Architecture and Design for VMware Cloud Automation Services

VMware Validated Design 5.1
VMware Cloud Automation Service
You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com
Contents

About Architecture and Design for VMware Cloud Automation Services  4

1 Architecture Overview  5
   VMware Cloud Services  5
   VMware Cloud Automation Services  6
      VMware Cloud Assembly  6
      VMware Service Broker  7
      VMware Code Stream  8

2 Detailed Design  10
   Logical Design of Cloud Automation Services  10
   VMware Cloud Services Design  11
      Identity and Access Management  12
      Cloud Proxy Design  13
      Status Notifications Design  26
   VMware Cloud Automation Services Design  27
      Cloud Assembly Design  27
      Service Broker Design  63
About Architecture and Design for VMware Cloud Automation Services

The Architecture and Design for VMware Cloud Automation Services documentation provides information about connecting your on-premises SDDC with VMware Cloud Automation Services for automated, policy-driven provisioning of workloads.

Intended Audience

Architecture and Design for VMware Cloud Automation Services is intended for architects and administrators who want to automate provisioning and lifecycle management of workloads across the Software Defined Data Center and public clouds by using a policy-driven, extensible, as-a-service cloud automation platform.

Prerequisites

You must have VMware Validated Design for Software-Defined Data Center 5.1 deployed in at least a single-region deployment. See the VMware Validated Design documentation page and VMware Validated Design Release Notes for more information about supported product versions.

In this design, both VMware NSX Data Center and VMware NSX-T Data Center are supported for workload domain deployment.

This design uses VMware Cloud Automation Services for provisioning and lifecycle management. vRealize Automation and vRealize Business are not required.

Required VMware Software and Services

In addition to VMware Validated Design for Software-Defined Data Center 5.1 deployment, you must have subscriptions to the following VMware Cloud Services.

- VMware Cloud Assembly
- VMware Service Broker
- VMware Code Stream (Optional.)
Architecture Overview

By implementing VMware Validated Design for VMware Cloud Automation Services you can automate provisioning and lifecycle management of workloads across the Software Defined Data Center and public clouds by using a policy-driven, extensible, as-a-service cloud automation platform.

- **VMware Cloud Services**
  VMware Validated Design for VMware Cloud Automation Services enables IT organizations that have deployed VMware Validated Design for Software-Defined Data Center 5.1 to provision workloads by using a policy-driven, extensible, as-a-service cloud automation platform.

- **VMware Cloud Automation Services**
  VMware Cloud Automation Services streamlines multi-cloud infrastructure and application delivery, enhances visibility and cross-functional collaboration, and provides continuous delivery and release automation.

**VMware Cloud Services**

VMware Validated Design for VMware Cloud Automation Services enables IT organizations that have deployed VMware Validated Design for Software-Defined Data Center 5.1 to provision workloads by using a policy-driven, extensible, as-a-service cloud automation platform.

Organizations looking to adopt cloud services typically pursue two strategies - hybrid cloud or public cloud. A public cloud strategy is predominantly centered around the evolution of application development to orchestrated container services and operations. This generally leads to use of multiple public clouds within an organization, resulting in a multi-cloud environment, including the hybrid cloud subset.

Hybrid cloud adoption is primarily driven by IT Operations teams that have established proven models for infrastructure within the Software-Defined Data Center and are extending the models to the public cloud.

Public cloud and hybrid cloud have a range of consumers of cloud services - from developers and business owners, to dedicated cloud operations teams. These groups are looking to take advantage of the agility and innovation that cloud native development supports.

VMware Cloud Services is focused on meeting the needs of each audience by providing services to deliver consistent operations across clouds. VMware Cloud Services provides visibility, operations, automation, security, and governance, to deliver technology service agility, without the complexity and risk, typically associated with leveraging heterogeneous cloud environments.

You sign up for VMware Cloud Services, access your entitled cloud services and manage all organizations, users, groups, and payment methods through a single intuitive portal.
Table 1-1. VMware Cloud Services Components Overview

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services</td>
<td>Services is used to view, access, and manage active VMware Cloud Services subscriptions or request access to additional services.</td>
</tr>
<tr>
<td>Identity &amp; Access Management</td>
<td>Identity &amp; Access Management is where user and group permissions are configured in VMware Cloud Services. You can invite new users, remove existing users, and assign organization and service roles to both users and/or groups.</td>
</tr>
<tr>
<td>Billing &amp; Subscriptions</td>
<td>Billing &amp; Subscriptions is used for billing and payment information. Current costs, statements, card payments, and credits are shown in this area.</td>
</tr>
<tr>
<td>Support Center</td>
<td>Support Center is used to help find documentation or log VMware support requests. These requests are linked to a VMware ID which is be required to use this feature.</td>
</tr>
<tr>
<td>Cloud Proxy</td>
<td>Cloud Proxy for VMware Cloud Services connects the VMware Cloud Services such as VMware Cloud Automation Services - which consists of Cloud Assembly, Service Broker, and Code Stream - and Log Intelligence, to on-premises and VMware Cloud on AWS Software-Defined Data Center instances.</td>
</tr>
</tbody>
</table>

VMware Cloud Automation Services

VMware Cloud Automation Services streamlines multi-cloud infrastructure and application delivery, enhances visibility and cross-functional collaboration, and provides continuous delivery and release automation.

VMware Cloud Automation Services is a bundled offering of Cloud Assembly, Service Broker, and Code Stream, and is available on-demand based on hourly consumption through VMware Cloud Services. Access to Cloud Automation Services is available globally.

- **VMware Cloud Assembly**
  VMware Cloud Assembly provides the ability to automate the delivery of cloud services across multiple clouds.

- **VMware Service Broker**
  VMware Service Broker aggregates native content from multiple clouds and platforms into a single catalog with role-based policies.

- **VMware Code Stream**
  VMware Code Stream accelerates software delivery and streamlines troubleshooting through release pipelines automation and analytics.

**VMware Cloud Assembly**

VMware Cloud Assembly provides the ability to automate the delivery of cloud services across multiple clouds.
VMware Cloud Assembly is a cloud service that enables you to create and deploy virtual machines, applications, and services to your cloud infrastructure. With Cloud Assembly you can:

- Curate Content - define and configure what content is available to projects.
- Design and Deploy - iteratively build and deploy blueprints for infrastructure and applications.

As a cloud administrator, you configure the cloud vendor infrastructure to support cloud-agnostic blueprint development and deployment for multiple clouds. You set up projects, add users, and enable access to resources in cloud accounts or regions. You import or develop blueprints, or delegate development to the project administrators and members.

As a project member, you use Cloud Assembly to develop and deploy blueprints with declarative and iterative approach. You deploy blueprints to the cloud accounts/regions, which are configured by the cloud administrator as part of the project, and manage their resources throughout the development life cycle. You integrate resource delivery into continuous integration and continuous delivery (CI/CD) processes.

Some of the Cloud Assembly capabilities include the following:

- **Cloud Agnostic Blueprints**
  Build blueprints on a set of building blocks that can be deployed on any supported cloud.

- **Infrastructure as Code**
  Define blueprints in YAML to facilitate deployment, configuration, definition repeatability, version control, and collaboration.

- **Policy Based Resource Delivery**
  Establish governance that includes access definition to resources and which clouds can support specific activities or teams.

- **Governance**
  Ensure optimal use of resources with tag-based governance and policies at the project level.

- **Development Model**
  Create environment definitions by writing code, drawing, or both in an intuitive canvas.

- **Declarative Approach**
  Modify and iterate to reach a desired endstate definition.

**VMware Service Broker**

VMware Service Broker aggregates native content from multiple clouds and platforms into a single catalog with role-based policies.

VMware Service Broker is a cloud service that aggregates content in native formats from multiple clouds and platforms into a simplified and efficient catalog. You use the catalog to manage the available catalog items, as well as how and where they are deployed in cloud accounts/regions.
As a cloud administrator, Service Broker is the portal that you provide to your users, such as, operations and development teams. You import content such as Cloud Assembly blueprints, AWS CloudFormation templates, and extensibility actions, and configure governance in the form of projects to control accessibility of resources and deployment location.

As a user, you request and monitor the provisioning process. After deployment, you manage the deployed catalog items throughout the deployment lifecycle.

Some of the Service Broker capabilities include the following:

- **Self Service**
  A portal for users that provides access to both infrastructure and application level services.

- **Governance**
  Policy based management that provide access control over resources and deployment locations.

- **Definition Abstraction**
  Support for integrating end user services, that have been designed by using a range of definition tools, such as VMware Cloud Assembly.

- **Multi-cloud Support**
  Unified delivery of predefined services, that run on different cloud environments, including VMware based private and hybrid clouds, as well as native public clouds.

**VMware Code Stream**

VMware Code Stream accelerates software delivery and streamlines troubleshooting through release pipelines automation and analytics.

VMware Code Stream is a cloud service that provides continuous integration and continuous delivery (CI/CD) that enables you to rapidly and reliably deliver software. Code Stream simplifies the ability to build, test, and deploy your applications, and increases productivity as you release source code from the development repository, through testing, to production. Code Stream integrates your release with custom and common developer tools and objects, such as Cloud Assembly blueprints.

You create a pipeline that runs actions to build, deploy, test, and release your software. Code Stream runs your software through each stage of the pipeline until it is ready to be released. You integrate the pipeline with one or more DevOps tools, which provide data for the pipeline to run. For example, when a developer checks in code to a Git repository, Code Stream can trigger the pipeline and automate the build, test, and deployment of an application.

You can integrate Code Stream with other VMware Cloud Service. For example you can publish your Code Stream pipeline to Service Broker as a catalog item that can be requested and deployed on cloud accounts/regions or deploy a Cloud Assembly blueprint and use the parameter values that the blueprint exposes.

Some of the Code Stream capabilities include the following:

- **Artifacts**
Ensure correct versions are used across all stages of the development lifecycle.

- **Dashboards**
  Facilitate collaboration by providing dashboards and reports for release pipelines KPIs.

- **Integration**
  Use existing investments in build, test, provisioning, deployment, and monitoring tools.

- **Multi-cloud Support**
  Run pipelines on different cloud environments including VMware based private and hybrid clouds, as well as native public clouds.

**Important** VMware Code Stream is out of scope for this VMware Validated Design.
Detailed Design

VMware Validated Design for VMware Cloud Automation Services considers both virtual infrastructure and services design. It includes numbered design decisions, and the justification and implications of each decision.

- **Logical Design of Cloud Automation Services**
  VMware Cloud Automation services communicates with the vCenter Servers and the NSX Managers in all regions of the SDDC through the Cloud Proxy provision and manage the lifecycle of workloads.

- **VMware Cloud Services Design**
  The VMware Cloud Services design includes the physical and software components that make up the service. The design provides guidance on the main elements of the service design, such as, identity and access management, networking, integration, and availability.

- **VMware Cloud Automation Services Design**
  The VMware Cloud Automation Services design includes Cloud Assembly and Service Broker design. The design provides guidance on configuration, organization, and consumption of the services to enable declarative blueprint orchestration in a multi-cloud environment.

**Logical Design of Cloud Automation Services**

VMware Cloud Automation services communicates with the vCenter Servers and the NSX Managers in all regions of the SDDC through the Cloud Proxy provision and manage the lifecycle of workloads.
Figure 2-1. Logical Design of the SDDC with Cloud Automation Services

VMware Cloud Services Design

The VMware Cloud Services design includes the physical and software components that make up the service. The design provides guidance on the main elements of the service design, such as, identity and access management, networking, integration, and availability.

- **Identity and Access Management**
  
  You manage access to your VMware Cloud Services organization by assigning users and groups to organization and service roles.
Cloud Proxy Design

The Cloud Proxy for VMware Cloud Services connects the VMware Cloud Services Platform (CSP) services, such as Cloud Automation Services and Log Intelligence, to on-premises and VMware Cloud on AWS Software-Defined Data Center instances.

Status Notifications Design

You can view and subscribe to notifications for services available in VMware Cloud Services by using a system powered by Atlassian Statuspage.

Identity and Access Management

You manage access to your VMware Cloud Services organization by assigning users and groups to organization and service roles.

As an organization owner, you add users to your organization and provide access to the VMware Cloud services associated with it.

You can assign users two types of role-based access:

Organization Role

A role within the organization - owner or member.

Cloud Service Role

A role within the cloud service to which you are inviting the user. Each VMware Cloud service has its own specific roles.

For more information, on organization roles and their permissions, refer to Organization Roles and Permissions in the VMware Cloud Services documentation.

Assigning roles to groups is more efficient than assigning the same permissions to individual users. As an organization owner, you determine the members that make up your groups and what roles they are assigned.

There are two types of groups available in VMware Cloud Services:

Custom Groups

You create custom groups by entering a name and a description, adding members, and then assigning roles for the organization and its resources. For example, you can create a custom group and give it an organization member role to your organization and a support role, and read-only access to specific services in the organization.

Enterprise Groups

Enterprise groups are groups that are synced from your corporate Active Directory after federating with VMware Cloud Services and available for use in your organization.

As an organization owner, you can create groups, manage, and edit groups. For custom groups, you can edit the name and description, add or remove members, and change the role assignment of the group. For enterprise groups, you can only change the role assignment of the group.

In this design, enterprise groups are used for organization and service roles.
As a consumer of VMware Cloud Services, you can set up federation with your corporate domain. Federating your corporate domain allows you to use your organization's single sign-on and identity source to sign in to VMware Cloud Services. You can also set up multi-factor authentication as part of federation access policy settings.

Federated identity management allows you to control authentication to your organization and its services by assigning organization and service roles to your enterprise's Active Directory security groups.

**Note** VMware requires users of VMware Cloud Services, who work with VMware for the purposes of billing and support, to have a VMware ID. Enterprise users, who need access to billing and support features, must link their corporate account with their VMware ID.

Federation is achieved by deploying and configuring the VMware Identity Manager connector with the user attributes and group synchronization from your corporate identity store. Configure your corporate identity provider instance by using the VMware Identity Manager service and register your corporate domain identity source. After you set up a federated identity, use your corporate credentials to sign in to VMware Cloud Services.

**Note** VMware Cloud Services federation is established with the VMware Cloud Services customer success team. Contact your customer success representative.

| Table 2-1. Design Decision for VMware Cloud Services Identity and Access Management |
|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Decision ID                     | Design Decision                 | Design Justification            | Design Implication              |
| SDDC-VMCS-IDM-001               | Enable VMware Cloud Services federation with your corporate identity source with the Customer Success team or representative. | Allows authentication, including multi-factor, to VMware Cloud Services and subscriptions using your corporate identity source. Allows the assignment of organization and cloud services roles to enterprise users and groups defined in your corporate identity source. | You must deploy and configure the VMware Identity Manager connector to establish federation and single sign-on to VMware Cloud Services using your corporate identity source. For subscription billing and support features in VMware Cloud Services, an enterprise user identity must be linked to a VMware ID. |
| SDDC-VMCS-IDM-002               | Assign organization and service roles to designated enterprise groups, synchronized from your corporate identity source through federation. | Allows access management and administration to VMware Cloud Services by using corporate security group membership. | You must define and manage security groups, group membership and, security controls in your corporate identity source for VMware Cloud Services. |

**Cloud Proxy Design**

The Cloud Proxy for VMware Cloud Services connects the VMware Cloud Services Platform (CSP) services, such as Cloud Automation Services and Log Intelligence, to on-premises and VMware Cloud on AWS Software-Defined Data Center instances.
A Cloud Proxy connects the on-premises and VMware Cloud on AWS SDDC instances to target VMware Cloud Services subscriptions. The Cloud Proxy is responsible for the following:

- Identifying items in VMware Cloud Services that must be executed against any on-premises endpoints.
- Capturing information about your environment that is pertinent to the VMware Cloud Services.
- Polling Cloud Assembly to check if there are any provisioning (or extensibility) requests.
- Executing those requests.

**Logical Design of the Cloud Proxy**

VMware Cloud Proxy connects VMware Cloud Services to on-premises networks and provides multiple capabilities, including a data pipeline service and agent lifecycle management. Cloud Proxy is supplied with your VMware Cloud service.

**Physical Design of the Cloud Proxy**

To enable integration of the Software-Defined Data Center with some VMware Cloud Services offerings, you deploy the Cloud Proxy virtual appliance and configure settings and registration.

**Networking Design for the Cloud Proxy**

For isolation and co-location with on-premises cloud accounts and service integrations, you deploy the Cloud Proxy virtual appliance in the region-specific management VXLAN. The networking design supports administrative access to the Cloud Proxy instances and outbound access for each Cloud Proxy instance to VMware Cloud Services endpoints.

**Availability Design for the Cloud Proxy**

You protect the Cloud Proxy virtual appliance to ensure availability of VMware Cloud Service integrations.

**Information Security and Access Control for the Cloud Proxy**

You protect Cloud Proxy appliance deployments by configuring secure communication and authentication with VMware Cloud Services and components in the Software-Defined Data Center. You use dedicated service accounts for communication between the Cloud Proxy virtual appliance instances and cloud account endpoints in the management domain.

**Logical Design of the Cloud Proxy**

VMware Cloud Proxy connects VMware Cloud Services to on-premises networks and provides multiple capabilities, including a data pipeline service and agent lifecycle management. Cloud Proxy is supplied with your VMware Cloud service.

Cloud Proxy connects to VMware Cloud Services, such as Cloud Assembly and Code Stream, to on-premises data centers. Cloud Proxy runs Docker containers for specific agents for the various VMware Cloud Services and supports data communications. It enables lifecycle management, as well as data delivery, and communications functionality. Cloud Proxy contains service agents that gather data and use a data pipeline service to provide high throughput and low latency data delivery. The data pipeline service also controls channel communication between Cloud Proxy and VMware Cloud Services.
Cloud Proxy is required when:

- You create or use vCenter Server, NSX Data Center, or VMware Cloud on AWS Cloud Accounts in Cloud Assembly.
- You are using Cloud Assembly blueprint components that communicate with on-premises systems, such as on-premise vRealize Orchestrator, Puppet, or Ansible integrations.
- You are integrating with vRealize Orchestrator to run extensibility workflows, based on event broker subscriptions or as a content source in Service Broker.
- You are integrating Code Stream with on-premises development tools, such as Jenkins.
- You provision to an on-premises VMware Enterprise PKS stack.
- You forward logs from VMware Cloud on AWS to on-premises by using Log Intelligence.

Figure 2-2. Logical Design of the Cloud Proxy a Multi-Region Deployment

The Cloud Proxy appliance runs several agents as Docker containers within PhotonOS.
Figure 2-3. Logical Architecture of the Cloud Proxy Appliance
<table>
<thead>
<tr>
<th>Agent</th>
<th>Used By</th>
<th>Description</th>
<th>Collects</th>
<th>Stores</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloudassembly-cmx-agent</td>
<td>Cloud Assembly</td>
<td>Responsible for the communication with Kubernetes clusters, such as, VMware Enterprise PKS.</td>
<td>Information about Kubernetes resources.</td>
<td>Logs.</td>
</tr>
<tr>
<td>cloudassembly-blueprint-agent</td>
<td>Cloud Assembly</td>
<td>Enables Cloud Assembly to integrate with on-premise endpoints, such as, Ansible and Puppet. Uses the data pipeline service to communicate with these endpoints.</td>
<td>Automatic collection from on-premises. Logs may be shared or uploaded with manual approval only.</td>
<td>Temporarily stores request inputs and outputs.</td>
</tr>
<tr>
<td>cloudassembly-sddc-agent</td>
<td>Cloud Assembly</td>
<td>Passes information between VMware Cloud Services and on-premises Cloud Accounts.</td>
<td>Inventory information, such as, host, machines, storage, networks, and templates from vCenter Server and NSX Data Center.</td>
<td>Endpoint certificate thumbprint and self-signed certificates.</td>
</tr>
<tr>
<td>codestream-lemans-agent</td>
<td>Code Stream</td>
<td>Passes information between VMware Cloud Services and on-premises Cloud Accounts.</td>
<td>Helps to proxy commands from code stream SaaS to on-premises endpoints.</td>
<td>Log files and proxy properties.</td>
</tr>
<tr>
<td>log-forwarder</td>
<td>Log Intelligence</td>
<td>Forwards vCenter Server and NSX Data Center logs to Log Intelligence.</td>
<td>Logs from vCenter Server and NSX Data Center.</td>
<td>No data.</td>
</tr>
</tbody>
</table>
**Note**  The is the current list of Cloud Proxy agents. Additional agents may be added for VMware Cloud Services as required.

To display status of the containers for these agents, the following command can be run on the Cloud Proxy appliance.

```
bash /data-collector-status --containers
```

**Physical Design of the Cloud Proxy**

To enable integration of the Software-Defined Data Center with some VMware Cloud Services offerings, you deploy the Cloud Proxy virtual appliance and configure settings and registration.

**Cloud Proxy Appliance Deployment**

The Cloud Proxy appliance is distributed as an OVA file through specific services in the VMware Cloud Services console, such as Cloud Assembly. You deploy a Cloud Proxy appliance from the OVA file or through the URL provided in the management domain cluster. During the deployment, you provide the One-Time Key (OTK) for the Cloud Proxy to register with your VMware Cloud Services organization. The OTK expires within 24 hours and cannot be reused. In a multi-region or multi availability zone SDDC, you deploy a Cloud Proxy instance in each region. Each Cloud Proxy deployment requires a unique OTK to register with your VMware Cloud Services organization.

**Note**  Cloud Proxy is also available as an Amazon Machine Image (AMI) for deployment in Amazon Web Services when used only with Log Intelligence. This deployment method is out of scope for this design.

**Compute and Storage Resource Sizing**

The Cloud Proxy appliance has the following resource requirements. Provide memory and CPUs for the operation of the appliance.

**Table 2-3. Resource Footprint of the Cloud Proxy Appliance**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>4 vCPUs</td>
</tr>
<tr>
<td>Memory</td>
<td>12 GB</td>
</tr>
<tr>
<td>Disk size</td>
<td>1.4 GB Thin Provisioned</td>
</tr>
<tr>
<td></td>
<td>80 GB Thick Provisioned</td>
</tr>
</tbody>
</table>

**Lifecycle Management**

Upgrades for VMware Cloud Proxy virtual appliance and Cloud Proxy agents are provided through the VMware Cloud Services subscriptions integrated with the proxy.

The upgrade process resolves any updates to the virtual appliance and its agents. For the Cloud Proxy agent upgrades, information such as version tag, artifact name, and new Docker run commands, are sent as part of the command from the Cloud Proxy service.
Networking Design for the Cloud Proxy

For isolation and co-location with on-premises cloud accounts and service integrations, you deploy the Cloud Proxy virtual appliance in the region-specific management VXLAN. The networking design supports administrative access to the Cloud Proxy instances and outbound access for each Cloud Proxy instance to VMware Cloud Services endpoints.

You deploy the Cloud Proxy appliance in the region-specific application virtual networks Mgmt-RegionA01-VXLAN and Mgmt-RegionB01-VXLAN.

Application Virtual Network Design

This networking design has the following features:

- Each Cloud Proxy instance has routed access to the management network through the universal distributed logical router (UDLR) for the SDDC cloud accounts endpoints deployed in the management cluster.
- Routing to the management network and the external network is dynamic, and is based on the Border Gateway Protocol (BGP).

For more information about the networking configuration of the application virtual networks for Cloud Proxy, see Application Virtual Network and Virtual Network Design Example.
Figure 2-4. Networking Design for the VMware Cloud Proxy

![Networking Diagram](image_url)

Table 2-4. Design Decisions on the Application Virtual Networks for the VMware Cloud Proxy

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CP-001</td>
<td>Deploy the Cloud Proxy instances on the region-specific application virtual networks.</td>
<td>Ensures localized connection to cloud accounts and integrations per region if a cross-region network outage occurs. Avoids cross-region bandwidth usage for cloud accounts and integrations. Provides a consistent deployment model for management applications.</td>
<td>Requires NSX Data Center for vSphere to support this network configuration.</td>
</tr>
</tbody>
</table>
IP Subnets

You can allocate the following example subnets to the Cloud Proxy deployment.

**Table 2-5. IP Subnets in the Application Virtual Networks for the Cloud Proxy Instances**

<table>
<thead>
<tr>
<th>Region</th>
<th>IP Subnet</th>
<th>VXLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region A</td>
<td>192.168.31.0/24</td>
<td>Mgmt-RegionA01-VXLAN</td>
</tr>
<tr>
<td>Region B</td>
<td>192.168.32.0/24</td>
<td>Mgmt-RegionB01-VXLAN</td>
</tr>
</tbody>
</table>

DNS Records

The name resolution for each Cloud Proxy appliance uses a region-specific suffix according to the region of deployment - sfo01.rainpole.local or lax01.rainpole.local.

**Table 2-6. FQDNs for the Cloud Proxy Instances**

<table>
<thead>
<tr>
<th>Region</th>
<th>FQDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region A</td>
<td>sfo01csp01.sfo01.rainpole.local</td>
</tr>
<tr>
<td>Region B</td>
<td>lax01csp01.lax01.rainpole.local</td>
</tr>
</tbody>
</table>

**Table 2-7. Design Decision on the DNS Records for the VMware Cloud Proxy Instances**

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CP-002</td>
<td>Configure forward and reverse DNS records for each Cloud Proxy appliance.</td>
<td>Each Cloud Proxy instance is accessible by using a fully qualified domain name instead of by using IP addresses only.</td>
<td>You must provide forward and reverse DNS records for each Cloud Proxy appliance.</td>
</tr>
</tbody>
</table>

External Connectivity

A Cloud Proxy instance uses network connections to securely pass information between VMware Cloud Services cloud accounts and integrations.

A Cloud Proxy instance requires external network connectivity to white-listed VMware Cloud Service URLs. You can use an HTTP proxy server for outbound connectivity but access to on-premises cloud account and integration endpoints must be direct. An HTTP proxy can be configured during OVA deployment or by using the `configure-network-proxy.sh` script in the root file system.

```
./configure-network-proxy.sh proxy.rainpole.local 3128
```

The Cloud Proxy connects to VMware Cloud Services, through the API gateway or through a data pipeline service and then to the VMware Cloud Service. All connections require TLS v1.2 over TCP port 443.

To display whitelisted URLs, run the command on the Cloud Proxy appliance.

```
bash /data-collector-status --requiredUrls
```
To test if these URLs are reachable, run the command on the Cloud Proxy appliance.

```
bash /data-collector-status --traceroute
```

Table 2-8. Design Decision on Network Connectivity for the VMware Cloud Proxy Instances

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CP-003</td>
<td>Provide direct or proxy HTTPS access to the white-listed URLs for the Cloud Proxy virtual appliances.</td>
<td>The Cloud Proxy virtual appliances requires outbound network connectivity to the VMware Cloud Services for data exchange.</td>
<td>You must provide the Cloud Proxy instances with direct or proxy HTTPS access to the VMware Cloud Services.</td>
</tr>
</tbody>
</table>

**Availability Design for the Cloud Proxy**

You protect the Cloud Proxy virtual appliance to ensure availability of VMware Cloud Service integrations. The Cloud Proxy virtual appliance for VMware Cloud Services does not offer native high-availability, but vSphere High-Availability (HA) may be used to restart the Cloud Proxy virtual appliance and services with minimal downtime.

High-availability for agents running on the Cloud Proxy appliance is also not provided. The appliance provides self-healing functionality that is facilitated using native Docker restart and monitoring capabilities for the container services.

Table 2-9. Design Decisions on Availability of the VMware Cloud Proxy

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CP-004</td>
<td>Protect all Cloud Proxy appliances by using vSphere HA.</td>
<td>Supports the availability objectives for Cloud Proxy virtual appliances without a required manual intervention during a failure event.</td>
<td>The Cloud Proxy becomes unavailable during a vSphere HA failover and may impact VMware Cloud Service integrations, such as Cloud Automation Services and Log Intelligence.</td>
</tr>
</tbody>
</table>

**Information Security and Access Control for the Cloud Proxy**

You protect Cloud Proxy appliance deployments by configuring secure communication and authentication with VMware Cloud Services and components in the Software-Defined Data Center. You use dedicated service accounts for communication between the Cloud Proxy virtual appliance instances and cloud account endpoints in the management domain.

**Security**

A one-time key (OTK) is generated as part of the initial registration workflow for the Cloud Proxy. The OTK is stored locally in the Cloud Proxy virtual appliance and expires after 24 hours. A self-signed certificate and an endpoint thumbprint are also stored in the Cloud Proxy.
Once registered, the Cloud Proxy connects to VMware Cloud Services through the API gateway or through a data pipeline service. All connections require TLS v1.2 over TCP port 443. Cross communication with and between available VMware Cloud Services, such as Cloud Assembly and Service Broker, occurs over HTTPS.

To mitigate man-in-the-middle (MITM) attacks, VMware Cloud Services validates that the network traffic, originating from a registered Cloud Proxy, is using public-key cryptography and Trust-on-First-Use (TOFU) to secure the connection.

Cloud Proxy can run a limited list of allowed commands, such as, Docker, VAMI, and Tail commands, as well as appliance upgrade and management scripts. In addition, data that is generated when the Cloud Proxy virtual appliance is started, such as log files, OTK, and private/public certificate pairs, is kept on the file system of the appliance.

TLS v1.2 or greater encryption is used between the Cloud Proxy appliance and the on-premises cloud account or integration (for example, vCenter Server, NSX-T Manager, or vRealize Orchestrator).

Authentication and Authorization

Cloud Proxy performs local authentication for the default administrator account only. The root account is the primary user account. You use this account to log in to the Cloud Proxy over SSH to check the status of connectivity to white-listed URLs and container services. The root password expires three (3) months after deploying the Cloud Proxy virtual appliance. Upon password expiry, the agent lifecycle management functionality continues to work but the OVA upgrade on reboot cannot occur until you set a new password.

Cloud Proxy service agents connect to on-premises cloud accounts and integrations, such as a workload domain vCenter Server, NSX Data Center Manager, and vRealize Orchestrator. You provide the details, such as the endpoint IP/FQDN/URL, Cloud Proxy instance, user name, password, name, description, and capability tags in VMware Cloud Service. These details are stored in VMware Cloud Services and sent to the data pipeline service agents. The agents then use these to call respective APIs.

You use a custom role in vSphere with permissions to perform Cloud Automation Services operations on cloud accounts in the Software-Defined Data Center. A dedicated service account is assigned a custom role for communication between VMware Cloud Services and the vCenter Server instances in the environment.
Table 2-10. Design Decision on Authentication and Authorization for VMware Cloud Proxy

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
</table>
| SDDC-VMCS-CP-005    | Use local authentication for Cloud Proxy.                                       | Cloud Proxy supports local authentication for SSH only.                               | - The accountability in tracking direct user interactions with the Cloud Proxy virtual appliance is limited.  
- You must control the access to the administrator account for vRealize Suite Lifecycle Manager. |
|                     |                                                                                 |                                                                                      |                                                                                     |
| SDDC-VMCS-CP-006    | Reset the root account password after deployment of the Cloud Proxy virtual appliances. | The root password expires three (3) months after deploying the Cloud Proxy virtual appliance. Upon password expiry, the agent lifecycle management functionality continues to work, but the OVA upgrade on reboot cannot occur until you set a new password. | You must manage the lifecycle of the root account password.                      |
| SDDC-VMCS-CP-007    | Define a custom vCenter Server role (e.g. VMware Cloud Services - Cloud Proxy to vSphere User) for the Cloud Proxy that has the minimum privileges required to support a vCenter Server-based cloud account. | VMware Cloud Services subscriptions interact with vSphere using the minimum set of permissions that are required to support the cloud account. | You must maintain the permissions required by the custom role.                     |
| SDDC-VMCS-CP-008    | Configure a service account (e.g svc-vmcs-vsphere) in vCenter Server for application-to-application communication from VMware Cloud Services to vSphere. | Provides the following access control features:                                      | You must maintain the life cycle and availability of the service account outside of the SDDC stack. |
|                     |                                                                                 | - VMware Cloud Services, such as Cloud Assembly, accesses vSphere with the minimum set of required permissions. |
|                     |                                                                                 | - If there is a compromised account, the accessibility in the destination application remains restricted. |
|                     |                                                                                 | - You can introduce improved accountability in tracking request-response interactions between the VMware Cloud Services and the SDDC cloud account. |
Table 2-10. Design Decision on Authentication and Authorization for VMware Cloud Proxy (continued)

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
</table>
| SDDC-VMCS-CP-009 | Assign global permissions for the Cloud Proxy-to-vSphere service account (e.g. svc-vmcs-vsphere). | VMware Cloud Services subscriptions access designated workload domains with the minimum set of permissions that are required to support vCenter Server-backed cloud accounts in the design.  
See the Credentials for Cloud Assembly Cloud Accounts documentation for the required minimum permissions.                                                                 | You must set the role for the management domain vCenter Server instances to **No Access** to ensure the account can not communicate with the management domain.         |
| SDDC-VMCS-CP-010 | Configure a service account in vCenter Server for application-to-application communication (e.g. svc-vmcs-nsx-v) from VMware Cloud Services to NSX Data Center for vSphere.  
**Note** Only applicable when using NSX Data Center for vSphere based Cloud.  
Provides the following access control features:  
- NSX Data Center accesses to vSphere with the minimum set of permissions, required to perform lifecycle management of virtual networking objects.  
- In the event of a compromised account, the accessibility in the destination application remains restricted.  
- You can introduce improved accountability in tracking request-response interactions between VMware Cloud Services and the SDDC cloud account.  
See the Credentials for Cloud Assembly Cloud Accounts documentation for the required minimum permissions. | You must maintain the service account's life cycle outside of the SDDC stack to ensure its availability.                                                                                                 |
Table 2-10. Design Decision on Authentication and Authorization for VMware Cloud Proxy (continued)

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CP-011</td>
<td>Use global permissions when you create the VMware Cloud Services to NSX Data Center for vSphere service account (e.g. svc-vmcs-nsx-v) in vCenter Server.</td>
<td>■ Simplifies and standardizes the deployment of the service account across all vCenter Server instances in the same vSphere domain.</td>
<td>All vCenter Server instances must be in the same vSphere domain.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Only applicable when using NSX Data Center for vSphere based Cloud Accounts.</td>
<td>■ Provides a consistent authorization layer.</td>
<td></td>
</tr>
<tr>
<td>SDDC-VMCS-CP-012</td>
<td>Use the admin account with NSX-T Data Center Enterprise Administrator role for application-to-application communication from VMware Cloud Services to NSX-T Data Center.</td>
<td>Although NSX-T Data Center supports the use of Workspace ONE Access (f.k.a. VMware Identity Manager) as an authentication source and access control, it is not used in this design. The default admin account is used.</td>
<td>■ The accountability in tracking interactions between VMware Cloud Services and the NSX-T Data Center cloud accounts in the SDDC is limited.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Only applicable when using NSX-T Data Center for vSphere based Cloud Accounts.</td>
<td>See the Credentials for Cloud Assembly Cloud Accounts documentation for the required minimum permissions.</td>
<td>■ You must control access to the admin account for NSX-T Data Center.</td>
</tr>
</tbody>
</table>

Status Notifications Design

You can view and subscribe to notifications for services available in VMware Cloud Services by using a system powered by Atlassian Statuspage.

VMware Cloud Services provides system status notifications to ensure your organization is aware of scheduled maintenance and all known service incidents. It reports the status of each VMware Cloud Services offering as:

- Service is Operational
- Service Maintenance
- Degraded Performance
- Service is Partially Accessible
- Service is Inaccessible

The status notifications also provide updates on all current incidents and history for all maintenance and incidents.
You can subscribe to the status of all or an individual VMware Cloud Services offering to receive notifications when they are affected by scheduled maintenance or an incident, such as in case VMware Cloud Services creates, updates, or resolves an incident.

Subscription methods include:

- Email
- SMS
- RSS
- Webhook (e.g. vRealize Log Insight, vRealize Operations, or Slack App.)

### Table 2-11. Design Decision for Status Notifications

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-STAT-001</td>
<td>Subscribe to status notifications using <a href="https://status.vmware-services.io">https://status.vmware-services.io</a> for the VMware Cloud Services to which your organization is subscribed.</td>
<td>Allows you to receive notifications when VMware Cloud Services have scheduled maintenance or are affected by an incident.</td>
<td>You must select and integrate the delivery mechanisms that work best for your organization.</td>
</tr>
</tbody>
</table>

## VMware Cloud Automation Services Design

The VMware Cloud Automation Services design includes Cloud Assembly and Service Broker design. The design provides guidance on configuration, organization, and consumption of the services to enable declarative blueprint orchestration in a multi-cloud environment.

- **Cloud Assembly Design**
  Cloud Assembly is a multi-cloud, declarative blueprint orchestration and automation solution that enables infrastructure as code for expedited infrastructure consumption and application delivery.

- **Service Broker Design**
  Service Broker aggregates content in native formats from multiple clouds and platforms into a common catalog with the ability to apply governance based on roles and projects.

## Cloud Assembly Design

Cloud Assembly is a multi-cloud, declarative blueprint orchestration and automation solution that enables infrastructure as code for expedited infrastructure consumption and application delivery.

## Logical Design

Cloud Automation Services, of which Cloud Assembly is a core component, enables a usage model that includes interaction between project users, the platform itself, the supporting infrastructure, and the provisioning infrastructure.
**Service Role Design**

You manage access to your Cloud Assembly by assigning enterprise groups to service roles in your organization.

Cloud Assembly has two (2) service roles assigned from the organization Identity and access management. You assign the service roles to designated enterprise groups, synchronized from your corporate identity source through federation.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>Example Enterprise Group</th>
</tr>
</thead>
</table>
| **Cloud Assembly Administrator** | ■ Read and write access to the entire user interface and API resources.  
■ Configure the cloud infrastructure and integrations.  
■ Set up projects and add service users and cloud resources.  
■ Delegate the user management and blueprints to Project Administrators. | rainpole.local\ug-vcms-cas-ca-admins                         |
| **Cloud Assembly User**      | Access to projects as a Project Administrator or Project Member.                                                                                                                                           | rainpole.local\ug-vcms-cas-ca-users                           |

**Cloud Accounts Design**

Cloud Accounts in Cloud Assembly provide a centralized authentication mechanism. You configure the necessary permissions to collect data and deploy blueprints to SDDC regions or data centers.
You create cloud accounts for the projects in which your organization's team members work. Resource information, such as network and security, compute, storage, and tags data, is collected from your cloud accounts.

Some cloud accounts require Cloud Proxy to route information and collect data from Cloud Assembly and your on-premises endpoints. After you create a cloud account, you can associate it with a Cloud Proxy instance. For organizations that are distributed across multiple geographic regions, you use multiple Cloud Proxy instances to spread the workload. You can set up Cloud Proxy instances by geography, by project, or by deployment intent. For multi-region environments with multiple Cloud Proxy instances, you keep the Cloud Proxy instance in the same region as the workload domain which it manages in Cloud Automation Services.

**Important** A cloud account can only be associated with a single Cloud Proxy instance.

As a public cloud, VMware Cloud on AWS does not require the intermediary services provided by Cloud Proxy. You can configure a direct connection between the public cloud service and Cloud Assembly. This allows Cloud Assembly to collect infrastructure data from the public cloud and enables you to deploy blueprints to one or more of the account regions in the public-cloud backed cloud account.
## Table 2-12. Design Decisions for Cloud Accounts

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
</table>
| SDDC-VMCS-CA-001 | Add a cloud account for vCenter Server for the workload domain in each Software-Defined Data Center region. | Allows you to integrate on-premises vSphere backed workload domains with Cloud Automation Services for workload provisioning.                                                                                                                                 | - You must associate the cloud account with the Cloud Proxy instance in each region.  
- You must manage the cloud account credentials along with the lifecycle management of the service account.  
- You must manage capability tags for the cloud account. |
| SDDC-VMCS-CA-002 | Add a Cloud Account for NSX Data Center for the workload domain in each Software-Defined Data Center region. | Allows you to integrate on-premises NSX Data Center backed workload domains with Cloud Automation Services for workload provisioning.                                                                                                                                 | - You must manage cloud account credentials along with the lifecycle management of the service account.  
- You must manage capability tags for the Cloud Account. |
| SDDC-VMCS-CA-003 | Add a cloud account for a VMware Cloud on AWS Software-Defined Data Center instance. | Allows you to integrate the vSphere and NSX Data Center resources provided through VMware Cloud on AWS with Cloud Automation Services for workload provisioning.                                                                                                                                 | - The VMware Cloud on AWS Token used during the cloud account registration expires after six (6) months and must be authenticated before the token expiration.  
- You must manage capability tags for the Cloud Account. |

### Cloud Zones Design

Cloud zones in Cloud Assembly are a method of partitioning the resources that cloud accounts provide to projects.

You create cloud zones to define a set of resources within a cloud account. Each cloud zone is associated with a Cloud Assembly project. Cloud zones can be shared across multiple groups of users and are not limited by a one-to-one relationship.

You can add or remove tags to filter the compute resources that are used in a cloud zone. For example, you may have multiple clusters within a workload domain provided through a vCenter Server-back cloud account. The clusters can serve different purposes due to specific functional or non-functional constraints or classifications.
Cloud zones do not control how CPU and memory are allocated during periods of resource contention. This function is performed by the use of vSphere resource pools. To ensure resource availability, all virtual machines must be deployed on one of the following resource pools on the default cluster (shared edge and compute cluster) for the workload domain.

Table 2-13. Default Cluster (Shared Edge and Compute) Resource Pools in a Workload Domain

<table>
<thead>
<tr>
<th>Resource Pool</th>
<th>Object Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>sfo01-w01rp-sddc-edge</td>
<td>NSX Data Center Edge components. Place user workload in other resource pools.</td>
</tr>
<tr>
<td>sfo01-w01rp-user-edge</td>
<td>Statically or dynamically deployed NSX Data Center components, such as NSX Edge gateways or load balancers, which serve specific customer workloads.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Applies only to NSX Data Center for vSphere-backed workload domain and cloud accounts.</td>
</tr>
<tr>
<td>sfo01-w01rp-user-vm</td>
<td>Statically or dynamically deployed virtual machines which contain organization workloads.</td>
</tr>
</tbody>
</table>

By default, on vCenter Server-backed cloud zones, workloads are placed on random hosts. Optionally, one of the following strategies may be applied to a cloud zone:

**BINPACK**

The most loaded host with enough resources to run the workload is selected for workload placement.

**SPREAD**

Evenly spreads workload across hosts during workload placement.

By default, workloads are placed in the default data center folder. Workload placement can be set by defining a relative path within the data center in the blueprint YAML code.

For example, blueprint expressions can evaluate configuration options to return a value. Due to restrictive permissions, in VMware Cloud on AWS, workloads must be deployed in the Workloads folder (or subfolder), but for an on-premises cloud account you may want to deploy the workload to an alternate folder. You can accomplish this with a blueprint expression that evaluates endstate values. If a user selects the `targetCloud` value of `vmc`, the `folderName` property value is set to `Workload`. The else option can apply an alternative value to `folderName` - a blank value (“””) or another input value `$ {input.myFolderName}`. See the sample YAML blueprint below.

```yaml
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: '${input.operatingSystem}'
      flavor: '${input.nodeSize}'
      count: '${input.nodeCount}'
    folderName: '${{input.targetCloud == "vmc" ? "Workload" : ${{input.targetEnvironment}}}'
    networks:
      - network: '${resource.Cloud_NSX_Network_1.id}'
    constraints:
```

VMware, Inc.
You can add capability tags to match blueprint constraints to a cloud zone. Capability tags are optional, because the tags on compute resources (e.g. vSphere Clusters and Resource Pools) are also used as capability tags for a cloud zone. For example: cloud:private, cloud:vmc, region:sfo, region:us-west, env:prod, env:dev, env:dmz.

**Table 2-14. Design Decisions for Cloud Zones**

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CA-004</td>
<td>Create a Cloud Zone for each Software-Defined Data Center region.</td>
<td>Enables region-specific workload provisioning</td>
<td>None.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-005</td>
<td>Add tags to compute resources instead of cloud zone.</td>
<td>Ensures that the scope of the workload tagging is managed with the compute resources.</td>
<td>- You must establish and manage an effective tagging strategy. - You must ensure that constraint tags are included in the blueprint YAML.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-006</td>
<td>Only add tags to the *-user-vm and *-user-edge resource pools in the default shared edge and compute cluster for cloud zones that contain NSX Data Center for vSphere-backed cloud accounts.</td>
<td>Ensures that virtual machines and on-demand network services, using NSX Data Center for vSphere, are deployed to the designated resource pools in the default cluster of the workload domain.</td>
<td>You must ensure that constraint tags are included in the blueprint YAML.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-007</td>
<td>Only add tags to the *-user-vm resource pool in the default shared edge and compute cluster for cloud zones that contain NSX-T Data Center-backed cloud accounts.</td>
<td>Ensures that virtual machines, using NSX-T Data Center, are deployed to the designated resource pool in the default cluster of the workload domain.</td>
<td>You must ensure that constraint tags are included in the blueprint YAML.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-008</td>
<td>Add tags to the vSphere cluster for additional clusters added to the workload domain.</td>
<td>Ensures that, as you add new vSphere clusters to a workload domain, you can enable workload provisioning readiness by adding the appropriate organizational tags.</td>
<td>- You must ensure that you manage the tagging of compute resources as you add clusters to a workload domain. - You must ensure that constraint tags are included in the blueprint YAML.</td>
</tr>
</tbody>
</table>
Table 2-14. Design Decisions for Cloud Zones (continued)

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
</tr>
</thead>
</table>
| SDDC-VMCS-CA-009  | Add a Workloads folder to each workload domains.                                 | - Ensures that blueprints, which do not include the folderName value, are deployed in a default Workloads folder.  
                                                                                  - Ensures that workloads deployed to VMware Cloud on AWS are provisioned to the default folder due to the restrictive security mode.  
                                                                                  - You must add the Workloads default folder to all workload domains.  
                                                                                  - To override the default folder, you must add logic in your blueprint YAML code to deploy blueprints in alternative folders.  
                                                                                  You must add the Workloads default folder to all workload domains.  
                                                                                  To override the default folder, you must add logic in your blueprint YAML code to deploy blueprints in alternative folders.  
                                                                                  Important: The destination folder, where the blueprints are deployed, must already exist and cannot be created by Cloud Assembly without extensibility. |
| SDDC-VMCS-CA-010  | Use the DEFAULT placement policy for each cloud zone.                           | All vSphere clusters have the vSphere Distributed Resource Scheduler enabled to optimize initial and ongoing workload placement within a cluster.  
                                                                                  You must ensure that vSphere Distributed Resource Scheduler is enabled for all workload domain clusters.                                                                                      |

Integration Design

Integrations in Cloud Assembly enables you to support other systems and services, allowing for extended automation and lifecycle management capabilities.

An integration creates a connection between Cloud Assembly and the target service. You configure an integration with a direct connection to a target service or you configure a Cloud Proxy instance to communicate with integrations, just as you do for cloud accounts, based on the integration requirements. Cloud Proxy manages the information exchange between Cloud Assembly and the integration service or application.

Table 2-15. Cloud Assembly Integrations

<table>
<thead>
<tr>
<th>Integration</th>
<th>Description</th>
<th>Cloud Proxy Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>vRealize Orchestrator</td>
<td>You can use vRealize Orchestrator workflows in extensibility subscriptions.</td>
<td>Yes</td>
</tr>
<tr>
<td>Ansible</td>
<td>You can manage deployments for configuration and drift.</td>
<td>Yes</td>
</tr>
<tr>
<td>Puppet</td>
<td>You can manage deployments for configuration and drift.</td>
<td>Yes</td>
</tr>
<tr>
<td>My VMware</td>
<td>You can use VMware Marketplace to consume existing blueprints.</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 2-15. Cloud Assembly Integrations (continued)

<table>
<thead>
<tr>
<th>Integration</th>
<th>Description</th>
<th>Cloud Proxy Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>You can use GitHub (SaaS only) repositories to manage blueprints and action scripts under source control, allowing you to use modern development (e.g. Visual Studio Code) for blueprint action based script authoring.</td>
<td>No</td>
</tr>
<tr>
<td>GitLab</td>
<td>You can use GitLab (SaaS) repositories to manage blueprints and action scripts under source control, allowing you to use modern development (e.g. Visual Studio Code) for blueprint and action based script authoring.</td>
<td>No</td>
</tr>
<tr>
<td>Kubernetes (BETA)</td>
<td>You can use the Kubernetes integration with Enterprise PKS (and native K8s) to create Kubernetes clusters using blueprints.</td>
<td>No</td>
</tr>
<tr>
<td>IPAM (BETA)</td>
<td>You can use an IPAM provider for IP Address Management.</td>
<td>No</td>
</tr>
</tbody>
</table>

With a vRealize Orchestrator integration to Cloud Assembly, you can use vRealize Orchestrator workflows in Extensibility subscriptions. Cloud Assembly supports the integration of multiple vRealize Orchestrator instances. You can use Cloud Assembly capability tags to manage the placement logic of your vRealize Orchestrator integrations. For more information on capability tags, see Tags Design.
<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
</table>
| SDDC-VMCS-CA-011       | Add an integration to the vRealize Orchestrator instance in each region.         | Allows you to use vRealize Orchestrator workflows as part of Extensibility.            | - You must use a service account, created for application-to-application integration for Cloud Automation Services to the SaaS-enabled vRealize Orchestrator, when configuring the integration.  
- You must manage the service account password in the integration configuration when the service account password is updated.  
- You must manage workflow packages for each vRealize Orchestrator instance. |
| SDDC-VMCS-CA-012       | For each vRealize Orchestrator integration, add capability tags for your organization to determine workflow execution during the deployment process. | You use capability tags to manage the placement logic of your vRealize Orchestrator integrations. For example, using a capability tag of region:sfo on each vRealize Orchestrator integration in Region A, would ensure that blueprints with a region:sfo:hard tag are deployed in the specified region. | You must manage capability tagging on each vRealize Orchestrator integration for workflow placement and execution. |
| SDDC-VMCS-CA-013       | Add an integration to GitHub or GitLab and assign a branch to each project.      | You can use Git repositories to manage infrastructure code under source control, allowing you to iterate using modern development (e.g. Visual Studio Code) for blueprint and action-based extensibility script authoring. | - Integration works only with GitHub or GitLab SaaS services.  
- You must establish a GitHub/GitLab repository and access control for blueprint and action-based extensibility source control.  
- Integration is only supported with GitHub and GitLab SaaS Services. |
| SDDC-VMCS-CA-014       | Integrate Cloud Assembly with My VMware.                                        | Provides the ability to download curated blueprints and images from VMware Marketplace to Cloud Assembly. | You must consider using a dedicated My VMware account for the integration with Cloud Assembly. (e.g. svc-cas-marketplace@vmware.com) with the minimum My VMware permissions to access the VMware Marketplace. |
Project Design

Projects in Cloud Assembly control who has access to blueprints and where blueprints may be deployed. A project in Cloud Assembly determines which cloud zones a set of user or group can deploy to, a priority value, the maximum number of virtual machine instances to deploy, and a maximum amount of memory that the deployment can use. A project is typically defined using an organisational structure, such as, a cost-center, or a specific business group or purpose.

Projects enable you to organize and govern what business users can do and to what cloud zones they can deploy blueprints in your cloud infrastructure. You create a project to which you add members and cloud zones so that the project members can deploy their blueprints to the associated cloud zones.

For example, as a Cloud Assembly administrator, you can create a project for a development team and then add the users and groups as project members or project administrators. You then only add the cloud zones that are designated for these workloads.

Projects provide a context that you can assign blueprints to, and moving beyond Cloud Assembly they also bind integration endpoints, such as, Git-based repositories in Code Stream.

When you assign a cloud zone within a project, you can include additional configurations that determines the project behavior.

- **Provisioning Priority**
  
The priority order for a cloud zone use in a project when all other constraints have been met.

- **Instances Limit**
  
The maximum number of instances that can be provisioned in the cloud zone for the project.

Resource tags can be created on cloud resources during provisioning. For vSphere-backed cloud accounts, the tags are associated with the workloads inside the workload domain.

Project constraints can be used as a governance definition to restrict or enable resource consumption. These include network constraints matched with capability tags on network profiles and subnets, storage constraints matched with capability tracks on storage profiles, and extensibility constraints matched with capabilities tags on vRealize Orchestrator integrations. Project constraint are define what resources the deployment request consumes or avoids in the project cloud zones.

At deployment time, a project blueprint requirements are evaluated against the cloud zones associated with the project to determine the best deployment location. Consider the following formats when you configure project constraints.
### Table 2-17. Design Decisions for Projects

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CA-015</td>
<td>For each project, add one or more enterprise groups to the Project Administrator and Project Member roles rather than using individual users.</td>
<td>An enterprise group is a collection of Active Directory users. You can use enterprise groups to specify the role permissions for a collection of users, which makes it easier to manage. For example, you could have a group called <code>sg-project-x-members</code> and give that group the Project Member role. Any user in the group automatically inherits the role assigned to the group and has access to catalog items in Service Broker. If a new user joins your organization and requires project access, you can assign the role by adding the user to the Active Directory security group. Access can be revoked by removing a user from the Active Directory security group. Refer to Cloud Assembly User Roles for more information on Project Administrator and Project Member permissions.</td>
<td>None.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-016</td>
<td>For each project, add one or more Cloud Zones based on the project requirements and allowed cloud resources.</td>
<td>Allows you to provide one or more cloud zones and its resources for project consumption.</td>
<td>None.</td>
</tr>
<tr>
<td>Decision ID</td>
<td>Design Decision</td>
<td>Design Justification</td>
<td>Design Implication</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-017</td>
<td>For each project, set a provisioning priority for each cloud zone based on your deployment prioritization.</td>
<td>Enables you to prioritize one cloud zone over another within a project. The default priority is 0 (highest priority).</td>
<td>You must manage the provisioning priority for each cloud zones in each project.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-018</td>
<td>For each project, set an instance limit for the project cloud zones.</td>
<td>Enables you to set the maximum number of workloads that can be provisioned in this cloud zone for the project. The default is instance limit is 0 (unlimited instances).</td>
<td>If a value greater than 0 (unlimited) is used for instance limit, you must manage the limit for each cloud zone in each project when requirements change.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-019</td>
<td>For each project, specify network, storage, and extensibility constraints that should be applied to all requests in the project.</td>
<td>Ensures proper placement of the workloads in a project and its cloud zones.</td>
<td>If the same constraint or the same constraint category are specified in both the project (e.g. region:sfo) and the blueprint (e.g. region:lax), the constraint specified in the project takes precedence.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-020</td>
<td>For each project, add one or more custom properties. (e.g. project:foo)</td>
<td>Custom properties can be used for provisioning or to capture additional metadata. For example, for reporting or extensibility actions.</td>
<td>If the same custom property is specified in both the project (e.g. project:foo) and the blueprint (e.g. project:bar), the property value specified in the project takes precedence (e.g. project:foo).</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-021</td>
<td>For each project, add a custom naming template to be used for virtual machine names provisioned in the project. (Note: The template provides custom virtual machine name and does not affect the virtual machine’s hostname.)</td>
<td>The template substitutes auto-generated virtual machine names using available properties, such as, resource properties, custom properties, endpoint properties, project properties: and/or a random number with a specified number of digits.</td>
<td>You must ensure that the template generates unique names for this project and between other projects.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-022</td>
<td>Add each project to a GitHub- or GitLab-backed integration and include a default &quot;BLUEPRINT&quot; type repository (e.g. github.com/rainpole/cas-blueprints &gt; master branch)</td>
<td>You can use Git repositories to manage blueprints under source control, allowing you to iterate using modern development (e.g. Visual Studio Code) for blueprint authoring.</td>
<td>You must establish a GitHub/ GitLab repository and access control for blueprint source control.</td>
</tr>
</tbody>
</table>
Mappings Design
Cloud Assembly mappings use natural language terminology to define compute resource sizes and virtual machine images for a specific cloud account or region.

- **Flavor Mapping Design**
  Cloud Assembly flavor mappings allow you to use natural language naming to define a group of deployment size specifications in a specified cloud account/region.

- **Image Mapping Design**
  Cloud Assembly image mappings allow you to use natural language naming to define a group of virtual machine operating system image specifications when used in a cloud account or region.

**Flavor Mapping Design**
Cloud Assembly flavor mappings allow you to use natural language naming to define a group of deployment size specifications in a specified cloud account/region.

A flavor mapping associates a defined cloud region with one or more resource sizing options or instance types. A mapping defines a common term that is mapped to the specific constructs in each of your platform environment. A flavor may equate to a specific number of CPUs and memory allocation, allowing you to scale your resources to the requirements of your target workload.

This is commonly expressed as "t-shirt" sizes. For example, *x-small* can represent 1 CPU and 512 MB memory, while *medium* can represent 4 CPUs and 16 GB memory for a vCenter Server-back cloud account in a named data center. The same resources can be mapped to *t2.nano* and *t2.large* for an Amazon Web Services account in a named region.

You can create a flavor mapping schema across your cloud accounts and regions. For example, a flavor map named *x-large* can contain a similar flavor sizing for some or all available account/regions in your project. When you build a blueprint, you pick an available flavor that fits your needs.
To simplify blueprint creation, you can select a pre-configured option when you add a new cloud account. When you select the pre-configuration option, your organization’s most popular flavor mapping and image mapping for the specified region are selected.

Table 2-18. Design Decision for Flavor Mappings

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CA-023</td>
<td>Create standardized flavor mappings based on a common taxonomy and deployment intent. For example, you can use “t-shirt sizes”.</td>
<td>Provides a simple, natural language naming to define common deployment size specifications.</td>
<td>You must publish and communicate the image mapping standards and updates to blueprint developers.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-024</td>
<td>For each flavor mapping, add each cloud zone.</td>
<td>Provides a simple, natural language naming to define common deployment size specifications when used in a specific cloud account/region.</td>
<td>You must maintain the mapping for any image mapping create or update operation.</td>
</tr>
</tbody>
</table>
**Image Mapping Design**

Cloud Assembly image mappings allow you to use natural language naming to define a group of virtual machine operating system image specifications when used in a cloud account or region.

Image mappings provide a method to define the virtual machine image that an environment can consume. An image mapping associates an image name with a virtual machine template in a region. You can create one or more image names and map to a metadata file that contain pre-defined value sets.

For example, if you want to deploy an Ubuntu virtual machine, you can map *Ubuntu* to a specific virtual machine template or an image from the vSphere Content Library in a vSphere-backed cloud region or account, or to an AMI file in Amazon Web Services, or a specific VHD in Microsoft Azure.

A mapping links a common term to the specific construct in each of your platform environments. In addition, an image may map to a virtual machine image that contains pre-populated cost or region specifications to import into a blueprint.

Cloud accounts backed by vCenter Server and NSX-based environments, such as on-premises and VMware Cloud on AWS Software-Defined Data Center environments, use image mappings to group a set of target deployment conditions together, including virtual machine configuration settings and guest operating system. Cloud accounts, such as Amazon Web Services and Microsoft Azure, also use a similar grouping mechanism to define a set of deployment conditions.

Cloud Assembly can consume images for vCenter Server and NSX-based environments by using traditional virtual machine templates, or Open Virtual Format (OVF) images, packaged in the Open Virtual Appliance (OVA) format in the vSphere Content Library or a source URL, such as a GIT-based repository. In this design, the vSphere Content Library is used to manage and distribute virtual machine images across these Software-Defined Data Center instances.

---

**Figure 2-6. Image Distribution Using the vSphere Content Library**

Image mappings can consume both pre-configured and private images, such as Amazon Machine Images (AMIs) on Amazon Web Services, or Azure Images (VHD/VHDX) on Microsoft Azure. Cloud Assembly automatically performs private image collection every 24 hours or you may manually trigger a synchronization. Synchronization is disabled during image enumeration and for 10 minutes after the last image enumeration has completed.
When you build, deploy, and iterate a blueprint, you pick an available image that best fits your requirements. To simplify blueprint creation, you can select a pre-configuration option when you add a new cloud account. When you select the pre-configuration option, your organization's most popular flavor mapping and image mapping for the specified region are selected.
<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CA-025</td>
<td>Use Open Virtual Format (OVF) images packaged in the Open Virtual Appliance (OVA) format for vCenter Server-backed Cloud Accounts.</td>
<td>Provides a simple, standards-based, and portable file format for all virtual machine images that can be replicated by the vSphere Content Library and natively consumed by Cloud Automation Services.</td>
<td>You must use OVFTool or PowerCLI to export virtual machines to the OVA format. Use of the OVA format precludes the ability to use linked clones or instant clones for blueprint images. If required, native vSphere templates can be used instead of OVA-based images.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-026</td>
<td>Use the vSphere Content Library to synchronize virtual machine images across workload domains and regions for vSphere-backed cloud accounts.</td>
<td>Important The vSphere Content Library can also be used by Software-Defined Data Center instances in VMware Cloud on AWS as a subscriber to the primary region. (e.g. Region A)</td>
<td>The vSphere Content Library is built into vSphere and meets all the requirements to synchronize virtual machine images across multiple regions in a Software-Defined Data Center, including VMware Cloud on AWS. VMware Cloud Automation Services can consume Open Virtual Format-based images from the vSphere Content Library for a cloud account. The vSphere Content Library permissions are connected to global permissions in the permissions hierarchy. To allow Cloud Assembly to sync and use images in the Content Library, the user and role must be applied at the global permissions level. You must provision storage space in each region for images. You must ensure that the service account used for the Cloud Account for vCenter Server has the minimum required permissions to consume images from the vSphere Content Library. You must ensure that the number, size, and structure of the OVA images are kept within the vSphere Content Library configuration maximums. You must ensure HTTPS communications between each workload domain vCenter Server instance. You must manage permissions to the vSphere Content Library. See Using Content Libraries.</td>
</tr>
</tbody>
</table>
Table 2-19. Design Decision for Image Mappings (continued)

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CA-027</td>
<td>Create standardized image mappings based on similar operating system, functional deployment intent, and cloud zone availability.</td>
<td>Allows you to create a simple taxonomy to map images to blueprints.</td>
<td>You must publish and communicate the image mapping standards and updates to blueprint developers.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-028</td>
<td>For each image in an image mapping, add constraint tag, as applicable.</td>
<td>Refines the image selection in an image mapping by matching constraints.</td>
<td>You must manage multiple images in each region based upon the use of constraint tags in your organization.</td>
</tr>
</tbody>
</table>

Profiles Design

Cloud Assembly profiles allow you to define network and storage profiles for a specific cloud account or region.

- **Network Profile Design**
  A Network Profile in Cloud Assembly describes the behavior of the cloud-specific network to be deployed. For example, a network may require specific network services or access.

- **Storage Profile Design**
  A Storage Profile in Cloud Assembly describes cloud-specific storage on which workloads can be deployed based on a set of characteristics.

**Network Profile Design**

A Network Profile in Cloud Assembly describes the behavior of the cloud-specific network to be deployed. For example, a network may require specific network services or access.

Network profile capability tags define a group of networks and workload-specific network characteristics that are available for a cloud account in a particular region. Network profile capability tags are matched to blueprint constraint tags during provisioning. Capability tags are applied to all networks in the network profile.

Network profiles can match the `key:value` constraint pair in a blueprint network configuration. If a network profile is enabled for a public IP, only networks that match `networkType:public` will be matched in the deployment process.
Network policies within a network profile allow you to define network settings, such as private or public network, on-demand networks and security groups generation.

For example, you can create an on-demand network for each deployment using a NSX-backed transport zone or a public cloud network domain. The existing and public and private networks within the network profile are used as the underlying or upstream networks. You can match network type tags with blueprint constraints to place the deployed workload in a specific network.

Table 2-20. Example Network Profiles

<table>
<thead>
<tr>
<th>Account / Region</th>
<th>Cloud Account Type</th>
<th>Network Profile Name</th>
<th>Network Characteristics</th>
<th>Capability Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>vmw-rainpole / sfo01dc01</td>
<td>VMware vSphere</td>
<td>vmw-rainpole-sfo01dc01</td>
<td>Name: subnet-j58d73kk, CIDR: 172.11.10.0/24</td>
<td>cloud:private, region:sfo, network:web, env:prod</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Name: subnet-89d738j5, CIDR: 172.11.20.0/24</td>
<td>cloud:private, region:sfo, network:app, env:prod</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Name: subnet-19jd8yt4, CIDR: 172.11.30.0/24</td>
<td>cloud:private, region:sfo, network:db, env:prod</td>
</tr>
<tr>
<td>vmc-rainpole / SDDC-Datacenter</td>
<td>VMware Cloud on AWS</td>
<td>vmc-rainpole-vmc01sddc01</td>
<td>Name: vmc01-routed-01, CIDR: 192.168.18.0/24</td>
<td>cloud:vmc, region:us-east-1, network:web, env:dev</td>
</tr>
<tr>
<td>aws-rainpole / us-east-1</td>
<td>Amazon Web Services</td>
<td>aws-us-east-1</td>
<td>Name: subnet-5da92b01, Zone: us-east-1a, Network Domain: vpc-7685580c, Subnet: 172.31.31.0/20</td>
<td>cloud:aws, region:us-east, network:api, env:prod</td>
</tr>
</tbody>
</table>
Table 2-20. Example Network Profiles (continued)

<table>
<thead>
<tr>
<th>Account / Region</th>
<th>Cloud Account Type</th>
<th>Network Profile Name</th>
<th>Network Characteristics</th>
<th>Capability Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>azure-rainpole / east us</td>
<td>Microsoft Azure</td>
<td>azure-us-east</td>
<td>Name: default Zone: eastus Network Domain: vnet-eff82c97 Subnet: 10.0.0.0/24</td>
<td>cloud:azure Zone: eastus Network Domain: vnet-eff82c97 Subnet: 10.0.0.0/24</td>
</tr>
</tbody>
</table>

Note  The scope of this design includes VMware enabled clouds. Examples include both private and public cloud network profiles for Cloud Assembly in a multi-cloud context.

Table 2-21. Design Decisions for Network Profiles

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CA-029</td>
<td>For each account / region, add one or more network profiles based on network characteristics available for consumption.</td>
<td>Allows you to add networks with pre-defined characteristics that can be consumed during a deployment process.</td>
<td>You must manage network profiles for each account / region.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-030</td>
<td>For each network in a network profile, add one or more capability tags.</td>
<td>You use capability tags to manage the workload network placement logic during the deployment process.</td>
<td>You must manage capability tagging on each network profile for workflow placement selection during a deployment process</td>
</tr>
</tbody>
</table>

Storage Profile Design

A Storage Profile in Cloud Assembly describes cloud-specific storage on which workloads can be deployed based on a set of characteristics.

Storage profiles are organized under cloud-specific regions. A public cloud account such as Amazon Web Services can have multiple regions (e.g. AWS us-east-1 and us-west-1), with multiple storage profiles for each; whereas, an on-premises vSphere-backed cloud account can have a single region (e.g. sfo), with multiple storage profiles.

Storage profiles include cloud-specific configurations and a means to identify the type of storage by capability tags. Tags are then matched against the provisioning service request constraints to create the required storage at deployment time. You can also specify if a profile should be used as the default for an account / region.
## Table 2-22. Example Storage Profiles

<table>
<thead>
<tr>
<th>Account / Region</th>
<th>Cloud Account Type</th>
<th>Storage Profile Name</th>
<th>Storage Characteristics</th>
<th>Capability Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>sfo01dc01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Datastore/ Datastore Cluster: sfo01-w01-vsan</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Supports Encryption: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Storage Policy: Datastore Default</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Disk Mode: Dependant</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Supports Encryption: Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Device Type: EBS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Volume Type: Provisioned IOPS SSD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Max IOPS: 15,000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Supports Encryption: Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Storage Type: Managed Disks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Disk Type: Premium SSD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Storage Type: Managed Disks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Disk Type: Premium SSD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Storage Type: Managed Disks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Disk Type: Premium SSD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Storage Type: Managed Disks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Disk Type: Premium SSD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Storage Type: Managed Disks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Disk Type: Premium SSD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Storage Type: Managed Disks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Disk Type: Premium SSD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Preferred Storage: Enabled</td>
<td></td>
</tr>
</tbody>
</table>

**Note** The scope of this design includes VMware enabled clouds. Examples include both private and public cloud storage profiles for Cloud Assembly in a multi-cloud context.
Cloud-independent placement is possible. For example, different cloud storage might have different performance characteristics but still be considered for deployment based on capability tags. For example, a platform can have two different vendor accounts and a region. A storage profile for each region is tagged with a `tier:platinum` constraint `key:value` pair. During provisioning, a request containing a `tier:platinum:hard` tag for a hard constraint looks for a matching capability, regardless of the cloud vendor. A match then applies the settings from the associated storage profile during creation of the deployed storage item.

**Table 2-23. Design Decisions for Storage Profiles**

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CA-031</td>
<td>For each account / region, add one or more storage profiles based on storage characteristics available for consumption.</td>
<td>Allows you to add storage with defined characteristics that can be consumed during a deployment process.</td>
<td>You must manage storage profiles for each account / region as storage is added, removed, and updated in a cloud environment.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-032</td>
<td>For each storage profile, add one or more capability tags.</td>
<td>You use capability tags to manage the workload storage placement logic during the deployment process.</td>
<td>You must manage capability tagging on each storage profile for workflow placement logic during a deployment process.</td>
</tr>
</tbody>
</table>

**Tags Design**

Tags express capabilities and constraints that determine how and where resources are allocated to workloads during the provisioning process.

Capability and constraint tagging serves as the foundation for deployment configuration in Cloud Assembly. Tags are labels that you apply to Cloud Assembly constructs that enable policy-driven placement by directing how and where Cloud Assembly uses resources and infrastructure to build deployable services across private and public clouds. Structurally, tags must follow the `key:value` pair convention (e.g. `region:sfo`), but their construction is largely open.

**Overview**

Tags can be divided into two categories - capabilities and constraints. Capability tags placed on cloud zones, network, and storage profiles, and individual infrastructure resources define required connectivity, functionality, and capabilities for deployments. Constraint tags on projects provide governance over those projects. These constraint tags are added to other constraints expressed in blueprints. During a provisioning operation, Cloud Assembly matches capabilities with constraints tags in blueprints to determine the deployment configuration.

Tags also enable the search and identification of compute, storage, and network resources, as well as provisioned machines, using logical and natural language context.

Tags can be user-defined or discovered from sources external to Cloud Assembly. While tags can be created, applied, and matched from within Cloud Assembly, tags can also be automatically imported from a cloud account instance. These tags might be discovered and imported from vSphere, NSX Data Center, and VMware Cloud on AWS, as well as from public clouds like Amazon Web Services and Microsoft Azure. When imported, these tags are available for use in the same manner as the user-defined tags.
Standard Tags
Standard tags are applied automatically during provisioning on Amazon Web Services, Microsoft Azure, and vSphere deployments. Unlike other tags, users cannot use standard tags during deployment configuration, and no constraints are applied. Standard tags are stored as system custom properties and are added to deployments after provisioning.

Strategy
Before you create and use tags in Cloud Assembly, you must establish well-defined and adaptive tagging strategy and taxonomy. This will ensure that users who create and use tags understand what they mean, how they should be used, and where/when they should be applied. For example, which tags should be discovered (e.g. vSphere tags) and which should be user-defined and managed through Cloud Assembly.

Some practices for an effective tagging strategy:
- **Plan and Communicate**
  Create, communicate, and execute a plan for tagging that relates to the structure of your organization. Your plan must support your deployment needs, use clear natural language, and be understandable to all applicable users.
- **Simple and Adaptive**
  Use simple, clear, and meaningful names and values for tags. Users can easily understand capabilities and constraints when using tags in blueprints or reviewing tag assignments for a resource.

Capability Tags
Capability tags enable you to define placement logic for deployment.

You can create capability tags on resources, such as cloud zones, storage and storage profiles, networks and network profiles. Capability tags on storage or network components affect only the components on which they are applied.

Cloud Assembly matches capability tags with constraints from cloud zones and on blueprints at deployment time.

Constraint Tags
Constraint tags on blueprints and components match capabilities defined on resources, cloud zones, network and storage profiles to generate deployments with the required configuration.

Constraint tags are applied to two main constructs.
- **Project and image configuration**
- **Blueprint deployment.**

Constraints applied in both areas are merged in blueprints to form a set of deployment requirements.
When configuring Cloud Assembly, you apply constraint tags on projects which provide governance directly at the project level. All constraints added at this level are applied to all blueprints, requested for the applicable project. If a tag on a project conflicts with a tag on a blueprint, the project tag takes precedence, allowing you to enforce governance rules.

On blueprints, you add constraint tags as YAML code to match the appropriate capability tags that your cloud administrator created on resources, cloud zones, and storage and network profiles. In addition, there are other more complex options for implementing constraint tags. For example, you can use a variable to populate one or more tags on a request. This enables you to specify one or more of the tags at request time.

Create constraint tags by using the tag label in the blueprint YAML code. Constraint tags from projects are added to the constraint tags created in blueprints.

In the following example, the blueprint constraint attempts to deploy on objects with the `cloud:private` capability tag applied.

```yaml
constraints:
  - tag: cloud:private
```

In this example, you pass a blueprint expression for user selection with a blueprint input.

```yaml
inputs:
  targetCloud:
    type: string
    enum:
      - private
      - vmc
      - aws
      - azure
      - gcp

......
constraints:
  - tag: '${"cloud:" + to_lower(input.targetCloud)}'
```

If the input of `private` is selected, the constraint tag would be `cloud:private`.

Consider the following formats when you configure constraints.

- `key:value` and `key:value:hard`
  Must be provisioned on resources with the matching capability tag. The deployment process fails when no matching capability tag is found.

- `key:value:soft`
  Preferred provisioning on resources with the matching capability tag. The deployment process proceeds without failing and accepts resources without a matching tag.

- `!key:value`
  With hard or soft, avoid resources with a matching capability tag.
Sequencing and Simulation

The following list summarizes the high level operations and sequence of capability and constraint tag processing:

1. Cloud zones are filtered by several criteria, including availability and profiles. Tags in profiles for the zone are matched.

2. Cloud zone and compute capability tags are used to filter the remaining cloud zones by hard constraints.

3. Provisioning priority is used to select a cloud zone from the remaining filtered cloud zones. If there are several cloud zones with the same provisioning priority, they are sorted by matching soft constraints, using a combination of the cloud zone and compute capabilities.

4. After a cloud zone is selected, a host is selected by matching a series of filters, including hard and soft constraints as expressed in blueprints.

You can simulate a provisioning request to validate your configurations. Based on the provided values, the request will go through the projects, cloud zones, and profiles configurations without executing the provisioning.

Important: Design Decisions regarding the use of tags on cloud zones, projects, integrations, and profiles are provided within the specific architecture topics.

Table 2-24. Design Decisions for Tags

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-VMCS-CA-033</td>
<td>Establish and publish well-defined strategy and taxonomy for tagging of cloud resources.</td>
<td>Capability and constraint tags enable to you organize and activate cloud resources and profiles for resource consumption by using the declarative nature of blueprints to define deployment configuration.</td>
<td>Your strategy must account (e.g. vSphere and NSX Data Center tags) and user-defined tags, managed through Cloud Assembly.</td>
</tr>
<tr>
<td>SDDC-VMCS-CA-034</td>
<td>Apply constraint tags to blueprints in YAML code.</td>
<td>During a provisioning operation, capabilities are matched with constraints, each expressed as tags, in blueprints and images to determine the deployment configuration.</td>
<td>You must manage the capability tags on your cloud resources, such as cloud zones, storage and storage profiles, networks and network profiles.</td>
</tr>
</tbody>
</table>
Extensibility Design

Cloud Assembly extensibility enables you to assign an extensibility action or workflow to an event by using subscriptions. When the specified event occurs, the subscription initiates the action or workflow to execute.

- **Workflow Extensibility**
  You can use your vRealize Orchestrator hosted workflows with Cloud Assembly to extend application lifecycle. Using this integration enables use of your existing on-premises workflows with extensibility subscriptions.

- **Action-Based Extensibility**
  Action-based extensibility uses lightweight code within Cloud Assembly to automate extensibility actions. You can assign an extensibility action to a Cloud Assembly subscription to extend your application lifecycle.

**Workflow Extensibility**

You can use your vRealize Orchestrator hosted workflows with Cloud Assembly to extend application lifecycle. Using this integration enables use of your existing on-premises workflows with extensibility subscriptions.

Cloud Assembly uses vRealize Orchestrator integration to import and link workflows to a subscription. You can create, modify, and delete workflows using the vRealize Orchestrator Client. Workflows are maintained on the vRealize Orchestrator server instance.

Cloud Assembly supports the integration of multiple vRealize Orchestrator instances that can be used in workflow subscriptions. You can manage which vRealize Orchestrator integration in used in workflow subscriptions by using project extensibility constraint and capability tags.

Both soft or hard extensibility constraints on the project enable you to manage which vRealize Orchestrator integrations are used in blueprint provisioning for a project. Capability tags are also used to manage which vRealize Orchestrator integrations are used in workflow subscriptions.

For example, when you deploy a blueprint, Cloud Assembly uses the capability tags, associated with a cloud account (e.g. `cloud:private` and `region:sfo`), to manage what vRealize Orchestrator integrations are used in workflow subscriptions.

**Important** Workflow and event subscriptions usage is beyond the scope of this design. The design for vRealize Orchestrator is in scope for the design.

Refer to the VMware Cloud Assembly Documentation on how to integrate your vRealize Orchestrator workflows into the provisioning lifecycle.

**vRealize Orchestrator Design**

VMware vRealize Orchestrator is a development and process automation platform that provides a library of extensible workflows. With vRealize Orchestrator workflows, you can create and run automated configurable processes to manage the VMware vSphere infrastructure and other VMware and third-party technologies.
Cloud Assembly uses vRealize Orchestrator to enable the use of extensible workflows that can be invoked during workload provisioning and post-provisioning actions.

- **Physical Design for vRealize Orchestrator**
  This design uses the SaaS-enable vRealize Orchestrator provided by and required for Cloud Assembly integration.

- **Networking Design for vRealize Orchestrator**
  For isolation and co-location with on-premises cloud accounts and other integrations, you place the SaaS-enabled vRealize Orchestrator virtual appliance in the region-specific management VXLAN. The networking design also supports administrative access to the vRealize Orchestrator instances and their access to cloud account endpoints.

- **Authentication Provider for vRealize Orchestrator**
  To authenticate and manage user permissions, vRealize Orchestrator requires a connection to vSphere.

- **Information Security and Access Control for vRealize Orchestrator**
  You use service accounts, security groups, and roles to manage authentication and authorization. You establish secure communication to the vCenter Server instances by using CA-signed certificates.

- **Configuration of vRealize Orchestrator**
  The vRealize Orchestrator configuration includes guidance on client configuration, database configuration, SSL certificates, and plug-ins.

**Physical Design for vRealize Orchestrator**
This design uses the SaaS-enable vRealize Orchestrator provided by and required for Cloud Assembly integration.

**Deployment**

The SaaS-enabled vRealize Orchestrator appliance, which includes the plug-in for VMware Cloud Services, is provided as an OVA from the vRealize Orchestrator integration in Cloud Assembly. You deploy a vRealize Orchestrator appliance from the URL or OVA provided to the management domain cluster. After the deployment, you configure and register the vRealize Orchestrator appliance as a Cloud Assembly integration for your VMware Cloud Services organization. In a multi-region or multi-availability zone Software-Defined Data Center, you deploy a SaaS-enabled vRealize Orchestrator instance in each region.

The non-SaaS implementation of vRealize Orchestrator supports native standalone or cluster deployment modes. In cluster mode, multiple vRealize Orchestrator instances with identical server and plug-in configurations, are joined together as a cluster and share a replicated database.

The SaaS-enabled vRealize Orchestrator appliance supports standalone deployment model only.
Table 2-25. Design Decisions on SaaS-enabled vRealize Orchestrator Deployment

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
</table>
| SDDC-CAS-CA-VRO-001  | Deploy a single, standalone, SaaS-enabled vRealize Orchestrator instance in each region. | - A clustered deployment for the SaaS-enabled vRealize Orchestrator is not supported by VMware Cloud Services.  
- Ensures localized connection to cloud account endpoints and integrations per region if a cross-region network outage occurs.  
- Provides a consistent deployment model for management applications. | You must deploy, configure and integrate the SaaS-enabled vRealize Orchestrator instance in each region. |

Sizing Compute and Storage Resources

The SaaS-enabled vRealize Orchestrator appliance has the following resource requirements. Provide memory and CPUs for the operation of the appliance.

Table 2-26. Resource Specification of the SaaS-Enabled vRealize Orchestrator Appliance

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>2 vCPUs</td>
</tr>
<tr>
<td>Memory</td>
<td>6 GB</td>
</tr>
<tr>
<td>Disk size</td>
<td>1.3 GB Thin Provisioned</td>
</tr>
<tr>
<td></td>
<td>26.9 GB Thick Provisioned</td>
</tr>
</tbody>
</table>

Networking Design for vRealize Orchestrator

For isolation and co-location with on-premises cloud accounts and other integrations, you place the SaaS-enabled vRealize Orchestrator virtual appliance in the region-specific management VXLAN. The networking design also supports administrative access to the vRealize Orchestrator instances and their access to cloud account endpoints.

You deploy the SaaS-enabled vRealize Orchestrator appliance in the region-specific application virtual networks Mgmt-RegionA01-VXLAN and Mgmt-RegionB01-VXLAN.

Application Virtual Network Design

This networking design has the following features:

- Each SaaS-enabled vRealize Orchestrator instance has routed access to the management network through the universal distributed logical router (UDLR) for the SDDC cloud accounts endpoints deployed in the management cluster.

- Routing to the management network and the external network is dynamic, and is based on the Border Gateway Protocol (BGP).
For more information about the networking configuration of the application virtual networks for the SaaS-enabled vRealize Orchestrator, see Application Virtual Network and Virtual Network Design Example.

**Figure 2-8. Networking Design for the SaaS-enabled vRealize Orchestrator**
Table 2-27. Design Decisions on the Application Virtual Networks for the SaaS-enabled vRealize Orchestrator

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
</table>
| SDDC-CAS-CA-VRO-002     | Deploy the SaaS-enabled vRealize Orchestrator instances on the region-specific management application virtual network. | ■ Ensures localized connection to cloud account endpoints and integrations per region if a cross-region network outage occurs.  
■ Avoids cross-region bandwidth usage for cloud accounts endpoint.  
■ Provides a consistent deployment model for management applications. | You must use NSX Data Center for vSphere to support this network configuration. |

IP Subnets
You can allocate the following example subnets to the SaaS-enabled vRealize Orchestrator deployment.

Table 2-28. IP Subnets in the Application Virtual Networks for the SaaS-Enabled vRealize Orchestrator Instances

<table>
<thead>
<tr>
<th>Region</th>
<th>IP Subnet</th>
<th>VXLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region A</td>
<td>192.168.31.0/24</td>
<td>Mgmt-RegionA01-VXLAN</td>
</tr>
<tr>
<td>Region B</td>
<td>192.168.32.0/24</td>
<td>Mgmt-RegionB01-VXLAN</td>
</tr>
</tbody>
</table>

DNS Records
The name resolution for each SaaS-enabled vRealize Orchestrator appliance uses a region-specific suffix according to the region deployment, such as, sfo01.rainpole.local or lax01.rainpole.local.

Table 2-29. FQDNs for the SaaS-Enabled vRealize Orchestrator Instances

<table>
<thead>
<tr>
<th>Region</th>
<th>FQDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region A</td>
<td>sfo01vro01.sfo01.rainpole.local</td>
</tr>
<tr>
<td>Region B</td>
<td>lax01vro01.lax01.rainpole.local</td>
</tr>
</tbody>
</table>

Table 2-30. Design Decision on the DNS Records for SaaS-Enabled vRealize Orchestrator Instances

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-CAS-CA-VRO-003</td>
<td>Configure forward and reverse DNS records for each SaaS-enabled vRealize Orchestrator appliance.</td>
<td>Each SaaS-enabled vRealize Orchestrator instance is accessible by using a fully qualified domain name instead of by using IP addresses only.</td>
<td>You must provide forward and reverse DNS records for each Cloud Proxy appliance.</td>
</tr>
</tbody>
</table>

Authentication Provider for vRealize Orchestrator
To authenticate and manage user permissions, vRealize Orchestrator requires a connection to vSphere.
The SaaS-enabled vRealize Orchestrator supports two authentication provider methods during configuration.

- **vSphere Authentication**
  
  Utilizes the Platform Service Controller in vSphere 6.0 and 6.5 environments. This mode is used for Cloud Assembly.

- **vRealize Automation Authentication**
  
  Utilizes the vRealize Automation component registry for authentication. This mode is used when deploying as an external solution with vRealize Automation 7.x and is not applicable to Cloud Assembly.

To authenticate to the SaaS-enabled vRealize Orchestrator, you must configure the vSphere mode as the authentication provider for each region.

**Table 2-31. Design Decisions on SaaS-enabled vRealize Orchestrator Authentication and Authorization**

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-CAS-CA-VRO-004</td>
<td>Configure the SaaS-enabled vRealize Orchestrator instances to use the vSphere mode as the Authentication Provider.</td>
<td>Required to authenticate to the SaaS-enabled vRealize Orchestrator instance.</td>
<td>If you extend the design to include VMware Cloud on AWS as a cloud account and require a SaaS-enabled vRealize Orchestrator instance be deployed in that SDDC instance, the Authentication Provider must be configured to your on-premises SDDC. Registering of a provider requires vSphere administrative privileges which customers do not have with the VMware Cloud on AWS administrative account (<a href="mailto:cloudadmin@vmc.local">cloudadmin@vmc.local</a>). You must configure the vSphere mode Authentication Provider to an existing on-premises vSphere instance.</td>
</tr>
<tr>
<td>SDDC-CAS-CA-VRO-005</td>
<td>Configure the SaaS-enabled vRealize Orchestrator Authentication Provider mode to use a directory services security group. (e.g. rainpole.local\ug-vro-admins) as the vRealize Orchestrator Admin Group.</td>
<td>Allows only users defined in the directory service user group to administer the SaaS-enabled vRealize Orchestrator instance.</td>
<td>You must maintain the security group lifecycle (e.g. add or remove users) outside of the SDDC stack to ensure its membership. <strong>Note</strong> In an Active Directory forest, consider using a Universal group.</td>
</tr>
</tbody>
</table>

**Information Security and Access Control for vRealize Orchestrator**

You use service accounts, security groups, and roles to manage authentication and authorization. You establish secure communication to the vCenter Server instances by using CA-signed certificates.
Authentication and Authorization

Configure a service accounts, security groups, and vCenter Server roles to control and manage administrative access to the SaaS-enabled vRealize Orchestrator instance and for the vRealize Orchestrator-to-workload domain vCenter Server endpoint instances. You define a service account with only the minimum set of permissions to perform desired operations on the workload domain instances defined in the SDDC.

Table 2-32. Design Decisions on SaaS-enabled vRealize Orchestrator Authentication and Authorization

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
</table>
| SDDC-CAS-CA-VRO-006 | Configure a service account in vCenter Server for application-to-application communication from the SaaS-enabled vRealize Orchestrator instance to the workload domain vCenter Server instances in each region. (e.g. svc-vro-vsphere). | Provides the following access control features:  
- Cloud Automation Services accesses the SaaS-enabled vRealize Orchestrator with the minimum set of required permissions.  
- If there is a compromised account, the accessibility in the destination application remains restricted.  
- You can introduce improved accountability in tracking request-response interactions between the components of the SDDC. | You must maintain the service account's lifecycle outside of the SDDC stack to ensure its availability. |
| SDDC-CAS-CA-VRO-007 | Configure a service account in vCenter Server for application-to-application communication from Cloud Automation Services to the SaaS-enabled vRealize Orchestrator instance in each region. (e.g. svc-cas-vro.) | Provides the following access control features:  
- The SaaS-enabled vRealize Orchestrator instance accesses the workload domain vCenter Server instance with the minimum set of required permissions.  
- If there is a compromised account, the accessibility in the destination application remains restricted.  
- You can introduce improved accountability in tracking request-response interactions between the components of the SDDC. | You must maintain the lifecycle and availability of the service account outside of the SDDC stack. |
Table 2-32. Design Decisions on SaaS-enabled vRealize Orchestrator Authentication and Authorization (continued)

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-CAS-CA-VRO-008</td>
<td>Add the service account used for application-to-application communication from the SaaS-enabled vRealize Orchestrator instance to a directory services security group (e.g. <code>rainpole.local\ug-vro-admins</code>). Add any named administrative groups or users to the security group.</td>
<td>Allows only groups and users defined in the directory services security group to administer the SaaS-enabled vRealize Orchestrator instance.</td>
<td>You must maintain the security group lifecycle (e.g. add or remove users) outside of the SDDC stack to ensure its membership.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> In an Active Directory forest, consider using a security group with a Universal scope. Add security groups with a Global scope that include service accounts and users from the domains in the Active Directory forest.</td>
<td><strong>Note</strong> In an Active Directory forest, consider using a security group with a Universal scope. Add security groups with a Global scope that include service accounts and users from the domains in the Active Directory forest.</td>
<td><strong>Note</strong> In an Active Directory forest, consider using a security group with a Universal scope. Add security groups with a Global scope that include service accounts and users from the domains in the Active Directory forest.</td>
</tr>
<tr>
<td>SDDC-CAS-CA-VRO-009</td>
<td>Create and apply a custom vCenter Server role for the service account(s) used to add workload domain vCenter Server instance(s) to the SaaS-enabled vRealize Orchestrator configuration (e.g. Cloud Assembly - vRealize Orchestrator to Workload Domain User ).</td>
<td>Provides the ability to limit the privileges for application-to-application integration.</td>
<td>You must define the minimum privileges for the custom role for your organization.</td>
</tr>
<tr>
<td>SDDC-CAS-CA-VRO-010</td>
<td>Use vCenter Server local permissions for the SaaS-enabled vRealize Orchestrator service account in vCenter Server by assigning the user and role to the workload domains vCenter Server instances.</td>
<td>Ensures that only the workload domain vCenter Server instance are accessible endpoints from SaaS-enabled vRealize Orchestrator instance.</td>
<td>As you add more workload domain vCenter Server instances, you must assign the service account local permissions in each additional vCenter Server so that each is a viable endpoint for the SaaS-enabled vRealize Orchestrator instance.</td>
</tr>
</tbody>
</table>

Encryption

The vRealize Orchestrator configuration interface uses a secure connection to communicate with workload domain vCenter Server instances, database systems, LDAP, and other servers. You can import an SSL certificate from a URL or PEM file to replace the certificate used for the SaaS-enabled vRealize Orchestrator instance as well as those SSL certificates that the instance must trust (e.g. an Enterprise Certificate Authority).
You can import workload domain vCenter Server instance SSL certificates from **Certificates > Trust Certificate** in the vRealize Orchestrator HTML5-based ControlCenter UI at https://saas_enabled_vro_hostname:8283/vco-controlcenter.

### Table 2-33. Design Decisions on SaaS-enabled vRealize Orchestrator Encryption

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-CAS-CA-VRO-011</td>
<td>Replace the default self-signed certificate of the SaaS-enabled vRealize Orchestrator appliance with a CA-signed certificate.</td>
<td>Configuring a CA-signed certificate ensures that the communication to the SaaS-enabled vRealize Orchestrator instance and between its integrations is encrypted.</td>
<td>■ Replacing the default certificates with trusted CA-signed certificates from a certificate authority increases the deployment preparation time as certificates requests are generated and delivered. ■ You must manage the lifecycle of the certificate replacement.</td>
</tr>
<tr>
<td>SDDC-CAS-CA-VRO-012</td>
<td>Import the certificate for each workload domain vCenter Server instance in a region to the SaaS-enabled vRealize Orchestrator instance in the same region.</td>
<td>Ensures that the certificate for each workload domain vCenter Server instance in a region is trusted by the SaaS-enabled vRealize Orchestrator instance in the same region.</td>
<td>As workload domains are added (or removed) you must add (or remove) the vCenter Server instance SSL certificate trust from the SaaS-enabled vRealize Orchestrator instance.</td>
</tr>
</tbody>
</table>

**Configuration of vRealize Orchestrator**

The vRealize Orchestrator configuration includes guidance on client configuration, database configuration, SSL certificates, and plug-ins.

**vRealize Orchestrator Client**

With the vRealize Orchestrator Client application, you can import packages, create, run, and schedule workflows, and manage user permissions.

The vRealize Orchestrator Client is available as a browser-based (HTML5) client from the vRealize Orchestrator start page at https://saas_enabled_vro_hostname:8281/vco.

You can also install the standalone version of the vRealize Orchestrator Client on Microsoft Windows, Mac OS, or Linux by downloading the vRealize Orchestrator Client installation files from the SaaS-enabled vRealize Orchestrator appliance page at https://saas_vro_hostname:8281/vco. Alternatively, you can run the vRealize Orchestrator Client using Java WebStart directly from https://saas_enabled_vro_hostname:8281/vco.

**vRealize Orchestrator Plug-ins**

You use vRealize Orchestrator plug-ins to access and control external services and applications. The external technologies that you can access by using plug-ins include visualization management tools, email systems, databases, directory services, and remote control interfaces. vRealize Orchestrator provides a set of standard plug-ins for technologies as the vCenter Server API and email capabilities.

**vCenter Server Plug-in**
You can use the vCenter Server plug-in to manage multiple workload domain vCenter Server instances. You can create workflows that use the vCenter Server plug-in API to automate tasks in your workload domain environment. The vCenter Server plug-in maps the vCenter Server API to JavaScript code that you can use in workflows. The plug-in also provides actions that perform individual vCenter Server tasks that you can include in workflows.

The vCenter Server plug-in provides a library of standard workflows that automate vCenter Server operations. For example, you can run workflows that create, clone, migrate, or delete virtual machines. Before managing and running workflows on the objects in your vSphere inventory, you must configure the vCenter Server plug-in and connect vRealize Orchestrator to the workload domain vCenter Server instances that you want to orchestrate.

Table 2-34. Design Decisions on SaaS-enabled vRealize Orchestrator Plug-ins

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implications</th>
</tr>
</thead>
</table>
| SDDC-CAS-CA-VRO-014 | Register each workload domain vCenter Server instance in a region with the corresponding SaaS-enabled vRealize Orchestrator instance in the same region. For example: Region A vRealize Orchestrator instance (sfo01vro01.sfo01.rainpole.local) > Region A Workload Domain vCenter Server instance (sfo01w01vc01.sfo01.rainpole.local) | Required for communication from the SaaS-enabled vRealize Orchestrator to vCenter Server instances. | - As workload domains are added (or removed) you must add (or remove) the vCenter Server instance in vRealize Orchestrator.  
- You must execute the Update a vCenter Server Instance workflow with new log-in properties when the service account password changes during its lifecycle. |

**Action-Based Extensibility**

Action-based extensibility uses lightweight code within Cloud Assembly to automate extensibility actions. You can assign an extensibility action to a Cloud Assembly subscription to extend your application lifecycle.

Action-Based Extensibility (ABX) provides a lightweight and flexible run-time where you can define small scriptable actions and configure them to initiate on particular events provided through the Event Broker Service (EBS). ABX supports both Node.js and Python run-time environments.
You can create or import extensibility action scripts in Cloud Assembly and assign them to subscriptions. Similarly to workflows, the extensibility action script triggers when an event included in an extensibility subscription occurs. Extensibility action scripts are used for more lightweight integrations and customizations. ABX is a functions-as-a-service (FaaS) through a cloud provider, such as Amazon Web Services Lambda or Microsoft Azure Functions, as opposed to workflows which are hosted on-premises by using a vRealize Orchestrator. Cloud accounts for Amazon Web Services or Microsoft Azure are required for production ABX use.

**Note** An on-premises Actions-Based Extensibility integration for Cloud Assembly is in Beta. The Beta integration requires the use of a dedicated Cloud Extensibility Proxy.

You can create extensibility actions by either writing a user-defined action script code or importing a predefined script code as a .ZIP package.

**Important** Action-Based Extensibility and event subscriptions usage is beyond the scope of the design. Refer to the VMware Cloud Assembly Documentation on integrating ABX into the provisioning lifecycle.

<table>
<thead>
<tr>
<th>Table 2-35. Design Decisions for Action-Based Extensibility</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision ID</strong></td>
</tr>
<tr>
<td>SDDC-VMCS-CA-042</td>
</tr>
</tbody>
</table>

**Service Broker Design**

Service Broker aggregates content in native formats from multiple clouds and platforms into a common catalog with the ability to apply governance based on roles and projects.

- **Service Roles Design**
  You manage access to Service Broker by assigning enterprise groups to service roles in your organization.

- **Catalog Content Design**
  The availability of the catalog items in Service Broker is determined by project membership. Projects link users, catalog items, and cloud resources.

- **Policies Design**
  Policies manage and process deployments requested from the Service Broker catalog.
Service Roles Design

You manage access to Service Broker by assigning enterprise groups to service roles in your organization.

Service Broker has two service roles assigned from the organization's Identity and access management. You assign the service roles to designated enterprise groups, synchronized from your corporate identity source through federation.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>Example Enterprise Group</th>
</tr>
</thead>
</table>
| Service Broker Administrator | - Read and write access to the entire user interface and API resources.  
                                 | - Configure content sources, sharing, and customizations.  
                                 | - Configure policies.                     | rainpole.local\ug-vcms-cas-sb-admins |
| Service Broker User        | Request services from projects.                                             | rainpole.local\ug-vcms-cas-sb-users        |

Catalog Content Design

The availability of the catalog items in Service Broker is determined by project membership. Projects link users, catalog items, and cloud resources.

Catalog content sources can include blueprints, actions, and workflows. Content sources are entitled to projects. Projects link a set of users (project members) with one or more target cloud zone regions or datastores.

When users requests a catalog item, the deployment location is dependant on the project. Projects might have one or more cloud zones.

Content Sources Design

Before you can release content, such as Cloud Assembly blueprints and actions, to the Service Broker catalog for Project Members, you must add a content source based on the content type and project.

You create content sources in Service Broker to import content types into the catalog.

<table>
<thead>
<tr>
<th>Content Source Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Blueprints          | Cloud Assembly Blueprints:  
                                 | Versioned and released blueprints, created and managed in a project, that are either synchronized from a Git repository or directly from within Cloud Assembly. |
| Actions             | Cloud Assembly Actions:  
                                 | Versioned and released extensibility actions, created and managed in a project, that are either synchronized from a Git repository or directly from within Cloud Assembly.  
                                 | Important | On-premises Action-Based Extensibility (ABX) is currently in Beta. The design will be updated to include ABX when the feature is Generally Available (GA).  
                                 | ABX using Amazon Web Services or Microsoft Azure is not applicable to this design. |
By default, content items are automatically refreshed every six hours. Any released changes in an item from the source are reflected in the catalog after the refresh. Content can also be refreshed outside the standard cycle by initiating an on-demand import operation.

**Table 2-36. Design Decisions for Content Sources**

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-CAS-SB-001</td>
<td>Add a blueprint content source for each Cloud Assembly project where blueprints are authored and released.</td>
<td>Provides the ability to share released blueprints with project members or other projects.</td>
<td>None.</td>
</tr>
</tbody>
</table>

**Customization Design**

You can customize icon and request forms for catalog items in Service Broker to enhance the user experience and capture additional workload inputs.

You use the content list to view the import source and entitled projects for each item. For each item you can customize the catalog item icon and request form that is presented to project members during a request. An icon customization allows you to use non-default icons to represent the catalog item and provide a better user experience to project members.

Form customizations allow you to add and configure elements, as well as data validation on a custom request form. By using input parameters, you can create useful forms and design how the information appears at request time, how the parameter values are populated, and add any specialized constraints. Custom forms can be enabled or disabled on a per catalog item basis.
Table 2-37. Design Decisions for Content Customization

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDDC-CAS-SB-002</td>
<td>For each shared content item, customize the icon based on the catalog item.</td>
<td>Allows you to provide a meaningful visual indicator on the catalog item or type to</td>
<td>Images must not exceed the dimension and file size maximums.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>provide a better user experience for project members.</td>
<td>PNG format images are not supported.</td>
</tr>
<tr>
<td>SDDC-CAS-SB-003</td>
<td>For each shared content item, customize the form based on the catalog item and user experience requirements.</td>
<td>Allows you to create an intuitive user experience by using simple and discoverable forms that capture additional user inputs and in-form validations.</td>
<td>Requires customization of request forms per catalog item.</td>
</tr>
</tbody>
</table>

Sharing

You publish imported content and make it available in the Service Broker catalog for project members. After you create a content source and import content items, you can share the items based on the release scope. When creating a blueprint in Cloud Assembly, you set the scope for sharing in Service Broker. You can restrict the blueprint to be shared only within its own project or make them available to any project in the organization.

For each project, you add released content items that are shared with project members. Items can be added by using two methods:

- **Content Source**
  A dynamic method to share all content items from the content source. As new items are added to a content source, each is made available in the catalog for the members of a target project.

- **All Content**
  A static method for sharing specific content items within a project. As new items are added to the content source, you must make them available in the catalog individually.

The content sharing method you choose is dependant on the your organization requirements, project structure, and the level of control required when releasing content to the catalog.

Policies Design

Policies manage and process deployments requested from the Service Broker catalog.

Policies include definitions - a set of rules or parameters for specific use. The definition commonly includes the scope and enforcement type. The scope of a definition determines if the policy is applicable to all deployments in an organization or only deployments in a selected project.

At the time of this publication, two policy types exist in Service Broker:

1. **Lease Policy** - allows you to set a workload lease policy.
2 Day-2 Actions Policy - allows you to select the actions that can be performed on a workload post-deployment.

**Note** Additional policy types are added to Service Broker as released to the SaaS platform.

When a project member requests a blueprint, there might be more than one policy that applies. An enforcement type defines how multiple policies are evaluated, ranked, and, where applicable, merged, to produce an effective policy. An effective policy produces the intended results but is not always a specific named policy.

<table>
<thead>
<tr>
<th>Enforcement Type</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard</td>
<td>Ranked higher than soft policies</td>
</tr>
<tr>
<td>Soft</td>
<td>Overridden by hard policies</td>
</tr>
</tbody>
</table>

During the evaluation phase, Service Broker first identifies and ranks policies.

1. Policy types are evaluated:
   a. If there are hard and soft policies, then only the hard policies are considered and ranked.
   b. If there are only soft policies, then the soft policies are ranked.

2. Policies with an organizational scope are ranked higher rank than policies with a project scope.

3. Policies with older creation dates are ranked higher than policies with newer creation dates. Post-ranking, policies are evaluated to identify the merge order.

4. The highest ranking policy becomes the baseline.

5. The second-level ranking policy is applied next, and so on.

6. If a policy is incompatible with the preceding policies, then it is discarded and marked as ineffective.

**Table 2-38. Example Lease Policy**

<table>
<thead>
<tr>
<th>Goal</th>
<th>Example Configuration</th>
<th>Effective Policy Behavior</th>
</tr>
</thead>
</table>
| A default organization-level policy that allows the project-level policy values to influence the applied values. | Organization Policy = Soft  
  - Grace period: 10  
  - Lease: 100  
  - Total Lease: 100  
  Project A Policy 1 = Soft  
  - Lease: 20  
  - Total Lease: 50  
  Project B Policy 1 = Soft  
  - Lease: 10  
  - Total Lease: 30 | A member of Project A requests a catalog item.  
Project B is not considered because it is not applicable to Project A deployments.  
**Merged Effective Policy:**  
  - Grace period: 10  
  - Lease: 20  
  - Total Lease: 100 |
Table 2-39. Design Decisions for Policies

<table>
<thead>
<tr>
<th>Decision ID</th>
<th>Design Decision</th>
<th>Design Justification</th>
<th>Design Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMCS-CAS-SB-004</td>
<td>Identify and apply goals for your organization and each project based on the applicability of available policy types.</td>
<td>By understanding how the policies are processed, you can meet organizational goals without creating an excessive and unmanageable number of policies.</td>
<td>For each policy type you must determine the applicability and your organization’s goals to design policy enforcement and scope that results in the desired effective policy.</td>
</tr>
</tbody>
</table>