WHITEPAPER

RED HAT OPENSHIFT CONTAINER PLATFORM PRODUCT APPLICABILITY GUIDE FOR FISMA MODERATE

APPLICABILITY TO ASSIST AGENCIES IN FISMA MODERATE DEPLOYMENTS

BYRON ESTRADA | SEC+
CHRIS KRUEGER | CISSP

FINAL V3.0

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 comprehensive 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 the Red Hat OpenShift Container Platform (OpenShift Container Platform) 4.4 on Red Hat Enterprise Linux Core OS.

The purpose of this product applicability guide is to identify the alignment of Red Hat OpenShift on Red Hat Enterprise Linux with the security controls of the Federal Information Security Management Act (FISMA) of 2014 security controls based on a Moderate impact level categorization. Consideration for alignment is based on an analysis of the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53 Rev. 4 Moderate impact security baseline provided by the NIST SP 800-190, *Application Container Security Guide*.

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.

Red Hat OpenShift is a container platform that natively integrates open-source Linux container technologies and Kubernetes to combine them into an enterprise solution running on Red Hat Enterprise Linux Core OS. OpenShift Container Platform provides an application programming interface (API), web console, and command-line interface (CLI) to manage the underlying container technologies and Kubernetes to help users orchestrate the creation and management of containers. OpenShift Container Platform 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 government agencies or other entities that want to utilize container technologies within a FISMA Moderate compliance program framework. This guide discusses the relevant FISMA Moderate requirements that apply to OpenShift Container Platform on Red Hat Enterprise Linux Core OS. The focus is on technical controls that are pertinent to and in alignment with OpenShift Container Platform capabilities.

COALFIRE OPINION

Security controls, features, and functionality that are built into OpenShift Container Platform on Red Hat Enterprise Linux Core OS can support or address relevant technical FISMA Moderate requirements and address relevant sections of NIST SP 800-190 as outlined in the compliance applicability detail section of this guide as it pertains to the orchestration, management, monitoring, and scaling of containerized workloads.

UNDERSTANDING FISMA

FISMA is a United States federal law enacted in 2002 that mandates a process to strengthen the security posture of the federal government's information systems, bureaus, departments, and their supporting entities, such as vendors and subcontractors. When most agencies – and their vendors – discuss being "FISMA compliant," they refer to meeting the controls identified in NIST SP 800-53 Rev. 4. Federal government agencies, their vendors, and subcontractors that provide information systems to agencies must prove they are meeting FISMA requirements through annual assessments.

The law is enforced through various processes, as described by the Office of Management and Budget (OMB) Circular A-130, which establishes definitions, processes, and requirements for federal agencies to follow. FISMA recommends guidance issued by NIST, such as Federal Information Processing Standard (FIPS) 199, *Standards for Security Categorization of Federal Information and Information Systems*; FIPS 200, *Minimum Security Requirements for Federal Information and Information Systems*; NIST SP 800-53A, *Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans*; and NIST SP 800-53 Rev. 4, *Security and Privacy Controls for Federal Information Systems and Organizations*. The control selection, implementation, and testing are where information technology (IT) professionals responsible for FISMA compliance perform most of their work. Meeting the compliance requirements is necessary to receive an Authority to Operate (ATO) by government agencies.

THE FISMA SCOPE

Beyond a selection of security controls, FISMA requires each agency to develop, document, and implement an agency-wide program to provide information security for the information and information systems that support the operations and assets of the agency, including those provided or managed by another agency, contractor, or other sources. The selection of security controls is part of a more comprehensive risk management framework based on NIST SP 800-37 Rev. 2, *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy*. Figure 1 illustrates the major components of the risk management framework and the documents that guide each component.

Figure 1: NIST Risk Management Framework

The selection of controls is based on the categorization and assignment of impact relative to the category or categories of data being protected.

The goal of an agency's security program is to conduct day-to-day operations of the agency and accomplish the agency's stated missions with adequate security or security commensurate with risk, including the

magnitude of harm resulting from the unauthorized access, use, disclosure, disruption, modification, or destruction of information.

Within NIST SP 800-53 Rev. 4, baselines (Low, Moderate, High) have been established that determine the minimum set of requirements necessary to be authorized to manage data at these relative impact levels.

ADDITIONAL RELEVANT PUBLICATIONS

NIST SP 800-190, *Application Container Security Guide*, was developed in accordance with its statutory responsibilities under FISMA 44 U.S.C. § 3551. The purpose of this document is to explain the potential security concerns associated with container technologies and make practical recommendations for addressing those concerns when planning for the implementation and maintenance of containers. This publication is a guide that provides recommendations to help ensure the security of container technology implementations and usage.

RED HAT OPENSHIFT CONTAINER PLATFORM

Red Hat OpenShift offers a consistent hybrid cloud foundation for building and scaling containerized applications. OpenShift Container Platform delivers a single, consistent Kubernetes platform anywhere Red Hat Enterprise Linux Core OS runs. OpenShift Container Platform 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. OpenShift Container Platform provides both streamlined automated installations and options for more customized installations where organizations have requirements not met by the automated defaults. OpenShift Container Platform 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.

OpenShift Container Platform 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 OpenShift Container Platform:

- **Red Hat Enterprise Linux Core OS** – OpenShift Container Platform uses Red Hat Enterprise Linux Core OS, a container-optimized operating system (OS) specifically configured and optimized for running the platform. Red Hat Enterprise Linux Core OS is installed and managed as part of OpenShift Container Platform similarly to an appliance.

  Red Hat Enterprise Linux Core OS is a single-purpose, container- and Kubernetes-optimized, minimal-footprint OS powered by the same binaries as Red Hat Enterprise Linux. 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 OpenShift Container Platform environment.

  Red Hat Enterprise Linux Core OS is designed to be more tightly managed than a default Red Hat Enterprise Linux installation. Management is performed remotely from the OpenShift Container Platform cluster. When the OpenShift Container Platform cluster is set up, and Red Hat Enterprise Linux Core OS is deployed, customers can only modify a few system settings.

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

- **Operating Environment** – Red Hat 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 Red Hat Enterprise Linux Core OS 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 OpenShift Container Platform, 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, OpenShift Container Platform 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 OpenShift Container Platform 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 controller model at the level of entire applications and are, in effect, application-specific custom controllers.

  OpenShift Container Platform 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. OpenShift Container Platform also leverages integrated components to automate application builds, deployments, scaling, and health management. Included in the automation capabilities of OpenShift Container Platform is the ability to configure and deploy Kubernetes container host clusters.

OpenShift Container Platform Components

The following components are specific to Red Hat OpenShift.

- **Red Hat OpenShift Operators** – As OpenShift Container Platform is a fully containerized platform that consists of many different components, OpenShift Container Platform takes advantage of Operators for driving the installation, configuration, management, and upgrades of OpenShift Container Platform and all its services. Operators are both the fundamental unit of the Red Hat OpenShift cluster and a way to deploy applications and software components for the OpenShift Container Platform 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 OpenShift Container Platform Kubernetes platform. Operators serve as the platform foundation that removes the need for manual upgrades of the OSs and OpenShift Container Platform 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 OpenShift Container Platform 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 Red Hat Enterprise Linux Core OS nodes, monitor for drift, and alert on unsupported configurations.
- **Red Hat OpenShift Nodes** – Nodes are instances of Red Hat Enterprise Linux with the OpenShift Container Platform software deployed. Nodes are where end-user applications are run in containers. Nodes will contain the necessary OpenShift Container Platform 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.
- **Red Hat OpenShift Master** – The Master is the control plane for OpenShift Container Platform. The Master maintains and understands the state of the environment and orchestrates all activity that occurs on the Nodes. Similarly, to Nodes, the OpenShift Container Platform 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 Container Platform Master should not be scheduled to run application instances (pods). The following are the four functions of the OpenShift Container Platform Master:
  - **API and Authentication** – The Master provides a single API where all tooling and systems interact. Everything that interacts with OpenShift Container Platform 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

OpenShift Container Platform who have been granted access can be authorized to work with specific projects.

  - **Desired and Current State** – The state of OpenShift Container Platform is held in the OpenShift Container Platform data store. The data store uses, etcd, a distributed key-value store. The data store houses information about the OpenShift Container Platform environment, including OpenShift Container Platform user account information and the RBAC rules, the Red Hat OpenShift environment state, application environment information, important environment variables, secrets data, and other information.
  - **Scheduler** – The scheduler determines pod placement within OpenShift Container Platform. 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 JSON file in combination with Node labels to carve up OpenShift Container Platform resources. This allows the placement of pods within OpenShift Container Platform 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 Container Platform Master is also responsible for monitoring pods' health and scaling the pods as desired to handle the additional load. The OpenShift Container Platform Master executes liveness and readiness tests using probes that are defined by users. The OpenShift Container Platform Master can detect failed pods and remediate failures as they occur.

- **Service Layer –** The OpenShift Container Platform 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. OpenShift Container Platform automatically and transparently handles load balancing across the services' instances. In conjunction with probes, the OpenShift Container Platform 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. Red Hat 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 OpenShift Container Platform.
- **Ingress Controller** – The Ingress Controller is commonly used to allow external access to an OpenShift Container Platform 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.

- **Red Hat OpenShift SDN** – The OpenShift Container Platform software-defined network (SDN) is a unified cluster network that enables the communication between pods across the OpenShift Container Platform cluster. The OpenShift Container Platform SDN configures an overlay network that uses Open vSwitch (OVS). Red Hat currently provides one SDN mode for the OpenShift Container Platform 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 mandatory access control (MAC) style, where all traffic is denied by default unless a rule explicitly exists, even for pods and containers on the same host. The network policy mode is the default mode.
- **Red Hat OpenShift Registry** – The Red Hat OpenShift Registry provides integrated storage and management for sharing container images. Red Hat OpenShift can utilize existing OCI-compliant container registries accessible to the Nodes and the Red Hat 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 Red Hat 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 OpenShift Container Platform 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 Red Hat OpenShift hosts, even for complex application debugging and troubleshooting tasks.

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

  – **Regular Users:** A user most commonly represents a real person who interacts with the OpenShift Container Platform cluster in some way. Since the OpenShift Container Platform 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 OpenShift Container Platform 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 OpenShift Container Platform annotations and metadata. It is the central vehicle by which access to regular users' resources is managed and is the tenancy model of OpenShift Container Platform 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 Container Platform 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

OpenShift Container Platform 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. OpenShift Container Platform runs on Red Hat Enterprise Linux Core OS 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 Container Platform 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, OpenShift Container Platform 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.

OpenShift Container Platform 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 a trusted distribution of content. The security of Red Hat 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, OpenShift Container Platform 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 Container Platform security features and capabilities:

- **Identity and Access Management** – OpenShift Container Platform 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. OpenShift Container Platform 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 OpenShift Container Platform user object represents an actor that may be granted permissions in the system by adding roles to the users or their groups. OpenShift Container Platform 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 OpenShift Container Platform itself, and all contained OpenShift Container Platform 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).

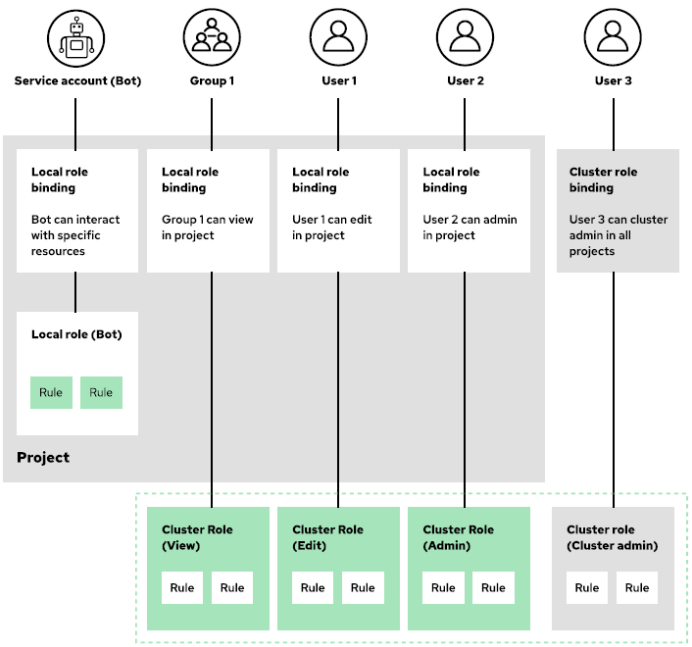

Figure 3: RBAC with Red Hat OpenShift Projects

- **Encryption** – Data in motion and data at rest is protected by encryption. OpenShift Container Platform 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 OpenShift Container Platform, 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 Red Hat Enterprise Linux Core OS 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 OpenShift Container Platform, Red Hat Enterprise Linux Core OS nodes must be configured for FIPS mode at the installation time. FIPS mode ensures that the OpenShift Container Platform components call only FIPS cryptographic modules. Custom containers built with Red Hat Enterprise Linux or UBI base images can be configured for FIPS compliance. Red Hat Enterprise Linux 8 and Red Hat Enterprise Linux Core OS cryptographic libraries have the status of Modules in Process per the NIST Computer Security Resource Center (CSRC).
- **Cluster Logging** – The OpenShift Container Platform cluster administrators can deploy cluster logging using a few CLI commands and the OpenShift Container Platform web console to install the Elasticsearch Operator and Cluster Logging Operator. The cluster logging components are based upon Elasticsearch, Fluentd, and Kibana (EFK). OpenShift Container Platform 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 log collection and 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 Red Hat Enterprise Linux and Red Hat Enterprise Linux Core OS.

- **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 other certified ISV partners.
- **OperatorHub** – The Operator Hub provides access to certified Kubernetes operators.
- **Container Registries** – OpenShift Container Platform includes an integrated container registry that provides basic functionality supporting build and deployment automation within the cluster, tied into the OpenShift Container Platform 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** – OpenShift Container Platform integrates tightly with Jenkins and can be easily integrated with other CI/CD tools to manage builds, code inspection, code scanning, and validation. OpenShift Container Platform includes Tekton for building Kubernetes native pipelines.
- **Deploying Containers** – By default, OpenShift Container Platform prevents containers from running as root or other specifically-named users. Also, OpenShift Container Platform enables granular deployment policies that allow operations, security, and compliance teams to enforce quotas, isolation, and access protections.
- **Container Orchestration** – OpenShift Container Platform integrates secure operational capabilities to support trust between users, applications, and security policies. Red Hat 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** – OpenShift Container Platform uses an SDN approach to provide a unified cluster network that enables the communication between pods across the Red Hat OpenShift cluster. OpenShift Container Platform uses the Multus CNI plug-in to allow chaining of CNI plug-ins. This Pod network is established and maintained by the Red Hat 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 Red Hat 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 plugins 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 Container Platform provides access to and integration with a broad range of storage platforms and protocols, allowing applications to store and encrypt application data securely. Red Hat OpenShift provides the option to encrypt sensitive data stored in, etcd, and encrypt the Red Hat Enterprise Linux Core OS volumes.
- **Red Hat OpenShift Service Mesh** – An optional feature of OpenShift Container Platform is 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 OpenShift Service Mesh, the customer can connect, secure, and monitor microservices in the OpenShift Container Platform environment. OpenShift Service Mesh adds a transparent layer on existing distributed applications without requiring any service code changes. The customer can add 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 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 OpenShift Service Mesh include A/B testing, canary releases, rate limiting, access control, and end-to-end authentication.
- **API Management** – Red Hat OpenShift and OpenShift Service Mesh can be integrated with the Red Hat 3scale API Management platform to authenticate, secure, and rate-limit API access to applications and services.
- **Updates** – Red Hat Enterprise Linux Core OS and OpenShift Container Platform 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 Red Hat 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 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 Container Platform 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 technology's applicability against the high-level FISMA Moderate impact baseline control objectives. This alignment was further narrowed down to specific requirements that were considered applicable to either Red Hat OpenShift or the underlying OS. An analysis of capability for the reviewed technology to address the applicable requirements was then conducted. This analysis primarily focused on what an assessor might review when following the FISMA Moderate testing guide during an assessment of applicable requirements.

SCOPE OF TECHNOLOGY AND SECURITY STANDARD TO REVIEW

Coalfire was tasked by Red Hat to review OpenShift Container Platform as deployed on Red Hat Enterprise Linux. The primary focus of the review included the components, features, and functionality of Red Hat 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 the assessment application pods or containers that an organization may deploy on Red Hat OpenShift. Containers deployed in the lab environment were used to demonstrate the platform's orchestration, deployment, and management capabilities. Furthermore, Coalfire did not assess available image registries or repositories that may be used for acquiring applications, services, dependencies, or other elements to be hosted on or used within OpenShift Container Platform.

For this review, Coalfire included requirements from NIST SP 800-53 Rev. 4, available from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf. For a broader understanding of the requirements and their applicability to technical solution implementation, Coalfire also reviewed supporting documentation provided by NIST, including assessment guidance. Applied knowledge of NIST SP 800-53 Rev. 4 requirements and guidance was supplemented by documentation and guidance on relevant subjects, many of which are referenced in the NIST SP 800-53 Rev. 4 requirements. As OpenShift Container Platform is a container platform, Coalfire also applied guidance from NIST SP 800-190.

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. The achievement of the required 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 Container Platform 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 essential factors to consider as it pertains to OpenShift Container Platform. Instead, OpenShift Container Platform does not natively 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 the requirement. For applicable requirements, Coalfire designated a qualitative category of capability, including whether the solution was determined to support the requirement, partially support the requirement fully, or unable to support the requirement. In cases where the requirement was determined to be applicable but unsupported, additional thought for using third-party solutions may be considered.

OPENSHIFT CONTAINER PLATFORM APPLICABILITY TO FISMA MODERATE

The table that follows details controls where OpenShift Container Platform applies to FISMA Moderate. This applicability applies either as an agency configurable item within Red Hat OpenShift or as a native and default capability to address the control requirement.

Overall, there are general statements of applicability that can be made for Red Hat OpenShift for each of the FISMA Moderate control families.

For the Access Control (AC) and Identity and Authentication (IA) families of controls, OpenShift Container Platform will primarily be dependent on an external third-party identity and authentication provider. OpenShift Container Platform provides several integration options for third-party identity and authentication services. Users and groups would be created in the external identity provider directory and assigned appropriate authenticators. The external provider would also be responsible for managing authenticators. The external provider would first authenticate users before gaining access to the Red Hat OpenShift API, web console, or CLI. The user is then tied to an identity within OpenShift Container Platform for authorization. OpenShift Container Platform provides services to enable this integration. The agency should determine which option for identity and authentication best suits their use case with the understanding that some options may fit better within the context of a security program intended to address FISMA Moderate impact level controls baseline.

Users and groups established within the third-party identity and authentication service can be assigned RBACs within Red Hat OpenShift. Administrators can use the CLI or web console to view RBAC resources and manage the roles and bindings. Roles can be used to grant various levels of access, both cluster-wide and at the project scope. Users and groups can be bound to multiple roles at the same time. The agency should carefully plan the implementation of OpenShift Container Platform regarding assigning roles and responsibilities to individuals and groups. It will be important for the organization to ensure that there is not uncontrolled administrative access provided to the container platform. Understanding the application of access controls to the platform and the projects will be important for ensuring that users or systems do not gain unauthorized access.

Awareness and Training (AT) requirements are also crucial to the security of OpenShift Container Platform. The agency should make sure that all personnel responsible for managing and using Red Hat OpenShift (administrators, developers, operations personnel) be trained in the proper design, implementation, and management of OpenShift Container Platform and the operational nuances specific to the agency's security program. The awareness of training of personnel engaged with OpenShift Container Platform should allow the community of users to be better equipped to identify potential issues with implementation and operation that could pose an increased risk.

The agency will be responsible for identifying auditable events (AU) relevant to Red Hat OpenShift as well as for the projects and containers that are orchestrated by the platform. While Red Hat OpenShift can produce audit logs with records necessary to support FISMA Moderate requirements, the handling of audit logs and events will best be served by a third-party tool that collects or aggregates logs from multiple sources and correlates activities to identify security events from a broader perspective than just the container platform. The agency will also be responsible for coordinating the log generation capabilities of the workloads that are deployed on the platform.

The agency should also consider the impacts and relevance of OpenShift Container Platform on their overall security program. Customers who are newly implementing OpenShift Container Platform should ensure that the culture and methods of containerization are considered when evaluating their Security

Assessment and Authorization (CA) policies and processes. Traditional security approaches have been less and less relevant to information systems as they become more agile, portable, flexible, scalable, and dynamic. As layers of the information systems are becoming increasingly abstracted, it will be necessary for the organization to implement hardening tactics to reduce the surface areas of attack. Securing the containers themselves becomes increasingly challenging to traditional security methods, as containers and container environments do not often provide as concrete a security boundary as physical and virtual machines.

Proper Configuration Management (CM) techniques relative to the software development and IT operations (DevOps) nature of containerization will also be essential to the environment's security. The organization will need to consider the software development lifecycle and security development lifecycle as it pertains to the use of Red Hat OpenShift.

Gates should be identified and managed whereby applications or application components are continually developed, tested, and deployed into production. Testing should include both application function and application security to limit the possibility of vulnerable code, libraries, or other application components from being introduced to the environment.

Once tested and approved, images from which containers are spawned should be digitally signed to prevent tampering. The organization should establish techniques to validate and establish baselines and standards by which containers are measured and implemented in the environment. To define, document, select, and implement configuration management policies, procedures, standards, controls, and management tools, the organization should consider the rapid and dynamic rate of change that can occur in container environments.

The portable nature of containers can be useful to organizations' Contingency Planning (CP) efforts. The organization should consider the container platform components that are necessary for recovery in case of failure and plan accordingly.

The agency should consider the implications of risk associated with the implementation and operationalization of Red Hat OpenShift, including the deployment of workloads or applications on the platform. A sound understanding of risk will help the agency properly implement mitigating controls or safeguards to reduce the residual risk to an acceptable level. The agency should include the selection and implementation of controls for Red Hat OpenShift and the applications it supports into their system security planning and establishes action items and milestones for continuous improvement of security per the Risk Assessment (RA) family of requirements.

The agency should understand the commitment that Red Hat has to its customers regarding its development of enterprise-class software. As part of the System and Services Acquisition (SA), the agency is responsible for ensuring that their vendors follow best practices for the development of the systems they provide to their customers. It will be necessary for the agency to address discovered vulnerabilities and apply security patches as Red Hat delivers them promptly.

Coalfire recommends that OpenShift Container Platform be implemented within the defined external FISMA Moderate authorization boundary with edge protections (e.g., routers, firewalls, IDS/IPS, WAF) between the Internet and out of scope systems and the in-scope systems. For east-west communications internal to the container environment, OpenShift Container Platform provides built-in SDN capabilities to provide management, control, and security of the data plane. These capabilities allow the organization to define internal boundaries between applications or application components. The agency should consider what is necessary to achieve the security posture that aligns with its compliance program.

The agency should consider approaches to System and Information Integrity (SI) requirements relevant to the nature of container environments and their workloads. Traditional toolsets (e.g., antivirus protections) may not apply naturally to OpenShift Container Platform or the supported workloads. The agency should choose controls, including tools that enable prevention, detection, and correction, designed for containerized workloads and their infrastructures. To help protect workloads, OpenShift Container Platform can be configured to selectively orchestrate workloads such that workloads of differing sensitivity levels run on different Nodes. These configuration options reduce the possibility of a lower security workload compromising a higher security workload. Additionally, OpenShift Container Platform has the capability of utilizing the security features of Red Hat Enterprise Linux. For example, SELinux is enabled by default, and containers are deployed with appropriate SELinux contexts to isolate the container's processes and prevent a rogue or exploited container from negatively impacting other containers, their persistent storage, or the container host.

OPENSHIFT CONTAINER PLATFORM FISMA MODERATE APPLICABILITY DETAIL

The following table provides detailed alignment of OpenShift Container Platform's capabilities for alignment and to address FISMA Moderate control requirements. The requirements that are listed are the requirements that Coalfire determined to be applicable for Red Hat OpenShift. Other FISMA requirements that are not listed in the following table are either not applicable to the technology or are intended to be addressed as a control response by the agency rather than the information system.

ID	CONTROL TITLE	CONTROL DESCRIPTION	APPLICABILITY OF RED HAT OPENSHIFT TO CONTROL
AC-3	Access Enforcement	The information system enforces approved authorizations for logical access to information and system resources per applicable access control policies.	Red Hat OpenShift uses granular RBAC 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 role. This allows platform administrators to use cluster roles and bindings to control who has various access levels to OpenShift Container Platform itself and all contained projects and resources and will enable developers to use local roles and bindings to control who has access to their projects. Authorization is managed using rules, roles, and bindings. Rules are a set of permitted verbs (actions) on a set of objects (e.g., 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. Anonymous requests are allowed but are subject to authorization with only specific permissions granted.
AC-4	Information Flow Enforcement	The information system enforces approved authorizations for controlling the flow of information within the system and between interconnected systems based on [*Assignment:*	Coalfire recommends placing OpenShift Container Platform on the inside of the overall information system's external boundary with boundary protections (e.g., firewalls, routers, application gateways, web proxies, web gateways, IDS/IPS) provided externally to OpenShift Container Platform. In this way, the information flow control to and from the Internet or uncontrolled network is

		organization-defined information flow control policies].	provided by the organization's third-party solution (e.g., firewalls, IDS/IPS, routers, application gateways, web proxies, web gateways). For communication between endpoints within OpenShift Container Platform, container networking capabilities control the flow of information within the container environment and between pods. Kubernetes tells a Node to attach a pod to a network. The network plug-in then allocates the pod an IP address from an internal network. This allocation ensures that all containers within the pod behave as if they are on the same host and causes all containers within a pod to share a network space. Giving each pod an IP address means that pods can be treated like physical hosts or virtual machines in port allocation, networking, naming, service discovery, load balancing, application configuration, and migration. OpenShift Container Platform uses an SDN approach to provide a unified cluster network that enables the communication between pods across the Red Hat OpenShift cluster. This pod network is established and maintained by the Red Hat OpenShift SDN, which configures an overlay network using OVS. Almost all packet delivery decisions are performed with OpenFlow rules in the OVS bridge br0, which provides flexible routing. Depending on the deployed SDN plug-in, network isolation can be enforceable and finely controlled. The OpenShift Container Platform SDN enables micro-segmentation with Kubernetes Network Policies. Pod networks can be isolated by project using network namespaces when Network Policies are configured in Red Hat OpenShift's multitenant mode. While each pod receives an IP and port range to bind to, the multitenant mode ensures each pod project has its own virtual network within the SDN, thereby isolating pod networks from each other even on the same Node. The default behavior for the multitenant mode is that pods from different projects cannot send packets to or receive packets from a different project's pods and services. This can be used to isolate development, test, and production environments within a cluster. Alternatively, a separate cluster can be deployed for production use.
AC-6 (10)	Prohibit Non-Privileged Users from Executing Privileged Functions	The information system prevents non-privileged users from executing privileged functions, including disabling, circumventing, or altering implemented security safeguards/countermeasures.	OpenShift Container Platform can prevent non-privileged users from executing privileged functions, including disabling, circumventing, or altering implemented security safeguards and countermeasures. Interaction with OpenShift Container Platform is associated with a user. An OpenShift Container Platform user object represents an actor that may be granted

			permissions in the system by adding roles to them or their groups. The users that can exist include regular users and service accounts. OpenShift Container Platform is configured with RBAC by default. RBAC allows for the granular determination of access for users or groups of users. RBAC objects determine whether a user can perform a given action system-wide. This allows platform administrators to use cluster roles and bindings to control who has various access levels to OpenShift Container Platform itself and all contained projects. It allows developers to use local roles and bindings to control who has access to their projects. Authorization is managed using rules, roles, and bindings. Rules are a set of permitted verbs (actions) on a set of objects, for example, 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.
AC-8	System Use Notification	The information system: a. Displays to users [*Assignment: organization-defined system use notification message or banner*] before granting access to the system that provides privacy and security notices consistent with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance and states that: a.1. Users are accessing a U.S. Government information system; a.2. Information system usage may be monitored, recorded, and subject to audit; a.3. Unauthorized use of the information system is prohibited and subject to criminal and civil penalties; and a.4. Use of the information system indicates consent to monitoring and recording; b. Retains the notification message or banner on the	Typically, users accessing the Red Hat OpenShift Web Console, CLI, or API would be doing so from an organizationally controlled machine where system use notifications would be presented to the user relative to access to the enterprise-wide network. The organization is responsible for establishing the system's content using notifications and defining where the system use notifications would be displayed. The choice in where system use notifications are displayed should consider the requirement to retain the notification banner on the screen until the user acknowledges the usage conditions and takes explicit action to login. For a FISMA deployment, it is not recommended to make the Red Hat OpenShift API, CLI, or Web Console directly publicly available. The Red Hat OpenShift Web Console can be customized to allow the organization to display a system use notification. However, the external authentication provider would more likely provide the system use notification with flow control that requires acceptance before logon or notify the user that logon indicates acknowledgment and acceptance of the system use notification.

		screen until users acknowledge the usage conditions and take explicit actions to log on to or further access the information system; and c. For publicly accessible systems: c.1. Displays system use information [*Assignment: organization-defined conditions*], before granting further access; c.2. Displays references, if any, to monitoring, recording, or auditing that is consistent with privacy accommodations for such systems that prohibit those activities; and c.3. Includes a description of the authorized uses of the system.	
AC-11	Session Lock	The information system: a. Prevents further access to the system by initiating a session lock after [*Assignment: organization-defined time period*] of inactivity or upon receiving a request from a user; and b. Retains the session lock until the user reestablishes access using established identification and authentication procedures.	OpenShift Container Platform provides the capability to enable a "soft" setting for inactivity or idle timeout. The soft setting is in masterConfig.oauthConfig.tokenConfig.accessTokenInactivityTimeoutSeconds. A hard setting option is available with the token expiration options in sessionConfig - sessionMaxAgeSeconds, which controls the maximum age of a session, after which the session token expires, and the user must log in again.
AC-11 (1)	Pattern-Hiding Displays	The information system conceals, via the session lock, information previously visible on the display with a publicly viewable image.	This is primarily applicable to the web console and CLI. After a specified period of idle time or when the user voluntarily disconnects, the session is terminated. The user would then be required to re-authenticate before further interaction with the interfaces.
AC-12	Session Termination	The information system automatically terminates a user session after [*Assignment: organization-defined conditions or trigger events requiring session disconnect*].	The organization can define session idle timeouts as well as session timeouts, after which re-authentication is required. The organization will be responsible for defining events or conditions requiring session termination. Through integration with the external authentication and authorization provider, the information system should respond accordingly by terminating sessions for events such as users being removed from groups that previously authorized access, or for user's accounts being disabled or deleted from the directory.

AU-3	Content of Audit Records	The information system generates audit records containing information that establishes what type of event occurred, when the event occurred, where the event occurred, the source of the event, the outcome of the event, and the identity of any individuals or subjects associated with the event.	OpenShift Container Platform is capable of supporting audit requirements by generating auditable events that contain information to establish what type of event occurred, when the event occurred, where the event occurred, the source of the event, the outcome of the event, and the identity of any individuals or subjects associated with the event.
AU-3 (1)	Additional Audit Information	The information system generates audit records containing the following additional information: [*Assignment: organization-defined additional, more detailed information*].	OpenShift Container Platform does not apply to the selection of additional auditable events by the organization; however, there is an expectation that a request for additional auditable events would be supported by OpenShift Container Platform as requested. OpenShift Container Platform may support organizations that consider limiting the additional audit information to only that information explicitly needed for specific audit requirements. A table of audit log fields for OpenShift Container Platform can be found at the following link: https://docs.openshift.com/container-platform/4.5/nodes/nodes/nodes-nodes-audit-log.html.
AU-8	Timestamps	The information system: a. Uses internal system clocks to generate timestamps for audit records; and b. Records time stamps for audit records that can be mapped to Coordinated Universal Time (UTC) or Greenwich Mean Time (GMT) and meets [*Assignment: organization-defined granularity of time measurement*].	OpenShift Container Platform is dependent on time synchronization to support sensitive operations, such as log keeping and time stamps. OpenShift Container Platform uses the time of the underlying host. By default, OpenShift Container Platform uses NTP to synchronize all Masters and Nodes. This is performed via the chrony RPM package. The Machine Config operator manages configuration. Additional information is available in the documentation. OpenShift Container Platform operations include etcd, leader election and health checks for pods, and can help prevent time skew problems.
AU-12	Audit Generation	The information system: a. Provides audit record generation capability for the auditable events defined in AU-2 a. at [*Assignment: organization-defined information system components*]; b. Allows [*Assignment: organization-defined personnel or roles*] to select which auditable events are to be audited by specific components of the information system; and	OpenShift Container Platform can generate audit records and events on actions taken by users or system components against the API, CLI, or Web Console interface. OpenShift Container Platform also generates log data and performance metrics relative to its normal operation. The organization is responsible for determining the information system's capabilities for auditing events that are defined by the organization. Once those capabilities are understood, the organization is responsible for ensuring that OpenShift Container Platform is configured properly to generate the required and defined logs and events. Coalfire recommends that the selection of auditable

		c. Generates audit records for the events defined in AU-2 d. with the content defined in AU-3.	events be managed through an external Syslog or SIEM solution. The organization will need to determine how the audit records are to be generated and how they meet the requirements defined in AU-2d and AU-3.
CP-9	Information System Backup	The organization: a. Conducts backups of user-level information contained in the information system [*Assignment: organization-defined frequency consistent with recovery time and recovery point objectives*]; b. Conducts backups of system-level information contained in the information system [*Assignment: organization-defined frequency consistent with recovery time and recovery point objectives*]; c. Conducts backups of information system documentation including security-related documentation [*Assignment: organization-defined frequency consistent with recovery time and recovery point objectives*]; and d. Protects the confidentiality, integrity, and availability of backup information at storage locations.	The organization can perform backups of Red Hat OpenShift to save state to separate storage. Red Hat provides documentation detailing methods for back up and restoration of Red Hat OpenShift.
IA-6	Authenticator Feedback	The information system obscures feedback of authentication information during the authentication process to protect the information from exploitation/use by unauthorized individuals.	Red Hat OpenShift natively obscures authentication information during the authentication process.
RA-5 (5)	Privileged Access	The information system implements privileged access authorization to [*Assignment: organization-identified information system components*] for selected [*Assignment: organization-defined vulnerability scanning activities*].	Container scanning tools can leverage continuously updated vulnerability databases to ensure that the agency always has the latest information on known vulnerabilities for container content. Red Hat OpenShift provides a pluggable API to support multiple scanners. Moreover, Red Hat Quay, an optional container registry sold by Red Hat, can be used to scan container images for known vulnerabilities.

			Red Hat OpenShift enables the leveraging of such scanners with the CI/CD process. Static code analysis tools can be integrated into the CI/CD pipeline to test for security flaws in source code and software composition analysis tools that identify open-source libraries to provide metadata on those libraries, including known vulnerabilities can also be integrated. Red Hat OpenShift makes use of object annotations to extend functionality. External tools, such as vulnerability scanners, may annotate image objects with metadata to summarize results and control pod execution.
SC-4	Information in Shared Resources	The information system prevents unauthorized and unintended information transfer via shared system resources.	Red Hat OpenShift uses features of the underlying OS to isolate the workloads to their processor, memory, and storage spaces. Unless the container is authorized to access or share a specific resource space, the container should not access that resource. In addition to the RBAC resources that control what a user can do, Red Hat OpenShift provides security context constraints (SCCs) that control the actions that a pod can perform and access. SCCs are objects that define a set of conditions that a pod must adhere to, to be accepted into the system. They allow an administrator to control: running of privileged containers, capabilities a container can request to be added, use of host directories as volumes, the SELinux context of the container, the user ID, the use of host namespaces and networking, allocating an FSGroup that owns the pod's volumes, configuring allowable supplemental groups, requiring the use of read-only root file system, controlling the usage of volume types, and configuring allowable seccomp profiles. SCCs are useful for managing access to host storage and can be managed by administrators. Additionally, seccomp can be used to restrict the set of system calls that applications can make. Sysctl settings are exposed via Kubernetes, allowing users to modify certain kernel parameters at runtime for namespaces within a container. Only sysctls that are namespaced can be set independently on pods. If a sysctl is not namespaced, called node-level, it cannot be set within Red Hat OpenShift. Moreover, only those sysctls considered safe are whitelisted by default; the agency can manually enable other unsafe sysctls on the Node to be available to the user.
SC-7	Boundary Protection	The information system: a. Monitors and controls communications at the external boundary of the system and key internal boundaries within the system; b. Implements subnetworks	The findings related to Red Hat OpenShift pertain primarily to key internal boundaries within a system for this control requirement. For external boundaries to the system, it is recommended to deploy Red Hat OpenShift within an external authorization boundary controlled by traditional network security approaches.

		for publicly accessible system components that are [*Selection: physically; logically*] separated from internal organizational networks; and c. Connects to external networks or information systems only through managed interfaces consisting of boundary protection devices arranged following an organizational security architecture.	For internal boundaries that may be defined by the agency for separation of workloads, Red Hat OpenShift uses the Red Hat OpenShift SDN to provide a unified cluster network that enables the communication between containers across the cluster. The level of control enabled for communication depends on the chosen SDN, the network policy configuration, and the use case for which the agency may choose it.
SC-8	Transmission Confidentiality and Integrity	The information system protects the [*Selection (one or more): confidentiality, integrity*] of transmitted information.	For Red Hat Enterprise Linux 7, the NIST Cryptographic Module Validation Program (CMVP) certificate number for this FIPS validated cryptographic module is 3563. Red Hat Enterprise Linux Core OS is built with Red Hat Enterprise Linux 8 libraries, currently identified as Modules In Process with the NIST CMVP. Red Hat OpenShift control plane components enforce industry standards, including HTTP/2 defaults and TLS 1.2 or TLS 1.3. Red Hat OpenShift out of the box provides containerized, stateless HAProxy as a default router for the container ecosystem. Red Hat OpenShift provides ways to perform TLS termination with secure routes: edge, re-encryption, and passthrough. With the passthrough route, TLS termination is handled at the pod level.
SC-8 (1)	Transmission Confidentiality and Integrity, Cryptographic or Alternate Physical Protection	The information system implements cryptographic mechanisms to [*Selection (one or more): prevent unauthorized disclosure of information; detect changes to information*] during transmission unless otherwise protected by [*Assignment: organization-defined alternative physical safeguards*]	Reference the above control, SC-8. Red Hat OpenShift support for TLS 1.2 and 1.3 communications satisfies this control for internal communications and across boundary interconnections both for native web services and containerized application transmissions. Regarding Ingress Cluster traffic, services accessed via HTTP/HTTPS, or TLS-encrypted protocols other than HTTPS (such as TLS with the SNI header), can use Ingress Controllers along with Ingress or Route resources to allow the service to be accessible from clients located outside of the cluster. For egress traffic, OpenShift Container Platform provides options for controlling the traffic leaving the cluster: egress firewall, egress routers, and egress static IP.
SC-12	Cryptographic Key Establishment and Management	The organization establishes and manages cryptographic keys for required cryptography employed within the information system in accordance with [*Assignment: organization-defined requirements for key*	Red Hat OpenShift supports key lifecycle requirements for the generation and distribution of keys. The OpenShift Container Platform includes multiple certificate authorities (CAs), providing independent chains of trust, increasing the cluster's security posture. These internal self-signing CAs enable automation because the key is known to the cluster. The certificates generated by each CA are used to identify a particular Red Hat OpenShift

		generation, distribution, storage, access, and destruction.]	platform component to another Red Hat OpenShift platform component. CA bundles are used when more than one communication path needs to be authenticated. Third-party Key Management Systems (KMS) would be recommended to augment any Agency solution, which relies on inherent support provided by Red Hat Core OS and Red Hat OpenShift for custom deployed applications.
SC-13	Cryptographic Protection	The information system implements [*Assignment: organization-defined cryptographic uses and type of cryptography required for each use*] in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards.	Red Hat OpenShift and Red Hat Core OS supply FIPS-140 Level 1 cryptography to satisfy this control both programmatically and for feature use.
SC-28	Protection of Information at Rest	The information system protects the [*Selection (one or more): confidentiality; integrity*] of [*Assignment: organization-defined information at rest*].	Red Hat Enterprise Linux Core OS disks can be encrypted. Data at the OpenShift Container Platform datastore layer can be configured to be encrypted. The capability allows for secrets (i.e., passwords, OAuth tokens, SSH keys) to be encrypted when at rest in etcd. Once enabled, encryption cannot be disabled. For applications, one way of passing secrets to containers is by mounting a volume into a container that contains a file with the secrets. Third-party integrations are also available to provide encryption capability for data at rest. Data at the etcd datastore layer may be encrypted using the AES-CBC cipher. Red Hat OpenShift automatically generates and manages the encryption key. Red Hat Enterprise Linux Core OS supports full-disk encryption for the system disk. As implemented, Red Hat Enterprise Linux Core OS disk encryption is FIPS compliant if FIPS mode is enabled. Red Hat Enterprise Linux Core OS 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.
SC-39	Process Isolation	The information system maintains a separate execution domain for each executing process.	Red Hat OpenShift makes use of underlying features of the OS to support process isolation.
SI-16	Memory Protection	The information system implements [*Assignment: organization-defined security safeguards*] to protect its memory from unauthorized code execution.	Red Hat OpenShift makes use of underlying features of the OS to support memory protection.

Table 1: FISMA Moderate Applicability Details for OpenShift Container Platform

CONCLUSION

OpenShift on Red Hat Enterprise Linux Core OS, as reviewed by Coalfire, can be useful in providing support for the outlined objectives and requirements of NIST 800-53 Rev. 4 Moderate baseline in support of a FISMA compliance program and with new compliance measures sought by the FIPS cryptographic components. Through proper implementation and integration into the organization's more significant technical infrastructure and information security management systems, OpenShift Container Platform may be useable in a FISMA Moderate controlled environment.

Care should be given for the implementation of OpenShift Container Platform regarding its platform for deploying containers in support of microservice architectures. An organization desiring to use OpenShift Container Platform should consider the guidance provided by NIST SP 800-190 when designing their implementation. The FIPS-series documents discussed at the beginning of the document may also be used to comply with FIPS compliance under the FISMA set of requirements.

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, including adherence to vendor best practices and hardening of configuration 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 security planning, and the overall organizational security and compliance 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 white paper 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 Red Hat 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 Red Hat OpenShift. These details are available at the following link: https://docs.openshift.com/container-platform/4.4/welcome/index.html.

NATIONAL INSTITUTE FOR STANDARDS AND TECHNOLOGY

NIST 800-53 Rev. 4 provides a catalog of security and privacy controls for federal information systems and organizations and a process for selecting controls to protect organizational operations (i.e., mission functions, image, and reputation). The publication can be found at the following link: https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final.

NIST 800-190 discusses application container technologies, also known as containers. Containers are a form of operating system virtualization combined with application software packaging. The document can be found at the following link: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf.

NIST 800-53A Rev. 4 provides a set of procedures for conducting assessments of security controls and privacy controls employed within federal information systems and organizations. The assessment procedures, executed at various phases of the system development lifecycle, are consistent with the security and privacy controls in NIST 800-53 Rev. 4. The publication can be found at the following link: https://csrc.nist.gov/publications/detail/sp/800-53a/rev-4/final.

FEDERAL INFORMATION SECURITY MANAGEMENT ACT

The FISMA implementation project information provides more essential details from the NIST Computer Security Resource Center (CRSC). Appendix G in NIST 800-53 Rev. 4 explains the FISMA requirements where organizations must comply. The publication can be found at the following link: https://csrc.nist.gov/Projects/risk-management/rmf-overview/security-controls.

OMB Circular A-130 underscores the mandatory nature of certain security and privacy controls while also enhancing the role of agency privacy officials in IT systems authorizations. One version of the publication has been published and can be found at the following link:https://obamawhitehouse.archives.gov/sites/default/files/omb/assets/OMB/circulars/a130/a130revised.pdf.

FEDERAL INFORMATION PROCESSING STANDARD

The hyperlinks embedded within the following FIPS standards provide more detail into this document's FIPS content. It was used as reference detail on how Red Hat OpenShift applies FIPS controls: FIPS 199 can be found at the following link: https://csrc.nist.gov/publications/detail/fips/199/final. FIPS 200 can be found at the following link: https://csrc.nist.gov/publications/detail/fips/200/final.

ABOUT THE AUTHOR

Byron Estrada | Senior Consultant, Solutions Engineering, Coalfire Systems

As a Senior 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.

Jason Macallister | Senior Consultant, Coalfire Systems

Jason consults on information security and regulatory compliance topics as they relate to advanced infrastructure and emerging products and solutions. Jason was the author of the previous Red Hat whitepaper titled *Red Hat OpenShift Container Platform Applicability Guide for FISMA Moderate v1.0.*

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.

December 8, 2020