Which configuration file on a Puppet Agent specifies the Puppet Master's address?

a) /etc/puppet/puppet.conf

b) /etc/hosts

c) /etc/ssh/sshd\_config

d) /etc/httpd/conf/httpd.conf

Answer: a

What is Advanced Puppet:

Question: In advanced Puppet, what is the purpose of using "Hiera" for data separation?

a) To store Puppet manifests

b) To keep sensitive data secure

c) To configure Puppet modules

d) To manage SSL certificates

Answer: b) To keep sensitive data secure

Operations Used in Puppet:

Question: When using Puppet, which operation ensures that the desired state of a resource is maintained?

a) Inspection

b) Execution

c) Compilation

d) Realization

Answer: b) Execution

Puppet on Command Line:

Question: What command can you use to apply a Puppet manifest from the command line?

a) puppet validate

b) puppet apply

c) puppet run

d) puppet execute

Answer: b) puppet apply

Managing Resources with the Puppet Apply Command:

Question: When using the puppet apply command, what does the -e flag allow you to do?

a) Specify the Puppet Master's address

b) Execute Puppet code directly from the command line

c) Enable debugging mode

d) Import external facts

Answer: b) Execute Puppet code directly from the command line

Puppet Manifests:

Question: In Puppet, what is a manifest?

a) A Puppet function

b) A custom resource type

c) A file that defines resource configurations

d) A type of SSL certificate

Answer: c) A file that defines resource configurations

Puppet Configuration:

Question: In Puppet, which configuration file is commonly used to specify the Puppet Master's settings?

a) puppet.conf

b) hiera.yaml

c) site.pp

d) manifests.pp

Answer: a) puppet.conf

Managing Packages in Puppet:

Question: Which Puppet resource type is typically used to manage package installations?

a) file

b) user

c) package

d) service

Answer: c) package

Puppet Module:

Question: What is the primary purpose of a Puppet module?

a) To execute custom scripts

b) To define Puppet classes

c) To store Puppet manifests

d) To encapsulate and share Puppet code

Answer: d) To encapsulate and share Puppet code

Monitoring Web Server:

Question: In Puppet, which tool is commonly used for monitoring web servers?

a) Nagios

b) PuppetDB

c) Hiera

d) Puppet Bolt

Answer: a) Nagios

Load Balancing the Cluster:

Question: What is the primary goal of load balancing in a cluster?

a) To distribute network traffic evenly among cluster nodes

b) To configure Puppet modules

c) To compile Puppet manifests

d) To manage SSL certificates

Answer: a) To distribute network traffic evenly among cluster nodes

Scaling Up the Puppet Environment:

Question: When scaling a Puppet environment, what is "horizontal scaling"?

a) Increasing the processing power of a single Puppet Master

b) Adding more Puppet Masters to the environment

c) Increasing the memory of a Puppet Agent

d) Managing Puppet modules

Answer: b) Adding more Puppet Masters to the environment

Connecting Puppet Agent with Puppet Master:

Question: How does a Puppet Agent communicate with the Puppet Master?

a) Using direct Ethernet connections

b) Via email notifications

c) Over HTTP using REST API

d) Through SSH connections

Answer: c) Over HTTP using REST API

Making the Configuration Dynamic:

Question: What is the primary goal of making Puppet configurations dynamic?

a) To reduce the number of Puppet modules

b) To automate configuration updates based on changing conditions

c) To optimize database queries

d) To create custom Puppet classes

Answer: b) To automate configuration updates based on changing conditions

Extending Puppet:

Question: In Puppet, what are "custom types and providers" used for?

a) To create custom Puppet classes

b) To manage SSL certificates

c) To extend Puppet's built-in resource types

d) To define external facts

Answer: c) To extend Puppet's built-in resource types

Puppet Classes:

Question: What is a Puppet class?

a) A type of SSL certificate

b) A module for managing network connections

c) A unit of configuration that can be included in Puppet manifests

d) A Puppet function

Answer: c) A unit of configuration that can be included in Puppet manifests

Puppet Function:

Question: What are Puppet functions used for?

a) To manage SSL certificates

b) To define custom resource types

c) To perform calculations and data transformations in Puppet code

d) To create custom Puppet classes

Answer: c) To perform calculations and data transformations in Puppet code

Puppet Custom Functions:

Question: What is a Puppet custom function?

a) A user-defined function for custom logic in Puppet code

b) A type of SSL certificate

c) A Puppet class for custom modules

d) A function provided by Puppet Labs for custom configurations

Answer: a) A user-defined function for custom logic in Puppet code

**Unit-3**

Continuous Monitoring: What is Continuous Monitoring:

Question: What is Continuous Monitoring in IT operations?

a) Periodic monitoring at fixed intervals

b) Monitoring that runs continuously to detect issues in real-time

c) Monitoring that occurs only during business hours

d) Manual monitoring conducted by IT administrators

Answer: b) Monitoring that runs continuously to detect issues in real-time

What is Nagios:

Question: What is Nagios?

a) A popular video game

b) An open-source monitoring and alerting system

c) A database management tool

d) A programming language

Answer: b) An open-source monitoring and alerting system

Why Nagios:

Question: Why is Nagios commonly used for monitoring IT infrastructure?

a) Because it's a proprietary, closed-source tool

b) Because it's a lightweight web browser

c) Because it provides real-time monitoring and alerting capabilities

d) Because it's primarily used for software development

Answer: c) Because it provides real-time monitoring and alerting capabilities

Nagios Architecture:

Question: In Nagios, what is the primary role of the Nagios Core component?

a) To generate reports

b) To provide a web-based interface

c) To perform actual monitoring and alerting

d) To manage configuration files

Answer: c) To perform actual monitoring and alerting

Introducing Plugins:

Question: What are Nagios plugins?

a) Add-ons for web browsers

b) Small programs that perform specific monitoring tasks

c) Hardware components of a monitoring system

d) Data visualization tools

Answer: b) Small programs that perform specific monitoring tasks

Benefits of Monitoring:

Question: What is one of the benefits of continuous monitoring with Nagios?

a) Reduced system downtime

b) Limited scalability

c) Manual alerting

d) Minimal configuration

Answer: a) Reduced system downtime

Main Features:

Question: Which of the following is a main feature of Nagios?

a) Virtual reality simulation

b) Real-time stock market data

c) Customizable alerting and reporting

d) Video streaming

Answer: c) Customizable alerting and reporting

Soft and Hard States:

Question: In Nagios, what does a "soft" state indicate?

a) A problem has been acknowledged but not yet resolved

b) A problem has been resolved

c) A problem is in a critical state

d) A problem is in a passive state

Answer: a) A problem has been acknowledged but not yet resolved

Installing Nagios: Installing Nagios using Package Managers: Installation with apt-get/dpkg:

Question: Which package management system is commonly used for installing Nagios on Debian-based systems?

a) apt-get/dpkg

b) yum/rpm

c) pip

d) brew

Answer: a) apt-get/dpkg

yum/rpm:

Question: Which package management system is commonly used for installing Nagios on Red Hat-based systems?

a) apt-get/dpkg

b) yum/rpm

c) pip

d) brew

Answer: b) yum/rpm

Installing Prerequisites:

Question: Before installing Nagios, what are some typical prerequisites that need to be installed?

a) A web browser and a text editor

b) A graphics card and monitor

c) A printer and scanner

d) Apache web server, PHP, and GCC

Answer: d) Apache web server, PHP, and GCC

Compiling and installing Nagios:

Question: In some cases, what might be required after downloading the Nagios source code?

a) Taking a coffee break

b) Running the make and make install commands

c) Uninstalling the operating system

d) Checking social media

Answer: b) Running the make and make install commands

Setting up web server:

Question: Which web server is often used in conjunction with Nagios?

a) Internet Explorer

b) Apache

c) Firefox

d) Chrome

Answer: b) Apache

Commands:

Question: In Nagios, what are "commands" used for?

a) Creating virtual machines

b) Executing checks and notifications

c) Sending emails

d) Browsing websites

Answer: b) Executing checks and notifications

Objects:

Question: In Nagios, what are "objects"?

a) Physical items

b) JavaScript files

c) Configuration elements such as hosts, services, and contacts

d) Software applications

Answer: c) Configuration elements such as hosts, services, and contacts

**Unit-4**

Configuring Nagios to Monitor Web Server:

Question: What is the primary configuration file in Nagios where you define host and service definitions?

a) nagios.cfg

b) hosts.conf

c) nagios.conf

d) objects.cfg

Answer: d) objects.cfg

Question: In Nagios, what is the purpose of defining host templates?

a) To specify custom commands for individual hosts

b) To group hosts with similar configurations

c) To define custom downtime periods

d) To manage comments for hosts

Answer: b) To group hosts with similar configurations

Using the Built-in Web Interface: The Nagios Web Interface:

Question: Which programming language is commonly used to develop the Nagios web interface?

a) JavaScript

b) Ruby

c) Python

d) PHP

Answer: d) PHP

Question: In the Nagios web interface, what does the "Downtime" feature allow you to do?

a) Manage host and service downtime periods

b) Create custom web pages

c) Manage user accounts

d) Execute commands on remote hosts

Answer: a) Manage host and service downtime periods

Managing Hosts:

Question: In Nagios, what is a "contact group"?

a) A group of hosts with similar configurations

b) A group of users or email addresses

c) A group of web pages

d) A group of monitoring commands

Answer: b) A group of users or email addresses

Question: How do you define a new host in Nagios configuration?

a) Using the "hosts.conf" file

b) Using the Nagios web interface

c) By editing the "nagios.cfg" file

d) By running a Python script

Answer: b) Using the Nagios web interface

Managing Services:

Question: What is a service dependency in Nagios?

a) A situation where one service relies on another to function correctly

b) A way to disable services temporarily

c) A configuration file for services

d) A group of services with similar characteristics

Answer: a) A situation where one service relies on another to function correctly

Question: In Nagios, how can you acknowledge a problem with a service?

a) By rebooting the server

b) By editing the "nagios.cfg" file

c) Using the Nagios web interface

d) By sending an email

Answer: c) Using the Nagios web interface

Managing Downtimes:

Question: Why might you schedule downtime for a host or service in Nagios?

a) To permanently disable it

b) To simulate a network outage

c) To perform maintenance without generating alerts

d) To speed up the monitoring process

Answer: c) To perform maintenance without generating alerts

Question: In Nagios, how do you schedule downtime for multiple hosts or services simultaneously?

a) By editing the "nagios.cfg" file

b) By using the command-line interface

c) By sending an email

d) By rebooting the server

Answer: b) By using the command-line interface

Managing Comments:

Question: In Nagios, what is the purpose of adding comments to hosts and services?

a) To disable them temporarily

b) To improve code readability

c) To provide additional information or context

d) To create new monitoring commands

Answer: c) To provide additional information or context

Question: How can you add a comment to a host or service in the Nagios web interface?

a) By running a PHP script

b) By using a text editor

c) By clicking the "Add Comment" button

d) By editing the "nagios.cfg" file

Answer: c) By clicking the "Add Comment" button

Nagios Information:

Question: In Nagios, what is the purpose of the status.dat file?

a) To store user account information

b) To store historical status information about hosts and services

c) To store PHP scripts for the web interface

d) To store database credentials

Answer: b) To store historical status information about hosts and services

Question: How can you view detailed information about the current status of hosts and services in Nagios?

a) By editing the "nagios.cfg" file

b) By running a Python script

c) By using the Nagios web interface

d) By sending an email

Answer: c) By using the Nagios web interface

Learning Command-Line Interfaces:

Question: Which command is commonly used to start, stop, or restart the Nagios service from the command line?

a) start-nagios

b) nagios-restart

c) systemctl

d) run-nagios

Answer: c) systemctl

Deploying a Simple Web Application on a Server:

Question: What is the purpose of deploying a web application on a server?

a) To monitor web applications using Nagios

b) To improve server performance

c) To write code for Nagios plugins

d) To test network connectivity

Answer: a) To monitor web applications using Nagios

**Unit 5**

Introduction to Ansible and Configuration Management:

Question: What is Ansible primarily used for in IT infrastructure?

a) Network routing

b) System monitoring

c) Configuration management

d) Web development

Answer: c) Configuration management

Question: What is the primary goal of configuration management in IT?

a) To maximize energy efficiency

b) To automate repetitive tasks and maintain system consistency

c) To build complex web applications

d) To improve network latency

Answer: b) To automate repetitive tasks and maintain system consistency

How Ansible Works:

Question: In Ansible, what is a "playbook"?

a) A book of Shakespearean plays

b) A list of registered Ansible users

c) A YAML file that defines automation tasks

d) A list of web servers

Answer: c) A YAML file that defines automation tasks

Question: What is the primary method Ansible uses to communicate with target hosts?

a) SSH

b) HTTP

c) FTP

d) Telnet

Answer: a) SSH

Modern Infrastructure Management:

Question: What does "Infrastructure as Code" (IaC) refer to?

a) The practice of manually configuring servers

b) Treating infrastructure configuration like software code

c) The use of physical hardware for all infrastructure

d) Using configuration files to manage network routers

Answer: b) Treating infrastructure configuration like software code

Ansible and RedHat:

Question: Who is the primary sponsor and maintainer of Ansible?

a) Microsoft

b) Amazon Web Services (AWS)

c) Red Hat

d) Google

Answer: c) Red Hat

Architecture of Ansible:

Question: In Ansible, what is the role of the "ansible-playbook" command?

a) To update the Ansible software

b) To execute automation tasks defined in playbooks

c) To configure network routers

d) To install third-party plugins

Answer: b) To execute automation tasks defined in playbooks

Question: In Ansible, what is an "Ansible Control Node"?

a) A server responsible for managing inventory

b) A target host being configured

c) A physical network device

d) An external database server

Answer: a) A server responsible for managing inventory

Ansible Infrastructure Management: On Snowflakes and Shell Scripts:

Question: What does the term "snowflake server" refer to in IT infrastructure?

a) A server with high performance

b) A server with unique, manual configurations

c) A server located in a snowy region

d) A server running a specific OS

Answer: b) A server with unique, manual configurations

Question: How does Ansible address the challenge of "snowflake servers"?

a) By promoting the use of more snowflake servers

b) By manually configuring each server

c) By using consistent automation to build and manage servers

d) By ignoring the issue of server uniqueness

Answer: c) By using consistent automation to build and manage servers

Installing Ansible:

Question: Which of the following package managers is commonly used to install Ansible on Linux systems?

a) brew

b) pip

c) yum/dnf

d) npm

Answer: c) yum/dnf

Creating a Basic Inventory File:

Question: In Ansible, what is an "inventory file"?

a) A list of playbooks

b) A file containing variables

c) A list of target hosts and their groupings

d) A file used for debugging

Answer: c) A list of target hosts and their groupings

Question: How can you specify a group of target hosts in an Ansible inventory file?

a) By using a wildcard character

b) By adding a "group" keyword

c) By assigning a group name in square brackets

d) By using a specific file extension

Answer: c) By assigning a group name in square brackets

Ansible and Vagrant: Setting up Vagrant:

Question: What is Vagrant primarily used for in the context of Ansible?

a) To deploy web applications

b) To manage virtualized development environments

c) To configure network routers

d) To monitor server performance

Answer: b) To manage virtualized development environments

Using Ansible with Vagrant:

Question: How does Ansible interact with virtual machines managed by Vagrant?

a) Ansible uses SSH to communicate with Vagrant-managed VMs

b) Ansible connects to Vagrant VMs using HTTP

c) Ansible requires a separate plugin for Vagrant integration

d) Ansible communicates with Vagrant through a physical network connection

Answer: a) Ansible uses SSH to communicate with Vagrant-managed VMs

**Unit-6**

Ansible Roles and Command Line Usage:

Question: What are Ansible roles used for, and how do they promote best practices in playbooks?

a) Ansible roles are used for creating complex playbooks, and they simplify code management.

b) Ansible roles are used for defining variables, and they enhance code readability.

c) Ansible roles are used for organizing and reusing playbooks, and they encourage modularization and code reuse.

d) Ansible roles are used for running playbooks with multiple inventories, and they optimize inventory management.

Answer: c) Ansible roles are used for organizing and reusing playbooks, and they encourage modularization and code reuse.

Ansible Playbook: Power Plays:

Question: What is idempotence in the context of Ansible playbooks, and why is it important?

a) Idempotence refers to playbooks that only run once and are not suitable for production environments.

b) Idempotence means that a playbook's result is the same, whether it's run once or multiple times, which is crucial for ensuring predictable and safe automation.

c) Idempotence is a deprecated feature in Ansible, no longer relevant in modern versions.

d) Idempotence refers to the speed at which playbooks execute, making them faster for large-scale deployments.

Answer: b) Idempotence means that a playbook's result is the same, whether it's run once or multiple times, which is crucial for ensuring predictable and safe automation.

Running Playbooks with ansible-playbook:

Question: How can you limit the execution of specific tasks within a playbook when using ansible-playbook from the command line?

a) Use the --limit flag followed by a specific task name.

b) Use the --tags flag followed by the task tag name.

c) Use the --only flag followed by the task name.

d) Use the --task flag followed by the task ID.

Answer: b) Use the --tags flag followed by the task tag name.

Real-world Playbook:

Question: In a real-world scenario, explain the role of a playbook in an application deployment pipeline.

a) Playbooks are used only for initial system setup; they are not part of the deployment pipeline.

b) Playbooks are used to automate the entire application deployment process, from configuring servers to deploying application code.

c) Playbooks are used solely for monitoring and alerting in the deployment pipeline.

d) Playbooks are used to create backups but not for application deployment.

Answer: b) Playbooks are used to automate the entire application deployment process, from configuring servers to deploying application code.

Handlers:

Question: How are Ansible handlers different from regular tasks in a playbook?

a) Handlers are executed immediately when triggered, while tasks are executed at the end of the playbook run.

b) Handlers are tasks that respond to events and are only executed when notified, while regular tasks are always executed.

c) Handlers can only be defined in roles, while regular tasks can be defined anywhere in a playbook.

d) Handlers are used for debugging and testing, while regular tasks are used in production.

Answer: b) Handlers are tasks that respond to events and are only executed when notified, while regular tasks are always executed.

Environment Variables:

Question: How can you set environment variables for an Ansible playbook to use during execution?

a) Environment variables cannot be set for Ansible playbooks.

b) Define them in the playbook's vars section.

c) Export them in the shell where you run the playbook.

d) Use the --environment flag with ansible-playbook.

Answer: c) Export them in the shell where you run the playbook.

Variables:

Question: What's the difference between Ansible's fact variables and user-defined variables?

a) Fact variables are predefined by Ansible and cannot be modified, while user-defined variables are set by the playbook author.

b) Fact variables are defined in the playbook, while user-defined variables are collected from target hosts during playbook execution.

c) Fact variables store facts about the Ansible control node, while user-defined variables store facts about target hosts.

d) Fact variables and user-defined variables are the same; there's no distinction.

Answer: a) Fact variables are predefined by Ansible and cannot be modified, while user-defined variables are set by the playbook author.

Facts:

Question: How can you gather facts about target hosts in Ansible?

a) Facts are automatically collected by Ansible during playbook execution; no specific action is required.

b) Use the ansible-facts command to manually collect facts.

c) Facts can only be gathered from AWS instances, not other types of hosts.

d) Define facts in the playbook's vars section.

Answer: a) Facts are automatically collected by Ansible during playbook execution; no specific action is required.

Prompts:

Question: In Ansible, what is the purpose of using prompts in playbooks?

a) Prompts are used to request user input during playbook execution.

b) Prompts are used for displaying notifications and alerts.

c) Prompts are used to pause playbook execution for a specified duration.

d) Prompts are not supported in Ansible playbooks.

Answer: a) Prompts are used to request user input during playbook execution.

Tags:

Question: How can you use tags in Ansible playbooks, and why are they useful?

a) Tags are used to categorize playbooks for documentation purposes.

b) Tags are used to filter which tasks in a playbook are executed, allowing for selective execution and better playbook control.

c) Tags are used to create dependencies between playbooks.

d) Tags are only applicable to roles, not individual tasks.

Answer: b) Tags are used to filter which tasks in a playbook are executed, allowing for selective execution and better playbook control.

Blocks:

Question: What is an Ansible block, and how can it simplify playbook structure?

a) An Ansible block is a unit of code that can be executed multiple times in a playbook.

b) An Ansible block is a way to group tasks together and apply common error handling or rescue actions to them.

c) An Ansible block is a deprecated feature and should not be used in modern playbooks.

d) An Ansible block is a complex data structure used to define variables.

Answer: b) An Ansible block is a way to group tasks together and apply common error handling or rescue actions to them.

Understanding the Relation of AWS and Ansible:

Question: How can Ansible be used to interact with AWS resources?

a) Ansible has no integration with AWS.

b) Ansible can only manage AWS EC2 instances.

c) Ansible provides AWS-specific modules to manage AWS resources, including EC2 instances, S3 buckets, and more.

d) Ansible can only manage AWS billing information.

Answer: c) Ansible provides AWS-specific modules to manage AWS resources, including EC2 instances, S3 buckets, and more.

Application Deployment with the Help of Ansible:

Question: How does Ansible contribute to the automation of application deployment processes?

a) Ansible automates the entire application development lifecycle.

b) Ansible automates server provisioning, configuration, and application deployment, ensuring consistent and repeatable deployments.

c) Ansible only manages application deployments for cloud-based applications.

d) Ansible is not suitable for application deployment.

Answer: b) Ansible automates server provisioning, configuration, and application deployment, ensuring consistent and repeatable deployments.

Ansible Roles and Command Line Usage:

Question: How can Ansible roles enhance code maintainability and reusability in playbooks?

a) Roles allow you to define variables, but they cannot be reused in other playbooks.

b) Roles encapsulate reusable tasks, variables, and handlers, making playbooks more modular and shareable.

c) Roles are used for specifying host groups, simplifying playbook structure.

d) Roles can only be applied to individual tasks, not entire playbooks.

Answer: b) Roles encapsulate reusable tasks, variables, and handlers, making playbooks more modular and shareable.

Ansible Playbook: Power Plays:

Question: Explain the concept of playbook "roles" in Ansible and how they differ from traditional playbooks.

a) Roles are used only for defining variables, while traditional playbooks contain tasks.

b) Roles are a way to organize tasks and variables within a playbook, improving playbook readability.

c) Roles are synonymous with traditional playbooks, and there is no difference between them.

d) Roles are deprecated in Ansible, and traditional playbooks should be used instead.

Answer: b) Roles are a way to organize tasks and variables within a playbook, improving playbook readability.

Running Playbooks with ansible-playbook:

Question: What is the purpose of the --check flag in the ansible-playbook command, and how does it affect playbook execution?

a) The --check flag performs a dry run of the playbook without making any changes to the target hosts, allowing you to preview the changes.

b) The --check flag is used to check if the playbook is syntactically correct but does not execute it.

c) The --check flag checks the health status of target hosts and reports any issues.

d) The --check flag skips tasks marked with the no\_log parameter.

Answer: a) The --check flag performs a dry run of the playbook without making any changes to the target hosts, allowing you to preview the changes.

Handlers:

Question: Explain the purpose of Ansible handlers and provide an example scenario where they are useful.

a) Handlers are used to perform cleanup tasks after playbook execution, such as deleting temporary files.

b) Handlers are used to execute tasks immediately when triggered by a specific event, like restarting a service after a configuration change.

c) Handlers are used to run tasks on the Ansible control node to prepare for playbook execution.

d) Handlers are used for debugging purposes only.

Answer: b) Handlers are used to execute tasks immediately when triggered by a specific event, like restarting a service after a configuration change.

Environment Variables:

Question: In what situations might you use environment variables to influence playbook behavior, and how can they be set?

a) Environment variables are used for defining variables within a playbook to improve code readability.

b) Environment variables are primarily used for securing sensitive information in playbooks.

c) Environment variables can be used to configure dynamic inventory sources or provide credentials and can be set in the shell environment where ansible-playbook is executed.

d) Environment variables are deprecated in Ansible.

Answer: c) Environment variables can be used to configure dynamic inventory sources or provide credentials and can be set in the shell environment where ansible-playbook is executed.

Variables:

Question: How can you override variable values defined in an Ansible role from within a playbook?

a) Variables defined in roles cannot be overridden in playbooks.

b) Use the --override-vars flag with ansible-playbook to provide new values.

c) Define the same variable with a different value in the playbook, and it will take precedence.

d) Variables defined in playbooks are always evaluated before those in roles.

Answer: c) Define the same variable with a different value in the playbook, and it will take precedence.

Facts:

Question: Explain the role of Ansible facts in playbooks, and provide an example of how they can be used.

a) Facts are user-defined variables that are available to playbooks for customization.

b) Facts are used to store static data in playbooks.

c) Facts are automatically collected information about target hosts, such as network interfaces, operating system, and available memory, and they can be used for conditional tasks or template rendering.

d) Facts are deprecated in modern Ansible versions.

Answer: c) Facts are automatically collected information about target hosts, such as network interfaces, operating system, and available memory, and they can be used for conditional tasks or template rendering.