سرفصل آموزشی

پک پیشرفته متخصص Automation

فهرست سرفصل‌های دوره‌های آموزشی

سرفصل‌های دوره آموزشی

Ansible

Introduction of DevOps

Understanding DevOps concepts

DevOps Automation

Continuous Integration

Continues Delivery

Continuous Deployment

The roles of Ansible in CI/CD

The benefit of CICD

What is Ansible?

Automation Deployment Pipeline

Need of Ansible

What Ansible can do?

Advantages of using Ansible?

Agent-Based VS Agentless systems

Ansible’s Agentless Architecture

Install Ansible

Validate Ansible Installation

Ansible Vs Puppet Vs Chef Vs SaltStack

Ansible Architecture

Host, Group and Host Inventory

Ansible Ad-Hoc commands

Playbooks, plays, tasks and modules

Ansible configuration

Ansible-playbook Structure

Taks, vars, files, templates, meta, defaults, handlers

Ansible-playbook Syntax

Run ansible playbook

Variables, variable types and priorities

Command, expect, script, shell and raw modules

file, copy and fetch modules

Group and user modules

zyper_repository, zypper, yum_repository and you modules

Template, lineinfile, replace and service module

Archive and unarchive module

Async actions and concurrent tasks

wait_for and wait_for_connection modules

Mail module

Subversion and git modules

get_url, timezone and iptables modules

Mariadb modules

Find module and local_action feature

Conditions

Loops

Standard loops

Nested loops

Import playbooks and tasks

Handlers

Ansible Vault

Encrypt files and strings

Vault ID

Implement an Ansible playBook to Setup a webserver

Integrate Jenkins & Ansible

CICD with Git, Jenkins and Ansible (Application Deployment)

Ansible & VMWare

Ansible & Cisco

Ansible & Mikrotik

Develop Custom Module

Module format

Module’s return value and error handling

Setup nginx servers behind haproxy via Ansible playBook

Ansible & Windows Hosts

Manage windows features

Manage windows services

Execute shell module on windows

Windows Package management

Package Silent Installation

Implement an Ansible PlayBook to Setup IIS

Integrate Ansible and Docker

Docker_image and docker_image modules

docker_container and docker_container modules

docker_network and docker_network_info modules

docker_volume and docker_volume_info modules

docker_swarm module

Ansible Galaxy

Ansible Tower

Ansible AWX

AWX prerequisites and Installation

AWX Dashboard

AWX - organizations, teams and users

AWX - hosts, groups and inventory

AWX - credentials

AWX - projects and templates

AWX - Schedule templates, notification and permissions

سرفصل‌های دوره آموزشی

Python

Introduction:

- Python History
- Python Features & usage
- Python versions & differences
- Interactive Environment and Interpreter of Python
- Python IDEs and PyCharm
- Running Python files from Terminal & IDE by example
- PyCharm Environment & Debuging with PyCharm

Python program structure:

- Storing code and running program
- Variables and Datatypes
- Naming rules & conventions
- Getting user input and displaying output to terminal
- Introduction to Object Oriented programming & Objects in Python
- Modularity & Python libraries
- Installing & using libraries in python

Computational Operators:

- Logical Operators
- Operator priority
- Exceptions & Exception Handling in Python
- Basic DataTypes & Literals
- Lists, Tuples, Sets, Sequences and dictionaries
- List & Tuple Methods Slicing And Concatenation of Sequences
- Dictionary methods
- List Comprehensions

Strings and coding:

- Unicode
- Excape Characters
- Multiline Strings
- Type casting in python
- String Methods
- String formatting

Conditional statements:

- Loops
- For loop
- While loop
- Loop controlling statements
- Nested Loops
- Using loops on Sequences & dictionaries

Working with files:

- Binary & Text files
- File Opening modes
- Working with file offset pointer
- Bytes and bytearrays
- With statement
- Working with csv files

Functions in python:

- Function definition structure
- Documenting objects in python and self documented concept
- Calling functions variable scope in functions (global, local , nonlocal)
- Optional function parameters
- Lambda functions

Libraries in python:

- Library Structure & Creating Libraries
- Separating program logic from helping entities
- Frequent libraries & their usage
- Sys library
- Getting script parameter from terminal
- Os library
- Working with OS directory structures using os library

Regular Expressions:

- Re library
- Urllib & request libraries
- Web scraping using urllib, request & re libraries

Class definition:

- Class initiation
- Inheritance
- Class methods and variables
- Example of using classes versus functional programming

سرفصل‌های دوره آموزشی

Terraform

Understand infrastructure as code (IaC) concepts:

- Explain what IaC is
- Describe advantages of IaC patterns

Understand Terraform’s purpose (vs other IaC):

- Explain multi-cloud and provider-agnostic benefits
- Explain the benefits of state

Understand Terraform basics:

- Handle Terraform and provider installation and versioning
- Describe plugin based architecture
- Demonstrate using multiple providers
- Describe how Terraform finds and fetches providers
- Explain when to use and not use provisioners and when to use local-exec or remote-exec

Use the Terraform CLI (outside of core workflow):

- Terraform format code
- Taint Terraform resources
- Import and reuse of terraform modules
- Terraform workspaces
- Terraform states
- Terraform verbose logging

Interact with Terraform modules:

- Contrast module source options
- Interact with module inputs and outputs
- Describe variable scope within modules/child modules
- Discover modules from the public Terraform Module Registry
- Defining module version

Navigate Terraform workflow:

- Describe Terraform workflow
- Initialize a Terraform working directory
- Validate a Terraform configuration
- Generate and review an execution plan for Terraform
- Execute changes to infrastructure with Terraform
- Destroy Terraform managed infrastructure

Implement and maintain state:

- Describe default local backend
- Outline state locking
- Handle backend authentication methods
- Describe remote state storage mechanisms and supported standard backends
- Describe effect of Terraform refresh on state
- Describe backend block in configuration and best practices for partial
- Configurations
- Understand secret management in state files

Read, generate, and modify configuration:

- Demonstrate use of variables and outputs
- Describe secure secret injection best practice
- Understand the use of collection and structural types
- Create and differentiate resource and data configuration
- Use resource addressing and resource parameters to connect resources together
- Use Terraform built-in functions to write configuration
- Configure resource using a dynamic block
- Describe built-in dependency management

Understand Terraform Cloud and Enterprise capabilities:

- Describe the benefits of Sentinel, registry, and workspaces
- Differentiate OSS and TFE workspaces
- Summarize features of Terraform Cloud