RH OVE References Documentation

Red Hat OpenShift Virtualization Ecosystem Team

2025-08-04

Table of Contents

# References

## Product Urls

### Product URLs and References

#### Overview

This document provides an exhaustive list of product URLs, documentation, and resources for all components used in the RH OVE ecosystem design and implementation.

#### Core Platform Components

##### Red Hat OpenShift

* **Product Homepage**: <https://www.redhat.com/en/technologies/cloud-computing/openshift>
* **Documentation**: <https://docs.openshift.com/>
* **Container Platform Docs**: <https://docs.openshift.com/container-platform/latest/>
* **Installation Guide**: <https://docs.openshift.com/container-platform/latest/installing/>
* **API Reference**: <https://docs.openshift.com/container-platform/latest/rest_api/>
* **Red Hat Customer Portal**: <https://access.redhat.com/>
* **OpenShift Blog**: <https://www.redhat.com/en/blog/channel/red-hat-openshift>

##### Red Hat OpenShift Virtualization Engine (RH OVE)

* **Product Page**: <https://www.redhat.com/en/technologies/cloud-computing/openshift/virtualization-engine>
* **Datasheet**: <https://www.redhat.com/en/resources/red-hat-openshift-virtualization-engine-datasheet>
* **Documentation**: <https://docs.redhat.com/en/documentation/openshift_container_platform/4.14/html-single/virtualization/>
* **Getting Started**: <https://docs.redhat.com/en/documentation/openshift_container_platform/4.15/html/virtualization/about>
* **Reference Implementation**: <https://access.redhat.com/sites/default/files/attachments/openshift_virtualization_reference_implementation_guide.pdf>

##### KubeVirt (Upstream)

* **Project Homepage**: <https://kubevirt.io/>
* **GitHub Repository**: <https://github.com/kubevirt/kubevirt>
* **Documentation**: <https://kubevirt.io/user-guide/>
* **Architecture**: <https://kubevirt.io/user-guide/architecture/>
* **Installation**: <https://kubevirt.io/user-guide/cluster_admin/installation/>
* **API Validation**: <https://kubevirt.io/user-guide/cluster_admin/api_validation/>
* **GitOps Guide**: <https://kubevirt.io/user-guide/cluster_admin/gitops/>

#### Networking Components

##### Cilium

* **Project Homepage**: <https://cilium.io/>
* **Documentation**: <https://docs.cilium.io/>
* **GitHub Repository**: <https://github.com/cilium/cilium>
* **OpenShift Installation**: <https://docs.cilium.io/en/stable/installation/k8s-install-openshift-okd.html>
* **Blog**: <https://cilium.io/blog/>
* **OpenShift Certification**: <https://cilium.io/blog/2021/04/19/openshift-certification/>
* **Learning Hub**: <https://www.tigera.io/learn/guides/cilium-vs-calico/cilium/>

##### Hubble (Network Observability)

* **Documentation**: <https://docs.cilium.io/en/stable/gettingstarted/hubble/>
* **GitHub Repository**: <https://github.com/cilium/hubble>
* **UI Repository**: <https://github.com/cilium/hubble-ui>

##### Multus CNI (Multi-Network)

* **Project Homepage**: <https://github.com/k8snetworkplumbingwg/multus-cni>
* **Documentation**: <https://github.com/k8snetworkplumbingwg/multus-cni/blob/master/docs/how-to-use.md>
* **OpenShift Documentation**: <https://docs.openshift.com/container-platform/latest/networking/multiple_networks/understanding-multiple-networks.html>
* **Network Attachment Definitions**: <https://docs.openshift.com/container-platform/latest/networking/multiple_networks/configuring-additional-network-types.html>
* **SR-IOV Network Operator**: <https://docs.openshift.com/container-platform/latest/networking/hardware_networks/about-sriov.html>
* **CNCF Landscape**: <https://landscape.cncf.io/card-mode?category=cni&grouping=category>
* **Kubernetes Network SIG**: <https://github.com/kubernetes/community/tree/master/sig-network>

##### SR-IOV Network Operator

* **GitHub Repository**: <https://github.com/k8snetworkplumbingwg/sriov-network-operator>
* **OpenShift SR-IOV Documentation**: <https://docs.openshift.com/container-platform/latest/networking/hardware_networks/installing-sriov-operator.html>
* **Configuration Guide**: <https://docs.openshift.com/container-platform/latest/networking/hardware_networks/configuring-sriov-device.html>
* **Performance Tuning**: <https://docs.openshift.com/container-platform/latest/networking/hardware_networks/using-dpdk-and-rdma.html>

##### Network Plumbing Working Group

* **GitHub Organization**: <https://github.com/k8snetworkplumbingwg>
* **Community Meetings**: <https://github.com/k8snetworkplumbingwg/community>
* **CNI Plugins**: <https://github.com/containernetworking/plugins>

#### Policy and Security

##### Kyverno

* **Project Homepage**: <https://kyverno.io/>
* **Documentation**: <https://kyverno.io/docs/>
* **GitHub Repository**: <https://github.com/kyverno/kyverno>
* **Installation**: <https://kyverno.io/docs/installation/>
* **Policy Examples**: <https://kyverno.io/policies/>
* **Admission Controllers Guide**: <https://kyverno.io/docs/introduction/admission-controllers/>
* **Helm Chart**: <https://github.com/kyverno/kyverno/tree/main/charts/kyverno>

#### GitOps and CI/CD

##### Argo CD

* **Project Homepage**: <https://argoproj.github.io/argo-cd/>
* **Documentation**: <https://argo-cd.readthedocs.io/>
* **GitHub Repository**: <https://github.com/argoproj/argo-cd>
* **Getting Started**: <https://argo-cd.readthedocs.io/en/stable/getting_started/>
* **OpenShift GitOps**: <https://docs.openshift.com/container-platform/latest/cicd/gitops/understanding-openshift-gitops.html>

##### Red Hat OpenShift GitOps

* **Product Page**: <https://www.redhat.com/en/technologies/cloud-computing/openshift/gitops>
* **Documentation**: <https://docs.openshift.com/container-platform/latest/cicd/gitops/>
* **Operator Hub**: <https://operatorhub.io/operator/openshift-gitops-operator>

#### Monitoring and Observability

##### Dynatrace

* **Product Homepage**: <https://www.dynatrace.com/>
* **Platform Documentation**: <https://docs.dynatrace.com/>
* **OpenShift Integration**: <https://docs.dynatrace.com/docs/ingest-from/setup-on-k8s/deployment/other/ocp-operator-hub>
* **Kubernetes Monitoring**: <https://docs.dynatrace.com/docs/observe/infrastructure-monitoring/container-platform-monitoring/kubernetes-monitoring/>
* **Operator GitHub**: <https://github.com/Dynatrace/dynatrace-operator>
* **Red Hat Partner Page**: <https://www.dynatrace.com/hub/detail/red-hat-openshift/>
* **Blog**: <https://www.dynatrace.com/news/blog/>

##### Prometheus

* **Project Homepage**: <https://prometheus.io/>
* **Documentation**: <https://prometheus.io/docs/>
* **GitHub Repository**: <https://github.com/prometheus/prometheus>
* **OpenShift Monitoring**: <https://docs.openshift.com/container-platform/latest/monitoring/>

##### Grafana

* **Project Homepage**: <https://grafana.com/>
* **Documentation**: <https://grafana.com/docs/>
* **GitHub Repository**: <https://github.com/grafana/grafana>

#### Backup and Recovery

##### Rubrik

* **Product Homepage**: <https://www.rubrik.com/>
* **OpenShift Solutions**: <https://www.rubrik.com/solutions/openshift>
* **Kubernetes Solutions**: <https://www.rubrik.com/solutions/kubernetes>
* **Documentation**: <https://docs.rubrik.com/>
* **Compatibility Matrix**: <https://docs.rubrik.com/en-us/compat_matrix/index.html>
* **Blog**: <https://www.rubrik.com/blog/>
* **Red Hat Partnership**: <https://www.redhat.com/en/blog/red-hat-openshift-virtualization-and-rubrik>

#### ITSM and CMDB Integration

##### ServiceNow

* **Product Homepage**: <https://www.servicenow.com/>
* **CMDB Documentation**: <https://docs.servicenow.com/bundle/xanadu-servicenow-platform/page/product/configuration-management/>
* **Developer Documentation**: <https://developer.servicenow.com/>
* **REST API Guide**: <https://docs.servicenow.com/bundle/xanadu-application-development/page/integrate/inbound-rest/concept/c_RESTAPI.html>
* **IntegrationHub**: <https://docs.servicenow.com/bundle/xanadu-servicenow-platform/page/administer/integrationhub/>

#### Storage Solutions

##### Container Storage Interface (CSI)

* **CSI Specification**: <https://github.com/container-storage-interface/spec>
* **Kubernetes CSI Documentation**: <https://kubernetes-csi.github.io/docs/>
* **OpenShift Storage**: <https://docs.openshift.com/container-platform/latest/storage/>

##### Containerized Data Importer (CDI)

* **GitHub Repository**: <https://github.com/kubevirt/containerized-data-importer>
* **Documentation**: <https://kubevirt.io/user-guide/storage/disks_and_volumes/>

#### Additional Tools and Utilities

##### virtctl

* **Documentation**: <https://kubevirt.io/user-guide/virtual_machines/accessing_virtual_machines/>
* **GitHub Repository**: <https://github.com/kubevirt/kubevirt/tree/main/cmd/virtctl>

##### OpenShift CLI (oc)

* **Documentation**: <https://docs.openshift.com/container-platform/latest/cli_reference/openshift_cli/getting-started-cli.html>
* **Download**: <https://mirror.openshift.com/pub/openshift-v4/clients/ocp/>

#### Community and Learning Resources

##### Red Hat Learning

* **Red Hat Training**: <https://www.redhat.com/en/services/training>
* **OpenShift Learning Portal**: <https://learn.openshift.com/>
* **Red Hat Developer**: <https://developers.redhat.com/>

##### Community Forums

* **Red Hat Customer Portal**: <https://access.redhat.com/discussions>
* **OpenShift Commons**: <https://commons.openshift.org/>
* **KubeVirt Community**: <https://kubevirt.io/community/>
* **CNCF Slack**: <https://slack.cncf.io/>

##### GitHub Organizations

* **Red Hat**: <https://github.com/redhat>
* **OpenShift**: <https://github.com/openshift>
* **KubeVirt**: <https://github.com/kubevirt>
* **Cilium**: <https://github.com/cilium>
* **Kyverno**: <https://github.com/kyverno>
* **Argo Project**: <https://github.com/argoproj>

#### Professional Services and Support

##### Red Hat Consulting

* **Services Overview**: <https://www.redhat.com/en/services/consulting>
* **OpenShift Consulting**: <https://www.redhat.com/en/services/consulting/openshift>

##### Partner Ecosystem

* **Red Hat Partner Directory**: <https://connect.redhat.com/en/partner-directory>
* **Certified Container Images**: <https://catalog.redhat.com/software/containers/explore>
* **Operator Hub**: <https://operatorhub.io/>

This comprehensive list provides direct access to all the resources, documentation, and tools needed for implementing and managing the RH OVE ecosystem.

## Best Practices

### Best Practices

#### Overview

This document outlines best practices for designing, deploying, and managing the multi-cluster RH OVE ecosystem, ensuring performance, security, and operational efficiency. This includes guidance on managing centralized services within the management cluster and distributing workloads across application clusters.

#### Multi-Cluster Architecture Best Practices

##### Cluster Design

* **Separation of Concerns**: Maintain clear separation between management and application clusters
* **Environment Isolation**: Use dedicated clusters for production, staging, and development
* **Resource Planning**: Size clusters appropriately for their intended workloads
* **Network Segmentation**: Implement proper network isolation between cluster environments

##### Management Cluster

* **High Availability**: Deploy management services with HA configuration
* **Resource Allocation**: Dedicate sufficient resources for centralized services
* **Backup Strategy**: Implement comprehensive backup for management cluster state
* **Security Hardening**: Apply strict security controls as this cluster manages the entire fleet

##### Application Clusters

* **Standardization**: Use consistent cluster configurations across environments
* **Agent Deployment**: Ensure proper deployment of management agents (ArgoCD, RHACS, monitoring)
* **Local Resources**: Optimize local resource allocation for workload requirements
* **Compliance**: Maintain consistent security and compliance postures

#### Architecture Best Practices

##### Namespace Design

* **Use Application Namespaces**: Segregate workloads by application or team-based namespaces for enhanced security and resource management
* **Environment Prefixes**: Use consistent naming conventions (e.g., prod-, staging-, dev-)
* **Label and Annotate**: Use consistent labeling and annotations for automation and policy application
* **Cross-Cluster Consistency**: Maintain similar namespace structures across clusters

##### Multi-Tenancy

* **RBAC Implementation**: Apply Role-Based Access Control to enforce access restrictions
* **Network Policies**: Utilize Cilium to enforce strict network policies between tenants
* **Resource Quotas**: Implement appropriate resource quotas per tenant/namespace
* **Policy Distribution**: Use centralized policy management with cluster-specific enforcement

#### Multi-Cluster GitOps Best Practices

##### Repository Structure

* **Centralized Repositories**: Use centralized Git repositories for all cluster configurations
* **Environment Branching**: Implement proper branching strategies for different environments
* **Application Separation**: Separate application definitions from infrastructure configurations
* **Policy as Code**: Store all policies and governance rules in version control

##### Deployment Strategies

* **Progressive Deployment**: Deploy to development, then staging, then production clusters
* **Automated Validation**: Implement automated testing and validation in CI/CD pipelines
* **Rollback Procedures**: Maintain clear rollback procedures for failed deployments
* **Change Management**: Implement proper change management processes for critical updates

#### Deployment Best Practices

##### Infrastructure Planning

* **Capacity Planning**: Assess resource needs well in advance and plan infrastructure accordingly
* **High Availability (HA)**: Configure HA for critical components and services
* **Cluster Sizing**: Right-size clusters based on workload requirements and growth projections
* **Geographic Distribution**: Consider geographic distribution for disaster recovery

##### Configuration Management

* **Infrastructure as Code (IaC)**: Use GitOps and Argo CD for configuration management and deployment consistency
* **Version Control**: Ensure all configurations and manifests are version controlled
* **Template Management**: Use Helm charts or Kustomize for template management
* **Secret Management**: Implement proper secret management across clusters

#### Security Best Practices

##### Network Security

* **Zero Trust Network**: Implement zero trust principles using Cilium’s microsegmentation and network policies.
* **Encryption**: Enforce encryption of data in transit and at rest.

##### Container and VM Security

* **Security Contexts**: Apply security contexts to restrict container capabilities and privileges.
* **Image Scanning**: Regularly scan container and VM images for vulnerabilities.

#### Operational Best Practices

##### Monitoring and Alerts

* **Comprehensive Monitoring**: Utilize tools like Dynatrace and Prometheus for end-to-end monitoring.
* **Alerting Systems**: Set up robust alerting and notification systems for proactive issue resolution.

##### Backup and Recovery

* **Regular Backups**: Schedule regular backups and test recovery procedures periodically.
* **Data Retention Policies**: Define and implement data retention and cleanup policies.

#### Continuous Improvement

##### Reviews and Audits

* **Performance Reviews**: Conduct regular performance reviews and optimizations.
* **Security Audits**: Perform periodic security audits and policy compliance checks.

##### Community Engagement

* **Stay Updated**: Engage with the community via forums and contribute to open-source projects.
* **Professional Development**: Encourage ongoing learning and certification for team members.

##### Documentation and Knowledge Sharing

* **Maintain Documentation**: Keep operational runbooks and architecture diagrams updated.
* **Knowledge Transfer**: Conduct regular training sessions and share lessons learned.

#### Conclusion

Adhering to these best practices ensures a well-architected, secure, and efficient RH OVE ecosystem that can adapt and scale with changing business needs.

## Sbom

### Software Bill of Materials (SBOM)

#### Overview

This document provides a comprehensive Software Bill of Materials (SBOM) for the RH OVE Multi-Cluster Ecosystem, consolidating all required software components, versions, and dependencies needed for successful deployment and operation.

#### Core Platform Components

##### Red Hat OpenShift Container Platform

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| OpenShift Container Platform | 4.12+ (recommended 4.14+) | Commercial | Kubernetes platform foundation | Red Hat |
| OpenShift CLI (oc) | Matches cluster version | Apache 2.0 | Command-line interface | Red Hat |
| OpenShift Web Console | Integrated with OCP | Commercial | Web-based management interface | Red Hat |

##### Virtualization Stack

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| OpenShift Virtualization Operator | 4.14+ | Commercial | VM management on OpenShift | Red Hat |
| KubeVirt | Latest (upstream) | Apache 2.0 | Kubernetes VM orchestration | KubeVirt Community |
| virtctl | Matches KubeVirt version | Apache 2.0 | VM command-line tool | KubeVirt Community |
| Containerized Data Importer (CDI) | Latest | Apache 2.0 | VM disk import/management | KubeVirt Community |
| libvirt | 7.0+ | LGPL 2.1+ | Virtualization API | Red Hat Enterprise Linux |
| QEMU/KVM | 6.0+ | GPL v2 | Hypervisor | Red Hat Enterprise Linux |

#### Networking Components

##### Container Network Interface (CNI)

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Cilium CNI | 1.12+ | Apache 2.0 | Primary network plugin with eBPF | Cilium |
| Cilium Operator | Matches Cilium version | Apache 2.0 | Cilium management operator | Cilium |
| Hubble | Integrated with Cilium | Apache 2.0 | Network observability | Cilium |
| Multus CNI | 3.8+ | Apache 2.0 | Multi-network support | Network Plumbing WG |
| SR-IOV Network Operator | 4.12+ | Apache 2.0 | High-performance networking | Red Hat |
| SR-IOV CNI | Latest | Apache 2.0 | SR-IOV network plugin | Network Plumbing WG |

##### Network Tools

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| iptables | 1.8+ | GPL v2 | Network filtering | Linux |
| eBPF | Kernel 4.14+ | GPL v2 | Network programming | Linux Kernel |
| OVS (Open vSwitch) | 2.15+ | Apache 2.0 | Virtual switching | Open vSwitch |

#### Security and Policy Management

##### Security Platforms

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Red Hat Advanced Cluster Security | Latest | Commercial | Security and compliance platform | Red Hat |
| Kyverno | 1.8+ | Apache 2.0 | Policy engine | Kyverno Community |
| Falco | 0.32+ | Apache 2.0 | Runtime security monitoring | CNCF |

##### Certificate and Identity Management

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| cert-manager | 1.10+ | Apache 2.0 | Certificate lifecycle management | CNCF |
| External Secrets Operator | 0.7+ | Apache 2.0 | Secret management | External Secrets |
| OpenShift OAuth | Integrated | Commercial | Authentication provider | Red Hat |

#### GitOps and Continuous Deployment

##### GitOps Platform

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Red Hat OpenShift GitOps | Latest | Commercial | GitOps platform based on Argo CD | Red Hat |
| Argo CD | 2.6+ | Apache 2.0 | GitOps continuous deployment | Argo Project |
| Argo Workflows | 3.4+ | Apache 2.0 | Workflow orchestration | Argo Project |
| Argo Rollouts | 1.4+ | Apache 2.0 | Progressive delivery | Argo Project |

##### Source Control Integration

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Git | 2.30+ | GPL v2 | Version control system | Git Community |
| GitHub/GitLab Webhooks | API v4+ | Various | Repository integration | GitHub/GitLab |

#### Monitoring and Observability

##### Metrics and Monitoring

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Dynatrace Operator | Latest | Commercial | Full-stack observability platform | Dynatrace |
| Prometheus | 2.40+ | Apache 2.0 | Metrics collection and storage | CNCF |
| Grafana | 9.0+ | AGPL v3 | Metrics visualization | Grafana Labs |
| AlertManager | 0.25+ | Apache 2.0 | Alert management | Prometheus |
| Node Exporter | 1.5+ | Apache 2.0 | Node metrics collection | Prometheus |
| kube-state-metrics | 2.7+ | Apache 2.0 | Kubernetes metrics | Kubernetes |

##### Logging

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| OpenShift Logging | 5.6+ | Commercial | Log aggregation platform | Red Hat |
| Elasticsearch | 7.17+ | Elastic License | Log storage and search | Elastic |
| Fluentd | 1.15+ | Apache 2.0 | Log collection and forwarding | CNCF |
| Kibana | 7.17+ | Elastic License | Log visualization | Elastic |

##### Distributed Tracing

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Jaeger | 1.40+ | Apache 2.0 | Distributed tracing | CNCF |
| OpenTelemetry Operator | 0.70+ | Apache 2.0 | Telemetry collection | CNCF |

#### Storage Solutions

##### Container Storage Interface (CSI)

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| AWS EBS CSI Driver | 1.15+ | Apache 2.0 | Block storage for AWS | Kubernetes |
| Azure Disk CSI Driver | 1.25+ | Apache 2.0 | Block storage for Azure | Kubernetes |
| GCE Persistent Disk CSI | 1.10+ | Apache 2.0 | Block storage for GCP | Kubernetes |
| Ceph CSI | 3.8+ | Apache 2.0 | Distributed storage | Ceph |
| NetApp Trident | 22.10+ | Apache 2.0 | Enterprise storage | NetApp |
| Dell CSI Driver | 2.8+ | Apache 2.0 | Dell enterprise storage | Dell Technologies |

##### Storage Management

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| OpenShift Data Foundation | 4.12+ | Commercial | Software-defined storage | Red Hat |
| Local Storage Operator | 4.12+ | Apache 2.0 | Local storage management | Red Hat |

#### Backup and Disaster Recovery

##### Backup Solutions

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Rubrik | Latest | Commercial | Enterprise backup platform | Rubrik |
| Velero | 1.10+ | Apache 2.0 | Kubernetes backup | VMware |
| OADP (OpenShift API for Data Protection) | 1.1+ | Apache 2.0 | Backup operator | Red Hat |

#### Multi-Cluster Management

##### Cluster Management

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Red Hat Advanced Cluster Management | Latest | Commercial | Multi-cluster management | Red Hat |
| Karmada | 1.6+ | Apache 2.0 | Multi-cluster orchestration | Karmada Community |
| Skupper | 1.2+ | Apache 2.0 | Application connectivity | Red Hat |

#### Development and Tooling

##### Command Line Tools

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| kubectl | Matches cluster version | Apache 2.0 | Kubernetes CLI | Kubernetes |
| helm | 3.10+ | Apache 2.0 | Package manager | CNCF |
| kustomize | 4.5+ | Apache 2.0 | Configuration management | Kubernetes |
| jq | 1.6+ | MIT | JSON processing | jq |
| yq | 4.30+ | MIT | YAML processing | yq |

##### Container Tools

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Podman | 4.3+ | Apache 2.0 | Container management | Red Hat |
| Buildah | 1.28+ | Apache 2.0 | Container image building | Red Hat |
| Skopeo | 1.10+ | Apache 2.0 | Container image operations | Red Hat |

#### Operating System Requirements

##### Base Operating System

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Red Hat Enterprise Linux CoreOS | 4.12+ | Commercial | Container-optimized OS | Red Hat |
| Red Hat Enterprise Linux | 8.6+ or 9.0+ | Commercial | General-purpose OS | Red Hat |

##### System Dependencies

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| systemd | 239+ | LGPL 2.1+ | System and service manager | systemd |
| Docker/Podman | 4.0+ | Apache 2.0 | Container runtime | Various |
| CRI-O | 1.25+ | Apache 2.0 | Container runtime | CRI-O |
| runc | 1.1+ | Apache 2.0 | Container runtime | OCI |

#### Integration and ITSM

##### ITSM Integration

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| ServiceNow | Latest | Commercial | ITSM platform | ServiceNow |
| ServiceNow MID Server | Latest | Commercial | Integration middleware | ServiceNow |

##### Event Management

| Component | Version | License | Purpose | Source |
| --- | --- | --- | --- | --- |
| Splunk | 8.2+ | Commercial | SIEM platform | Splunk |
| Elastic Security | 7.17+ | Elastic License | Security analytics | Elastic |

#### Hardware Requirements

##### Minimum Hardware Specifications

| Component | Requirement | Purpose |
| --- | --- | --- |
| CPU (Master Nodes) | 4+ cores per node | Control plane operations |
| Memory (Master Nodes) | 16GB+ per node | Control plane operations |
| Storage (Master Nodes) | 120GB+ per node | etcd and system data |
| CPU (Worker Nodes) | 8+ cores per node | Workload execution |
| Memory (Worker Nodes) | 32GB+ per node | VM and container workloads |
| Storage (Worker Nodes) | 500GB+ per node | Application data |
| Network | 10Gbps+ | High-performance networking |
| Virtualization | Intel VT-x/AMD-V | Hardware virtualization support |

#### Network Requirements

##### Port Requirements

| Port Range | Protocol | Purpose |
| --- | --- | --- |
| 6443 | TCP | Kubernetes API server |
| 22623 | TCP | Machine config server |
| 80/443 | TCP | HTTP/HTTPS ingress |
| 9000-9999 | TCP | Host level services |
| 10250-10259 | TCP | Kubernetes node ports |
| 30000-32767 | TCP | NodePort services |

#### License Summary

##### Commercial Licenses Required

* Red Hat OpenShift Container Platform
* OpenShift Virtualization
* Red Hat Advanced Cluster Security
* Red Hat Advanced Cluster Management
* Red Hat Enterprise Linux / CoreOS
* Dynatrace (monitoring platform)
* Rubrik (backup platform)
* ServiceNow (ITSM platform)

##### Open Source Components

* KubeVirt and related components (Apache 2.0)
* Cilium networking (Apache 2.0)
* Argo CD and GitOps tools (Apache 2.0)
* Kyverno policy engine (Apache 2.0)
* Prometheus monitoring stack (Apache 2.0)
* Various Kubernetes ecosystem tools (Apache 2.0)

#### Version Compatibility Matrix

##### Supported OpenShift Versions

| OpenShift Version | KubeVirt | Cilium | RHACM | RHACS |
| --- | --- | --- | --- | --- |
| 4.12.x | 4.12+ | 1.12+ | 2.7+ | 4.2+ |
| 4.13.x | 4.13+ | 1.13+ | 2.8+ | 4.3+ |
| 4.14.x | 4.14+ | 1.14+ | 2.9+ | 4.4+ |
| 4.15.x | 4.15+ | 1.15+ | 2.10+ | 4.5+ |

#### Security Considerations

##### CVE Monitoring

All components should be regularly updated to address security vulnerabilities. Subscribe to security advisories from:

* Red Hat Security Advisories
* CNCF Security SIG
* Individual project security lists
* National Vulnerability Database (NVD)

##### Supply Chain Security

* Verify image signatures for all container images
* Use Red Hat certified operators when available
* Implement image scanning in CI/CD pipelines
* Maintain software inventory and track dependencies

#### Maintenance and Updates

##### Update Frequency

* **Security patches**: As soon as available
* **Minor versions**: Monthly evaluation
* **Major versions**: Quarterly evaluation
* **OpenShift**: Follow Red Hat support lifecycle

##### End-of-Life Planning

Track EOL dates for all components and plan migrations:

* OpenShift: 18-month support lifecycle per version
* Kubernetes: 12-month support window
* Third-party components: Vendor-specific lifecycles

**Document Version**: 1.0  
**Last Updated**: 2025-08-04  
**Next Review**: 2025-11-04

This SBOM should be reviewed and updated quarterly or whenever significant changes are made to the RH OVE ecosystem architecture.

## Downloads

### Downloads and Exports

This page provides access to downloadable files and exports generated from the RH OVE Multi-Cluster Ecosystem documentation.

#### Documentation Exports

##### Complete Documentation

* [**RH OVE Complete Documentation (DOCX)**](../export/RH_OVE_Complete_Documentation.docx) - Comprehensive documentation in Microsoft Word format including all sections, diagrams, and technical specifications
* [**RH OVE Complete Documentation with Rendered Diagrams (DOCX)**](../export/RH_OVE_Complete_Documentation_Filtered.docx) - Enhanced version with natively rendered Mermaid diagrams using pandoc-mermaid-filter

##### Project Management Files

* [**Weekly Workload Breakdown (XLSX)**](../export/RH_OVE_Weekly_Workload_Breakdown.xlsx) - Detailed project timeline and resource allocation spreadsheet
* [**Project Charges (CSV)**](../export/project-charges.csv) - Budget breakdown and cost analysis in CSV format

#### Architecture Diagrams (Draw.io Format)

##### Core Architecture

* [**Global Overview**](../export/global-overview_advanced.drawio) - Multi-cluster architecture overview
* [**Single Cluster Overview**](../export/overview_advanced.drawio) - Individual cluster architecture
* [**Design Principles**](../export/design-principles_advanced.drawio) - Architecture design principles and patterns
* [**Network Architecture**](../export/network_advanced.drawio) - Network topology and connectivity
* [**Storage Architecture**](../export/storage_advanced.drawio) - Storage solutions and data management
* [**IAM Strategy**](../export/iam_advanced.drawio) - Identity and access management

##### ADR Diagrams

* [**ADR Table**](../export/adr-table_advanced.drawio) - Architecture decision records overview
* [**ADR-008 IAM Strategy**](../export/adr-008-iam-strategy_advanced.drawio) - IAM implementation decisions

##### Deployment & Management

* [**Prerequisites**](../export/prerequisites_advanced.drawio) - Deployment prerequisites flowchart
* [**Installation Guide**](../export/installation_advanced.drawio) - Installation process workflow
* [**Admission Control**](../export/admission-control_advanced.drawio) - Security and admission control
* [**GitOps Operations**](../export/gitops_advanced.drawio) - GitOps workflow and processes
* [**Monitoring**](../export/monitoring_advanced.drawio) - Monitoring and observability architecture

##### Operations

* [**Performance Tuning**](../export/performance_advanced.drawio) - Performance optimization strategies
* [**Troubleshooting**](../export/troubleshooting_advanced.drawio) - Troubleshooting workflows and decision trees

##### Use Cases

* [**Use Cases Table**](../export/use-cases-table_advanced.drawio) - Use cases overview and mapping
* [**VM Import & Migration**](../export/vm-importation_advanced.drawio) - Virtual machine migration processes
* [**VM Template Management**](../export/vm-template-management_advanced.drawio) - VM template lifecycle
* [**VM Scaling & Performance**](../export/vm-scaling-performance_advanced.drawio) - VM performance optimization
* [**VM Backup & Recovery**](../export/vm-backup-recovery_advanced.drawio) - Backup and disaster recovery
* [**Hybrid Applications**](../export/hybrid-applications_advanced.drawio) - Hybrid application deployment
* [**Multi-Environment Setup**](../export/setup-multi-env-application_advanced.drawio) - Multi-environment application deployment
* [**Database Services**](../export/database-services-paas_advanced.drawio) - PaaS database services
* [**Legacy Modernization**](../export/legacy-modernization_advanced.drawio) - Legacy system modernization
* [**Disaster Recovery**](../export/disaster-recovery_advanced.drawio) - Enterprise disaster recovery
* [**End-to-End Observability**](../export/end-to-end-observability_advanced.drawio) - Comprehensive monitoring solution
* [**WAF & Firewalling**](../export/waf-firewalling_advanced.drawio) - Web application firewall and security
* [**Events to CMDB/SIEM**](../export/publishing-events-to-cmdb-siem_advanced.drawio) - Event integration workflows

##### Project Planning

* [**Detailed Timeline**](../export/detailed-project-timeline_advanced.drawio) - Project timeline and milestones
* [**Home Page Diagram**](../export/index_advanced.drawio) - Main documentation overview

#### File Formats and Usage

##### DOCX Files

* **Purpose**: Complete documentation for offline reading, sharing, and printing
* **Software**: Microsoft Word, LibreOffice Writer, Google Docs
* **Best For**: Executive summaries, client presentations, offline documentation
* **Conversion Methods**:
  + Standard version: Docker-based mermaid-cli for diagram rendering
  + Filtered version: Native pandoc-mermaid-filter for enhanced diagram quality

##### XLSX Files

* **Purpose**: Project management data, timelines, and resource planning
* **Software**: Microsoft Excel, LibreOffice Calc, Google Sheets
* **Best For**: Project tracking, budget analysis, resource allocation

##### CSV Files

* **Purpose**: Data interchange and analysis
* **Software**: Any spreadsheet application, data analysis tools
* **Best For**: Data import, cost analysis, reporting

##### Draw.io Files (.drawio)

* **Purpose**: Editable architecture diagrams and flowcharts
* **Software**: [Draw.io](https://app.diagrams.net/), [Draw.io Desktop](https://github.com/jgraph/drawio-desktop)
* **Best For**: Diagram customization, architecture updates, visual documentation

#### How to Use These Files

##### Opening Draw.io Files

1. Visit [app.diagrams.net](https://app.diagrams.net/)
2. Click “Open Existing Diagram”
3. Upload the .drawio file from your downloads
4. Edit, customize, and export as needed

##### Viewing Documentation

* DOCX files can be opened directly in most word processors
* Use the table of contents for easy navigation
* All diagrams are embedded as high-resolution images

##### Project Management

* XLSX files contain detailed project timelines and resource allocations
* CSV files can be imported into project management tools
* Use filters and pivot tables for custom analysis

#### Version Information

* **Generated**: 2025-08-04
* **Documentation Version**: 1.1.0
* **Last Updated**: Based on latest documentation changes
* **Export Methods**:
  + Standard DOCX: task docs:export-docx
  + Enhanced DOCX: task docs:export-docx-filter

#### Support

For questions about these exports or to request additional formats, please: - Create an issue in the project repository - Contact the documentation team - Refer to the main documentation for detailed technical information

**Note**: All files are generated automatically from the source documentation. For the most up-to-date information, always refer to the online documentation.

## Tasks

### RH OVE Ecosystem - Task Management

This document describes the task management system for the RH OVE Ecosystem project.

#### Task Runner

This project uses Go’s **Task** (gotask) as a task runner to simplify common development and maintenance tasks.

##### Installing Task

If you don’t have task installed:

### macOS  
brew install go-task/tap/go-task  
  
### Linux  
sh -c "$(curl -ssL https://taskfile.dev/install.sh)"  
  
### Or via Go  
go install github.com/go-task/task/v3/cmd/task@latest  
  
### Or download from releases  
### https://github.com/go-task/task/releases

#### Global Tasks (Project Root)

From the project root directory, run:

task --list

##### Available Global Tasks

* task setup - Setup the entire project environment
* task clean - Clean all temporary files and caches
* task export-workload - Export workload data to XLSX
* task docs:build - Generate project documentation
* task docs:serve - Serve documentation locally
* task health-check - Run project health check
* task update - Update all dependencies
* task status - Show project status
* task init - Initialize a new development environment
* task ci - Run continuous integration checks

#### Scripts-Specific Tasks

From the scripts/ directory, run:

cd scripts  
task --list

##### Available Scripts Tasks

* task setup - Setup the scripts environment
* task install - Install dependencies
* task add <dep> - Add a new dependency (with prompt)
* task export-workload - Run the workload export script
* task export-workload-py - Run export script directly with Python
* task check - Check Python syntax
* task format - Format code
* task lint - Lint code
* task test - Run tests
* task clean - Clean Python cache files
* task update - Update dependencies
* task info - Show project information
* task dev-setup - Install development tools
* task dev - Run development workflow
* task build - Build/validate the project
* task watch - Watch for changes and run export
* task validate - Validate all aspects of the project

#### Quick Start

1. **Initial Setup:**

* task init

1. **Export Workload Data:**

* task export-workload

1. **Check Project Health:**

* task health-check

1. **Development Workflow (from root):**

* task scripts:dev-setup  
  task scripts:dev

1. **Watch for Changes (from root):**

* task scripts:watch

1. **Or work directly in scripts directory:**

* cd scripts  
  task dev-setup  
  task watch

#### File Structure

.  
├── Taskfile.yml # Global task definitions  
├── scripts/  
│ ├── Taskfile.yml # Scripts-specific task definitions  
│ ├── pyproject.toml # Python project configuration  
│ └── \*.py # Python scripts  
└── docs/  
 └── export/ # Generated export files

#### Advanced Features

##### Task Dependencies

Tasks can depend on other tasks:

tasks:  
 build:  
 deps:  
 - check  
 - lint  
 cmds:  
 - echo "Building..."

##### File Watching

Task supports file watching with sources and generates:

tasks:  
 export-workload:  
 sources:  
 - "{{.SOURCE\_FILE}}"  
 - export\_workload\_to\_xlsx.py  
 generates:  
 - "{{.EXPORT\_FILE}}"  
 cmds:  
 - uv run export-workload

##### Variables and Templating

Use variables for configuration:

vars:  
 PROJECT\_NAME: RH OVE Ecosystem  
 SCRIPTS\_DIR: scripts  
  
tasks:  
 info:  
 cmds:  
 - echo "Project: {{.PROJECT\_NAME}}"

#### Adding New Tasks

To add new tasks, edit the appropriate Taskfile.yml:

* Global tasks: Edit ./Taskfile.yml
* Scripts tasks: Edit ./scripts/Taskfile.yml

##### Example Task

tasks:  
 my-task:  
 desc: Description of what the task does  
 deps:  
 - other-task  
 sources:  
 - src/\*\*/\*.py  
 generates:  
 - dist/output.txt  
 cmds:  
 - echo "Running task..."  
 - command-to-run  
 - echo "Task complete!"

#### Dependencies

* task - Go-based task runner
* uv - Python package manager (for scripts)
* mkdocs - Documentation generator (optional)
* watchexec - File watcher (optional, for task watch)

## Glossary

### Glossary

#### Overview

This glossary provides definitions for key terms and concepts used throughout the RH OVE ecosystem documentation.

#### A

**Admission Control**

A Kubernetes mechanism that validates and mutates API requests before they are persisted to etcd. In RH OVE, this includes OpenShift built-in controllers, KubeVirt webhooks, and Kyverno policies.

**Ansible**

An open-source, agentless IT automation tool used for configuration management, application deployment, orchestration, and task automation across multiple systems.

**Argo CD**

A declarative GitOps continuous delivery tool for Kubernetes that automatically synchronizes applications with their desired state defined in Git repositories.

#### B

**Backup Policy**

A set of rules and schedules that define how, when, and what data should be backed up in the RH OVE environment, typically managed by Rubrik.

#### C

**CDI (Containerized Data Importer)**

A Kubernetes extension that provides facilities for enabling Persistent Volume Claims (PVCs) to be used as disks for KubeVirt VMs by importing, uploading, and cloning disk images.

**Cilium**

An open-source software for providing and transparently securing network connectivity and load balancing between application workloads using eBPF technology.

**CMDB (Configuration Management Database)**

A repository that acts as a data warehouse for IT installations, containing information about configuration items and their relationships, often integrated with ServiceNow.

**CNI (Container Network Interface)**

A specification and libraries for writing plugins to configure network interfaces in Linux containers, with Cilium being the recommended CNI for RH OVE.

**CRD (Custom Resource Definition)**

A Kubernetes extension mechanism that allows users to define custom resources that extend the Kubernetes API, extensively used in KubeVirt for VM management.

**CSI (Container Storage Interface)**

A standard for exposing arbitrary block and file storage systems to containerized workloads on Kubernetes, enabling storage vendors to develop plugins that work across different container orchestration systems.

#### D

**DataVolume**

A KubeVirt CRD that provides a declarative way to import, upload, and clone data into PVCs, serving as the primary storage mechanism for VM disks.

**Day-2 Operations**

Post-deployment operational activities including maintenance, monitoring, updates, scaling, and optimization of the RH OVE environment.

**Dynatrace**

An application performance monitoring and observability platform that provides full-stack monitoring for RH OVE environments.

#### E

**eBPF (extended Berkeley Packet Filter)**

A kernel technology that allows programs to run in kernel space without changing kernel source code or loading kernel modules, used by Cilium for high-performance networking.

**etcd**

A distributed, reliable key-value store used by Kubernetes to store all cluster data, providing a consistent and highly-available data store for cluster state.

#### G

**GitOps**

An operational framework that takes DevOps best practices used for application development and applies them to infrastructure automation, using Git as the single source of truth.

**Grafana**

An open-source platform for monitoring and observability that enables visualization, alerting, and exploration of metrics from multiple data sources including Prometheus, Elasticsearch, and others.

#### H

**Hugepages**

Large memory pages that can improve performance for memory-intensive applications by reducing memory management overhead in virtual machines.

**Hubble**

The network observability layer for Cilium that provides deep visibility into network flows, security policies, and performance metrics.

**Helm**

A Kubernetes package manager that helps you manage Kubernetes applications through charts, which are packages of pre-configured Kubernetes resources.

**HyperConverged**

A top-level CRD in OpenShift Virtualization that manages the deployment and configuration of all virtualization components.

#### I

**Ingress**

A Kubernetes API object that manages external access to services in a cluster, typically HTTP, providing load balancing, SSL termination, and name-based virtual hosting.

**Istio**

An open-source service mesh that provides a uniform way to secure, connect, and monitor microservices, offering traffic management, security, and observability features.

#### K

**KubeVirt**

An open-source Kubernetes add-on that enables running virtual machines alongside containers in a Kubernetes cluster, forming the foundation of OpenShift Virtualization.

**Karmada**

A Kubernetes management system that enables multi-cluster application management and provides centralized control plane for managing workloads across multiple Kubernetes clusters.

**Kyverno**

A policy engine designed for Kubernetes that validates, mutates, and generates configurations using admission controller webhooks and background scans.

#### M

**MacVLAN**

A Linux networking driver that allows creating multiple virtual network interfaces with different MAC addresses on a single physical network interface, commonly used with Multus for VM networking.

**Multus CNI**

A Container Network Interface (CNI) plugin that enables attachment of multiple network interfaces to pods and VMs in Kubernetes, allowing complex networking scenarios beyond single-network configurations.

#### N

**NAD (Network Attachment Definition)**

See NetworkAttachmentDefinition.

**NetworkAttachmentDefinition**

A CRD used by Multus that defines additional network interfaces for pods and VMs, enabling multi-network configurations beyond the default cluster network.

**Network Plumbing Working Group**

A Kubernetes community working group focused on developing networking enhancements, including Multus CNI and related multi-networking technologies.

**NUMA (Non-Uniform Memory Access)**

A computer memory design used in multiprocessing where memory access time depends on the memory location relative to the processor, important for VM performance tuning.

#### O

**OLM (Operator Lifecycle Manager)**

A component of the Operator Framework that helps users install, update, and manage the lifecycle of Kubernetes operators and their associated services.

**OpenShift Virtualization**

Red Hat’s enterprise virtualization solution that allows running virtual machines alongside containers on the same OpenShift platform.

#### P

**Prometheus**

An open-source systems monitoring and alerting toolkit with a dimensional data model, flexible query language (PromQL), efficient time series database, and modern alerting approach.

**PVC (Persistent Volume Claim)**

A request for storage by a user or application in Kubernetes, used extensively in RH OVE for VM disk storage.

#### Q

**QEMU Guest Agent**

A daemon that runs inside virtual machines to provide enhanced integration between the VM and the hypervisor, enabling better monitoring and management.

#### R

**RBAC (Role-Based Access Control)**

A method of restricting system access to authorized users based on their roles within an organization, fundamental to multi-tenant security in RH OVE.

**RH OVE (Red Hat OpenShift Virtualization Engine)**

Red Hat’s solution for running virtual machines on OpenShift, based on the upstream KubeVirt project.

**Rubrik**

An enterprise data management platform that provides backup, recovery, and data protection services, certified for integration with RH OVE.

#### S

**ServiceNow**

An IT service management platform that provides CMDB functionality and can be integrated with RH OVE for automated configuration tracking.

**SR-IOV (Single Root I/O Virtualization)**

A specification that allows efficient sharing of PCIe devices between virtual machines, enabling high-performance networking for VMs.

#### T

**Tekton**

A Kubernetes-native open-source framework for creating continuous integration and delivery (CI/CD) systems, allowing developers to build, test, and deploy applications.

**Terraform**

An open-source infrastructure as code tool that allows users to define and provision data center infrastructure using a declarative configuration language.

#### V

**VirtualMachine (VM)**

A KubeVirt CRD that represents a virtual machine definition, including CPU, memory, storage, and network configurations.

**VirtualMachineInstance (VMI)**

A KubeVirt CRD that represents a running virtual machine instance, showing the actual runtime state of a VM.

**VirtualMachineInstanceReplicaSet**

A KubeVirt CRD that ensures a specified number of VMI replicas are running, similar to Kubernetes ReplicaSets for pods.

**virtctl**

A command-line tool for managing KubeVirt virtual machines, providing functionality to start, stop, console access, and manage VMs.

**VLAN (Virtual Local Area Network)**

A network configuration that enables the logical partitioning of a physical network into multiple broadcast domains, improving security and network management.

**VPC (Virtual Private Cloud)**

A logically isolated section of a cloud provider’s infrastructure where users can launch resources in a virtual network that they define.

#### W

**WebAssembly (WASM)**

A binary instruction format for a stack-based virtual machine that enables high-performance applications on web browsers and server environments, increasingly used for cloud-native applications.

#### Common Acronyms

* **ADR**: Architecture Decision Record
* **API**: Application Programming Interface
* **CDI**: Containerized Data Importer
* **CI/CD**: Continuous Integration/Continuous Deployment
* **CMDB**: Configuration Management Database
* **CNI**: Container Network Interface
* **CPU**: Central Processing Unit
* **CRD**: Custom Resource Definition
* **CSI**: Container Storage Interface
* **DNS**: Domain Name System
* **HA**: High Availability
* **IAM**: Identity and Access Management
* **I/O**: Input/Output
* **IOPS**: Input/Output Operations Per Second
* **ITSM**: IT Service Management
* **JSON**: JavaScript Object Notation
* **LDAP**: Lightweight Directory Access Protocol
* **NFS**: Network File System
* **OAuth**: Open Authorization
* **OIDC**: OpenID Connect
* **OLM**: Operator Lifecycle Manager
* **RBAC**: Role-Based Access Control
* **REST**: Representational State Transfer
* **SAML**: Security Assertion Markup Language
* **SIEM**: Security Information and Event Management
* **SLA**: Service Level Agreement
* **SSD**: Solid State Drive
* **TLS**: Transport Layer Security
* **VLAN**: Virtual Local Area Network
* **VM**: Virtual Machine
* **VMI**: Virtual Machine Instance
* **VPC**: Virtual Private Cloud
* **WAF**: Web Application Firewall
* **WASM**: WebAssembly
* **YAML**: YAML Ain’t Markup Language

This glossary provides essential terminology for understanding and working with the RH OVE ecosystem. Terms are regularly updated as the technology and documentation evolve.