

HP Helion OpenStack 2.0: Object Storage (Swift)

Contents

HP Helion OpenStack® 2.0: Object Storage (Swift)..... 3

HP Helion OpenStack® 2.0: Object Storage (Swift)

What is Object Storage (Swift)?

Object Storage (Swift) is Software Defined Storage (SDS) architecture layered on top of industry-standard servers using native storage devices. Swift presents an object paradigm using an underlying set of disk drives. The disk drives are managed by a data structure called a "ring". Swift also enable you to store, retrieve, and delete objects in a containers through RESTful API.

Object Storage (Swift) Services

A Swift system comprises a number of services:

- Swift-Proxy provides the API for all requests to the Swift system.
- Account and Container services provide storage management of the accounts and containers.
- Object services provide storage management for object storage.

HP Helion OpenStack® Object Storage(Swift)

Based on OpenStack® Object Storage, the HP Helion OpenStack Object Storage service provides a highly available, resilient, and scalable storage pool for unstructured data. It is highly durable architecture with no single point of failure.

In addition, HP Helion OpenStack introduces the concept of example cloud models. This model allows the user to install swift as per your requirement.

In this release we are supporting the following cloud input models:

- entry-scale-with-vs-a of cloud model
- mid-size cloud model

The Swift system services can be co-located in a number of ways. In the example cloud models the following two general patterns exist:

- The Swift-proxy, Account, Container, and Object services run on the same (PACO) node type in the control plane. This is used for smaller clouds or where Swift is a minor element in a larger cloud.
- The Swift-Proxy, Account, and Container services run on one (PAC) node type in a control plane and the Object services run on another (OBJ) node type in a resource pool. This is used for in larger clouds or where a larger Swift system is in use or planned.

Each of these node types can be scaled independently to meet workload and redundancy requirements. The storage service can be scaled up as necessary. It can also scaled horizontally to handle an increased number of simultaneous connections and a larger number of objects.