Automation یا اتوماتیک کردن کارهای روزمره در شبکه Traditional قابل استفاده می‌باشد. تصور کنید می‌خواهیم ۱۰۰۰ تا VLAN را، بروی ۱۰۰ تا Switch بسازیم. اگه بخوایم این Configuration را به صورت Manual بروی Switch‌ها اعمال کنیم، خیلی زمان بر خواهد بود. از طرفی استفاده از VTP، Protocol دارای مشکلات خاص خود می‌باشد، که در دوره‌های CCNA و CCNP به آن پرداخته می‌شود. راه کار بهتر استفاده از یک Script ساده Python است.

Python برای Network Engineers اهمیت دارد به دلیل اینکه، یک زبان برنامه‌نویسی ساده و عمومی است. دلیل مهم‌تر این است که، Industry یا صنعت از آن پشتیبانی می‌کند. مثلاً Cisco Nexus های جدید، Interpreter یا Python Shell بر روی آن قرار دارد و می‌توان مستقیماً بر روی Command Line، Cisco از آن بهره برد.

سرفصل دوره:

- 1- Introduction - How to be a Network Engineer in a Programmable Age
- 2- Why Python
- 3- Python 2x Vs 3x
- 4- Python Interpreter
- 5- Argument Passing
- 6- User Input
- 7- Help Dir
- 8- Variables
- 9- Python Datatypes
- 10- Strings
- 11- Numbers
- 12- Booleans
- 13- Lists
- 14- Sets
- 15- Tuples
- 16- Dictionaries
- 17- If elif else conditions
- 18- For Loops
- 19- While Loops
- 20- Nesting
- 21- Break Continue Pass
- 22- Exceptions
- 23- Functions
- 24- Name Spaces
- 25- Class Object Oriented Programming
- 26- Colorama Module

- 27- Regular Expression Module
- 28- Time Date Module
- 29- Threading Module
- 30- File Operation
- 31- Quick Start Guide to Network Automation (Telnetlib Module, Paramiko Module)
- 32- Netmiko - Use SSH for Network Automation
- 33- NAPALM
- 34- NAPALM and BGP
- 35- Use NAPALM for device configuration audit and changes
- 36- Netmiko Scaling
- 37- Data Formats- Understanding and using JSON, XML and YAML
- 38- APIs are Everywhere... but what are they
- 39- REST APIs - HTTP is for more than Web Browsing
- 40- REST APIs - Making REST API Calls with Postman
- 41- Python - Working with Libraries and Virtual Environments
- 42- Python - Useful Python Libraries for Network Engineers
- 43- Getting the “YANG” of it with Standard Data Models
- 44- Goodbye SNMP hello NETCONF
- 45- Learn to CRUD with GET, POST and DELETE using RESTCONF_with annotation
- 46- NX-API - Get Started with the Native Nexus API
- 47- NX-API - Dive into the Nexus Object Model
- 48- Introducing Cisco DNA Center Platform APIs and Programmability
- 49- Got SDN- Understanding the ACI Programmability Options
- 50- Network Control in the Cloud - Developing with Cisco Meraki
- 51- Cisco DNA Center Platform APIs - Exploring Apps and Tools
- 52- Cisco DNA Center Platform APIs - Network Troubleshooting
- 53- ACI Programmability - The ACI Object Model
- 54- ACI Programmability - Using the ACI Toolkit
- 55- Linux at the Edge- Introduction to Guest Shell
- 56- Python at the Edge- Super Charged Network Event Management

- 57- Package, Deploy and Run Applications in the Network with IOx
- 58- Configuration Management and the Network
- 59- Ansible - What you need to Get Started
- 60- Ansible Quick Start - Ansible infrastructure requirements
- 61- Ansible Quick Start Ad hoc commands
- 62- Ansible Quick Start Raw Playbooks and Cisco IOS Command module Playbooks
- 63- Ansible Quick Start Ansible CLI Playbook
- 64- Ansible Quick Start Playbooks and Cisco IOS Config module
- 65- Cumulus Linux and Ansible Automation