

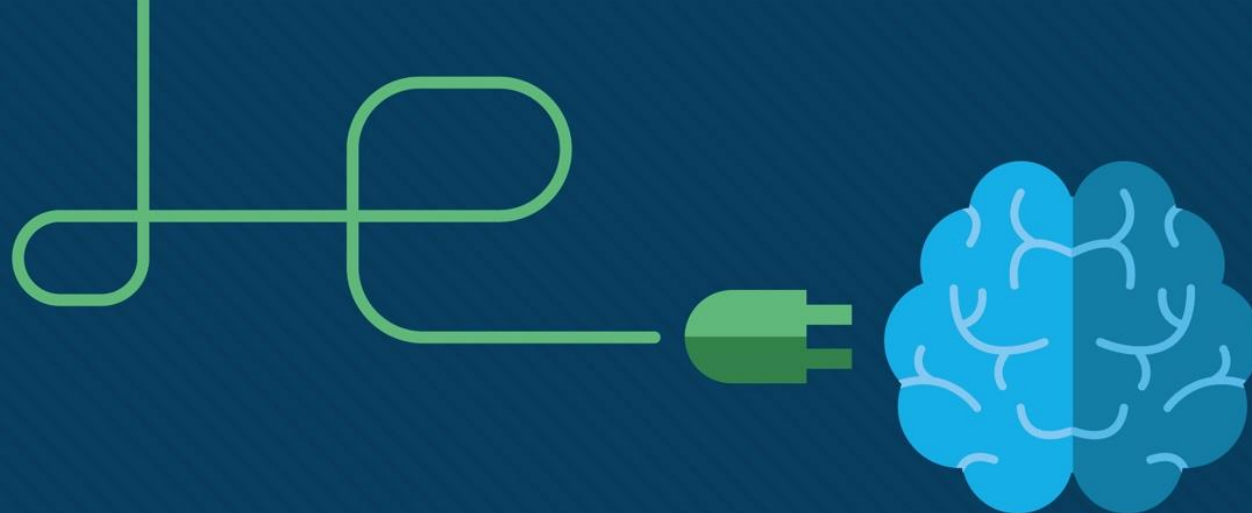
# Virtual Networking Academy Day 2020

11 juni 2020



# Agenda

- 13:00 – 13:15      Opening (Rik Bleeker)
- 13:15 – 14:00      Waarom digitaliseren we de netwerkinfrastructuur? (André Brugman)
- 14:00 – 14:30      NetAcad updates (Willem-Jan Derks)
  
- 14:30 – 14:45      Pauze
  
- 14:45 – 15:30      Network engineering in a programmable age (Hank Preston)
- 15:30 – 15:45      DevOps in het onderwijs (Willem-Jan Derks)
  
- 15:45 – 16:00      Afsluiting



# DevOps in het onderwijs

Networking Academy Day 11 juni 2020  
15:30-15:45

Willem-Jan Derks  
CSR Country Manager Netherlands



# Module 1: Course introduction

<b>Module Title/ Topic Title</b>	<b>Objective</b>
Module 1. Course Introduction	Use basic Python programming and Linux skills.
1.1 Your Lab Environment	Install a virtual lab environment.
1.2 Linux	Manage the Linux file system and permissions.
1.3 Python	Use basic Python programming.

# Module 2: The DevNet Developer Environment

<b>Module Title/ Topic Title</b>	<b>Objective</b>
Module 2. The DevNet Developer Environment	Investigate DevNet
2.1 DevNet Overview	Explain how DevNet encourages communities of network programmers.
2.2 Exploring DevNet Online Resources	Investigate DevNet online resources.

# Module 3: Software Development and Design

Module Title/ Topic Title	Objective
Module 3. Software Development and Design	Use software development and design best practices.
3.1 Software Development	Compare software development methodologies.
3.2 Software Design Patterns	Describe the benefits of various software design patterns.
3.3 Version Control	Implement software version control using GIT.
3.4 Coding Basics	Explain coding best practices.
3.5 Code Review and Testing	Use Python Unit Test to evaluate code.
3.6 Understanding Data Formats	Use Python to parse different messaging and data formats.

# Module 4: Understanding and Using APIs

Module Title/ Topic Title	Objective
Module 4. Understanding and Using APIs	Create a secure REST API.
4.1 Introducing APIs	Explain the use of APIs.
4.2 API Design Styles	Compare synchronous and asynchronous API design styles.
4.3 API Architecture Styles	Describe common API architecture styles.
4.4 Introduction to REST APIs	Explain the functions of REST APIs.
4.5 Authenticating to a REST API	Create a secure REST API.
4.6 API Rate Limits	Explain the purpose of API rate limits.
4.7 Working with Webhooks	Explain the use of webhooks.
4.8 Troubleshooting API Calls	Explain how to troubleshoot REST APIs

# Module 5: Network Fundamentals

Module Title/ Topic Title	Objective
Module 5. Network Fundamentals	Explain the processes and devices that support network connectivity.
5.1 Introduction to Network Fundamentals	Explain basic network terms and processes.
5.2 Network Interface Layer	Explain the features and functions of the OSI network layer.
5.3 Internetwork Layer	Explain the features and functions of the OSI internetwork layer.
5.4 Network Devices	Explain the features and functions of common network devices.
5.5 Networking Protocols	Explain common networking protocols.
5.6 Troubleshooting Application Connectivity Issues	Troubleshoot basic network connectivity.



# Module 6: Application Deployment and Security

Module Title/ Topic Title	Objective
Module 6. Application Deployment and Security	Use current technologies to deploy and secure applications and data in a cloud environment.
6.1 Understanding Deployment Choices with Different Models	Explain common cloud deployment models.
6.2 Creating and Deploying a Sample Application	Use container technology to deploy a simple application.
6.3 Continuous Integration/Continuous Deployment (CI/CD)	Explain the use of Continuous Integration/Continuous Deployment (CI/CD) in application deployment.
6.4 Networks for Application Development and Security	Explain the network technology required for application development in a cloud environment.
6.5 Securing Applications	Use common application security techniques to secure data.

# Module 7: Infrastructure and Automation

Module Title/ Topic Title	Objective
Module 7. Infrastructure and Automation	Compare software testing and deployment methods in automation and simulation environments.
7.1 Automating Infrastructure with Cisco	Describe deployment environments that benefit from automation.
7.2 DevOps and SRE	Explain the principles of DevOps
7.3 Basic Automation Scripting	Describe the use of scripting in automation.
7.4 Automation Tools	Explain automation tools that speed the development and deployment of code.
7.5 Infrastructure as Code	Explain the benefits of storing infrastructure as code.
7.6 Automating Testing	Explain how automation tools are used in the testing of application deployments.
7.7 Network Simulation	Describe the use of the Cisco VIRL network simulation test environment.

# Module 8: Cisco Platform and Development

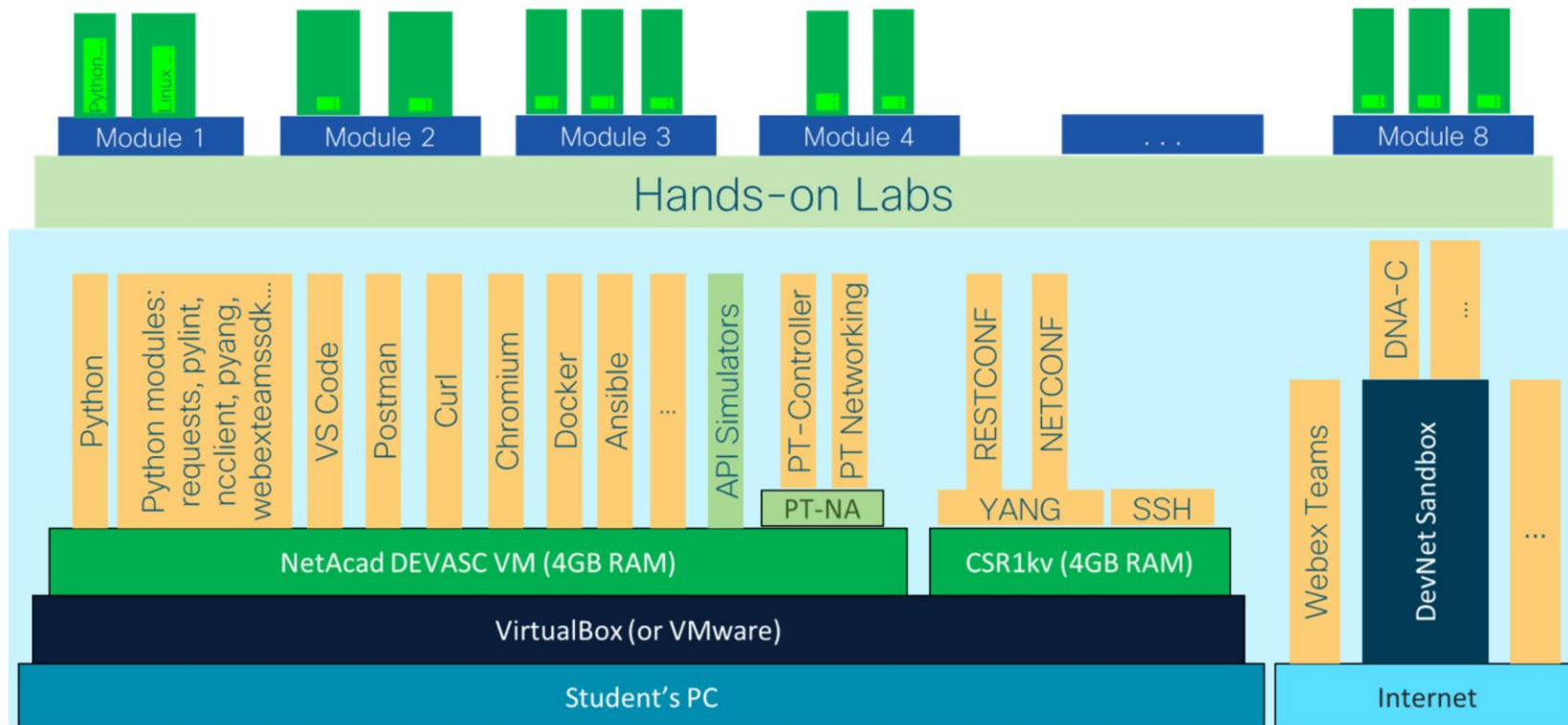
Module Title/ Topic Title	Objective
Module 8. Cisco Platforms and Development	Compare Cisco platforms used for collaboration, infrastructure management, and automation.
8.1 Introduction to Cisco Platforms	Describe the Cisco API platform.
8.2 Cisco SDKs	Explain how Cisco SDKs assist in the development of applications.
8.3 Understanding Network Programmability and Device Models	Compare network programmability models.
8.4 Cisco Network Management	Compare Cisco network management platforms.
8.5 Cisco Compute Management	Describe Cisco compute management solutions.
8.6 Cisco Collaboration Platforms	Describe Cisco collaboration platforms.
8.7 Cisco Security Platforms	Describe Cisco security platforms.



# DevAsc VM

2 Virtual Machines:

- Cisco CSR1000v
- Student's Lab VM bundled with all the software tools including Packet Tracer for Network Automation



# Packet Tracer for Network Automation

The screenshot displays a Packet Tracer environment with three main components:

- Python Script (getTicket.py):** A script that uses the `requests` library to send a POST request to a controller. The script defines two controller URLs, authentication data, and headers. It iterates over the URLs and prints the response if the status code is 200.
- Network Diagram:** A logical network diagram showing a central `NetworkController PT-Controller1` (IP: 10.0.0.254) connected to a `3550-24-S Multilayer Switch0` (IP: 10.0.0.3). The switch is connected to several `ISR4331 R-Left` and `ISR4331 R-LAN` routers, which are in turn connected to a `2960-24-TT SW-LAN` switch. This switch is connected to four `PC-PT` devices (PC0, PC1, PC2, PC3).
- Web Browser (PC0):** A browser window showing the Cisco Network Controller interface. It displays a table of host devices connected to the network.

Host Device		Connected Network Device				
MAC	IP	Hostname	Type	IP	Hostname	Port
0060.2F33.D7B3	10.0.1.129	PC2	Pc	10.0.12	SW-LAN	FastEthernet0/3
00D0.FFG6.A201	10.0.1.131	PC1	Pc	10.0.12	SW-LAN	FastEthernet0/2
0090.0C76.101A	10.0.1.132	PC3	Pc	10.0.12	SW-LAN	FastEthernet0/4
0001.972C.908B	10.0.1.130	PC0	Pc	10.0.12	SW-LAN	FastEthernet0/1
0001.6333.9280	10.0.0.4	PC5	Pc	10.0.0.3	SW-Left3550	GigabitEthernet1/0/1
0090.21B2.6DC4	10.0.1.128	PC4	Pc	10.0.12	SW-LAN	FastEthernet0/5

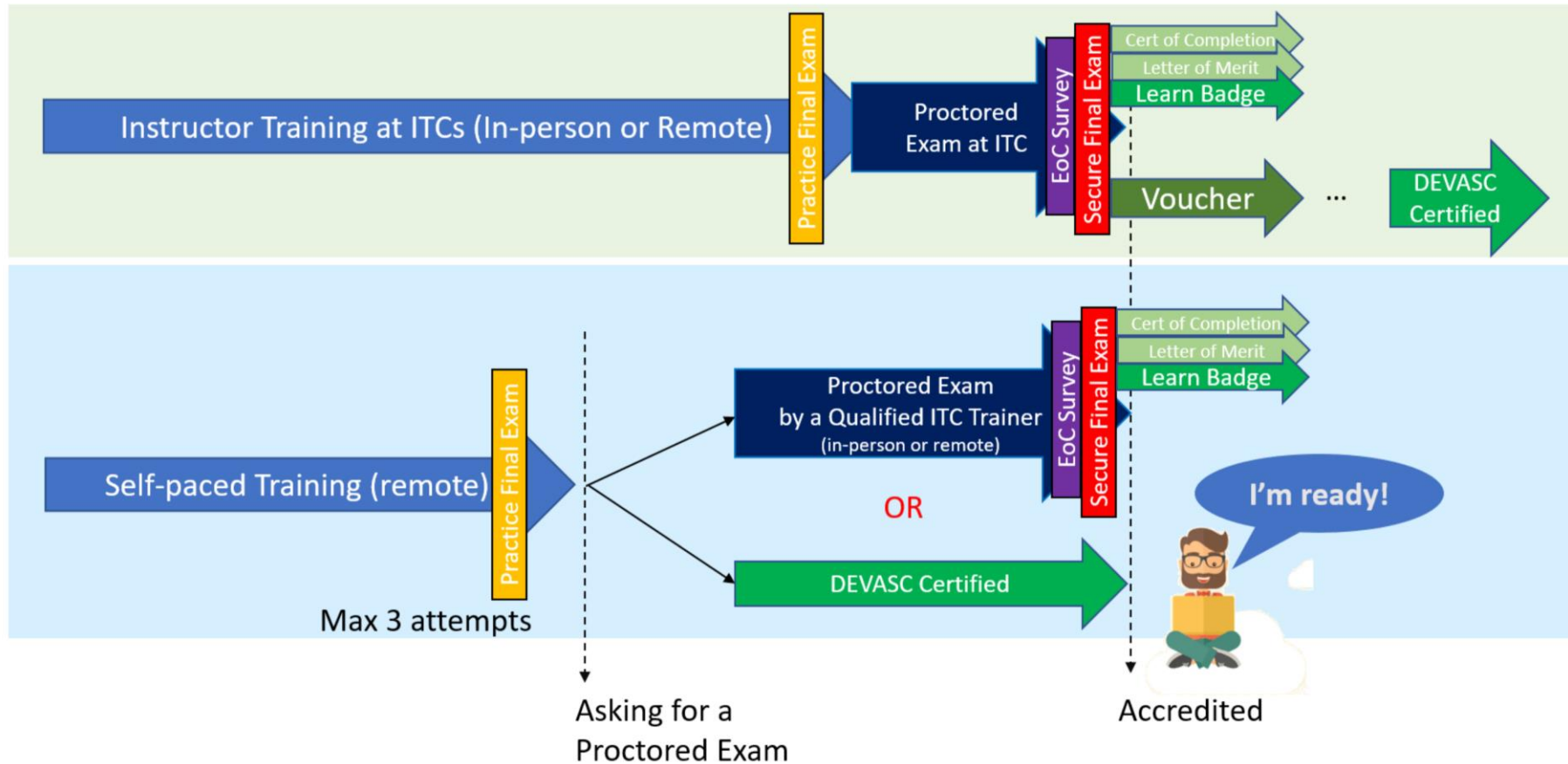
```
getTicket.py > ...
1 import requests
2 import json
3
4 # Controller1 = http://127.0.0.1:5001
5 # Controller2 = http://127.0.0.1:5002
6 URLOfController1 = "http://127.0.0.1:5001/api/v1/ticket"
7 URLOfController2 = "http://127.0.0.1:5002/api/v1/ticket"
8
9 authData = {
10     "username": "cisco",
11     "password": "cisco"
12 }
13 headers = {
14     "content-type": "application/json"
15 }
16 for URL in [URLOfController1, URLOfController2]:
17     r = requests.post(URL,
18                     headers=headers,
19                     data=json.dumps(authData),
20                     verify = False
21 )
22     if r.status_code == 200:
23         print(
24             r.json()['response']['serviceTicket']
25         )
26
27
28
29
30
31
32
```

```
getTicketInPT (Python) - main.py
1 import requests
2 import json
3
4 URLOfController1 = "http://10.0.0.254/api/v1/ticket"
5
6 authData = {
7     "username": "cisco",
8     "password": "cisco"
9 }
10 headers = {
11     "content-type": "application/json"
12 }
13 for URL in [URLOfController1]:
14     r = requests.post(URL,
15                     headers=headers,
16                     data=json.dumps(authData),
17                     verify = False
18 )
19     if r.status_code == 200:
20         print(
21             r.json()['response']['serviceTicket']
22         )
23
```

Starting getTicketInPT (Python)...  
WC-12-d6b44352ef94dbb413nbi  
getTicketInPT (Python) finished running.



# DevAsc Instructor Training



# Keuzedelen DevOps

- Keuzedeel Goedgekeurd op 9 Juni!
- DevOps voor IT Experts & DevOps voor Software Developers
- Indieners
  - Cisco
  - FastLane
  - Friesland College
  - Graafschap College
  - Noorderpoort
  - ROC v Amsterdam
  - ROC Midden Nederland
  - Techniek College Rotterdam



# Keuzedelen DevOps

## IT Expert

### Kerntaken/werkprocessen

Wat zie je de beroepsbeoefenaar doen op de werkvloer?

Gebruikmakend van de eigen vakkennis wordt een netwerk geautomatiseerd, waardoor resources heel flexibel ingezet kunnen worden. Hierbij wordt gebruik gemaakt van kleine development cyclussen volgens de DevOps principes

Hiervoor:

- Wordt een concept design gemaakt van de opzet of wijzigingen
- Wordt een CI/CD pijplijn toegepast
- Wordt een softwareoplossing gerealiseerd,
- Wordt de software oplossing uitgerold op netwerk apparatuur,
- Wordt het geautomatiseerde netwerk getest,
- Wordt de productie omgeving gemonitord

### Vaardigheden

Welke vaardigheden heeft de beroepsbeoefenaar nodig om de kerntaken/werkprocessen uit te voeren?

Om een netwerk te kunnen automatiseren moet de devops beschikken over de volgende vaardigheden:

- Informatiebehoefte van de klantorganisatie kunnen inventariseren,
- Problemen inzake dataopslag / communicatie kunnen oplossen,
- Risico's kunnen inschatten zoals downtime, capaciteit en security
- Social skills zoals samenwerken, presenteren en anticiperen

### Kennis

Welke kennis heeft de beroepsbeoefenaar nodig om de kerntaken/werkprocessen uit te voeren?

- Specialistische kennis over een programmeertaal zoals bijvoorbeeld Python
- Kennis over versie beheer binnen het programmeren
- Kennis over het gebruik van API's
- Kennis over het programmeren van netwerk infrastructuur devices
- Kennis over containers (Docker)
- Kennis over Kubernetes

## Software Developer

### Kerntaken/werkprocessen

Wat zie je de beroepsbeoefenaar doen op de werkvloer?

Gebruikmakend van de eigen vakkennis wordt het gehele traject van development tot aan uitrollen van een applicatie. Hierbij wordt gebruik gemaakt van kleine development cyclussen volgens de DevOps principes

Hiervoor:

- Wordt een concept design gemaakt van de opzet of wijzigingen
- Wordt een CI/CD pijplijn toegepast
- Wordt een softwareoplossing gerealiseerd,
- Wordt een softwarematige oplossing uitgerold in een productie omgeving
- Wordt de productie omgeving gemonitord

### Vaardigheden

Welke vaardigheden heeft de beroepsbeoefenaar nodig om de kerntaken/werkprocessen uit te voeren?

Om een omgeving te realiseren moet de devops beschikken over de volgende vaardigheden:

- Informatiebehoefte van de klantorganisatie kunnen inventariseren
- Problemen inzake dataopslag / communicatie kunnen oplossen
- Risico's kunnen inschatten zoals downtime, capaciteit en security
- Social skills zoals samenwerken, presenteren en anticiperen

### Kennis

Welke kennis heeft de beroepsbeoefenaar nodig om de kerntaken/werkprocessen uit te voeren?

- Kennis van netwerkarchitectuur en netwerk platformen
- Kennis van API's
- Kennis over Cloud omgevingen
- Kennis over virtuele machines.
- Kennis over containers. (Docker)

# Vragen?



