Airship V1 Installer Deployment Guide

Overview

This document provides concepts and procedures for deploying an NFVi with Airship 1 Installer in a bare metal infrastructure.

This document includes the following content:

- Introduction to the upstream tool set used by the Airship Installer, for example, Airship Project, OpenStack Helm, Treasuremap, and so on.
- Instructions for preparing a site manifest in declarative YAML, including hardware profile and software stack, according to the hardware infrastructure and software component model specified in the NFVi reference model and reference architecture.
- Instructions for customizing the settings in the site manifest.
- Instructions for running the deployment script.
- Instructions for setting up a CI/CD pipeline for automating deployment and testing.

Intel Pod 17 is used to deploy reference NFVi. Therefore, the examples in this document are based on the hardware profile of Intel Pod 17. Instructions are either referenced (to the upstream document) or provided (in this document) so that the reader can modify the settings of the hardware profile and/or software stack accordingly.

Airship

Airship is a collection of loosely coupled and interoperable open source tools that declaratively automate cloud provisioning.

Airship is a robust delivery mechanism for organizations who want to embrace containers as the new unit of infrastructure delivery at scale. Starting from raw bare metal infrastructure, Airship manages the full lifecycle of data center infrastructure to deliver a production-grade Kubernetes cluster with Helm deployed artifacts, including OpenStack-Helm. Airship allows operators to manage their infrastructure deployments and lifecycle through the declarative YAML documents that describe an Airship environment.

For more information, see https://www.airshipit.org/.

OpenStack Helm

OpenStack-Helm is a set of Helm charts that enable deployment, maintenance, and upgrading of loosely coupled OpenStack services and their dependencies individually or as part of complex environments.

For more information, see https://wiki.openstack.org/wiki/Openstack-helm/.

Treasuremap

Treasuremap is a deployment reference as well as CI/CD project for Airship.

Airship site deployments use the `treasuremap` repository as a `global` manifest set (YAML configuration documents) that are then overridden with site-specific configuration details (networking, disk layout, and so on).

For more information, see https://airship-treasuremap.readthedocs.io/ .

Manifests

Airship is a declarative way of automating the deployment of a site. Therefore, all the deployment details are defined in the manifests.

The manifests are divided into three layers: `global`, `type`, and `site`. They are hierarchical and meant as overrides from one layer to another. This means that `global` is baseline for all sites, `type` is a subset of common overrides for a number of sites with common configuration patterns (such as similar hardware, specific feature settings, and so on), and finally the `site` is the last layer of site-specific overrides and configuration (such as specific IP addresses, hostnames, and so on). See Deckhand documentation for more details on layering.

The `global` and `type` manifests can be used *as is* unless any major differences from a reference deployment are required. In the latter case, this may introduce a new type, or even contributions to the `global` manifests.

The site manifests are specific for each site and are required to be customized for each new deployment. The specific documentation for customizing these documents is located here:

- Airship Site Authoring and Deployment Guide
- Code comments in the manifests themselves, for example common-addresses.yaml
- As well as each individual chart of components, for example, Deckhand chart values.yaml

Global

Global manifests, defined in Airship Treasuremap, contain base configurations common to all sites. The versions of all Helm charts and Docker images, for example, are specified in versions.yaml.

Type

The type `cntt` will eventually support specifications published by the CNTT community. See CNTT type.

Site

The site documents reside under the `site` folder. While the folder already contains some sites, and will contain more in the future, the `intel-pod17` site shall be considered the Airship OPNFV reference site. See more at POD17 manifests.

The site-definition.yaml ties together `site` with the specific type and `global` manifests.

```
data:
  site_type: cntt

  repositories:
    global:
      revision: v1.7
      url: https://opendev.org/airship/treasuremap.git
```

Deployment

As Airship is tooling to declaratively automate site deployment, the automation from the installer side is light. See deploy.sh.

You will need to export environment variables that correspond to the new site (keystone URL, node IPs, and so on). See the beginning of the deploy script for details on the required variables.

Once the prerequisites that are described in the Airship deployment guide (such as setting up Genesis node), and the manifests are created, you are ready to execute `deploy.sh` that supports Shipyard actions: `deploy_site` and `update_site`.

```
$ tools/deploy.sh
Usage: deploy.sh <deploy_site|update_site>
```

CI/CD

TODO: Describe pipelines and approach

https://build.opnfv.org/ci/view/airship/

OpenStack

The `treasuremap` repository contains a wrapper script for running OpenStack clients tools/openstack. The wrapper uses `heat` image that already has openstack client installed.

Clone latest `treasuremap` code

```
$ git clone https://github.com/airshipit/treasuremap.git
```

Setup the needed environment variables, and execute the script as openstack CLI

```
$ export OSH_KEYSTONE_URL='http://identity-airship.intel-pod17.opnfv.org/v3'
$ export OS_REGION_NAME=intel-pod17
$ treasuremap/tools/openstack image list
```