CHAPTER 3

# System Design

This chapter describes the topology, hardware, software, and configuration of the design as tested. While configurations in this document may be used as guidelines for other configurations, results reflect hardware and software configurations exclusive to testing conducted for this document.

# IaaS Topology

As mentioned in Chapter 2, "System Overview", the scope of this document does not include the design and validation of the IaaS layer components. The test bed for OpenShift integration requires an IaaS layer component. OpenStack is used to provide the test bed. For the sake of completeness, the design of the IaaS test bed dependency is outlined below.

## Hardware and Software Components of the Architecture

Table 3-1 describes hardware and software components.

*Table 3-1 Architectural Hardware and Software Components*

| Vendor | Name | Specification | Purpose |
|---|---|---|---|
| Cisco | Cisco UCS B200 M3 Server | 2.1(3a)B 2CPU 16 Cores,96GB Mem,1TB Storage(1xHDD) | OpenStack Controller |
| Cisco | Cisco UCS B200 M3 Server | 2.1(3a)B 2CPU 16 Cores,96GB Mem,1TB Storage(1xHDD) | OpenStack Compute, Network Node |
| Cisco | Cisco UCS VIC 1240, 1280 | VIC 1240, 1280 | Cisco Virtual Interface Card (adapter) firmware |
| Cisco | Cisco UCS 6248UP Fabric Interconnect | Kernel Version: 5.0(3)N2(2.1.1.3a) | Blade Network and Management |
| Red Hat | Red Hat OpenStack Platform | RHOS 4 (Havana) | OpenStack IaaS |
| Red Hat | Red Hat OpenShift Enterprise | OSE 2.0 | OpenShift PaaS |

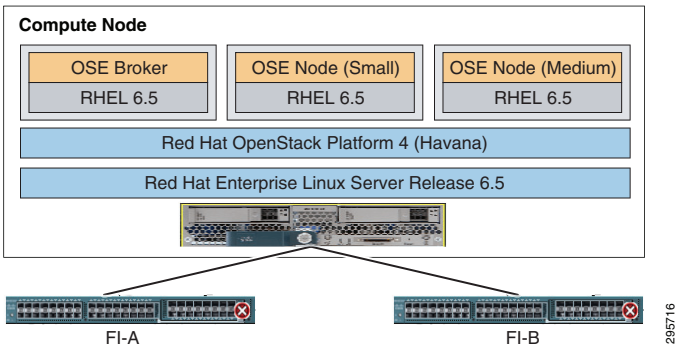
# OpenStack Services Placement

Table 3-2 describes OpenStack services placement. Figure 3-1 depicts the hardware/software stock.

*Table 3-2 OpenStack Services Placement*

| Host Name | Role | Services |
|---|---|---|
| node01 | OpenStack Compute | httpd (horizon), mongod, mysqld, qpid, tgtd, ntpd, neutron-server, cinder-, glance-, nova-api,<br>nova-conductor, nova-scheduler, nova-console, nova-metadata-api, nova-novncproxy, ceilometer-*, heat-* |
| node02 | OpenStack Controller, Network Node | libvirtd, ntpd, messagebus, openvswitch, dnsmasq, neutron-dhcp-agent, neutron-l3-agent,<br>neutron-metadata-agent,neutron-openvswitch-agent,ceilometer-compute |

*Figure 3-1 Hardware/Software Stack for the Compute Node used for Testing*




# OpenShift Topology

OpenShift, by default, assumes a flat network layout for its deployment, although the deployment is highly configurable based on needs. This document lays out the OpenShift nodes on two different network segments:

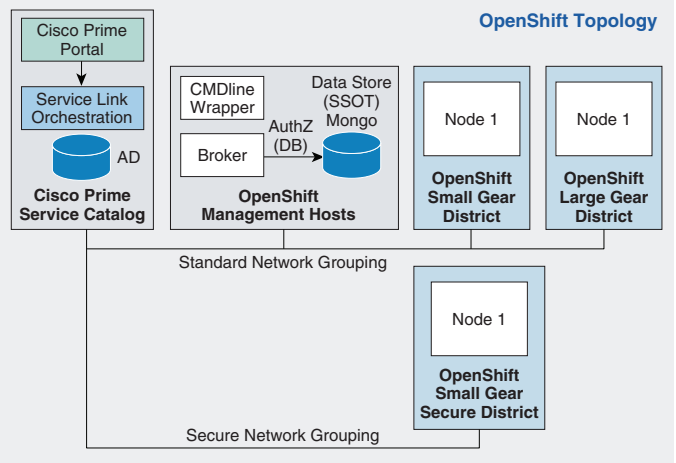
- Standard network segment
- Secure network segment

The OpenShift management nodes and the standard application nodes are placed on one network segment (standard network) and another set of application nodes are placed on a different network segment (secure network).

The secure network segment is protected using ACLs in security groups to permit only http and https traffic from the general network. In the opposite flow, activemq/storm traffic originating from the secure network is blocked from the activemq process on the broker host on the standard network.

The intention is to show that a single OpenShift installation can be deployed across Layer 3 network boundaries and still be managed as a single cohesive OpenShift installation (Figure 3-2).

*Figure 3-2 OpenShift Deployment Logical Network Topology Component Layout on IaaS*




# Design Principles

The following design principles are provided for consideration.

## Tenancy Model

Implementation of tenancy (and, by extension, multi-tenancy) can be different at different levels of the Cloud Stack even though the core notion of a tenant remains the same. The tenant is a user/set of users/organizations that owns resources, is accountable for the cost of the usage of such resources, and has full control over how those resources are used.

Examples of how tenancy may be broken down at the various layers of the cloud stack are:

- **IaaS Tenancy**—Virtual Data Center, VMs, Subnets, VLANs, Storage pools, total bandwidth.
- **PaaS Tenancy**—Secure OS Containers, DNS namespaces/ domains, application URLs.
- **SaaS Tenancy**—Application-level authorization, groups of users, roles, data encryption.

This solution does not directly address tenancy models and tenancy management. The solution allows for a flexible tenancy model, where a tenant can be defined by the provider with a combination of IaaS-level resources, groups of users, organizational boundaries, and PaaS-level resources.

For the purposes of this solution, an end user is the same as a tenant.

This requires that certain administration tasks for Day Zero be performed as part of the implementation of the solution, either manually or by running installation and configuration scripts and actions.

## Identity and Entitlement

This solution does not directly address identity, access management, or entitlement for different components in the stack. The assumption is that providers have an existing identity management solution with which the components need to integrate. For completeness of the flows, Microsoft Active Directory (AD) was chosen as the single identity store and authentication provider for all components in the stack.

- Each component integrates with the same AD instance(s) to provide a user store and an lDAP- and ADSI-based authentication mechanism.
- CIAC PSC (PSC) uses AD integration for authentication purposes.
- OpenStack Keystone uses AD/LDAP integration for authentication purposes.
- OpenShift Broker uses AD/LDAP integration for authentication purposes.
- System accounts is used wherever possible to identify a component to another component wherever possible; e.g. PSC uses a system account to talk to OpenShift.
- Authorization is performed at each component.

This solution does not take into account the user provisioning and management work flows for AD. For the purposes of verification, users will be created out of band in the Active Directory installation.

# Request Work Flows

The following design principles for implementing user request work flows are provided for consideration.

## IaaS Work Flow

Full lifecycle management of IaaS resources is out of scope of this document. For the purposes of the overall solution proposed in this document, a user can order a virtual machine (VM) and use the existing integration that CIAC has with the hypervisor managers to manage the VM.

## OpenShift PaaS Work Flow

As describe in the Tenancy Model section above, the tenancy for the purposes of this solution is based on the notion of an end user.

To facilitate the creation of resources on the OpenShift environment, a system user (admin user) will be created in OpenShift that will create /delete resources on behalf of the end-user.

1. System user owns all namespaces createdxaaslabs.cisco.com (e.g):
   a. System user is allowed large number of gears; e.g. 1000
2. System user creates app in namespace on behalf of end user:
   a. System user creates / checks end-user in OpenShift.
   b. System user creates namespace for end-user.
   c. System user adds end-user to the app as an edit user.

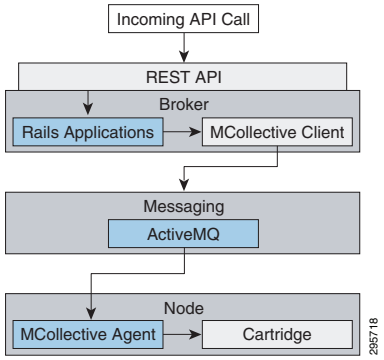
# OpenShift Internal Architecture

This section highlights the internal design of OpenShift. The design is internal to OpenShift and this document does not modify the inherent design in the OpenShift product.

# Communication Mechanisms

Communication from external clients, such as the client tools or the Management Console, occurs through the REST API that is hosted by the broker. The broker then communicates to the nodes using the messaging service component. MCollective queries a set of nodes and communicates securely with individual nodes. Figure 3-3 provides a high-level description of this communication.

***Figure 3-3 OpenShift Enterprise Communication Mechanisms***



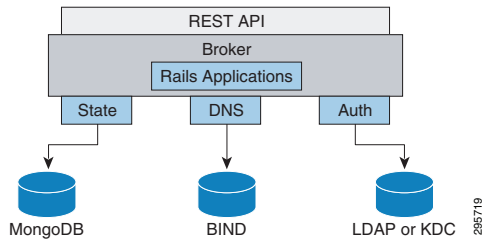

# State Management

The broker is responsible for managing persistent data for OpenShift Enterprise using three distinct interfaces that represent the complete state. Three interfaces are used because each data store is pluggable and each type of data is usually managed by a separate system. Table 3-3 describes each section of application data, and Figure 3-4 depicts enterprise state management.

***Table 3-3 Sections of Application Data***

| Section | Description |
|---|---|
| State | This is the general application state where the data is stored using MongoDB by default. |
| DNS | This is the dynamic DNS state where BIND handles the data by default. |
| Auth | This is the user state for authentication and authorization. This state is stored using any authentication mechanism supported by Apache, such as mod_auth_ldap and mod_auth_kerb. |

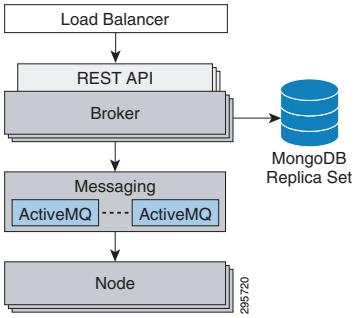
***Figure 3-4 OpenShift Enterprise State Management***



## Redundancy

OpenShift Enterprise incorporates redundancy where each architectural component can be configured redundantly. The broker applications themselves are stateless and can be set up behind a simple HTTP load balancer. The messaging tier is also stateless and MCollective can be configured to use multiple ActiveMQ endpoints. Multiple MongoDB instances can be combined into a replica set for fault tolerance and high availability (Figure 3-5).

***Figure 3-5 Implementing Redundancy in OpenShift Enterprise***




# Configuration Guidelines

Standard configuration guidelines for PSC are followed during installation. Any customizations and deviations from the standard configurations are outlined in the implementation section.

Standard RHOS OpenStack configuration guidelines are followed during installation. Any customizations and deviations from the standard configurations are outlined in the implementation section.

The following design details are provided for consideration.



## PSC Portal

The following components are to be created to fulfill the portal content requirements for the integration.

### User Workspace Portal Page—PaaS Order Portlet

PSC and CIAC ships with Portlets that allow end-users to provision IaaS resources. An additional Portlet is added to the existing User Workspace portal that allows users to order PaaS resources. The Portlet will invoke a PaaS order form that is described further below.

### Application Inventory Portal Page

PSC and CIAC ships with a portal page that allows end-users to view and manage ordered IaaS resources. An additional Portal page is added to PSC. The portal page has content and Ajax call functions embedded in it to allow it to call OpenShift APIs directly on the OpenShift Broker. The page exposes data retrieved from the APIs as viewable and actionable elements in the portal page.

### PaaS Order Form

A PaaS order form is added to PSC. The form uses table values to store the following elements:

- Form display and value capture attributes.
- Pre-defined OpenShift gear sizes.
- Costs associated with order elements.

The form:

- Computes and sets the OpenShift namespace based on the username of the logged in user.
- Retrieves available OpenShift cartridges via an API call to the OpenShift broker.
- Retrieves available additional cartridges via an API call to the OpenShift broker.
- Upon capture of the user-entered values, submits the request to a service defined in the Service Link component, which is outlined below.

## PSC ServiceLink

Several services are created in Service link to allow it to integrate with the back end OpenShift Broker:

### Creating OpenShift Application Service

A service that can be called by PSC service plan components to pass application meta data to the OpenShift Broker asking it to create applications and wait for a reply back.

### Add User Service

A service that can be called by PSC service plan components to pass user meta data to the OpenShift Broker and asking it to create user and namespaces, and wait for a reply back.

### ServiceLink Openssh Adapter

The Add User service described above requires ServiceLink to be able to SSH into the broker box to call a script.

The Add User service is configured to use the OpenSSH adapter.

# Service Proxies

To avoid cross-site scripting and to enable encapsulation of broker API commands, two service proxies are created that run as simple jsp processes inside the ServiceLink j2ee container. Full stateful agents inside ServiceLink in updates to the document will eventually replace the JSPs.

## Creating OpenShift App Service Proxy

Translates ServiceLink XML to OpenShift-supported JSON and calls the OpenShift Broker service on behalf of ServiceLink.

## Broker Read-Only Services

Traps HTTP REST calls from PSC and ServiceLink, proxies the API calls to OpenShift Broker, and returns the results. This service is used by AJAX UI elements in the PSC portal.

# OpenShift Command Line Wrapper

Certain administrative commands in OpenShift are not exposed via REST API calls. Instead, command line APIs are available on the OpenShift Broker host.

These APIs are available as oo-* shell commands on the broker host and are internally implemented using OpenShift Ruby libraries.

Although OpenShift allows pluggable authentication on the broker, it requires that it keep track of the users and userids that are using the OpenShift system. The creation of users in the system can occur via two methods:

- The first time a user authenticates to the OpenShift Console or the OpenShift Broker directly using user credentials.
- An oo-* command is run on the broker hosts for forcing pre-creation of a user.

The PSC integration in the document allows a user to be logged in to PSC. PSC acts on behalf of the user to manipulate resources on the OpenShift system. As such, it is required that PSC be able to create / pre-create OpenShift users on the OpenShift system.

A Ruby wrapper script is created and installed in the OpenShift broker. This aids in the following:

- Create a user in the OpenShift system.
- Disallow user to create namespaces and applications directly.
- Create a namespace (sub-domain container) in the OpenShift System.
- Assign the user as an edit user for the Namespace.

# OpenShift Prime Entitlement Integration

This solution does not directly address entitlement for different components in the stack. The assumption is that providers have an existing identity management solution with which components integrate. Each component integrates with the same AD instance(s) to provide a user store and an LDAP- and ADSI-based authentication mechanism.

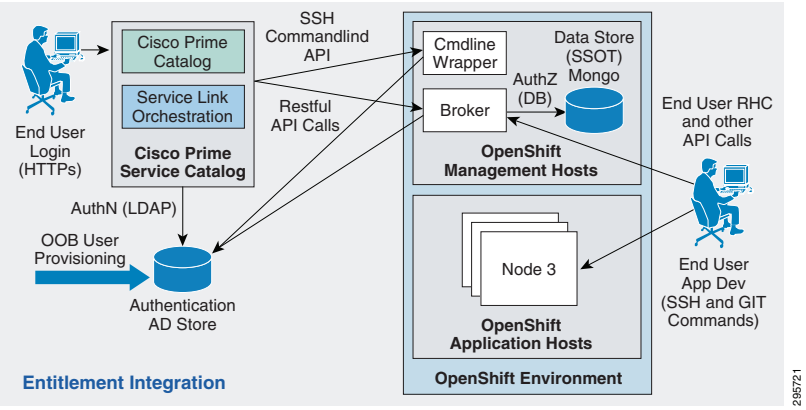
1. CIAC PSC (PSC) uses AD integration for authentication purposes.

2. OpenStack Keystone uses AD/LDAP integration for authentication purposes.
3. OpenShift Broker uses AD/LDAP integration for authentication purposes.
4. System accounts is used wherever possible to identify a component to another component wherever possible; e.g. PSC uses a system account to talk to OpenShift.
5. Authorization is performed at each component.

This solution does not take into account the user provisioning and management work flows for AD. For the purposes of verification, users will be created out of band in the Active Directory installation.

Figure 3-6 shows the AD flows for OpenShift and Cisco Prime Service Catalog in the integrated solution.

***Figure 3-6 OpenShift Prime Entitlement Integration***




# System Level Design Considerations

The following system level design considerations are defined.



## Scalability

PSC scalability is not in the scope of this document. The scalability of PSC is addressed directly in the PSC deployment guides. Further assistance is available from Cisco Advanced Services to guide in the deployment of PSC.

OpenShift scalability is not in the scope of this document. The scalability of OpenShift and design considerations for it is addressed directly in the Red Hat OpenShift Enterprise deployment guides. Further assistance is available from Cisco Advanced Services working in conjunction with Red Hat Professional Services to guide in the deployment of OpenShift.

# Availability

PSC availability is not in scope of this document and is addressed directly in the PSC deployment guides. Further assistance is available from Cisco Advanced Services to guide in the deployment of PSC.

OpenShift availability is not in the scope of this document. Availability of OpenShift and design considerations for it is addressed directly in the Red Hat OpenShift Enterprise deployment guides. Further assistance is available from Cisco Advanced Services working in conjunction with Red Hat Professional Services to guide in the deployment of OpenShift.

# Security

All components are web-based and connectivity is either browser or API-based. As such, all web-based threat vectors apply to this system. Although there are multiple deployment options for the system components, the following general security elements apply.

## Connectivity

All browser-based connectivity to the Cloud Portal and Admin consoles (PSC) is authenticated (LDAP credentials) and authorized (LDAP roles).

All API-based connectivity is authenticated (LDAP credentials) and authorized (LDAP roles).

SSL connectivity is, at a minimum, available as optional for all connections.

## Tenant Security Inside OpenShift

All application code runs in pseudo containers that are controlled by Linux control groups (CGroups). CGroups allow grouping of resources (CPU slots, memory, storage, I/O) such that they can be assigned limits and constraints. The CGroups can be applied to OS level processes such that the processes run within the constraints defined in the croup policies.

All application code is further protected by SElinux policies, which are turned by default. Explicit SElinux policies are applied to each CGroups policy on an application-by-application basis.

Connectivity to the Broker nodes for API interaction is done via an http over SSL (https) session.

Connectivity to the Broker nodes for administration purposes is done via authenticated SSH calls into the Linux host that the Brokers and Application hosts are running on.

Connectivity to the OpenShift Application nodes for end-users /developers to push code, view log etc., are done via SSH using an OpenShift-supplied Linux PAM plugin that allows authenticating and identifying applications based on SSH keys and application names.

## Cross Site Scripting (XSS) and Buffer Overflow

Cisco owns and develops all PSC components. As such, the components follow the Cisco Secure Development Lifecycle (CSDL) Rules and Guidelines for Development, including safe libraries and data input validation.

### Data Protection

All connectivity to the databases for the components is at a minimum authenticated via username and password.

SSL connectivity to the databases is supported as an option.

The components do not support any additional data encryption above that provided by the specific vendor database (MS SQL Server, Oracle).

## Manageability

PSC manageability is not in scope of the document and is addressed directly in the PSC deployment guides. Further assistance is available from Cisco Advanced Services to guide in the deployment of PSC.

OpenShift manageability is not in the scope of the document and is addressed directly in the Red Hat OpenShift Enterprise deployment guides. Further assistance is available from Cisco Advanced Services working in conjunction with Red Hat Professional Services to guide in the deployment of OpenShift.

## Service Assurance and Monitoring

Service assurance and monitoring for the two major components (PSC and OpenShift) are not in the scope of this document and their respective deployment guides address this. Further assistance is available from Cisco Advanced Services working in conjunction with Red Hat Professional Services to guide in the deployment of OpenShift.