

Cisco DevNet Professional (DEVCOR) Developing Applications using Cisco Core Platforms and APIs

-- Scheda Corso --

Pre-Requisiti

- Conoscenza dei concetti di base delle API
- Conoscenza dei processi per lo sviluppo di applicazioni
- Consigliata certificazione Cisco DevNet Associate (DEVASC)

A chi è rivolto

Questo corso è progettato per chiunque svolga o cerchi un ruolo di sviluppatore e ha uno o più anni di esperienza pratica nello sviluppo e nella manutenzione di applicazioni costruite su piattaforme Cisco. Questo corso tratta materiale specializzato sulla progettazione, lo sviluppo e il debug di applicazioni che utilizzano API e piattaforme Cisco e la gestione e la distribuzione di applicazioni sull'infrastruttura Cisco. Per beneficiare a pieno di questo corso, è preferibile avere dai tre ai cinque anni di esperienza nella progettazione e implementazione di applicazioni basate su piattaforme Cisco.

Moduli del programma

- 1) Sviluppo e progettazione software
- 2) Utilizzo delle API
- 3) Piattaforme Cisco
- 4) Distribuzione e sicurezza delle applicazioni
- 5) Infrastruttura e Automazione

Descrizione del programma

Il conseguimento della certificazione Cisco DevNet Professional certifica le abilità nello sviluppo e nella gestione di applicazioni basate su piattaforme Cisco. Per ottenere la certificazione DevNet Professional, si superano due esami: uno che copre lo sviluppo e la progettazione del software di base per le piattaforme Cisco e un esame di concentrazione degli sviluppatori di software a scelta, in modo da poter personalizzare la certificazione in base all'area di interesse.

Modulo 1		
20%		Sviluppo e progettazione software
	1.1	Descrivere le applicazioni distribuite e relativi concetti di front-end, back-end e bilanciamento del carico

	1.2	Valutare la progettazione di applicazioni considerando scalabilità e modularità
	1.3	Valutare la progettazione di un'applicazione considerando l'alta affidabilità e la resilienza (inclusi on-premise, ibrido e cloud)
	1.4	Valutare la progettazione di un'applicazione considerando la latenza e il rate-limiting
	1.5	Valutare la progettazione e l'implementazione di un'applicazione tenendo conto della manutenibilità
	1.6	Valutare la progettazione e l'implementazione di un'applicazione considerando l'osservabilità
	1.7	Diagnosticare i problemi di un'applicazione attraverso dei log correlati a un evento
	1.8	Valutare la scelta dei tipi di database in relazione ai requisiti dell'applicazione (ad esempio relazionale, documento, grafico, colonnare e Time Series)
	1.9	Spiegare i modelli architetturali (monolitici, orientati ai servizi, microservizi e basati su eventi)
	1.10	Utilizzare le operazioni avanzate di controllo della versione con Git 1.10.a Merge a branch 1.10.b Risolvere i conflitti 1.10.c git reset 1.10.d git checkout 1.10.e git revert
	1.11	Spiegare i concetti di creazione di pacchetti di rilascio e gestione delle dipendenze
	1.12	Costruire un diagramma di sequenza che include chiamate API

Modulo 2

20%		Utilizzo delle API
	2.1	Implementare una gestione affidabile degli errori REST API per i timeout e i rate-limit
	2.2	Implementare il flusso di controllo del codice consumer per gli errori irreversibili delle REST API
	2.3	Identificare i modi per ottimizzare l'utilizzo delle API tramite i controlli della cache
	2.4	Costruire un'applicazione che utilizza le REST API con supporto per la pagination
	2.5	Descrivere gli step per l'autorizzazione OAuth2 three-legged

Modulo 3

20%		Piattaforme Cisco
	3.1	Creare richieste API per implementare chatop con l'API Webex Teams

	3.2	Creare richieste API per creare ed eliminare oggetti utilizzando Firepower Device Management (FDM)
	3.3	Creare richieste API utilizzando la piattaforma Meraki per eseguire queste attività 3.3.a Utilizzare le API della dashboard Meraki per abilitare un SSID 3.3.b Utilizzare le location API di Meraki per recuperare i dati sulla posizione
	3.4	Creare chiamate API per recuperare i dati da Intersight
	3.5	Costruire uno script Python usando le API UCS per eseguire il provisioning di un nuovo server UCS in base a un template
	3.6	Costruire uno script Python utilizzando le API di Cisco DNA per recuperare e visualizzare le informazioni sulla rete wireless
	3.7	Descrivere le funzionalità di AppDynamics
	3.8	Descrivere i passaggi per creare una dashboard personalizzata per presentare i dati raccolti dalle API Cisco

Modulo 4		
20%		Distribuzione e sicurezza delle applicazioni
	4.1	Diagnosticare un errore della pipeline CI/CD (ad esempio dipendenza mancante, versioni incompatibili dei componenti e test non riusciti)
	4.2	Integrare un'applicazione in un ambiente CD predefinito sfruttando Docker e Kubernetes
	4.3	Descrivere i vantaggi dei test continui e dell'analisi statica del codice in una pipeline CI
	4.4	Usare Docker per creare containers
	4.5	Descrivere i principi delle applicazioni a 12 fattori
	4.6	Descrivere una strategia di loggign efficace per un'applicazione
	4.7	Spiegare i problemi relativi alla privacy dei dati per l'archiviazione e la trasmissione dei dati
	4.8	Identificare l'approccio di archiviazione segreta pertinente a un determinato scenario
	4.9	Configurare i certificati SSL specifici dell'applicazione
	4.10	Implementare strategie di mitigazione per le minacce OWASP (ad esempio XSS, CSRF e SQL injection)
	4.11	Descrivere in che modo i principi di crittografia end-to-end si applicano alle API

Modulo 5		
20%		Infrastruttura e Automazione
	5.1	Spiegare le considerazioni sul model driven programmability (incluso il consumo e l'archiviazione dei dati)
	5.2	Utilizzare RESTCONF per configurare un dispositivo di rete, incluse interfacce, rotte statiche e VLAN (solo XE IOS)
	5.3	Creare un flusso di lavoro per configurare i parametri di rete con: 5.3.a Playbook Ansible 5.3.b Manifesti Puppet
	5.4	Identificare una soluzione di gestione della configurazione per soddisfare i requisiti tecnici e aziendali
	5.5	Descrivere come ospitare un'applicazione su un dispositivo di rete (inclusi i dispositivi abilitati come Catalyst 9000 e Cisco IOx)

Laboratori

Si farà un uso intensivo delle risorse messe a disposizione dalla nostra Academy e delle Sandbox messe a disposizione da Cisco. Il POD sarà attivo e accessibile per tutta la durata del corso, le configurazioni verranno riutilizzate e aggiornate in ogni modulo.

Un POD studente è composto da:

2 x CSR1000v
2 x Nexus Switches NX-OS
1 x ASAv

Un Server Linux con:

Ubuntu 18.04 Server con GUI (gnome)

Lo studente durante il corso installerà i seguenti software:

- Python 3
- Python 2.7
- Git CLI
- Visual Studio
- Wireshark
- Postman
- Ansible