Module 1: Introduce Ansible

1. **Overview of Ansible**
    - Introduction to Ansible
    - Benefits of using Ansible for automation
    - Differences between Ansible and other automation tools
2. **Ansible Architecture**
    - Master and managed node relationship
    - Ansible control node and managed node setup
3. **Core Components of Ansible**
    - Inventory, modules, tasks, and playbooks
    - Understanding how these components interact
4. **Installation of Ansible**
    - Installing Ansible on a control node
    - Installation dependencies for Red Hat Ansible Automation Platform
5. **Introduction to Red Hat Ansible Automation Platform**
    - What is Red Hat Ansible Automation Platform?
    - Overview of its capabilities and use cases
6. **How Ansible Executes Commands**
    - Connection types: SSH, WinRM
    - Command execution workflow
7. **Using Ansible Galaxy**
    - What is Ansible Galaxy?
    - Finding, using, and contributing to Ansible roles from Galaxy
8. **Security Best Practices in Ansible**
    - Authentication methods (SSH keys, passwordless authentication)
    - Ansible Vault for securing sensitive data

Module 2: Implement an Ansible Playbook

1. **Understanding Ansible Playbooks**

- Introduction to playbooks and their syntax
- Structure of a basic playbook

2. **Creating an Inventory of Managed Hosts**
    - Static vs dynamic inventories
    - Format of an inventory file (INI, YAML)
3. **Writing Simple Ansible Playbooks**
    - Defining hosts and tasks
    - Running commands and managing services
4. **Variables in Playbooks**
    - Using variables to customize playbooks
    - Defining variables in playbooks and inventories
5. **Using Handlers in Playbooks**
    - What are handlers and when to use them
    - Defining handlers for specific tasks
6. **Execution Flow in Ansible Playbooks**
    - Task ordering and dependencies
    - Controlling playbook execution flow
7. **Using Loops and Conditionals**
    - Implementing loops in playbooks
    - Writing conditional tasks for specific situations
8. **Running and Debugging Playbooks**
    - Running playbooks with the ansible-playbook command
    - Debugging and troubleshooting playbook execution

Module 3: Manage Variables and Facts

1. **Introduction to Variables**
    - Defining and using variables in Ansible
    - Variable types: strings, integers, lists, and dictionaries
2. **Variable Precedence**

- Understanding variable precedence
- Where and how to define variables in Ansible

3. **Using Facts in Ansible**
    - What are facts and how are they gathered?
    - Using the setup module to gather facts
4. **Managing Variable Scope**
    - Global vs local variables
    - Setting variables per host, group, or playbook
5. **Facts and Conditional Logic**
    - Using facts to drive task execution
    - Writing conditional tasks based on gathered facts
6. **Using Ansible Vault to Manage Secrets**
    - Introduction to Ansible Vault for sensitive data
    - Encrypting and decrypting variables with Vault
7. **Best Practices for Managing Variables**
    - Organizing variables across playbooks
    - Keeping variables secure and maintainable
8. **Debugging Variables and Facts**
    - Using the debug module for troubleshooting
    - Viewing gathered facts during playbook runs

Module 4: Implement Task Control

1. **Task Execution Control**
    - Controlling task execution flow using when, until, and retries
    - Task skipping and conditional execution
2. **Managing Handlers**
    - Defining and using handlers for task events
    - Best practices for efficient handler use
3. **Understanding Task Failures**

- Handling task failures with the failed_when condition
- Using the ignore_errors directive

4. **Task Tags and Skipping Tasks**
    - Using tags to selectively run tasks
    - Skipping tasks with specific tags during execution
5. **Looping through Tasks**
    - Looping through a list of items with with_items
    - Using loop for more advanced looping techniques
6. **Task Delegation and Asynchronous Tasks**
    - Delegating tasks to other hosts
    - Running tasks asynchronously
7. **Task Notifications**
    - Sending notifications upon task completion
    - Integrating with external systems for task notifications
8. **Testing and Debugging Task Control**
    - Using verbose output for troubleshooting
    - Validating playbook task flow

Module 5: Deploy Files to Managed Hosts

1. **Understanding File Deployment in Ansible**
    - Using the copy, template, and fetch modules
    - Deploying configuration files to remote hosts
2. **Managing Files with the copy Module**
    - Copying local files to remote hosts
    - Setting permissions and ownership
3. **Using the template Module**
    - Rendering files with Jinja2 templates
    - Using variables to customize file deployment
4. **Managing Directories on Managed Hosts**

- Using the file module to create, modify, and remove directories
- Setting ownership and permissions for directories

5. **Synchronizing Files with the synchronize Module**
    - Syncing files between local and remote systems
    - Managing directories efficiently with rsync
6. **File Management Best Practices**
    - Managing large files and directories
    - Ensuring idempotency in file deployments
7. **Error Handling in File Deployment**
    - Handling errors when copying files
    - Ensuring deployment is successful even in case of errors
8. **Testing File Deployment**
    - Verifying files are deployed correctly
    - Using checksums and diffs to verify file integrity

Module 6: Manage Complex Plays and Playbooks

1. **Complex Playbook Structures**
    - Organizing plays into multiple tasks
    - Writing modular, reusable playbooks
2. **Managing Large Inventories**
    - Handling large numbers of hosts in inventory
    - Grouping hosts by roles and attributes
3. **Handling Complex Variables**
    - Using complex data structures like dictionaries and lists
    - Passing variables between plays and tasks
4. **Writing Plays with Multiple Hosts**
    - Defining multiple hosts per play
    - Running tasks across a variety of hosts
5. **Conditionally Executing Tasks Across Multiple Hosts**

- Using the when directive for conditional task execution
- Managing host-specific tasks in large playbooks

6. **Running Playbooks in Parallel**
    - Using async and poll for parallel task execution
    - Optimizing playbooks for speed
7. **Reusing Playbooks for Different Environments**
    - Using variables to customize playbook behavior
    - Creating reusable playbook templates for different environments
8. **Debugging Complex Playbooks**
    - Troubleshooting issues in large playbooks
    - Using verbosity and error messages for debugging

Module 7: Simplify Playbooks with Roles

1. **Introduction to Ansible Roles**
    - What are roles and why use them?
    - The structure of a role
2. **Creating Custom Roles**
    - Writing a custom role with tasks, handlers, and defaults
    - Organizing playbooks using roles
3. **Using Predefined Roles from Ansible Galaxy**
    - Installing and using roles from Ansible Galaxy
    - Customizing downloaded roles for specific use cases
4. **Role Variables and Defaults**
    - Managing variables in roles
    - Setting default variables and overriding them
5. **Role Dependencies**
    - Defining dependencies between roles
    - Using dependencies field to manage role interactions
6. **Optimizing Playbooks with Roles**

- Reusing roles across multiple playbooks
- Managing roles in large-scale environments

7. **Managing Role Files and Templates**
    - Using files and templates within roles
    - Organizing and managing role content
8. **Testing and Debugging Roles**
    - Using ansible-lint to ensure role quality
    - Debugging role execution

Module 8: Troubleshoot Ansible

1. **Understanding Common Ansible Errors**
    - Analyzing error messages and troubleshooting playbook failures
    - Interpreting syntax, task, and connection errors
2. **Verbose Mode for Debugging**
    - Using -v, -vv, -vvv options to get more detailed output
    - Understanding the output from debug mode
3. **Using the debug Module**
    - Adding debug tasks to help troubleshoot playbooks
    - Inspecting variables and facts with the debug module
4. **Handling Network and SSH Issues**
    - Troubleshooting SSH connectivity problems
    - Verifying network configurations
5. **Error Handling in Playbooks**
    - Using failed_when and ignore_errors to manage playbook failures
    - Implementing task retries and timeouts
6. **Validating Playbooks with ansible-lint**
    - Using Ansible Lint to check for best practices
    - Identifying and fixing issues before execution
7. **Troubleshooting with Logs**

- Enabling and analyzing Ansible logs
- Using ansible-playbook with logging enabled

8. **Best Practices for Troubleshooting Playbooks**
    - Structuring playbooks for easier debugging
    - Common strategies for isolating issues

Module 9: Automate Linux Administration Tasks

1. **Automating Package Management**
    - Installing, updating, and removing packages with Ansible
    - Ensuring package idempotency in playbooks
2. **Automating Service Management**
    - Starting, stopping, and restarting services
    - Ensuring services are running as expected
3. **User and Group Management**
    - Creating and managing users and groups
    - Modifying user privileges and managing access
4. **Managing System Files and Directories**
    - Automating configuration file edits
    - Creating and managing system directories
5. **Network Configuration Automation**
    - Automating network configuration changes
    - Managing firewall settings and network interfaces
6. **Disk and File System Management**
    - Automating disk partitioning, mounting, and file system management
    - Ensuring disk usage limits are not exceeded
7. **Scheduled Task Automation**
    - Managing cron jobs with Ansible
    - Ensuring periodic tasks are automated and reliable
8. **Security and Compliance Automation**

- Applying security patches with Ansible
- Automating system hardening tasks

Module 10: Advanced Playbook Concepts

1. **Defining Roles in Ansible**
    - Structure and purpose of roles
    - Breaking down a playbook into roles for modularity
2. **Role Dependencies**
    - Defining role dependencies to create complex setups
    - Using the dependencies keyword to automate role chaining
3. **Managing Role Variables**
    - Defining variables in roles
    - Overriding role variables in playbooks and inventories
4. **Using Ansible Galaxy for Roles**
    - Installing and utilizing roles from Ansible Galaxy
    - Customizing downloaded roles to fit specific needs
5. **Managing Files and Templates in Roles**
    - Using the file and template modules within roles
    - Creating custom configuration files using Jinja2 templates
6. **Role Handlers**
    - Understanding handlers within roles
    - Triggering actions only when necessary in a role
7. **Best Practices for Role Structure**
    - Organizing roles for efficiency and clarity
    - Structuring roles to ensure scalability and maintainability
8. **Debugging and Testing Roles**
    - Debugging issues within roles
    - Using ansible-lint for quality checks on roles

Module 11: Advanced Inventory Management

1. **Introduction to Dynamic Inventory**
    - What is dynamic inventory?
    - Setting up dynamic inventories using cloud providers like AWS, Azure, or Google Cloud
2. **Inventory Plugins**
    - Using different inventory plugins (e.g., AWS EC2, VMware, etc.)
    - Setting up inventory sources with plugins
3. **Grouping Hosts in Inventories**
    - Creating and managing host groups
    - Using group variables for better organization
4. **Variables in Inventories**
    - Defining and overriding variables in inventory files
    - Using group and host-specific variables effectively
5. **Host Facts and Dynamic Variables**
    - Using ansible_facts to gather system information dynamically
    - Customizing inventory with host facts
6. **Inventory Hostnames and Aliases**
    - Managing hostnames and creating aliases in inventories
    - Using ansible_ssh_host for controlling SSH connections
7. **Using Group Variables for Configuration Management**
    - Defining common variables for a group of hosts
    - Ensuring consistent configuration across multiple hosts
8. **Inventory File Best Practices**
    - Organizing inventory files for large-scale environments
    - Using .ini, .yml, or .json formats and deciding when to use each

---

Module 12: Ansible Facts and Variables

1. **What Are Ansible Facts?**
    - Understanding facts and how they are gathered
    - How to use the setup module to gather facts

2. **Using Facts in Playbooks**
    - Accessing and using facts in tasks and templates
    - Fact-based conditional execution
3. **Defining and Using Variables**
    - Scope of variables in Ansible
    - Defining variables in playbooks, inventories, and files
4. **Handling Complex Data Structures**
    - Working with lists, dictionaries, and nested variables
    - Using Jinja2 filters to manipulate data structures
5. **Managing Secrets and Sensitive Data**
    - Using Ansible Vault to store sensitive variables
    - Best practices for securing passwords and API keys
6. **Overriding Variables**
    - Understanding variable precedence
    - Overriding variable values using command line or extra-vars
7. **Dynamic Variables and Facts**
    - Using dynamic inventories to pass variables to playbooks
    - Modifying facts at runtime
8. **Debugging Facts and Variables**
    - Using the debug module to print variables and facts
    - Troubleshooting issues related to variables and facts

Module 13: Ansible Automation with APIs

1. **Using Ansible with APIs**
    - Automating API calls with Ansible
    - Using the uri module to interact with RESTful APIs
2. **Interacting with Cloud APIs**
    - Automating cloud infrastructure management with Ansible (e.g., AWS, GCP)
    - Example playbooks to provision and manage resources on cloud platforms

3. **Handling Authentication with APIs**
    - Managing API keys and credentials
    - Using Ansible Vault to secure API tokens
4. **API Error Handling**
    - Handling API responses and errors in Ansible playbooks
    - Retrying failed API requests with Ansible task control
5. **Using Ansible to Automate Web Applications**
    - Automating web application deployment using APIs
    - Managing DNS, CDN, and load balancers through API integration
6. **Scheduling API Calls with Cron Jobs**
    - Automating API calls periodically
    - Setting up cron jobs using Ansible for periodic API interactions
7. **Automating Network Devices via APIs**
    - Using Ansible to interact with network device APIs (e.g., Cisco, Juniper)
    - Example playbooks to automate network configurations
8. **Best Practices for API Integration**
    - Using Ansible in a RESTful automation environment
    - Ensuring idempotency and managing retries when interacting with APIs

---

Module 14: Using Ansible for Configuration Management

1. **Configuration Management Overview**
    - What is configuration management and why it's important
    - Key components: idempotency, versioning, and consistency
2. **Ansible vs Other Configuration Management Tools**
    - Comparing Ansible to Chef, Puppet, and SaltStack
    - Benefits of using Ansible for configuration management
3. **Managing System Configuration Files**
    - Using the copy, template, and lineinfile modules to manage config files
    - Example use cases: managing /etc/hosts, /etc/nginx/nginx.conf, etc.

4. **Automating Software Installation and Configuration**
    - Using Ansible to ensure consistent software installations
    - Automating configuration settings for different environments
5. **Managing Package Updates with Ansible**
    - Keeping systems up-to-date by automating package updates
    - Managing software repositories using Ansible
6. **Automating Server Hardening and Security Configuration**
    - Using Ansible to apply security patches and enforce best practices
    - Automating firewall and SELinux settings
7. **Ensuring Compliance with Desired State**
    - Using Ansible for compliance auditing
    - Ensuring servers meet security baselines and configuration standards
8. **Best Practices for Configuration Management**
    - Managing configurations across large infrastructures
    - Organizing playbooks and roles for maintainability

Module 15: Using Ansible for Application Deployment

1. **Automating Application Installation**
    - Using Ansible to install and configure applications
    - Best practices for deploying applications consistently across environments
2. **Managing Deployment Artifacts**
    - Managing binaries, configuration files, and packages in deployments
    - Using the unarchive and copy modules for deployment artifacts
3. **Deploying Web Applications**
    - Automating deployment for common web servers (Apache, Nginx)
    - Managing SSL certificates, virtual hosts, and application configurations
4. **Automating Database Deployment**
    - Using Ansible to deploy and configure databases (MySQL, PostgreSQL, MongoDB)
    - Example playbooks for creating and managing databases

5. **Scaling Applications with Ansible**
    - Using Ansible to scale applications horizontally
    - Adding or removing application servers automatically
6. **Continuous Integration (CI) with Ansible**
    - Integrating Ansible into CI/CD pipelines
    - Automating testing, staging, and production deployments
7. **Rolling Updates and Downtime Minimization**
    - Performing rolling updates with Ansible to ensure zero downtime
    - Techniques for updating servers with minimal service disruption
8. **Post-Deployment Configuration**
    - Automating post-deployment tasks like database migrations, cache clearing, etc.
    - Ensuring the application is properly configured after deployment

---

Module 16: Ansible for Continuous Delivery (CD)

1. **Introduction to Continuous Delivery**
    - What is Continuous Delivery and why is it important?
    - The role of automation in achieving continuous delivery
2. **Ansible in a CI/CD Pipeline**
    - How Ansible integrates with Jenkins, GitLab CI, or other CI tools
    - Automating deployments from code commit to production
3. **Managing Deployment Stages**
    - Using Ansible to automate deployment stages (development, staging, production)
    - Managing different environments using variables and inventories
4. **Automating Test Environments**
    - Using Ansible to spin up and tear down test environments
    - Running unit, integration, and acceptance tests automatically
5. **Rollback Mechanisms with Ansible**
    - Creating rollback strategies for failed deployments
    - Using Ansible to restore systems and applications to a previous state

6. **Version Control Integration**
    - Managing code deployments with version control systems (Git)
    - Automating code pull and deployment to remote servers
7. **Zero-Downtime Deployments**
    - Strategies for achieving zero-downtime deployments with Ansible
    - Using load balancers and rolling updates for continuous availability
8. **Best Practices for Continuous Delivery with Ansible**
    - Organizing playbooks, roles, and inventories for CI/CD success
    - Ensuring maintainability and scalability in a continuous delivery pipeline

---

Module 17: Ansible for Cloud Automation

1. **Introduction to Cloud Automation**
    - The benefits of automating cloud resources with Ansible
    - Cloud providers supported by Ansible (AWS, GCP, Azure, etc.)
2. **Provisioning Cloud Resources**
    - Automating instance creation, networking, and storage on AWS/GCP
    - Using Ansible modules for managing cloud resources
3. **Managing Cloud Infrastructure with Ansible**
    - Provisioning and managing EC2 instances, networks, and storage
    - Creating VPCs, subnets, and configuring security groups using Ansible
4. **Cloud Application Deployment with Ansible**
    - Deploying applications in cloud environments (AWS, GCP, Azure)
    - Using cloud-native services like ECS, Lambda, and GKE with Ansible
5. **Cost Optimization and Scaling Cloud Infrastructure**
    - Using Ansible to scale cloud resources based on load
    - Automating the shutdown and startup of cloud resources to optimize costs
6. **Integrating Cloud with On-Premise Systems**
    - Hybrid cloud automation strategies
    - Managing both cloud and on-premise environments with a single Ansible setup

7. **Security and Compliance in Cloud Automation**
    - Managing cloud security groups, IAM roles, and firewalls using Ansible
    - Automating security policies and audits in the cloud
8. **Best Practices for Cloud Automation**
    - Organizing cloud automation with Ansible for large-scale environments
    - Ensuring scalability, reliability, and maintainability in cloud environments

---

Module 18: Ansible for Network Automation

1. **Introduction to Network Automation**
    - What is network automation and why it's needed
    - Benefits of automating network configurations and tasks
2. **Using Ansible for Network Configuration**
    - Automating network device configuration with Ansible
    - Managing routers, switches, and firewalls using Ansible modules
3. **Managing Network Devices with Ansible**
    - Interacting with network device APIs and protocols (SSH, SNMP)
    - Using modules like ios_config, nxos_config, and eos_config
4. **Network Topology Automation**
    - Automating the creation of network topologies
    - Managing VLANs, interfaces, and routing configurations with Ansible
5. **Automating Network Troubleshooting**
    - Using Ansible to automate network diagnostics and logging
    - Troubleshooting network configurations with Ansible commands
6. **Network Security Automation**
    - Using Ansible to configure firewall rules and VPNs
    - Managing network security compliance through automation
7. **Monitoring Network Devices with Ansible**
    - Integrating Ansible with monitoring tools like Nagios or Prometheus
    - Automating the collection of network metrics and alerts

8. **Best Practices for Network Automation with Ansible**
    - Organizing network automation playbooks and roles
    - Ensuring scalability and maintainability in network automation

---

Module 19: Ansible for System Administration Automation

1. **Introduction to System Administration Automation**
    - What tasks can be automated for system administrators
    - The benefits of using Ansible for daily sysadmin tasks
2. **User and Group Management**
    - Automating user creation, deletion, and modification
    - Managing group memberships and privileges using Ansible
3. **Package Management Automation**
    - Installing, updating, and removing software packages with Ansible
    - Managing package repositories and updates
4. **Service Management Automation**
    - Managing system services (start, stop, enable) using Ansible
    - Configuring systemd units and init scripts with Ansible
5. **File System and Disk Management**
    - Automating disk partitioning, file systems, and mounts with Ansible
    - Configuring quotas and file system security with Ansible
6. **Scheduled Jobs Automation**
    - Managing cron jobs, at jobs, and systemd timers with Ansible
    - Ensuring scheduled tasks are idempotent and reliable
7. **Networking Automation for System Admins**
    - Automating network interface configurations and DNS settings
    - Managing firewalls, proxy servers, and routing
8. **Security and Patch Management**
    - Automating system hardening and patching
    - Managing security updates and compliance checks with Ansible

---

Module 20: Ansible Troubleshooting and Best Practices

1. **Debugging Playbooks and Tasks**
    - Using debug and verbose modes for troubleshooting
    - Common debugging strategies for resolving playbook issues
2. **Handling Errors in Ansible**
    - Using ignore_errors and failed_when to manage task failures
    - Troubleshooting connectivity and authentication issues
3. **Best Practices for Writing Efficient Playbooks**
    - Organizing playbooks for clarity and maintainability
    - Using loops, conditionals, and handlers effectively
4. **Using Ansible Linting and Testing Tools**
    - Introduction to ansible-lint for checking playbook quality
    - Setting up continuous integration (CI) for Ansible code testing
5. **Optimizing Playbook Performance**
    - Techniques for reducing playbook execution time
    - Managing large inventories and scaling playbooks for efficiency
6. **Managing Large-Scale Infrastructure**
    - Best practices for handling hundreds or thousands of nodes
    - Using parallelism and async execution to scale playbooks
7. **Version Control for Ansible**
    - Using Git to manage playbook versions
    - Collaboration and versioning strategies for team environments
8. **Best Practices for Ansible Security**
    - Securing sensitive data with Ansible Vault
    - Limiting access to sensitive playbooks and variables