WHITEPAPER

RED HAT OPENSHIFT CONTAINER PLATFORM APPLICABILITY GUIDE FOR ISO/IEC 27001:2013

TO ASSIST CUSTOMERS WITH APPLICABILITY OF OPENSHIFT CONTAINER PLATFORM TO ISO/IEC 27001:2013

BYRON ESTRADA | SEC+
CHRIS KRUEGER | CISSP
AL MAHDI MIFDAL | ISO SME

FINAL V2.1

Red Hat

COALFIRE

North America | Europe
877.224.8077 | info@coalfire.com | Coalfire.com

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Red Hat, Inc. (Red Hat) delivers a portfolio of products and services built from open source software components using a subscription and support model. Red Hat engaged Coalfire, a cybersecurity engineering, advisory, and assessment company, to conduct an independent technical assessment of Red Hat OpenShift Container Platform (OCP) 4.4 on Red Hat Enterprise Linux CoreOS (RHEL CoreOS), referred to as RHCOS in specific sections throughout the whitepaper.

For a broader understanding of the requirements and their applicability to technical solution implementation, Coalfire also reviewed supporting documentation from Coalfire's International Organization for Standardization (ISO) team (CFISO), including implementation and assessment guidance. CFISO is an ISO/International Electrotechnical Commission (IEC) 27001:2013 and ISO 9001:2015 certification body accredited by both the American National Standards Institute-American Society for Quality (ANSI-ASQ) National Accreditation Board (ANAB) and the United Kingdom Accreditation Service (UKAS). CFISO provides ISO/IEC 27001:2013 and ISO 9001:2015 audit and certification services to clients, utilizing the framework required in the ISO 17021-1:2015 and ISO 27006 standards.

The purpose of this product applicability guide is to identify the alignment of OCP on RHEL CoreOS to the ISO/IEC 27001 standards published in 2013. ISO/IEC 27001:2013 Information technology – Security techniques – Information security management system – Requirements specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system (ISMS) within the context of the organization. It also includes requirements for the assessment and treatment of information security risks tailored to the organization's needs.

Coalfire assessed control capabilities applicable to OCP on RHEL CoreOS for ISO/IEC 27001:2013 requirements with guidance for control implementation provided by ISO/IEC 27002:2013 Information technology – Security techniques – Code of practice for information security controls. The findings provided in this product applicability guide do not intend to claim conformity to ISO/IEC 27001:2013. Each organization is individually responsible for conforming to the standard to address all ISO requirements.

Containerization can provide benefits to businesses that incorporate it into their service development and delivery model. Some of these benefits may include increased developer productivity; a decrease in time to application deployment; increased application portability, agility, and scalability to align with changes in service demand; and increased compute efficiencies.

OCP is a container platform that natively integrates open-source Linux container technologies and Kubernetes to combine them into an enterprise solution running on RHEL CoreOS. OCP provides an application programming interface (API), web interface, and command-line interface (CLI) to manage the underlying container technologies and Kubernetes to help users orchestrate the creation and management of containers. OCP provides self-service build and deployment automation for containers in addition to operational container features, including scaling, monitoring, and management capabilities.

From a packaging perspective, the primary advantage of containerization is that applications in container images can be packaged with only their code and dependency requirements instead of other technologies, like virtualization, where the entire operating system (OS) must be included in the image.

This product applicability guide may be useful for organizations that want to utilize container technologies within the framework of an ISO program of compliance. This guide discusses relevant ISO/IEC 27001:2013 requirements that apply to OpenShift on RHEL CoreOS. This paper's focus is on technical controls that are pertinent to and in alignment with OCP capabilities.

COALFIRE OPINION

Security controls, features, and functionality built into OCP on RHEL CoreOS can support or address relevant technical ISO/IEC 27001:2013 requirements. OCP can provide granular control and improved security at scale for containerized workloads.

UNDERSTANDING ISO/IEC 27001:2013

ISO/IEC 27001:2013 is a globally recognized standard for the establishment and certification of an organization's ISMS. The framework establishes processes for organizations to implement, monitor, operate, maintain, and continually improve an ISMS in accordance with the organization's cyber risk tolerance to help organizations secure financial information, intellectual property, employee information, or information entrusted to third parties. ISO/IEC 27001:2013 conformance can be frequently leveraged for other compliance efforts, including, but not limited to, General Data Protection Regulation (GDPR), Payment Card Industry Data Security Standard (PCI DSS), Health Insurance Portability and Accountability Act (HIPAA), and Sarbanes-Oxley (SOX).

The ISO/IEC 27001:2013 standard is divided into two parts. Annex A focuses on the ISMS design within the context of the continuous improvement cycle through clauses 4 - 10. Annex A is comprised of 114 control objectives divided into 14 categories (e.g., human resources [HR], security, cryptography, access control). Figure 1 provides a high-level illustration of the two sections of the standard.

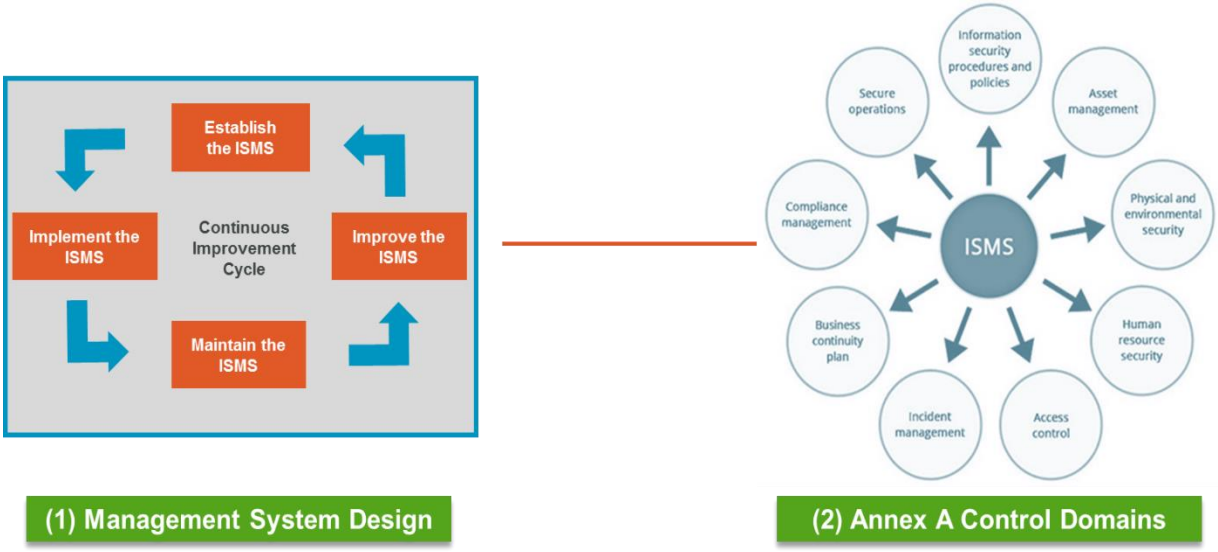

Figure 1: High-Level ISO/IEC 27001:2013 Standard

ISO/IEC 27001:2013 uses a top-down, risk-based approach to security that is technology neutral. The first section focuses on the ISMS establishment, implementation, maintenance, and continuous improvement within the organization's context. Annex A (normative) provides a reference for control objectives and controls. This paper focuses on the assessed technology capabilities to address ISO/IEC 27001:2013 controls and control objectives. This paper does not make any claims against the management system design, as no actual organization was assessed.

OPENSHIFT CONTAINER PLATFORM

OpenShift offers a consistent hybrid cloud foundation for building and scaling containerized applications. OCP delivers a single, consistent Kubernetes platform anywhere that RHEL CoreOS runs. OCP comes

with a three-year enterprise support lifecycle from a Kubernetes contributor. It is an enterprise-grade application platform built for containers with Kubernetes. It is an integrated platform to run, orchestrate, monitor, and scale containers. OCP provides both streamlined automated installations and options for more customized installations where organizations have requirements not met by the automated defaults. OCP allows organizations to control, defend, and extend the application platform throughout an application's lifecycle. It enables a secure software supply chain to make applications more secure.

OCP helps maximize developer productivity with specially configured toolsets and tools. One of these tools is a workflow that includes built-in Continuous Integration/Continuous Delivery (CI/CD) pipelines and a source-to-image capability that enables developers to go directly from application code to container. These updated toolsets help provide consistent operations and management experience across infrastructures and in support of many teams.

OPENSHIFT CONTAINER PLATFORM ARCHITECTURE

The following is a list of components and roles that support OCP:

- **RHEL CoreOS** – OCP uses RHEL CoreOS as a container-optimized operating system specifically configured and optimized for running the platform. RHEL CoreOS is installed and managed as part of OpenShift, which is similar to an appliance.

  RHEL CoreOS is a single-purpose, container- and Kubernetes-optimized, minimal-footprint OS powered by the same binaries as RHEL. This pre-hardened OS can help organizations meet requirements for system hardening with the least functionality through its lightweight, purpose-built nature; it only includes the necessary features, functions, and services to host containers in an OCP environment.

  RHCOS is designed to be more tightly managed than a default RHEL installation. Management is performed remotely from the OCP cluster. When the OCP cluster is setup and RHCOS is deployed, customers can only modify a few system settings.

  RHEL and RHEL CoreOS have built-in security features and functionality that, as configured in an OCP installation, provide a secure platform for supporting the OCP components and the workloads in containers that OCP orchestrates. While the option exists to utilize RHEL for worker nodes managed by OCP, there are additional security benefits for using RHEL CoreOS, including a reduced attack surface, pre-hardened OS, and automated updates. Customers should consider their use cases when determining the desired OCP architecture.

- **Operating Environment** – OpenShift can be deployed on bare-metal physical hardware, VMware vSphere, Red Hat Virtualization (RHV), OpenStack, or other major cloud providers. It can be deployed on private or certified public cloud environments, depending on the organization's specific use cases.

- **Open Container Initiative (OCI) Runtime** – Container Runtime Interface-Orchestration (CRI-O) is an OCI-compatible runtime installed on every RHEL CoreOS host. As such, CRI-O enables the use of OCI-compatible containers. OCI has an open governance structure used to create open industry standards around container formats and runtimes. The CRI-O engine focuses on features needed by Kubernetes platforms, such as OCP, and offers specific compatibility with different Kubernetes versions.

- **Kubernetes** – Kubernetes is an open-source container orchestration engine for automating the deployment, scaling, scheduling, and management of containerized applications across the cluster. Kubernetes provides orchestration for complex multi-container services – serving as the de facto standard for orchestrating containers.

- **Containers** – End-user application instances, application components, or other services are run in Linux containers. The container only includes the necessary libraries, functions, elements, and code required to run the application.
- **Pods** – While application components run in containers, OCP orchestrates and manages pods. A Kubernetes pod is a group of containers that are deployed together on the same host. A pod is an orchestrated unit in OCP made up of one or more containers. A pod should only contain a single function, such as an application server or web server, and should not include multiple functions, such as both a database and application server.
- **Operators** – An Operator is a method of packaging, deploying, and managing a Kubernetes- native application. Operators automate the lifecycle management of containerized applications within Kubernetes. A Kubernetes-native application is an application that is both deployed on Kubernetes and managed using the Kubernetes APIs and kubectl tooling. A controller is a core concept of Kubernetes. It is implemented as a software loop that runs continuously on the Kubernetes master nodes, compares, and, if necessary, reconciles the expressed desired state and the current state of an object. Objects are well-known resources like Pods, Services, ConfigMaps, or PersistentVolumes. Operators apply the model of controller at the level of entire applications and are, in effect, application-specific custom controllers.

  OCP adds developer- and operations-centric tools to Kubernetes that help enable rapid application development, simplified deployment and scaling, and long-term lifecycle maintenance for applications. OCP also leverages integrated components to automate application builds, deployments, scaling, and health management. Included in the automation capabilities of OCP is the ability to configure and deploy Kubernetes container host clusters.

OpenShift Container Platform Components

The following components are specific to OpenShift.

- **OpenShift Operators** – As OCP is a fully containerized platform that consists of many different components, OCP takes advantage of Operators for driving the installation, configuration, management, and upgrades of OCP and all its services. Operators are both the fundamental unit of the OpenShift cluster and a way to deploy applications and software components for the OCP customer's applications to use, including the Kubernetes core services along with monitoring (e.g., Prometheus), logging (e.g., Elasticsearch, Fluentd, Kibana), software-defined networking, storage, registry, and other components that make up the OCP Kubernetes platform. Operators serve as the platform foundation that removes the need for manual upgrades of the OSs and OCP control plane applications. The Cluster Version Operator and Machine Config Operator allow simplified cluster-wide management of those critical components. All the components of the platform are managed throughout their lifecycle with Operators.
- **Operator Lifecycle Manager** – The Operator Lifecycle Manager (OLM) is the backplane that facilitates the management of Operators on a Kubernetes cluster. OLM helps administrators of the cluster control which Operators are available in which namespaces, and who can interact with the running Operators. The permissions of an Operator are configured automatically to follow a least-privilege approach. The OLM helps users install, update, and manage the lifecycle of all Operators and their associated services running across their clusters. The OLM runs by default in OCP 4, which aids administrators in installing, upgrading, and granting access to Operators running on their cluster.

- **Machine Config Operator (MCO)** – The MCO manages OS updates and OS configuration changes. The MCO, allows platform administrators to ensure consistent configuration across all RHEL CoreOS nodes, monitoring for drift, and alerting on unsupported configurations.
- **OpenShift Nodes** – Nodes are instances of RHEL with the OpenShift software deployed. Nodes are where end-user applications are run in containers. Nodes will contain the necessary OCP node daemon, the Kubelet, the container runtime, and other services required to support the hosting of containers. Most of the software components that run above the OS (e.g., the software-defined network [SDN] daemon) run in containers on the Node.
- **OpenShift Master** – The Master is the control plane for OpenShift. The Master maintains and understands the state of the environment and orchestrates all activity that occurs on the Nodes. Similarly to Nodes, the OCP Master is run on RHEL. While the Master is technically also a Node and can participate in the software-defined network, for separation of function, the OCP Master should not be scheduled to run application instances (pods). The following are the four functions of the OCP Master:
  - **API and Authentication** – The Master provides a single API that all tooling and systems interact with. Everything that interacts with OCP must go through this API. All API requests are Secure Sockets Layer (SSL) encrypted and must be authenticated. Authorizations are handled by fine-grained role-based access control (RBAC). It is recommended to tie the Master to an external identity and access management system using Lightweight Directory Access Protocol (LDAP), OAuth, or other providers. The Master evaluates requests for both authentication (AuthN) and authorization (AuthZ). Users of OCP who have been granted access can be authorized to work with specific projects.
  - **Desired and Current State** – The state of OpenShift is held in the OCP data store. The data store uses etcd, a distributed key-value store. The data store houses information about the OCP environment including OCP user account information and the RBAC rules, the OpenShift environment state, application environment information and important environment variables, secrets data, and other information.
  - **Scheduler** – The scheduler determines pod placement within OCP. It uses a combination of configuration and environment state (CPU, memory, and other environmental factors) to determine the best fit for running pods across the Nodes in the environment. The scheduler is configured with a simple JSON file in combination with Node labels to carve up OCP resources. This allows the placement of pods within OCP to be based on the real-world topology, making use of concepts such as regions, zones, or other constructs relevant to the enterprise. These factors can contribute to the scheduled placement of pods in the environment and can ensure that pods run on appropriate Nodes associated with their function.
  - **Health and Scaling** – The OCP Master is also responsible for monitoring the health of pods and scaling the pods as desired to handle additional load. The OCP Master executes liveness and readiness tests using probes that are defined by users. The OCP Master can detect failed pods and remediate failures as they occur.
- **Service Layer** – The OCP Service Layer helps application components more easily communicate with one another. For instance, a front-end web service containing multiple web servers would connect to database instances by communication via the database service. OCP automatically and transparently handles load balancing across the services' instances. In conjunction with probes, the OCP Service Layer ensures that traffic is only directed toward healthy pods, maintaining component availability.

- **Persistent Storage** – Linux containers are natively ephemeral and only maintain data for as long as they are running. Applications or application components may require access to a long-term persistent storage repository, required for a database engine. OpenShift provides the means to connect pods to real-world external storage, which allows for stateful applications to be used on the platform. Persistent storage types that are usable include iSCSI, Fiber Channel, and NFS and cloud-type storage and software-defined storage options such as Red Hat OpenShift Container Storage. Persistent storage can be dynamically provisioned upon the user's request, provided the storage solution has integration with OCP.
- **Ingress Controller** – The Ingress Controller is commonly used to allow external access to an OCP cluster. The Ingress Operator manages Ingress Controllers. An Ingress Controller is configured to accept external requests and proxy them based on the configured routes. External requests are limited to HTTP and HTTPS using Server Name Indication (SNI) and Transport Layer Security (TLS) using SNI, which is enough for web applications and services that work over TLS with SNI. Ingress works in partnership with the service layer to provide automated load balancing to pods for external clients. The Ingress Controller uses the service endpoint information to determine where to route and load balance traffic; however, it does not route traffic through the Service Layer.
- **OpenShift SDN** – The OCP software-defined network (SDN) is a unified cluster network that enables the communication between pods across the OCP cluster. The OCP SDN configures an overlay network that uses Open vSwitch (OVS). Red Hat currently provides one SDN mode for the OCP pod network – the networkpolicy mode. The network policy mode provides fine-grained access control via user-defined rules. Network policy rules can be built in a mandatory access control (MAC) style, where all traffic is denied by default unless a rule explicitly exists, even for pods/containers on the same host. The network policy mode is the default mode.
- **OpenShift Registry** – The OpenShift Registry provides integrated storage and management for sharing container images. OpenShift can utilize existing OCI-compliant container registries accessible to the Nodes and the OpenShift Master via the network.

Figure 2 below provides a high-level OpenShift Container Platform Overview.

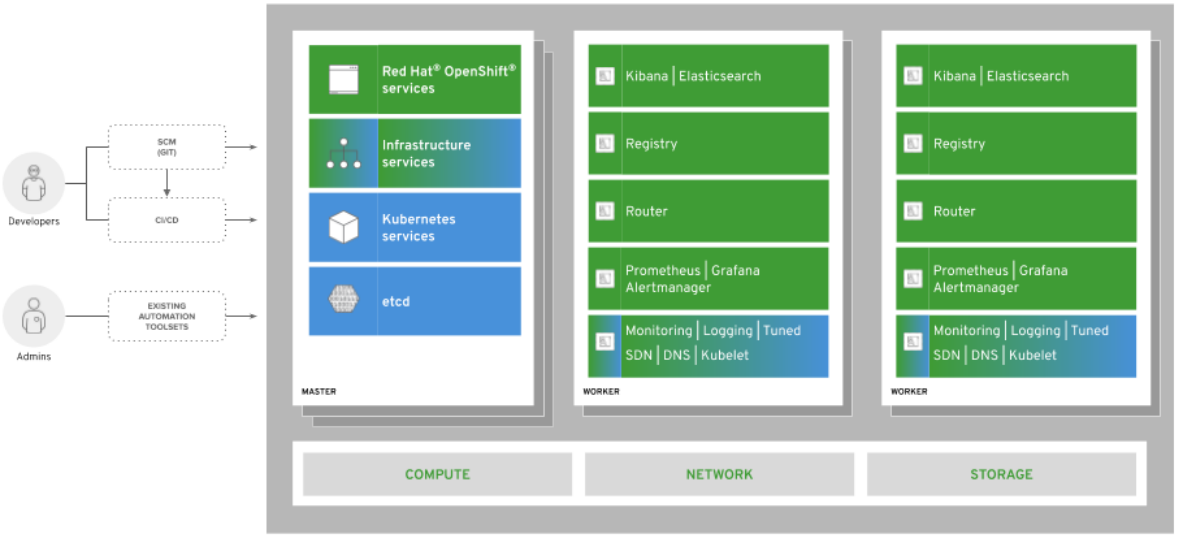

Figure 2: High-level OpenShift Container Platform Overview

Types of Users

When it comes to direct use and management of an OpenShift cluster, regular users (who typically run workloads and may do some administration) and system users (who can interact with the API), and service accounts used for automation purposes.

- **Users** – User (operators, developers, application administrators) access to OCP is provided through standard interfaces, including the Web UI, CLI, and IDEs. These interfaces go through the authenticated and RBAC-controlled API. Users do not require system-level access to any of the OCP hosts, even for complex application debugging and troubleshooting tasks.

  Three types of users can exist in an OCP environment: regular users, system users, and service accounts.

  - **Regular Users:** A user most commonly represents a real person who interacts with the OCP cluster in some way. Since the OCP management is performed on the API, this includes both administrators of the cluster and regular cluster users who run their workloads.
  - **System Users:** Many of the system users are created automatically when the OCP infrastructure is defined to enable the infrastructure to interact with the API securely. System users include a cluster administrator (with access to everything), a per-node user, users for use by ingress controllers and registries, and various others. There is also an anonymous system user that is used by default for unauthenticated requests. Examples of these users are system:admin, system:openshift-registry, and system:node:node1.example.com.
  - **Service Accounts:** These are non-human system users, often associated with projects and used for API access in automation situations. Some default service accounts are created and associated when a project is first created. Project and cluster administrators can create additional service accounts for defining access to the contents of each project.
- **Projects** – A project is a Kubernetes namespace with additional OCP annotations and metadata. It is the central vehicle by which access to regular users' resources is managed and is the tenancy model of OCP and Kubernetes. A project allows a community of users to organize and manage their content in isolation from other communities.

For more information on OpenShift concepts, features, and functions, please refer to Red Hat's product documentation; links are provided in this paper's references section.

OPENSHIFT CONTAINER PLATFORM SECURITY

OCP enables continuous security with defense-in-depth capabilities and Red Hat's secured software supply chain. Security controls can be applied dynamically to the platform and the applications the platform supports. This allows security controls to keep up with the scale and agility of applications deployed on the platform. OCP runs on RHEL CoreOS and makes heavy use of the existing security features built into the OS. Red Hat manages the OS packages and provides a trusted distribution of content. The security of OCP includes and utilizes hardened technologies such as Security-Enhanced Linux (SELinux); process, network, and storage separation; proactive monitoring of capacity limits (e.g., CPU, disk, memory); and encrypted communications for infrastructure support including Secure Shell (SSH) and SSL. Additionally, OCP provides integration with third-party identity management solutions to support secure authentication and authorization options aligned with organization compliance requirements.

Container and host security capabilities are derived from four major areas:

- Linux namespaces create a partitioning of system resources, forming the logical boundaries around the container's structure and segregating the view of the process table, host file system, and network.

- Secure computing (seccomp) features provide a way to filter system call availability within a container.
- Linux capabilities allow the reduction of privileges that would normally be permitted within the root user context.
- SELinux enforces mandatory access control for system processes, services, files, and network resources.

OCP runs on RHEL and makes heavy use of the existing security features built into the OS. Red Hat manages the OS packages and provides a trusted distribution of content. The security of OpenShift includes and utilizes hardened technologies such as SELinux; process, network, and storage separation; proactive monitoring of capacity limits (CPU, disk, memory, etc.); and encrypted communications for infrastructure support including SSH and SSL. Additionally, OCP provides integration with third-party identity management solutions to support secure authentication and authorization options aligned with organization compliance requirements.

Regarding security for configurations, operators will reset unsupported changes to supported configurations, or in the case of the MCO, it will mark the node as degraded.

The following is a high-level list of OpenShift security features and capabilities:

- **Identity and Access Management** – OCP includes an embedded OAuth server for token-based authentication. Red Hat recommends using a supported third-party identity and authentication provider for centralized management of user identities and authenticators. OCP supports multiple identity providers, including HTPasswd, Keystone, LDAP, basic authentication, request header, GitHub, GitLab, Google, and OpenID.

  With a token-based authentication system, users obtain an OAuth access token to authenticate themselves to the API. When a user requests a new OAuth token, the OAuth server uses the configured identity provider to determine the person's identity making the request. The server then determines what user the identity maps to, creates an access token for that user, and returns the token for use. Users can use bound service account tokens, which improves the ability to integrate with cloud provider identity access management (IAM) services, such as Amazon Web Services (AWS) IAM.
- **Role-Based Access Control (RBAC)** – An OCP user object represents an actor that may be granted permissions in the system by adding roles to the users or their groups. OCP is configured to use RBAC, allowing for a granular determination of access for users or groups of users. RBAC objects determine whether a user can perform a given action on the system, based on the user's assigned roles. This allows platform administrators to use cluster roles and bindings to control who has various access levels to OCP itself, and all contained OCP projects and resources. This also allows developers to use local roles and bindings to control who has access to their projects. The authorization process is managed using rules, roles, and bindings. Rules are a set of permitted verbs (actions) on a set of objects, such as whether something or someone can create pods. Roles are a collection of rules. Users and groups can be associated with roles or bound to multiple roles at the same time. Bindings are associations between users or groups with a role.

Figure 3 illustrates the relationship between projects and role binding. Users or groups can only view the content in their assigned projects and where they are assigned specific roles (role binding).

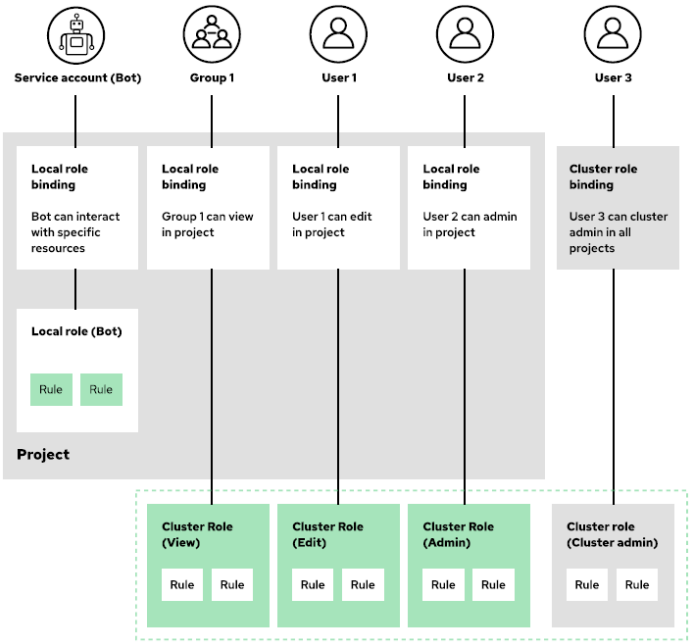

Figure 3: RBAC with OpenShift Projects

- **Encryption** – Data in motion and data at rest is protected by encryption. OCP control plane components enforce industry standards, including HTTP/2 defaults and TLS 1.2 or TLS 1.3. Certificate key sizes are not configurable (RSA certificates are 2048, ECDSA for kubelet certificates). In OCP, all traffic between the API server and the worker nodes is encrypted, ensuring that secrets stored in etcd and transmitted to pods are encrypted in transit. Additional protection for secrets at-rest can be provided by encrypting the RHEL CoreOS disks and the etcd datastore. The AES-CBC cipher is used to encrypt the datastore.
- **Federal Information Processing Standard (FIPS)** – To use FIPS validated components with OCP, RHEL CoreOS nodes must be configured for FIPS mode at the installation time. FIPS mode ensures that the OCP components call only FIPS cryptographic modules. Custom containers built with RHEL or UBI base images can be configured for FIPS compliance. RHEL 8 and RHEL CoreOS cryptographic libraries have the status of Modules in Process per the NIST Computer Security Resource Center (CSRC).
- **Cluster Logging** – The OCP cluster administrators can deploy cluster logging using a few CLI commands and the OCP web console to install the Elasticsearch Operator and Cluster Logging Operator. The cluster logging components are based upon Elasticsearch, Fluentd, and Kibana (EFK). OCP uses Elasticsearch (ES) by default to store log data. RBAC controls are applied to ES that enabled controlled access of the logs to the developers.

  The cluster logging Elasticsearch instance is optimized and tested for short-term storage (i.e., seven days). For administrators wanting to retain their logs over a longer-term, it is recommended they move the data to a third-party storage system.
- **Audit** – Audit provides a security-relevant chronological set of records documenting the sequence of activities that have affected the system by individual users, administrators, or other system components. Audit works at the API server level, logging all requests coming to the server. Each audit log contains several fields of information that provide an in-depth view of each log's behavior. An identity can be used to audit what actions a user has performed during a specific point in time. The host operating system audits consist of standard auditing capabilities in RHEL and RHEL CoreOS.

- **Container Host and Platform Multitenancy** – RHEL can manage multitenancy for the container runtime by using Linux namespaces, SELinux, CGroups, and Secure Computing Mode (seccomp) to isolate and protect containers, which can be useful for maintaining separation for workloads of differing classifications.
- **Container Content** – The Red Hat Container Catalog delivers validated application content from Red Hat and certified ISV partners.
- **OperatorHub** – The Operator Hub provides access to certified Kubernetes operators.
- **Container Registries** – OpenShift includes an integrated container registry that provides basic functionality supporting build and deployment automation within the cluster, tied into the OCP RBAC. Within the context of an organization needing to adhere to FISMA Moderate requirements, Red Hat Quay is an additional product that provides a registry with capabilities for both RBAC and the vulnerability scanning of applications and software in images and more.
- **Building Containers** – OCP integrates tightly with Jenkins and can be easily integrated with other CI/CD tools to manage builds, code inspection, code scanning, and validation. OCP includes Tekton for building Kubernetes native pipelines.
- **Deploying Containers** – By default, OCP prevents containers from running as root or other specifically-named users. Also, OCP enables granular deployment policies that allow operations, security, and compliance teams to enforce quotas, isolation, and access protections.
- **Container Orchestration** – OCP integrates secure operational capabilities to support trust between users, applications, and security policies. OpenShift can be used to guide how and where containers can be deployed, deployment of containers based on the availability of capacity, how containers can access each other based on least privilege, how access to shared resources and management is controlled, how container health is monitored, how the applications can be scaled automatically to meet a need, and can enable developer self-service while maintaining and meeting designed security requirements.
- **Network Isolation** – OCP uses an SDN approach to provide a unified cluster network that enables the communication between pods across the OpenShift cluster. OCP uses the Multus CNI plug-in to allow chaining of CNI plug-ins. This Pod network is established and maintained by the OpenShift SDN, configuring an overlay network using Open vSwitch (OVS). The NetworkPolicy SDN mode is available from Red Hat for the customer to configure network policies. Other third-party SDN solutions exist that are capable of being integrated into OpenShift as well.

  Multus is a multi-CNI plug-in to support the Multi-Networking feature in Kubernetes using customer resource definition (CRD) based network objects in Kubernetes. During cluster installation, the customer configures the default pod network. The default network handles all ordinary network traffic for the cluster. The customer can then define additional networks based on the available CNI plug-ins and attach one or more of these networks to their pods. The customer can define more than one additional network for their cluster, depending on their needs. This gives the customer flexibility when configuring Pods that deliver network functionality, such as switching or routing.

  The capability to create additional networks enables the customer to enhance network isolation, including data plane and control plane separation. The customer can send sensitive traffic onto a network plane managed specifically for security considerations and can separate private data that must not be shared between tenants or customers.
- **Secure the Data** – OpenShift provides access to and integration with a broad range of storage platforms and protocols, allowing applications to store and encrypt application data securely.

OpenShift provides the option to encrypt sensitive data stored in etcd and to encrypt the RHEL CoreOS volumes.

- **Service Mesh** – An optional feature of OCP is Red Hat OpenShift Service Mesh. This provides a platform for behavioral insights and operational control over the networked microservices in a service mesh. A service mesh is a network of microservices that make up applications in a distributed microservice architecture and the interactions between those microservices. When a service mesh grows and complexity, it can become harder to understand and manage. With Red Hat Service Mesh, the customer can connect, secure, and monitor microservices in the OCP environment. Red Hat OpenShift Service Mesh adds a transparent layer on existing distributed applications without requiring any changes to the service code. The customer can add Red Hat OpenShift Service Mesh support to services by deploying a special sidecar proxy to relevant services in the mesh that intercepts all network communications between microservices. The service mesh is configured and managed using the control plane features.

  The Red Hat OpenShift Service Mesh gives customers a way to create a network of deployed services to provide discovery, load balancing, service-to-service authentication, failure recovery, metrics, and monitoring capabilities. More complex operational functions provided by Red Hat OpenShift Service Mesh include A/B testing, canary releases, rate limiting, access control, and end-to-end authentication.
- **API Management** – OpenShift and Service Mesh can be integrated with the 3scale API Management platform to authenticate, secure, and rate-limit API access to applications and services.
- **Updates** – RHEL CoreOS and OCP updates are delivered in the same stream and automatically applied in a rolling fashion across the cluster's nodes. Administrators are notified when updates are available and can choose when to apply them.

SCOPE AND APPROACH FOR REVIEW

The understanding of OpenShift and RHEL and their combined capabilities was gained through product specification, installation, configuration, administration, and integration documentation provided by Red Hat and made available from Red Hat's public-facing web site. Coalfire further conducted interviews and engaged in live product demonstrations with Red Hat personnel. For live product demonstration purposes, OCP was also implemented on RHEL in a lab environment to provide hands-on testing and analysis of the system's capabilities to support compliance.

Coalfire's review of OpenShift on Red Hat Enterprise Linux began with a general alignment of the technology's applicability against the high-level ISO/IEC 27001:2013 ISMS requirements with guidance for the requirements provided by ISO/IEC 27002:2013. This was further narrowed down to specific requirements that may be considered applicable to a secure operation of OpenShift. An analysis of capability for the reviewed technology to address the applicable requirements was then conducted. Coalfire considered the inherent capability of OpenShift to enable security controls for the protection of supported workloads and data. Where inherent capabilities did not exist by design, consideration was made to integrate recommended adjacent people, processes, and other technologies to support the control requirements.

SCOPE OF TECHNOLOGY AND SECURITY STANDARD TO REVIEW

Coalfire was tasked by Red Hat to review OCP deployed on RHEL CoreOS. The primary focus of the review included the components, features, and functionality of OpenShift along with the supporting underlying OS features and functionality when the components are deployed on RHEL. Coalfire did not include in this assessment the pods or containers (workloads) that an organization (Red Hat's customers) may deploy in OCP. Containers deployed in the lab environment were used to demonstrate the platform's orchestration, deployment, and management capabilities. Furthermore, Coalfire did not assess available public or private image registries or repositories that may be used for acquiring application code, services, dependencies, or other elements to be hosted on or used within OCP.

For this review, Coalfire included requirements from ISO/IEC 27001:2013 Information technology – Security techniques – Information security management systems – Requirements Second Edition, October 1, 2013 publication available from www.iso.org. For broader understanding of the ISO/IEC 27001:2013 requirements, Coalfire referenced the ISO/IEC 27002:2013 Information technology – Security techniques - Code of practice for information security controls and CFISO as a certification body.

COALFIRE EVALUATION METHODOLOGY

Coalfire initially examined the FISMA Moderate impact baseline requirements and identified them as either procedural (organizational) or technical (implementation). Qualification of a procedural or technical requirement was based on reviewing the requirement narrative, testing procedures, and guidance.

Non-technical procedural requirements that include the definition and documentation of policies, procedures, and standards were not considered directly applicable to the technical solution. Likewise, non-technical requirements, including operational procedures that describe manual processes, were not assessed against the technology's capability. Examples of this type of non-technical requirement included maintaining facility visitor logs, verifying an individual's identity before granting physical or logical access, the performance of periodic physical asset inventories, or generation of network topology flow diagrams.

Technical requirements were then assessed to determine applicability to the solution or solution components' capabilities to enable supporting controls. Where the achievement of the requirement objectives was more likely to be met using an external and non-adjacent mechanism, the requirement was determined to be not applicable to the assessed technology. Examples of requirements that Coalfire determined to be not applicable to OCP on RHEL CoreOS included the use of encryption key management, wireless networking, technical, physical access controls, and antivirus solutions. That is not to say that these are not important factors to consider as it pertains to OCP, but that OCP does not natively or inherently provide these capabilities to the extent necessary to achieve compliance.

Where the requirement was qualified as applicable, Coalfire further assessed the capability of the solution to address or enable controls in support of meeting the requirement objectives. Each applicable requirement is described in the table in the following section. This table includes the findings of applicability along with a short narrative describing the capability.

OPENSHIFT APPLICABILITY TO ISO/IEC 27001:2013

The following table details the applicability of OpenShift providing control enablement through either default or configurable implementation. ISO/IEC 27001:2013 requirements that are not listed in the following table were determined to be not applicable to the reviewed technology's capabilities. Every requirement of ISO/IEC 27001:2013 must be addressed by the organization seeking certification. All requirements are the organization's responsibility, including how controls are enabled or configured to meet those requirements.

The enablement of technical controls is highly dependent on the knowledge and application of people and processes to ensure proper operation of controls is in alignment with supported requirements.

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
A.9.1.2 Access to networks and network services	Users shall only be provided with access to the network and network services that they have been specifically authorized to use.	In OCP, each user is assigned a role, either individually or through group assignment. The assigned role's access then determines access to global cluster resources or local project resources. Roles are divided into cluster and local roles. Users with the cluster-admin default cluster role bound cluster-wide can perform any action on any resource. Users with the administrator default cluster role bound locally to a specific project can manage roles and bindings in that project. Roles and the rule sets associated with them are very granular. For example, the role "networkpolicies.extensions" has the rule "create" associated with it. A user or group associated with this role will only be able to create those specific policies. Also, suppose the cluster is configured to use the multitenant isolation mode for the OpenShift SDN CNI plugin. In that case, project networks can be specifically isolated from other projects using the "isolate-projects" function. Project networks can also be specifically joined using the "join-project" function.
A.9.2.1 User registration and de-registration	A formal user registration and de-registration process shall be implemented to enable assignment of access rights.	The creation of users is dependent upon the selected identity provider. OpenShift utilizes RBAC to grant various access levels to users and groups, both cluster-wide and at the project level. Users and groups can be bound to one or more RBAC roles. Access to global cluster resources or local project resources is then determined by the assigned role's access. Roles are divided into cluster and local roles. Users with the cluster administrator default cluster role bound cluster-wide can perform any action on any resource. Users with the admin default cluster role bound locally can manage roles and bindings in that project. Roles can be added and removed to and from users and groups using "oc adm policy" command. Users can be removed by deleting their user record and removing the user identity from the selected identity provider. Group identities can by synchronized using an external LDAP provider.
A.9.2.2 User access provisioning	A formal user access provisioning process shall be implemented to assign or revoke access rights for all user types to all systems and services.	The desired identity integration is configured after installation. The system:admin (cluster administrator) account would be used to grant cluster administrator privileges to individual user or groups.

		A user in OCP is an entity that can make requests to the OCP API. The authorization layer then uses information about the requesting user to determine if the request should be allowed. A user can be assigned to one or more groups, each representing a certain set of users. Groups are useful when managing authorization policies to grant permissions to multiple users at once (e.g., allowing access to objects within a project) versus granting them to users individually. In general, user identification and authentication take place external to OCP where OCP supports authentication integration. OCP can be configured to authenticate using an identity provider, such as LDAP, GitHub, or Google (see OpenShift documentation for more detail). RBAC is used to grant various levels of access cluster-wide and at the project level. Users and groups can be bound to one or more roles. Users may also be added to groups, and groups may be assigned one or more roles. To revoke user privileges, the role is removed from the user or the user is removed from the group. Roles can be added and removed to and from users and groups using "oc adm policy" command.
A.9.2.3 Management of privileged access rights	The allocation and use of privileged access rights shall be restricted and controlled.	The allocation and use of privileged access rights are managed by a cluster administrator and enabled by a user through roles. The initial cluster administrator is defined at installation. Users may be added by a cluster administrator and allocated access rights by the assignment of appropriate roles or by being added to one or more groups that have been assigned roles. For more detailed information on cluster administrators, see the summary of Requirement Title 9.4.1.
A.9.2.4 Management of secret authentication of users	The allocation of secret authentication information shall be controlled through a formal management process.	OCP supports the configuration of authentication using several different external identity providers, including classic LDAP. Identities are created using the provider, and users are created in OCP and mapped to the identities. This is the recommended implementation model for security and compliance.
A.9.2.5 Review of user access rights	Asset owners shall review users' access rights at regular intervals.	Entering the OCP command "oc get users" will produce a list of current users in OCP. Each user is assigned one or more roles, either individually or through membership in a group. Roles can be very granular and are the assignment of specific access. Roles are either cluster roles (cluster-wide) or project roles (confined to a specific project). To view a list of all users bound to the projects and their roles, enter the command "oc get rolebindings." To view a list of users and what they have access to across the entire cluster, enter the command "oc get clusterrolebindings."
A.9.2.6 Removal or adjustment of access rights	The access rights of all employees and external party users to information	OCP allows the immediate removal of a user. First, the user record must be deleted using the command "oc delete user." If using an external identity provider, the

	and information processing facilities shall be removed upon termination of their employment, contract, or agreement, or adjusted upon change.	identity name that has been mapped to the username must also be removed using the command "oc delete identity." The user must also be removed from the external identity provider itself if appropriate. The user's authentication will fail at next login. The user must also be removed from any group memberships.
A.9.4.1 Information access restriction	Access to information and application system functions shall be restricted in accordance with the access control policy.	Each OCP installation initially defines a cluster administrator: kubeadmin. Additional cluster administrators can be defined. Once an additional administrator is defined, deleting the kubeadmin user is recommended. Cluster administrators can manage the access level of every other user. This may be through RBAC directly or access granted to groups via roles and group membership. Normal users use local roles to control who has access to their projects and what rules are applied to that access. Administrators use cluster roles to determine who has access to OpenShift. Roles contain rules that specify specific access, and the possible actions contained in a rule are get, list, create, update, delete, delete collection, and watch. OCP also includes the security context constraint (SCC) admission controller plugin, allowing the cluster administrator to control the actions that a pod can perform, and access based on the SCC assigned.
A.9.4.2 Secure log-on procedures	Where required by the access control policy, access to systems and applications shall be controlled by a secure log-on procedure.	OCP has a built-in OAuth server that is used to obtain tokens to authenticate users. When a person requests an OAuth token for log on, the OAuth server uses the configured identity provider to determine the person's identity making the request. It then determines what user the identity maps to, creates an access token for that user, and returns the token for use. From that point on, RBAC determines whether a user can perform a given action based on the roles assigned to that user.
A.9.4.3 Password management system	Password management systems shall be interactive and shall ensure quality passwords.	After installation, an identity provider can be defined for OpenShift. The AllowAllPasswordIdentityProvider identity provider will accept any non-empty username or password for login. The DenyAllPasswordIdentityProvider identity provider denies all username and passwords for login. The HTPasswdPasswordIdentityProvider identity provider validates the username and password against a flat file generated using the HTPasswd utility. Passwords are entered interactively using the HTPasswd utility and are stored in a hashed format. These three authentication providers are designed for special use cases, and Red Hat recommends that they not be used for production systems. For a production installation, Red Hat recommends that OpenShift be configured to utilize an external authentication provider such as LDAP (see the documentation for a full list of

		supported external authentication providers). In this case, it is up to the external authentication system to enforce password hygiene and management practices.
A.9.4.4 Use of privileged utility programs	The use of utility programs that might be capable of overriding system and application controls shall be restricted and tightly controlled.	Software in general can only be introduced to an OCP cluster through a project. Software and programs are instantiated from container images, which must come from a repository. OCP is capable of building software into container images and storing them in its image repository. OCP can run container images from any OCI-compliant image repository to which the OCP environment has access. When a container is created within a pod on a specific node, the image is pulled from the repository. Images may be signed, and the signatures are validated to assure their source and assure that the image has not been tampered with. Images that are not unsigned or whose signatures are invalid may be prohibited from being run in the environment. While pods have credentials automatically injected, the default credentials have extremely limited permissions. Various layers of validation and RBAC would need to be configured for a deployed program to make material changes to system or application controls.
A.10.1.1 Policy on the use of cryptographic controls	A policy on the use of cryptographic controls for protection of information shall be developed and implemented.	OCP offers several cryptographic protections for the information. Services are exposed outside of the cluster via network routes, and these routes may be configured as secured routes with TLS 1.2 encryption. Termination (decryption) may be performed at the network edge, at the destination, or a combination where termination occurs at the edge. Then the communication is re-encrypted to the endpoint. Container images may be digitally signed to assure that the image has not been tampered with and validate the source of the image. Data at the etcd datastore layer may be encrypted using the AES-CBC cipher. OCP automatically generates and manages the encryption key. RHCOS supports full-disk encryption for the system disk. As implemented, RHCOS disk encryption is FIPS compliant if FIPS mode is enabled. RHCOS disk encryption will use the AES256-CBC block cipher. This cipher is named in various places on the system and config files as cbc(aes), aescbc, aes cbc and similar.
A.10.1.2 Key management	A policy on the use, protection, and lifetime of cryptographic keys shall be developed and implemented through their whole lifecycle.	Certificates and keys for platform components are managed by the OCP platform and automatically rotated. Data at the etcd datastore layer may be encrypted using the AES-CBC cipher. OCP automatically generates and manages the encryption key.

		Applications deployed to the cluster can use external certificate and key management solutions or choose to use the OCP Service CA.
A.12.1.2 Change management	Changes to the organization, business processes, information processing facilities, and systems that affect information security shall be controlled.	The OCP Master tightly controls changes to the implementation and operations of the OCP environment. The OCP Master controls the creation of containers and pods, replicating pods, user authentication, and the API interface. Changes to containers running in pods are tightly controlled through defined SDLC processes and authentication/authorization to specific projects. Cluster upgrades are automated. OCP uses the Operator Lifecycle Manager and cluster Operators to automate the tasks needed to upgrade an OCP cluster. OCP supports integration with solutions such as ArgoCD for change management of day two configurations, such as NetworkPolicy configurations.
A.12.1.3 Capacity management	The use of resources shall be monitored, tuned, and projections made of future capacity requirements to ensure the required system performance.	The OCP administrator can collect and view cluster metrics from all containers and components in one interface (i.e., Prometheus). CPU, memory, and network-based metrics are viewable from the OCP web console. These metrics are also utilized by the pod auto scalers to determine when to scale up to add additional resources. External systems can tie into these metrics via OCP's various APIs. Additionally, many third-party solutions can be utilized to monitor and predict OCP utilization and capacity.
A.12.1.4 Separation of development, testing, and operational environments	Development, testing, and operational environments shall be separated to reduce the risks of unauthorized access or changes to the operational environment.	With node labeling, OCP can deploy development, testing, and operational software onto separate hosts, effectively separating these environments. Alternatively, separate clusters can be deployed for production use. Using the included Jenkins or Tekton CI/CD pipelines or an external pipeline, an organization's SDLC processes can include either automated or manual promotion of software through these environments.
A.12.2.1 Controls against malware	Detection, prevention, and recovery controls to protect against malware shall be implemented, combined with appropriate user awareness.	Red Hat certified images, free of vulnerabilities and compatible across the RHEL platforms, are supported by Red Hat or the third-party software owner. Red Hat Advisories alert administrators to newly discovered issues and direct the administrator to use updated images. OCP provides a pluggable API to support multiple vulnerability scanners. Red Hat Quay is an optional container registry that can be leveraged with built-in vulnerability scanning capability to scan stored applications and images for known vulnerabilities.
A.12.3.1 Information backup	Backup copies of information, software and system images shall be taken and tested regularly	The organization can perform backups of OCP to save state to a separate storage. Red Hat provides documentation detailing methods for back up and restoration of OCP.

	in accordance with an agreed backup policy.	
A.12.4.1 Event logging	Event logs recording user activities, exceptions, faults, and information security events shall be produced, kept, and regularly reviewed.	Events in OpenShift are specific to the namespace of the resource they are related to or to the OpenShift namespaces for cluster events. Events are automatically collected and stored. They must be explicitly searched using grep or extracted and searched using the jq tool against JSON output for events of interest. The OCP audit features provide a security-relevant chronological set of records documenting the sequence of activities that have affected the system by individual users, administrators, or other system components. The host operating system audits consist of standard auditing capabilities in RHEL and RHEL CoreOS. Audit also works at the API server level, logging all requests coming to the server. Each audit log contains several fields of information that provide an in-depth view of each log's behavior. An identity can be used to audit what actions a user has performed during a specific point in time. It is recommended to send logs to an external syslog server for aggregated collection, correlation, analysis, and retention.
A.12.4.2 Protection of log information	Logging facilities and log information shall be protected against tampering and unauthorized access.	OCP cluster administrators can deploy cluster logging using the OCP web console or CLI to install the Elasticsearch Operator and Cluster Logging Operator to aggregate node system audit logs, application container logs, and cluster logs. Elasticsearch is an object store designed to store and protect all logs in a central location. It is also possible to use the fluent-plug-in-remote-syslog plug-in on the host to send logs to an external syslog server. Administrators and application developers can view the logs of the projects for which they have view access. Application log information collected by the OCP logging solution is protected via the same RBAC that isolates projects and namespaces providing for protection against unauthorized access. Non-repudiation of audit data (logs) involves the configuration of auditing services to securely collect and store audit logs, protecting from unauthorized access.
A.12.4.3 Administrator and operator logs	System administrator and system operator activities shall be logged, and the logs protected and regularly reviewed.	Administrators and application developers can view the logs of the projects for which they have view access. OCP Master API audit logs OCP Master API requests by users, administrators, and system components. Audit logs can be viewed for the OCP API server or the Kubernetes API server for each master node. Users with cluster-admin, cluster-bound roles can access all container logs. It is recommended to send logs to an external syslog server for aggregated collection, correlation, analysis, review, and retention.

		The customer will be responsible for the regular review of pertinent logs.
A.12.4.4 Clock synchronization	The clocks of all relevant information processing systems within an organization or security domain shall be synchronized to a single reference time source.	By default, OpenShift uses NTP to synchronize all Masters and Nodes. This is performed via the chrony RPM package. The user space daemon updates the system clock, which is a software clock running in the kernel. Linux uses a software clock as its system clock for better resolution than the typical embedded hardware clock referred to as the Real Time Clock (RTC). The system clock can keep time by using various clock sources. Usually, the Time Stamp Counter (TSC) is used. The TSC is a CPU register which counts the number of cycles since it was last reset. Configuration information is available in the documentation.
A.12.5.1 Installation of software on operational systems	Procedures shall be implemented to control the installation of software on operational systems.	In OCP, developed and tested software can be built into an image and placed into the registry that was deployed at installation. OCP detects, based on pre-defined triggers, that the registry's image has changed and automatically deploys the new application image using a pre-defined deployment configuration (template for running applications). This deployment incorporates the new code and ensures that the production code in the target pod is identical to the most current image in the repository. The deployment process also supports rollback, either manual or automatic, to a previous version of the application in the case of deployment failure. More complex build and deployment scenarios can be implemented using the provided Jenkins or Tekton pipeline solutions or via integration with other third-party CI/CD tools. The RBAC within OCP determines who can deploy software in containers and in which projects/namespaces. Installation of software on the OCP control plane (Masters/Nodes) is controlled via the installer, the upgrade process, or must be initiated by a cluster administrator with sufficient privileges to modify them. Standard Linux OS controls enforce whether system-level users can install the software. The only users that exist on an RHCOS OpenShift node are root and core. The core user is a member of the wheel group, which permits it to use sudo for running privileged commands. It is strongly recommended that system-level access to OpenShift hosts (Masters and Nodes) be tightly controlled, restricted, and audited.

A.12.6.1 Management of technical vulnerabilities	Information about technical vulnerabilities of information systems being used shall be obtained in a timely fashion, the organization's exposure to such vulnerabilities evaluated and appropriate measures taken to address the associated risk.	Vulnerability management and notification in OCP is accomplished using capabilities from the Red Hat portfolio. For instance, the Container Security Operator (CSO) can be deployed to OCP. When deployed on a connected cluster, known vulnerability information for OCP components is visible in the cluster console. The CSO exposes vulnerabilities via the ImageManifestVuln object in the Kubernetes API. Additionally, Red Hat makes updates for OCP components, which run as containers, available through the Red Hat container registry. Customers can sign up to be automatically notified of newly discovered vulnerabilities. Customers are also responsible for discovering and managing vulnerabilities in their deployed software and applications run in workload containers. To assist with identifying known vulnerabilities, Red Hat Quay, an add-on product, can be implemented and configured to scan container images in the registry for vulnerabilities.
A.12.6.2 Restrictions on software installation	Rules governing the installation of software by users shall be established and implemented.	Rules governing the installation of software by users will be established and enforced by the customer. Red Hat products can provide the capability to support the implementation and enforcement of these rules by automating many of the processes. The implementation of the rules can be handled through a combination of OCP features (e.g., automated deployment triggers), and external systems (e.g., Jenkins) or other CI tools capable of driving the customer's SDLC. OCP also provides the ability to allow or disallow deployment of software from specific registries.
A.13.1.1 Network controls	Networks shall be managed and controlled to protect the information in systems and applications.	OCP uses an SDN to create the cluster network that allows pods to communicate with each other. The OCP SDN supports Network Policies for microsegmentation and can support this requirement. Additionally, there are third-party SDN solutions that can be integrated with OCP to provide similar functionality. In situations where several security zones are implemented on an OCP Cluster, distinct Ingress Controllers are typically deployed for each security zone to enforce segregation of ingress traffic.
A.13.1.3 Segregation in networks	Groups of information services, users, and information systems shall be segregated on networks.	OCP utilizes the SDN in combination with project and network namespaces to isolated pod networks. When configured for multitenant mode, pods from different projects cannot send or receive packets from different projects' pods and services. In addition to project/pod isolation, this enables the isolation of developer, test, and production environments. IP whitelisting is also available for additional control. Pods (groups of containers) inside an OCP cluster are reachable only via their IP addresses on the cluster network. An edge load balancer is utilized to proxy the traffic to internal destinations to access the pods from an outside network.

		Traffic between containers is accomplished by OCP using an OVS overlay network.
A.13.2.1 Information transfer policies and procedures	Formal transfer policies, procedures and controls shall be in place to protect the transfer of information through the use of all types of communication facilities.	Customers can utilize the OCP SDN multitenant mode to isolate pod networks. With this configuration, pods from different projects cannot send or receive packets from a different project's pods and services unless a cluster administrator explicitly joins those projects' networks. In addition to project/pod isolation, this enables the isolation of developer, test, and production environments. More fine-grained network controls can be configured. IP whitelisting is also available for additional control. All communication with registries utilizes TLS 1.2 encryption. The OCP router is the ingress point for all external traffic destined for services in an OCP environment and is configured for TLS 1.2 encryption by default. Communication between the OCP Master and Worker Nodes is encrypted via TLS 1.2 encryption. Nodes do not communicate directly with each other in the cluster.
A.14.1.2 Securing application services on public networks	Information involved in application services passing over public networks shall be protected from fraudulent activity, contract dispute, and unauthorized disclosure and modification.	The OCP router is the ingress point for all external traffic destined for services in an OCP environment, and this traffic can be TLS 1.2 encrypted, assuming the route is configured as such.
A.14.1.3 Protecting application services transactions	Information involved in application service transactions shall be protected to prevent incomplete transmission, misrouting, unauthorized message alteration, unauthorized disclosure, unauthorized message duplication, or replay.	Control plane traffic between the OCP Master and all Nodes in a cluster is configured to be SSL encrypted using TLS 1.2 encryption. All communication with internal OCP registries is configured to utilize TLS 1.2 encryption. OCP can be configured to allow or disallow communication with public registries. It will be up to the customer to determine and apply necessary security controls for public registries communication. The OCP router is the ingress point for all external traffic destined for services in an OCP environment, and this traffic is configured for TLS 1.2 encryption.
A.17.1.2 Implementing information security continuity	The organization shall establish, document, implement and maintain processes, procedures, and controls to ensure the required level of continuity for information security during an adverse situation.	In OCP, if the Pod Restart Policy is set to Always or On Failure, OCP will attempt to restart a failed pod. Restarting a pod will cause the creation of the containers that run in that pod. A container is created from an image stored in either an internal registry or the standard external registry. It is endowed with a specific set of security and other attributes at creation time. When a container is duplicated or restarted, all relevant security controls are also started or duplicated.

		By default, an OCP cluster is installed with three Master nodes for redundancy, allowing for failover of service to a secondary OCP Master should the primary OCP Master fail. The persistent OCP Master state is stored in etcd, deployed on each master node for a fully redundant, high availability etcd storage arrangement. There are numerous optional external methods of configuring persistent networked storage for OCP, including NFS, OCP Cinder, Azure disk or file, iSCSI, Azure Disk, AWS Elastic Block Store (EBS), OpenShift Container Storage, and others. Many of these options include local redundancy and geo-dispersed redundancy options.
A.17.2.1 Availability of information processing facilities	Information processing facilities shall be implemented with redundancy sufficient to meet availability requirements.	OCP allows the deployment of the same image in multiple pods and containers across multiple hosts with load balancing between them, providing redundancy of service. This redundancy is driven by the OCP Master and the set of hosts containing the OCP Master components, which can restart failed applications (containers in pods). For redundancy of the management and control plane, it is recommended to maintain clustered Masters for redundancy. Access to the API would need to be supported with a load balancer.
A.18.1.3 Protection of records	Records shall be protected from loss, destruction, falsification, unauthorized access, and unauthorized release, in accordance with legislative, regulatory, contractual, and business requirements.	RHEL CoreOS disks can be encrypted. OCP offers the capability of encrypting datastores in the datastore layer. An OCP Master can be deployed redundantly. The persistent OCP Master state is stored in the etcd, configured in a fully redundant, high availability arrangement by default. Pods may also be deployed redundantly across multiple hosts. TLS 1.2 encryption is enabled for communications. There are numerous optional methods of configuring persistent networked storage for OCP including NFS, OpenStack Cinder, Azure disk or file, iSCSI, Azure Disk, AWS Elastic Block Store (EBS), OpenShift Container Storage, and others. Many of these options include local redundancy and geo-dispersed redundancy options.
A.18.1.5 Regulation of cryptographic controls	Cryptographic controls shall be used in compliance with all relevant agreements, legislation, and regulations.	OCP recommends and enables the protection of repositories with TLS 1.2 communication encryption via custom certificates. It is also recommended to enable TLS encryption, usually terminated at the edge, for communication external to containers. OCP offers the capability of encrypting the etcd datastore using the AES-CBC encryption providers. Keys are generated in OCP explicitly, base64 encoded and then stored in the configuration file or in etcd. Keys are automatically rotated.

		Third-party SDN providers may natively implement encryption and cryptographic controls to meet the requirement. Both datastore layer encryption and cluster traffic encryption are optional.

Table 1: ISO/IEC 27001:2013 Applicability Details for OCP on RHEL CoreOS

CONCLUSION

OpenShift RHEL CoreOS, as reviewed by Coalfire, can be useful in providing support for the outlined objectives and requirements of ISO/IEC 27001:2013. Through proper implementation and integration into the organization's more significant infrastructure and ISMS, OCP may be useable to support an ISO/IEC 27001:2013 controlled environment.

Care should be given for the implementation of OCP concerning classification and categorization of data such that applications that process, transmit, and store data of differing security classification are not comingled on the same Nodes. Likewise, the OCP Master should be dedicated to managing and controlling plane functions and keeping separate from the data plane.

Coalfire's opinion is based on observations and analysis of the provided documentation, interviews with Red Hat personnel, and hands-on engagement with a lab environment. The presented conclusions are based upon several underlying presumptions and caveats. These caveats include adherence to vendor best practices and hardening of configuration as supported by the system components. This solution should be implemented in alignment with the organization's mission, values, business objectives, general approach to security, and concerning the overall ISMS.

Inclusion into the organization's overall compliance program includes considerations for supporting network infrastructure, isolation, or network segmentation of workloads representing different levels of risk, physical security, personnel security, vulnerability testing, and an ongoing risk and compliance evaluation and improvement program.

A COMMENT REGARDING REGULATORY COMPLIANCE

Coalfire disclaims generic suitability of any product to cause a federal agency to use that product to achieve regulatory compliance. Agencies attain compliance through a Governance, Risk Management, and Compliance (GRC) program, not via the use of a specific product. This is true for federal agencies subject to FISMA and customers targeting compliance with other regulations.

LEGAL DISCLAIMER

Coalfire expressly disclaims all liability with respect to actions taken or not taken based on the contents of this whitepaper and the supporting controls workbook and the opinions contained therein. The opinions and findings within this evaluation are solely those of Coalfire and do not represent any assessment findings, or opinions, from any other parties. This document's contents are subject to change at any time based on revisions to the applicable regulations and standards (e.g., Health Information Portability and Accountability Act [HIPAA], PCI-DSS, et al.) Consequently, any forward-looking statements are not predictions and are subject to change without notice. While Coalfire has endeavored to ensure that the information contained in this document has been obtained from reliable sources, there may be regulatory, compliance, or other reasons that prevent Coalfire from doing so. Consequently, Coalfire is not responsible for any errors or omissions, or for the results obtained from the use of this information. Coalfire reserves the right to revise any or all of this document to reflect an accurate representation of the content relative to the current technology landscape. To maintain this document's contextual accuracy, all references to this document

must explicitly reference the entirety of the document inclusive of the title and publication date. Neither party will publish a press release referring to the other party or excerpting highlights from the document without the other party's prior written approval. For questions regarding any legal or compliance matters referenced herein, you should consult legal counsel, your security advisor, and/or the relevant standard authority.

ADDITIONAL INFORMATION, RESOURCES, AND REFERENCES

This section contains a description of the links, standards, guidelines, and reports used for the materials used to identify and discuss the features, enhancements, and security capabilities of OpenShift 4.4.

RED HAT

The Red Hat OpenShift Container Platform documentation used to provide depth and context for this document is available at the Welcome page. The left column of the page provides further details into every capability for OpenShift. These details are available at the following link: https://docs.openshift.com/container-platform/4.4/welcome/index.html.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION

ISO/IEC 27002:2013 gives guidelines for organizational information security standards and information security management practices, including the selection, implementation, and management of controls, considering the organization's information security risk environments. This publication is located at the following link: https://www.iso.org/standard/54533.html.

ISO/IEC 27001:2013 specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system within the organization's context. It also includes requirements for the assessment and treatment of information security risks tailored to the organization's needs. This publication is located at the following link: https://www.iso.org/standard/54534.html.

ISO/IEC 27017:2015 gives guidelines for information security controls applicable to cloud services guidance provision and use. This is completed for cloud service providers and cloud service customers by providing additional implementation guidance for relevant controls specified in ISO/IEC 27002. Additional controls with implementation guidance specifically relate to cloud services. This publication is located at the following link: https://www.iso.org/standard/43757.html.

Additional ISO links are listed below:

- https://www.iso.org/isoiec-27001-information-security.html
- https://www.iso.org/standard/73906.html

COALFIRE ISO INFORMATION

The following links provide information about Coalfire's ISO support for writing this whitepaper.

- https://www.coalfire.com/Solutions/Audit-and-Assessment/ISO-27001
- http://www.coalfireiso.com/

ABOUT THE AUTHOR

Byron Estrada | Sr. Consultant, Solutions Engineering, Coalfire Systems

As Sr. Consultant, Byron is an author and thought leader on information security topics for Coalfire's clientele with a focus in enterprise security.

ABOUT THE CONTRIBUTORS

Chris Krueger | Principal II, Solutions Engineering, Coalfire Systems

As a Principal, Chris contributes as an author and thought leader on information security and regulatory compliance topics for Coalfire's clientele in new and emerging technical areas.

Al Mahdi Mifdal | Principal, Global Assurance, Coalfire ISO

As Principal, Al Mahdi contributes as an author and thought leader on information security and regulatory compliance topics for Coalfire ISO's clientele in compliance and assurance services.

Jason Macallister | Senior Consultant, Cyber Engineering, Coalfire Systems

As Senior Consultant, Jason consults on information security and regulatory compliance topics as they relate to advanced infrastructure, emerging technology, and cloud solutions. Jason was the author of the previous Red Hat whitepaper titled *Red Hat OpenShift Container Platform Applicability Guide for ISO/IEC 27001:2013 v1.0.*

Mitch Ross | Director, Cyber Engineering, Coalfire Systems

As Director, Mitch contributes as an author and thought leader on information security and regulatory compliance topics for Coalfire's clientele with an emphasis in security in the cloud.

ABOUT COALFIRE

Coalfire is the trusted cybersecurity advisor that helps private and public sector organizations avert threats, close gaps, and effectively manage risk. By providing independent and tailored advice, assessments, technical testing, and cyber engineering services, we help clients develop scalable programs that improve their security posture, achieve their business objectives, and fuel their continued success. Coalfire has been a cybersecurity thought leader for nearly 20 years and has offices throughout the United States and Europe. For more information, visit Coalfire.com.

October 27, 2020