**Introduction**

**Overview of StarlingX Distributed Cloud**

Distributed cloud configuration supports an edge computing solution by providing central management and orchestration for a geographically distributed network of Kubernetes-only StarlingX systems.

The Distributed Cloud Kubernetes-only StarlingX deployment is designed to meet the needs of edge-based data centers in which NFC worker resources are localized for maximum responsiveness, while management and control functions are centralized for efficient administration. The system supports a scalable number of Kubernetes-only StarlingX edge systems, centrally managed and synchronized over L3 networks from a central region. Each edge system is also highly scalable, from a single Kubernetes-only StarlingX deployment to a full Standard Cloud configuration with storage nodes.

The architecture features a synchronized distributed control plane for reduced latency, with an autonomous control plane such that all subcloud local services are operational even during loss of northbound connectivity to Central Region.

**Distributed Cloud Architecture**

A Distributed Cloud system consists of a Central Cloud, and one or more subclouds connected to the System Controller region Central Cloud over L3 networks.

The Central Cloud has two regions: RegionOne, used to manage the nodes in the Central Cloud, and SystemController, used to manage the subclouds in the Distributed Cloud system.

* Central Cloud

The Central Cloud provides a RegionOne region for managing the physical platform of the Central Cloud and the SystemController region for managing and orchestrating over the subclouds.

* RegionOne

RegionOne is the name of the access mode, or region, for managing the Central Cloud’s physical platform.

* SystemController

SystemController is the name of the access mode, or region, for managing the subclouds.

You can use the SystemController to add subclouds, synchronize select configuration data across all subclouds and monitor subcloud operations and alarms. System software updates for the subclouds are also centrally managed and applied from the SystemController.

DNS, NTP, and other select configuration settings are centrally managed at the System Controller and pushed to the subclouds in parallel to maintain synchronization across the Distributed Cloud.

* Subclouds

The subclouds are StarlingX Kubernetes-only systems used to host containerized applications. Any type of StarlingX Kubernetes configuration, i.e. simplex, duplex or standard with or without storage nodes, can be used for a subcloud.

Alarms raised at the subclouds are sent to the System Controller for central reporting.

**Installation**

**Networks**

Subclouds are connected to the System Controller over L3 networks. Because each subcloud is on a separate L3 subnet, the management and PXE boot L2 networks are local to the subclouds and not connected via L2 to the Central Cloud, they are only connected via L3 routing. The settings required to connect a subcloud to the System Controller are specified when a subcloud is defined. For more information, see Installing a Subcloud. However, a gateway router is required to complete the L3 connections, which will provide IP routing between the subcloud management subnet and the SystemController management subnet.

**Install a Distributed Cloud System on Bare Metal**

Installing and Provisioning the Central Cloud

Installing the Central Region Cloud is similar to installing a standard StarlingX Kubernetes system. The Central Cloud supports either an AIO-Duplex deployment configuration or a Standard with Dedicated Storage Nodes deployment configuration.

To configure controller-0 as a distributed cloud central controller, you must set the system parameter ‘distributed\_cloud\_role’ to ‘systemcontroller’ in the Ansible bootstrap override file during the initial bootstrapping of controller-0 prior to Ansible bootstrapping.

Also during initial controller-0 bootstrapping, limit the management network IP address range to exclude IP addresses reserved for gateway routers providing routing to the subclouds’ management subnets.

NOTE: Worker hosts and data networks are not used in the Central Region Cloud.

Procedure:

·         Follow the StarlingX R3.0 Installation procedures at:

*For AIO-Duplex:*

[*https://docs.starlingx.io/deploy\_install\_guides/r3\_release/bare\_metal/aio\_duplex.html*](https://docs.starlingx.io/deploy_install_guides/r3_release/bare_metal/aio_duplex.html)

*or*

*For Standard with Dedicated Storage Nodes:*

[*https://docs.starlingx.io/deploy\_install\_guides/r3\_release/bare\_metal/dedicated\_storage.html*](https://docs.starlingx.io/deploy_install_guides/r3_release/bare_metal/dedicated_storage.html)

·         *With the following exception:*

*@ “Bootstrap system on controller-0” STEP*

Add the following additional parameters to Ansible bootstrap override file in order to indicate that this cloud will play the role of the Central Cloud / SystemController, and restrict the range of addresses for the management network to exclude the IP addresses reserved for gateway routers providing routing to the subclouds’ management subnets.

 *distributed\_cloud\_role: systemcontroller

management\_start\_address: < X.Y.Z.2 >*

*management\_end\_address: < X.Y.Z.50 >*

Installing a Subcloud

There are two stages to installing a subcloud.

At the subcloud location:

1. Physically install and cable all subcloud servers,

2. Physically install the top of rack switch and configure it for the required networks,

3. Physically install the gateway router which will provide IP routing between the subcloud management subnet and the SystemController management subnet

4. On the server designated for controller-0, you must install the StarlingX Kubernetes software from USB or a PXE Boot Server,

5. Enable the management interface on the subcloud controller.

This is required in order to establish an L3 connection to the System Controller. An L3 connection is required in order to configure the subcloud successfully.

To enable the management interface, run the config\_management script on the subcloud.

Be prepared to provide the following information:

• The subcloud management interface name (for example, enp0s8)

• The subcloud management interface address, in CIDR format (for example, 192.168.101.2/24)

• NOTE: This must match the Management Start IP supplied in the subcloud’s ansible bootstrap override file.

• The subcloud gateway address on the management network (for example, 192.168.101.1). A default value is shown

• The System Controller management subnet (for example, 192.168.204.0)

NOTE: To exit without completing the script, use CTRL+C. Allow few minutes for the script to finish.

$ sudo config\_management

Enablng interfaces... DONE

Waiting 120 seconds for LLDP neighbor discovery... Retrieving neighbor details... DONE

... DONE

Available interfaces:

local interface remote port

--------------- ----------

enp0s3 08:00:27:c4:6c:7a

enp0s8 08:00:27:86:7a:13

enp0s9 unknown

Enter management interface name:

enp0s8

Enter management address CIDR:192.168.101.2/24

Enter management gateway address [192.168.101.1]:

Enter System Controller subnet:192.168.204.0/24

Disabling non-management interfaces... DONE

Configuring management interface... DONE

RTNETLINK answers: File exists

Adding route to System Controller... DONE

At the SystemController:

1. Create a bootstrap-values.yml overrides file for the subcloud

For example:

system\_mode: duplex

name: "subcloud1"

description: "Ottawa Site"

location: "YOW"

management\_subnet: 192.168.101.0/24

management\_start\_address: 192.168.101.2

management\_end\_address: 192.168.101.50

management\_gateway\_address: 192.168.101.1

external\_oam\_subnet: 10.10.10.0/24

external\_oam\_gateway\_address: 10.10.10.1

external\_oam\_floating\_address: 10.10.10.12

systemcontroller\_gateway\_address: 192.168.204.101

1. Add the subcloud using dcmanager subcloud add CLI command

**dcmanager subcloud add --bootstrap-ip <ip\_address> --bootstrap-values <config\_file>**

Where:

ip\_address is the bootstrap\_ip set earlier on the subcloud

config\_file is the Ansible override configuration file, bootstrap-values.yml, created earlier in step 1 above.

You will be prompted for the Linux password of the subcloud.

This command will take 5- 10 minutes to complete. You can monitor the progress of the subcloud bootstrap through logs:

tail –f /var/log/dcmanager/< subcloud name>\_bootstrap\_<time stamp>.log

1. Confirm that the subcloud was deployed successfully

 dcmanager subcloud list

+----+-----------+------------+--------------+---------------+---------+

| id | name | management | availability | deploy status | sync

| +----+-----------+------------+--------------+---------------+---------+

| 1 | subcloud1 | unmanaged | online | complete | unknown |

+----+-----------+------------+--------------+---------------+---------+

1. Continue provisioning the subcloud system as required using the StarlingX R3.0 Installation procedures and starting from the ‘Configure controller-0’ step.

For AIO-Simplex:

<https://docs.starlingx.io/deploy_install_guides/r3_release/bare_metal/aio_simplex.html>

For AIO-Duplex:

<https://docs.starlingx.io/deploy_install_guides/r3_release/bare_metal/aio_duplex.html>

For Standard with controller storage:

<https://docs.starlingx.io/deploy_install_guides/r3_release/bare_metal/controller_storage.html>

For Standard with dedicated storage nodes:

<https://docs.starlingx.io/deploy_install_guides/r3_release/bare_metal/dedicated_storage.html>

1. Add routes from the subcloud to the controller management network.

system host-route-add <host id> <mgmt. interface> <system controller mgmt. subnet> <prefix> <subcloud mgmt. gateway ip>

For example:

system host-route-add 1 enp0s8 192.168.204.0 24 192.168.101.1

Repeat this step for each controller of the subcloud