

How to Modify Input Model

Contents

How to Modify Input Model.....3

How to Modify Input Model

This page describes the configuration of Swift services with the various types of input models supported in this release.

Types of Input Models

- **Entry-scale-with-vsa** This has a single control plane cluster running all services (Nova, Swift, Glance, etc) with an additional 3 resource nodes used as Nova compute servers.

Entry-scale-with-vsa

The `entry-scale-with-vsa` is relatively simple and constrained. You can ensure that the disk drives exists and are correctly assigned to swift nodes. Hence, you need not modify this model for any Swift-related changes before implementing it.

Before starting the deployment, you must validate, and possibly update the data in the cloud model files as follows:

- Examine the `data/disks_controller.yml` file and validate that the drives exist and have the appropriate preconditions as described in [Requirements for a disk device](#).
- If you plan to add more Swift storage nodes or drives, you should read the information about partition power in [Specifying the rings \(ring specifications\)](#). However, in most cases you do not need to change the `config/swift/rings.yml` file.

Node Type

Proxy, Container, Account, Object (PACO) Node: The Swift-proxy, Account, Container and Object services run on the same (PACO) node type in the control plane. All other control services also run on this node type.

YML files associated with Swift Services

The following `yml` files are configured for **entry-scale-with-vsa** model:

<code>data/control_plane.yml</code>	This ROLE-CONTROLLER node type runs on the c1 cluster as specified in this file.
<code>data/disk_controller.yml</code>	The controller node type allocates two disk drives for use by Swift.
<code>config/swift/rings.yml</code>	Swift account, container, and object storage are managed by Swift using a data structure known as a <i>ring</i> . This <code>yml</code> file provides the specification of the rings. For more information on ring specification, refer to ring specification .
<code>data/network_groups.yml</code>	This <code>yml</code> file specifies the MGMT and Swift networks. The Swift-proxy service uses the Swift network to communicate with the other Swift services and among themselves.

Mid Size Cloud Model

This example model describes a system using multiple servers and networks. It is representative of how a mid sized cloud is configured. You are expected to use the example model as the basis for building a cloud model for your actual cloud – using the servers and networks you plan to deploy. This document describes the Swift-specific changes you need to make to the provided example.

Before starting the deployment, you must validate, and possibly update the data in the cloud model files as follows:

- You need to assign servers for use by Swift. See [Allocating Servers for use by Swift](#)
- These servers have disk drives. See [Allocating disk drives for use by Swift](#)
- You need to specify the rings for your system. See [Specifying the rings \(ring specifications\)](#)

Node Types

- **Proxy, Container, Account (PAC) Node:** This node runs the Swift-proxy, Swift-account, and Swift-container services. The Swift-proxy service processes API requests and directs them to the Swift-account, Swift-container or Swift-object services for processing. The Swift-account and Swift-container handle requests to accounts and containers respectively.
- **Object (OBJ) Node:** This node runs the Swift-object service. The Swift-object service handles requests for objects.

YML files associated with Swift Services

In the example mid sized cloud model, the Swift services are configured using the following yml files:

data/control_plane.yml	<p>The Proxy, Container, and Account node types are assigned to a dedicated <i>cluster of nodes</i> as specified in the swpac cluster in yml file. These nodes are dedicated only to Swift services.</p> <p>The Object node type is assigned to a dedicated <i>resource nodes</i> group as specified in the swobj group in yml file. Requests to Swift are directed to a virtual IP address (VIP) that is managed by a cluster as specified in the core cluster in data/control_plane.yml file. The requests are then directed on the MGMT network to the Swift-proxy service on one of the Proxy, Container, Account nodes.</p>
data/disks_swpac.yml	The Proxy, Container, Account node type uses two disk drives to store account and container databases which is specified in this file. For more information about allocating a disk drives, refer to Allocating Disk Drives
data/disks_swobj.yml	The Object node type uses two disk drives to store account and container databases which is specified in this file. For more information about allocating a disk drives, refer to Allocating Disk Drives .
config/swift/rings.yml	Swift account, container, and object storage are managed by Swift using a data structure known as a <i>ring</i> . This yml file provides the specification of the rings. For more information on ring specification, refer to ring specification .
data/network_groups.yml	This yml file specifies the MGMT and Swift networks. The Swift-proxy service uses the Swift network to communicate with the other Swift services and among themselves. For more information about allocating a disk drives, refer to

The Swift-proxy service uses two other cloud services. Both of these are configured in the example to run on the **core** cluster:

- Swift validates tokens by making requests to the Keystone service

- Swift caches tokens and other data using the memcached service