cisco Live!
Let's go
#CiscoLiveAPJC

The bridge to possible

Breaking down the DevNet Expert Lab Exam

Everything you need to know to get ready to test!

Hank Preston, Principal Engineer Learning and Certifications
ccie 38336 | devnet expert 20220001
@hfpreston

BRKDEV-2014

Cisco Webex App

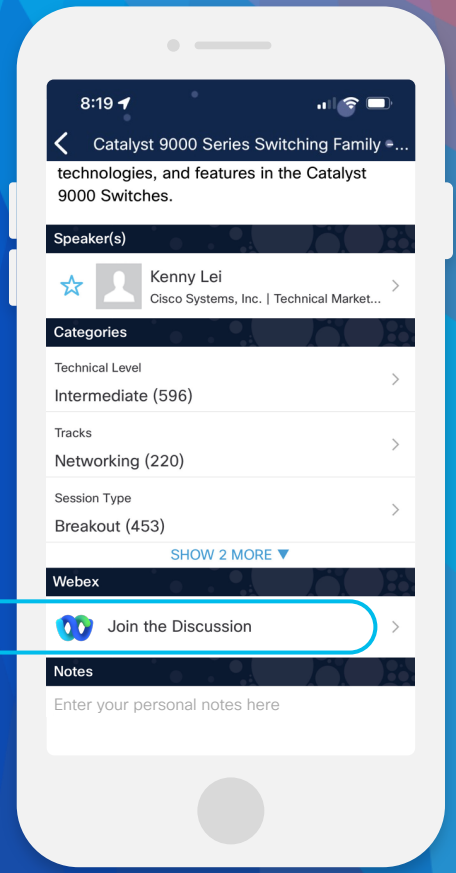
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until December 22, 2023.



https://ciscolive.ciscoevents.com/ciscolivebot/#BRKDEV-2014

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Hank Preston

- First GitHub commit on December 13, 2014
- Once used Legos, a ball of string, IR819, and an Arduino to demonstrate why network programmability mattered
- Joined DevNet in December 2016 to make network automation less scary
- Spent the last 5 years applying NetDevOps to DevNet's and Learning's lab systems
- Why do something by hand in 30 minutes when you can spend 8 hours automating it?

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

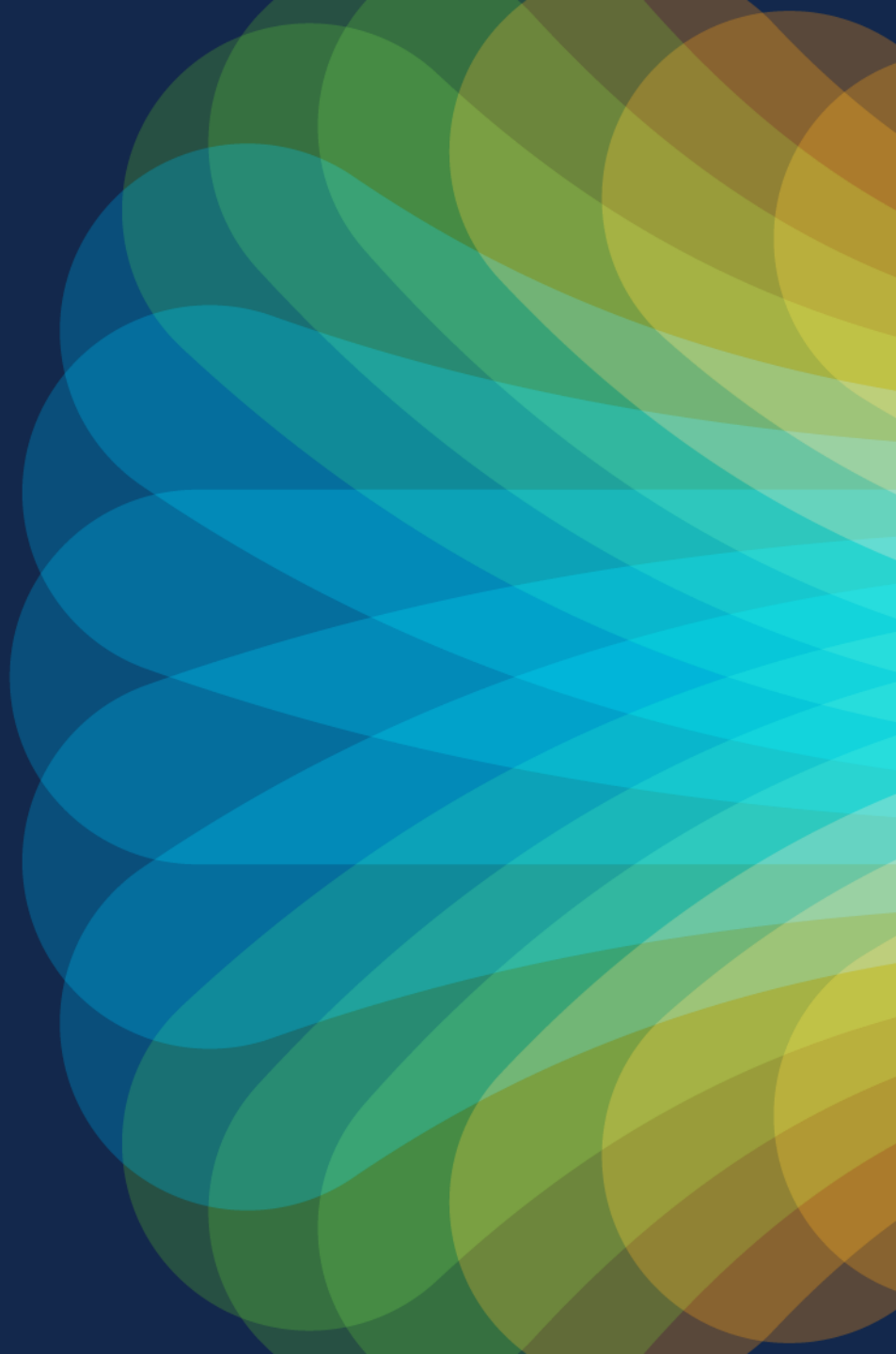
Some fun with Trivia Time!!!

Agenda

- Who is the DevNet Expert for?
- What does it take to become a DevNet Expert?
- What topics are covered on the lab exam?
- What is the lab exam format?
- What kinds of questions will I see?
- What questions do YOU have?

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Who is the DevNet Expert for?



Senior level infrastructure automation engineers

Expert (CCIE) in building and designing networks

Understand how networks work

5+ years experience doing network automation

Build and use scripts to make my job easier

Build applications, scripts, utilities used by OTHER people

Advanced to expert level Python development

Comfortable working with git on collaborative projects

Designed applications for hybrid cloud deployment

Spend all day, every day writing business apps

Comfortable building and using REST APIs

Can make API calls to ACI, Webex and Meraki without documentation

Experience automating variety of Cisco platforms

Fluent in multiple Infrastructure as Code tools

Not scared by NETCONF, RESTCONF, YANG

Prefer your telemetry model driven

Experience building CICD pipelines for projects

Deployed example Docker apps to Minikube

Experienced with packaging and deploying applications with Docker and Kubernetes

What does it take to become a DevNet Expert?

Qualification Exam

- Developing Applications using Cisco Core Platforms and APIs 350-901 DEVCOR*
- 120 minute multiple choice exam

* Same exam required to earn DevNet Professional

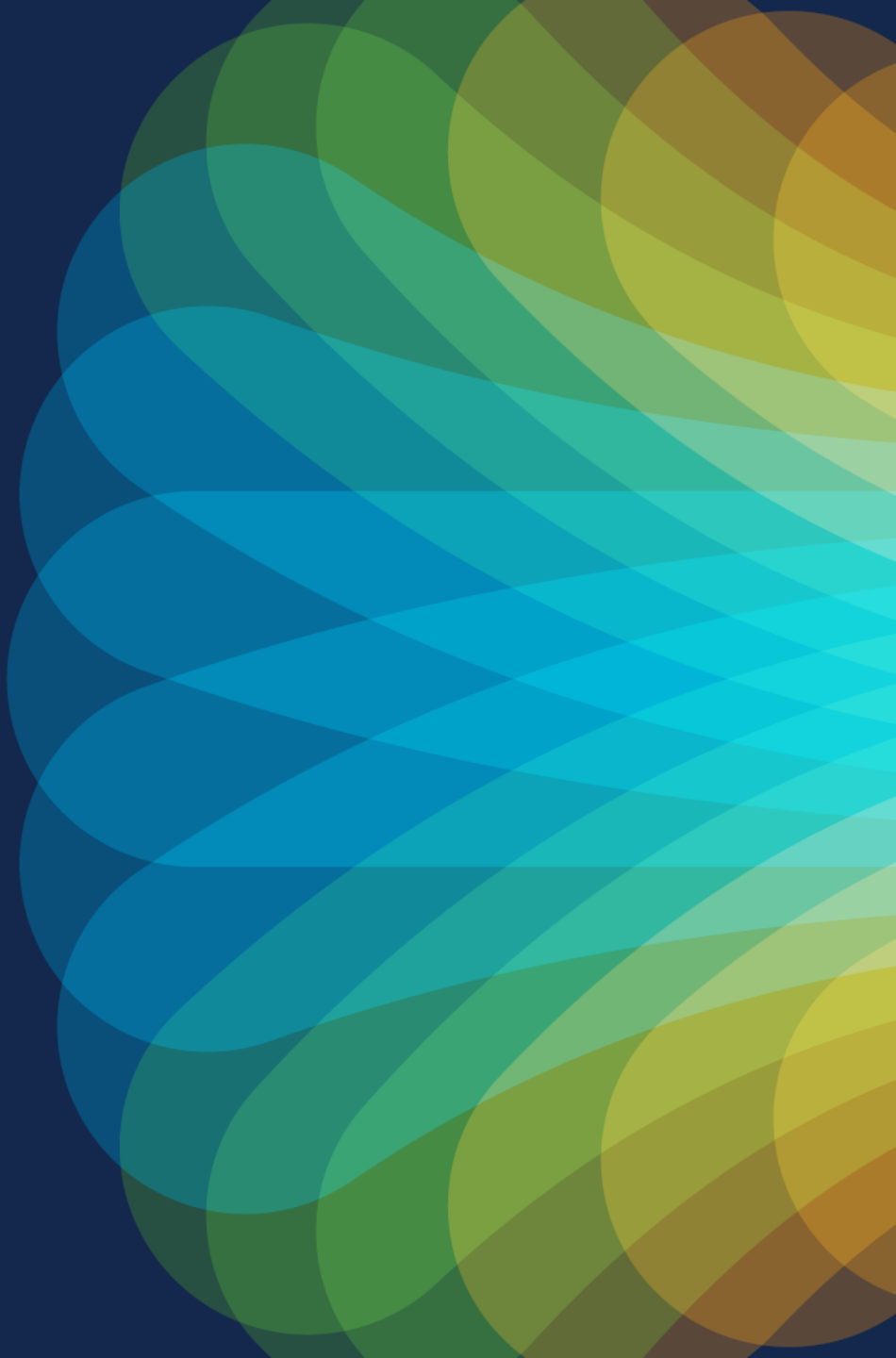
Lab Exam

- DevNet Expert (v1.0) Lab Exam
- 8 hour practical exam
  - 3 hour Design module
  - 5 hour Develop, Test, Deploy, Maintain module

https://learningnetwork.cisco.com/s/devnet-expert

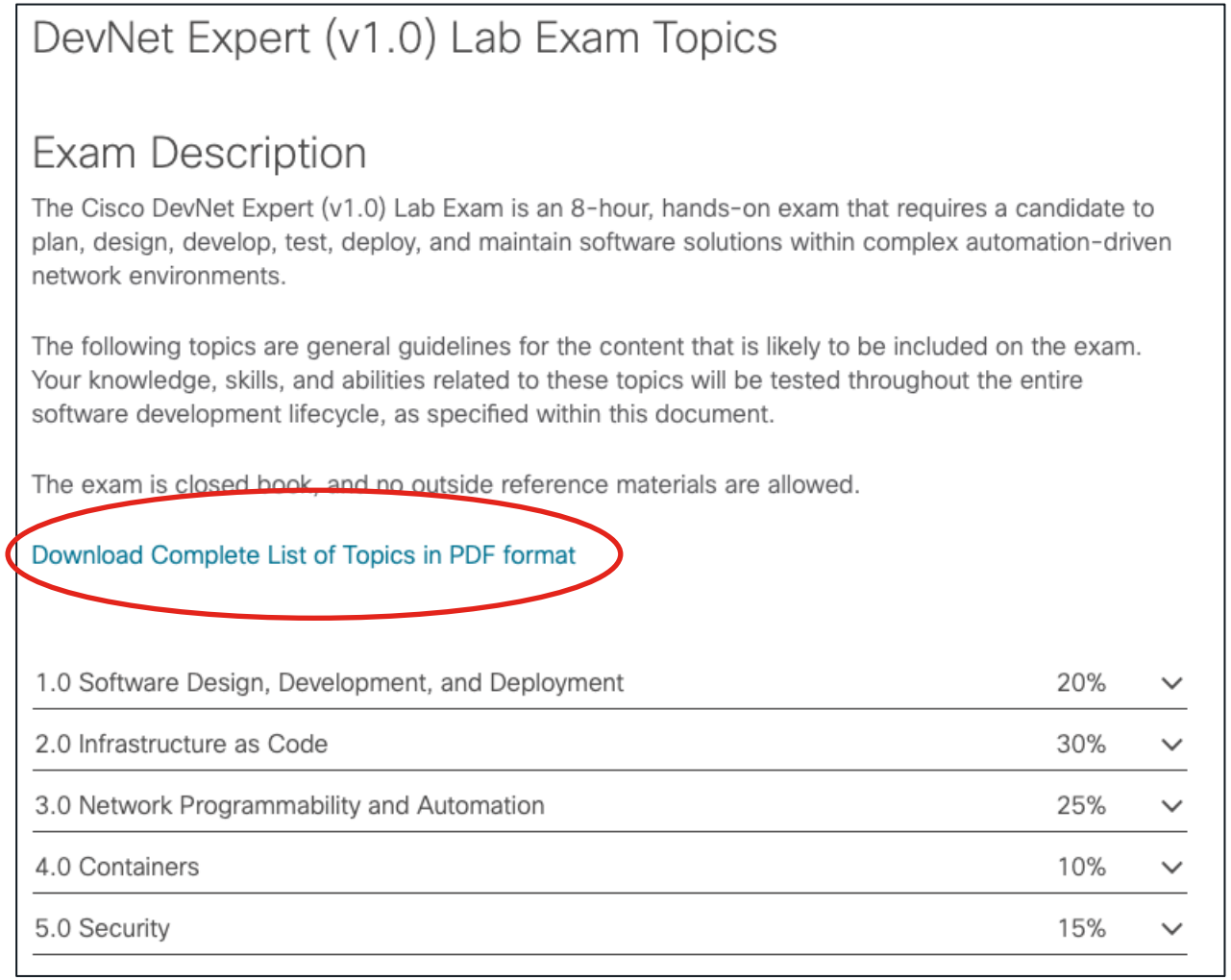
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

What topics are covered on the lab exam?



Exam Blueprint

https://learningnetwork.cisco.com



DevNet Expert (v1.0) Lab Exam Topics

Exam Description

The Cisco DevNet Expert (v1.0) Lab Exam is an 8-hour, hands-on exam that requires a candidate to plan, design, develop, test, deploy, and maintain software solutions within complex automation-driven network environments.

The following topics are general guidelines for the content that is likely to be included on the exam. Your knowledge, skills, and abilities related to these topics will be tested throughout the entire software development lifecycle, as specified within this document.

The exam is closed book, and no outside reference materials are allowed.

Download Complete List of Topics in PDF format

1.0 Software Design, Development, and Deployment	20%
2.0 Infrastructure as Code	30%
3.0 Network Programmability and Automation	25%
4.0 Containers	10%
5.0 Security	15%

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Deciphering the Blueprint:
Implementing Cisco Enterprise Network Core Technologies (350-401)

Weight	Domain
20%	Software Design, Development and Deployment

1.1 **Design** a solution based on an on-premises, hybrid, or public cloud deployment, considering these factors: — Task

- 1.1.a Deployment: maintainability, modularity
- 1.1.b Reliability: high availability and resiliency
- 1.1.c Performance: scalability, latency, and rate limiting
- 1.1.d Infrastructure: monitoring, observability, and metrics

Subtasks

Verb

1.2 **Recommend** a deployment strategy to mitigate risk and impact on service availability, for a given scenario

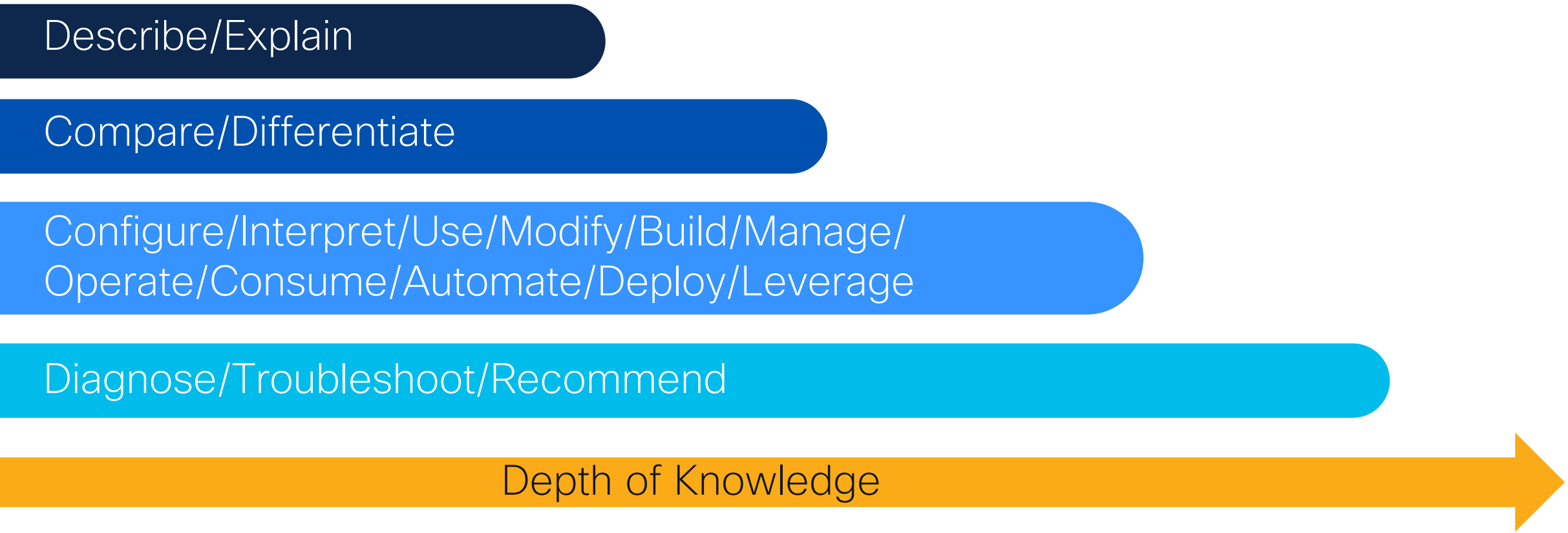
1.3 **Modify** an existing network automation solution based on business and technical requirements (includes gap analysis)

1.4 **Use** Git in a CI/CD development workflow

1.5 **Troubleshoot** issues with a CI/CD pipeline

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Blueprint Verbs



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

DevNet Expert Lab Exam v1.0

Exam Blueprint

Five major areas of knowledge

Domain / Weight	
1. Software Design, Development & Deployment	20%
2. Infrastructure as Code	30%
3. Network Programmability and Automation	25%
4. Containers	10%
5. Security	15%

Make note of domain weights

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

20%

Tasks

Domain 1: Software Design, Development and Deployment

1.1 Design a solution based on an on-premises, hybrid, or public cloud deployment, considering these factors:

- 1.1.a Deployment: maintainability, modularity
- 1.1.b Reliability: high availability and resiliency
- 1.1.c Performance: scalability, latency, and rate limiting
- 1.1.d Infrastructure: monitoring, observability, and metrics

1.2 Recommend a deployment strategy to mitigate risk and impact on service availability, for a given scenario

1.3 Modify an existing network automation solution based on business and technical requirements (includes gap analysis)

1.4 Use Git in a CI/CD development workflow

1.5 Troubleshoot issues with a CI/CD pipeline

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

30%

Tasks

Domain 2: Infrastructure as Code

(page 1 of 3)

2.1 Create a scalable solution for infrastructure automation (considering areas such as network impact, risk, and tool selection)

2.2 Build, manage, and operate a Python-based REST API with a web application framework (endpoints, HTTP request, and response)

2.3 Build, manage, and operate a Python-based CLI application to use a REST API

2.4 Consume and use a new REST API, given the documentation

2.5 Create a RESTCONF or NETCONF payload based on a given YANG module, and interpret the response

2.6 Create a NETCONF filter by using XPath

2.7 Configure network devices on an existing infrastructure by using NETCONF or RESTCONF, given YANG analysis tools

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

30%

Tasks

Domain 2: Infrastructure as Code

(page 2 of 3)

2.8 Create and use a role by utilizing Ansible to manage infrastructure, given support documentation

- 2.8.a Loop control
- 2.8.b Conditionals
- 2.8.c Use of variables and templating
- 2.8.d Use of connection plug-ins such as network CLI, HTTPAPI, and NETCONF

2.9 Use Terraform to statefully manage infrastructure, given support documentation

- 2.9.a Loop control
- 2.9.b Resource graphs
- 2.9.c Use of variables
- 2.9.d Resource retrieval
- 2.9.e Resource provision
- 2.9.f Management of the state of provisioned resources

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

30%

Tasks

Domain 2: Infrastructure as Code

(page 3 of 3)

2.10 Create a basic Cisco NSO service package to meet given business and technical requirements. The service would generate a network configuration on the target device platforms using the "cisco-ios-cli" NED and be of type "python-and-template"

- 2.10.a Create a service template from a provided NSO device configuration
- 2.10.b Create a basic YANG module for the service containers (including lists, leaf lists, data types, leaf references, and single argument "when" and "must" conditions)
- 2.10.c Create basic actions to verify operational status of the service
- 2.10.d Monitor service status by reviewing the NCS Python VM log file

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

25%

Tasks

Domain 3: Network Programmability and Automation

(page 1 of 3)

3.1 Create, modify, and troubleshoot scripts by using Python libraries and SDK documentation to automate against APIs (ACI, AppDynamics, DNA Center, FDM, Intersight, IOS XE, Meraki, NSO, Webex)

3.2 Automate the configuration of a Cisco IOS XE network device (based on a provided architecture and configuration), including these components:

- 3.2.a Interfaces
- 3.2.b Static routes
- 3.2.c VLANs
- 3.2.d Access control lists
- 3.2.e BGP peering
- 3.2.f BGP and OSPF routing tables
- 3.2.g BGP and OSPF neighbors

3.3 Deploy an application on a Cisco IOS XE device by leveraging the technologies of Guest Shell and application hosting

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

25%

Tasks

Domain 3: Network Programmability and Automation

(page 2 of 3)

3.4 Modify and troubleshoot an automated test by using pyATS to meet requirements

3.4.a Create a testbed file for connecting to Cisco IOS, IOS XE, or NX-OS devices

3.4.b Gather current configuration and operational state from devices using the Genie parsers and models included with pyATS

3.4.c Develop and execute test jobs and scripts using AEtest to verify network health

3.5 Design a model-driven telemetry solution based on given business and technical requirements by using gNMI dial-in, gRPC dial-out, and NETCONF dial-in

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

25%

Tasks

Domain 3: Network Programmability and Automation

(page 3 of 3)

3.6 Create YANG model-driven telemetry subscriptions

- 3.6.a Identify model elements and cadence
- 3.6.b On-change or event drive
- 3.6.c Optimize frequency
- 3.6.d Dial-out subscription
- 3.6.e Secure telemetry streams
- 3.6.f Confirm data transmission
- 3.6.g Identify network issues and make changes

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

10%

Tasks

Domain 4: Containers

(page 1 of 2)

4.1 Create a Docker image (including Dockerfile)

- 4.1.a From a provided image
- 4.1.b Expose ports
- 4.1.c Add or copy files
- 4.1.d Run commands during image build
- 4.1.e Manipulate entry point and initial commands
- 4.1.f Establish working directories
- 4.1.g Environment variables as part of a definition to control an application
- 4.1.h Docker ignore file
- 4.1.i Volumes

4.2 Package and deploy a solution by using Docker Compose

- 4.2.a Deploy and manage containers
- 4.2.b Define services, networks, volumes, and links

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

10%

Tasks

Domain 4: Containers

(page 2 of 2)

4.3 Package and deploy a solution by using Kubernetes
- 4.3.a Use deployments, secrets, services, ingress, volumes, namespaces, and replicas
- 4.3.b Manage the lifecycle of pods (e.g., scale up, scale down, help status, logs)
- 4.3.c Monitor pods by building health checks)
- 4.3.d Use the kubectl interface

4.4 Create, consume, and troubleshoot a Docker host and bridge-based networks and integrate them with external networks

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

15%

Tasks

Domain 4: Security

5.1 Leverage OWASP secure coding practices into all solutions to meet given requirements

- 5.1.a Input validation
- 5.1.b Authentication and password management
- 5.1.c Access control
- 5.1.d Cryptographic practices
- 5.1.e Error handling and logging
- 5.1.f Communication security

5.2 Create a certificate signing request (CSR) by using OpenSSL; send CSR to a provided certificate authority; and use the certificate to secure a web application

5.3 Use OAuth2+ to obtain an authentication token

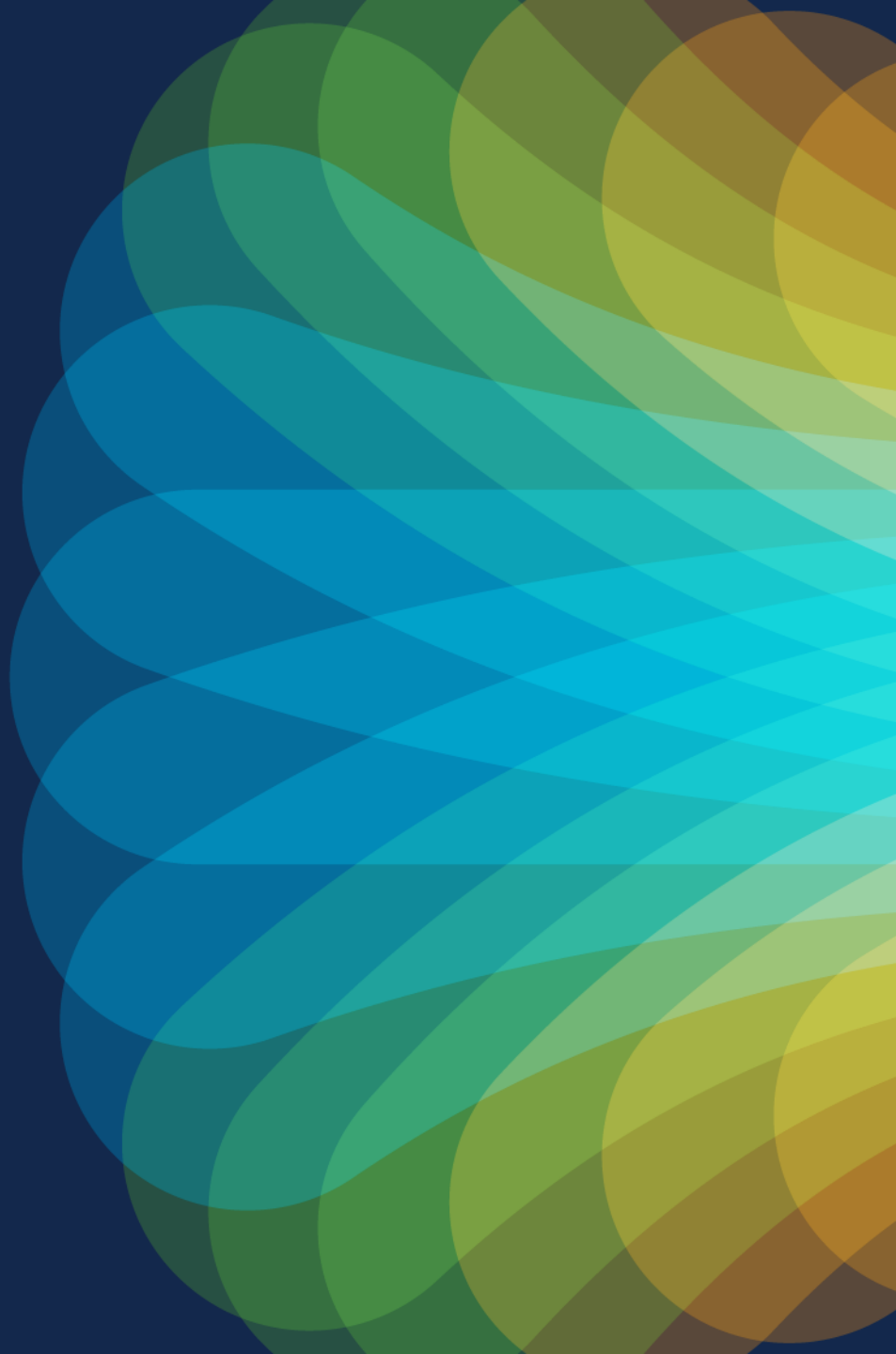
5.4 Use a secret management system to secure an application

5.5 Use tokens, headers, and secrets to secure a REST API

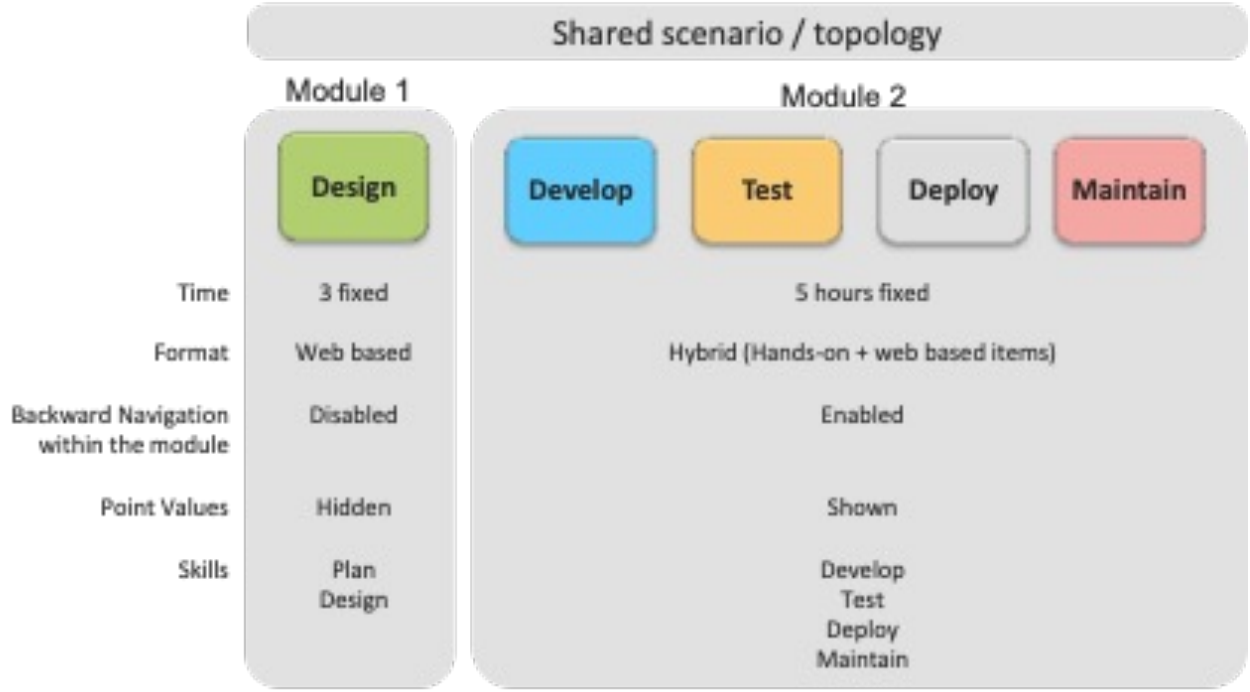
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Some fun with Trivia Time!!!

What is the lab exam format?



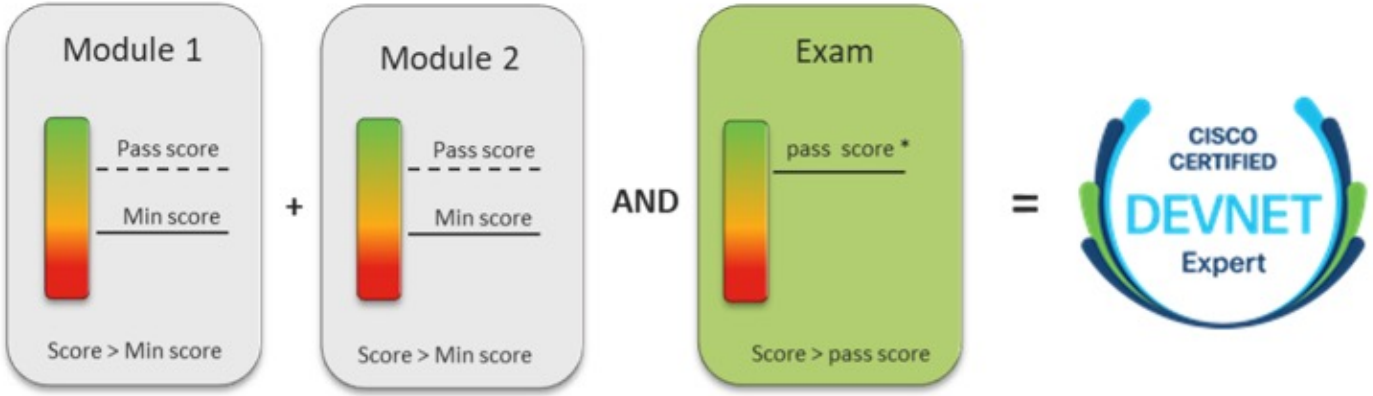
But what exactly do you DO?



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

DevNet Expert : How to pass?

To **pass** the exam, candidates need to score higher than the aggregated pass score of both modules and score higher than the minimum score set on each of the modules individually.



* Exam pass score = Pass score (Mod 1 + Mod 2)

Total points per module:

Mod 1: 37 points (3/8*100)
Mod 2: 63 points (5/8*100)

If candidates score less than the overall aggregated pass score of both modules OR score less than the minimum score set on 1 (or more) of the individual modules, they will fail the exam.

The pass score of each module is set based upon the difficulty level of the module itself.

Note: You can compensate for a (slightly) weaker performance in one module, by scoring higher than the pass score of the other exam module.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Design Module Question Details

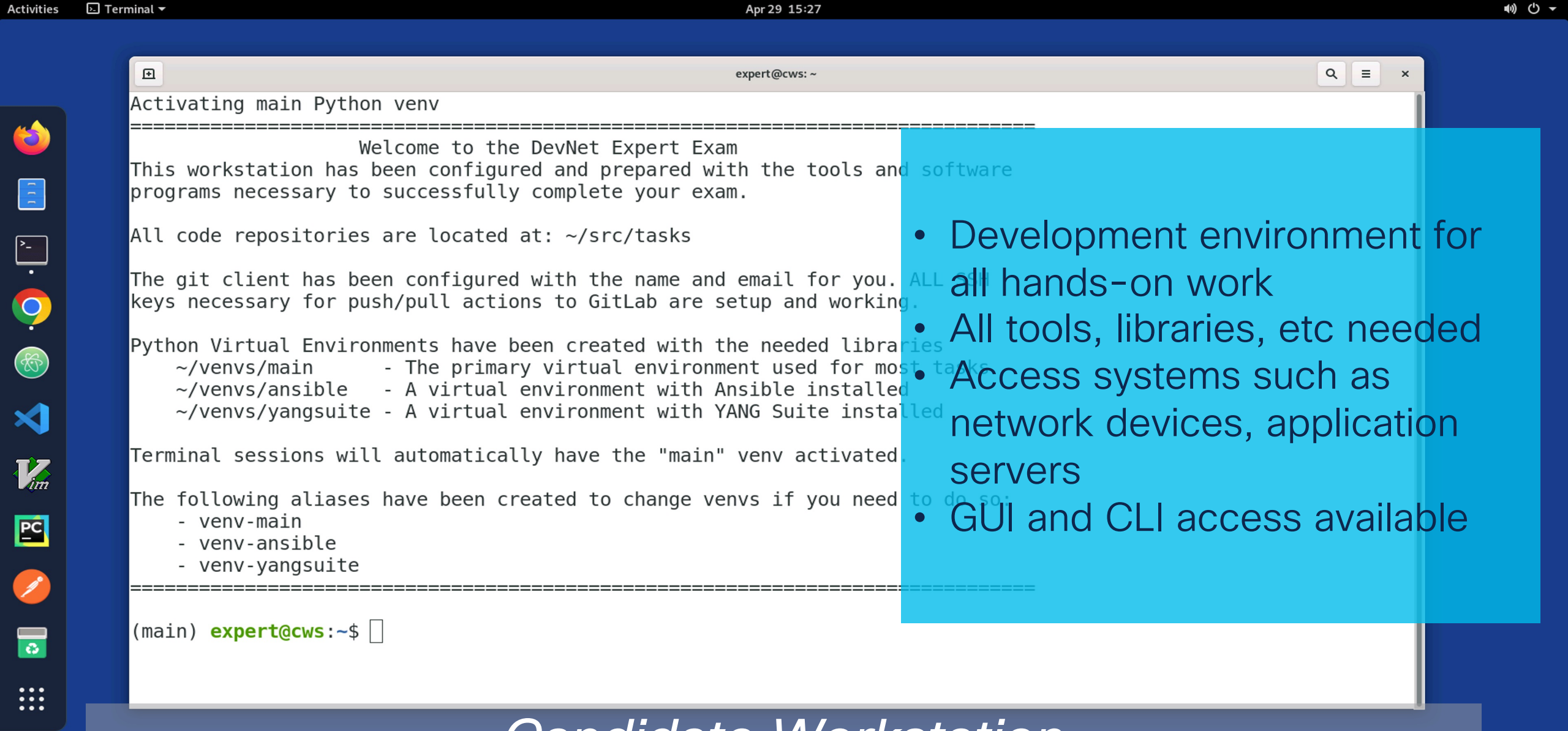
- Web based questions
  - Multiple Choice
  - Drag and Drop
  - Matrix
- Scenario based questions
  - Read, understand, and interpret presented details
  - Provide recommendations, finds problems, address conflicting requirements

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Hands On DTDM Module Question Details

- Create, complete, modify, or fix “code”
- Tasks include
  - Specific list of TODOs
  - Details on expected results when complete
  - Starting code / scripts
  - Details for network or relevant systems
- You must commit and push solutions with git

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public



Candidate Workstation

Download CWS from Cisco Learning Network

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Tools, Languages, and Platforms

- Python
- Ansible
- git
- GitLab
- Terraform
- Vault
- Telegraf, InfluxDB, Grafana
- Kubernetes
- Docker

Full list with versions available on Cisco Learning Network

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

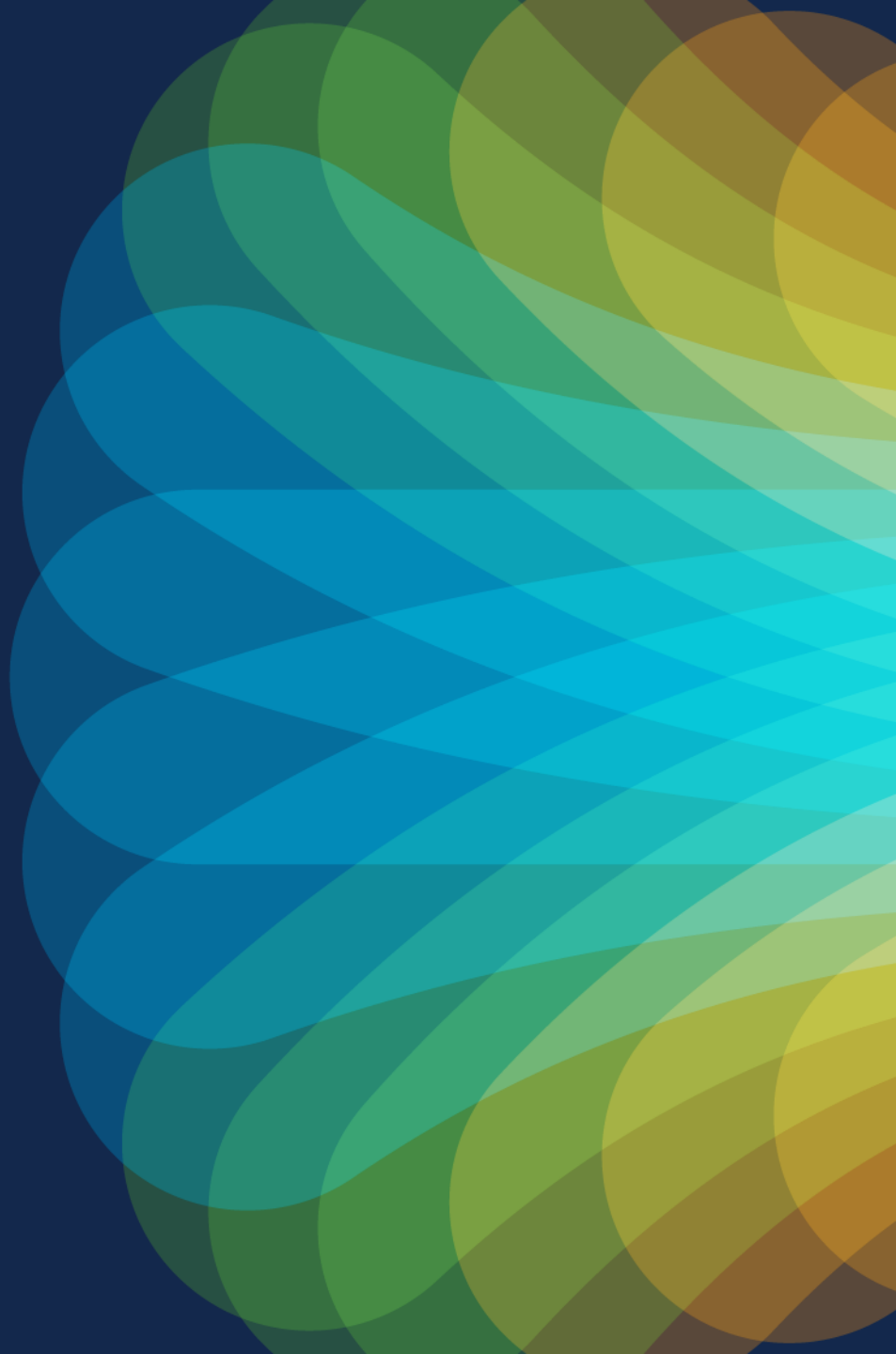
Cisco Platforms

- IOSv, IOSvL2
- Catalyst 8000v
- Nexus 9300v
- APIC Simulator
- Network Service Orchestrator (NSO)
- pyATS
- YANG Suite

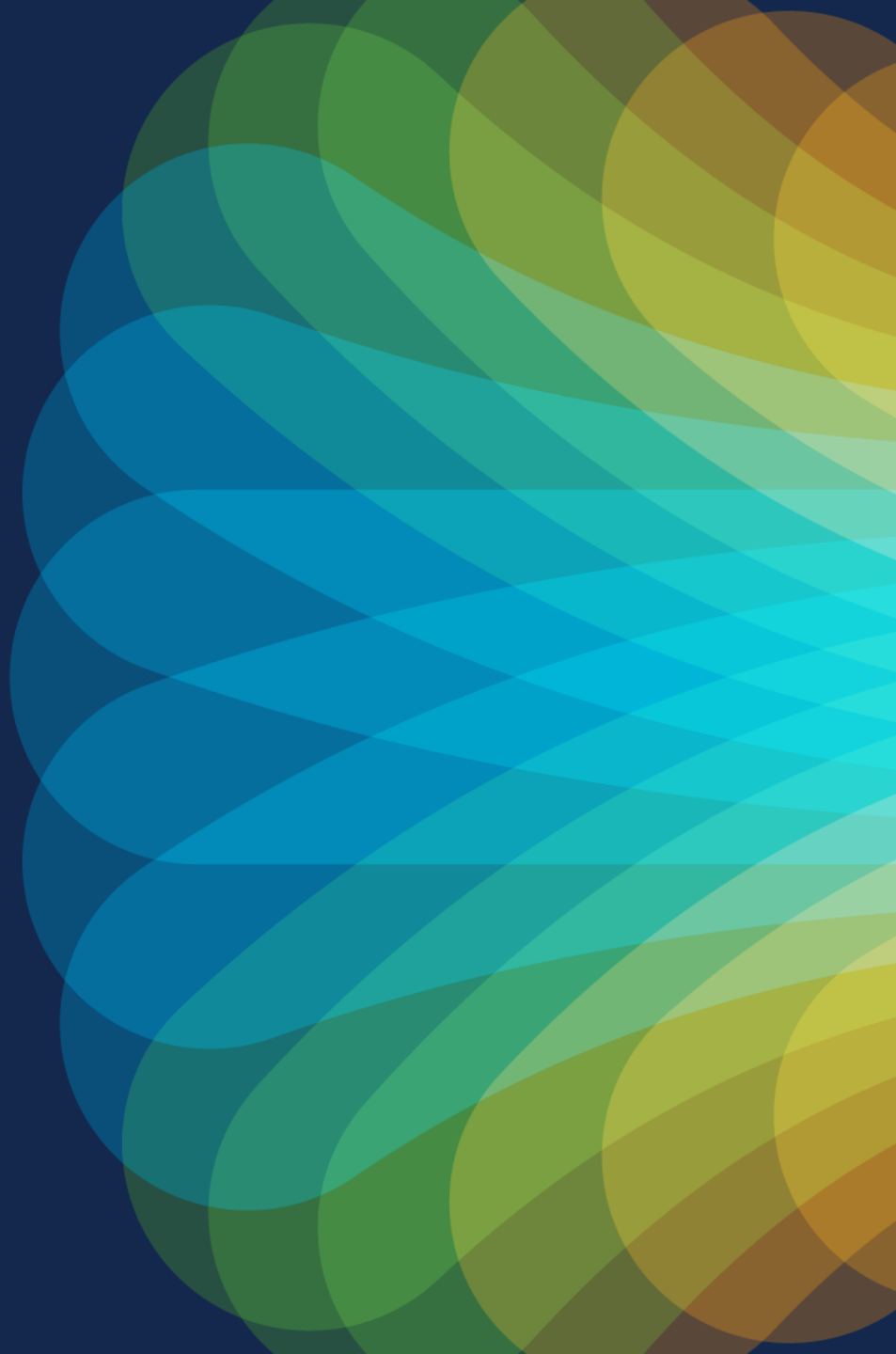
Full list with versions available on Cisco Learning Network

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

What kinds of questions will I see?



Design Question Walkthrough



Designing Solutions to Meet Business and Technical Requirements

Relevant Blueprint Items: 1.2, 2.1

Which approach do you suggest the automation team take to ease the management of access-control lists?

1. Create a Python tool using the Click library with commands to manage the ACLs
2. Share a Postman collection of RESTCONF API requests to manage the device ACLs
3. Build an Ansible role that can be used to manage the ACLs
4. Create an NSO service that manages the ACLs

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Supporting Materials

Email from Director of IT

Expert,
The new processes implemented by architecture and operations to make changes to the external access lists at the request of security have resulted in us missing our SLA with partners 4 times in the last week alone. Not to mention barely catching serious configuration errors at the last minute.

Please sync up with the Ops Lead and come up with a permanent solution to this issue.

~ Director of IT

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Chat with Ops Lead

Expert, it is not that bad. The process has only been in place for 3 weeks. My team is doing a great job.

I understand, but after 3 weeks you must have ideas for improving.

True. The problem is creating the correct configs. Too much variety between requests.

What if we could standardize the workflow? Common inputs, fill in the blanks?

That would actually help a lot. The tools your team has given us to gather network state have been so helpful.

Email from Network Architect

Expert,
Per the standup call this morning, here are the details on the network devices that are in scope for the new access-list policies and solutions.

Most of our locations have a standard deployment of ASR routers running IOS XE 17.2 code. However, this new standard is still being deployed. There is still 15% of the locations to be updated. Those sites have a variety of router platforms. Multiple vendors, and some that are 10 years old or more. We won't complete the refresh for 12 months.

And I confirmed that NETCONF/RESTCONF is now standard in all new deployments.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Automation Solutions Best Practices

- Don’t overengineer a solution, keep it simple
- Focus on business problems, not technical ones
- Build for 95% of the cases. Don’t double work for exceptions.
- Don’t automate halfway. If the desired change can’t be fully completed, do nothing
- Prefer modern interfaces and approaches, avoid CLI solutions whenever possible.
- Provide user of our tools consistent and simple interfaces

Chat with Service Desk Developer

We have just finished development and testing of new workflows to approve the ACL requests in the service desk system.

That’s great news. So once a request is approved, what happens?

It is routed to the operations team to execute the work.

Does the user request include all the source & destination details?

Yes… And the workflow could continue after approval. But the Ops Lead told us each request is too unique.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

What do we need to do?

- Define the ***actual*** problem to solve and the relevant users
- Use the supporting documentation to find the best option for this case study
- Find reasons to ***NOT*** pick each answer
- Find reasons to ***CHOOSE*** one answer

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

What do we need to avoid?

- Bringing our personal opinions into the answer
- Jumping to conclusions
- Focusing on a solution that is NOT an option

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Designing Solutions to Meet Business and Technical Requirements

Relevant Blueprint Items: 1.2, 2.1

Which approach do you suggest the automation team take to deliver ease the management of access-control lists?

1. Create a Python tool using the Click library with commands to manage the ACLs
2. Share a Postman collection of REST API requests to manage the device ACLs
3. Build an Ansible role that can be used to manage the ACLs
4. Create an NSO service that manages the ACLs

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Designing Solutions to Meet Business and Technical Requirements

Relevant Blueprint Items: 1.2, 2.1

Which approach do you suggest the automation team take to deliver ease the management of access-control lists?

1. Create a Python tool using the Click library with commands to manage the ACLs

- Great solution for Operations Users.
- Not ideal for the Service Desk Developers

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Designing Solutions to Meet Business and Technical Requirements

Relevant Blueprint Items: 1.2, 2.1

Which approach do you suggest the automation team take to deliver ease the management of access-control lists?

2. Share a Postman collection of REST API requests to manage the device ACLs

- Not all the devices support REST APIs
- Unique REST APIs per device does not provide "consistent interface to users"

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Designing Solutions to Meet Business and Technical Requirements

Relevant Blueprint Items: 1.2, 2.1

Which approach do you suggest the automation team take to deliver ease the management of access-control lists?

3. Build an Ansible role that can be used to manage the ACLs

A good option, however...

- "Don't automate halfway.
- If the desired change can't be fully completed, do nothing"

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Designing Solutions to Meet Business and Technical Requirements

Relevant Blueprint Items: 1.2, 2.1

Which approach do you suggest the automation team take to deliver ease the management of access-control lists?

4. Create an NSO service that manages the ACLs

- Modern tool, provide a REST API to Service Desk Team
  - Support for both old and new network devices
- No need to change northbound interface as devices change

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Designing Solutions to Meet Business and Technical Requirements

Relevant Blueprint Items: 1.2, 2.1

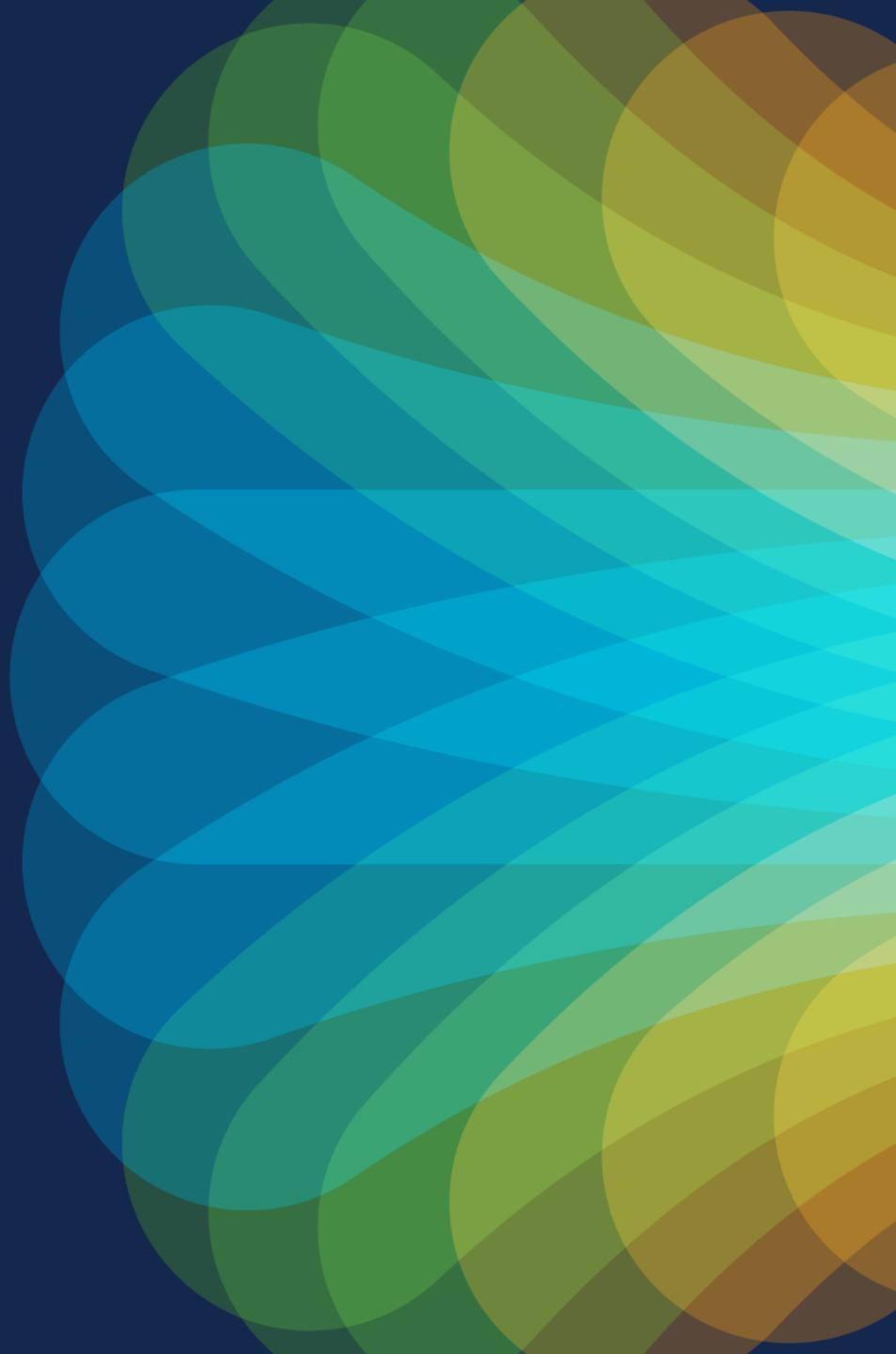
Which approach do you suggest the automation team take to deliver ease the management of access-control lists?

1. Create a Python tool using the Click library with commands to manage the ACLs
2. Share a Postman collection of REST API requests to manage the device ACLs
3. Build an Ansible role that can be used to manage the ACLs
4. Create an NSO service that manages the ACLs

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Some fun with Trivia Time!!!

DTDM Question Walkthrough



Working with Model Driven APIs
Relevant Blueprint Items: 2.5, 2.7, 3.2

A network service is being created to manage access control lists on CPE routers at branch locations using NSO. CRUD requests of this service will come from multiple systems including web applications, service catalogs, and orchestrators.

You will be completing example code for integrating with this service.

Context

Given the YANG model for this new service you must:

- Determine the RESTCONF resource path for creating a new rule on an existing access-list
- Evaluate the results of the creation API call to validate success and indicate error causes

Instructions

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Working with Model Driven APIs

Here are details for the NSO service and API endpoint.

```
admin@ncs# show running-config \
  access-list full-rule-set rule host-to-host

access-list full-rule-set
 rule host-to-host
  description "Allow inbound smtp access"
  action      permit
  protocol    tcp
  source address 192.168.1.1
  destination address 192.168.10.11
  destination port 25
  log
```

Resources

```
module: acl-service
  +--rw access-list* [name]
     +--rw name              string
     +--rw device*           -> /ncs:devices/device/name
     +--rw default
     |  +--rw rule?          enumeration
     +--rw rule* [name]
        +--rw name           string
        +--rw description?   string
        +--rw action         enumeration
        +--rw protocol       enumeration
        +--rw source
        |  +--rw address     union
        |  +--rw port?       uint16
        +--rw destination
        |  +--rw address     union
        |  +--rw port        uint16
        +--rw log?           empty
```

```
curl -u admin:admin http://0.0.0.0:8080/.well-known/host-meta

<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
    <Link rel='restconf' href='/restconf'/>
</XRD>
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Working with Model Driven APIs

Constraints

Keep the following in mind while completing your solution

- The access-list service is created and functional. No changes to the service can be made.
- Limit changes to the Python file “add_rule.py” and the specific TODO indications. Do not make changes outside of the functions with TODOs.
- Do not add, remove, or change any ”raise Error” statements in the code.
- Do not add “try/except” blocks.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Working with Model Driven APIs

The following are some examples of how the script should function. Validation

- Successfully adding a new rule to the access-list (no output)

```
./add_rule.py \
  full-rule-set newrule deny tcp 0.0.0.0/0 192.168.1.10/32 80 \
  --nso-username admin --nso-password admin \
  --nso-server http://0.0.0.0:8080
```

- Attempting to add a rule that already exists (same command)

```
ValueError: rule newrule already exists on access-list full-rule-set
```

- Attempting to add a rule to an access-list that does not exist

```
./add_rule.py \
  invalid-acl newrule deny tcp 0.0.0.0/0 192.168.1.10/32 80 \
  --nso-username admin --nso-password admin \
  --nso-server http://0.0.0.0:8080
ValueError: access-list invalid-acl not found.
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Working with Model Driven APIs

Starting Code – Part 1

```
def add_rule(server: str, username: str, password: str, args: argparse.Namespace) -> bool:
    """Add a new rule to an ACL Service on an NSO Server"""

    test_nso(server, username, password)

    # TODO: Set the value of URL to the correct path to add a new rule to an ACL
    # Notes:
    #   - the variable `server` is equal to the NSO address including the protocol and port
    #     for example: http://127.0.0.1:8080
    #   - The following arguments are available from the `args` parameter
    #     args.acl > access-list name
    #     args.rule > rule name
    #     args.action > permit/deny value
    #     args.protocol > tcp/udp
    #     args.source > source address as ipaddress.IPv4Address or IPv4.Network object
    #     args.destination > destination address as ipaddress.IPv4Address or IPv4.Network object
    #     args.sport / args.dport > source/destionation ports
    #     args.description > optional description for rule entry
    #     args.log > true/false on whether to log hits to rule
    url = f""
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Working with Model Driven APIs

Starting Code – Part 2

```
def add_rule(server: str, username: str, password: str, args: argparse.Namespace) -> bool:
    """Add a new rule to an ACL Service on an NSO Server"""
    # Code emitted

    response = requests.post(url, json=body, headers=headers, auth=(username, password))

    # Check response status code for expected values for conditions
    # TODO: Leverage the response status code to determine results based on comments for each test
    # 1. A successful creation of new rule
    if True == True:
        return True
    # 2. An error due to the rule already existing
    elif True == True:
        raise ValueError(f"rule {args.rule} already exists on access-list {args.acl}")
    # 3. An error due to the named access-list not existing
    elif True == True:
        raise ValueError(f"access-list {args.acl} not found.")
    # 4. An error due to the access-list service not being available in NSO
    elif True == True:
        raise ValueError(f"The acl-service:access-list not found on the NSO server.")
    else:
        return False
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Solution

Working with Model Driven APIs

Solution Code – Part 1

```
def add_rule(server: str, username: str, password: str, args: argparse.Namespace) -> bool:
    """Add a new rule to an ACL Service on an NSO Server"""

    test_nso(server, username, password)

    # TODO: Set the value of URL to the correct path to add a new rule to an ACL
    # Notes:
    #   - the variable `server` is equal to the NSO address including the protocol and port
    #     for example: http://127.0.0.1:8080
    #   - The following arguments are available from the `args` parameter
    #     args.acl > access-list name
    #     args.rule > rule name
    #     args.action > permit/deny value
    #     args.protocol > tcp/udp
    #     args.source > source address as ipaddress.IPv4Address or IPv4.Network object
    #     args.destination > destination address as ipaddress.IPv4Address or IPv4.Network object
    #     args.sport / args.dport > source/destionation ports
    #     args.description > optional description for rule entry
    #     args.log > true/false on whether to log hits to rule

    url = f"{server}/restconf/data/acl-service:access-list={args.acl}"
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Working with Model Driven APIs

Solution Code – Part 2

```
def add_rule(server: str, username: str, password: str, args: argparse.Namespace) -> bool:
    """Add a new rule to an ACL Service on an NSO Server"""

    # Check response status code for expected values for conditions
    # TODO: Leverage the response status code to determine results based on comments for each test

    # 1. A successful creation of new rule
    if response.status_code == 201:
        return True

    # 2. An error due to the rule already existing
    elif response.status_code == 409: # 409 Conflict
        raise ValueError(f"rule {args.rule} already exists on access-list {args.acl}")

    # 3. An error due to the named access-list not existing
    elif response.status_code == 400: # 400 Bad Request
        raise ValueError(f"access-list {args.acl} not found.")

    # 4. An error due to the access-list service not being available in NSO
    elif response.status_code == 404: # 404 Not Found
        raise ValueError(f"The acl-service:access-list not found on the NSO server.")
    else:
        return False
```

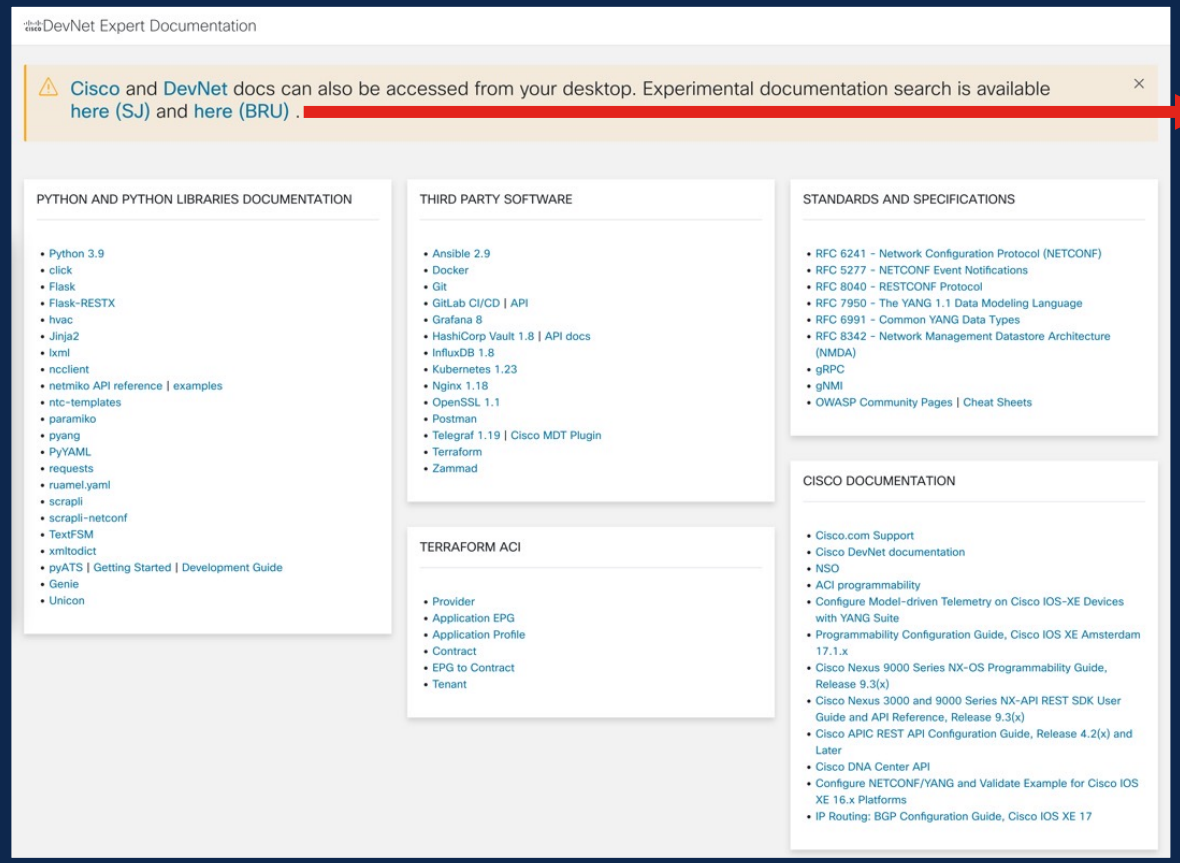

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

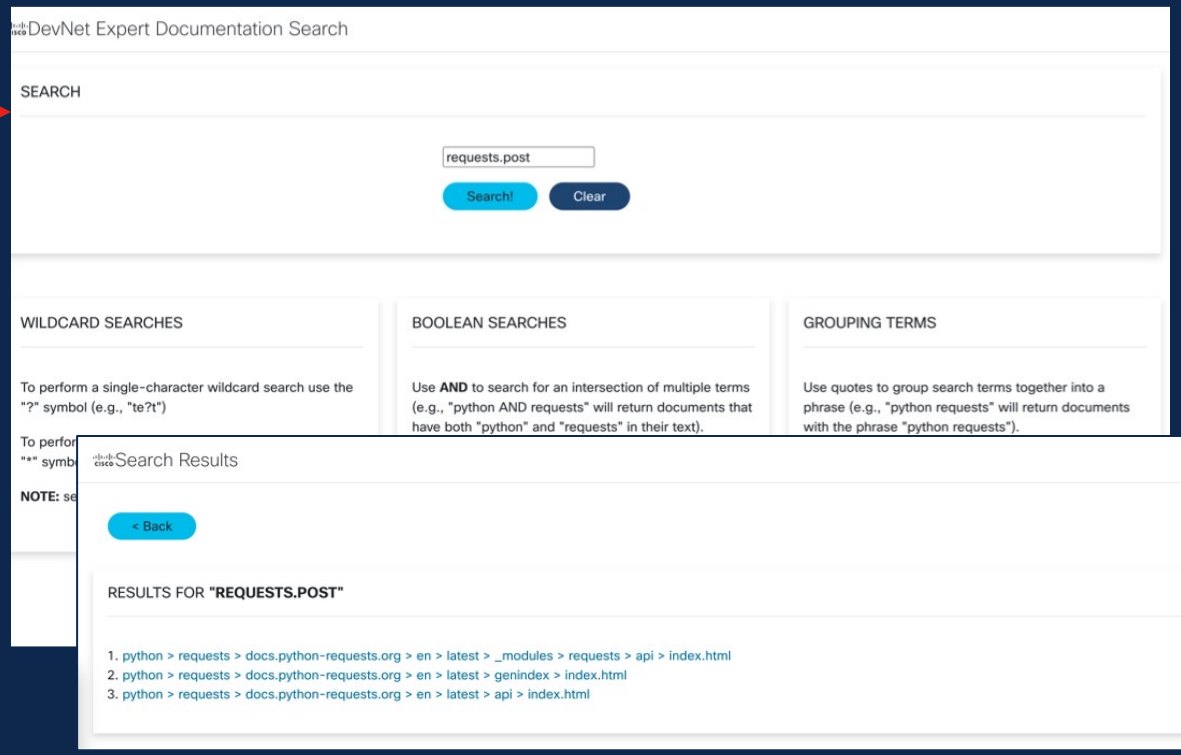
Common Questions about the Lab Exam

- What documentation is available?
- Will I need to write code from scratch?
- How is the exam graded?
- What libraries can I use?
- Can I pick what library to use?
- What does it take to pass?

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

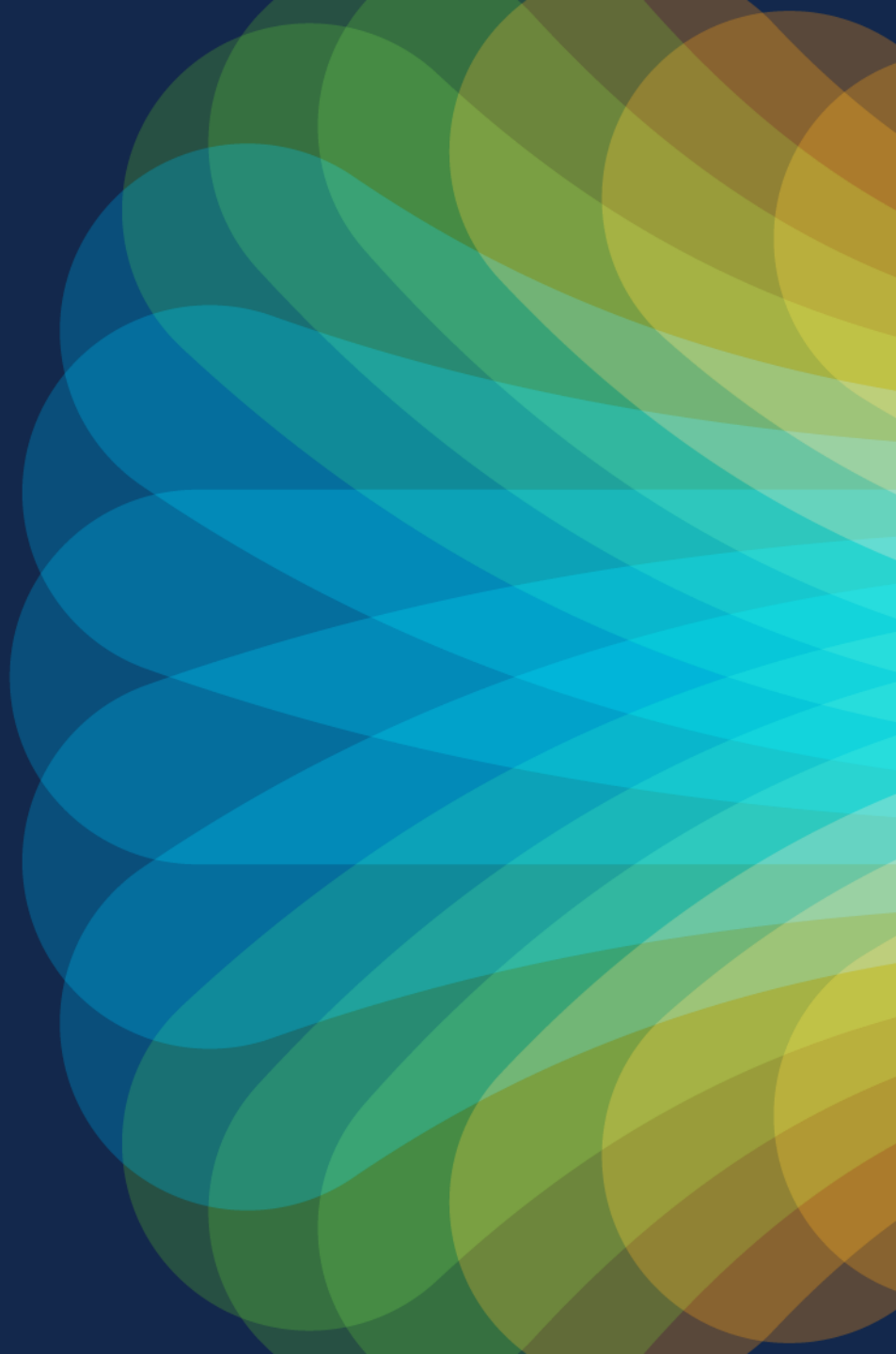
Documentation and Search Engine





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

What questions do YOU have?



Hank Preston

- Resources
  - Exam Topics List
  - Software and Equipment List (including CWS)
  - DevNet Expert Prep Webinar Series
  - Videos/Podcasts
    - Cisco Champion's Interview Hank
    - Q/A with Eric Chou and Hank
    - Snack Minute with Joe Clarke
- Stay in touch:
  - Webex/Email: hapresto@cisco.com
  - Blogs: https://blogs.cisco.com/author/hankpreston
    - Hank's Reflections on the Journey to DevNet Expert

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Resources

DevNet Expert 5 Prep Tips:

- Join your community, Cisco Learning Network.
- Find free Expert Prep self-study resources.
- Build your study plan with exam topics.
- Find the right training for you and your organization.
- Follow us @CiscoUtube for tech videos and more.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

cisco Live!

Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.

Meet the Expert Opening Hours:

Tuesday	**3:00pm – 7:00pm**
Wednesday	**11:15am – 7:00pm**
Thursday	**9:30am – 4:00pm**
Friday	**10:30am – 1:30pm**

Session Surveys

We would love to know your feedback on this session!

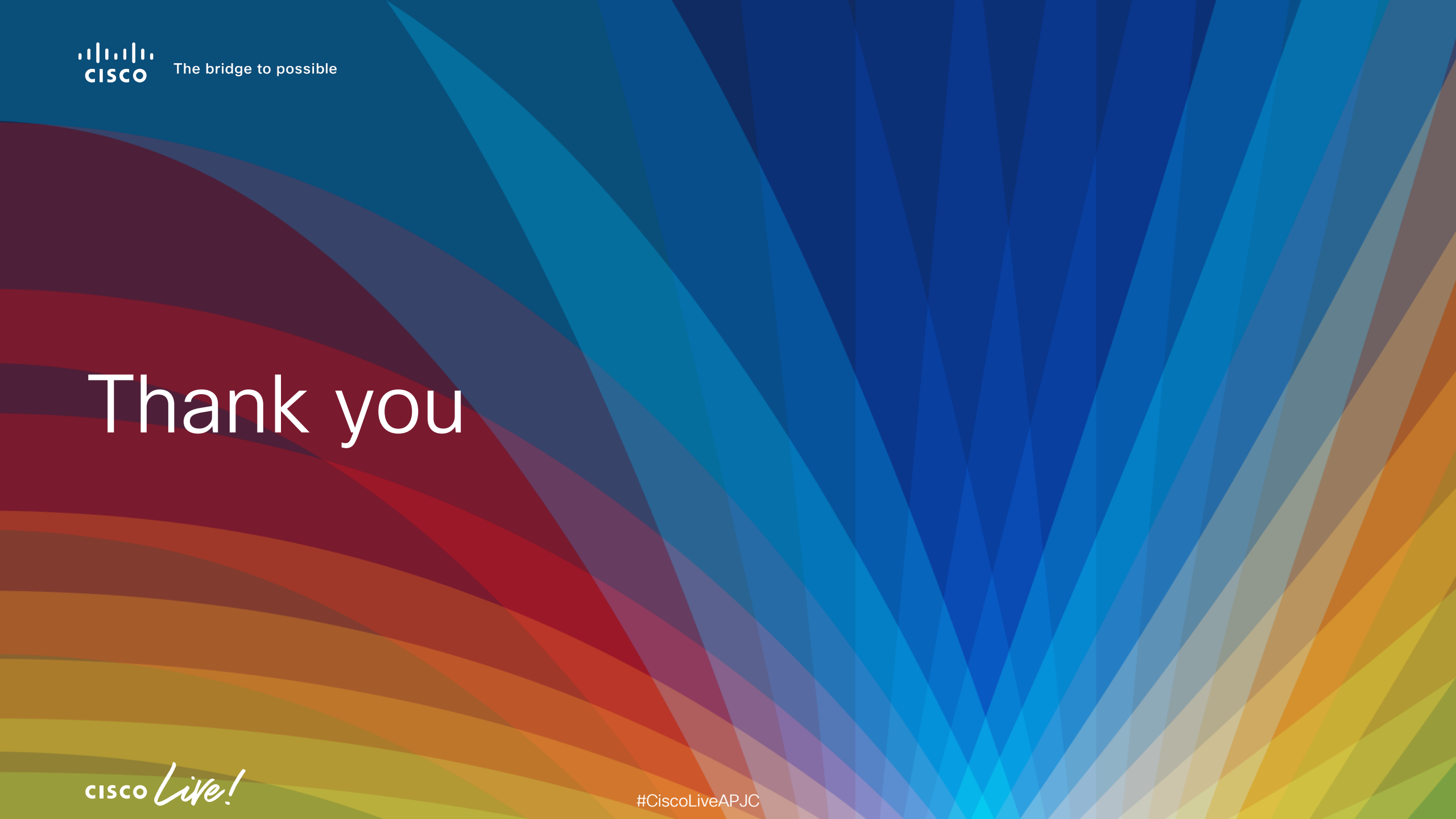
- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public


cisco
The bridge to possible
Thank you
cisco Live!
#CiscoLiveAPJC

Other Example Questions

Application design and architecture

Relevant Blueprint Items: 1.1, 1.2, 2.1, 4.x, 5.1, 5.2

Your team built a proof of concept application allowing users to update configuration of network connections in the lab area.

Context

As the senior engineer on the project you must provide design guidance in these areas as it moves into beta testing phase.

- While we expect the application to be popular, actual usage is unknown. What should be done to avoid overloading the app?
- How can users authenticate to the application securely to make requests?
- How should the user request services and network change services interact?

Questions

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Creating scripts and utilities for other teams

Relevant Blueprint Items: 1.3, 2.2, 3.2

The help desk team needs a tool to help them quickly gather information when troubleshooting network issues.

The output of the following commands need to be captured and formatted into a spreadsheet.

Context

Finish the script by:

- Adding arguments for device address and output file name
- Parsing returned command output for data
- Generate a CSV file with relevant data

Instructions

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Ensuring CICD pipelines operate as designed

Relevant Blueprint Items: 1.4, 1.5, 4.1

The CICD pipeline that builds, tests, and deploys the network service request application isn't working as required.

Context

Investigate the errors and ensure the following:

- The Docker images used to run the pipeline and build the images are correctly assigned and used
- Notification messages sent when errors occur include a link to the job
- Tests run on development branches leverage the correct infrastructure environment

Instructions

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Integrating Network Source of Truth into Change Control

Relevant Blueprint Items: 2.4, 2.3, 5.5

After numerous human error issues with change control submissions involving devices and IP addressing, the decision has been made to integrate the systems together. The developers for the change control system have asked for the following information.

Context

A curl command for each of the following targeting the NetBox API:

- Generation of a new authentication token for a user
- A listing of all management switches currently in inventory at site “DC1” unassigned to any tenant
  - Management switches include `mgmt` in the device name

Constraints

- A list of all possible device interface type options

Instructions

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Ensuring the health of the network with testing

Relevant Blueprint Items: 3.1, 3.4

The operations team is tired of being surprised by memory issues and process crashes on network devices and have asked for a new pyATS test to monitor CPU and memory on IOS, IOS XE, and NX-OS routers and switches.

Context

Building on the provided code, you are to:

- Capture current memory and CPU status using provided commands
- Compare results to known good results, looking for significant changes
- Send Webex warning and error notifications based on provided thresholds.

Instructions

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Creating a service to manage Access Lists

Relevant Blueprint Items: 2.10

A network service is being created to manage access control lists on CPE routers at branch locations using NSO. CRUD requests of this service will come from multiple systems including web applications, service catalogs, and orchestrators.

Context

Given the below desired final configuration for an example service instance, you are to:

- Complete the XML configuration template that will be used by the service to render final device configurations
  - TODOs have been included in the XML template to align to requirements
- Update the YANG model for the service match the service definition resources and requirements below
  - TODOs have been included in YANG model to align to requirements
  - Be sure to compile and include the final YANG model in your solution
- Apply the XML configuration template in the Python script passing in the variables that are created from YANG data.
  - TODOs have been included in the Python file to align to requirements

Instructions

Several “PASS” and “FAIL” configurations are included for testing your solution.

Validation

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Creating a service to manage Access Lists

Constraints

The following requirements have been provided by the network architect:

- Every rule will include a `LABEL` and `DESCRIPTION` remark directly before the rule entry in the configured access-list
  - See below examples for formating details
- Variable names and values for the XML template have been defined in Python already. Use these names in the XML template.

The following must be enforced in the YANG model

- Access list and rule names cannot contain any white spaces.
  - The regex pattern `[\S]*` can be used to match valid names
- Valid actions for a rule are "deny" or "permit" only. Valid protocols for a rule are "tcp" or "udp" only.
- Source and Destination address leafs should support either IPv4 address or prefixes using the `ietf-inet-types`.
- Source ports can be any valid port number, however destination ports must be limited to well known ports (1 - 1024).
- The log attribute of a rule should be a leaf that is either present or not.
- The default rule action, rule description, source port, and log flag are the only optional leafs in the model

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Example Resulting Configuration

Resources

```
data +access-list web-servers {
  + device [ ios-1 ];
  + rule https-web1 {
  +  description "Allow inbound https access";
  +  action permit;
  +  protocol tcp;
  +  source {
  +   address 0.0.0.0/0;
  +  }
  +  destination {
  +   address 192.168.10.11;
  +   port 443;
  +  }
  + }
  +}
```

```
access-list web-servers
 device [ ios-1 ]
 rule https-web1
  description "Allow inbound https access"
  action      permit
  protocol    tcp
  source address 0.0.0.0/0
  destination address 192.168.10.11
  destination port 443
 !
!
```

```
devices {
      device ios-1 {
       config {
        ip {
         access-list {
          extended {
    +      ext-named-acl web-servers {
    +       ext-access-list-rule "10 remark Rule https-web1 in ACL Service web-servers";
    +       ext-access-list-rule "11 remark Description Allow inbound https access";
    +       ext-access-list-rule "12 permit tcp any host 192.168.10.11 eq 443";
    +       ext-access-list-rule "100 remark Default Rule in ACL Service web-servers";
    +       ext-access-list-rule "101 remark Description deny all";
    +       ext-access-list-rule "102 deny ip any any log";
    +      }
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public