DEVCOR 350-901 Study Sheet

The Cisco Certified DevNet Professional certification validates your skills in developing and maintaining applications built on Cisco platforms. This study sheet is designed to help DEVCOR 350-901 exam candidates review common exam topics as reflected on the official exam blueprint.

COMING SOON: **Cisco Certified DevNet Professional DEVCOR 350-901 Official Cert Guide** (ISBN: 9780137370443) by Stuart Clark, Hazim Dahir, and Jason Davis will be available in Spring 2022.

Sign up for Cisco Press Promotional Mailings and Special Offers to receive product updates and pre-order discounts at **ciscopress.com/newsletter**

Browse all Cisco Press self-study products at **ciscopress.com/store**

Git Reference

git config --global user.name "[first_name last_name]"	configure a name associated in version history/logs
git config --global user.email "[email_address]"	configure an email address associated in version history/logs
cd project_dir; git init	initialize an existing directory as a Git repository
git clone [url]	retrieve a whole repository from a hosted location/URL
git add [file]	add a file in its current state to the next commit
git status	show modified files in working directory, staged for next commit
git diff	examine differences of what is changed but not staged
git diff –staged	examine differences of what is staged but not yet committed
git commit -m "[commit message]"	commit staged content as a new commit instance
git branch	list branches with an asterisk (*) next to the current active branch
git branch [branch_name]	create a new branch at the current commit
git checkout	switch to another branch / check it out into your working directory
git merge [branch_name]	merge the specified branch's history into the current one
git log	show all commits in the current branch's history
git remote add [alias] [url]	add a git URL as an alias
git fetch [alias]	fetch (download) all branches from aliased Git remote
git merge [alias]/[branch]	merge an aliased Git remote (and branch) into the current branch bringing it up-to-date
git push [alias] [branch]	push (upload) local branch commits to the aliased Git remote repository branch
git pull	fetch (download) and merge any commits from the tracking remote branch
git show [SHA_reference]	show any object in Git in human-readable format

API Methods

HTTP Verb	Idempotent	Safe
GET	Yes	Yes
POST	No	No
PUT	Yes	No
PATCH	No	No
DELETE	Yes	No

API Specifications and Protocols

	REST	RPC	GraphOL	SOAP
Style	Architectural Style	RPC	Query Language	Protocol
Protocol	HTTP/1	HTTP/2	HTTP/1	HTTP/1
Format	XML, JSON, HTML, Plain Text	JSON, XML, Protobuf	JSON	XML
Security	TLS/ SSL / HTTPS	TLS / SSL	TLS / SSL	WS security / SSL
State	Stateless	Stateless	Stateless	Stateful/Stateless
ACID Compliant	No	No	No	Yes

RESTful and NETCONF APIs

	RESTful API	Netconf API
Transport	HTTP	SSHv2, SOAP, TLS
Message	HTTP/1.1	RCP, RCP REPLY
Content	JSON/XML	XML
Operations	HTTP GET, HTTP POST, HTTP PUT, HTTP DELETE	get-config, get, copy-config, lock, unlock, edit-config, delete-config, kill-session, close-session
State	Stateless	Stateful

HTTP Status Codes

Status Code	Reason Phrase	
1xx	Informational	Request received, continuing process
2xx	Success	The action was successfully received, understood, and accepted
3xx	Redirection	Further action must be taken in order to complete the request
4xx	Client Error	The request contains bad syntax or cannot be fulfilled
5xx	Server Error	The server failed to fulfill an apparently valid request

SNMP Versions and Capabilities

	Level	Auth	Encryption	What Happens	
SNMPv1	noAuthNoPriv	Community String		Uses a Community String Match for Authentication	👎
SNMPv2c	noAuthNoPriv	Community String		Uses a Community String Match for Authentication	👎
SNMPv3	noAuthNoPriv	Username		Uses a Username Match for Authentication	👎
SNMPv3	authNoPriv	MD5 or SHA		Provides Authentication Based on HMAC-MD5 or HMAC-SHA Algorithms	👍
SNMPv3	authPriv	MD5 or SHA	CBC-DES AES-128	DES 56-Bit Encryption RFC 3826 added AES-128	👍

128/192/256-Bit AES and 168-Bit 3DES Available in 12.4(2)T

Logical Plane Models

Model	Function	Example
Control Plane	Determines/calculates packet or frame forwarding decisions	Software processes, such as OSPF, EIGRP, BGP, IS-IS, LDP, ARP, MAC address table, etc.
Data Plane	Executes on packet or frame forwarding	Interfaces
Management Plane	Protocols/methods for provisioning the Control Plane	CLI/SSH, SNMP, NETCONF/RESTCONF

Operational Lifecycles

Operational Perspective	Function
Day-0	Initial Installation
Day-1	Configuration for production purpose
Day-2	Compliance & optimization
Day-X	Migration/Decommissioning

Software Development Methodologies and Frameworks

Method Name	Description
Agile	Flexible and incremental design process focused on collaboration
Kanban	Visual framework promoting what/when/how to develop in small, incremental changes; complements Agile
Lean	Create efficiencies and remove waste to produce more with less
Scrum	Fixed-length iterations (sprints); follows roles, responsibilities and meetings for well-defined structure; derivative of Agile
Waterfall	Sequential design process; fully planned; execution through phases

Software-Defined Protocols and Solutions

Protocol/Solution		Function
OpenFlow		Layer-2 programmable forwarding protocol and specification for switch manufacturing
I2RS	Interface to Routing System	Layer-3 programmable protocol to the routing information base (RIB); allowed manipulation and creation of new routing metrics
PCEP	Path Computation Element Protocol	L3 protocol capable of computing a network path or route based on a network graph and applying computational constraints
BGP-LS/FS	BGP Link-State / Flow Spec	Provides ability to gather IGP topology of the network and export to a central SDN Controller / alternative method to Remotely Triggered Black Hole filtering useful for DDoS mitigation
OpenStack		Hypervisor technology for virtualization of workloads
OMI	Open Management Infrastructure	Open-source Common Information Model with intent to normalize management
Puppet		Agent-based configuration management solution embedded in devices
Ansible		Agentless configuration management solution
NETCONF	Network Configuration standard	IETF working group specification normalizing configuration across vendors using XML schemas (later updated with YANG)
YANG	Yet Another Next Generation	Data modeling language for the definition of data sent over network representing configurations and/or services

NETCONF Protocol Operations

Operation	Description
get	Retrieve running configuration and device state information
get-config	Retrieve all or part of a specified configuration
edit-config	loads all or part of a specified configuration to the specified target configuration
copy-config	Create or replace an entire configuration datastore with the contents of another complete configuration datastore
delete-config	Delete a configuration datastore
lock	Lock an entire configuration datastore of a device
unlock	Release a configuration lock previously obtained with the <lock> operation
close-session	Request graceful termination of a NETCONF session
kill-session	Force the termination of a NETCONF session

IETF YANG Types Compared to SNMP/SMIv2 types

YANG Type	Equivalent SMIv2 type (module)
counter32	Counter32 (SNMPv2-SMI)
zero-based-counter32	ZeroBasedCounter32 (RMON2-MIB)
counter64	Counter64 (SNMPv2-SMI)
zero-based-counter64	ZeroBasedCounter64 (HCNUM-TC)
gauge32	Gauge32 (SNMPv2-SMI)
gauge64	CounterBasedGauge64 (HCNUM-TC)
object-identifier	-
object-identifier-128	OBJECT IDENTIFIER
yang-identifier	-
date-and-time	-
timeticks	TimeTicks (SNMPv2-SMI)
timestamp	TimeStamp (SNMPv2-TC)
phys-address	PhysAddress (SNMPv2-TC)
mac-address	MacAddress (SNMPv2-TC)
xpath1.0	-
hex-string	-
uuid	-
dotted-quad	-