THE OPENSTACK KICKSTART GUIDE FOR THE DSS 9000 (MAAS EDITION)

Transforming management for scale out infrastructure

ABSTRACT / SCOPE

This document intends to demonstrate how Dell EMC ® has greatly simplified the management of OpenStack cloud platforms by integrating Intel ® Rack Scale Design (RSD) and MAAS (Metal as a Service). To that end, the Dell EMC Extreme Scale Infrastructure (ESI) group provides integration software that combines the Intel Pod Manager (PODM) component of RSD, and MAAS to provide an easier and seamless “kickstart” for workload deployment. An example reference implementation and step-by-step description are included to illustrate how an administrator uses PODM and MAAS to quickly and easily initialize, provision and configure workloads in an OpenStack cluster environment, on the DSS 9000 infrastructure solution.

April, 2017

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The increasingly complex nature of today's innovative software solutions require more and more IT resources, which in turn accentuate the need for more efficient IT solutions. The resulting virtualization of all aspects of IT has resulted in truly software defined data centers (SDDC), where all the resources needed for a particular workload can be (virtually) mustered from anywhere in the physical datacenter - across machines, operating systems, vendor brands – transparently to the application. The applications, users, and business owners now operate in a world that is simpler, faster and more efficient for them. Thanks to these innovations they can readily scale up to thousands of nodes.

But now, transformation is needed in the management of large scale IT environments - not just hyperscale data centers, but growing scale customers as well. To justify IT outlays these companies need to "get to live" or "get to money" more rapidly than has been available in the past. They need a simpler, faster way to deploy and manage massively-scaled IT environments.

Dell EMC ® and Intel have collaborated using Intel ® Rack Scale Design and OpenStack to allow for more simplified and automated management of scale-out resources on the latest rack scale solution from Dell EMC – the DSS 9000. This paper details the various software components involved, the new orchestration capabilities they bring to rack scale data centers, and how to implement the kickstart management process.

Audience

This paper is intended for storage architects, engineers, and IT administrators who want to understand how to use these new tools to manage scale out infrastructure more easily and to understand the interaction of the various components.

1 Overview of DSS 9000 management components

To understand how the kickstart process works, it is helpful to have an understanding of each of the hardware and software components involved in cloud management on the DSS 9000. Those are:

- DSS 9000 hardware - rack, compute, storage with single point of management
- Top of Rack switch - networking component
- Utility node
- Intel Rack Scale Design (RSD) - architecture and APIs that enable disaggregation
- Redfish API - Open standard management APIs – the basis of much integration
- Intel POD Manager - management software that is part of RSD
- MaaS – an open source bare metal management and provisioning tool
- Dell EMC integration components – software that transparently integrates management components

The following sections describe these components in more detail.

1.1 DSS 9000 hardware

The DSS 9000 is a rack level solution comprised of pools of compute, storage and networking resources which are managed through a single point of rack management. Each DSS 9000 allows combinations of ompute and storage sleds, along with shared power, cooling and networking to provide maximum configuration flexibility.
Each DSS 9000 also has a hardware controller (rack manager) that serves as the single point of management for the resources in the rack. This rack manager conforms to the Distributed Management Task Force's (DMFT) Redfish API and the Redfish extensions for Intel POD Manager, enabling easier integration of the management components. (http://www.dmtf.org)

1.2 Top of Rack (ToR) network switch

A DSS 9000 rack solution is agnostic to the Top of Rack switch a customer may choose. The reference implementation this document uses to illustrate the kickstart process is the S4048-ON. It is a x48 SFP+ port network switch running Cumulus Linux OS 2.5.X ,with its ports configured to allow MAAS and PODM to connect to the same switch. PODM traffic is configured to use a VLAN. More details are provided in the switch configuration section.

1.3 Utility node

The utility node is an additional server, connected to the DSS 9000 through the ToR, that is used as a management station and runs the DSS 9000 management software. This reference implementation uses a PowerEdge R430 server.

1.4 Intel Rack Scale Design

RSD is a software-defined architecture that allows IT administrators to consider compute, storage, and networking as disaggregated resources that can be assembled dynamically, as needed, to meet various demands in a data center/cloud.

Disaggregation, in addition to allowing hardware refresh at different rates for each of the storage, compute, and networking resources, supports more efficient resource utilization based on right-sized assignments. Imagine a cloud that grows and shrinks to meet optimum utilization by virtue of being comprised of racks of resources which allows dynamic assignment and release, where one might assemble a system of resources where some nodes have copious storage and others provide pure compute horsepower.

To bring such a vision to light, the industry collaborated to define and standardize a RESTful set of APIs – DMTF's Scalable Platform Management Forum (SPMF) specification is called Redfish (described in more detail in the following section). As part of RSD, Intel also extended the Redfish APIs to allow support of

Dell - Internal Use - Confidential

managing node power control, discovering hardware capabilities, and collecting advanced telemetry information.

Dell EMC takes advantage of these open standards to provide transparent integration of the various management components using the kickstart integration software.

1.5 Redfish API

The Redfish API is an infrastructure management schema and API standard defined by the Distributed Management Task Force (DMTF). Redfish has the following characteristics:

- RESTFul Interface over HTTP in JSON format based on Odata v4
- Useable by client applications and browser-based GUIs
- A Secure interfaces over HTTPS to replace IPMI-over-LAN
- Multi-Node capable replacement for IPMI
- Example Python code to retrieve serial number:
  - rawData = urllib.urlopen('http://192.168.1.117/redfish/v1/Systems/1')
  - jsonData = json.loads(rawData)
  - Print(jsonData["SerialNumber"]
  - Output: 1A87C4442K

The Redfish API is an essential element of the overall integration of management software in the DSS 9000 solution, allowing the various components to interact seamlessly.

1.6 Intel POD Manager

PODM is a software component of RSD that supports the assembling and releasing of nodes. POD Manager acts as a resource manager, for disaggregated pools of hardware resources, by communicating using standard Redfish APIs to hardware-aware software within the DSS 9000.

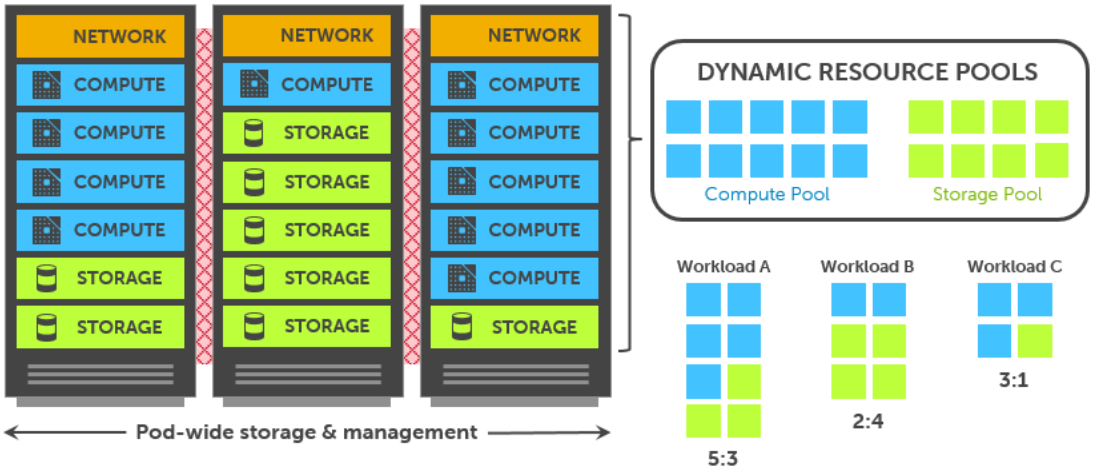


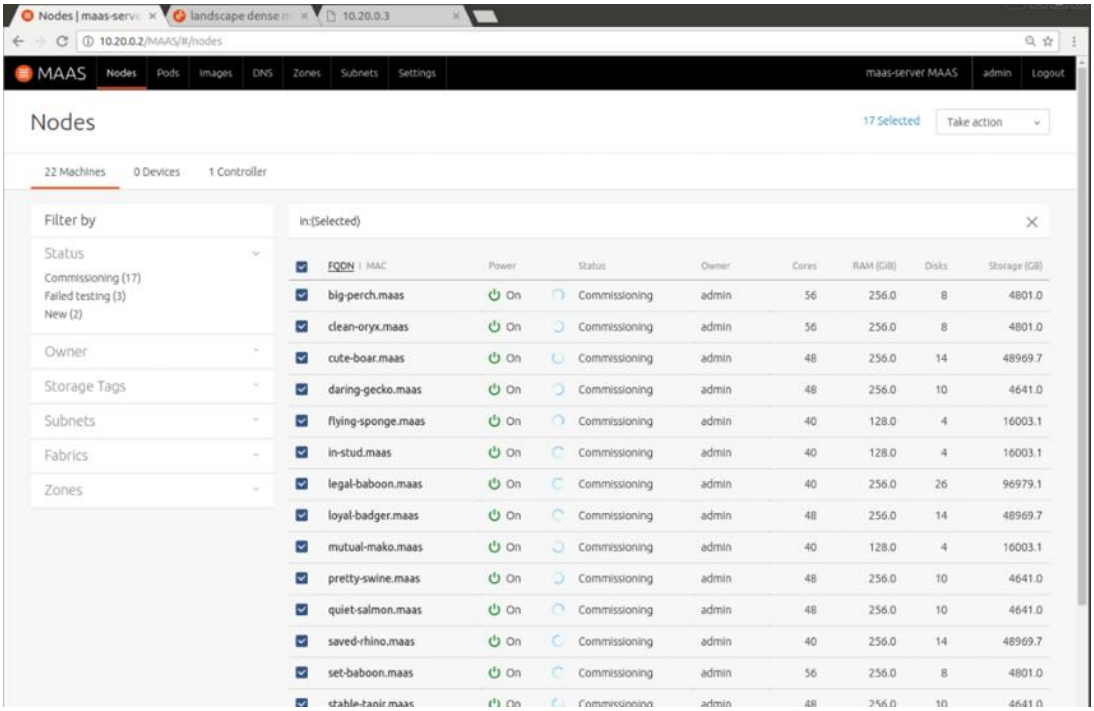
Figure 1: HW resources from 3 racks are virtually pooled and allocated to 3 workloads

1.7 MAAS

MAAS (*Metal As A Service) is a* bare metal management and provisioning tool. It lets you treat physical servers like virtual machines (instances) in the cloud. Rather than having to manage each server individually, MAAS turns your bare metal into an elastic cloud-like resource.

Dell - Internal Use - Confidential

Figure 2: OpenStack MAAS Web UI (cropped)



MAAS provides management of a large number of physical machines by creating a single resource pool out of them. Participating machines can then be provisioned automatically and used as normal. When those machines are no longer required they are "released" back into the pool. MAAS integrates all the tools you require in one smooth experience. It includes:

- A user-friendly web UI
- Full API/CLI support
- High availability (optional)
- IPv6 support
- Open source IP address management (IPAM)
- Ubuntu, CentOS, Windows, RHEL and SUSE installation support
- inventory of components
- DHCP and DNS for other devices on the network
- VLAN and fabric support
- NTP for the entire infrastructure

You can find more details about MAAS at https://docs.ubuntu.com/maas/

1.8 Dell EMC kickstart integration components

Dell EMC has created integration software that also implements the Redfish/RSD APIs and works with MAAS and RSD, allowing you to easily and quickly construct nodes to deploy an OpenStack Cloud on the DSS 9000. With these components working together, transparently, you can grow and shrink the cloud to meet utilization needs as they arise. How to install, configure and use these components is described in later sections.

Dell - Internal Use - Confidential

2 Configuration and preparation of the reference implementation

The following sections describe the different preparations you need to take to ensure a smooth OpenStack implementation on the DSS 9000.

2.1 The DSS 9000 reference implementation

The DSS 9000 rack scale solution can contain a variable number of compute and storage "blocks" and implements shared power, cooling, and networking infrastructure. Below are the hardware specifications for configurations used on this DSS 9000 reference implementation:

	DSS 9500 (full-width)	**DSS 9520 (half-width)**
Processors	2x Intel Xeon E5-2600v4	2x Intel Xeon E5-2600v4
TDP Max	135W	135W
Chipset	Intel C610	Intel C610
# DIMMs	Up to 16	Up to 16
DIMM Type	DDR4	DDR4
Hot-swap HDDs/SSDs	Up to12x 3.5" SAS/SATA	Up to 2x 2.5" SATA (on-board controller)
Non-HS HDDs/SSDs	Up to 4x 2.5" SAS*/SATA	Up 4x 3.5" or 8x 2.5" SAS/SATA, Up to 2x 2.5" NVMe* SSD
PCIe Gen3 slots	x16 half-height, half-length x8 mezzanine	x16 half-height, half-length x8 mezzanine
LOM	Dual 10GbE SFP+	Dual 10GbE SFP+
Management	Dedicated RJ45, BMC, iDRAC, IPMI 2.0/DCMI 1.5, Redfish	Dedicated RJ45, BMC, iDRAC, IPMI 2.0/DCMI 1.5, Redfish

2.2 Management Controller (MC) configuration

The Management Controller (MC) on the DSS 9000 must have firmware 3.30 or newer and the management interface must be set to DHCP. The hostname on the MC must include the string "*psme*", for example: *psme-mc-hostname*. PODM uses DHCP and the hostname to discover Pooled System Management Engines (PSME). The MC can be configured via serial or Ethernet by logging into the MC and running the MC CLI. Please refer to the MC CLI user guide for configuration related issues. It is also important to enable RSD features in the MC. Usually this is done by setting properties in the MC configuration file found at: */opt/dell/mc/conf/Redfish.conf.*

Below are the properties that are required to be set in order to enable RSD features:

R2Apis=true
R2ApiDataSource=Dynamic
R2DynApiCacheUpdate=AutoAndPoll
R2DynApiCachPollInterval=20

Once changes are made to the Redfish.conf file the MC must be rebooted for the changes to take effect. You can simply reboot the MC by typing "reboot" at the prompt

Dell - Internal Use - Confidential

2.3 Network configuration

In the reference implementation for this paper, each node in the DSS 9000 rack scale solution has two onboard 10Gb LOMs which are connected to the S4040-ON switch. All ports on the switch are setup in trunk mode except for two ports that are used to connect to the Management Controller and Management port of the switch. Those ports are configured to be in access mode. Below is a sample /etc/network/interfaces file that you can use as a reference to setup your network switch. In the below example port 47 (swp47) is used to connect to the management controller and port 48 (swp48) is used to connect to the management port of the switch.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5), ifup(8)
#
# Please see /usr/share/doc/python-ifupdown2/examples/ for examples
#
#

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp


auto br0
iface br0
bridge-ports glob swp1-54
bridge-stp on
bridge-vlan-aware yes
bridge-vids  4094


auto swp47
iface swp47
      link-speed 1000
      link-autoneg on
  bridge-access 4094
  bridge-pvid   4094


auto swp48
iface swp48
      link-speed 1000
      link-autoneg on
  bridge-access 4094
  bridge-pvid   4094
```


Dell - Internal Use - Confidential

2.4 Utility node configuration

The utility node is used as the management station and for this reference implementation, a PowerEdge R430 is used. The utility node is required to run Ubuntu 16.04 and needs to have Virtual Box 5.x.x installed in order to host PODM. PODM is provided in OVA format that can be imported into VirtualBox.

For networking, the utility node is required to have two network interfaces at a minimum; one NIC connected to the outside internet and another NIC is used for MAAS PXE + PODM. For illustration, refer to the two interfaces as eth0 and eth1. Since eth1 is used for both MAAS and PODM, a VLAN needs to be created on eth1. Below is an example of an /etc/network/interfaces configuration to setup a vlan on eth1.

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 10.20.0.1
netmask 255.255.252.0

auto eth1.4094
iface eth1.4094 inet static
      address 10.3.0.2
      netmask 255.255.252.0
      gateway 10.3.0.2
      vlan-raw-device e
```

Please note that there are two networks in the above configuration 1) 10.20.0.0/22 and 2) 10.3.0.0/22. The 10.20.x.x network is used for MAAS and the 10.3.x.x network is used for PODM.

Once the above network configuration is applied, install the vlan package using "sudo apt-get install vlan".

2.5 MAAS installation and configuration

MAAS is simple to use and configure. To install MAAS on the utility node, simply run:

"apt-get install maas".

Once fully installed, login to the WEB UI by running your favorite browser and navigate to:

http://10.20.0.1/MAAS

The first time you bring up the MAAS UI, you will be prompted to create an admin login. You can create an admin login account by running the following in a terminal window:

sudo maas createadmin

Now that you created an admin account, you can login to the WebUI and navigate through the different options. Please refer to the online MAAS documentation at:

https://docs.ubuntu.com/maas/

Also, it is important to enable DHCP on the 10.20.0.0/22 subnet. You can do that by clicking on the subnet TAB and selecting your subnet.

Dell - Internal Use - Confidential

2.6 Kickstart integration software installation and configuration

The Dell EMC kickstart integration components consist of several tools that are installed on the utility node that can be used to get your systems up and running quickly, and simplify deployment of resources.

The integration components can be obtained from Dell EMC on request and are delivered as a single file named *kickstart.tar.gz* which has separate folders containing the following components:

- **Intel PODM virtual box VM**

  As part of the kickstart, Dell EMC has created an OVA file that enables you to easily import Intel PODM into VirtualBox. The VM you create during this step requires a single bridged interface to the PODM network. By default, the PODM interface is setup to a static IP 10.3.0.1. PODM also runs a DHCP server that will lease IPs to interfaces connected to the management network.

  Once PODM VM is started, you can login to the VM from the utility node by using SSH. The default username is “user” and default password is “password”

  If you are not able to login to the POMD VM, insure that you have a route from the utility node to the PODM VM over the 10.3.0.0/22 network.

  You can import PODM into virtualbox by running:

  ```
  sudo VBoxManage import <OVA filename>
  ```

  After you have successfully imported the OVA image, you need to attach the PODM VM to the management network eth1.4094. You can accomplish this by running:

  ```
  sudo VBoxManage modifyvm “PODM” –nic1 bridged – cableconnected1 on –bridgeadapter1 eth1.4094
  ```

  Finally, start the new VM by running:

  ```
  sudo VBoxManage startvm “PODM” –type headless
  ```

- **Kickstart management tools to be installed on utility node**

  The *podm_tool.py* tool allows you to interface with PODM to do things like power control or set PXE on multiple nodes. This tool also allows you to assemble nodes from a pool of resources.

  The *switch_tool.py* tool is used to configure and manage VLANs on an RSD enabled switch. For example, the S4048-ON running the network PSME under Cumulus can be used as an RSD-enabled switch.

  The *podm_maas.py* tool needs to run on the utility node where MAAS is installed. It is a service that runs in a loop that constantly polls PODM to discover newly assembled nodes and insure they can be added to MAAS and start commissioning. Commissioning is a step in MAAS that puts the nodes into “Ready” state where they can be used for deployment. Start this service by running:

  ```
  python podm_maas.py <interval in seconds>
  ```

  Note: This service requires that you have a network interface on the MAAS server that can reach PODM network. Usually this is the eth1.4094

Dell - Internal Use - Confidential

- **A customized commisioning script for MAAS to support the DSS 9000**

  This is a Linux shell script that that is imported into MAAS to perform custom commissioning. The commissioning script will add configuration tools into the MAAS commissioning image. These tools include *racadm, syscfg, perccli,* and *arcconf*, which are used to update BIOS/Firmware and configure RAID controllers. The commissioning script will run when MAAS initiates the commissioning process. In order for nodes to be used for deployment they must be first commissioned from within MAAS. Once the nodes are fully commissioned they will be transitioned to a "Ready" state.

- **Firmware/BIOS update tools.**

  You can also run firmware/BIOS update by simply running an ansible playbook. These tools run on the MAAS server to apply firmware and BIOS configuration to multiple nodes. Ansible must be installed on the MAAS server to run the ansible playbook. The playbook is invoked by running a script called "update_firmare.sh". This script initiates a firmware and BIOS update on all discovered nodes. There is also a configuration parameter file called "parameters" that describes the location/version/payload information for the firmware update process. To start the firmware/BIOS update process, you first edit the parameter file and then run the script *update_firmware.sh*

  The next page has an example excerpt from a text file for the configuration of a node.

Dell - Internal Use - Confidential

```
##

### Address of the repository where you will store your files.
### Supported protocols are HTTP, HTTPS, and FTP.
### Example: http://10.20.0.2:8080/dell/rom_file.rom
### Example: REPO_ADDRESS="http://10.20.0.2:8080";

REPO_ADDRESS="http://10.20.0.2:8080";


### Path to the repository where you will store your files.
### Example: http://10.20.0.2:8080/dell/rom_file.rom
### Example: REPO_PATH="dell";

REPO_PATH="dell";


### Current iDRAC version as of October 2016.
FIRMWARE_VERSION_IDRAC="2.40.40.40";
FIRMWARE_FILE_IDRAC="iDRAC-with-Lifecycle-
Controller_Firmware_2091K_LN_2.40.40.40_A00.BIN";

### Previous iDRAC version.
#FIRMWARE_VERSION_IDRAC="2.30.109.30";
#FIRMWARE_FILE_IDRAC="iDRAC-with-Lifecycle-
Controller_Firmware_8PG3M_LN_2.30.109.30_A00.BIN";


### Current BIOS version as of October 2016.
FIRMWARE_VERSION_BIOS="2.0.3";
FIRMWARE_FILE_BIOS="6400_BIOS_JRNDK_LN_2.0.3.BIN";

### Previous BIOS version.
#FIRMWARE_VERSION_BIOS="2.0.2";
#FIRMWARE_FILE_BIOS="6400_BIOS_JRNDK_LN_2.0.2.BIN";


### This script will automatically restart nodes if necessary as part of the
### firmware update process. However, if you would like to force all nodes to
### restart after the script runs then uncomment the following and set it to 1.
#POWERCYCLE=0;


### AVAGO MegaRAID 9361-4i, 9361-8i, 9380-4i4e, 9380-8e firmware.
        ## Available versions:
                # 24.15.0-0032
                # 24.15.0-0026
                # 24.15.0-0016
                # 24.12.0-0020

#FIRMWARE_VERSION_MEGARAID="24.15.0-0032";
#FIRMWARE_FILE_MEGARAID="mr3108fw.rom";


### Dell PERC H730, H830, FD33x series firmware.
        ## Available versions:
                # 25.4.1.0004
                # 25.4.0.0017
                # 25.3.0.0016
                # 25.2.2-0004

#FIRMWARE_VERSION_PERC="25.4.1.0004";
#FIRMWARE_FILE_PERC="SAS-RAID_Firmware_4CGCG_LN_25.4.1.0004_A07.BIN";
```

Once configured, the nodes are ready to deploy.

Dell - Internal Use - Confidential

The figure below illustrates how the utility node should be set up and how the different components interact.

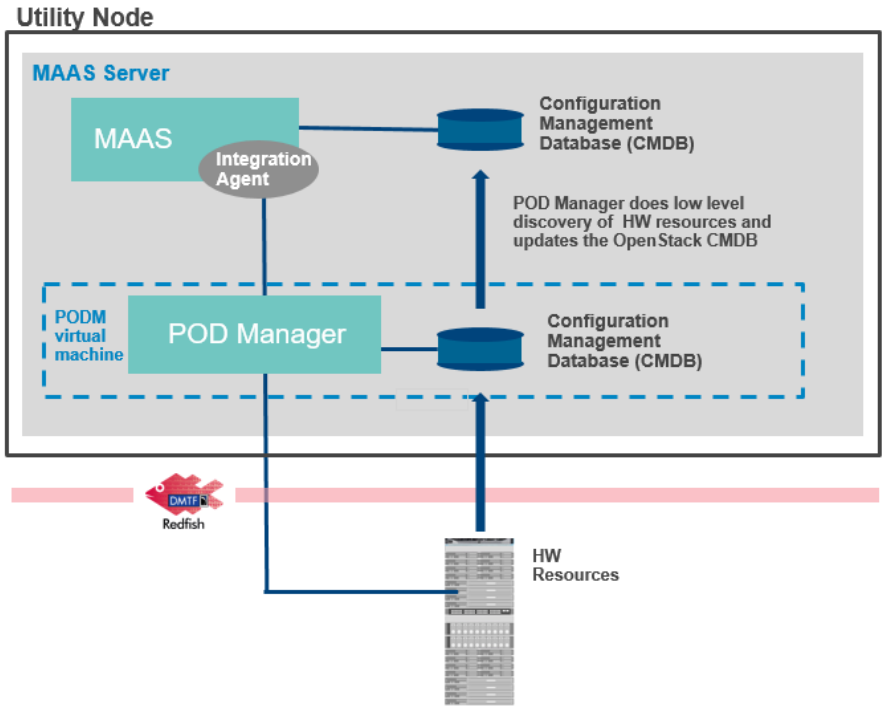


Figure 3: Management utility node configuration

You are now ready to begin allocating resources on the DSS 9000.

3 OpenStack cloud deployment: allocate, provision, deploy

Once you have done the set-up described in the installation and configuration section, allocating resources on your OpenStack cloud comes down to just a few simple (yet powerful) commands.

3.1 Allocate resources for a workload

Allocate a workload from PODM using the command line on the utility node, for example, to allocate a node made up of 3 servers each with specifically 2 CPUs and 6400 MB of memory, type at the python prompt:

```
podm_tool.py node allocate “Sample-Workload” 3  CPU=2  MEM=64000
```

3.2 Assemble the nodes

Once the 3 nodes are allocated, assemble them to make them ready to be used, using this command:

```
python podm_tool.py  node assemble 1-5
```

At this point, the PODM-MAAS agent will detect the newly allocated workload and automatically power on the nodes, if they are off. It will also automatically set the nodes to “boot to PXE” so all the new nodes can boot to MAAS’s PXE network.

Node locale information
Once the nodes are discovered by MAAS, their locale information will be displayed in the MAAS Web-UI.

Dell - Internal Use - Confidential

3.3 Commissioning Nodes

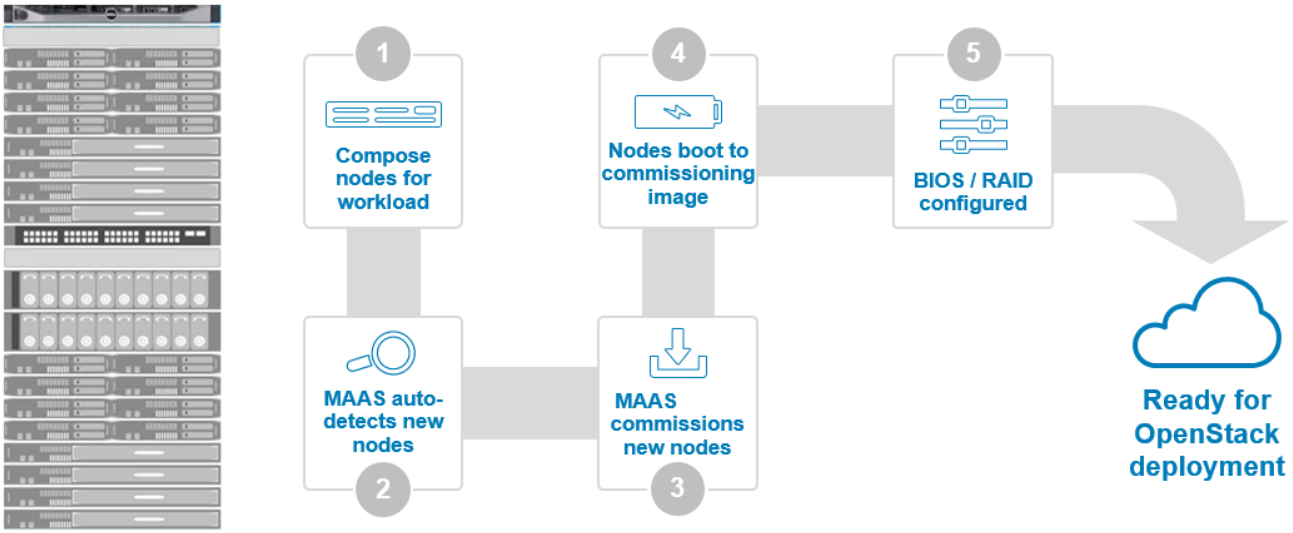
Once your nodes are allocated and assembled, MAAS will start commissioning the new nodes. This process will update BIOS/Firmware and configure RAID on the newly assembled nodes. Once the commissioning process is complete, the nodes will be ready for deployment.

3.4 Deploy the nodes into the OpenStack Cluster

You can use MAAS to deploy an operating systems or you can use it in conjunction with JUJU to deploy an OpenStack cluster.

4 Step by Step approach to dynamic workload assignment

Below is a flow chart illustration of the dynamic workload assignment process. This section provides a step-by step description of what you need to do to implement this process.



Prerequisite checklist

- The utility node must run Ubuntu 16.04 server
- The utility node needs to have Virtual Box 5.x.x installed
- MAAS 2.1 needs to be installed on the utility node. Please refer to installation instructions at:

  https://docs.ubuntu.com/maas/2.1/en/installconfig-package-install

- MAAS server needs to have PODM running as a VM in VirtualBox.
- MAAS needs to have a VLAN interface for PODM communication. Usually this is eth1.4094
- The Management Controller (MC) on the DSS 9000 must have firmware 3.30 or newer. The management interface must be set to DHCP
- The hostname on the MC and the network switch must include the string "*psme*" so that podm can discover them. For example: *psme-mc-hostname* and psme-cumulus

Dell - Internal Use - Confidential

Step-by-step tasks

1) Power on the DSS 9000.

2) The S4048-ON switch should be setup for PODM and MAAS netowrk. Please refer to networking configuration in this guide for instructions.

3) On a Utility Node (R430), you should have PODM running as a VM and the Kickstart integration components for PODM.

4) The utility node will have two networks (NIC1/NIC2). One connected to the outside network and the other is for MAAS plus PODM network

5) Login to the Utility node using your credential.

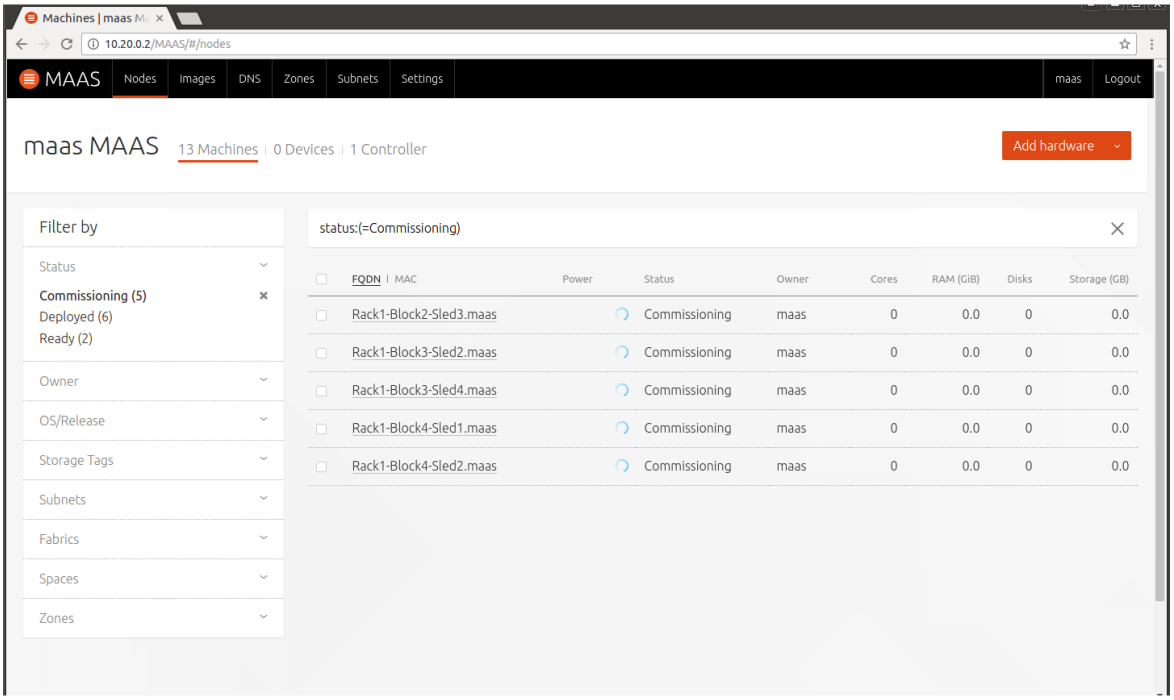
6) Using the Kickstart tools, allocate a workload from PODM. For example, you may have a need for 5 servers with specific CPU and memory requirements:

   python podm_tool.py node allocate "Example Workload" 5 CPU=16 MEM=64000

7) Once the allocation is complete, assemble the nodes to make the new nodes ready.

   python podm_tool.py node assemble 1-5

8) The PODM MAAS agent will detect the newly allocated workload. The agent will automatically bring those nodes into MAAS and list them under the nodes tab. The MAAS agent will insure that the new nodes are powered-on and start commissioning. (See image below.)



Dell - Internal Use - Confidential

9) Dell EMC provides a custom commissioning script that can be uploaded to MAAS using the MAAS WebUI under the settings TAB. The commissioning script will insure that every component in the system in updated to the latest firmware including BIOS/BMC/RAID

10) Once the nodes are added to MAAS, they will each be identified by LOCALE information. For example, each node will be identified as RackXX, BlockXX, SledXX". Where XX is the Rack, Block, and Sled location in the rack.

11) Now that the nodes are ready within MAAS, you can use those to nodes deploy an operating system.

For more information

For more information about the DSS 9000 or this OpenStack kickstart process, contact your Dell EMC sales representative or send email to: ESI@dell.com .

Dell - Internal Use - Confidential