

Red Hat Certification Exam Objectives by Version

[EX180K - Red Hat Certified Specialist in Containers and Kubernetes](#)

[EX183K - Red Hat Certified Enterprise Application Developer](#)

[EX188K - Red Hat Certified Specialist in Containers](#)

[EX200K - Red Hat Certified System Administrator](#)

[EX210K - Red Hat Certified System Administrator in Red Hat OpenStack](#)

[EX221K - Red Hat Certified Specialist in Cloud-native Integration](#)

[EX229K - Red Hat Certified Specialist in ROSA](#)

[EX240K - Red Hat Certified Specialist in API Management](#)

[EX248K - Red Hat Certified Specialist in Enterprise Application Server Administration](#)

[EX260K - Red Hat Certified Specialist in Ceph Cloud Storage](#)

[EX267K - Red Hat Certified Specialist in OpenShift AI](#)

[EX280K - Red Hat Certified Specialist in OpenShift Administration](#)

[EX288K - Red Hat Certified Specialist in OpenShift Application Development](#)

[EX294K - Red Hat Certified Engineer](#)

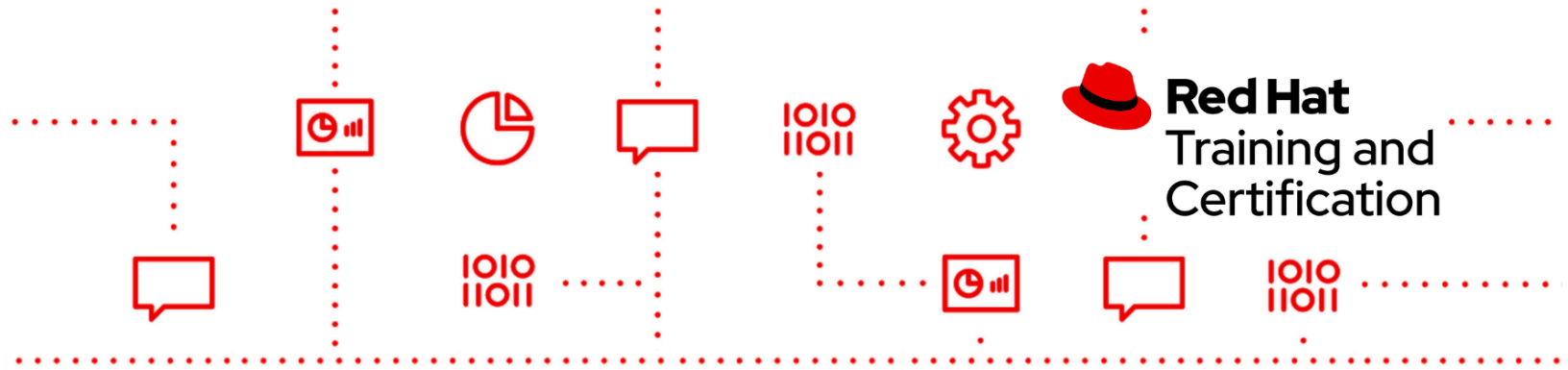
[EX316K - Red Hat Certified Specialist in OpenShift Virtualization Exam](#)

[EX318K - Red Hat Certified Specialist in Virtualization](#)

[EX328K - Red Hat Certified Specialist in Building Resilient Microservices](#)

[EX342K - Red Hat Certified Specialist in Enterprise Linux Diagnostics and Troubleshooting](#)

[EX358K - Red Hat Certified Specialist in Services Management and Automation](#)



[EX362K - Red Hat Certified Specialist in Identity Management](#)

[EX364K - Red Hat Certified Specialist in Decision Management](#)

[EX370K - Red Hat Certified Specialist in OpenShift Data Foundation](#)

[EX374K - Red Hat Certified Specialist in Developing Automation with Ansible Automation Platform](#)

[EX378K - Red Hat Certified Cloud-native Developer](#)

[EX380K - Red Hat Certified Specialist in OpenShift Automation and Integration](#)

[EX403K - Red Hat Certified Specialist in Enterprise Deployment and Systems Management](#)

[EX415K - Red Hat Certified Specialist in Security: Linux](#)

[EX427K - Red Hat Certified Specialist in Business Process Design](#)

[EX436K - Red Hat Certified Specialist in High Availability Clustering](#)

[EX440K - Red Hat Certified Specialist in Messaging Administration](#)

[EX442K - Red Hat Certified Specialist in Performance Tuning](#)

[EX447K - Red Hat Certified Specialist in Advanced Automation: Ansible Best Practices](#)

[EX457K - Red Hat Certified Specialist in Ansible Network Automation](#)

[EX467K - Red Hat Certified Specialist in Managing Automation with Ansible Automation Platform](#)

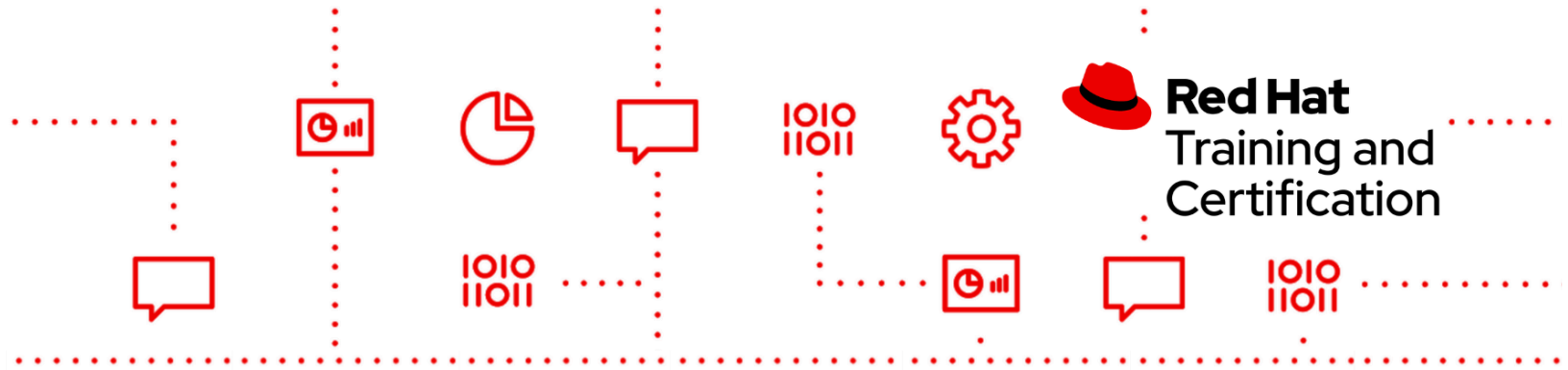
[EX480K - Red Hat Certified Specialist in MultiCluster Management](#)

[EX482K - Red Hat Certified Specialist in Event-Driven Development with Kafka](#)

[PE110K - Preliminary Exam in Red Hat OpenStack Administration](#)

[PE124K - Preliminary Exam in Red Hat OpenStack Administration](#)

[PE180K - Preliminary Exam in Red Hat OpenShift Administration](#)



[EX180K - Red Hat Certified Specialist in Containers and Kubernetes](#)

EX180V46K - Retired

- Implement images using Podman
 - Understand and use FROM (the concept of a base image) instruction.
 - Understand and use RUN instruction.
 - Understand and use ADD instruction.
 - Understand and use COPY instruction.
 - Understand the difference between ADD and COPY instructions.
 - Understand and use WORKDIR and USER instructions.
 - Understand security-related topics.
 - Understand the differences and applicability of CMD vs. ENTRYPOINT instructions.
 - Understand ENTRYPOINT instruction with param.
 - Understand when and how to expose ports from a Docker file.
 - Understand and use environment variables inside images.
 - Understand ENV instruction.
 - Understand container volume.
 - Mount a host directory as a data volume.
 - Understand security and permissions requirements related to this approach.
 - Understand lifecycle and cleanup requirements of this approach.
- Manage images
 - Understand private registry security.
 - Interact with many different registries.
 - Understand and use image tags.
 - Push and pull images from and to registries.



- Back up an image with its layers and meta data vs. backup a container state.
- Run containers locally using Podman
 - Get container logs.
 - Listen to container events on the container host.
 - Use Podman inspect.
- Basic OpenShift knowledge
- Creating applications in OpenShift
 - Create, manage and delete projects from a template, from source code, and from an image.
 - Customize catalog template parameters.
 - Specifying environment parameters.
 - Expose public applications.
- Troubleshoot applications in OpenShift
 - Understand the description of application resources.
 - Get application logs.
 - Inspect running applications.
 - Connecting to containers running in a pod.
 - Copy resources to/from containers running in a pod.

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX180V410K - Retired

- Implement images using Podman
 - Understand and use FROM (the concept of a base image) instruction.
 - Understand and use RUN instruction.
 - Understand and use ADD instruction.
 - Understand and use COPY instruction.
 - Understand the difference between ADD and COPY instructions.
 - Understand and use WORKDIR and USER instructions.



- Understand security-related topics.
 - Understand the differences and applicability of CMD vs. ENTRYPOINT instructions.
 - Understand ENTRYPOINT instruction with param.
 - Understand when and how to expose ports from a Docker file.
 - Understand and use environment variables inside images.
 - Understand ENV instruction.
 - Understand container volume.
 - Mount a host directory as a data volume.
 - Understand security and permissions requirements related to this approach.
 - Understand lifecycle and cleanup requirements of this approach.
- Manage images
 - Understand private registry security.
 - Interact with many different registries.
 - Understand and use image tags.
 - Push and pull images from and to registries.
 - Back up an image with its layers and meta data vs. backup a container state.
- Run containers locally using Podman
 - Get container logs.
 - Listen to container events on the container host.
 - Use Podman inspect.
- Basic OpenShift knowledge
- Creating applications in OpenShift
 - Create, manage and delete projects from a template, from source code, and from an image.
 - Customize catalog template parameters.
 - Specifying environment parameters.
 - Expose public applications.
- Troubleshoot applications in OpenShift
 - Understand the description of application resources.
 - Get application logs.
 - Inspect running applications.



- Connecting to containers running in a pod.
- Copy resources to/from containers running in a pod.

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX183K - Red Hat Certified Enterprise Application Developer](#)

EX183V70K

Bean validation

- Annotate a POJO to enable and set up data validation.
- Use the documentation to find useful standard validators.

JPA mapping

- Annotate a POJO to map it to its persistent state representation in the database.
- Apply basic property mapping and be able to use the documentation to identify correct property annotations (e.g., @Temporal).
- Map a bidirectional OneToMany relationship between two entities, including both sides of the association.
- Understand default fetching behavior and be able to override the fetching strategy per association.

JPA query

- Implement basic JPA queries using named parameters.
- Create and use a named query.
- Use a query to eager fetch an association.

Messaging

- Understand point-to-point vs. publish/subscribe models.
- Understand JMS queues, topics, and connection factories.
- Understand and use the `javax.jms.MessageListener` interface.



- Implement a message-driven bean.
- Use the @MessageDriven and @ActivationConfigProperty annotations.

REST services with JAX-RS

- Understand REST concepts, particularly the application and use of the HTTP PUT, DELETE, GET, and POST methods.
- Know and use standard HTTP return codes.
- Implement RESTful Root resource class.
- Expose a REST service using JAX-RS.
- Demonstrate ability to define @Path.
- Understand and use @Produce and @Consume.
- Be able to both consume and produce xml- and json-formatted content using JAX-RS.

Security

- Understand basic JAAS terms and concepts.
- Understand the JAAS authentication details that will be provided to you.
- Secure server-side services (REST services and EJBs) using JAAS annotations.

CDI

- Understand contextual scopes.

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX188K - Red Hat Certified Specialist in Containers](#)

EX188V4K

- Implement images using Podman
 - Understand and use FROM (the concept of a base image) instruction.
 - Understand and use RUN instruction.



- Understand and use ADD instruction.
 - Understand and use COPY instruction.
 - Understand the difference between ADD and COPY instructions.
 - Understand and use WORKDIR and USER instructions.
 - Understand security-related topics.
 - Understand the differences and applicability of CMD vs. ENTRYPOINT instructions.
 - Understand ENTRYPOINT instruction with param.
 - Understand when and how to expose ports from a Containerfile.
 - Understand and use environment variables inside images.
 - Understand ENV instruction.
 - Understand container volume.
 - Mount a host directory as a data volume.
 - Understand security and permissions requirements related to this approach.
 - Understand the lifecycle and cleanup requirements of this approach.
- Manage images
 - Understand private registry security.
 - Interact with many different registries.
 - Understand and use image tags.
 - Push and pull images from and to registries.
 - Back up an image with its layers and meta data vs. backup a container state.
- Run containers locally using Podman
 - Run containers locally using Podman
 - Get container logs.
 - Listen to container events on the container host.
 - Use Podman inspect.
 - Specifying environment parameters.
 - Expose public applications.
 - Get application logs.
 - Inspect running applications.
- Run multi-container applications with Podman
 - Create application stacks



- Understand container dependencies
 - Working with environment variables
 - Working with secrets
 - Working with volumes
 - Working with configuration
- Troubleshoot containerized applications
 - Understand the description of application resources
 - Get application logs
 - Inspect running applications
 - Connecting to running containers

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

During the exam you may be required to work with one or more pre-written applications. You will not be required to modify application code however in some cases you may need to utilize supplied documentation in order to produce a new deployment of a given application.

[EX200K - Red Hat Certified System Administrator](#)

EX200V8K - Retired

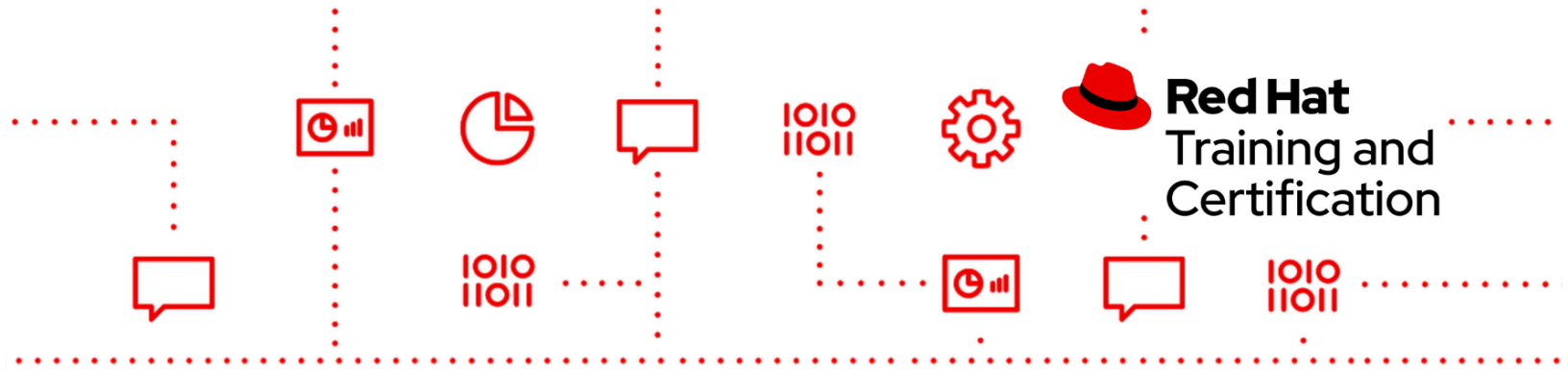
- Understand and use essential tools
 - Access a shell prompt and issue commands with correct syntax
 - Use input-output redirection (>, >>, |, 2>, etc.)
 - Use grep and regular expressions to analyze text
 - Access remote systems using SSH
 - Log in and switch users in multi-user targets
 - Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2
 - Create and edit text files
 - Create, delete, copy, and move files and directories
 - Create hard and soft links
 - List, set, and change standard ugo/rwx permissions



- Locate, read, and use system documentation including man, info, and files in /usr/share/doc
- Create simple shell scripts
 - Conditionally execute code (use of: if, test, [], etc.)
 - Use Looping constructs (for, etc.) to process file, command line input
 - Process script inputs (\$1, \$2, etc.)
 - Processing output of shell commands within a script
 - Processing shell command exit codes
- Operate running systems
 - Boot, reboot, and shut down a system normally
 - Boot systems into different targets manually
 - Interrupt the boot process in order to gain access to a system
 - Identify CPU/memory intensive processes and kill processes
 - Adjust process scheduling
 - Manage tuning profiles
 - Locate and interpret system log files and journals
 - Preserve system journals
 - Start, stop, and check the status of network services
 - Securely transfer files between systems
- Configure local storage
 - List, create, and delete partitions on MBR and GPT disks
 - Create and remove physical volumes
 - Assign physical volumes to volume groups
 - Create and delete logical volumes
 - Configure systems to mount file systems at boot by universally unique ID (UUID) or label
 - Add new partitions and logical volumes, and swap to a system non-destructively
- Create and configure file systems
 - Create, mount, unmount, and use vfat, ext4, and xfs file systems
 - Mount and unmount network file systems using NFS
 - Extend existing logical volumes
 - Create and configure set-GID directories for collaboration



- Configure disk compression
 - Manage layered storage
 - Diagnose and correct file permission problems
- Deploy, configure, and maintain systems
 - Schedule tasks using at and cron
 - Start and stop services and configure services to start automatically at boot
 - Configure systems to boot into a specific target automatically
 - Configure time service clients
 - Install and update software packages from Red Hat Network, a remote repository, or from the local file system
 - Work with package module streams
 - Modify the system bootloader
- Manage basic networking
 - Configure IPv4 and IPv6 addresses
 - Configure hostname resolution
 - Configure network services to start automatically at boot
 - Restrict network access using firewall-cmd/firewalld
- Manage users and groups
 - Create, delete, and modify local user accounts
 - Change passwords and adjust password aging for local user accounts
 - Create, delete, and modify local groups and group memberships
 - Configure superuser access
- Manage security
 - Configure firewall settings using firewall-cmd/firewalld
 - Create and use file access control lists
 - Configure key-based authentication for SSH
 - Set enforcing and permissive modes for SELinux
 - List and identify SELinux file and process context
 - Restore default file contexts
 - Manage SELinux port labels
 - Use boolean settings to modify system SELinux settings
 - Diagnose and address routine SELinux policy violations



As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX200V9K

- Understand and use essential tools
 - Access a shell prompt and issue commands with correct syntax
 - Use input-output redirection (>, >>, |, 2>, etc.)
 - Use grep and regular expressions to analyze text
 - Access remote systems using SSH
 - Log in and switch users in multi-user targets
 - Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2
 - Create and edit text files
 - Create, delete, copy, and move files and directories
 - Create hard and soft links
 - List, set, and change standard ugo/rwx permissions
 - Locate, read, and use system documentation including man, info, and files in /usr/share/doc
- Create simple shell scripts
 - Conditionally execute code (use of: if, test, [], etc.)
 - Use Looping constructs (for, etc.) to process file, command line input
 - Process script inputs (\$1, \$2, etc.)
 - Processing output of shell commands within a script
- Operate running systems
 - Boot, reboot, and shut down a system normally
 - Boot systems into different targets manually
 - Interrupt the boot process in order to gain access to a system



- Identify CPU/memory intensive processes and kill processes
- Adjust process scheduling
- Manage tuning profiles
- Locate and interpret system log files and journals
- Preserve system journals
- Start, stop, and check the status of network services
- Securely transfer files between systems
- Configure local storage
 - List, create, and delete partitions on GPT disks
 - Create and remove physical volumes
 - Assign physical volumes to volume groups
 - Create and delete logical volumes
 - Configure systems to mount file systems at boot by universally unique ID (UUID) or label
 - Add new partitions and logical volumes, and swap to a system non-destructively
- Create and configure file systems
 - Create, mount, unmount, and use vfat, ext4, and xfs file systems
 - Mount and unmount network file systems using NFS
 - Configure autofs
 - Extend existing logical volumes
 - Create and configure set-GID directories for collaboration
 - Diagnose and correct file permission problems
- Deploy, configure, and maintain systems
 - Schedule tasks using at and cron
 - Start and stop services and configure services to start automatically at boot
 - Configure systems to boot into a specific target automatically



- Configure time service clients
 - Install and update software packages from Red Hat Content Delivery Network, a remote repository, or from the local file system
 - Modify the system bootloader
- Manage basic networking
 - Configure IPv4 and IPv6 addresses
 - Configure hostname resolution
 - Configure network services to start automatically at boot
 - Restrict network access using firewall-cmd/firewalld
- Manage users and groups
 - Create, delete, and modify local user accounts
 - Change passwords and adjust password aging for local user accounts
 - Create, delete, and modify local groups and group memberships
 - Configure privileged access
- Manage security
 - Configure firewall settings using firewall-cmd/firewalld
 - Manage default file permissions
 - Configure key-based authentication for SSH
 - Set enforcing and permissive modes for SELinux
 - List and identify SELinux file and process context
 - Restore default file contexts
 - Manage SELinux port labels
 - Use boolean settings to modify system SELinux settings
 - Diagnose and address routine SELinux policy violations
- Manage containers
 - Find and retrieve container images from a remote registry



- Inspect container images
- Perform container management using commands such as podman and skopeo
- Build a container from a Containerfile
- Perform basic container management such as running, starting, stopping, and listing running containers
- Run a service inside a container
- Configure a container to start automatically as a systemd service
- Attach persistent storage to a container

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX200V93K

- Understand and use essential tools
 - Access a shell prompt and issue commands with correct syntax
 - Use input-output redirection (>, >>, |, 2>, etc.)
 - Use grep and regular expressions to analyze text
 - Access remote systems using SSH
 - Log in and switch users in multi-user targets
 - Archive, compress, unpack, and uncompress files using tar, gzip, and bzip2
 - Create and edit text files
 - Create, delete, copy, and move files and directories
 - Create hard and soft links
 - List, set, and change standard ugo/rwx permissions
 - Locate, read, and use system documentation including man, info, and files in /usr/share/doc
- Create simple shell scripts
 - Conditionally execute code (use of: if, test, [], etc.)



- Use Looping constructs (for, etc.) to process file, command line input
- Process script inputs (\$1, \$2, etc.)
- Processing output of shell commands within a script
- Operate running systems
 - Boot, reboot, and shut down a system normally
 - Boot systems into different targets manually
 - Interrupt the boot process in order to gain access to a system
 - Identify CPU/memory intensive processes and kill processes
 - Adjust process scheduling
 - Manage tuning profiles
 - Locate and interpret system log files and journals
 - Preserve system journals
 - Start, stop, and check the status of network services
 - Securely transfer files between systems
- Configure local storage
 - List, create, and delete partitions on GPT disks
 - Create and remove physical volumes
 - Assign physical volumes to volume groups
 - Create and delete logical volumes
 - Configure systems to mount file systems at boot by universally unique ID (UUID) or label
 - Add new partitions and logical volumes, and swap to a system non-destructively
- Create and configure file systems
 - Create, mount, unmount, and use vfat, ext4, and xfs file systems
 - Mount and unmount network file systems using NFS
 - Configure autofs
 - Extend existing logical volumes
 - Create and configure set-GID directories for collaboration
 - Diagnose and correct file permission problems



Red Hat Training and Certification

- Deploy, configure, and maintain systems
 - Schedule tasks using at and cron
 - Start and stop services and configure services to start automatically at boot
 - Configure systems to boot into a specific target automatically
 - Configure time service clients
 - Install and update software packages from Red Hat Content Delivery Network, a remote repository, or from the local file system
 - Modify the system bootloader
- Manage basic networking
 - Configure IPv4 and IPv6 addresses
 - Configure hostname resolution
 - Configure network services to start automatically at boot
 - Restrict network access using firewall-cmd/firewalld
- Manage users and groups
 - Create, delete, and modify local user accounts
 - Change passwords and adjust password aging for local user accounts
 - Create, delete, and modify local groups and group memberships
 - Configure privileged access
- Manage security
 - Configure firewall settings using firewall-cmd/firewalld
 - Manage default file permissions
 - Configure key-based authentication for SSH
 - Set enforcing and permissive modes for SELinux
 - List and identify SELinux file and process context
 - Restore default file contexts
 - Manage SELinux port labels
 - Use boolean settings to modify system SELinux settings
 - Diagnose and address routine SELinux policy violations



- Manage containers
 - Find and retrieve container images from a remote registry
 - Inspect container images
 - Perform container management using commands such as podman and skopeo
 - Perform basic container management such as running, starting, stopping, and listing running containers
 - Run a service inside a container
 - Configure a container to start automatically as a systemd service
 - Attach persistent storage to a container

[EX210K - Red Hat Certified System Administrator in Red Hat OpenStack](#)

EX210V13K - Retired

- Understand and work with director-based deployments
 - Use identity environment files to connect to the undercloud
 - Use identity environment files to connect to the overcloud
 - Use template files, environment files, and other resources to obtain information about an OpenStack environment
 - Work with containerized services
- Configure OpenStack domains
 - Create projects
 - Create groups
 - Create users
 - Manage quotas
- Create resources
 - Create virtual machine flavors
 - Add existing images to an overcloud
 - Create security groups
 - Create key pairs
- Configure networking
 - Create and assign networks to projects
 - Configure network routers
 - Configure software-defined networks
 - Work with Open Virtual Networks
- Configure floating IP addresses



- Configure an instance to use a floating IP address
 - Configure a service to be accessible via a floating IP address
- Manage block storage
 - Create a block storage volume
 - Attach block storage volumes to an instance
 - Snapshot a storage volume
- Work with Red Hat® Ceph Storage
 - Configure Ceph storage
 - Monitor Ceph Storage
 - Diagnose and troubleshoot Ceph Storage issues
- Work with object storage
 - Create a Swift container
 - Utilize a Swift container
- Work with shared storage
 - Create shared file systems
 - Configure instances to use shared file systems
- Manage instances
 - Launch instances
 - Associate instances with specified projects and networks
 - Use key pairs to connect to instances
 - Configure an instance during deployment
- Create a Heat stack
 - Create a Heat template
 - Diagnose and correct a broken Heat template
 - Launch a Heat stack
- Work with images
 - Modify an existing image
 - Create and associate flavors to customized images
 - Launch an instance from a customized image
 - Launch an instance on a second compute node
- Work with OpenStack services
 - Manage Identity Service tokens
 - Enable tracing in RabbitMQ
 - Display statistics using Ceilometer

EX210V16K

- Manage the Red Hat OpenStack Platform control plane



Red Hat Training and Certification

- Manage control plane services
 - Backup and restore control plane
 - Start and stop an overcloud
- Manage infrastructure security
 - Manage end-to-end secure services
 - Manage file-based component security with AIDE
- Manage user security
 - Manage an integrated IdM back-end configuration
 - Manage scoped service access
 - Manage project organization
 - Create a Red Hat OpenStack Platform domain for a client organization
 - Maintain token keys
 - Customize user roles
- Manage application deployment resources
 - Create images and flavors
 - Create and customize images
 - Initialize an instance during deployment
- Manage storage in Red Hat OpenStack Platform
 - Manage a Red Hat OpenStack Platform-dedicated Ceph storage cluster
 - Implement storage choices in Red Hat OpenStack Platform
 - Manage a Ceph storage cluster
 - Configure storage infrastructure
 - Manage Swift storage
- Manage networking
 - Configure Open Virtual Networking (OVN) services
 - Create and manage shared networks
- Manage compute node operations
 - Administer compute nodes
 - Manage compute resource capacity
 - Manage hyperconverged resource capacity
 - Re-balance compute node workloads
- Monitor operations



Red Hat Training and Certification

- Implement the Service Telemetry Framework
- Understand the flow of Red Hat OpenStack Platform services and components logging
- Automate cloud application deployment
 - Manage mass-scale application deployment
 - Write heat orchestration templates
 - Deploy applications using Ansible
 - Create a load-balanced application stack
- Troubleshoot operations
 - Diagnose issues
 - Troubleshoot common core issues
 - Trace heat stack events and troubleshoot failures

EX221K - Red Hat Certified Specialist in Cloud-native Integration

EX221V710K

Create and maintain Camel routes

- Use the Java™ language
- Use the CamelContext XML

Exchange messages

- Body
- In/out
- Headers
- Attachments

Consume and produce files

- CSV format
- JSON format
- XML format



Work with databases and the Camel JPA component

Understand transactional routes

Use EIPs

- Content-based routing
- Wire tap
- Splitter
- Aggregator
- Recipient list

Use Camel components

- File2
- FTP
- JPA2
- JMS
- Direct
- Rest

Use Camel Test

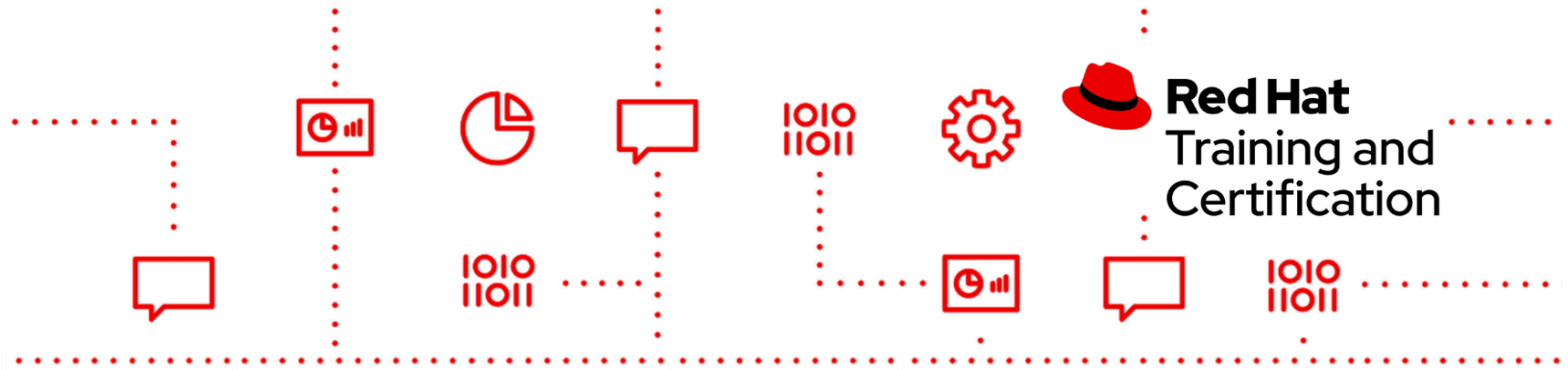
- Mock endpoints
- JUnit 4 integration and Camel Test APIs

Dynamically route messages

- Use conditional routes
- Use data-driven routes

Handle exceptions

- Catch and handle exceptions
- Use the dead-letter queue



As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX229K - Red Hat Certified Specialist in ROSA](#)

EX229V414K

- Deploy applications from external registries
 - Configure ROSA projects with Quay.io
 - Configure ROSA projects with Amazon Elastic Container Registry
- Provide Amazon storage volumes for applications
 - Configure storage classes for different Elastic Block Storage volume types.
 - Configure ROSA to use Elastic Filesystem Storage shares
- Configure application access to AWS services
 - Understand IAM Roles for Service Accounts (IRSA)
 - Provision AWS infrastructure and services using AWS Controllers for Kubernetes
- Configure observability for OpenShift on AWS
 - Configure structured application logging
 - Query application logs from Cloudwatch
 - Configure user workload monitoring with AWS Managed Service for Prometheus
- Configure node and pod autoscaling
 - Create node pools
 - Schedule pods to node pools
 - Configure workload autoscaling
 - Configure cluster node autoscaling

[EX240K - Red Hat Certified Specialist in API Management](#)

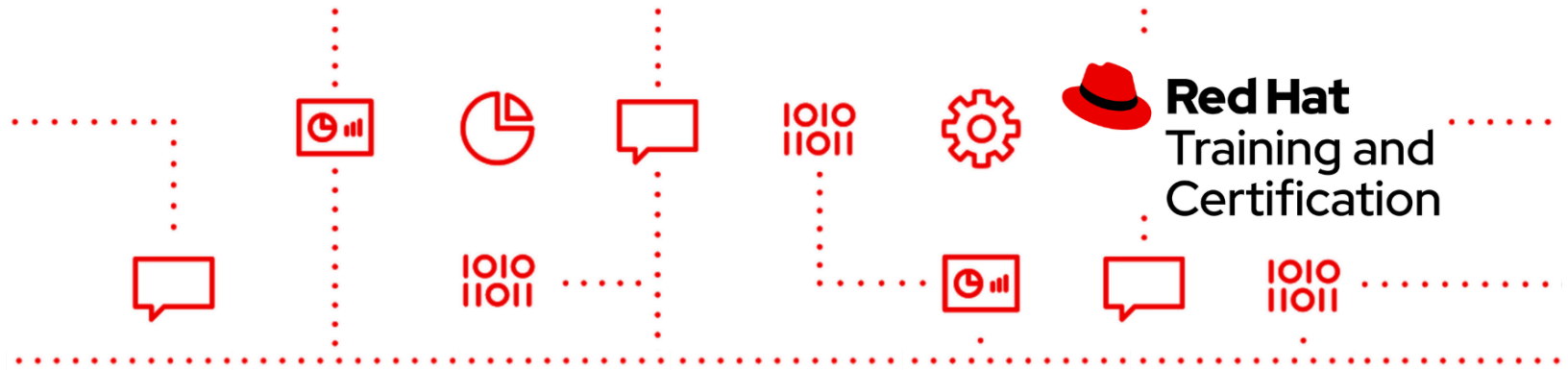
EX240V24K - Retired

- Access control
 - Create and maintain application plans
 - Configure limits and pricing rules
- Accounts
 - Create and update user accounts



Red Hat Training and Certification

- Configure user credentials
 - Send user invitations
 - Maintain group memberships
 - Configure usage rules and information gathered from users
 - Maintain application states and plans
- Analytics
 - Monitor usage, averages, and top applications
 - Analyze alerts, traffic, and integration errors
- API authentication
 - Maintain API credentials
 - Configure access tokens
 - Provide access control using OpenID Connect, API key, and/or App_ID, App_Key Pair
 - Configure SSO integration
- API documentation
 - Configure ActiveDocs using OpenAPI standards
- Billing
 - Configure payment gateway
 - Configure credit card Policies
 - Configure billing periods
 - Update users billing state
 - Configure billing details and invoice items
 - Issue, charge, and cancel invoices
- Developer portal
 - Customize developer portal
 - Configure legal terms
 - Configure portal settings
- Service discovery
 - Add services from an underlying OpenShift cluster
- Service definition
 - Configure methods and metrics
- Service integration



- Configure API gateway to point to private API
- Promote API gateway to staging and production
- Customize service authentication settings and service response
- Create and maintain custom service policies

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX240V211K

- Access control
 - Create and maintain application plans
 - Configure limits and pricing rules
- Accounts
 - Create and update user accounts
 - Configure user credentials
 - Send user invitations
 - Maintain group memberships
 - Configure usage rules and information gathered from users
 - Maintain application states and plans
- Analytics
 - Monitor usage, averages, and top applications
 - Analyze alerts, traffic, and integration errors
- API authentication
 - Maintain API credentials
 - Configure access tokens
 - Provide access control using OpenID Connect, API key, and/or App_ID, App_Key Pair
 - Configure SSO integration
- API documentation
 - Configure ActiveDocs using OpenAPI standards
- Billing



- Configure payment gateway
 - Configure credit card Policies
 - Configure billing periods
 - Update users billing state
 - Configure billing details and invoice items
 - Issue, charge, and cancel invoices
- Developer portal
 - Customize developer portal
 - Configure legal terms
 - Configure portal settings
- Service discovery
 - Add services from an underlying OpenShift cluster
- Service definition
 - Configure methods and metrics
- Service integration
 - Configure API gateway to point to private API
 - Promote API gateway to staging and production
 - Customize service authentication settings and service response
 - Create and maintain custom service policies

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX248K - Red Hat Certified Specialist in Enterprise Application Server Administration](#)

EX248V70K

Install and manage JBoss EAP

- Install JBoss EAP to a specific location on a system
- Install additional, operating system specific native libraries to enhance JBoss EAP performance
- Configure minimal security requirements for accessing and managing JBoss EAP
- Access and manage JBoss EAP using the provided tools



- Configure and start a multi-node, multi-server JBoss EAP domain spanning at least two hosts with multiple servers per host

Configure JBoss EAP domains, hosts and servers

- Create and remove JBoss EAP domains, hosts and servers
- Start, monitor and stop individual JBoss EAP domains, hosts and servers
- Configure Java™ memory usage at the host, server group and server level

Configure JBoss EAP to support clustered and HA operation

- Configure persistent network bindings for JBoss EAP services (both addresses and ports)
- Configure high-availability clustering using either TCP unicast or UDP multicast networking
- Secure the communications channels between clustered nodes
- Configure an Apache based load balancer for handling HTTP session fail over in a HA environment

Monitoring and managing JBoss EAP

- Create and restore configuration snapshots
- Configure JBoss EAP logging
- Configure and secure JBoss EAP JMX interface for external monitoring

Configuring Java Messaging Service

- Create and configure JMS topics and queues
- Secure access to JMS destinations

Manage applications



- Select appropriate JBoss EAP server profiles based on application requirements
- Configure DataSources (both XA and non-XA compliant)
- Deploy and undeploy applications
- Deploy and undeploy additional libraries and drivers
- Deploy a web application to the root context

Configure JBoss Web Connectors

- Tune and configure JBoss EAP web connector properties as requested
- Configure Apache integration with JBoss EAP
- Configure an SSL encrypted connection

Configure JBoss EAP security

- Create, modify, and use security domains
- Connect JBoss EAP to specified external security sources such as LDAP and DBMS
- Secure access to JBoss EAP services

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX248V74K

Install and manage JBoss EAP

- Install JBoss EAP to a specific location on a system
- Configure minimal security requirements for accessing and managing JBoss EAP
- Access and manage JBoss EAP using the provided tools
- Configure and start a multi-node, multi-server JBoss EAP domain spanning at least two hosts with multiple servers per host

Configure JBoss EAP domains, hosts and servers

- Create and remove JBoss EAP domains, hosts and servers
- Start, monitor and stop individual JBoss EAP domains, hosts and servers
- Configure Java™ memory usage at the host, server group and server level



Configure JBoss EAP to support clustered and HA operation

- Configure persistent network bindings for JBoss EAP services (both addresses and ports)
- Configure high-availability clustering using TCP unicast networking
- Secure the communications channels between clustered nodes

Monitoring and managing JBoss EAP

- Create and restore configuration snapshots
- Configure JBoss EAP logging

Configuring Java Messaging Service

- Remove, create and configure JMS topics and queues
- Secure access to JMS destinations

Manage applications

- Select appropriate JBoss EAP server profiles based on application requirements
- Configure DataSources (both XA and non-XA compliant)
- Deploy and undeploy applications
- Deploy and undeploy additional libraries and drivers
- Deploy a web application to the root context

Configure JBoss Web Connectors

- Tune and configure JBoss EAP web properties as requested
- Configure an SSL encrypted connection

Configure JBoss EAP legacy security subsystem

- Create, modify, and use security domains
- Connect JBoss EAP to specified external security sources such as LDAP and DBMS
- Secure access to JBoss EAP services

[EX260K - Red Hat Certified Specialist in Ceph Cloud Storage](#)

EX260V42K

- Install Red Hat Ceph Storage server



- Install a containerized Red Hat Ceph Storage server on both physical and virtual systems
 - Utilize and modify Red Hat Ansible Automation Platform installation files provided with Red Hat Ceph Storage to configure and install Red Hat Ceph Storage server
- Work with existing Red Hat Ceph Storage server appliances
 - Be able to change a Red Hat Ceph Storage server configuration
 - Add monitor (MON) nodes and object storage device (OSD) nodes
- Configure Red Hat Ceph Storage server
 - Configure a replicated storage pool
 - Store objects in storage pool
 - Store objects within a namespace within a storage pool
 - Create and configure erasure-coded pools
 - Create an erasure-coded pool profile with specified parameters
 - Upload a file to an erasure-coded pool
 - Change default settings in the Red Hat Ceph Storage configuration files
 - Manage Red Hat Ceph Storage authentication
 - Create a Red Hat Ceph Storage client with restricted read or write access to MONs, OSDs, pools, and namespaces
 - Managing OSDs Using Ceph-volume
 - Configure placement group auto-scaling
- Provide block storage with RBD
 - Create a RADOS block device image
 - Obtain information about a RADOS block device image
 - Map a RADOS block device image on a server
 - Use a RADOS block device image
 - Create an RBD snapshot
 - Create an RBD clone
 - Configure RBD mirrors
 - Deploy a RBD mirror agent
 - Configure one-way RBD mirroring in pool mode
 - Configure one-way RBD mirroring in image mode
 - Check the status of the mirroring process



- Import and export RBD images
 - Export a RADOS block device to an image file
 - Create an incremental RBD image file
 - Import a full RBD image file
 - Import a full RBD image file updated with an incremental RBD image file
- Provide object storage with RADOSGW
 - Deploy a RADOS gateway
 - Deploy a multisite RADOS gateway
 - Provide object storage using the Amazon S3 API
 - Be able to create a RADOSGW user that will use the S3 client commands
 - Be able to upload and download objects to a RADOSGW using the S3 client commands
 - Export S3 objects using NFS
 - Provide object storage for Swift
 - Be able to create a RADOSGW user that will use the Swift interface
 - Be able to upload or download objects to a RADOSGW using Swift commands
 - Configure Ceph Object Gateway for In-Transit Encryption
- Provide file storage with CephFS
 - Create a Red Hat Ceph Storage file system
 - Mount a Red Hat Ceph Storage file system on a client node persistently
 - Configure CephFS quotas
 - Create a CephFS snapshot
- Configure a CRUSH map
 - Be able to create a bucket hierarchy in a CRUSH map that can be used in an erasure profile or a replicant rule
 - Be able to remap a PG
 - Be able to remap all PG's in a pool for an optimal redistribution
- Manage and update cluster maps
 - Manage MON and OSD maps
 - Be able to monitor and change OSD storage limits for monitoring available space on an OSD
- Manage a Red Hat Ceph Storage cluster



- Determine the general status of a Red Hat Ceph Storage cluster
 - Troubleshoot problems with OSDs and MONs
- Tune Red Hat Ceph Storage
 - Specify and tune key network tuning parameters for a Red Hat Ceph Storage cluster
 - Control and manage scrubbing and deep scrubbing
 - Control and manage recovery and rebalancing processes
 - Control and manage RAM utilization against I/O performance
- Troubleshoot Red Hat Ceph Storage server problems
 - Troubleshoot client issues
 - Enable debugging mode on RADOS gateway
 - Optimize RBD client access using key tuning parameters
- Integrate Red Hat Ceph Storage with Red Hat OpenStack Platform
 - Integrate Red Hat Ceph Storage using both Glance and Cinder
 - Modify key Glance configuration files to use Red Hat Ceph Storage
 - Configure Glance to use Red Hat Ceph Storage as a backend to store images in the Red Hat Ceph Storage cluster
 - Modify key Cinder configuration files to use Red Hat Ceph Storage
 - Configure Cinder to use Red Hat Ceph Storage RBDs for block storage backing volumes

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX260V50K

- Install Red Hat Ceph Storage server
 - Install a containerized Red Hat Ceph Storage server on both physical and virtual systems
 - Utilize and modify Red Hat Ansible Automation Platform installation files provided with Red Hat Ceph Storage to configure and install Red Hat Ceph Storage server
- Work with existing Red Hat Ceph Storage server appliances



- Be able to change a Red Hat Ceph Storage server configuration
 - Add monitor (MON) nodes and object storage device (OSD) nodes
- Configure Red Hat Ceph Storage server
 - Configure a replicated storage pool
 - Store objects in storage pool
 - Store objects within a namespace within a storage pool
 - Create and configure erasure-coded pools
 - Create an erasure-coded pool profile with specified parameters
 - Upload a file to an erasure-coded pool
 - Change default setting in the Red Hat Ceph Storage configuration files
 - Manage Red Hat Ceph Storage authentication
 - Create a Red Hat Ceph Storage client with restricted read or write access to MONs, OSDs, pools, and namespaces
 - Managing OSDs using Ceph-volume
 - Configure placement group auto-scaling
- Provide block storage with RBD
 - Create a RADOS block device image
 - Obtain information about a RADOS block device image
 - Map a RADOS block device image on a server
 - Use a RADOS block device image
 - Create an RBD snapshot
 - Create an RBD clone
 - Configure RBD mirrors
 - Deploy a RBD mirror agent
 - Configure one-way RBD mirroring in pool mode
 - Configure one-way RBD mirroring in image mode
 - Check the status of the mirroring process
 - Import and export RBD images
 - Export a RADOS block device to an image file
 - Create an incremental RBD image file
 - Import a full RBD image file
 - Import a full RBD image file updated with an incremental RBD image file



- Provide object storage with RADOSGW
 - Deploy a RADOS gateway
 - Deploy a multisite RADOS gateway
 - Provide object storage using the Amazon S3 API
 - Be able to create a RADOSGW user that will use the S3 client commands
 - Be able to upload and download objects to a RADOSGW using the S3 client commands
 - Export S3 objects using NFS
 - Provide object storage for Swift
 - Be able to create a RADOSGW user that will use the Swift interface
 - Be able to upload or download objects to a RADOSGW using Swift commands
 - Configure Ceph Object Gateway for In-Transit Encryption
- Provide file storage with CephFS
 - Create a Red Hat Ceph Storage file system
 - Mount a Red Hat Ceph Storage file system on a client node persistently
 - Configure CephFS quotas
 - Create a CephFS snapshot
- Configure a CRUSH map
 - Be able to create a bucket hierarchy in a CRUSH map that can be used in an erasure profile or a replicant rule
 - Be able to remap a PG
 - Be able to remap all PGs in a pool for an optimal redistribution
- Manage and update cluster maps
 - Manage MON and OSD maps
 - Be able to monitor and change OSD storage limits for monitoring available space on an OSD
- Manage a Red Hat Ceph Storage cluster
 - Determine the general status of a Red Hat Ceph Storage cluster
 - Troubleshoot problems with OSDs and MONs
- Tune Red Hat Ceph Storage
 - Specify and tune key network tuning parameters for a Red Hat Ceph Storage cluster



Red Hat Training and Certification

- Control and manage scrubbing and deep scrubbing
 - Control and manage recovery and rebalancing processes
 - Control and manage RAM utilization against I/O performance
- Troubleshoot Red Hat Ceph Storage server problems
 - Troubleshoot client issues
 - Enable debugging mode on RADOS gateway
 - Optimize RBD client access using key tuning parameters
- Integrate Red Hat Ceph Storage with Red Hat OpenStack Platform
 - Integrate Red Hat Ceph Storage using both Glance and Cinder
 - Modify key Glance configuration files to use Red Hat Ceph Storage
 - Configure Glance to use Red Hat Ceph Storage as a backend to store images in the Red Hat Ceph Storage cluster
 - Modify key Cinder configuration files to use Red Hat Ceph Storage
 - Configure Cinder to use Red Hat Ceph Storage RBDs for block storage backing volumes

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX267K - Red Hat Certified Specialist in OpenShift AI](#)

EX267V28K

- Install Red Hat OpenShift AI (RHOAI)
- Configure and manage RHOAI
 - Manage user and group permissions and resources
 - Manage DataScienceCluster object
 - Create and publish custom notebook images
 - Import custom notebook images
 - Manage idle notebook culling
 - Customize default workbench and model server sizes
- Work with data science projects
 - Create, modify, and delete data science projects



Red Hat Training and Certification

- Manage data science project permissions
- Use data science workbenches
 - Understand Jupyter ecosystem
 - Create, modify, and delete workbenches
 - Start and stop workbenches
 - Manage data connections
 - Manage Persistent Volume Claim objects
 - Inspect workbench resources
- Use Git to manage Jupyter notebooks collaboratively
 - Upload an existing notebook from a Git repository
 - Push updated notebooks to a Git repository
- Work with machine learning models
 - Understand basic machine learning concepts
 - Train models in Python using popular foundational libraries
 - Load data in a scalable way
 - Monitor and evaluate the training process
- Save and load models
 - Save, export, and share models
 - Deploy models as Python applications
 - Create a custom runtime in KServe
 - Deploy a model using ModelMesh
- Create data science pipelines
 - Create pipelines with Elyra
 - Create pipelines with KubeFlow

[EX280K - Red Hat Certified Specialist in OpenShift Administration](#)

EX280V410K, EX280V45K, EX280V46K

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

- Manage OpenShift Container Platform
 - Use the command-line interface to manage and configure an OpenShift cluster
 - Use the web console to manage and configure an OpenShift cluster



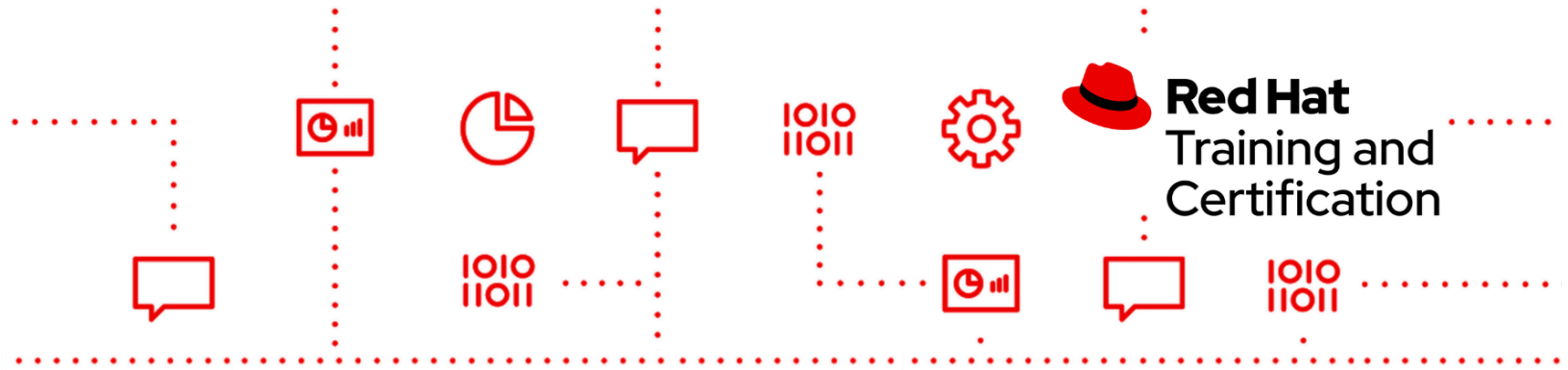
- Create and delete projects
 - Import, export, and configure Kubernetes resources
 - Examine resources and cluster status
 - View logs
 - Monitor cluster events and alerts
 - Troubleshoot common cluster events and alerts
 - Use product documentation
- Manage users and policies
 - Configure the HTTPBasic identity provider for authentication
 - Create and delete users
 - Modify user passwords
 - Modify user and group permissions
 - Create and manage groups
- Control access to resources
 - Define role-based access controls
 - Apply permissions to users
 - Create and apply secrets to manage sensitive information
 - Create service accounts and apply permissions using security context constraints
- Configure networking components
 - Troubleshoot software defined networking
 - Create and edit external routes
 - Control cluster network ingress
 - Create a self signed certificate
 - Secure routes using TLS certificates
- Configure pod scheduling
 - Limit resource usage
 - Scale applications to meet increased demand
 - Control pod placement across cluster nodes

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

- Manage OpenShift Container Platform
 - Use the web console to manage and configure an OpenShift cluster
 - Use the command-line interface to manage and configure an OpenShift cluster
 - Create and delete projects
 - Locate and examine container images
 - Query, format, and filter attributes of Kubernetes resources
 - Import, export, and configure Kubernetes resources
 - Examine resources and cluster status
 - Monitor cluster events and alerts
 - View logs
 - Assess the health of an OpenShift cluster
 - Troubleshoot common container, pod, and cluster events and alerts
 - Use product documentation
- Deploy applications
 - Deploy applications from resource manifests
 - Use Kustomize overlays to modify application configurations
 - Deploy applications from images, OpenShift templates, and Helm charts
 - Deploy jobs to perform one-time tasks
 - Manage application deployments
 - Work with replica sets
 - Work with labels and selectors
 - Configure services
 - Expose both HTTP and non-HTTP applications to external access
 - Work with operators such as MetalLB and Multus
- Manage storage for application configuration and data
 - Create and use secrets
 - Create and use configuration maps
 - Provision persistent storage volumes for block and file-based data
 - Use storage classes
 - Manage non-shared storage with stateful sets



- Configure applications for reliability
 - Configure and use health probes
 - Reserve and limit application compute capacity
 - Scale applications to meet increased demand
- Manage application updates
 - Identify images using tags and digests
 - Roll back failed deployments
 - Manage image streams
 - Use triggers to manage images
- Manage authentication and authorization
 - Configure the HTPasswd identity provider for authentication
 - Create and delete users
 - Modify user passwords
 - Create and manage groups
 - Modify user and group permissions
- Configure network security
 - Configure networking components
 - Troubleshoot software defined networking
 - Create and edit external routes
 - Control cluster network ingress
 - Secure external and internal traffic using TLS certificates
 - Configure application network policies
- Enable developer self-service
 - Configure cluster resource quotas
 - Configure project quotas
 - Configure project resource requirements
 - Configure project limit ranges
 - Configure project templates
- Manage OpenShift operators
 - Install an operator
 - Uninstall an operator
 - Delete an operator
- Configure application security



- Configure and manage service accounts
 - Run privileged applications
 - Manage and apply permissions using security context constraints
 - Create and apply secrets to manage sensitive information
 - Configure application access to Kubernetes APIs
 - Configure Kubernetes cron jobs
- Update OpenShift
 - Update an OpenShift cluster
 - Identify deprecated Kubernetes API usage
 - Update OpenShift operators

[EX288K - Red Hat Certified Specialist in OpenShift Application Development](#)

EX288V46K - Retired

- Work with Red Hat OpenShift Container Platform
 - Create and work with multiple OpenShift projects
 - Deploy single container and multi-container applications
 - Create and use a Helm chart
 - Customize deployments with Kustomize
 - Implement application health monitoring
 - Understand basic Git usage and work with Git in the context of deploying applications in OpenShift
 - Configure the OpenShift internal registry to meet specific requirements
 - Create and deploy cloud-native applications for OpenShift
 - Manage applications with the web console
- Deploy multi-container applications
 - Create Helm charts
 - Customizing deployments with Kustomize
- Work with container images
 - Use command line utilities to create and manipulate container images
 - Optimize container images
 - Publish container images to a registry



- Troubleshoot application deployment issues
 - Diagnose and correct minor issues with application deployment
- Work with image streams
 - Create custom image streams to deploy applications
 - Pull applications from existing Git repositories
 - Debug minor issues with application deployment
- Work with configuration maps
 - Create configuration maps
 - Use configuration maps to inject data into applications
- Work with the source-to-image (S2I) tool
 - Deploy applications using S2I
 - Customize existing S2I builder images
- Work with hooks and triggers
 - Create a hook that runs a provided script
 - Test and confirm proper operation of the hook
 - Manage and trigger application builds
- Work with templates
 - Create an OpenShift template
 - Use pre-existing templates written in either JSON or YAML format
 - Work with multi-container templates
 - Add custom parameters to a template

During the exam you may be required to work with one or more pre-written applications. You will not be required to modify application code however in some cases you may need to utilize supplied documentation in order to fully test and complete the deployment of a given application.

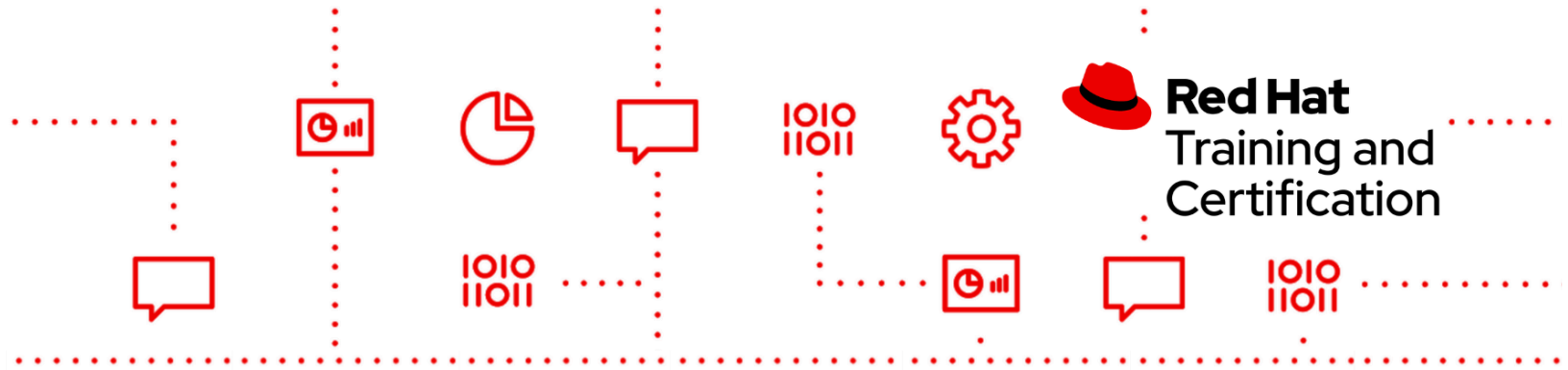
EX288V410K

As part of this exam, you should be able to perform these tasks:

- Work with Red Hat OpenShift Container Platform
 - Create and work with multiple OpenShift projects



- Deploy single container and multi-container applications
 - Create and use a Helm chart
 - Customize deployments with Kustomize
 - Implement application health monitoring
 - Understand basic Git usage and work with Git in the context of deploying applications in OpenShift
 - Configure the OpenShift internal registry to meet specific requirements
 - Create and deploy cloud-native applications for OpenShift
 - Manage applications with the web console
- Deploy multi-container applications
 - Create Helm charts
 - Customizing deployments with Kustomize
- Work with container images
 - Use command line utilities to create and manipulate container images
 - Optimize container images
 - Publish container images to a registry
- Troubleshoot application deployment issues
 - Diagnose and correct minor issues with application deployment
- Work with image streams
 - Create custom image streams to deploy applications
 - Pull applications from existing Git repositories
 - Debug minor issues with application deployment
- Work with configuration maps
 - Create configuration maps
 - Use configuration maps to inject data into applications
- Work with the source-to-image (S2I) tool
 - Deploy applications using S2I
 - Customize existing S2I builder images
- Work with hooks and triggers
 - Create a hook that runs a provided script
 - Test and confirm proper operation of the hook
 - Manage and trigger application builds



- Work with templates
 - Create an OpenShift template
 - Use pre-existing templates written in either JSON or YAML format
 - Work with multi-container templates
 - Add custom parameters to a template

During the exam you may be required to work with one or more pre-written applications. You will not be required to modify application code however in some cases you may need to utilize supplied documentation in order to fully test and complete the deployment of a given application.

EX288V412K and EX288V414K

As part of this exam, you should be able to perform these tasks:

- Work with Red Hat OpenShift Container Platform
 - Create and work with multiple OpenShift projects
 - Create and deploy single container and multi-container applications
 - Use application health monitoring
 - Understand basic Git usage and work with Git in the context of deploying applications in OpenShift
 - Configure the OpenShift internal registry to meet specific requirements
 - Manage applications with the web console
- Deploy multi-container applications
 - Create and use Helm charts
 - Customizing deployments with Kustomize
- Work with container images
 - Use command line utilities to create and manipulate container images
 - Optimize container images
 - Publish container images to a registry
- Troubleshoot application builds and deployment issues
 - Diagnose and correct minor issues with application deployment



- Diagnose and correct minor issues with build process
- Work with image streams
 - Create custom image streams to deploy applications
 - Pull applications from existing Git repositories
 - Trigger updates on image stream changes
 - Debug minor issues with application deployment
- Work with configuration maps and secrets
 - Create configuration maps
 - Create secret resources
 - Use configuration maps to inject data into applications
- Work with the source-to-image (S2I) framework
 - Build and deploy applications using S2I
 - Customize existing S2I builder images
- Work with build hooks and build triggers
 - Create a build hook that runs a provided script
 - Test and confirm proper operation of the hook
 - Manage and trigger application builds
- Work with templates
 - Create an OpenShift template
 - Use pre-existing templates written in either JSON or YAML format
 - Work with multi-container templates
 - Add custom parameters to a template
- Work with OpenShift Pipelines
 - Understanding CI/CD process inside of OpenShift
 - Understand and work with standard Tekton custom resource definitions (CRDs) for defining CI/CD pipelines
 - Design and define, and troubleshoot CI/CD workflows for applications
 - Configure and trigger Pipeline workflows for applications
- Work with Operators accessible for users to run in their applications
 - Creating applications from installed Operators

During the exam you may be required to work with one or more pre-written applications. You will not be required to modify application code however in some cases you may need to utilize supplied documentation in order to fully test and complete the deployment of a given application.

EX288V414K

- Work with Red Hat OpenShift Container Platform
 - Create and work with multiple OpenShift projects
 - Create and deploy single container and multi-container applications
 - Use application health monitoring
 - Understand basic Git usage and work with Git in the context of deploying applications in OpenShift
 - Configure the OpenShift internal registry to meet specific requirements
 - Manage applications with the web console
- Deploy multi-container applications
 - Create and use Helm charts
 - Customizing deployments with Kustomize
- Work with container images in OpenShift Container Platform
 - Understand how to create container images based on pre-built images
 - Understand and work with image builds and image build configurations
 - Work with custom builder workflows to create images to use with OpenShift Container Platform
 - Publish container images to a the OpenShift image registry
- Troubleshoot application builds and deployment issues
 - Diagnose and correct minor issues with application deployment
 - Diagnose and correct minor issues with build process
- Work with image streams
 - Create custom image streams to deploy applications
 - Pull applications from existing Git repositories
 - Trigger updates on image stream changes
 - Debug minor issues with application deployment



Red Hat Training and Certification

- Work with configuration maps and secrets
 - Create configuration maps
 - Create secret resources
 - Use configuration maps to inject data into applications
- Work with the source-to-image (S2I) framework
 - Build and deploy applications using S2I framework
 - Customize existing S2I builder images
- Work with build hooks and build triggers
 - Create a build hook that runs a provided script
 - Test and confirm proper operation of the hook
 - Manage and trigger application builds
- Work with templates
 - Create an OpenShift template
 - Use pre-existing templates written in either JSON or YAML format
 - Work with multi-container templates
 - Add custom parameters to a template
- Work with OpenShift Pipelines
 - Understanding CI/CD process inside of OpenShift
 - Understand and work with standard Tekton custom resource definitions (CRDs) for defining CI/CD pipelines
 - Design and define, and troubleshoot CI/CD workflows for applications
 - Configure and trigger Pipeline workflows for applications
- Work with Operators accessible for users to run in their applications
 - Creating applications from installed Operators

[EX294K - Red Hat Certified Engineer](#)

EX294V8K - Retired

- Be able to perform all tasks expected of a Red Hat Certified System Administrator
 - Understand and use essential tools



- Operate running systems
 - Configure local storage
 - Create and configure file systems
 - Deploy, configure, and maintain systems
 - Manage users and groups
 - Manage security
- Understand core components of Ansible
 - Inventories
 - Modules
 - Variables
 - Facts
 - Plays
 - Playbooks
 - Configuration files
- Install and configure an Ansible control node
 - Install required packages
 - Create a static host inventory file
 - Create a configuration file
- Configure Ansible managed nodes
 - Create and distribute SSH keys to managed nodes
 - Configure privilege escalation on managed nodes
 - Validate a working configuration using ad hoc Ansible commands
- Script administration tasks
 - Create simple shell scripts
 - Create simple shell scripts that run ad hoc Ansible commands
- Create and use static inventories to define groups of hosts
- Create Ansible plays and playbooks
 - Know how to work with commonly used Ansible modules
 - Use variables to retrieve the results of running a command
 - Use conditionals to control play execution
 - Configure error handling
 - Create playbooks to configure systems to a specified state



- Use Ansible modules for system administration tasks that work with:
 - Software packages and repositories
 - Services
 - Firewall rules
 - File systems
 - Storage devices
 - File content
 - Archiving
 - Scheduled tasks
 - Security
 - Users and groups
- Create and use templates to create customized configuration files
- Work with Ansible variables and facts
- Create and work with roles
- Download roles from an Ansible Galaxy and use them
- Manage parallelism
- Use Ansible Vault in playbooks to protect sensitive data
- Use provided documentation to look up specific information about Ansible modules and commands

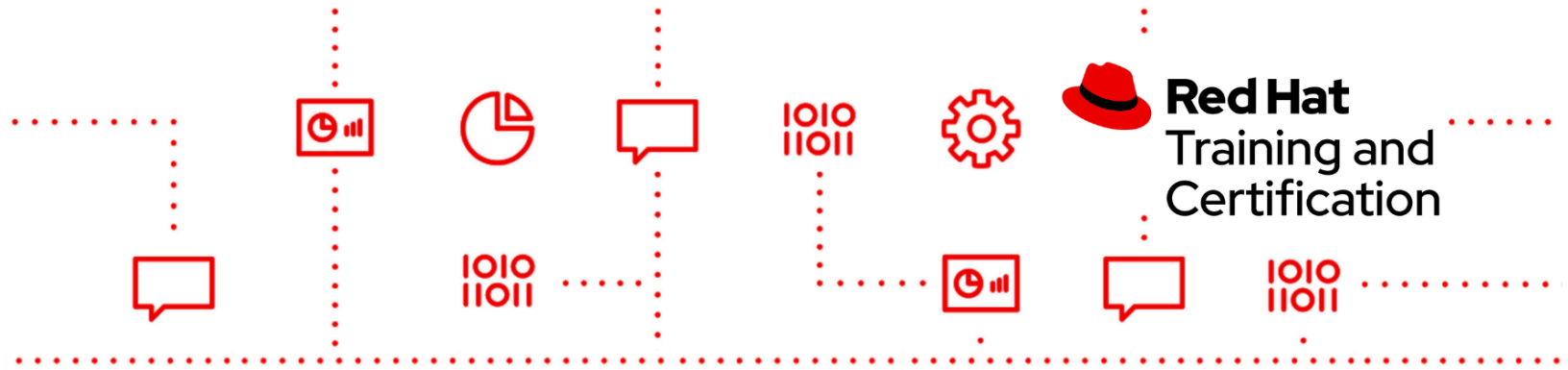
As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX294V83K - Retired

- Be able to perform all tasks expected of a Red Hat Certified System Administrator
 - Understand and use essential tools
 - Operate running systems
 - Configure local storage
 - Create and configure file systems
 - Deploy, configure, and maintain systems
 - Manage users and groups



- Manage security
- Understand core components of Ansible
 - Inventories
 - Modules
 - Variables
 - Facts
 - Plays
 - Playbooks
 - Configuration files
- Install and configure an Ansible control node
 - Install required packages
 - Create a static host inventory file
 - Create a configuration file
- Configure Ansible managed nodes
 - Create and distribute SSH keys to managed nodes
 - Configure privilege escalation on managed nodes
 - Validate a working configuration using ad hoc Ansible commands
- Script administration tasks
 - Create simple shell scripts
 - Create simple shell scripts that run ad hoc Ansible commands
- Create and use static inventories to define groups of hosts
- Create Ansible plays and playbooks
 - Know how to work with commonly used Ansible modules
 - Use variables to retrieve the results of running a command
 - Use conditionals to control play execution
 - Configure error handling
 - Create playbooks to configure systems to a specified state
- Use Ansible modules for system administration tasks that work with:
 - Software packages and repositories
 - Services
 - Firewall rules
 - File systems



- Storage devices
- File content
- Archiving
- Scheduled tasks
- Security
- Users and groups
- Create and use templates to create customized configuration files
- Work with Ansible variables and facts
- Create and work with roles
- Download roles from an Ansible Galaxy and use them
- Use Ansible Vault in playbooks to protect sensitive data
- Use provided documentation to look up specific information about Ansible modules and commands

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX294V84K

- Be able to perform all tasks expected of a Red Hat Certified System Administrator
 - Understand and use essential tools
 - Operate running systems
 - Configure local storage
 - Create and configure file systems
 - Deploy, configure, and maintain systems
 - Manage users and groups
 - Manage security
- Understand core components of Ansible
 - Inventories
 - Modules
 - Variables
 - Facts
 - Plays



- Playbooks
 - Configuration files
- Install and configure an Ansible control node
 - Install required packages
 - Create a static host inventory file
 - Create a configuration file
- Configure Ansible managed nodes
 - Create and distribute SSH keys to managed nodes
 - Configure privilege escalation on managed nodes
 - Validate a working configuration using ad hoc Ansible commands
- Script administration tasks
 - Create simple shell scripts
 - Create simple shell scripts that run ad hoc Ansible commands
- Create and use static inventories to define groups of hosts
- Create Ansible plays and playbooks
 - Know how to work with commonly used Ansible modules
 - Use variables to retrieve the results of running a command
 - Use conditionals to control play execution
 - Configure error handling
 - Create playbooks to configure systems to a specified state
- Use Ansible modules for system administration tasks that work with:
 - Software packages and repositories
 - Services
 - Firewall rules
 - File systems
 - Storage devices
 - File content
 - Archiving
 - Scheduled tasks
 - Security
 - Users and groups
- Create and use templates to create customized configuration files



Red Hat Training and Certification

- Work with Ansible variables and facts
- Create and work with roles
- Download roles from an Ansible Galaxy and use them
- Use Ansible Vault in playbooks to protect sensitive data
- Use provided documentation to look up specific information about Ansible modules and commands

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX294V9K

- Be able to perform all tasks expected of a Red Hat Certified System Administrator
 - Understand and use essential tools
 - Operate running systems
 - Configure local storage
 - Create and configure file systems
 - Deploy, configure, and maintain systems
 - Manage users and groups
 - Manage security
- Understand core components of Ansible
 - Inventories
 - Modules
 - Variables
 - Facts
 - Loops
 - Conditional tasks
 - Plays
 - Handling task failure
 - Playbooks
 - Configuration files
 - Roles



- Use provided documentation to look up specific information about Ansible modules and commands
- Use roles and Ansible Content Collections
 - Create and work with roles
 - Install roles and use them in playbooks
 - Install Content Collections and use them in playbooks
 - Obtain a set of related roles, supplementary modules, and other content from content collections, and use them in a playbook.
- Install and configure an Ansible control node
 - Install required packages
 - Create a static host inventory file
 - Create a configuration file
 - Create and use static inventories to define groups of hosts
- Configure Ansible managed nodes
 - Create and distribute SSH keys to managed nodes
 - Configure privilege escalation on managed nodes
 - Deploy files to managed nodes
 - Be able to analyze simple shell scripts and convert them to playbooks
- Run playbooks with Automation content navigator
 - Know how to run playbooks with Automation content navigator
 - Use Automation content navigator to find new modules in available Ansible Content Collections and use them
 - Use Automation content navigator to create inventories and configure the Ansible environment
- Create Ansible plays and playbooks
 - Know how to work with commonly used Ansible modules
 - Use variables to retrieve the results of running a command
 - Use conditionals to control play execution
 - Configure error handling
 - Create playbooks to configure systems to a specified state
- Automate standard RHCSA tasks using Ansible modules that work with:
 - Software packages and repositories



- Services
- Firewall rules
- File systems
- Storage devices
- File content
- Archiving
- Task scheduling
- Security
- Users and groups
- Manage content
 - Create and use templates to create customized configuration files
 - Use Ansible Vault in playbooks to protect sensitive data

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX316K - Red Hat Certified Specialist in OpenShift Virtualization Exam](#)

EX316V410K

Candidates for this exam should be able to perform the following tasks:

- Deploy OpenShift Virtualization Operator
 - Understand how Red Hat OpenShift Virtualization components work
 - Deploy Red Hat OpenShift Virtualization Operator using Red Hat OpenShift Container Platform Operator Lifecycle Management
- Run and access virtual machines
 - Understand how to provision virtual machines using the Red Hat OpenShift Virtualization operator
 - Manage virtual machines from the web interface or the command line interface
 - Allow different users to access different virtual machine resources
- Configure Kubernetes networking for virtual machines
 - Understand virtual machine communication on the Kubernetes SDN
 - Create ClusterIP services and network policies for virtual machines



- Configure external access for virtual machines
- Connect virtual machines to external networks
 - Understand the Multus CNI plug-in and use cases
 - Configure multihomed nodes and virtual machines using the NMstate operator and Multus
- Configure Kubernetes storage for virtual machines
 - Configure appropriate services for persistent virtual machine storage
 - Attach disks to a virtual machine
 - Detach disks from a virtual machine
 - Connect virtual machines to external storage using Multus
- Manage virtual machine templates
 - Use and modify preconfigured templates to provision virtual machines.
 - Generate custom templates to provision virtual machines
 - Manage cloud-init in templates adding access credentials, repositories and running arbitrary commands
- Manage virtual machine snapshots
 - Understand how to create and manage virtual machine snapshots.
 - Create a snapshot of a VM
 - Restore a VM from a snapshot
- Clone virtual machines
 - Prepare virtual machines for cloning
 - Clone virtual machines using OpenShift web console
 - Clone virtual machines' disks using data volumes
- Perform virtual machine live migrations
 - Understand limitations of virtual machine live migration
 - Configure virtual machines to have affinity for certain nodes
 - Understand how to start, monitor and cancel virtual machine migrations
- Perform node maintenance and OpenShift Virtualization updates
 - Configure node maintenance resources
 - Prepare nodes for maintenance using the CLI
 - Drain the nodes correctly in order to be set in maintenance mode
- Manage virtual machine load balancing with Kubernetes networking resources



- Configure virtual machine load balancing with Kubernetes networking resources
- Create and configure NodePort type services
- Create customized routes to expose generic services
- Configure health probes for virtual machines
 - Configure and update the run strategy parameters
 - Configure health probes and watchdog devices to monitor health and responsiveness of a virtual machine and its services
- Prepare virtual machines for node failure
 - Determine the eviction strategies used to survive cluster node loss
 - Understand how to use node selectors, affinity, anti-affinity, tolerations and taints.
 - Use watchdog monitoring and health checks for virtual machine applications and nodes.

[EX318K - Red Hat Certified Specialist in Virtualization](#)

EX318V43K

Install Red Hat Virtualization Manager (RHVM) and Red Hat Virtualization Hypervisor (RHHV)

- Install RHVM and any dependencies
- Configure RHVM to authenticate against an external IPA service
- Install Red Hat Virtualization Hypervisor
- Add one or more Red Hat Virtualization Hypervisors to Red Hat Virtualization Manager
- Create and manage datacenters and clusters

Manage storage

- Configure NFS services
- Configure storage domains
- Import installation media into a storage domain

Create and manage virtual machines

- Install virtual machines



- Access virtual machine consoles with SPICE and VNC
- Start and stop virtual machines
- Edit virtual machine hardware characteristics
- Configure automatic migration for virtual machines

Work with virtual machine images

- Create snapshots
- Create virtual machines from snapshots
- Import existing virtual machine images into RHVM

Create and manage RHVM users

- Create internal users
- Create external users
- Configure roles and assign users to those roles
- Configure access using roles

Automatically deploy virtual machines

- Create virtual machine templates
- Deploy virtual machines using templates
- Configure virtual machines using cloud-init

Create and manage networking objects

- Create logical networks
- Assign logical networks to hosts
- Create MAC address pools
- Assign MAC address pools to clusters

Create and manage cluster policies

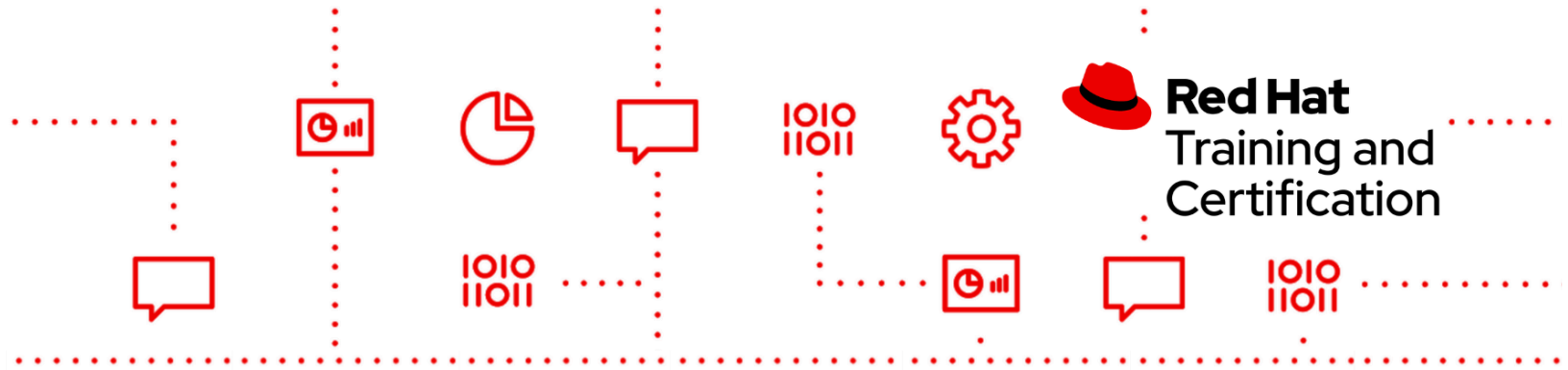
- Create scheduling and migration policies
- Assign policies to clusters

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX328K - Red Hat Certified Specialist in Building Resilient Microservices](#)

EX328V20K

- Understand and work with Red Hat OpenShift Service Mesh Custom Resources
- Deploy and configure applications on Service Mesh:
 - Install sidecar manually in pod applications
 - Automatic sidecar injection using annotations
 - Understand the configuration of network policies (Mesh members, external services, etc.)
- Work with request routing and traffic management.
 - Be able to configure static and dynamic request routing to different versions of an application
- Understand the deployment/release pattern strategies that Red Hat OpenShift Service Mesh® can help with, providing more complex operational functionality, including A/B testing and canary releases
- Configure and manage advanced routing techniques to control the flow and API calls between services
 - Traffic shifting migration within the mesh producing between different services A/B testing, canary rollouts, and staged rollouts with percentage-based traffic splits, and dark launches for a selective group of users
 - Take profit of the traffic mirroring capabilities to bring changes to the microservices. Be able to produce shadowing launches copying the live traffic
- Send the inbound and outbound traffic from and to the mesh, managing the ingress and egress traffic control policies
 - Define and control gateway entry points into the mess for the incoming traffic allowing pass requests through the services
 - Enable controlled access to external publicly accessible services from within the istio cluster
- Be able to configure the network resilience and the fault tolerance dynamically at runtime to ensure the failing nodes and prevent localized failures from cascading
 - Control the waiting time for replies defining timeouts



- Enhance service availability specifying the number of request attempts with retry strategies
- Limit for calls within a service and prevent access to an overloaded or failing host applying a circuit breaker mechanism
- Specify the connection and ejection pool policies configuring the load balancing destination rules
- Work and configure Service Mesh policy checks
 - Define enforcement features through policies, configure local and global rate limiting, and define access quotas
 - Enable and configure the authorization for denial and allow policies applied to a workload
- Understand and configure the workload-to-workload communication using the implemented architecture for authentication and authorization security in Service Mesh
 - Provide service-to-service communication with secure naming authorization
 - Tunnel the service-to-service communication using mutual TLS communication. Map the identity of the service name with secure naming
 - Define peer authentication policies to enforce the mutual TLS mode
 - Define the required end-user authentication policy check. Define and configure access authorization rules for service and end-user to workload communications
- Understand and work with the fault injection mechanisms to introduce errors and chaos testing into the system to test the failure recovery capacity of the applications
 - Inject timing failures producing delays to mimic increased network latency or overloaded services
 - Produce crash failures with error response injections and TCP connection failures

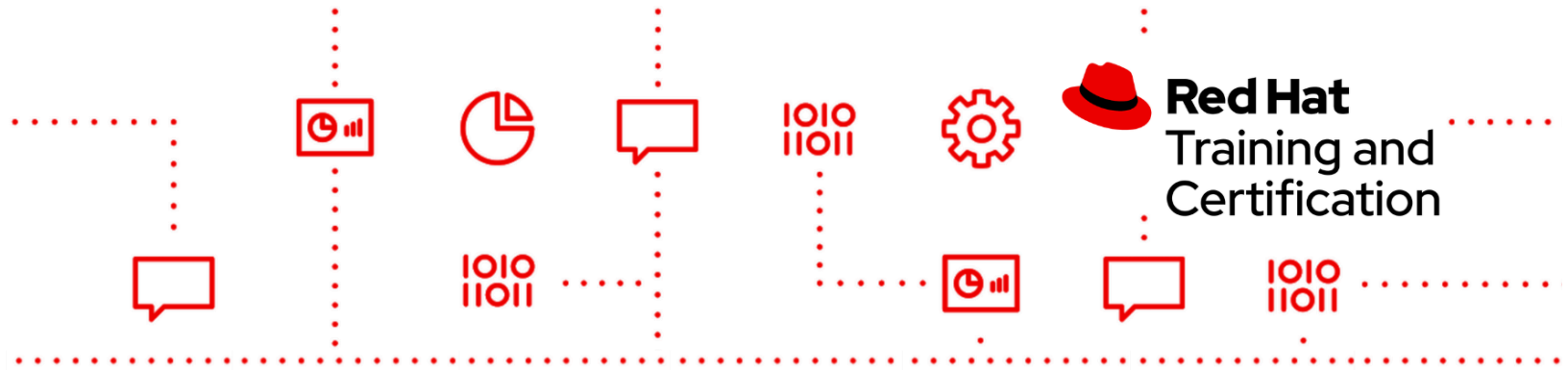
[EX342K - Red Hat Certified Specialist in Enterprise Linux Diagnostics and Troubleshooting](#)

EX342V72K - Retired

- Understand and employ general methods for troubleshooting
 - Collect system information to aid in troubleshooting
 - Consult documentation resources to aid in troubleshooting
 - Monitor systems for vital characteristics



- Configure systems to send log messages to a centralized host
- Diagnose and troubleshoot system start up issues
 - Identify and resolve service failures affecting boot
 - Regain root control of a system
 - Troubleshoot boot issues
 - Identify hardware and hardware problems
 - Manage kernel modules and their parameters
- Diagnose and troubleshoot file system issues
 - Recover corrupted file systems
 - Recover mis-configured or broken LVM configurations
 - Recover data from encrypted file systems
 - Identify and fix iSCSI issues
- Resolve package management issues
 - Resolve package management dependency issues
 - Recover a corrupted RPM database.
 - Identify and restore changed files
- Troubleshoot and fix network connectivity issues
 - Use standard tools to verify network connectivity
 - Identify and fix network connectivity issues
 - Inspect network traffic to aid troubleshooting
- Diagnose application issues
 - Identify library dependencies for third-party software
 - Identify if an application suffers from memory leaks
 - Use standard tools to debug an application
 - Identify and fix issues related to SELinux
- Identify and fix authentication issues
 - Identify and fix pluggable authentication module (PAM) issues
 - Identify and fix LDAP and Kerberos identity management issues
- Gather information to aid third party investigation of issues
 - Create kernel crashdumps



- Compile and execute SystemTap modules

As with all Red Hat performance-based exams, configurations must persist after restart without intervention.

EX342V84K

- Understand and employ general methods for troubleshooting
 - Collect system information to aid in troubleshooting
 - Consult documentation resources to aid in troubleshooting
 - Monitor systems for vital characteristics
 - Monitor systems with the RHEL Web console
 - Configure systems using Ansible
 - Configure systems to send log messages to a centralized host
- Diagnose and troubleshoot system startup issues
 - Identify and resolve service failures affecting boot
 - Regain root control of a system
 - Troubleshoot boot issues
 - Identify hardware and hardware problems
 - Manage kernel modules and their parameters
- Diagnose and troubleshoot file system issues
 - Recover corrupted file systems
 - Recover mis-configured or broken LVM configurations
 - Recover data from encrypted file systems
 - Identify and fix iSCSI issues
- Resolve package management issues
 - Resolve package management dependency issues
 - Recover a corrupted RPM database.
 - Identify and restore changed files
- Troubleshoot and fix network connectivity issues
 - Use standard tools to verify network connectivity



- Identify and fix network connectivity issues
- Inspect network traffic to aid troubleshooting
- Diagnose application issues
 - Identify library dependencies for third-party software
 - Identify if an application suffers from memory leaks
 - Use standard tools to debug an application
 - Identify and fix issues related to SELinux
 - Identify and fix issues related to containerized applications
- Identify and fix authentication issues
 - Identify and fix pluggable authentication module (PAM) issues
 - Identify and fix identity management issues
- Gather information to aid third party investigation of issues
 - Create kernel crashdumps
 - Compile and execute SystemTap modules

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX358K - Red Hat Certified Specialist in Services Management and Automation](#)

EX358V8K

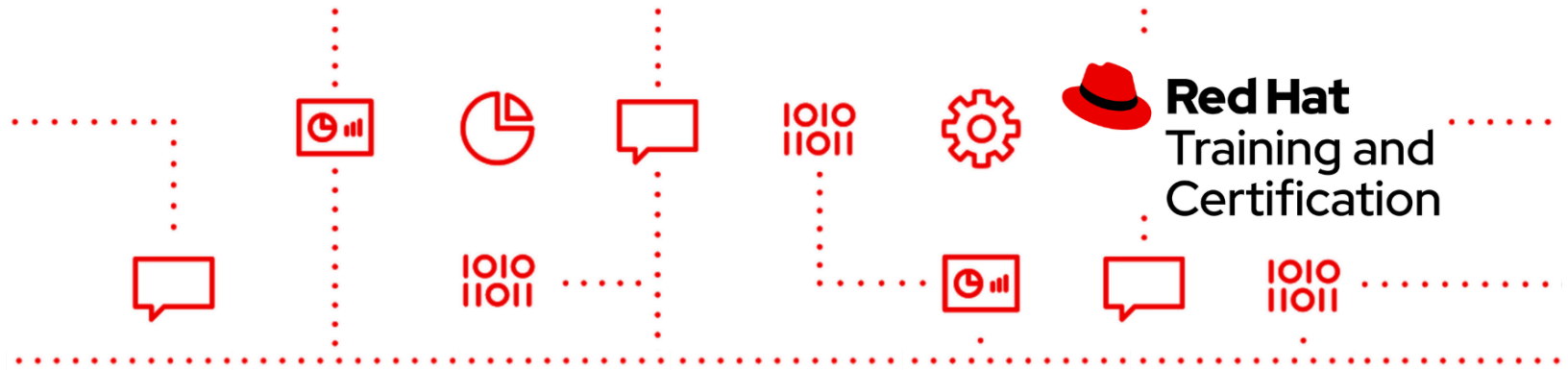
- Manage Network Services
 - Configure network clients to use either a dynamically or statically assigned address
 - Work with both IPv4 and IPv6
- Manage Firewall Services
 - Configure system firewalls to allow access to specific services or ports
 - Configure system firewalls to allow or deny access from only specific network domains or IP subnets
- Manage SELinux
 - Configure SELinux booleans for a given service
 - Configure a file or directory's SELinux context



- Manage system processes
 - Configure system processes to start on boot
 - Prevent a system process from starting
- Manage Link Aggregation
 - Create a network team interface consisting of two network interfaces
 - Make a network team interface persistent across boots
 - Assign a network address to a network team interface
 - Configure a teamd runner
- Manage DNS
 - Configure a caching nameserver
 - Configure an authoritative nameserver using a partially completed zone file
 - Configure forward and reverse lookups for both IPv4 and IPv6 addresses
- Manage DHCP
 - Configure address assignment within a specified address range
 - Configure a specific address assignment to a designated host
 - Configure address assignments for both IPv4 and IPv6
- Manage printers
 - Create and manage a printer queue for a network printer
 - Manage existing printer queues
- Manage Email services
 - Configure an email server to forward email to an outbound mail relay
 - Use mail clients to read or send email
- Manage a MariaDB database server
 - Install and configure a basic MariaDB service
 - Restrict access to a MariaDB server to specific network addresses
 - Create a MariaDB database
 - Manage MariaDB database users and access rights
 - Add records to an existing MariaDB database
 - Issue simple SQL queries against a MariaDB database
 - Create a MariaDB backup
 - Import a MariaDB database from a backup
- Manage HTTPD web access



- Install and configure Apache
 - Install and configure NGINX
 - Configure an alternate document root
 - Configure an alternate web access port
 - Configure name-based virtual hosts
 - Configure a secure web server (HTTPS)
 - Provide a static cache to speed up HTTP response time
 - Configure an HTTP HAProxy load balancer
 - Terminate HTTPS Connections
- Manage iSCSI
 - Provide and configure iSCSI targets
 - Configure iSCSI initiators to connect persistently to iSCSI targets
 - Restrict access to iSCSI services to specific clients and networks
- Manage NFS
 - Configure persistent NFS exports
 - Configure an NFS client to mount an NFS export persistently
 - Restrict access to NFS exports to specific clients and networks
- Manage SMB
 - Configure SMB shares
 - Create and manage SMB users
 - Create SMB only users
 - Restrict access to SMB shares
 - Mount a SMB share
 - Perform a multiuser SMB mount
- Use Ansible to Configure Standard Services
 - Create and modify playbooks
 - Understand and utilize inventory files
 - Use variables in playbooks
 - Work with RHEL System Roles



As with all Red Hat performance-based exams, configurations must persist after reboot without intervention. Red Hat reserves the right to add, modify, and remove objectives.

[EX362K - Red Hat Certified Specialist in Identity Management](#)

EX362V45K

- IdM Server Installation and Configuration
 - Install RedHat Identity Management (IdM) in a scalable, fault tolerant environment
 - Create Users, Groups and Policies
 - Implement a SSO
- IdM Client Installation and Configuration
 - Install and configure IdM clients
 - Configure Kerberized services
- IdM HA Configuration
 - Configure and Manage a Certificate Authority
 - Create Secret Vaults
- IdM Users and Policies management
 - Configure Policies and User Access
 - Configure roaming or automounted home directories
 - Configure IdM as LDAP backend for external services such as Red Hat Satellite Server or Ansible Tower
- Creating Trust with AD domain
 - Create trust relationships with Active Directory
 - Authenticate users within an Active Directory domain
- IdM maintenance
 - Back up an IdM infrastructure
 - Perform a backup without interruption of services

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX362V48K



Red Hat Training and Certification

- Install and configure Red Hat Identity Management (IdM)
 - Install IdM using either scripts or using Ansible Automation Platform
 - Install and configure a replica IdM server
 - Create users, groups and policies
- Implement Single Sign On (SSO)
 - Create an SSO client
 - Verify SSO client operation
- Install and configure an IdM Client
 - Install and configure IdM clients
 - Configure Kerberized services
- Manage the IdM integrated certificate authority
 - Configure and manage a certificate authority
 - Create secret vaults
- Create and configure IdM users and user policies
 - Configure policies and user access
 - Configure roaming or automounted home directories
 - Use Ansible Tower to configure and manage IdM users
 - Configure IdM as an LDAP backend for external services such as Red Hat Satellite Server or Ansible Automation Platform automation controller
- Maintain IdM services
 - Configure NTP on all IdM components
 - Configure firewall on all IdM components
 - Back up an IdM infrastructure
 - Perform a backup without interruption of services

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX364K - Red Hat Certified Specialist in Decision Management](#)

EX364V78K

Design business rules:

- Create and use an intermediate Drools model with associated facts.
- Organize rules and manage facts.



- Understand and use effective rule design patterns.

Business Rules Authoring:

- Basic rules
- Calendars
- Decision tables
- Rule templates

Designing a decision service using DMN models:

- Input, output definition
- Data types
- Decision design and implementation

Conflict resolution:

- Prioritize and order rule execution.
- Group rules appropriately to avoid conflicts.
- Use rule flow groups.

Session management:

- Build a common object.
- Understand stateless KIE sessions.
- Understand stateful KIE sessions.
- Add, remove, and update facts.

Test and maintain:

- Understand JUnit test for testing rules.
- Identify potential issues, including poorly designed rules.

Performance:

- Understand the impacts of different rule designs and implementations.
- Identify and avoid common, inefficient results, such as Cartesian products.

[EX370K - Red Hat Certified Specialist in OpenShift Data Foundation](#)

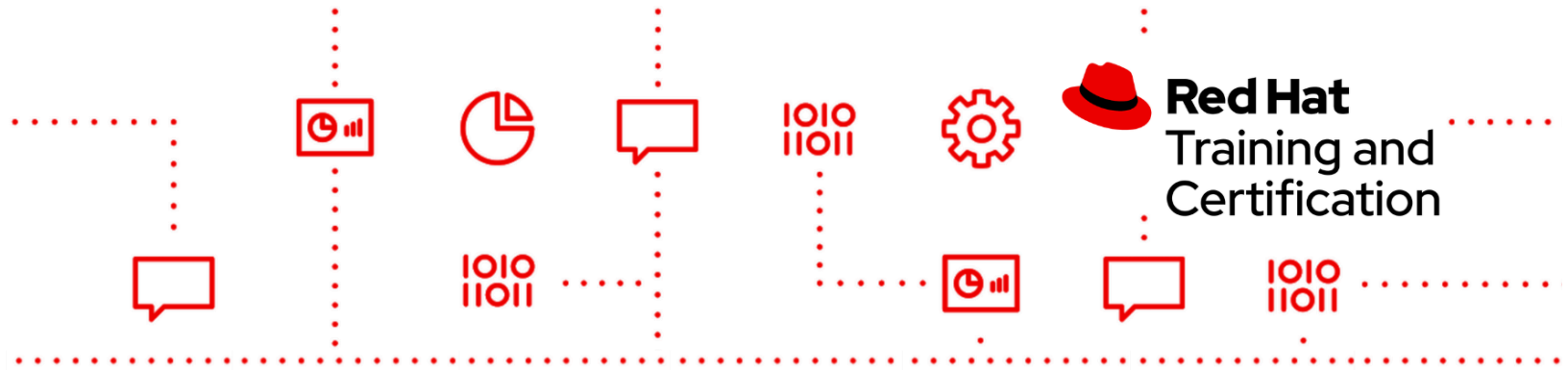
EX370V47K

- Manage OpenShift Container Platform
 - Use the command line interface to manage and configure an OpenShift Container Platform cluster
 - Use the web console to manage and configure an OpenShift Container Platform cluster
 - Create and delete projects
 - Create, edit, and delete ConfigMaps and secrets



Red Hat Training and Certification

- Import, export, and configure Kubernetes resources
- Examine resources and cluster status
- View logs
- Monitor cluster events and alerts
- Troubleshoot common cluster events and alerts
- Use product documentation
- Deploy OpenShift Data Foundation using internal mode
 - Deploy OpenShift Data Foundation from the web console
 - Deploy OpenShift Data Foundation from the command line
- Configure OpenShift cluster services to use Red Hat OpenShift Data Foundation
 - Configure the internal image registry to Use Red Hat OpenShift Data Foundation
 - Configure monitoring to use Red Hat OpenShift Data Foundation
- Deploy and manage applications and images
 - Deploy applications to Red Hat OpenShift Container Platform using the web console
 - Deploy applications to Red Hat OpenShift Container Platform using the command line
 - Configure applications to use ConfigMaps and secrets
 - Understand and use image tags
 - Create and edit external routes
- Troubleshoot applications in OpenShift
 - Understand the description of application resources.
 - View and use application logs
 - Inspect running applications
 - Connect to containers running in a pod
 - Copy resources to and from containers running in a pod
- Configure application workloads to use Red Hat OpenShift Data Foundation file and block storage
 - Identify Ceph components needed for a Red Hat OpenShift Data Foundation Implementation
 - Configure applications to use Red Hat OpenShift Data Foundation file storage
 - Configure applications to use Red Hat OpenShift Data Foundation block storage



- Configure custom storage classes
- Manage Red Hat OpenShift Data Foundation block and file storage capacity
 - Monitor Red Hat OpenShift Data Foundation cluster health
 - Configure storage quotas and permissions
 - Extend application storage for Red Hat OpenShift Data Foundation
 - Add disks to a Red Hat OpenShift Data Foundation Cluster
- Perform backup and restore of Kubernetes block and file volumes
 - Backup and restore Kubernetes applications
 - Create volume snapshots and clones
- Configure applications to use Red Hat OpenShift Data Foundation object storage
 - Create object bucket claims and access object storage
 - Configure applications to use Red Hat OpenShift Data Foundation object storage
 - Monitor Red Hat OpenShift Data Foundation object buckets
- Control access to resources
 - Define role-based access controls
 - Apply permissions to users
 - Create and apply secrets to manage sensitive information
 - Create service accounts and apply permissions using security context constraints
- Configure pod scheduling
 - Limit resource usage
 - Control pod placement across cluster nodes

[EX374K - Red Hat Certified Specialist in Developing Automation with Ansible Automation Platform](#)

EX374V20K, EX374V22K

- Understand and use Git
 - Clone a Git repository
 - Create, modify and push files in a Git repository
- Manage inventory variables
 - Structure host and group variables using multiple files per host or group
 - Use special variables to override the host, port, or remote user for a specific host



- Set up directories containing multiple host variable files for managed hosts
 - Override names used in inventory files with a different name or IP address
- Manage task execution
 - Control privilege execution
 - Run selected tasks from a playbook
- Transform data with filters and plugins
 - Populate variables with data from external sources using lookup plugins
 - Use lookup and query functions to incorporate data from external sources into playbooks and deployed template files
 - Implement loops using structures other than simple lists using lookup plugins and filters
 - Inspect, validate, and manipulate variables containing networking information with filters
- Delegate tasks
 - Run a task for a managed host on a different host
 - Control whether facts gathered by a task are delegated to the managed host or the controlling host
- Manage content collections
 - Create a content collection
 - Install a content collection
 - Publish a content collection
- Manage execution environments
 - Build an execution environment
 - Run playbooks in a execution environment
 - Upload execution environments into automation hub
 - Using execution environments in automation controller
- Manage inventories and credentials
 - Manage advanced inventories
 - Create a dynamic inventory from an identity management server or a database server
 - Create machine credentials to access inventory hosts
 - Create a source control credential



- Manage automation controller
 - Run playbooks in automation controller
 - Pull content into automation controller from either Git or automation hub
 - Pull an execution environment from automation hub and run a playbook in it.

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX378K - Red Hat Certified Cloud-native Developer](#)

EX378V17K - Retired

Provide and obtain configuration properties through several environment-aware sources made available through dependency injection or lookup

- Externalize data into configured values.
- Inject configured values into beans using the @Inject and the @ConfigProperty qualifier.
- Access or create a configuration.
- Understand default and custom ConfigSource and ConfigSource ordering.

Build fault-tolerant Quarkus-based microservices using Microprofile Fault Tolerance strategies

- Understand the relationship to MicroProfile Config.
- Understand async versus sync execution type.
- Use @Timeout.
- Understand Retry policies and apply using @Retry.
- Understand and define Fallback.
- Understand and apply CircuitBreaker.
- Understand and apply Bulkhead.
- Understand and set up fault tolerance configuration.

Probe the state of a Quarkus application from another machine using MicroProfile Health Check

- Understand and implement the HealthCheck interface.
- Understand and apply @Liveness and @Readiness annotation.
- Understand and implement HealthCheckResponse.
- Construct human-friendly HealthCheckResponse.



Export monitoring data to management agents from a running Quarkus application using Microprofile Metrics

- Understand and use three sets of sub-resource (scopes): Base, vendor, application.
- Understand tags (labels).
- Understand and use metadata.
- Understand metric registry and @Metric.
- Expose metrics via REST API.
- Know required metrics.
- Understand Quarkus application metrics programming model.

MicroProfile Interoperable JWT RBAC on Quarkus applications: OpenID Connect (OIDC)-based JSON Web Tokens(JWT) for role-based access control (RBAC) of microservice endpoints

- Understand token-based authentication.
- Use JWT bearer tokens to protect services.
- Mark a JAX-RS application as requiring MP-JWT access control.
- Map MP-JWT tokens to Java EE container APIs.

Implement a Quarkus application and expose REST service endpoints with JAX-RS

- Understand REST concepts, particularly the application and use of the HTTP PUT, DELETE, GET, and POST methods.
- Know and use standard HTTP return codes.
- Implement RESTful root resource class.
- Expose a REST service using JAX-RS.
- Understand and use @Path, @Produce, and @Consume.
- Using CDI to integrate components.
- Using bean validation to ensure data format and consistency.

Simplified JPA mapping with Panache

- Understand the difference between the Active Record Pattern and the Repository Pattern.



- Use basic JPA to create, read, update, and delete persistent objects and their relationships.
- Map a bi-directional one-to-many relationship between two entities, including both sides of the association.
- Demonstrate the ability to perform the most common Panache operations and add custom entity methods.

Microprofile OpenAPI specification to document RESTful APIs

- Understand OpenAPI documents and the Swagger UI to discover remote services APIs.
- Demonstrate the ability to link to semantic versioning (semver) remote service endpoints.
- Understand how to produce the default and custom OpenAPI document to JAX-RS endpoints.

Interacting with REST APIs in Quarkus using the MicroProfile REST Client

- Understand the type-safe approach to invoke RESTful services over HTTP using the JAX-RS APIs.
- Understand REST concepts, particularly the application and use of the HTTP PUT, DELETE, GET, and POST methods.
- Demonstrate the ability to create and use a REST client to connect with a remote service.
- Parameterize and configure the REST client URI to invoke a specific remote microservice.
- Understand and use special additional client headers.

As with all Red Hat performance-based exams, configurations must persist after restart without intervention.



Provide and obtain configuration properties through several environment-aware sources made available through dependency injection or lookup

- Externalize data into configured values.
- Inject configured values into beans using the @Inject and the @ConfigProperty qualifier.
- Access or create a configuration.
- Understand default and custom ConfigSource and ConfigSource ordering.

Build fault-tolerant Quarkus-based microservices using Microprofile Fault Tolerance strategies

- Understand the relationship to MicroProfile Config.
- Understand async versus sync execution type.
- Use @Timeout.
- Understand Retry policies and apply using @Retry.
- Understand and define Fallback.
- Understand and apply CircuitBreaker.
- Understand and apply Bulkhead.
- Understand and set up fault tolerance configuration.

Probe the state of a Quarkus application from another machine using MicroProfile Health Check

- Understand and implement the HealthCheck interface.
- Understand and apply @Liveness and @Readiness annotation.
- Understand and implement HealthCheckResponse.
- Construct human-friendly HealthCheckResponse.

Export monitoring data to management agents from a running Quarkus application using Microprofile Metrics

- Understand tags (labels).
- Understand and use metadata.



- Understand micrometer annotation and registry.
- Expose metrics via REST API.
- Know required metrics.
- Understand Quarkus application metrics programming model.

MicroProfile Interoperable JWT RBAC on Quarkus applications: OpenID Connect (OIDC)-based JSON Web Tokens(JWT) for role-based access control (RBAC) of microservice endpoints

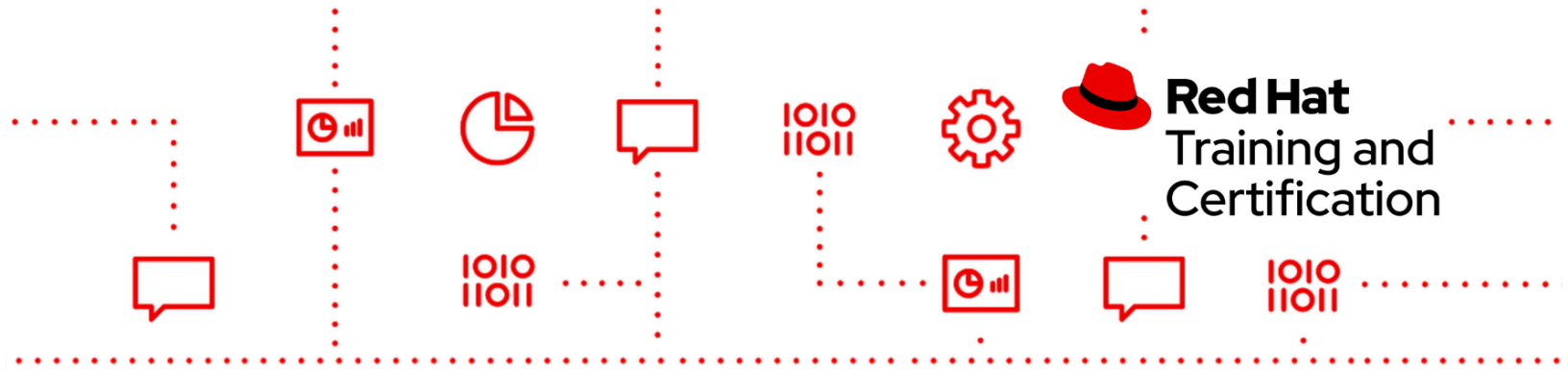
- Understand token-based authentication.
- Use JWT bearer tokens to protect services.
- Mark a JAX-RS application as requiring MP-JWT access control.
- Map MP-JWT tokens to Java EE container APIs.

Implement a Quarkus application and expose REST service endpoints with JAX-RS

- Understand REST concepts, particularly the application and use of the HTTP PUT, DELETE, GET, and POST methods.
- Know and use standard HTTP return codes.
- Implement RESTful root resource class.
- Expose a REST service using JAX-RS.
- Understand and use @Path, @Produce, and @Consume.
- Using CDI to integrate components.
- Using bean validation to ensure data format and consistency.

Simplified JPA mapping with Panache

- Understand the difference between the Active Record Pattern and the Repository Pattern.
- Use basic JPA to create, read, update, and delete persistent objects and their relationships.
- Map a bi-directional one-to-many relationship between two entities, including both sides of the association.



- Demonstrate the ability to perform the most common Panache operations and add custom entity methods.

Reactive messaging

- Understand reactive programming
- Understand messaging core concept
- Understand Channels
- Understand Incoming and Outgoing
- Understand message acknowledgment

Microprofile OpenAPI specification to document RESTful APIs

- Understand OpenAPI documents and the Swagger UI to discover remote services APIs.
- Demonstrate the ability to link to semantic versioning (semver) remote service endpoints.
- Understand how to produce the default and custom OpenAPI document to JAX-RS endpoints.

Interacting with REST APIs in Quarkus using the MicroProfile REST Client

- Understand the type-safe approach to invoke RESTful services over HTTP using the JAX-RS APIs.
- Understand REST concepts, particularly the application and use of the HTTP PUT, DELETE, GET, and POST methods.
- Demonstrate the ability to create and use a REST client to connect with a remote service.
- Parameterize and configure the REST client URI to invoke a specific remote microservice.
- Understand and use special additional client headers.



As with all Red Hat performance-based exams, configurations must persist after restart without intervention.

EX378V38K

Provide and obtain configuration properties through several environment-aware sources made available through dependency injection or lookup

- Externalize data into configured values
- Inject configured values into beans using the @Inject and the @ConfigProperty qualifier
- Demonstrate ability to create and access a configuration mapping to a Java Object
- Understand the default ConfigSource ordering and how to extend the configuration support with custom ConfigSource
- Understand how to set custom configuration properties per target environment with Profiles

Build fault-tolerant Quarkus-based microservices using Microprofile Fault Tolerance strategies

- Understand the relationship to MicroProfile Config
- Understand async versus sync execution type
- Understand resilience using Timeouts
- Understand Retry policies and apply using @Retry annotation
- Understand and define Fallback strategy
- Understand and apply CircuitBreaker
- Understand and apply Bulkhead
- Understand and set up fault tolerance configuration

Probe the state of a Quarkus application from another machine using MicroProfile Health Check

- Understand and implement the HealthCheck interface
- Understand and apply @Startup, @Liveness and @Readiness annotation
- Understand and implement Reactive health checks



- Understand and implement HealthCheckResponse
- Construct human-friendly HealthCheckResponse
- Understand and use the Health UI
- Understand and apply Health Groups and the @HealthGroup annotation
- Understand and apply the @Wellness annotation

Observe, collect and measure data from a running Quarkus application using Micrometer Metrics

- Understand and use dimensional data aggregation with tags
- Understand and instrument the code with timers, gauges, counters, distribution summaries, and long task timers
- Understand micrometer annotations and the registry API
- Understand and use the exposed metrics via REST API
- Understand Quarkus application metrics programming model
- Understand how to export monitoring data to management agents

MicroProfile Interoperable JWT RBAC on Quarkus applications: OpenID Connect (OIDC)-based JSON Web Tokens(JWT) for role-based access control (RBAC) of microservice endpoints

- Understand token-based authentication
- Use JWT bearer tokens to protect services
- Mark a JAX-RS application as requiring MP-JWT access control
- Map MP-JWT tokens to Java EE container APIs

Implement a Quarkus application and expose RESTful service endpoints with Resteasy Reactive and Jakarta REST (formerly known as JAX-RS)

- Understand RESTful concepts, particularly the application and use of the HTTP PUT, DELETE, GET, and POST methods
- Know and use standard HTTP return codes and return response body
- Understand how to write JSON REST services with Quarkus
- Implement RESTful root resource class
- Expose a RESTful service using Jakarta REST (formerly known as JAX-RS)



- Understand and apply declaring endpoints: URI mapping and HTTP representation and content types
- Understand and apply the async/reactive support to declare non-blocking endpoints
- Use CDI to integrate components
- Using bean validation to ensure data format and consistency

Simplified JPA mapping with Panache

- Understand the difference between the Active Record Pattern and the Repository Pattern
- Use basic JPA to create, read, update, and delete persistent objects and their relationships
- Map a bi-directional one-to-many relationship between two entities, including both sides of the association
- Demonstrate the ability to perform the most common Panache operations and add custom entity methods

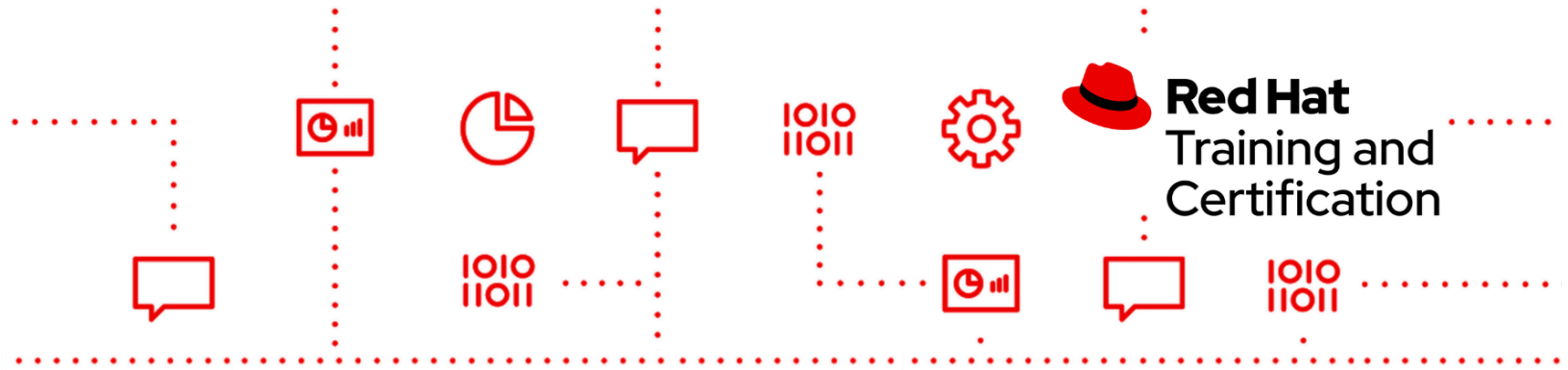
Reactive messaging on Quarkus applications

- Understand reactive programming
- Understand messaging core concept
- Understand Channels
- Understand Incoming and Outgoing
- Understand message acknowledgment

Microprofile OpenAPI specification to document RESTful APIs

- Understand OpenAPI documents and the Swagger UI to discover remote services APIs
- Demonstrate the ability to link to semantic versioning (semver) remote service endpoints
- Understand how to produce the default and custom OpenAPI document to JAX-RS endpoints

Interacting with REST APIs in Quarkus using the REST Client Reactive



- Understand the type-safe approach to invoke RESTful services over HTTP using the Jakarta REST And Microprofile annotations
- Understand REST concepts, particularly the application and use of the HTTP PUT, DELETE, GET, and POST methods
- Understand and use the non-blocking flavor of REST Client Reactive with async support
- Demonstrate the ability to create and use a REST client to connect with a remote service
- Parameterize and configure the REST client URI to invoke a specific remote microservice
- Understand and apply additional custom client headers to the client requests
- Demonstrate the ability to convert HTTP responses to handle exceptions
- Understand and use special additional client headers

Use OpenTelemetry (OTel) to instrument distributed tracing for interactive web applications

- Understand and implement distributed tracing across microservices using OpenTelemetry
- Demonstrate the ability to follow and understand the full path through your distributed application
- Understand the Tracer creates Spans and the Context Propagation assembled into a trace in the correlation identifiers
- Understand how contextual information is passed between Spans using the key-store Baggage

As with all Red Hat performance-based exams, configurations must persist after restart without intervention.

[EX380K - Red Hat Certified Specialist in OpenShift Automation and Integration](#)

EX380V46K - Retired

- Deploy Kubernetes applications on OpenShift
 - Assemble an application from Kubernetes components
 - Understand and use Kustomize
 - Use an image stream with a Kubernetes deployment
- Configure and automate OpenShift tasks
 - Create a simple script to automate a task



- Deploy an existing script to automate a task
 - Troubleshoot and correct a script
 - Understand and query the REST API using CLI tools
 - Create a custom role
 - Create a cron job
 - Create a simple Ansible playbook
- Work with and manage OpenShift Operators
 - Install an operator
 - Update an operator
 - Delete an operator
 - Subscribe an operator
 - Troubleshoot an operator
- Work with registries
 - Pull/push content from remote registries
 - Tag images in remote registries
- Implement GitOps with Jenkins
 - Deploy a Jenkins master
 - Create a Jenkins pipeline to remediate configuration drift
- Configure Enterprise Authentication
 - Configure an LDAP identity provider
 - Configure RBAC for an LDAP provided user account
 - Synchronize OpenShift groups with LDAP
- Understand and manage ingress
 - Use the `oc route` command to expose services
 - Understand how ingress components relate to OpenShift deployments and projects
 - Configure trusted TLS Certificates
 - Work with certificates using the web and CLI interfaces
 - Renew and apply a certificate
- Work with machine configurations
 - Understand MachineConfig object structure
 - Create custom machine configurations



- Configure Persistent Storage
 - Provision shared storage for applications
 - Provision block storage
 - Configure and use storage quotas, classes, and policies
 - Troubleshoot storage issues
- Manage Cluster Monitoring and Metrics
 - Manage OpenShift alerts
 - Use monitoring to troubleshoot cluster issues
- Provision and Inspect Cluster Logging
 - Deploy cluster logging
 - Query cluster logs
 - Diagnose cluster logging problems
- Utilize and configure cluster logging
 - Query cluster logs
 - Utilize cluster logs to diagnose problems
 - Deploy the cluster logging operator
- Recover Failed Worker Nodes
 - Diagnose worker node failures
 - Recover a node that has failed

EX380V410K

As part of this exam, you should be able to perform these tasks:

- Deploy Kubernetes applications on OpenShift
 - Assemble an application from Kubernetes components
 - Understand and use Kustomize
 - Use an image stream with a Kubernetes deployment
- Configure and automate OpenShift tasks
 - Create a simple script to automate a task
 - Deploy an existing script to automate a task
 - Troubleshoot and correct a script
 - Understand and query the REST API using CLI tools



- Create a custom role
 - Create a cron job
 - Create a simple Ansible playbook
- Work with and manage OpenShift Operators
 - Install an operator
 - Update an operator
 - Delete an operator
 - Subscribe an operator
 - Troubleshoot an operator
- Work with registries
 - Pull/push content from remote registries
 - Tag images in remote registries
- Implement GitOps with Jenkins
 - Deploy a Jenkins master
 - Create a Jenkins pipeline to remediate configuration drift
- Configure Enterprise Authentication
 - Configure an LDAP identity provider
 - Configure RBAC for an LDAP provided user account
 - Synchronize OpenShift groups with LDAP
- Understand and manage ingress
 - Use the `oc route` command to expose services
 - Understand how ingress components relate to OpenShift deployments and projects
 - Configure trusted TLS Certificates
 - Work with certificates using the web and CLI interfaces
 - Renew and apply a certificate
- Work with machine configurations
 - Understand MachineConfig object structure
 - Create custom machine configurations
- Configure Persistent Storage
 - Provision shared storage for applications
 - Provision block storage



- Configure and use storage quotas, classes, and policies
 - Troubleshoot storage issues
- Manage Cluster Monitoring and Metrics
 - Manage OpenShift alerts
 - Use monitoring to troubleshoot cluster issues
- Provision and Inspect Cluster Logging
 - Deploy cluster logging
 - Query cluster logs
 - Diagnose cluster logging problems
- Recover Failed Worker Nodes
 - Diagnose worker node failures
 - Recover a node that has failed

EX380V414K

- Configure and manage OpenShift with LDAP and OIDC identity providers
 - Integrate OpenShift with LDAP
 - Integrate OpenShift with Red Hat SSO (Keycloak)
 - Manage multiple identity providers
 - Configure authentication
 - Configure and synchronize OpenShift groups
 - Resolve synchronization conflicts
 - Maintain group synchronization on a scheduled basis
 - Configure RBAC roles with users and groups
 - Create and use authentication tokens with kubeconfig files
 - Create and use authentication certificates with kubeconfig files
- Back up and restore applications with OpenShift API for Data Protection (OADP)
 - Export and import a Kubernetes application
 - Export and import container images
 - Deploy the OADP operator
 - Configure a full application backup including data and resources



- Configure and use volume snapshots
 - Schedule a recurring backup
 - Restore applications
- Manage workloads with cluster partitioning
 - Dedicate multiple cluster nodes to a specific workload
 - Configure and work with node labels and selectors
 - Configure and work with machine configurations
 - Use the machine configuration operator and special purpose operators
- Manage workloads with pod scheduling
 - Configure workloads to run on dedicated cluster nodes
 - Prevent workloads from running on dedicated cluster nodes
 - Configure and work with node taints and tolerations
 - Configure and work with project tolerations
 - Configure application resiliency using pod disruption budgets
 - Configure cluster and project default settings
 - Configure and use affinity rules
- Implement OpenShift GitOps
 - Deploy OpenShift GitOps for managing cluster administration
 - Deploy OpenShift GitOps for managing applications
- Manage cluster monitoring and metrics
 - Troubleshoot performance and availability issues with applications and clusters
 - Configure and silence alerts
- Provision and inspect cluster logging
 - Configure OpenShift logging to forward logs to an external aggregator
 - Configure OpenShift logging for short-term log retention and aggregation
 - Deploy OpenShift logging with Vector and Loki
 - Deploy the Event Router to log Kubernetes events
 - Query logs from cluster services and node services
 - Diagnose cluster logging problems



Red Hat Training and Certification

EX403K - Red Hat Certified Specialist in Enterprise Deployment and Systems Management

EX403V66K

- Configure Red Hat Satellite server
 - Create and configure organizations and locations
 - Create and configure users and roles
 - Create and configure Ansible® roles
 - Create and configure Ansible variables
- Manage content and software channels
 - Synchronize Red Hat Enterprise Linux content
 - Configure subscriptions, content, and content views
 - Create custom content channels and populate them with software
 - Create and configure development life cycles
 - Sign an RPM
- Manage systems
 - Create activation keys
 - Create host group keys
 - Configure bare-metal deployments
- Manage Ansible roles
 - Import Ansible roles
 - Configure Ansible roles
- Manage Ansible variables
 - Import Ansible variables from an Ansible role
 - Configure an Ansible variable
- Manage clients
 - Register an existing client to a Satellite server
 - Apply errata to clients
 - Install custom software on clients
 - Use Satellite server to remotely execute scripts on clients
 - Assign an Ansible role to an existing client
- Provision clients
 - Provision bare metal and virtual clients using kickstarts
 - Ensure Ansible configurations are applied upon client installation



As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX403V611K

- Configure Red Hat Satellite server
 - Create and configure organizations and locations
 - Create and configure users and roles
- Manage content and software products
 - Synchronize Red Hat Enterprise Linux content
 - Configure subscriptions, content, and content views
 - Create custom products and repositories and populate them with software
 - Create products with repository discovery
 - Create and configure development life cycles
 - Manage signing of custom RPMs
- Manage systems
 - Register existing hosts
 - Install custom software on clients
 - Manage hosts with host collections
 - Create and use activation keys
 - Deploy Satellite capsule servers
 - Publish and synchronize content to a Satellite capsule server
- Filter content with Content Views
 - Apply ~~emergency~~ updates to selected hosts
 - Create content filters using Content Views
- Manage Ansible roles and remote execution
 - Configure Ansible remote execution
 - Manage Ansible roles and variables in Satellite
 - Run remote jobs on managed hosts
- Provision clients
 - Configure bare-metal deployments
 - Prepare network configuration for provisioning on either a Satellite or a capsule server



- Provision bare metal and virtual clients using kickstarts
- Ensure remote executions are applied upon client installation
- Maintain Satellite
 - Perform Satellite maintenance
 - Backup Satellite

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX415K - Red Hat Certified Specialist in Security: Linux](#)

EX415V75K

- Use Red Hat Ansible® Engine
 - Install Red Hat Ansible Engine on a control node.
 - Configure managed nodes.
 - Configure simple inventories.
 - Perform basic management of systems.
 - Run a provided playbook against specified nodes.
- Configure intrusion detection
 - Install AIDE.
 - Configure AIDE to monitor critical system files.
- Configure encrypted storage
 - Encrypt and decrypt block devices using LUKS.
 - Configure encrypted storage persistence using NBDE.
 - Change encrypted storage passphrases.
- Restrict USB devices
 - Install USBGuard.
 - Write device policy rules with specific criteria to manage devices.
 - Manage administrative policy and daemon configuration.
- Manage system login security using pluggable authentication modules (PAMs)
 - Configure password quality requirements.
 - Configure failed login policy.
 - Modify PAM configuration files and parameters.



- Configure system auditing
 - Write rules to log auditable events.
 - Enable prepackaged rules.
 - Produce audit reports.
- Configure SELinux
 - Enable SELinux on a host running a simple application.
 - Interpret SELinux violations and determine remedial action.
 - Restrict user activity with SELinux user mappings.
 - Analyze and correct existing SELinux configurations.
- Enforce security compliance
 - Install OpenSCAP and Workbench.
 - Use OpenSCAP and Red Hat Insights to scan hosts for security compliance.
 - Use OpenSCAP Workbench to tailor policy.
 - Use OpenSCAP Workbench to scan an individual host for security compliance.
 - Use Red Hat Satellite server to implement an OpenSCAP policy.
 - Apply OpenSCAP remediation scripts to hosts.

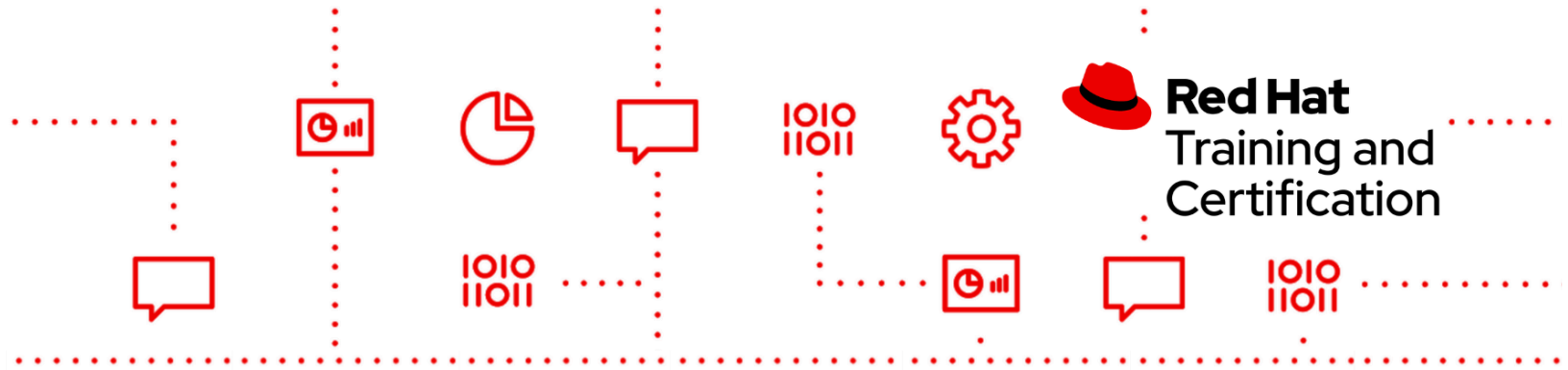
As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

EX415V92K

- Configure Ansible Core
 - Install Red Hat Ansible Core on a control node
 - Configure managed nodes
 - Configure simple inventories
 - Perform basic management of systems
 - Run a provided playbook against specified nodes
- Implement access controls for automation controller
 - Create and restrict an inventory to an automation controller user
 - Restrict a credential and/or a project to an automation controller user
 - Be able to create and launch a template as an automation controller user
- Configure intrusion detection



- Install AIDE
- Configure AIDE to monitor critical system files
- Configure encrypted storage
 - Encrypt and decrypt block devices, either physical or logical, using LUKS
 - Configure encrypted storage persistence using NBDE
 - Change encrypted storage passphrases
- Restrict USB devices
 - Install USBGuard
 - Write device policy rules with specific criteria to manage devices
 - Manage administrative policy and daemon configuration
- Manage system login security using pluggable authentication modules (PAM)
 - Configure password quality requirements
 - Configure failed login policy
 - Modify PAM configuration files and parameters
- Configure system auditing
 - Write rules to log auditable events
 - Enable prepackaged rules
 - Produce audit reports
- Configure SELinux
 - Enable SELinux on a host running a simple application
 - Interpret SELinux violations and determine remedial action
 - Restrict user activity with SELinux user mappings
 - Analyze and correct existing SELinux configurations
- Enforce security compliance using OpenSCAP
 - Install OpenSCAP and OpenSCAP Workbench
 - Scan hosts for security compliance
 - Tailor security policy
 - Scan individual hosts for security compliance
 - Generate and apply a playbook from customized XML for remediation of inventory hosts



As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX427K - Red Hat Certified Specialist in Business Process Design](#)

EX427V73K - Retired

- User Task
- Script
- Service
- Custom work item
- Business rule task
- Sub process

Define and manipulate Business Process variables, including:

- Global variables
- Process variables
- Task variables
- Mapping variables to activity parameters

Understand and use flow control in business processes, including:

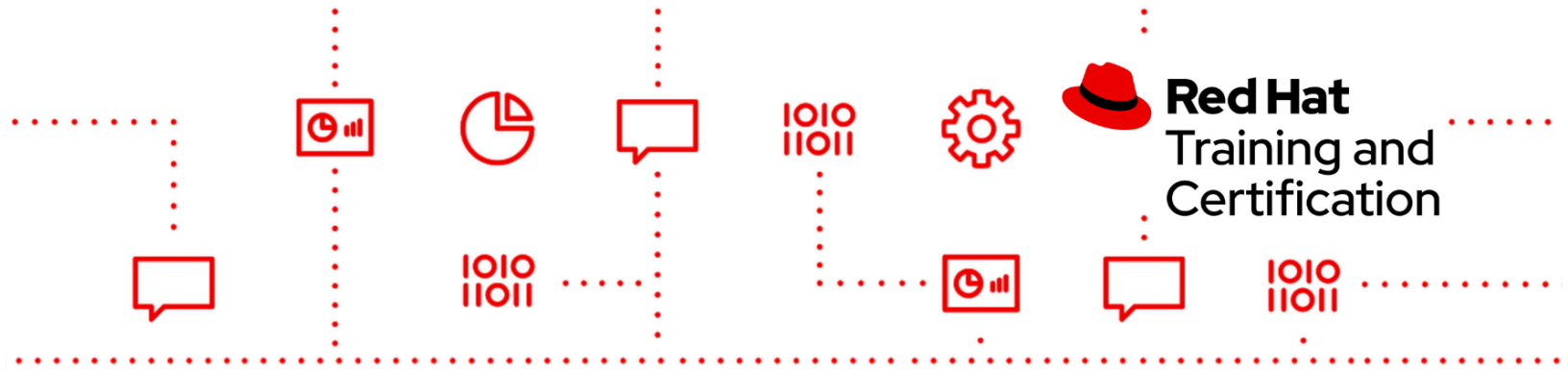
- Understanding the meaning and use of the different start and end events
- Using gateways to branch process flows

Understand the specific benefits and weaknesses of different Business Process elements, including but not limited to:

- Script vs. work item
- Stop vs. terminate

Handle technical exceptions and process errors:

- Using signal
- Using error code



- Triggered by event subprocess
- Integrate with external processes

Familiarize yourself with the official [Red Hat Process Automation Manager documentation](#) for the version used in the exam (RHPAM 7.3). The exam environment includes a snapshot of the official RHPAM 7.3 docs so being familiar with them will significantly help if you need to look up anything during the exam. In addition to the official documentation for the product version being tested, there may also be sample files or additional documentation included with the product itself. We will not strip out anything that is normally included, but you will not be able to use the Internet to search for or use other documentation. It's a good idea to keep these restrictions in mind as you study. Make sure you're familiar with, and only rely on, the official product documentation for the version specified. Due to technical limitations in the exam environment, we strongly recommend that you use the +HTML-SINGLE+ version of the docs as that is the version that is easiest to use and view during the exam.

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX436K - Red Hat Certified Specialist in High Availability Clustering](#)

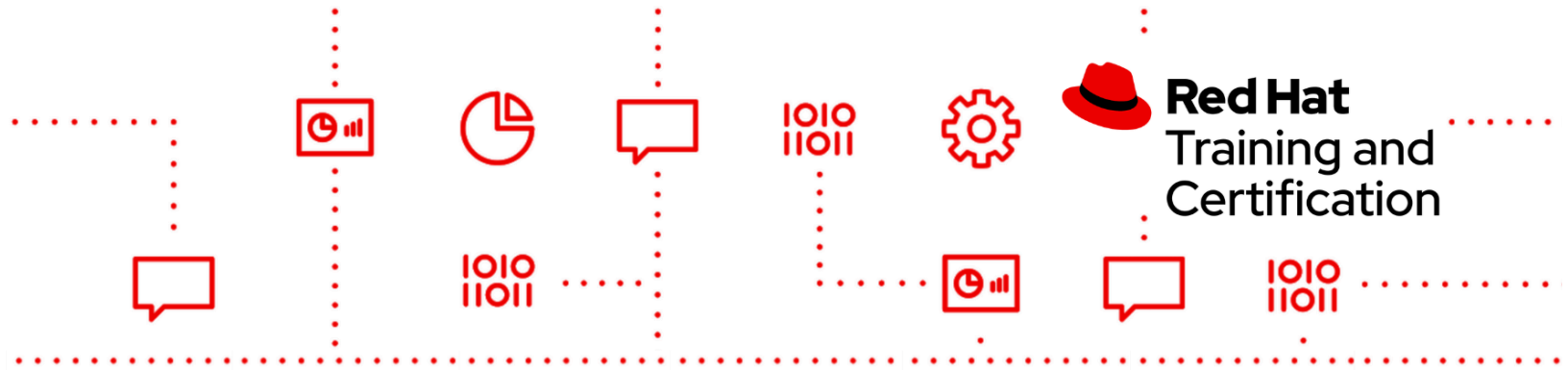
EX436V83K

- Configure a high availability cluster
 - Install high availability clustering
 - Install and configure a high availability cluster either manually or using Ansible
 - Configure cluster quorum options
- Configure cluster fencing
 - Configure standard fence mechanisms such as fence_ipmilan
 - Test fencing configurations using standard tools
 - Configure fencing so that any cluster member can fence any other cluster member
- Configure cluster logging and monitoring



- Configure cluster logging so that each node system activity is logged to a separate file
 - Configure cluster logging so that logging messages will be forwarded to journald
- Configure cluster monitoring
 - Create and configure a cluster monitoring resource
 - Log cluster events and send notification emails to a specific address
 - Configure cluster alerts
- Configure a clustered fail-over service
 - Create and configure a cluster highly available service
 - Configure a specific resource group
 - Configure services to use shared storage
 - Configure services to use a specific IP
- Configure cluster service behavior
 - Restrict where services run
 - Configure service failover behavior
- Configure storage
 - Configure an iSCSI initiator
 - Create and configure shared storage using provided iSCSI volumes
 - Configure multipath access to shared storage
 - Configure shared LVM devices
 - Configure highly available LVM devices
- Configure GFS2 filesystems
 - Create GFS2 filesystems on logical volumes
 - Configure GFS filesystems to be shared between multiple nodes simultaneously
 - Manage GFS2 filesystems
 - Add journals to existing GFS2 filesystems
- Grow a GFS2 filesystem

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.



[EX440K - Red Hat Certified Specialist in Messaging Administration](#)

EX440V7K - Retired

- Create a JBoss AMQ simple broker instance
- Configure addresses, queues, and topics
- Configure point-to-point and publisher-subscriber messaging
- Understand JMS messaging naming convention
- Configure expiring message, moving to Dead Letter queue
- Configure fine-grained address settings
- Apply AMQ broker wildcard syntax
- Configure paging
- Understand and tune performances
- Enable/disable access to guest and configure guest access
- Configure authorization and authentication
- Secure network connection, acceptors, and connectors
- Configure a cluster of brokers
- Configure high availability and failover

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX442K - Red Hat Certified Specialist in Performance Tuning](#)

EX442V80K

- Use utilities to analyze system behavior
 - Use utilities such as vmstat, iostat, mpstat, sar, gnome-system-monitor, top, powertop, and others to analyze and report system and application behavior
 - Use utilities such as Performance Co-Pilot (PCP) to analyze system behaviour
 - Use utilities such as dmesg, dmidecode, and sosreport to profile system hardware configurations
- Monitor and alter kernel behavior



- Use /proc/sys, sysctl, and /sys to examine, modify, and set kernel run-time parameters
 - Configure kernel behavior by altering module parameters
- Analyze system and application performance
 - Analyze system and application behavior using tools such as ps, top, and Valgrind
 - Configure systems to run SystemTap scripts
 - Use the eBPF family of tools (e.g. syscount, gethostlatency and others) to diagnose system and application behavior
 - Given multiple versions of applications that perform the same or similar tasks, choose which version of the application to run on a system based on its observed performance characteristics
- Tune running systems
 - Alter process priorities of both new and existing processes
 - Select and configure tuned profiles
 - Manage system resource usage using control groups
- Tune memory utilization
 - Configure systems to support alternate page sizes for applications that use large amounts of memory
- Configure disk and file subsystems
 - Select proper I/O scheduling algorithm
 - Tune file system layout for a given use
- Tune network performance
 - Calculate network buffer sizes based on known quantities such as bandwidth and round-trip time
 - Set system buffer sizes based on those calculations

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.



- Understand and use Git
 - Clone a Git repository
 - Update, modify and create files in a Git repository
 - Add those modified files back into the Git repository
- Manage inventory variables
 - Structure host and group variables using multiple files per host or group
 - Use special variables to override the host, port, or remote user Ansible uses for a specific host
 - Set up directories containing multiple host variable files for some of your managed hosts
 - Override the name used in the inventory file with a different name or IP address
- Manage task execution
 - Control privilege execution
 - Run selected tasks
- Transform data with filters and plugins
 - Populate variables with data from external sources using lookup plugins
 - Use lookup and query functions to template data from external sources into playbooks and deployed template files
 - Implement loops using structures other than simple lists using lookup plugins and filters
 - Inspect, validate, and manipulate variables containing networking information with filters
- Delegate tasks
 - Run a task for a managed host on a different host, then control whether facts gathered by that task are delegated to the managed host or the other host
- Install Ansible Tower
 - Perform basic configuration of Ansible Tower after configuration
- Manage access for Ansible Tower
 - Create Ansible Tower users and teams and make associations of one to the other
- Manage inventories and credentials
 - Manage advanced inventories



Red Hat Training and Certification

- Create a dynamic inventory from an identity management server or a database server
 - Create machine credentials to access inventory hosts
 - Create a source control credential
- Manage projects
 - Create a job template
- Manage job workflows
 - Create a job workflow template
- Work with the Ansible Tower API
 - Write an API scriptlet to launch a job
- Back up Ansible Tower
 - Back up an instance of Ansible Tower

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX457K - Red Hat Certified Specialist in Ansible Network Automation](#)

EX457V25K - Retired

- Work from a Red Hat Enterprise Linux command line environment
 - Use a text editor to create and modify files
 - Run commands from the shell
 - Use SSH to log in to remote systems
- Work with Git and perform basic Git operations
 - Clone a Git repository
 - Add files to a Git repository
 - Upload changes to a remote Git repository
 - Perform simple Git customizations
- Configure Ansible
 - Customize the configuration file
 - Configure the connection method
 - Configure authentication details

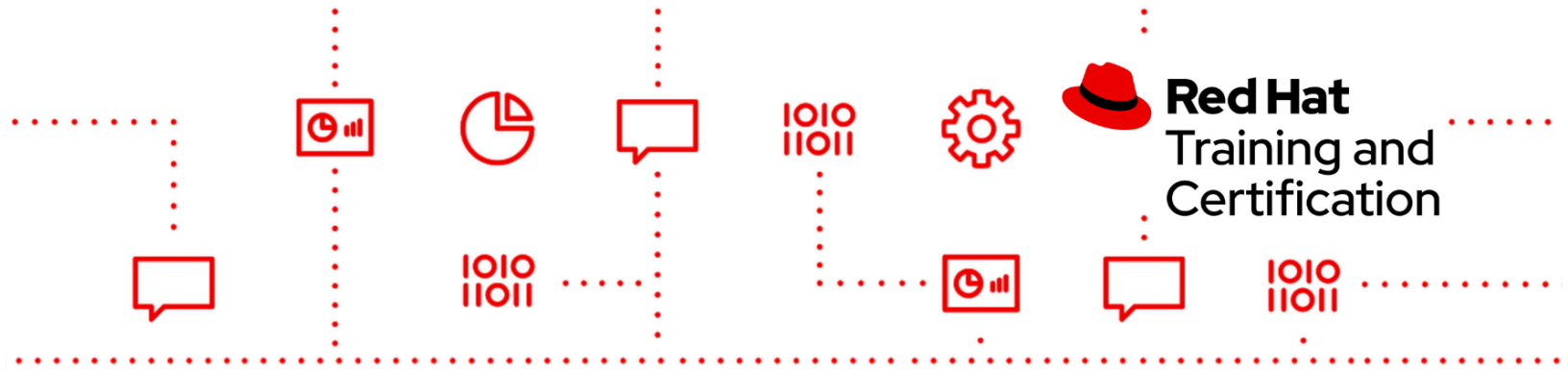


- Verify the configuration
- Run Ansible commands and playbooks
 - Execute Ansible ad hoc commands
 - Create Ansible playbooks
 - Modify existing Ansible playbooks
- Use standard Ansible constructs
 - Work with variables
 - Loop over a list of IP addresses
 - Perform conditional tests
 - Manipulate data with filters
 - Control playbook execution with tags
 - Work with templates
 - Use registers to capture the output of a task
 - Use Ansible vault to secure sensitive information
- Work with Ansible roles
 - Use an existing role
 - Create a new role
- Automate network device management
 - Gather facts about devices and systems
 - Configure routers, switches and ports
 - Configure OSPF
 - Configure BGP
 - Configure VLANs
 - Create a multi-play playbook to back up a device configuration
 - Configure devices to use syslog and SNMP
- Use Ansible Tower
 - Create credentials
 - Create organizations, users and teams
 - Create a dynamic inventory
 - Create a project from a Git repository
 - Create a job template
 - Create a survey



EX457V23K

- Work from a Red Hat Enterprise Linux command line environment
 - Use a text editor to create and modify files
 - Run commands from the shell
 - Use SSH to log in to remote systems
- Configure Automation content navigator to work with Ansible Automation Platform
 - Build inventories
 - Configure Automation content navigator
- Manage playbooks using Git repositories
 - Write and run playbooks.
 - Work with Git and perform basic Git operations
- Manage variables and facts.
 - Create flexible playbooks using variables
 - Gather facts from managed network devices
 - Transform data with filters
- Implement task control
 - Write loops and conditional tasks
 - Handle task failures
- Work with automation controller
 - Create machine and source credentials
 - Create projects
 - Create workflow templates
 - Run jobs
- Utilize roles and Ansible Content Collections
 - Include and import files
 - Create network automation roles
 - Use roles and modules from Ansible Content Collections
- Automate network administration tasks
 - Generate configuration settings from Jinja2 templates
 - Use platform independent modules from the ansible.netcommon collection
 - Perform configuration backups



[EX467K - Red Hat Certified Specialist in Managing Automation with Ansible Automation Platform](#)

EX467V22K

- Install Ansible Automation Platform
 - Install Private Automation Hub
 - Install automation controller
 - Configure automation controller after installation
- Configure Private Automation Hub
 - Upload content collections to Private Automation Hub
 - Upload execution environments to Private Automation Hub
- Manage access to Private Automation Hub
 - Create groups and assign different permissions to groups
 - Create users and assign them to groups
 - Configure user types
- Manage access for automation controller
 - Create automation controller users and teams
 - Associate users with teams associations
 - Configure organization roles
 - Configure user types
- Manage inventories and credentials
 - Create host groups
 - Assign systems to host groups
 - Configure access to inventories
 - Configure inventory variables
 - Create and configure machine credentials to access hosts
- Manage automation controller projects
 - Create projects
 - Create job templates
 - Control user access to job templates
 - Launch jobs
 - Create a job template survey
 - Use variables from a survey in a Jinja template



- Schedule jobs and manage notifications
- Manage automation controller workflows
 - Create workflow templates
 - Launch workflow jobs
 - Configure workflow jobs for automatic approval
- Manage advanced inventories
 - Import external static inventories
 - Filter hosts with Smart Inventories
- Work with the automation controller API
 - Understand how to use an API script to obtain required parameters for launching jobs
 - Write an API scriptlet to launch a job
- Install and configure automation mesh
 - Install automation mesh
 - Create and manage instance groups
 - Assign default instance groups to controller inventories or templates
 - Run controller job templates in a specific instance group
- Back up Ansible Automation Platform
 - Back up an instance of automation controller
 - Back up an instance of Ansible Private Hub

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[EX480K - Red Hat Certified Specialist in MultiCluster Management](#)

EX480V410K

- Manage and deploy Red Hat Advanced Cluster Management (RHACM) for Kubernetes
 - Understand how RHACM components work
 - Deploy RHACM using Red Hat OpenShift Container Platform Operator Lifecycle Management
 - Install RHACM agents in a managed cluster
 - Use RHACM from the web terminal or the command line interface



Red Hat Training and Certification

- Work with multiple OpenShift Container Platform clusters
 - Import a cluster
 - Remove an existing cluster
- Configure access control for multi cluster management
 - Be able to configure objects related to access control in a fleet cluster such as roles, cluster sets, cluster selectors and placement rules
 - Understand how to use the multi cluster management dashboard search engine
 - Create and manage role based access controls (RBAC)
 - Manipulate policies for group and users
 - Manipulate cluster sets
- Deploy and manage policies for multiple clusters
 - Understand governance architecture and related components
 - Manipulate RHACM policies and related components
 - Configure and use the Compliance Operator for multiple clusters
 - Understand how Compliance Operator policy cluster resource definitions works and be able to manipulate them
- Manage the RHACM observability service
 - Understand the RHACM observability architecture
 - Enable the observability service
 - Manage the observability service
 - Customize the observability service
- Manage multi cluster application resources
 - Be able to work with Git from the CLI
 - Organize Git repositories to support GitOps
 - Manipulate applications using Git repositories
 - Use the application console to manage manipulate applications
- Use Kustomize to manage resources
 - Understand how to manage templates
 - Manage apps using Kustomize
 - Deploy and Configure Red Hat Quay
 - Understand the Red Hat Quay architecture and components
 - Configure the Quay Operator and related objects



- Understand the Red Hat Quay tenancy model
- Use Podman and Skopeo to manipulate images in repositories
- Integrate RHACM with OpenShift pipelines and GitOps
 - Understand RHACM application policies
 - Understand pipeline operators
 - Deploy OpenShift GitOps in a hub cluster
 - Integrate CI/CD pipeline with RHACM
- Install and Configure Red Hat Advanced Cluster Security (RHACS)
 - Understand the security considerations in Kubernetes multi cluster environments
 - Understand the RHACS architecture
 - Manage advanced cluster security
 - Customize advanced cluster security settings
- Import managed clusters into RHACS
 - Import a secured clusters to RHACS
 - Explore the RHACS by web console or cli
 - Use Helm to deploy applications
- Integrate RHACS with external registries
 - Understand how to manipulate images with podman
 - Know how to configure external registry
- Perform multi cluster vulnerability management with RHACS
 - Scan and detect vulnerabilities using RHACS web console
 - Create RHACS policy to prevent deployment of vulnerable images
- Manage multi cluster configuration with RHACS
 - Identify configuration management issues affecting a deployment
 - Verify that configurations issue are resolved

[EX482K - Red Hat Certified Specialist in Event-Driven Development with Kafka](#)

EX482V79K, EX482V18K

- Understand and work with event-driven applications with AMQ Streams API.
 - Know how to send and read data from Kafka.
 - Be able to develop microservices and other types of applications to share data with extremely high throughput and low latency.



- Understand the Kafka ecosystem and architecture:
 - How to create, configure, and manage topics.
 - How to configure the ecosystem to share data with extremely high throughput and low latency.
 - How to scale and guarantee message ordering.
 - Message compaction to remove old records, and how to set them.
 - Configuration and use of the replication of data to control fault tolerance.
 - Retention of high volumes of data for immediate access.
- Understand and work with a Quarkus application connected to Kafka
 - Connect to Kafka with Reactive Messaging
 - Connect to Apache Kafka with its native API
 - Produce and consume messages and implement event-driven and data-streaming applications
 - Be familiar with the reactive libraries used by Quarkus : Asynchronous Java or Publisher API, RxJava or Reactor APIs, Mutiny, etc.
- Provide and configure access to a Kafka cluster.
 - Be able to access the external listeners of Kafka on the cloud. In the cases of Kubernetes or Red Hat OpenShift, connect via node ports, load balancers, and externally, using an ingress or OpenShift route.
 - Understand how to configure the security of the communications between the Kafka client and the cluster.
 - Understand and provide the Kafka client configuration for the required authentication and authorization security.
- Provide and use the schema Red Hat Service Registry to decouple the data from client applications, share and manage the data types at runtime:
 - Understand how to manage the registry content using the web console, REST API command, Maven plug-in, or Java client.
 - Be able to understand the rule definitions for content validation and version compatibility for schema governance.
 - Know and use the schema registry support in Apache Kafka, and understand the integration with Kafka Connect for external systems.
 - Be able to work with the client SerDes (Serializer/Deserializer) to validate Kafka and other message types at runtime.
- Understand, produce, test, and secure data-stream processing using the Kafka Streams API to perform an efficient management and real-time querying of application state.
 - Understand and work with the different Kafka Streams APIs like Streams DSL and Processor API.



- Configure and provide the proper Kafka SerDes (Serializer/Deserializer) for the records to correctly materialize the data
- Be able to receive data from one or more input streams, execute complex operations like mapping, filtering or joining, repartition and/or grouping, and write the results into one or more output streams.
- Understand the stream-table duality and perform stateful operations like joins, aggregations, and windowed joins.
- Understand how to define and connect custom processors and transformers to interact with state stores using the Processor API.
- Understand the event manipulation deriving new collections from existing ones and describing changes between them.
- Data integration with Kafka Connect:
 - Understand how Kafka Connect provides reliability and scalability data transferring between Kafka and other heterogeneous data systems.
 - Understand how Kafka Connect facilitates data conversion, transformation, and offset management.
 - Apply the detecting and capturing data changes (CDC) with Debezium.
 - Understand the different stand-alone/distributed running modes and their use cases.
 - Use the pre-built AMQ Streams connectors.
- Understand and use advanced event-driven patterns in applications based on Apache Kafka:
 - Recognize and work in an application with Event Sourcing and CQRS patterns
 - Know and work in advanced techniques like long-running business transactions with Saga orchestration and outbox patterns to exchange data between different services.
- Troubleshoot most common problems in event-driven applications like maintaining message ordering, retries and idempotency, handling duplicate events, implement Streams test cases

During the exam you will be required to work with one or more pre-written Java applications. You will be required to modify some parts of the application code.

As with all Red Hat performance-based exams, configurations and changes must persist after reboot without intervention.



[PE110K - Preliminary Exam in Red Hat OpenStack Administration](#)

PE110V16K

- Understand and work with Director based deployments
 - Use identity environment files to connect to the undercloud
 - Use identity environment files to connect to the overcloud
 - Use template files, environment files and other resources to obtain information about an OpenStack environment
 - Work with containerized services
- Configure OpenStack domains
 - Create projects
 - Create groups
 - Create users
 - Manage quotas
- Create resources
 - Create virtual machine flavors
 - Add existing images to an overcloud
 - Create security groups
 - Create key pairs
- Configure networking
 - Create and assign networks to projects
 - Configure network routers
 - Configure software defined networks
 - Work with Open Virtual Networks (OVN)
- Configure floating IP addresses
 - Configure an instance to use a floating IP address
 - Configure a service to be accessible via a floating IP address
- Manage instances
 - Launch instances
 - Associate instances with specified projects and networks
 - Use key pairs to connect to instances
 - Configure an instance during deployment



Red Hat reserves the right to add, modify, and remove objectives. Such changes will be made public in advance through revisions to this document.

[PE124K - Preliminary Exam in Red Hat OpenStack Administration](#)

PE124V80K

- Understand and use essential tools
 - Access a shell prompt and issue commands with correct syntax
 - Use input-output redirection (>, >>, |, 2>, etc.)
 - Access remote systems using SSH
 - Log in and switch users in multi-user targets
 - Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2
 - Create and edit text files
 - Create, delete, copy, and move files and directories
 - Create hard and soft links
 - List, set, and change standard ugo/rwx permissions
 - Locate, read, and use system documentation including man, info, and files in /usr/share/doc
- Operate running systems
 - Boot, reboot, and shut down a system normally
 - Boot systems into different targets manually
 - Identify CPU/memory intensive processes and kill processes
 - Locate and interpret system log files and journals
 - Preserve system journals
 - Start, stop, and check the status of network services
 - Securely transfer files between systems
- Create and configure file systems
 - Create and configure set-GID directories for collaboration
 - Diagnose and correct file permission problems
 - Manually mount and unmount existing filesystem
- Deploy, configure, and maintain systems
 - Start and stop services and configure services to start automatically at boot
 - Configure systems to boot into a specific target automatically



- Configure time service clients
 - Install and update software packages from Red Hat Network, a remote repository, or from the local file system
 - Work with package module streams
 - Modify the system bootloader
- Manage basic networking
 - Configure IPv4 and IPv6 addresses
 - Configure hostname resolution
 - Configure network services to start automatically at boot
- Manage users and groups
 - Create, delete, and modify local user accounts
 - Change passwords and adjust password aging for local user accounts
 - Create, delete, and modify local groups and group memberships
 - Configure superuser access

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

PE124V90K

- Understand and use essential tools
 - Access a shell prompt and issue commands with correct syntax
 - Use input-output redirection (>, >>, |, 2>, etc.)
 - Access remote systems using SSH
 - Log in and switch users in multi-user targets
 - Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2
 - Create and edit text files
 - Create, delete, copy, and move files and directories
 - Create hard and soft links
 - List, set, and change standard ugo/rwx permissions
 - Locate, read, and use system documentation including man, info, and files in /usr/share/doc



- Operate running systems
 - Boot, reboot, and shut down a system normally
 - Boot systems into different targets manually
 - Identify CPU/memory intensive processes and kill processes
 - Locate and interpret system log files and journals
 - Preserve system journals
 - Start, stop, and check the status of network services
 - Securely transfer files between systems
- Create and configure file systems
 - Create and configure set-GID directories for collaboration
 - Diagnose and correct file permission problems
 - Manually mount and unmount existing filesystem
- Deploy, configure, and maintain systems
 - Start and stop services and configure services to start automatically at boot
 - Configure systems to boot into a specific target automatically
 - Configure time service clients
 - Install and update software packages from Red Hat Network, a remote repository, or from the local file system
 - Work with package module streams
 - Modify the system bootloader
- Manage basic networking
 - Configure IPv4 and IPv6 addresses
 - Configure hostname resolution
 - Configure network services to start automatically at boot
- Manage users and groups
 - Create, delete, and modify local user accounts
 - Change passwords and adjust password aging for local user accounts
 - Create, delete, and modify local groups and group memberships
 - Configure superuser access

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

[PE180K - Preliminary Exam in Red Hat OpenShift Administration](#)

PE180V412K

- Manage OpenShift Container Platform
 - Use the web console to manage and configure an OpenShift cluster
 - Deploy and use the command-line interface to manage and configure an OpenShift cluster
 - Create and delete projects
 - Locate and examine container images
 - Query, format, and filter attributes of Kubernetes resources
 - Import, export, and configure Kubernetes resources
 - Examine resources and cluster status
 - Monitor cluster events and alerts
 - View logs
 - Assess the health of an OpenShift cluster
 - Troubleshoot common container, pod, and cluster events and alerts
 - Use product documentation
- Deploy applications
 - Deploy applications from images and templates
 - Deploy jobs to perform one-time tasks
 - Manage application deployments
 - Work with replica sets
 - Work with labels and selectors
 - Configure services
 - Expose applications to external access
- Manage storage for application configuration and data
 - Create and use secrets
 - Create and use configuration maps
 - Provision persistent storage volumes for block and file-based data
 - Use storage classes
 - Manage non-shared storage with stateful sets



Red Hat Training and Certification

- Configure applications for reliability
 - Configure and use health probes
 - Reserve and limit application compute capacity
 - Scale applications to meet increased demand
- Manage application updates
 - Identify images using tags and digests
 - Roll back failed deployments
 - Manage image streams
 - Use triggers to manage images

As with all Red Hat performance-based exams, configurations must persist after reboot without intervention.

Red Hat reserves the right to add, modify, and remove objectives. Such changes will be made public in advance through revisions to this document.