WHITE PAPER

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

JASON MACALLISTER
AL MAHDI MIFDAL | ISO 27001 MASTER-QSA-CISA-CISM
MITCH ROSS | CISSP

VERSION 1.0

redhat.

COALFIRE.

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

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Red Hat, Inc. (Red Hat) delivers a comprehensive portfolio of products and services built from open source software components using an affordable, predictable subscription and support model. Red Hat engaged Coalfire, a respected cybersecurity engineering, advisory, and assessment company, to conduct an independent technical assessment of Red Hat OpenShift Container Platform (OpenShift) on Red Hat Enterprise Linux. For a broader understanding of the requirements and their applicability to technical solution implementation, Coalfire also reviewed supporting documentation from Coalfire ISO (CFISO), including implementation and assessment guidance. CFISO is an ISO/IEC 27001 certification body accredited by both the ANSI-ASQ National Accreditation Board (ANAB) and the United Kingdom Accreditation Service (UKAS). CFISO provides ISO/IEC 27001:2013 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 OpenShift on Red Hat Enterprise Linux to the International Organization for Standardization (ISO)/International Electrotechnical Commission (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 needs of the organization. Coalfire assessed control capabilities applicable to OpenShift with respect to 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 that are provided in this product applicability guide are in no way a claim of conformity to ISO/IEC 27001:2013. It is up to each organization desiring to conform to the standard to address all requirements of the standard.

Containerization provides valuable benefits to businesses that incorporate it into their service development and delivery model. Some of the benefits include increased developer productivity; decrease in time to application deployment; increased application portability, agility, and scalability to align with changes in service demand; and increased compute efficiencies. OpenShift is a container platform that natively integrates open source Linux container technologies and Kubernetes, combining them in an enterprise solution running on Red Hat Enterprise Linux. OpenShift provides an API, web interface, and CLI to manage the underlying container technologies and Kubernetes to allow users to orchestrate the creation and management of containers. OpenShift provides self-service build and deployment automation for containers in addition to operational container features including scaling, monitoring, and management capabilities.

This product applicability guide may be useful for organizations desiring to utilize container technologies within the framework of an ISO program of compliance. The guide discusses relevant ISO/IEC 27001:2013 requirements that are applicable to OpenShift on Red Hat Enterprise Linux. The focus of this paper is on technical controls that are pertinent to and in alignment with OpenShift capabilities.

COALFIRE OPINION

Security controls, features, and functionality that are built into OpenShift on Red Hat Enterprise Linux can support and/or address relevant technical ISO/IEC 27001:2013 requirements. OpenShift provides granular control and improved security at scale for containerized workloads.

INTRODUCING 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, helping organizations secure financial information, intellectual property, employee information, or information entrusted to third parties. Frequently, ISO/IEC 27001:2013 conformance can be leveraged for other compliance efforts, including, but not limited to, 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 (ISO 27001) is divided into two sections, namely clauses 4 through 10, which focus on the design of the ISMS within the context of the continuous improvement cycle, and Annex A, comprising of 114 control objectives across 14 domains (e.g., human resources security, cryptography, access control). Figure 1 provides a high-level illustration of the two sections of the standard.

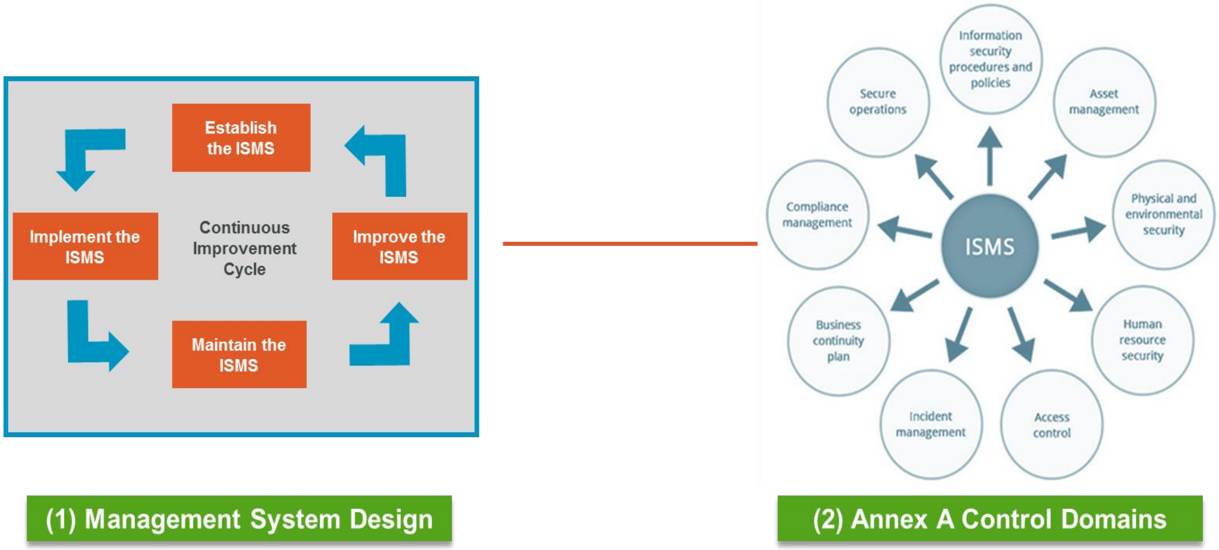

Figure 1 – High-level ISO/IEC 27001:2013 Standard

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 context of the organization. Annex A (normative) provides a reference of control objectives and controls. This paper focuses on the capabilities of the assessed technology 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.

INTRODUCING OPENSHIFT CONTAINER PLATFORM

OpenShift is a comprehensive enterprise-grade application platform built for containers with Kubernetes. It is an integrated platform to run, orchestrate, monitor, and scale containers. OpenShift 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 without reducing developer productivity and provides a consistent operations and management experience across any infrastructure in support of many teams.

OPENSHIFT CONTAINER PLATFORM ARCHITECTURE

The following is a list of components and roles that support OpenShift.

Operating System (OS) - OpenShift can be deployed on either Red Hat Enterprise Linux or, soon, Red Hat CoreOS.

Red Hat CoreOS is a container- and Kubernetes-optimized, minimal footprint OS powered by much of the same source as Red Hat Enterprise Linux. This pre-hardened OS will assist organizations with meeting requirements for least functionality due to its lightweight, purpose-built nature, as it only includes necessary features, functions, and services to host containers in an OpenShift environment.

Red Hat Enterprise Linux has built in security features and functionality that, as configured in an OpenShift installation, provide a secure platform for supporting the OpenShift components and the workloads in containers that OpenShift orchestrates.

Operating Environment - OpenShift can be deployed on bare-metal physical hardware, on virtual infrastructure, or in the cloud. It can be deployed on private or certified public cloud environments, depending on the organization's specific use cases.

OCI Runtime - OpenShift uses an Open Container Initiative (OCI)-compatible runtime for the execution of Linux containers. OCI is an open governance structure for the express purpose of creating open industry standards around container formats and runtime.

Kubernetes – Kubernetes provides orchestration for complex multi-container services. Kubernetes also provides scheduling for services across a container host cluster. To Kubernetes, OpenShift adds developer- and operations-centric tools that enable rapid application development, easy deployment and scaling, and long-term life-cycle maintenance for applications. OpenShift also leverages integrated components from Kubernetes to automate application builds, deployments, scaling, health management, and more. Included in the automation capabilities of OpenShift is the ability to configure and deploy Kubernetes container host clusters.

OpenShift Container Platform Components

The following components are specific to OpenShift itself.

OpenShift Nodes – Nodes are instances of Red Hat Enterprise Linux with the OpenShift software installed. Nodes are where end-user applications are ultimately run in containers. Nodes will contain the necessary OpenShift node daemon, the container runtime, and other necessary services to support the hosting of containers. Most of the software components that run above the OS (e.g., the software-defined network daemon) all run in containers themselves on the Node.

Containers – End-user application instances, application components, or other services are run in Linux containers. This OCI-compatible container technology provides an open source software development and delivery platform that allows applications to be packaged for portability. The container only includes the necessary libraries, functions, elements, and code required to run the application.

Pod –While application components run in containers, OpenShift orchestrates and manages pods. A pod is an orchestrated unit in OpenShift made up of one or more containers. OpenShift will schedule and run all containers in a pod together on the same host. Generally, a pod should only contain a single function such as app server or web server and should not include multiple functions such as database and app server.

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. Just like the Nodes, the OpenShift Master is run on Red Hat Enterprise Linux. While the Master is technically also a Node and can participate in the software-defined network, for separation of function, the OpenShift Master should NOT be scheduled to run application instances (pods). The following are the four functions of the OpenShift Master:

API and Authentication – The Master provides the single API that all tooling and systems interact with. Everything that interacts with OpenShift must go through this API. All API requests are 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 LDAP, OAuth, or other providers. The Master evaluates requests for both authentication (AuthN) and authorization (AuthZ). Users of OpenShift 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 OpenShift data store. The data store uses etcd, a distributed key-value store. The data store houses information about the OpenShift environment and pertaining to the OpenShift Master, including user account information and the RBAC rules; the OpenShift environment state, including application environment information and non-application user data; and important environment variables, secrets data, and other information.

Scheduler – The scheduler determines pod placement within OpenShift. 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 OpenShift. This allows placement of pods within OpenShift 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 OpenShift Master is also responsible for monitoring the health of pods and scaling the pods as desired to handle additional load. The OpenShift Master executes liveness and readiness tests using probes that are defined by users. The OpenShift Master can detect failed pods and remediate failures as they occur.

Service Layer – The OpenShift Service Layer allows for application components to 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. OpenShift automatically and transparently handles load balancing across the services' instances. In conjunction with probes, the OpenShift Service Layer ensures that traffic is only directed toward healthy pods, which helps to maintain component availability.

Persistent Storage – Linux containers are natively ephemeral and only maintain data for as long as they are running. Applications and/or application components may require access to a long-term persistent storage repository, such as may be required for a database engine. OpenShift provides the means to connect pods to external real-world storage, which allows for stateful applications to be used on the platform. Persistent storage types that are usable include iSCSI, Fiber Channel, and NFS, as well as 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 an integration with OpenShift.

OpenShift Router – The routing layer provides external access to applications hosted in OpenShift. The routing layer operates in partnership with the Service Layer and provides automated load balancing to pods for external clients. The OpenShift Router runs in pods on the platform but receives traffic from the outside world and proxies the traffic to the appropriate pods. The OpenShift Router 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 OpenShift software-defined network (SDN) is a unified cluster network that enables communication between pods across the OpenShift cluster. The OpenShift SDN configures an overlay network that uses Open vSwitch (OVS). Red Hat currently provides three SDN plug-ins for use with OpenShift. The ovs-subnet plug-in provides a “flat” pod network where every pod can communicate with every other pod and service cluster-wide. The ovs-multitenant plug-in provides project-level isolation for pods and services. Each project receives a unique Virtual Network ID (VNID) that identifies traffic from pods assigned to the project. Pods from different projects cannot send packets to or receive packets from pods and services of a different project by default. Administrators of OpenShift can join or isolate projects as required. Lastly, the ovs-networkpolicy plug-in provides extremely fine-grained access control via user-defined rules. Network policy rules can be built in a “mandatory access control” style, where all traffic is denied by default unless a rule explicitly exists, even for pods/containers on the same host.

OpenShift Registry – The OpenShift Registry provides integrated storage and management for sharing container images, but OpenShift can utilize existing OCI-compliant container registries that are accessible to the Nodes and the OpenShift Master via the network.

Figure 2 is a high-level illustration of the OpenShift components.

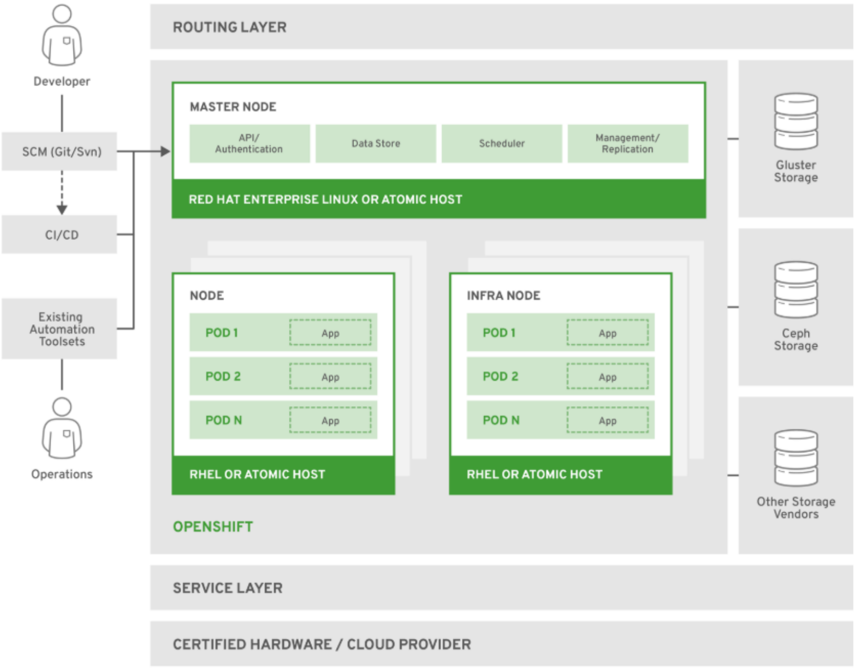

Figure 2: High-Level OpenShift Architecture

Users – User (operators, developers, application administrators) access to OpenShift 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 OpenShift hosts, even for complicated application debugging and troubleshooting tasks.

There are three types of users that can exist in an OpenShift environment: regular users, system users, and service accounts.

Regular users are created automatically in the system upon first logon or via the API. Most interactive OpenShift users, including operators, developers, and application administrators, will be represented by this type of user account.

System users This is the system:admin account which is a regular user type of account with elevated privileges. The system:admin account is the cluster administrator account that gets created when the system is setup for the first time. This account has special privileges and can only be logged on via a certificate from the console of the OpenShift Master.

Service accounts are non-human system users, often associated with projects, 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 OpenShift annotations and metadata. It is the central vehicle by which access to resources for regular users is managed and is essentially the tenancy model of OpenShift 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.

OPENSHIFT CONTAINER PLATFORM SECURITY

OpenShift enables continuous security with a defense-in-depth and secure software supply chain to the application platform. 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 in the platform. OpenShift runs on Red Hat Enterprise Linux and makes heavy use of the existing security features built into the OS. Red Hat manages the OS packages and provides trusted distribution of content. Red Hat is committed to responsive action to security vulnerabilities. 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, SSL, etc. Additionally, OpenShift provides integration with third-party identity management solutions to support secure authentication and authorization options in alignment with organization compliance requirements.

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

Container Host and Platform Multitenancy – Red Hat Enterprise Linux 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.

Container Registries – OpenShift includes an integrated container registry that provides basic functionality supporting build and deployment automation within the cluster, tied into the OpenShift 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 vulnerability scanning of applications and software in images and more.

Building Containers – OpenShift integrates tightly with Jenkins and can be easily integrated with other Continuous Integration/Continuous Delivery (CI/CD) tools to manage builds, code inspection, code scanning, and validation.

Deploying Containers – By default, OpenShift prevents containers from running as root or other specifically-named users. In addition, OpenShift enables granular deployment policies that allow operations, security, and compliance teams to enforce quotas, isolation, and access protections.

Container Orchestration – OpenShift integrates secure operational capabilities to support trust between users, applications, and security policies.

Network Isolation – OpenShift uses a SDN approach to provide a unified cluster network that enables communication between pods across the OpenShift cluster. The pod network is established and maintained by the OpenShift SDN plug-ins, which create an overlay network using OVS. There are three SDN plug-ins available from Red Hat as options for the customer to deploy: the ovs-subnet, ovs-multi-tenant, and ovs-networkpolicy. Other third-party SDN solutions exist that are capable of being integrated into OpenShift.

Secure the data – OpenShift provides access to and integration with a broad range of storage platforms and protocols, allowing applications to securely store and encrypt application data.

API management – OpenShift can be integrated with the 3scale API Management platform to authenticate, secure, and rate-limit API access to applications and services.

SCOPE AND APPROACH FOR REVIEW

The understanding of OpenShift and Red Hat Enterprise Linux and their combined capabilities was gained through product specification, installation, configuration, administration, and integration documentation provided by Red Hat and generally 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, OpenShift was also implemented on Red Hat Enterprise Linux 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 applicability of the technology 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 inherent capability of OpenShift to enable security controls for the protection of supported workloads and data. Where inherent capabilties did not exist by design, consideration was made for the integration of 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 OpenShift as deployed on Red Hat Enterprise Linux. 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 Red Hat Enterprise Linux. Coalfire did not include in this assessment the pods or containers (workloads) that an organization (Red Hat's customers) may deploy in OpenShift. Containers that were deployed in the lab environment were strictly used for the purposes of demonstrating 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 OpenShift.

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 ISO/IEC 27001:2013 requirements and identified them as either procedural (organizational) or technical (implementation). Qualification of a requirement as procedural or technical was based on a review of the requirement narrative, testing procedures, and guidance.

“Non-technical” procedural requirements that include definition and documentation of policies, procedures, and standards were not considered directly applicable to the technical capabilities of the solution though should have bearing on the management and use of the 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 maintenance of facility visitor logs, verification of an individual’s identity prior to granting physical or logical access, performance of periodic physical asset inventories, or generation of network topology or flow diagrams.

Technical requirements were then assessed to determine applicability to the capabilities of the solution and/or solution components to enable supporting controls. Where 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 OpenShift on Red Hat Enterprise Linux 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 OpenShift, but rather that OpenShift 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 capabilities of the reviewed technology to address. Every requirement of ISO/IEC 27001:2013 must be addressed by the organization seeking certification. All requirements are the responsibility of that organization, 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 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 OpenShift, each user is assigned a role, either individually or through group assignment. Access to global cluster resources or to local project resources is then determined by the access defined by the assigned role. 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 admin default cluster role bound locally can manage roles and bindings in that project. Roles and the rule sets associated with them are very granular in nature. 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. In addition, if the cluster is configured to use the ovs-multitenant SDN plug-in, project networks can be specifically isolated from other projects using the "isolate-projects" function. Or, project networks can be specifically joined together 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 levels of access 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 to local project resources is then determined by the access defined by the assigned role. 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 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 can be configured at install time. The Deny All identity provider is used by default for new OpenShift deployments but can be configured at initial installation or post-installation. The system:admin (cluster admin) account would be used to grant cluster-admin privileges to individual user or groups. A user in OpenShift is an entity that can make request to the OpenShift API. The authorization layer then uses information about the requesting user to determine if the

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
		request should be allowed. A user can be assigned to one or more groups, each of which represent 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, identification and authentication takes place external to OpenShift where OpenShift supports authentication integration. As an administrator, OAuth can be configured to authenticate using an identity provider, such as LDAP, GitHub, or Google, among others (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 and/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 is managed by a cluster administrator and enabled on a user basis through the assignment of 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.
A.9.2.4 Management of secret authentication of users	The allocation of secret authentication information shall be controlled through a formal management process.	OpenShift supports the configuration of authentication using several different external identity providers including classic LDAP. Identities are created using the provider, and then users are created in OpenShift 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 OpenShift command "oc get users" will produce a list of current users in OpenShift. 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 that are 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". Performing this check at regular intervals is a highly recommended practice

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
A.9.2.6 Removal or adjustment of access rights	The access rights of all employees and external party users to information and information processing facilities shall be removed upon termination of their employment, contract, or agreement, or adjusted upon change.	OpenShift 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 identity name that has been mapped to the user name 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 OpenShift installation initially defines a cluster administrator: system: admin. Additional cluster administrators can be defined. 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, deletecollection, and watch. OpenShift also provides security context constraints (SCC) that control the actions that a pod can perform and what it can access.
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.	OpenShift has a built in OAuth server that users use to obtain tokens to authenticate themselves. When a person requests an OAuth token for log-on, the OAuth server uses the configured identity provider to determine the identity of the person making the request. It then determines what user that 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.	During installation, an identity provider can be defined for OpenShift. The AllowAllPasswordIdentityProvider identity provider will accept any non-empty username/password for login. The DenyAllPasswordIdentityProvider identity provider denies all username/passwords for login. The HTPasswdPasswordIdentityProvider identity provider validates the username/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

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
		special use cases, and it is not recommended that they be used for production systems. For a production installation, it is highly recommended that OpenShift be configured to utilize an external authentication provider such as LDAP (see 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.	Utility programs, or software in general, can only be introduced to an OpenShift cluster through a project. Software/programs are instantiated from container images, which must come from a repository. OpenShift is capable of building software into container images and storing them in its own image repository. OpenShift can run container images from any OCI-compliant image repository to which the OpenShift environment has access. When a container is created within a pod on a specific Node, the image(s) is pulled from the repository. Images may be signed and the signatures validated to assure their source and to 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. For a deployed program to make material changes to system or application controls, various layers of validation and RBAC would need to be configured.
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.	OpenShift Container Platform offers several cryptographic protections for 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 and 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 as well as validate the source of the image. Data at the datastore layer may be encrypted using one of three encryption providers and a generated key.
A.10.1.2 Key management	A policy on the use, protection and lifetime of cryptographic keys shall be developed and	Keys can be created and managed for the encryption of data at the datastore level. Three different encryption algorithms are available - AESCBC, SECRETBOX, and AESGCM. The AESGCM option requires rotation every

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
	implemented through their whole lifecycle.	200,000 writes. All keys may be manually rotated on a schedule determined by an administrator.
A.12.1.2 Change management	Changes to the organization, business processes, information processing facilities and systems that affect information security shall be controlled.	Changes to the implementation and operations of the OpenShift environment are tightly controlled by the OpenShift Master. The OpenShift Master controls the creation of containers and pods, the replication of pods, user authentication, and the API interface. Changes to containers that are running in pods are tightly controlled through defined SDLC processes and authentication/authorization to specific projects. Cluster upgrades are automated. The automated method uses Ansible playbooks and Operators to automate the tasks needed to upgrade an OpenShift cluster. Either in-place upgrades or blue-green upgrades are supported by Red Hat.
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 OpenShift administrator can collect and view cluster metrics from all containers and components in one interface. When this is configured, CPU, memory, and network-based metrics are viewable from the OpenShift 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 OpenShift's various APIs. Additionally, many third-party solutions can be utilized to monitor and predict OpenShift 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, OpenShift provides the ability to deploy development, testing, and operational software onto separate hosts, effectively separating these environments. Using the included Jenkins, or driven externally, 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 Red Hat Enterprise Linux platforms, are supported by Red Hat and/or the third-party software owner. Red Hat Advisories alert administrators to newly discovered issues and direct the administrator to use updated images. OpenShift 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.

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
A.12.3.1 Information backup	Backup copies of information, software and system images shall be taken and tested regularly in accordance with an agreed backup policy.	The organization can perform backups of OpenShift to save state to a separate storage. Red Hat provides documentation detailing methods for back up and restoration of OpenShift.
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 DEFAULT namespace for cluster events. Events are automatically collected and stored. They must be explicitly searched using grep or can be extracted and searched using the jq tool against JSON output for events of interest. There are two types of logs in OpenShift - cluster logs and service logs. Service logs are associated with each systemd service that is running on a host and can be retrieved per host with the journalctl command. In clusters with the aggregated logging stack deployed, they may be found in the logging .operations indexes. OpenShift Master API logging logs OpenShift Master API requests by users, administrators, and system components. This feature must be enabled and can be located in the service logs by searching for AUDIT. User activity logs will always be associated with the OpenShift Master API logs. 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.	In OpenShift, the EFK stack (Elasticsearch, Fluentd, and Kibana) can be deployed to aggregate logs for a range of OpenShift services. Elasticsearch is an object store designed to store and protect all logs in a central location. In addition, it is possible to use the fluent-plugin-remote-syslog plug-in on the host to send logs to an external syslog server. Non-administrators have no control over log retention policies within the OpenShift logging solution, and application log information collected by the OpenShift logging solution is protected via the same RBAC that isolates projects/namespaces providing for protection against 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.	OpenShift Master API logging logs OpenShift Master API requests by users, administrators, and system components. This feature must be enabled and can be located in the service logs by searching for AUDIT. User activity logs will always be associated with the OpenShift Master API logs. It is recommended to send logs to an external syslog server for aggregated collection, correlation, analysis, review, and

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
		retention. The customer will be responsible for 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 either the NTP or the chrony RPM package, and one of these packages must be installed and enabled.
A.12.5.1 Installation of software on operational systems	Procedures shall be implemented to control the installation of software on operational systems.	In OpenShift, developed and tested software can be built into an image and placed into the registry that was deployed at installation. OpenShift detects, based on pre-defined triggers, that the image in the registry 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 solution or via integration with other third-party CI/CD tools. The RBAC within OpenShift determines who can deploy software in containers in which projects/namespaces. Installation of software on the OpenShift control plane (Masters/Nodes) is controlled via the installer and/or 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 software. It is strongly recommended that system-level access to OpenShift hosts (Masters/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 OpenShift is accomplished using capabilities from the Red Hat portfolio. For instance, Red Hat Satellite can be used for managing the core OS packages in an OpenShift environment as part of a systems management methodology to address vulnerability management. Additionally, Red Hat makes updates for OpenShift components, which run as containers, available through the Red Hat container registry.

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
		Customers will be responsible for discovery and managing vulnerabilities in their deployed software and applications that are run in workload containers. To assist with identification of 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 capability to support implementation and enforcement of these rules by automating many of the processes. The implementation of the rules can be handled through a combination of OpenShift features, e.g., automated deployment triggers, and “external” systems, e.g., Jenkins or other CI tools that are capable of driving the customer’s SDLC.
A.13.1.1 Network controls	Networks shall be managed and controlled to protect information in systems and applications.	OpenShift uses an SDN approach to creating the cluster network that allows pods to communicate with each other. Red Hat offers two SDN solutions capable of supporting this requirement that can be implemented as plug-ins that are deployed and run within containers and interface with the underlying OVS. Additionally, there are third-party SDN solutions that can be integrated with OpenShift to provide similar functionality
A.13.1.3 Segregation in networks	Groups of information services, users and information systems shall be segregated on networks.	OpenShift utilizes SDN namespaces to isolated pod networks. Pods from different projects cannot send or receive packets from pods and services of a different projects. 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 OpenShift cluster are reachable only via their IP addresses on the cluster network. To access the pods from an outside network, an edge load balancer is utilized to proxy the traffic to internal destinations. Traffic between containers is accomplished by OpenShift 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 OpenShift SDN multitenant plug-in to isolate pod networks. Pods from different projects cannot send or receive packets from pods and services of a different project unless those projects’ networks are explicitly joined by a cluster administrator. In addition to project/pod isolation, this enables the isolation of developer, test, and production environments. More fine-grained network controls can be implemented when using the OpenShift NetworkPolicy SDN plug-in. IP whitelisting is also available for additional control. Traffic over the SDN can be encrypted with TLS 1.2 encryption tunneled over IPsec.

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
		All communication with registries utilizes TLS 1.2 encryption. The OpenShift router is the ingress point for all external traffic destined for services in an OpenShift environment and can be configured for TLS 1.2 encryption. Communication between the OpenShift Master and 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 OpenShift router is the ingress point for all external traffic destined for services in an OpenShift 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, mis-routing, unauthorized message alteration, unauthorized disclosure, unauthorized message duplication or replay.	Control plane traffic between the OpenShift Master and all Nodes in a cluster can be configured to be SSL encrypted using TLS 1.2 encryption. All communication with internal OpenShift registries can be configured to utilize TLS 1.2 encryption. Communication with public registries is not controlled by OpenShift. It will be up the customer to determine and apply necessary security controls for communication to public registries. The OpenShift router is the ingress point for all external traffic destined for services in an OpenShift environment, and this traffic can be 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 OpenShift, if the Pod Restart Policy is set to Always or On Failure, OpenShift 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 that is stored in either an internal registry or the standard external registry and 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 within the container are also started or duplicated. An OpenShift Master can be deployed in a redundant fashion, allowing for failover of service to a secondary OpenShift Master should the primary OpenShift Master fail. The persistent OpenShift Master state is stored in etcd, which can be configured in a fully redundant, high availability storage arrangement. There are numerous optional external methods of

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
		configuring persistent networked storage for OpenShift including NFS, OpenStack Cinder, Azure disk or file, iSCSI, Azure Disk, AWS Elastic Block Store (EBS), 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.	OpenShift allows the deployment of the same image in multiple pods/containers across multiple hosts with load balancing between them, providing redundancy of service. This redundancy is driven by the OpenShift Master and the set of hosts that contains the OpenShift Master components, which have the capability of restarting 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.	OpenShift offers the capability of encrypting datastores in the datastore layer. An OpenShift Master can be deployed in a redundant fashion. The persistent OpenShift Master state is stored in etcd, which can be configured in a fully redundant, high availability arrangement. Pods may also be deployed in a redundant fashion across multiple hosts. TLS1.2 encryption is enabled for communications. There are numerous optional methods of configuring persistent networked storage for OpenShift including NFS, OpenStack Cinder, Azure disk or file, iSCSI, Azure Disk, AWS Elastic Block Store (EBS), 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.	OpenShift 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. OpenShift offers the capability of encrypting datastores in the datastore layer using one of three available encryption providers - aescbc, secretbox, and aesgcm. Keys are generated in OpenShift explicitly, base64 encoded, and then stored in the configuration file. Keys are rotated manually. OpenShift provides the capability to tunnel the provided SDN solution traffic over IPSec. Likewise, third-party SDN providers may natively implement encryption and cryptographic controls to meet the requirement.

REQUIREMENT TITLE	REQUIREMENT DESCRIPTION	CONTROL CAPABILITY SUMMARY
		Both datastore layer encryption and cluster traffic encryption are optional.

Table 1 - OpenShift Container Platform Applicability to ISO/IEC 27001:2013 Detail

CONCLUSION

OpenShift hosted on Red Hat Enterprise Linux, as reviewed by Coalfire, can be effective in providing support for the outlined objectives and requirements of ISO/IEC 27001:2013. Through proper implementation and integration into the organization's greater technical infrastructure and ISMS, OpenShift may be useable in support of an ISO/IEC 27001:2013 controlled environment. Care should be given for the implementation of OpenShift with respect to classification/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 OpenShift Master should be dedicated to management and control plane functions and kept 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 provided 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 with respect to the overall ISMS. Inclusion into the organization's overall program of compliance 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.

ADDITIONAL INFORMATION, RESOURCES, AND REFERENCES

https://docs.openshift.com/container-platform/3.11/welcome/index.html

https://docs.openshift.com/container-platform/3.11/admin_guide/ipsec.html

https://docs.openshift.com/container-platform/3.11/architecture/networking/routes.html

https://developers.redhat.com/blog/2017/01/24/end-to-end-encryption-with-openshift-part-1-two-way-ssl/

https://docs.openshift.com/container-platform/3.11/install/prerequisites.html

https://docs.openshift.com/container-platform/3.11/admin_guide/managing_networking.html

https://docs.openshift.com/container-platform/3.11/admin_guide/encrypting_data.html

https://docs.openshift.com/container-platform/3.11/security/monitoring.html

https://docs.openshift.com/container-platform/3.11/install_config/master_node_configuration.html#master-node-config-audit-config

https://docs.openshift.com/container-platform/3.11/install_config/master_node_configuration.html#master-node-config-advanced-audit

https://docs.openshift.com/container-platform/3.11/dev_guide/authorization.html

https://docs.openshift.com/container-platform/3.11/admin_guide/manage_rbac.html

https://docs.openshift.com/container-platform/3.11/admin_guide/manage_users.html

https://docs.openshift.com/container-platform/3.11/architecture/core_concepts/projects_and_users.html

https://docs.openshift.com/container-platform/3.11/admin_guide/manage_scc.html

https://docs.openshift.com/container-platform/3.11/architecture/index.html#architecture-index

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/selinux_users_and_administrators_guide/

https://docs.openshift.com/container-platform/3.11/admin_guide/seccomp.html

https://github.com/kubernetes/kubernetes/blob/release-1.4/docs/design/seccomp.md

https://docs.openshift.com/container-platform/3.11/admin_guide/sysctls.html

https://docs.openshift.com/container-platform/3.11/admin_guide/ipsec.html

https://developers.redhat.com/blog/2017/01/24/end-to-end-encryption-with-openshift-part-1-two-way-ssl/

https://docs.openshift.com/container-platform/3.11/architecture/networking/networking.html

https://docs.openshift.com/container-platform/3.11/security/network_security.html

https://docs.openshift.com/container-platform/3.11/install_config/router/index.html

https://docs.openshift.com/container-platform/3.11/install_config/configuring_sdn.html

https://docs.openshift.com/container-platform/3.11/security/container_content.html

https://www.iso.org/standard/54533.html

https://www.iso.org/standard/54534.html

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

https://www.iso.org/standard/73906.html

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

http://www.coalfireiso.com/

ABOUT THE AUTHORS

Al Mahdi Mifdal | Principal, ISO/SOC Assurance, Coalfire

Mr. Mifdal is a certified ISO/IEC 27001 Master, Lead Auditor, and Lead Implementer and also maintains Certified Information Systems Auditor (CISA), Certified Information Security Manager (CISM), PCI Qualified Security Assessor (QSA), and CSA STAR Auditor designations. His ISO/IEC 27001, 27002, 27017, and 27018 implementations, pre-assessment, risk assessment, certification audit, and surveillance audit experience extends to Fortune 500 companies and leading service providers, such as IBM Cloud, Oracle, Microsoft, Google, F5 Networks, and Teradata.

Jason Macallister | Senior Consultant, Cyber Engineering, Coalfire

Mr. Macallister consults on Information Security and regulatory compliance topics as they relate to advanced infrastructure, emerging technology, and cloud solutions.

Mitch Ross | Contributor | Director, Cyber Engineering, Coalfire

As Director, Mr. Ross 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.

Published November 2018

ABOUT COALFIRE

Coalfire is the 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 more than 16 years and has offices throughout the United States and Europe. Coalfire.com

Coalfire is solely responsible for the contents of this document as of the date of publication. The contents of this document are subject to change at any time based on revisions to the applicable regulations and standards (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 us 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. In order to maintain contextual accuracy of this document, 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 prior written approval of the other party. If you have questions with regard to any legal or compliance matters referenced herein you should consult legal counsel, your security advisor and/or your relevant standard authority.

Red Hat OpenShift Container Platform Product Applicability Guide for ISO/IEC 27001:2013 November 2018