Hewlett Packard Enterprise

HPE Reference Architecture for deploying Oracle 12c with HPE Synergy Image Streamer

Use cases for Oracle single-instance database and adding Oracle RAC node to existing cluster

Technical white paper

Contents

Executive summary

The demands of Oracle database implementations continue to escalate. Faster transaction processing speeds, scalable capacity, and increased flexibility are required to meet the needs of today's business. In the day-to-day management of Oracle database environments, administrators need to be able to quickly deploy new servers, easily update existing systems, and upgrade processing capabilities for scale-out performance. With traditional infrastructure, these activities are disruptive and time consuming.

HPE Synergy is an ideal platform for Oracle databases, offering fluid resource pools which can be customized for specific database needs. Oracle resources can be deployed rapidly through the software-defined intelligence embedded in the HPE Synergy Composer and HPE Synergy Image Streamer. An administrator can utilize HPE Synergy Image Streamer to develop a deployment plan to install and configure both the operating system and application software. A server profile defined in the HPE Synergy Composer can use that deployment plan to configure a new server in a matter of minutes, compared to hours or days utilizing traditional infrastructure.

This Reference Architecture demonstrates how to use the functionality provided by HPE Synergy Composer and HPE Synergy Image Streamer to quickly deploy two specific use cases: (1) an Oracle 12c single-instance database, and (2) adding a node to an existing Oracle RAC cluster. More specifically, it shows the following benefits of utilizing HPE Synergy for Oracle solutions:

- HPE Synergy Composer with embedded HPE OneView seamlessly manages the entire environment, including configuration of network resources required for Oracle compute nodes, creation and management of HPE 3PAR StoreServ volumes for the Oracle single-instance database and RAC nodes, and deploying the OS and application software on the compute nodes.
- Testing shows that HPE Synergy Composer plus HPE Synergy Image Streamer allows administrators to configure a new system for Oracle in less than three minutes, which is a significant reduction as compared to traditional deployment times of hours or days.

Target audience: This Hewlett Packard Enterprise white paper is designed for IT professionals who use, program, manage, or administer large databases that require high availability and high performance. Specifically, this information is intended for those who evaluate, recommend, or design new IT high performance architectures.

Document purpose: The purpose of this document is to describe a Reference Architecture, highlighting the usage of HPE Synergy Image Streamer to deploy Oracle database and Oracle RAC environments.

Solution overview

HPE Synergy enables IT organizations to accelerate application and service delivery through a single interface that composes physical and virtual compute, storage and fabric pools into any configuration for any application. Composable resources are provisioned together with their state (determined by variables such as BIOS settings, firmware, drivers, and protocols) and their OS and application image using repeatable templates. This is ideal for applications such as Oracle database, because it eliminates time-consuming provisioning processes.

The key components of this solution are the HPE Synergy Composer and HPE Synergy Image Streamer. The combination of these tools allow automating the customization of an OS image and configuration of Oracle database software, to quickly deploy a new environment.

HPE Synergy Composer

HPE Synergy Composer provides the enterprise-level management to compose and deploy system resources to your application needs. This management appliance uses software-defined intelligence with embedded HPE OneView to aggregate Compute, Storage and Fabric resources in a manner that scales to your application needs, instead of being restricted to the fixed ratios of traditional resource offerings.

HPE Synergy Image Streamer

HPE Synergy Image Streamer is a new approach to deployment and updates for composable infrastructure. This management appliance works with HPE Synergy Composer for fast software-defined control over physical compute modules with operating system and application provisioning. HPE Synergy Image Streamer enables true stateless computing combined with the capability for image lifecycle management. This management appliance rapidly deploys and updates infrastructure.

HPE Synergy Image Streamer adds a powerful dimension to 'infrastructure as code'—the ability to manage physical servers like virtual machines. In traditional environments, deploying an OS and applications or hypervisor is time consuming because it requires building or copying the software image onto individual servers, possibly requiring multiple reboot cycles. In HPE Synergy, the tight integration of HPE Synergy Image Streamer with HPE Synergy Composer enhances server profiles with images and personalities for true stateless operation.

HPE Synergy Composer, powered by HPE OneView, captures the physical state of the server in the server profile. HPE Synergy Image Streamer enhances this server profile (and its desired configuration) by capturing your golden image as the 'deployed software state' in the form of bootable image volumes. These enhanced server profiles and bootable OS plus application images are software structures ('infrastructure as code')—no compute module hardware is required for these operations. The bootable images are stored on redundant HPE Synergy Image Streamer appliances, and they are available for fast implementation onto multiple compute nodes at any time. This enables bare-metal compute modules to boot directly into a running OS with applications and multiple compute nodes to be quickly updated.

Figure 1 shows how HPE Synergy Composer and HPE Synergy Image Streamer manage a compute node via a server profile.

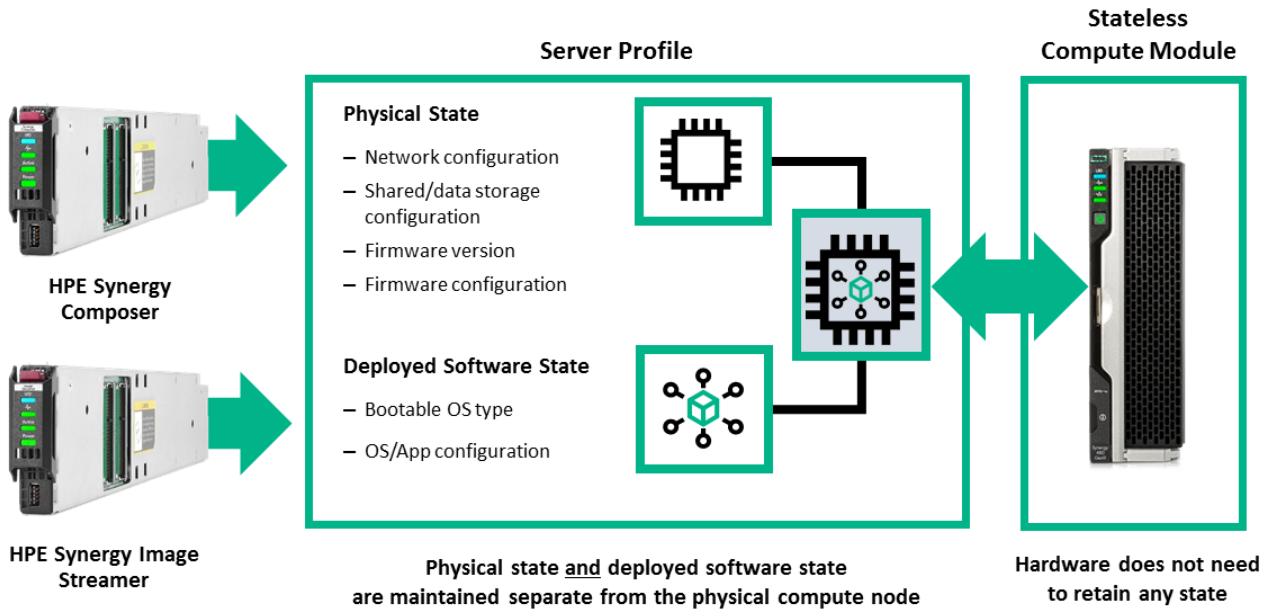

Figure 1. HPE Synergy Composer and HPE Synergy Image Streamer managing compute module with a server profile

HPE Synergy Image Streamer building blocks

HPE Synergy Image Streamer uses the following components for capture and deployment of images:

- **Plan script:** A guestfish[1] script used by OS build plans to personalize OS volumes based upon the values of custom attributes.
- **OS build plan:** A set of plan scripts used to modify the configuration of an OS volume during the deployment or capture process.
- **Golden image:** A generic format of an application and operating system image that can be customized for multiple deployments.
- **Deployment plan:** A combination of an OS build plan and golden image that is used by a server profile for the deployment of a server.

[1] For more information about the guestfish scripting language, see http://libguestfs.org/guestfish.1.html

Figure 2 shows the HPE Synergy Image Streamer Dashboard, which displays the resources available to create and modify OS images.

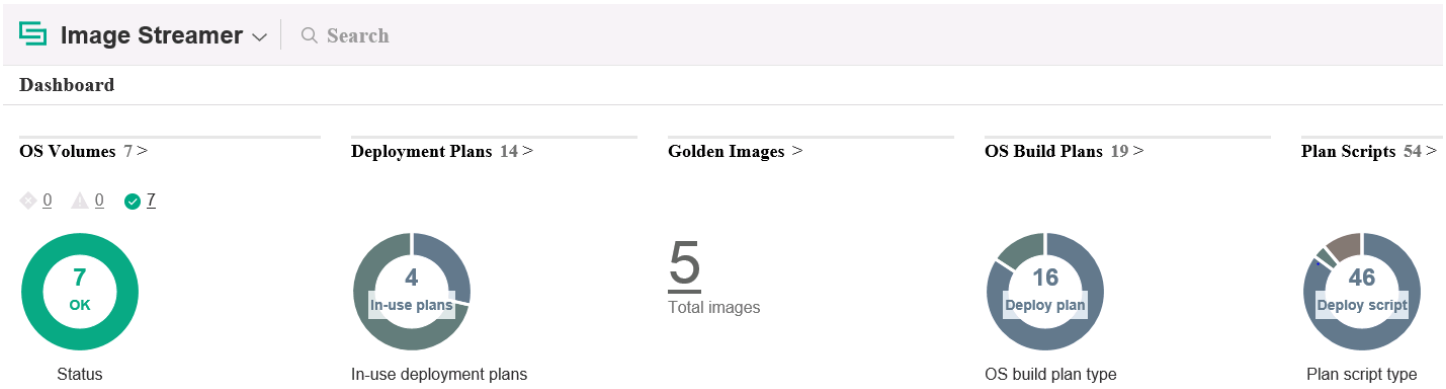

Figure 2. HPE Synergy Image Streamer Dashboard

HPE Synergy Image Streamer supports a variety of operations for flexibility in how you handle your images. For example, you can capture golden images for your use, import golden images from another location, or to modify some of your 'known good' images for re-use. This flexibility allows you to easily establish your desired images for use. A variety of images can be used on HPE Synergy Image Streamer. Reference implementations provide artifacts for recent versions of VMware® ESXi (5.0, 6.0, 6.5), and application images based on Red Hat Enterprise Linux® (RHEL 7.2, 7.3) and SUSE Linux® (SLES 12 SP1) using ext3 and ext4 file systems. You can also enable your own specific images and image types using the tools provided with HPE Synergy Image Streamer.

Solution components

The HPE Synergy components used in this solution included an HPE Synergy 12000 Frame, with one HPE Synergy Composer, one HPE Synergy Image Streamer, nine HPE Synergy compute modules, and a single Virtual Connect SE 40 Gb F8 Module for Synergy. The Virtual Connect Module for Synergy provides FCoE connectivity to external SAN storage. For this solution, HPE 3PAR StoreServ 8450 all-flash storage was used for the Oracle database table spaces, indexes and logs, plus the voting disk for Oracle RAC. Shared storage is required for Oracle RAC implementations, and the all-flash performance, as well as mission-critical resiliency of the HPE 3PAR StoreServ 8450 make it ideal for both Oracle single-instance and Oracle RAC environments.

Figure 3 shows the components that were used for this effort.

Note

While a single frame with a single Image Streamer was used for this testing, the recommended production configuration for high availability is a pair of Image Streamer appliances, and a pair of Virtual Connect SE 40 Gb F8 Modules for Synergy, which provides Active-Active HA for volume storage. HPE Synergy Image Streamer use in production environments requires a minimum of three Synergy Frames with two Virtual Connect SE 40 Gb F8 Modules for Synergy, two HPE Synergy Image Streamers, and two HPE Synergy Composers.

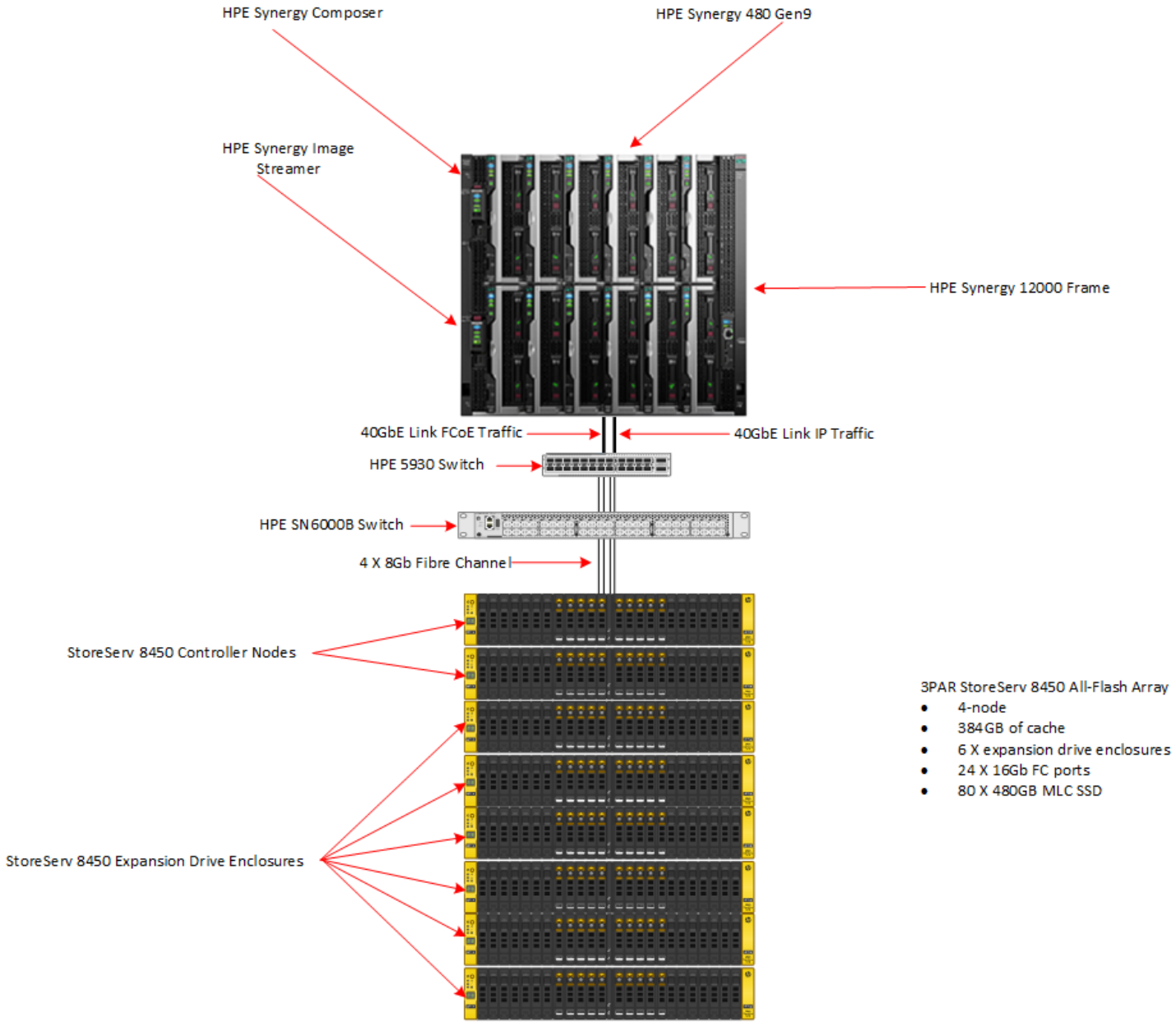

Figure 3. HPE Synergy and HPE 3PAR StoreServ configuration

Hardware

The HPE Synergy 12000 Frame had the following components:

- One HPE Synergy Composer
- One HPE Synergy Image Streamer
- 9 X HPE Synergy 480 Gen9 compute nodes
- One Virtual Connect SE 40 Gb F8 Module for Synergy

The HPE 3PAR StoreServ 8450 was configured as follows:

- 1 X 3PAR StoreServ 8450 all-flash array
  - 4-node
  - 384GiB of cache
  - 6 X expansion drive enclosures
  - 80 X 480GB MLC SSD
  - 24 X 16Gb Fibre Channel ports (running at 8Gb)

Software

- Red Hat Enterprise Linux version 7.2

Application software

- Oracle 12c (12.1.0.2.0) Enterprise Edition
- Oracle 12c Real Application Clusters

Best practices and configuration guidance for the solution

The high level steps for creating 1) a deployment plan, and 2) a server profile for an Oracle database environment are listed below. The details for implementing each of these steps are provided in the following sections.

Steps to develop an HPE Synergy Image Streamer deployment plan

1. Create golden image
2. Create plan scripts to customize golden image
3. Create OS build plan using plan scripts
4. Create deployment plan using OS build plan and golden image

Steps to deploy a compute node using the HPE Synergy Image Streamer deployment plan

1. Create required networks
2. Create 3PAR volumes required for Oracle data, logs and voting disk groups
3. Create server profile utilizing networks, storage and OS deployment plan

Develop an HPE Synergy Image Streamer deployment plan

Step 1: Create golden image

Use Case 1: Prepare RHEL 7 golden image for Oracle single-instance deployment

The following steps were required to create a RHEL 7 golden image for deploying an Oracle single-instance database.

Note that the Oracle binaries were located on the OS volume so that Image Streamer plan scripts could configure the software. Placing the binaries on the OS volume also makes it easy to apply Oracle patches. The Oracle Grid Infrastructure software is included to implement Oracle ASM Disk Groups for the Oracle database. Further details are provided in Appendix B and the Oracle database Installation Guide 12c Release 1 for Linux.

Note

An OS deployment plan must be in place to create an empty OS volume for step one below.

1. Create an HPE OneView server profile with an OS deployment plan that creates an empty OS volume of size 40 GB and assign the profile to a compute node.
2. Obtain the Driver Update Disk (DUD) for the QLogic iSCSI driver, since it is not included with RHEL 7.2.[2] This driver is required for the Image Streamer OS volume.
3. Install RHEL 7.2 on the empty volume, selecting the driver dud for the QLogic iSCSI driver during the installation, and adding the driver and ip=ibft parameters as install kernel options.
4. Create /etc/resolv.conf with address of DNS server.
5. Create oracle and grid user accounts and groups.
6. Create the directory /u01/app on the root partition.
7. Set the kernel parameters required for an Oracle database installation.
8. Set limits required by oracle and grid users in /etc/security/limits.d/oracle-limits.conf.
9. Create udev permissions file /etc/udev/rules.d/12-oracle-permissions.rules
10. Install all RHEL packages required for an Oracle database installation.
11. Unpack the zip files for the Grid Infrastructure kit in /home/grid and the Oracle database kit in /home/oracle.
12. Create the response files grid-single-instance.rsp in /home/grid and db-single-instance.rsp in /home/oracle (see Appendix C).

Note

Some of the settings listed above will be modified each time that the golden image is used to deploy a new OS volume via Image Streamer. These settings include the hostname, domain, some kernel parameters, and some account limits.

[2] This step is not needed when using RHEL 7.3, which includes the QLogic iSCSI driver in the distribution. The ip=ibft option for the install kernel listed in step 3 is still required.

After the OS has been customized according to the steps listed above, the Image Streamer "Create Golden Image" interface is used to create an image which is stored on the Image Streamer appliance, as shown in Figure 4. Do the following to create the golden image:

1. Shut down the OS.
2. Find the OS volume number in the HPE OneView server profile created in step one above. It is listed under the OS Deployment section of the profile.
3. On the Image Streamer Golden Image screen, select "Create golden image" and specify a name ("RHEL 7.2 for Oracle"), description, OS volume, and Capture OS build plan. Note that the Capture OS build plan, "HPE – Foundation 1.0 – capture OS Volume as is" is chosen in this case.

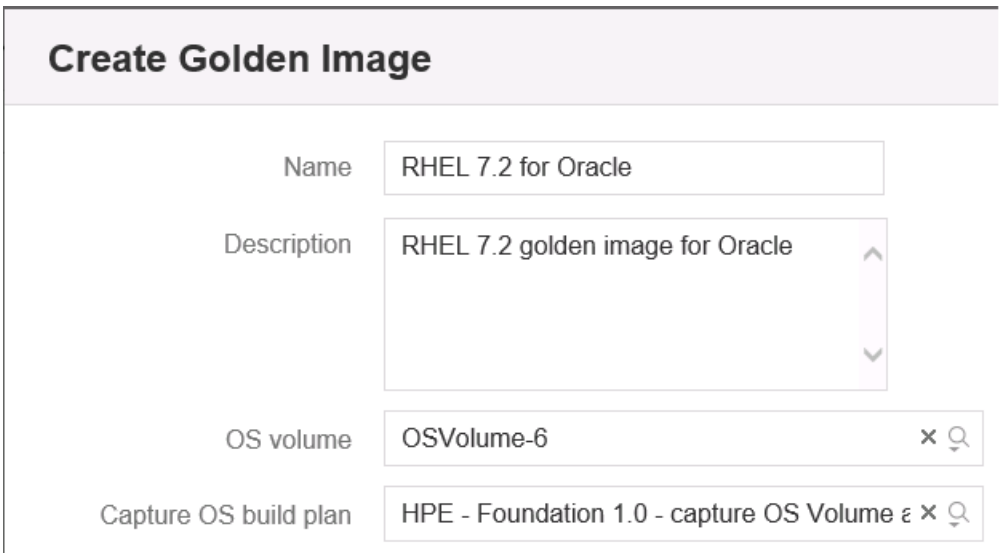

Figure 4. Create golden image

Use Case 2: Prepare RHEL 7 golden image for new Oracle RAC node

Note
The information provided here assumes that a two-node RAC environment already exists. The existing environment includes shared volumes for the Oracle data, logs and voting disk groups.

In addition to the steps listed above for the Oracle single-instance golden image, the following steps are required when creating a golden image for a new Oracle RAC node to be added to an existing cluster:

1. On the RHEL 7.2 volume, create /etc/multipath.conf with an alias for each shared volume.
2. Create a file in /etc/udev/rules.d to set ownership of the volumes.
3. Obtain and install packages for Oracle ASM libraries (oracleasmlib-2.0.12-1.el7.x86_64.rpm and oracleasm-support-2.1.8-3.el7.x86_64.rpm).
4. Configure oracleasm service.
5. Install cvuqdisk package from the rpm directory in the grid infrastructure kit.
6. Disable the avahi-daemon service.
7. Install and configure sshpass from https://sourceforge.net/projects/sshpass. This is used by the sshkey_config.sh scripts to provide the passwords required when configuring ssh keys.

After the OS has been customized according to the steps listed above, the Image Streamer "Create Golden Image" interface is used to create an image named "RHEL 7.2 Oracle RAC".

Steps 2 and 3: Create HPE Synergy Image Streamer plan scripts and OS build plans

Use Case 1: Create plan scripts and OS build plan for Oracle single instance deployment

Table 1 lists the HPE Synergy Image Streamer plan scripts that were developed to create an OS build plan named "RHEL 7 Deploy Oracle" to customize a RHEL 7 golden image for an Oracle database. The contents of the plan scripts are included in Appendix D. Since the server must be booted to complete the Oracle installation and configuration, the plan scripts create a service that is run the first time that the newly deployed server is booted. Note that the last plan script is used for debugging purposes. When the custom attribute "DeploymentTesting" is set to "Intentionally_Fail_OS_Deployment", a deployment log is created that allows the user to inspect the output of the plan scripts, without actually creating a new OS volume.

Table 1. Plan scripts for RHEL 7 OS deployment for Oracle database environment

Plan script name	Purpose
RHEL 7 – mount – deploy	Mount root filesystem
RHEL 7 – change root volume name	Use new hostname in root volume name
RHEL 7 – hostname configuration	Set hostname and add to /etc/hosts
RHEL 7 – update resolv.conf	Set search domain in resolv.conf
RHEL 7 – public network configuration	Configure network device for public network
RHEL 7 - Oracle – update kernel params	Configure kernel parameters required for Oracle
RHEL 7 – Oracle – update limits	Configure user limits for oracle account
RHEL 7 – Oracle – update udev rules	Set WWN for Oracle data disk in udev permissions file
RHEL 7 – Oracle - create firstboot service	Create service to install Oracle on initial boot
RHEL 7 – Oracle - create firstboot.sh script	Create shell script used by service
RHEL 7 – Oracle – create install script	Create script to perform silent install of Oracle
RHEL 7 – umount	Unmount root file system
HPE - Foundation 1.0 – attempt or fail deployment – v1.00	Allow failing deployment for debugging purposes

Use Case 2: Create plan scripts and OS build plan for new Oracle RAC node

Table 2 lists the HPE Synergy Image Streamer plan scripts that were developed to create an OS build plan named "RHEL 7 RAC Add Node". This build plan adds an Oracle RAC node to an existing cluster of two nodes. The contents of the plan scripts are included in Appendix E. Since the Oracle-provided addnode.sh script must be run from one of the existing RAC nodes, plan scripts are used to create services on the new node that are run the first time that the server is booted. These services update the /etc/hosts files on all three servers, and ssh keys are configured to allow all three nodes to access each other, and then the addnode.sh script is run from one of the existing nodes to complete the installation.

Note

All of these steps are completed with no interaction required by the user.

Table 2. Plan scripts for RHEL 7 OS deployment that adds Oracle RAC node

Plan script name	Purpose
RHEL 7 – mount – deploy	Mount root filesystem
RHEL 7 – change root volume name	Use new hostname in root volume name
RHEL 7 – RAC hostname configuration	Set hostname and add to /etc/hosts
RHEL 7 – update resolv.conf	Set search domain in resolv.conf
RHEL 7 – RAC public network configuration	Configure network device for public network
RHEL 7 - RAC private network configuration	Configure network device for RAC private network

Plan script name	Purpose
RHEL 7 - Oracle – update kernel params	Configure kernel parameters required for Oracle
RHEL 7 – Oracle – update limits	Configure user limits for oracle account
RHEL 7 – RAC create firstboot services	Create services to configure ssh keys and add RAC node
RHEL 7 - RAC create firstboot.sh scripts	Create shell scripts used by services
RHEL 7 – RAC grid create sshkey.config.sh	Create script to update hosts file and configure ssh keys for grid account
RHEL 7 – RAC oracle create sshkey.config.sh	Create script to configure ssh keys for oracle account
RHEL 7 – RAC create addRACnode.sh script	Create script to add new RAC node
RHEL 7 – umount	Unmount root file system
HPE - Foundation 1.0 – attempt or fail deployment – v1.00	Allow failing deployment for debugging purposes

Figure 5 shows the set of steps to configure a RHEL 7.2 installation with the components needed in the golden image for an Oracle RAC node (in the green boxes), followed by the steps conducted by the Image Streamer plan scripts to customize the golden image (in the blue boxes), followed by the set of scripts (in the orange boxes) that are run to add the new server to the RAC cluster when the server is booted for the first time.

Note

The first boot scripts are created by the Image Streamer plan scripts, and run automatically when the server is booted for the first time.

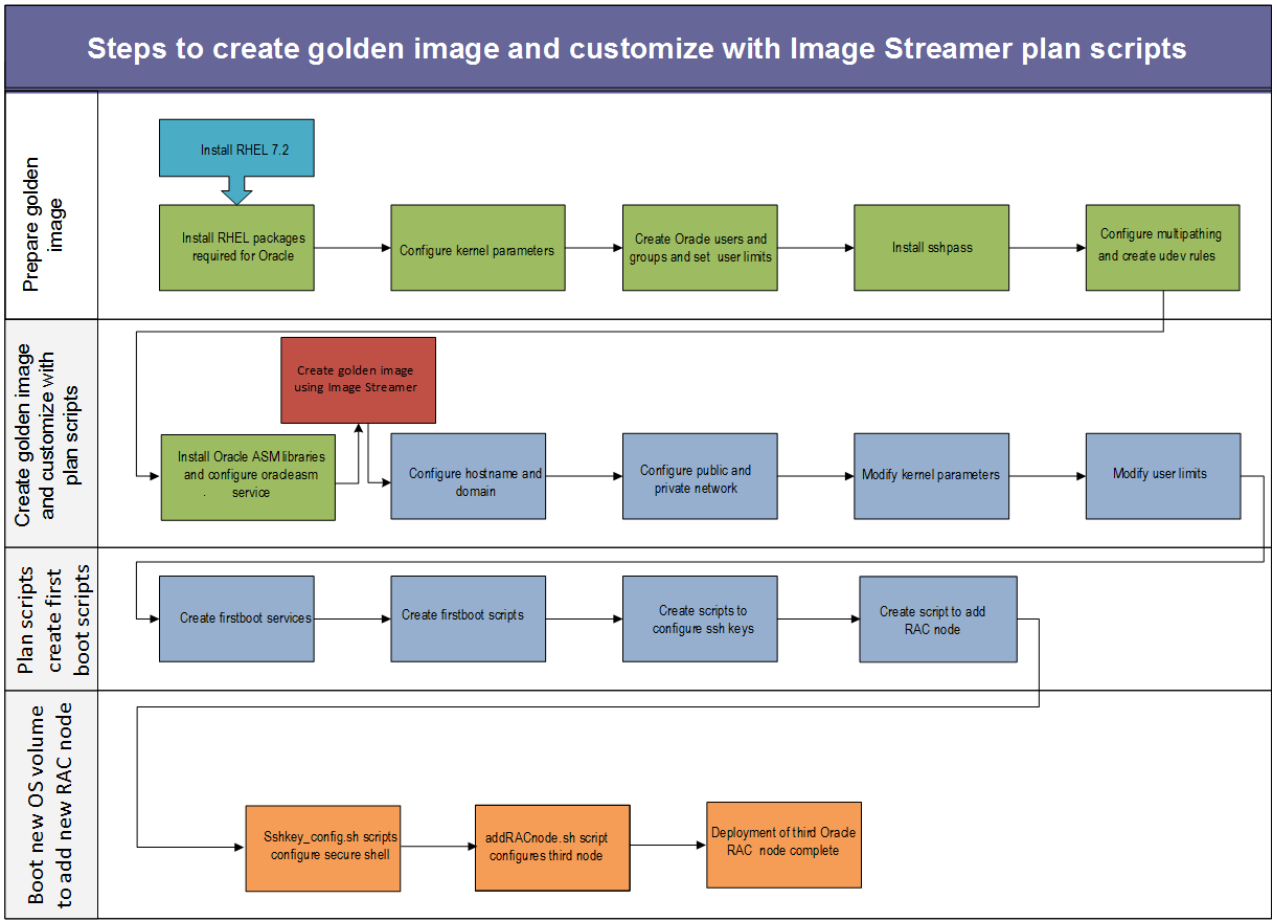

Figure 5. Steps to create golden image and customize with HPE Synergy Image Streamer plan scripts

Step 4: Create HPE Synergy Image Streamer deployment plans

An Image Streamer deployment plan is created by specifying an OS build plan and a golden image, using the "Create Deployment Plan" dialog box shown in figure 6. Note that the name of the OS build plan has not yet been filled in, because once it is entered, the list of custom attributes will be displayed, making it difficult to view the entire dialog box. The list of custom attributes and their default settings for the deployment plan is included in Appendix F.

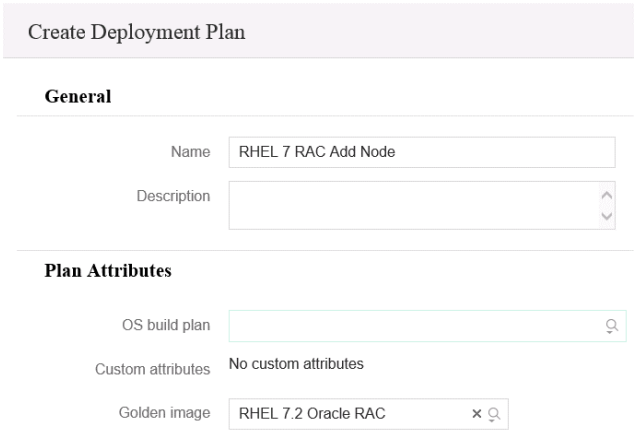

Figure 6. Create Deployment Plan

Deploy a compute node using HPE Synergy Image Streamer deployment plan

This section describes the steps required for deploying a Synergy compute node with an Image Streamer deployment plan. Prior to deploying a compute node, the Synergy environment must be configured with the networks and storage volumes required for an Oracle database or Oracle RAC implementation.

Step 1: Create required networks

The following networks are required in the Synergy environment, for usage in the server profiles that deploy Oracle single-instance database and Oracle RAC:

- Public network
- Private network for RAC interconnect traffic (heartbeat)
- FCoE network to connect to external storage (3PAR)
- Image Streamer deployment network

Table 3 summarizes the configuration for each network. All networks were created with a preferred bandwidth of 2.5 Gb/second and a maximum bandwidth of 20 Gb/second since they all shared a single Virtual Connect SE 40 Gb F8 Module. Note that the private network for the RAC heartbeat is not assigned a VLAN or uplink set, since communications are only needed between the Synergy compute modules.

Table 3. Networks for Oracle deployments

Network name	Type	VLAN	Subnet ID	Uplink set
Management_VLAN_145	Ethernet	145	10.145.40.0	VLAN_145
RAC_private	Ethernet	None	192.168.0.0	None
FCoE_EB04_1145	FCoE	1145	None	FCoE_1145
I3S_Deployment	Ethernet	100	10.100.0.0	I3S

Figure 7 shows the Connections section of the server profile for an Oracle RAC node. The server profile for an Oracle single-instance database server is similar, but does not use the RAC private network. Note that the Image Streamer deployment network is automatically added to the server profile when an OS deployment plan is selected, and it must be on port 3:1a. The iSCSI boot configuration is also automatically added to the profile.

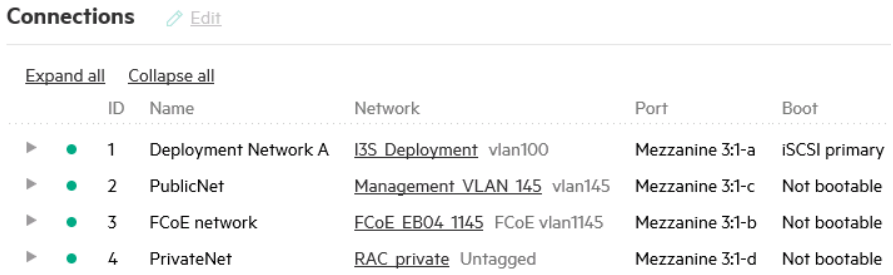

Figure 7. Connections section of server profile for Oracle RAC node

Step 2: Storage configuration

The HPE OneView Create Volume screen was used to configure 3PAR storage for the Oracle data, logs and RAC voting disks. For the RAC configuration, all volumes were created with the sharing attribute set to Shared, as shown in figure 8.

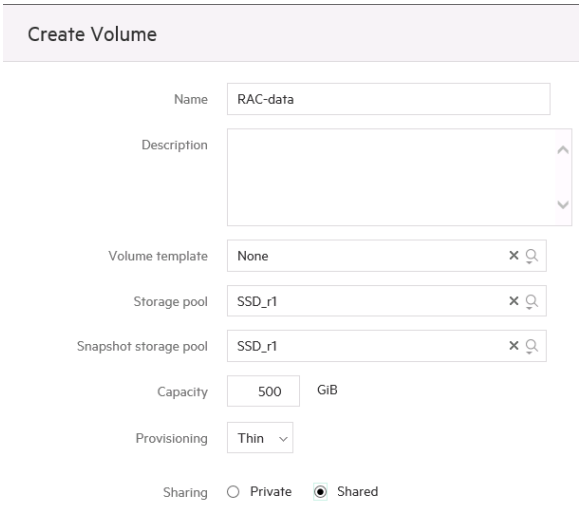

Figure 8. Create shared volume for Oracle RAC nodes

When creating the server profile for the new RAC node, all that was needed was to add the existing volumes to the server profile. Figure 9 shows the SAN Storage portion of the server profile for the new RAC node.

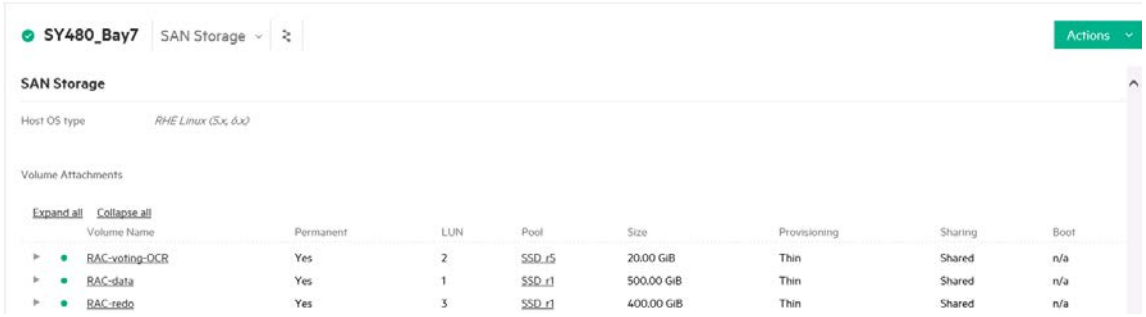

Figure 9 SAN Storage section of server profile for Oracle RAC node

Step 3: Create a server profile

After the required networks and storage volumes have been configured, a server profile can be created that utilizes these components along with an OS deployment plan that will configure the software for an Oracle 12c database or add a new node to an Oracle RAC cluster.

Note

Deploying a RAC node also requires that the DNS server for the environment be updated to include the hostname and all IP addresses required for the new server.

Figure 10 shows the Create Server Profile dialog box, with the OS Deployment section specifying the OS deployment plan "RHEL 7 RAC Add Node". At this point, the user may also modify the deployment settings listed for the deployment plan. This includes settings such as hostnames, IP addresses, kernel parameters and passwords. For a new RAC node, this includes IP addresses for the public network, the private network for RAC heartbeat, and the VIP address. The full set of attributes that can be customized for this deployment plan are listed in Appendix F. When the server profile is created, the plan scripts specified in the OS build plan are used to customize the new OS volume. After the profile creation has completed, the compute node can be powered on, and the steps required to add the new server to the RAC cluster are run by the firstboot services.

Note

The creation of a server profile also allows the user to specify firmware updates, boot settings, and BIOS settings to be applied to the server. These steps are not shown here as the focus is on OS deployment.

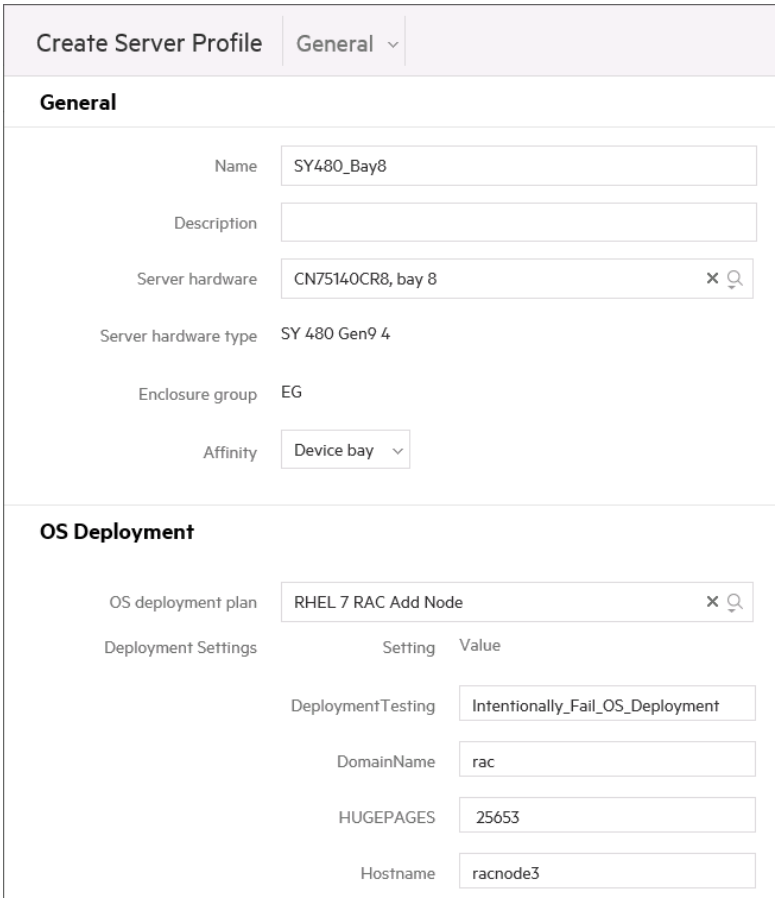

Figure 10. Create server profile

Capacity and sizing

Using HPE OneView server profiles with HPE Synergy Image Streamer deployment plans allows users to very quickly deploy a server for an Oracle single-instance database or a new node for an Oracle RAC environment. Figure 11 shows the Activity section of a server profile for deploying a compute node for a single-instance Oracle database. The total time to complete the deployment was 2 minutes and 10 seconds. This includes the time to customize the OS volume, using the plan scripts, at 14 seconds, and the time to apply the profile, at 1 minute and 51 seconds.

Note

The creation of a server profile also allows the user to specify firmware updates, boot settings, and BIOS settings to be applied to the compute node. These steps can add a significant and highly variable amount of time to the deployment. They are not included here as the focus is on OS deployment using Image Streamer. Firmware updates may be conducted when the compute node is added to the environment, avoiding the need to do this at OS deployment time. The iSCSI boot configuration for the HPE Synergy Image Streamer OS volume is set automatically in the server profile and is included in the timings shown here.

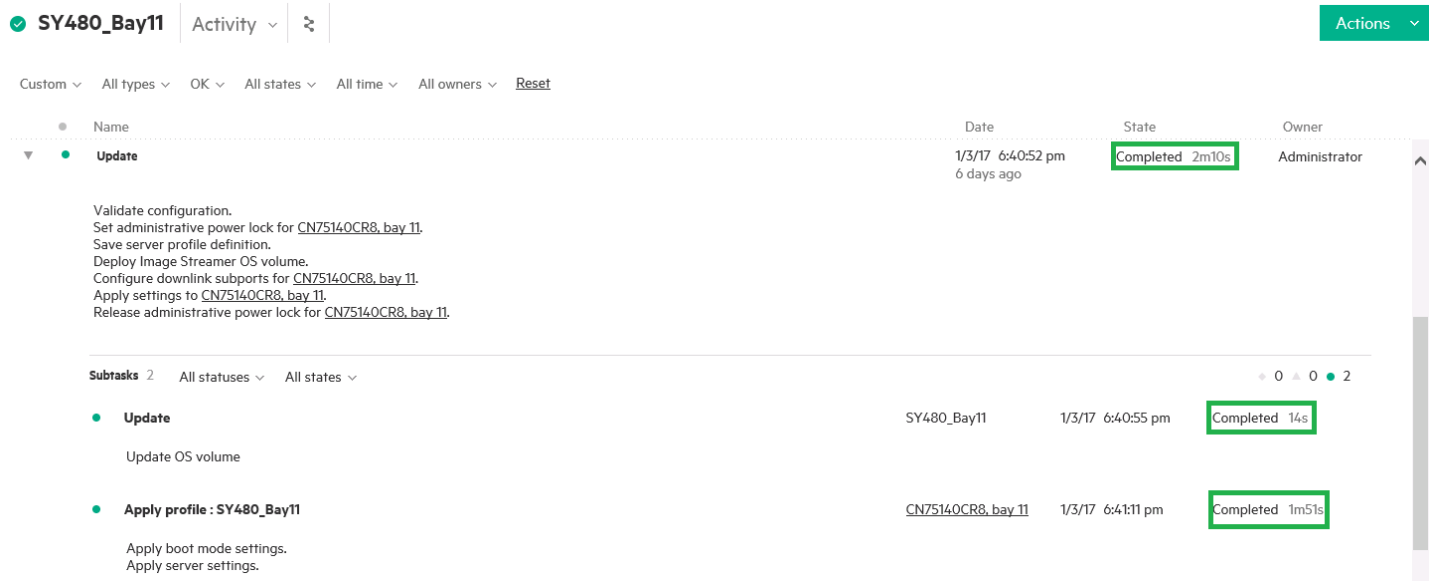

Figure 11. Activity log for deploying compute node for Oracle single-instance database

Figure 12 shows the Activity section of a server profile for deploying a new RAC node. The total time to complete the deployment was 2 minutes and 14 seconds. The time required to customize the OS volume (i.e. apply the plan scripts) was a mere 15 seconds. Applying the profile to the server, including the boot mode settings and server settings took 1 minute and 52 seconds.

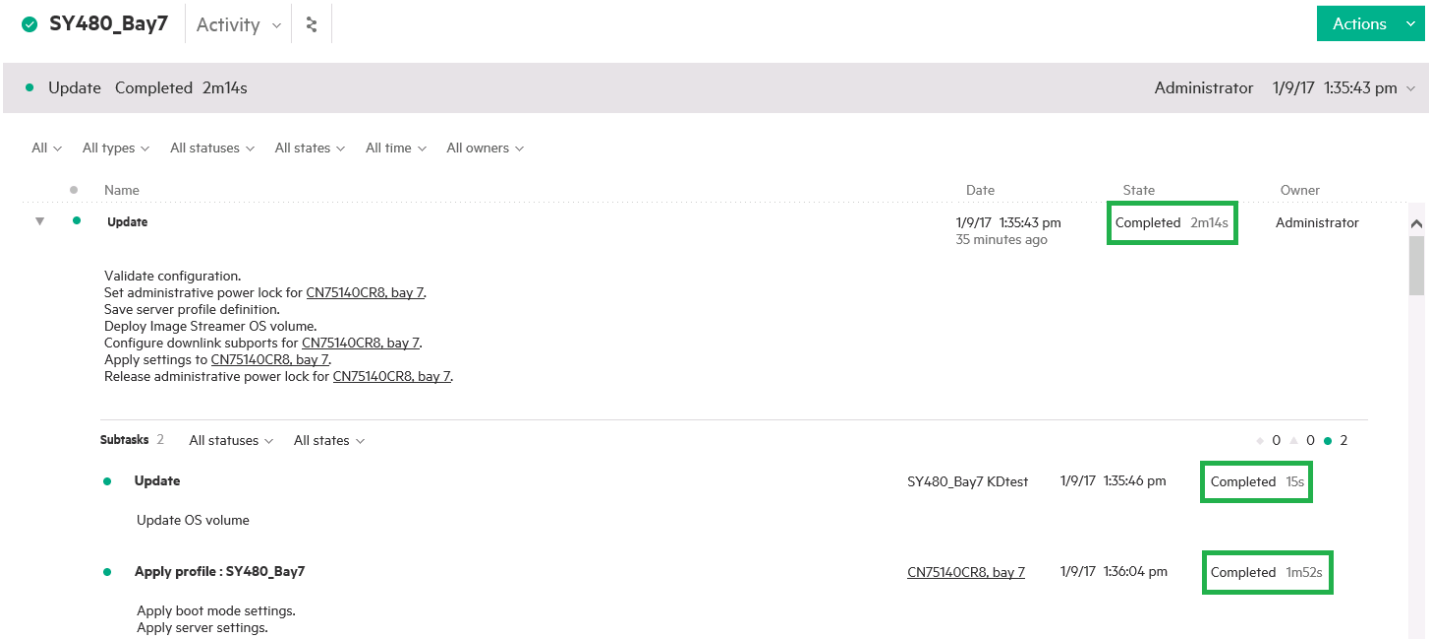

Figure 12. Activity log for deploying new Oracle RAC node

Summary

This Reference Architecture demonstrates how HPE Synergy enables Oracle database administrators to accelerate Oracle 12c database deployment and easily update their environments. HPE Synergy Composer and HPE Synergy Image Streamer can be utilized to create deployment plans to install and configure single-instance databases as well as expand capacity for Oracle RAC clusters. The fluid resource pools and software-defined intelligence of HPE Synergy allow administrators to rapidly compose any configuration required, reducing deployment time from hours or days down to minutes. More specifically, this Reference Architecture shows the following benefits of utilizing HPE Synergy for Oracle solutions.

- HPE Synergy Composer with embedded HPE OneView seamlessly manages the entire environment, including configuration of network resources required for Oracle compute nodes, creation and management of HPE 3PAR StoreServ volumes for the Oracle single-instance database and RAC nodes, and deploying the OS and application software on the compute nodes.
- Testing shows that HPE Synergy Composer plus HPE Synergy Image Streamer allows administrators to configure a new system for Oracle in less than three minutes, which is a significant reduction as compared to traditional deployment times of hours or days.

This Reference Architecture describes solution testing performed in December 2016.

Implementing a proof-of-concept

As a matter of best practice for all deployments, Hewlett Packard Enterprise recommends implementing a proof-of-concept using a test environment that matches as closely as possible the planned production environment. In this way, appropriate performance and scalability characterizations can be obtained. For help with a proof-of-concept, contact an HPE Services representative (hpe.com/us/en/services/consulting.html) or your HPE partner.

Appendix A: Bill of materials

The following BOMs contain electronic license to use (E-LTU) parts. Electronic software license delivery is now available in most countries. HPE recommends purchasing electronic products over physical products (when available) for faster delivery and for the convenience of not tracking and managing confidential paper licenses. For more information, please contact your reseller or an HPE representative.

Note

Part numbers are at time of publication/testing and subject to change. The bill of materials does not include complete support options or other rack and power requirements. If you have questions regarding ordering, please consult with your HPE Reseller or HPE Sales Representative for more details. hpe.com/us/en/services/consulting.html

Table 4. Bill of materials for HPE Synergy solution with HPE 3PAR StoreServ 8450 all-flash array

Qty	Part Number	Description
		Rack and Network Infrastructure
1	BW908A	HPE 642 1200mm Shock Intelligent Rack
1	BW909A	HPE 42U 1200mm Side Panel Kit
1	BW930A	HPE Air Flow Optimization Kit
3	BW928A	HPE 10pk 1U Universal Filler Panel
		HPE Synergy Frame Components
1	797740-B21	HPE Synergy 12000 Configure-to-order Frame with 1x Frame Link Module 10x Fans
1	804942-B21	HPE Synergy Frame Link Module
1	798096-B21	HPE 6X 2650W AC Titanium Hot Plug FIO Power Supply Kit
1	804353-B21	HPE Synergy Composer
1	804937-B21	HPE Synergy Image Streamer
1	804942-B21	HPE Synergy Frame Link Module
1	794502-B21	HPE Virtual Connect SE 40Gb F8 Module for HPE Synergy
		HPE Synergy Compute Module Components
9	732350-B21	HPE Synergy 480 Gen9 Configure-to-order Compute Module
9	826985-L21	HPE Synergy 480 Gen9 Intel® Xeon® E5-2697A v4 FIO Processor Kit
9	826985-B21	HPE Synergy 480 Gen9 Intel Xeon E5-2697A v4 Processor Kit
36	805349-B21	HPE 16GB (1x16GB) Single Rank x4 DDR4-2400 RDIMMs
9	777430-B21	HPE Synergy 3820C 10/20Gb Converged Network Adapter
		Storage
1	BW904A	HPE 42U 600X1075mm Enterprise Shock Rack
1	H6Z25A	HPE 3PAR StoreServ 8450 4N Stor Cent Base
4	H6Z00A	HPE 3PAR 8000 4-pt 16Gb FC Adapter
16	K2Q95A	HPE 3PAR 8000 480GB SFF SSD
1	L7C17A	HPE 3PAR 8450 OS Suite Base LTU
80	L7C18A	HPE 3PAR 8450 OS Suite Drive LTU
2	QR480B	HPE SN6000B 16Gb 48/48 FC Switch
96	QK724A	HPE B-series 16Gb SFP+SW XCVR
8	H6Z26A	HPE 3PAR 8000 SFF(2.5in) SAS Drive Encl
64	K2Q95A	HPE 3PAR 8000 480GB SFF SSD
1	K2R28A	HPE 3PAR StoreServ SPS Service Processor

Qty	Part Number	Description
1	TK808A	HPE Rack Front Door Cover Kit
80	QK735A	HPE Premier Flex LC/LC OM4 2f 15m Cbl
16	QK734A	HPE Premier Flex LC/LC OM4 2f 5m Cbl
4	H5M58A	HPE Basic 4.9kVA/L6-30P/C13/NA/J PDU
1	BW906A	HPE 42U 1075mm Side Panel Kit
1	BD362A	HPE 3PAR StoreServ Mgmt/Core SW Media
1	BD363A	HPE 3PAR OS Suite Latest Media
1	BD365A	HPE 3PAR SP SW Latest Media
1	TC472A	HPE Intelligent Inft Analyzer SW v2 LTU

Appendix B: RHEL 7 golden image settings

The following settings were used for the RHEL 7.2 golden images.

The /etc/sysctl.conf file included the following settings required for Oracle database installations. This file is edited by the plan script "RHEL 7 – Oracle – update kernel parameters" to modify the settings for shmmax and nr_hugepages which are both dependent upon the memory configuration of a server.

```
kernel.sem = 250 32000 100 128
kernel.shmall = 4294967295
kernel.shmmax = 53798230426
fs.file-max = 6815744
kernel.shmmni = 4096
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048586
vm.nr_hugepages = 25653
vm.hugetlb_shm_group = 507
kernel.numa_balancing = 0
```

The /etc/security/limits.d/oracle-limits.conf file included the following settings for the oracle and grid user accounts. This file is edited by the plan script "RHEL 7 – Oracle – update limits" to modify the setting for memlock, which is dependent upon the memory configuration of a server:

```
oracle   soft   nofile    1024
oracle   hard   nofile    65536
oracle   soft   nproc    16384
oracle   hard   nproc    16384
oracle   soft   stack    10240
oracle   hard   stack    32768
# set memlock to 90% of memory
oracle   hard   memlock   59104501
oracle   soft   memlock   59104501
grid     soft   nofile    1024
grid     hard   nofile    65536
grid     soft   nproc    16384
grid     hard   nproc    16384
grid     soft   stack    10240
grid     hard   stack    32768
# set memlock to 90% of memory
grid     hard   memlock 59104501
grid     soft   memlock 59104501
```

The /etc/multipath.conf file included the following settings, with the multipaths section only included for the “RHEL 7.2 Oracle RAC” golden image:

```
defaults {
polling_interval 10
user_friendly_names yes
find_multipaths yes
}
devices {
device {
vendor "3PARdata"
product "VV"
path_grouping_policy group_by_prio
path_selector "round-robin 0"
path_checker tur
features "0"
hardware_handler "1 alua"
prio alua
failback immediate
rr_weight uniform
no_path_retry 18
rr_min_io_rq 1
detect_prio yes
#uncomment these since using Broadcom CNA
fast_io_fail_tmo 10
dev_loss_tmo 14
}
}
multipaths {
        multipath {
                wwid "360002ac0000000000000000c0001b69e"
                alias data
        }
        multipath {
                wwid "360002ac0000000000000000d0001b69e"
                alias redo
        }
        multipath {
                wwid "360002ac0000000000000000e0001b69e"
                alias voting
        }
}
```

The /etc/udev/rules.d/12-oracle-permissions.rules file included the following settings for the Oracle data LUN for the “RHEL 7.2 Oracle” golden image. This file is modified by the plan script “RHEL 7 – Oracle – update udev rules” to specify the WWN for the 3PAR volume used as the Oracle data disk.

```
KERNEL=="dm*", \
PROGRAM=="scsi_id --page=0x83 --whitelisted --device=/dev/%k", \
RESULT=="360002ac0000000000000000090001b69e", \
OWNER="oracle", GROUP="oinstall", MODE="660"
```

The /etc/udev/rules.d/12-oracle-permissions.rules file included the following settings for the “RHEL 7.2 Oracle RAC” golden image for the Oracle data, redo and voting LUNs.

```
ENV{DM_NAME}=="data", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="voting", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
```

Appendix C: Oracle response files

The Oracle Grid Infrastructure and Database products each provide sample response files for silent installations of the products. These files were modified as follows and included in the "RHEL 7.2 for Oracle" golden image used by the "RHEL 7 Deploy Oracle" deployment plan.

The file /home/grid/grid-single-instance.rsp included the following entries. The ORACLE_HOSTNAME and oracle.install.asm.diskGroup.disks settings are modified by the plan script "RHEL 7 – Oracle – create install script".

```
ORACLE_HOSTNAME=aps41-19
SELECTED_LANGUAGES=en
oracle.install.option=HA_CONFIG
ORACLE_BASE=/u01/app/base
ORACLE_HOME=/u01/app/grid/12c
oracle.install.asm.OSDBA=asmdba
oracle.install.asm.OSOPER=asmoper
oracle.install.asm.OSASM=asmadmin
oracle.install.asm.SYSASMPassword=Password1234
oracle.install.asm.diskGroup.name=DATA1
oracle.install.asm.diskGroup.redundancy=EXTERNAL
oracle.install.asm.diskGroup.AUSize=1
oracle.install.asm.diskGroup.disks=/dev/mapper/mpathd
oracle.install.asm.diskGroup.diskDiscoveryString=/dev/mapper
oracle.install.asm.monitorPassword=Password1234
```

The file /home/oracle/db-single-instance.rsp included the following entries. The ORACLE_HOSTNAME setting is modified by the plan script "RHEL 7 – Oracle – create install script".

```
oracle.install.option=INSTALL_DB_SWONLY
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oracle/oraInventory
SELECTED_LANGUAGES=en
ORACLE_HOME=/u01/app/oracle/12c
ORACLE_BASE=/u01/app/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.DBA_GROUP=dba
oracle.install.db.BACKUPDBA_GROUP=dba
oracle.install.db.DGDBA_GROUP=dba
oracle.install.db.KMDBA_GROUP=dba
oracle.install.db.config.starterdb.type=GENERAL_PURPOSE
oracle.install.db.ConfigureAsContainerDB=false
oracle.install.db.config.starterdb.memoryOption=false
oracle.install.db.config.starterdb.installExampleSchemas=false
oracle.install.db.config.starterdb.managementOption=DEFAULT
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
```

Appendix D: HPE Synergy Image Streamer plan scripts for Oracle database deployment

Note
The scripts in Appendix D and Appendix E are provided as is by HPE. HPE has no obligation to maintain or support this software.

The following plan scripts were used for the OS deployment plan "RHEL 7 Deploy Oracle". Note that custom attributes (i.e. variables) in the scripts are enclosed by the @ character. These custom attributes are set at deployment time in the deployment plan.

RHEL 7 – mount - deploy

```
!echo "Logical Volumes in Golden Image:"
lvs
!echo "File Systems in Golden Image:"
list-filesystems
!echo "Mounting File Systems:"
mount /dev/rhel_@OrigHostname@/root /
mount /dev/sda2 /boot
mount /dev/sda1 /boot/efi
mkdir-p /tmp/ImageStreamer
```

RHEL 7 – change root volume name

```
# Rename volume group for new hostname
!echo "---Original volume group name---"
vgs
vgrename rhel_@OrigHostname@ rhel_@Hostname@
!echo "---New volume group name---"
vgs
upload -<<END /tmp/ImageStreamer/grub_configure
#! /bin/bash
echo "-------- original grub ----------"
cat /etc/default/grub
echo "-------- original grub.cfg ----------"
cat /boot/efi/EFI/redhat/grub.cfg
echo "------- original fstab ---------------"
cat /etc/fstab
ex -c '1,$s/@OrigHostname@/@Hostname@/g' -c wq /etc/default/grub
ex -c '1,$s/@OrigHostname@/@Hostname@/g' -c wq /boot/efi/EFI/redhat/grub.cfg
ex -c '1,$s/@OrigHostname@/@Hostname@/g' -c wq /etc/fstab
echo "-------- new grub ----------"
cat /etc/default/grub
echo "-------- new grub.cfg ----------"
cat /boot/efi/EFI/redhat/grub.cfg
echo "------- new fstab ---------------"
cat /etc/fstab
exit 0
END
chmod 755 /tmp/ImageStreamer/grub_configure
command /tmp/ImageStreamer/grub_configure
```

RHEL 7 – hostname configuration

```
# Set hostname
upload -<<EOF /etc/hostname
@Hostname@.@DomainName@
EOF
# Add /etc/hosts entry for hostname
upload -<<END /tmp/ImageStreamer/hosts_configure
#! /bin/bash
```

```
cp /etc/hosts /tmp/ImageStreamer/hosts
sed '/127.0.0.1/a @Hostname_ipaddress@ @Hostname@.@DomainName@ @Hostname@
' </tmp/ImageStreamer/hosts >/etc/hosts
exit 0
END
chmod 755 /tmp/ImageStreamer/hosts_configure
command /tmp/ImageStreamer/hosts_configure
echo "-------- new /etc/hostname----------"
cat /etc/hostname
echo "-------- new /etc/hosts ----------"
cat /etc/hosts
```

RHEL 7 – update resolv.conf

```
upload -<<EOF /tmp/ImageStreamer/update_resolv
#!/bin/bash
ex -c '1,$s/search *.*/search @DomainName@/' -c wq /etc/resolv.conf
echo "-------------new resolv.conf-----------------"
cat /etc/resolv.conf
EOF
chmod 755 /tmp/ImageStreamer/update_resolv
command /tmp/ImageStreamer/update_resolv
```

RHEL 7 – public network configuration

```
# config eth4 network configuration
upload -<<END /tmp/ImageStreamer/eth4_configure
#! /bin/bash
if [ "@eth4_dhcp:DHCP@" = "DHCP"  ]; then

        cat <<EOF > /etc/sysconfig/network-scripts/ifcfg-eth4
        DEVICE=eth4
        BOOTPROTO=dhcp
        HWADDR=@eth4_mac@
        ONBOOT=yes
        EOF
else
        cat <<EOF > /etc/sysconfig/network-scripts/ifcfg-eth4
        DEVICE=eth4
        BOOTPROTO=static
        HWADDR=@eth4_mac@
        IPADDR=@eth4_ipaddress@
        NETMASK=@eth4_netmask@
        GATEWAY=@eth4_gateway@
        ONBOOT=yes
        EOF
fi
exit 0
END
chmod 755 /tmp/ImageStreamer/eth4_configure
command /tmp/ImageStreamer/eth4_configure
echo "-------- new /etc/sysconfig/network-scripts/ifcfg-eth4 ----------"
cat /etc/sysconfig/network-scripts/ifcfg-eth4
```

RHEL 7 – Oracle – update kernel params

```
upload -<<EOF /tmp/ImageStreamer/update_kernel_params
#!/bin/bash
ex -c '1,$s/vm.nr_hugepages *= *[0-9]*/vm.nr_hugepages = @HUGEPAGES@/g'  -c '1,$s/kernel.shmmax *= *[0-
9]*/kernel.shmmax = @SHMMAX@/g' -c wq /etc/sysctl.conf
EOF
chmod 755 /tmp/ImageStreamer/update_kernel_params
```

```
command /tmp/ImageStreamer/update_kernel_params
```

RHEL 7 – Oracle – update limits

```
upload -<<EOF /tmp/ImageStreamer/update_limits
#!/bin/bash
ex -c '1,$s/memlock *[0-9][0-9]*/memlock @MEMLOCK@/g' -c wq /etc/security/limits.d/oracle-limits.conf
EOF
chmod 755 /tmp/ImageStreamer/update_limits
command /tmp/ImageStreamer/update_limits
```

RHEL 7 – Oracle – update udev rules

```
# update udev rules
upload -<<EOF /tmp/ImageStreamer/configure_udev
#!/bin/bash
#
# set WWN for Oracle data disk in udev permissions file
#
sed -i 's/^RESULT==.*/RESULT=="@DataDiskWWN@", \\ /' /etc/udev/rules.d/12-oracle-permissions.rules
#
echo "-------- updated 12-oracle-permissions.rules----------"
cat /etc/udev/rules.d/12-oracle-permissions.rules
EOF
chmod 0755 /tmp/ImageStreamer/configure_udev
command /tmp/ImageStreamer/configure_udev
```

RHEL 7 – Oracle – create firstboot service

```
upload -<<EOF /etc/systemd/system/firstboot.service
[Unit]
Description= Install and configure Oracle single instance
After=network.target multi-user.target multipathd.service
[Service]
Type=idle
TimeoutStartSec=0
User=root
ExecStart=/root/bootstrap/firstboot.sh
[Install]
WantedBy=multi-user.target
EOF
chmod 0664 /etc/systemd/system/firstboot.service
ln-s /etc/systemd/system/firstboot.service /etc/systemd/system/multi-user.target.wants/firstboot.service
echo "------ created firstboot.service---------------"
cat /etc/systemd/system/firstboot.service
```

RHEL 7 – Oracle – create firstboot.sh script

```
echo "--------------- create firstboot.sh script ----------------"
upload -<<END /tmp/ImageStreamer/firstboot_script
#!/bin/bash
mkdir -p /root/bootstrap
chmod 0744 /root/bootstrap
cat > /root/bootstrap/firstboot.sh <<EOF
#!/bin/bash
if [ ! -f  /root/bootstrap/.firstboot ]; then
        echo "First time boot, .firstboot not found"
        if [ -f /root/bootstrap/install.sh ]; then
                /root/bootstrap/install.sh
        else
                echo "could not find /root/bootstrap/install.sh"
        fi
        # this is a one time script
```

```
        touch /root/bootstrap/.firstboot
else
        echo "firstboot check complete"
fi
EOF
chmod 0754 /root/bootstrap/firstboot.sh
END
chmod 0755 /tmp/ImageStreamer/firstboot_script
command /tmp/ImageStreamer/firstboot_script
echo "--------------- created firstboot.sh script ----------------"
cat /root/bootstrap/firstboot.sh
```

RHEL 7 – Oracle – create install script

```
# create install.sh script
upload -<<EOF /root/bootstrap/install.sh
#!/bin/bash
#
# find and set name of device for ASM diskgroup
#
multipath -ll |grep @DataDiskWWN@ > mpath
sed -i 's/ [.*$//g' mpath
sed -i 's/^/\\\/dev\\\/mapper\\\//g' mpath
REPLACE=`cat mpath`
SED_COMMAND='s/^oracle.install.asm.diskGroup.disks=.*/oracle.install.asm.diskGroup.disks='${REPLACE}'/g'
su - grid -c "sed -i '${SED_COMMAND}' /home/grid/grid-single-instance.rsp"
#
# set passwords in grid response file
#
su - grid -c "sed -i 's/^oracle.install.asm.SYSASMPassword=.*/oracle.install.asm.SYSASMPassword=@asmpassword@/'
/home/grid/grid-single-instance.rsp"
su - grid -c "sed -i
's/^oracle.install.asm.monitorPassword=.*/oracle.install.asm.monitorPassword='@asmmonitorpassword@/'
/home/grid/grid-single-instance.rsp"
#
# replace hostname in response files
#

su - grid -c "sed -i 's/^ORACLE_HOSTNAME=.*/ORACLE_HOSTNAME=@Hostname@/' /home/grid/grid-single-instance.rsp"
su - oracle -c "sed -i 's/^ORACLE_HOSTNAME=.*/ORACLE_HOSTNAME=@Hostname@/' /home/oracle/db-single-instance.rsp"
#
# grid install
#
su - grid -c "/home/grid/runInstaller -waitforcompletion -ignoreSysPrereqs -ignorePrereq -silent -responseFile
/home/grid/grid-single-instance.rsp"
/u01/app/oraInventory/orainstRoot.sh
/u01/app/grid/12c/root.sh
#
# create cfgrsp.properties since it contains passwords
#
su - grid -c "echo 'oracle.assistants.asm|S_ASMPASSWORD=@asmpassword@' > /home/grid/cfgrsp.properties"
su - grid -c "echo 'oracle.assistants.asm|S_ASMMONITORPASSWORD=@asmmonitorpassword@' >>
/home/grid/cfgrsp.properties"
#
su - grid -c "/u01/app/grid/12c/cfgtoollogs/configToolAllCommands RESPONSE_FILE=cfgrsp.properties"
#
# oracle install
#
su - oracle -c "/home/oracle/database/runInstaller -waitforcompletion -silent -ignoreSysPrereqs -ignorePrereq -
responseFile /home/oracle/db-single-instance.rsp"
/u01/app/oracle/12c/root.sh
```

```
EOF
chmod 0754 /root/bootstrap/install.sh
echo "-------- created install.sh----------"
cat /root/bootstrap/install.sh
```

RHEL 7 – umount

```
rm-rf /tmp/ImageStreamer
umount /boot/efi
umount /boot
umount /
```

HPE – Foundation 1.0 – attempt or fail deployment – v1.0

```
# This plan script may be used to test failure of OS deployment.
echo "Attempt or Intentionally Fail OS Deployment as Requested."
@DeploymentTesting:Intentionally_Fail_OS_Deployment@
```

Appendix E: Image Streamer plan scripts for Oracle RAC deployment

The following plan scripts were used by the OS build plan "RHEL 7 RAC Add Node". Note that plan scripts that were used for both this build plan and the "RHEL 7 Deploy Oracle" build plan are not duplicated here.

RHEL 7 – RAC hostname configuration

```
# Set hostname
upload -<<EOF /etc/hostname
@HostnameNode3@.@DomainName@
EOF
# Add /etc/hosts entry for hostname
upload -<<END /tmp/ImageStreamer/hosts_configure
#! /bin/bash
cp /etc/hosts /tmp/ImageStreamer/hosts
sed '/127.0.0.1/a @HostnameNode3_IP@   @HostnameNode3@.@DomainName@   @HostnameNode3@\n@HostnameNode1_IP@
@HostnameNode1@.@DomainName@ @HostnameNode1@\n @HostnameNode2_IP@ @HostnameNode2@.@DomainName@
@HostnameNode2@\n@PrvNode1_IP@ @PrvNode1@.@DomainName@ @PrvNode1@\n@PrvNode2_IP@ @PrvNode2@.@DomainName@
@PrvNode2@\n@PrvNode3_IP@ @PrvNode3@.@DomainName@ @PrvNode3@\n@VipNode1_IP@ @VipNode1@.@DomainName@
@VipNode1@\n@VipNode2_IP@ @VipNode2@.@DomainName@ @VipNode2@\n@VipNode3_IP@ @VipNode3@.@DomainName@
@VipNode3@\n@Scan1_IP@ @Scan1@.@DomainName@ @Scan1@\n@Scan2_IP@ @Scan2@.@DomainName@ @Scan2@\n@Scan3_IP@
@Scan3@.@DomainName@ @Scan3@\n
' </tmp/ImageStreamer/hosts >/etc/hosts
exit 0
END
chmod 755 /tmp/ImageStreamer/hosts_configure
command /tmp/ImageStreamer/hosts_configure
echo "-------- new /etc/hostname----------"
cat /etc/hostname
echo "-------- new /etc/hosts ----------"
cat /etc/hosts
```

RHEL 7 – RAC public network configuration

```
# config eth4 network configuration
upload -<<END /tmp/ImageStreamer/eth4_configure
#! /bin/bash
# Check if this is a request to configure for DHCP
if [ "@eth4_dhcp:DHCP@" = "DHCP"  ]; then
        cat <<EOF > /etc/sysconfig/network-scripts/ifcfg-eth4
        DEVICE=eth4
        BOOTPROTO=dhcp
```

```
        HWADDR=@HostnameNode3_mac@
        ONBOOT=yes
        EOF
else
        cat <<EOF > /etc/sysconfig/network-scripts/ifcfg-eth4
        DEVICE=eth4
        BOOTPROTO=static
        HWADDR=@HostnameNode3_mac@
        IPADDR=@HostnameNode3_IP@
        NETMASK=@HostnameNode3_netmask@
        GATEWAY=@HostnameNode3_gateway@
        ONBOOT=yes
        EOF
fi
exit 0
END
chmod 755 /tmp/ImageStreamer/eth4_configure
command /tmp/ImageStreamer/eth4_configure
echo "-------- new /etc/sysconfig/network-scripts/ifcfg-eth4 ----------"
cat /etc/sysconfig/network-scripts/ifcfg-eth4
```

RHEL 7 – RAC private network configuration

```
# config Oracle RAC private network configuration
upload -<<END /tmp/ImageStreamer/privatenet_configure
#! /bin/bash
# Check if this is probably a request to configure for DHCP
if [ "@eth4_dhcp:DHCP@" = "DHCP"  ]; then
        cat <<EOF > /etc/sysconfig/network-scripts/ifcfg-ens3f6
        DEVICE=ens3f6
        BOOTPROTO=dhcp
        HWADDR=@PrvNode3_mac@
        ONBOOT=yes
        EOF
else
        cat <<EOF > /etc/sysconfig/network-scripts/ifcfg-ens3f6
        DEVICE=ens3f6
        BOOTPROTO=none
        HWADDR=@PrvNode3_mac@
        IPADDR=@PrvNode3_IP@
        NETMASK=@PrvNode3_netmask@
        ONBOOT=yes
        EOF
fi
exit 0
END
chmod 755 /tmp/ImageStreamer/privatenet_configure
command /tmp/ImageStreamer/privatenet_configure
echo "-------- new /etc/sysconfig/network-scripts/ifcfg-ens3f6 ----------"
cat /etc/sysconfig/network-scripts/ifcfg-ens3f6
```

RHEL 7 – RAC create firstboot services

```
# create firstboot services
upload -<<EOF /etc/systemd/system/firstboot-grid.service
[Unit]
Description=Configure ssh keys for grid
After=network.target
[Service]
Type=oneshot
User=grid
ExecStart=/home/grid/bootstrap/firstboot.sh
```

```
[Install]
WantedBy=multi-user.target
EOF
chmod 664 /etc/systemd/system/firstboot-grid.service
ln-s /etc/systemd/system/firstboot-grid.service /etc/systemd/system/multi-user.target.wants/firstboot-
grid.service
upload -<<EOF /etc/systemd/system/firstboot-oracle.service
[Unit]
Description=Configure ssh keys for oracle
After=network.target firstboot.service
[Service]
Type=oneshot
User=oracle
ExecStart=/home/oracle/bootstrap/firstboot.sh
[Install]
WantedBy=multi-user.target
EOF
chmod 664 /etc/systemd/system/firstboot-oracle.service
ln-s /etc/systemd/system/firstboot-oracle.service /etc/systemd/system/multi-user.target.wants/firstboot-
oracle.service
upload -<<EOF /etc/systemd/system/firstboot-addnode.service
[Unit]
Description= Add Oracle RAC node
After=network.target firstboot.service firstboot-oracle.service multi-user.target
[Service]
Type=idle
TimeoutStartSec=0
User=root
ExecStart=/root/bootstrap/firstboot-addnode.sh
[Install]
WantedBy=multi-user.target
EOF
chmod 664 /etc/systemd/system/firstboot-addnode.service
ln-s /etc/systemd/system/firstboot-addnode.service /etc/systemd/system/multi-user.target.wants/firstboot-
addnode.service
```

RHEL 7 – RAC create firstboot.sh scripts

```
# create firstboot.sh scripts
upload -<<END /tmp/ImageStreamer/firstboot_scripts
#!/bin/bash
mkdir -p /root/bootstrap
chmod 0744 /root/bootstrap
cat > /root/bootstrap/firstboot-addnode.sh <<EOF
#!/bin/bash
FIRST_BOOT_CHECK="/root/bootstrap/.firstboot_addnode"
if [ ! -f $FIRST_BOOT_CHECK ]; then
        echo "First time boot, .firstboot_addnode not found"
        if [ -f /root/bootstrap/addRACnode.sh ]; then
                /root/bootstrap/addRACnode.sh
        else
                echo "could not find /root/bootstrap/addRACnode.sh"
        fi
        # this is a one time script
        touch $FIRST_BOOT_CHECK
else
        echo "firstboot check complete"
fi
EOF
chmod 0754 /root/bootstrap/firstboot-addnode.sh
mkdir -p /home/grid/bootstrap
```

```
chmod 0744 /home/grid/bootstrap
chown grid:oinstall /home/grid/bootstrap
cat > /home/grid/bootstrap/firstboot.sh <<EOF
#!/bin/bash
FIRST_BOOT_CHECK="/home/grid/bootstrap/.firstboot"
if [ ! -f $FIRST_BOOT_CHECK ]; then
        echo "First time boot, .firstboot not found"
        if [ -f /home/grid/bootstrap/sshkey_config.sh ]; then
                /home/grid/bootstrap/sshkey_config.sh
        else
                echo "could not find /home/grid/bootstrap/sshkey_config.sh"
        fi
        # this is a one time script
        touch $FIRST_BOOT_CHECK
else
        echo "firstboot check complete"
fi
EOF
chmod 0754 /home/grid/bootstrap/firstboot.sh
chown grid:oinstall /home/grid/bootstrap/firstboot.sh
mkdir -p /home/oracle/bootstrap
chmod 0744 /home/oracle/bootstrap
chown oracle:oinstall /home/oracle/bootstrap
cat > /home/oracle/bootstrap/firstboot.sh <<EOF
#!/bin/bash
FIRST_BOOT_CHECK="/home/oracle/bootstrap/.firstboot"
if [ ! -f $FIRST_BOOT_CHECK ]; then
        echo "First time boot, .firstboot not found"
        if [ -f /home/oracle/bootstrap/sshkey_config.sh ]; then
                /home/oracle/bootstrap/sshkey_config.sh
        else
                echo "could not find /home/oracle/bootstrap/sshkey_config.sh"
        fi
        # this is a one time script
        touch $FIRST_BOOT_CHECK
else
        echo "firstboot check complete"
fi
EOF
chmod 0754 /home/oracle/bootstrap/firstboot.sh
chown oracle:oinstall /home/oracle/bootstrap/firstboot.sh
END
chmod 755 /tmp/ImageStreamer/firstboot_scripts
command /tmp/ImageStreamer/firstboot_scripts
```

RHEL 7 – RAC grid create sshkey_config.sh

```
# create sshkey_config.sh for grid  user
upload -<<EOF /home/grid/bootstrap/sshkey_config.sh
#!/bin/bash
# script to update hosts files and configure ssh keys
#
# add this server to hosts file on other nodes
#
# Node1
#
/usr/local/bin/sshpass -p@rootPassword@ ssh -o StrictHostkeyChecking=no -l root @HostnameNode1@ "cp /etc/hosts
/tmp/hosts; sed '/127.0.0.1/a @HostnameNode3_IP@  @HostnameNode3@.@DomainName@ @HostnameNode3@\n@PrvNode3_IP@
@PrvNode3@.@DomainName@ @PrvNode3@\n@VipNode3_IP@ @VipNode3@.@DomainName@ @VipNode3@' < /tmp/hosts > /etc/hosts"
#
# Node2
```

```
#
/usr/local/bin/sshpass -p@rootPassword@ ssh -o StrictHostkeyChecking=no -l root @HostnameNode2@ "cp /etc/hosts
/tmp/hosts; sed '/127.0.0.1/a @HostnameNode3_IP@  @HostnameNode3@.@DomainName@ @HostnameNode3@\n@PrvNode3_IP@
@PrvNode3@.@DomainName@ @PrvNode3@\n@VipNode3_IP@  @VipNode3@.@DomainName@ @VipNode3@' < /tmp/hosts > /etc/hosts"
#
# generate keys for node3
#
ssh-keygen -b 1024 -f /home/grid/.ssh/id_rsa -N ""
#
# copy public key to node1
#
/usr/local/bin/sshpass -p@gridPassword@  scp -o StrictHostkeyChecking=no  /home/grid/.ssh/id_rsa.pub
@HostnameNode1@:./.ssh/node3_rsa
#
# copy public key into authorized_keys file on node1
#
/usr/local/bin/sshpass -p@gridPassword@ ssh -o StrictHostkeyChecking=no @HostnameNode1@  "cat
/home/grid/.ssh/node3_rsa >> /home/grid/.ssh/authorized_keys"
#
# copy public key to node2
#
/usr/local/bin/sshpass -p@gridPassword@  scp -o StrictHostkeyChecking=no  /home/grid/.ssh/id_rsa.pub
@HostnameNode2@:./.ssh/node3_rsa
#
# copy public key into authorized_keys file on node2
#
/usr/local/bin/sshpass -p@gridPassword@ ssh -o StrictHostkeyChecking=no @HostnameNode2@ "cat
/home/grid/.ssh/node3_rsa >> /home/grid/.ssh/authorized_keys"
#
# copy node1's public key to node3
#
/usr/local/bin/sshpass -p@gridPassword@  scp -o StrictHostkeyChecking=no  @HostnameNode1@:./.ssh/id_rsa.pub
/home/grid/.ssh/node1_rsa
#
# copy node1's public key into authorized_keys file on node3
#
cat /home/grid/.ssh/node1_rsa >> /home/grid/.ssh/authorized_keys
#
# copy node2's public key to node3
#
/usr/local/bin/sshpass -p@gridPassword@  scp -o StrictHostkeyChecking=no  @HostnameNode2@:./.ssh/id_rsa.pub
/home/grid/.ssh/node2_rsa
#
# copy node2's public key into authorized_keys file on node3
#
cat /home/grid/.ssh/node2_rsa >> /home/grid/.ssh/authorized_keys
#
# copy node3's public key into authorized_keys file on node3
#
cat /home/grid/.ssh/id_rsa.pub >> /home/grid/.ssh/authorized_keys
#
# add Node3 to known_hosts on all three nodes
#
/usr/local/bin/sshpass -p@gridPassword@ ssh -o StrictHostkeyChecking=no @HostnameNode1@ "cp .ssh/known_hosts
.ssh/known_hosts_tmp; sed '/@HostnameNode2@/a @HostnameNode3@,@HostnameNode3_IP@ @KnownHostsString@' <
.ssh/known_hosts_tmp > .ssh/known_hosts"
#
/usr/local/bin/sshpass -p@gridPassword@ ssh -o StrictHostkeyChecking=no @HostnameNode2@ "cp .ssh/known_hosts
.ssh/known_hosts_tmp; sed '/@HostnameNode1@/a @HostnameNode3@,@HostnameNode3_IP@ @KnownHostsString@' <
.ssh/known_hosts_tmp > .ssh/known_hosts"
```

```
#
cp /home/grid/.ssh/known_hosts /home/grid/.ssh/known_hosts_tmp; sed '/@HostnameNode2@/a
@HostnameNode3@,@HostnameNode3_IP@ @KnownHostsString@' < /home/grid/.ssh/known_hosts_tmp >
/home/grid/.ssh/known_hosts
EOF
chmod 754 /home/grid/bootstrap/sshkey_config.sh
```

RHEL 7 – RAC oracle create sshkey_config.sh

```
# create sshkey_config.sh for oracle  user
upload -<<EOF /home/oracle/bootstrap/sshkey_config.sh
#!/bin/bash
# script to configure ssh keys for oracle user
#
# generate keys for node3
#
ssh-keygen -b 1024 -f /home/oracle/.ssh/id_rsa -N ""
#
# copy public key to node1
#
/usr/local/bin/sshpass -p@oraclePassword@  scp -o StrictHostkeyChecking=no  /home/oracle/.ssh/id_rsa.pub
@HostnameNode1@:./.ssh/node3_rsa
#
# copy public key into authorized_keys file on node1
#
/usr/local/bin/sshpass -p@oraclePassword@ ssh -o StrictHostkeyChecking=no @HostnameNode1@  "cat
/home/oracle/.ssh/node3_rsa >> /home/oracle/.ssh/authorized_keys"
#
# copy public key to node2
#
/usr/local/bin/sshpass -p@oraclePassword@  scp -o StrictHostkeyChecking=no  /home/oracle/.ssh/id_rsa.pub
@HostnameNode2@:./.ssh/node3_rsa
#
# copy public key into authorized_keys file on node2
#
/usr/local/bin/sshpass -p@oraclePassword@ ssh -o StrictHostkeyChecking=no @HostnameNode2@ "cat
/home/oracle/.ssh/node3_rsa >> /home/oracle/.ssh/authorized_keys"
#
# copy node1's public key to node3
#
/usr/local/bin/sshpass -p@oraclePassword@  scp -o StrictHostkeyChecking=no  @HostnameNode1@:./.ssh/id_rsa.pub
/home/oracle/.ssh/node1_rsa
#
# copy node1's public key into authorized_keys file on node3
#
cat /home/oracle/.ssh/node1_rsa >> /home/oracle/.ssh/authorized_keys
#
# copy node2's public key to node3
#
/usr/local/bin/sshpass -p@oraclePassword@  scp -o StrictHostkeyChecking=no  @HostnameNode2@:./.ssh/id_rsa.pub
/home/oracle/.ssh/node2_rsa
#
# copy node2's public key into authorized_keys file on node3
#
cat /home/oracle/.ssh/node2_rsa >> /home/oracle/.ssh/authorized_keys
#
# copy node3's public key into authorized_keys file on node3
#
cat /home/oracle/.ssh/id_rsa.pub >> /home/oracle/.ssh/authorized_keys
#
# add Node3 to known_hosts on all three nodes
```

```
#
/usr/local/bin/sshpass -p@oraclePassword@ ssh -o StrictHostkeyChecking=no @HostnameNode1@ "cp .ssh/known_hosts
.ssh/known_hosts_tmp; sed '/@HostnameNode2@/a @HostnameNode3@,@HostnameNode3_IP@ @KnownHostsString@' <
.ssh/known_hosts_tmp > .ssh/known_hosts"
#
/usr/local/bin/sshpass -p@oraclePassword@ ssh -o StrictHostkeyChecking=no @HostnameNode2@ "cp .ssh/known_hosts
.ssh/known_hosts_tmp; sed '/@HostnameNode1@/a @HostnameNode3@,@HostnameNode3_IP@ @KnownHostsString@' <
.ssh/known_hosts_tmp > .ssh/known_hosts"
#
cp /home/oracle/.ssh/known_hosts /home/oracle/.ssh/known_hosts_tmp; sed '/@HostnameNode2@/a
@HostnameNode3@,@HostnameNode3_IP@ @KnownHostsString@' < /home/oracle/.ssh/known_hosts_tmp >
/home/oracle/.ssh/known_hosts
EOF
chmod 754 /home/oracle/bootstrap/sshkey_config.sh
```

RHEL 7 – RAC create addRACnode.sh script

```
# create addRACnode.sh script
upload -<<EOF /root/bootstrap/addRACnode.sh
#!/bin/bash
#
# run addnode.sh for both grid and oracle accounts
#
# grid account
#
su - grid -c "ssh @HostnameNode1@ '/u01/app/grid/12c/addnode/addnode.sh -ignoreSysPrereqs -ignorePrereq -silent
"CLUSTER_NEW_NODES={@HostnameNode3@}" "CLUSTER_NEW_VIRTUAL_HOSTNAMES={@VipNode3@}"'"
/u01/app/oraInventory/orainstRoot.sh
/u01/app/grid/12c/root.sh
#
# oracle account
#
su - oracle -c "ssh @HostnameNode1@ '/u01/app/oracle/12c/addnode/addnode.sh -silent
"CLUSTER_NEW_NODES={@HostnameNode3@}"'"
/u01/app/oracle/12c/root.sh
EOF
chmod 754 /root/bootstrap/addRACnode.sh
echo "-------- created addRACnode.sh----------"
cat /root/bootstrap/addRACnode.sh
```

Appendix F: Custom attributes for deployment plan

Table 5. Default values for custom attributes for deployment plan "RHEL 7 RAC Add Node"

Custom attribute name	Default value	Comment
DeploymentTesting	Intentionally_Fail_OS_Deployment	For debugging purposes. Set to #Attempt_OS_Deployment to do actual deployment.
DomainName	rac	Domain name
eth4_DHCP	NO	Don't use DHCP for public network
gridPassword	Password\!1234	Password for grid user account. Special characters such as ! must be escaped for usage in scripts
Hostname	racnode3	Hostname for new RAC node
HostnameNode1	racnode1	Hostname for first RAC node
HostnameNode1_IP	10.145.41.54	IP address for first RAC node
HostnameNode2	racnode2	Hostname for second RAC node
HostnameNode2_IP	10.145.41.55	IP address for second RAC node
HostnameNode3	racnode3	Hostname for third RAC node
HostnameNode3_gateway	10.145.40.1	Gateway for third RAC node
HostnameNode3_IP	10.145.41.56	IP address for third RAC node
HostnameNode3_mac	No default value	Obtain MAC address from public network connection in server profile
HostnameNode3_netmask	255.255.248.0	Netmask for new RAC node
HUGEPAGES	25653	Used to set value for vm.nr_hugepages in /etc/sysctl.conf(dependent on memory size)
KnownHostsString	ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAy NTYAAAAIbmlzdHAyNTYAAABBBKA G+nUzrnAQqzQ9VKg+50LZMWYeqiT FiHuHMVfs55XXDry65MbeeJJkWOsL X0sWGyFmuYU8nc9AWZ1ICyvEOAs=	Use value found in known_hosts file on existing nodes
MEMLOCK	59104501	Used to set value of memlock in limits file (dependent on memory size)
oraclePassword	Password\!1234	Password for oracle account. Note that special characters must be escaped for usage in scripts
OrigHostname	aps41-19	Hostname in golden image
PrvNode1	racnode1-prv	Name for private (heartbeat) network on first RAC node
PrvNode1_IP	192.168.0.54	IP address for private network on first RAC node
PrvNode2	racnode2-prv	Name for private network on second RAC node
PrvNode2_IP	192.168.0.55	IP address for private network on second RAC node
PrvNode3	racnode3-prv	Name for private network on third RAC node
PrvNode3_IP	192.168.0.56	IP address for private network on third RAC node
PrvNode3_mac	No default	Use MAC address from server profile connection for private network
PrvNode3_netmask	255.255.255.0	Netmask for private network
rootPassword	Password1234	Password for root account

Custom attribute name	Default value	Comment
Scan1	synergy-rac	Cluster scan name 1
Scan1_IP	10.145.41.42	Cluster scan address 1
Scan2	synergy-scan2	Cluster scan name 2
Scan2_IP	10.145.41.43	Cluster scan address 2
Scan3	synergy-scan3	Cluster scan name 3
Scan3_IP	10.145.41.44	Cluster scan address 3
SHMMAX	53798230426	Setting for kernel.shmmax in /etc/sysctl.conf (dependent on memory size)
VipNode1	racnode1-vip	Name for virtual IP 1
VipNode1_IP	10.145.41.39	Virtual IP for node1
VipNode2	racnode2-vip	Name for virtual IP 2
VipNode2_IP	10.145.41.40	Virtual IP for node2
VipNode3	racnode3-vip	Name for virtual IP 3
VipNode3_IP	10.145.41.41	Virtual IP for node3

Resources and additional links

HPE Reference Architectures, hpe.com/info/ra

HPE Synergy, hpe.com/info/synergy

HPE Synergy Image Streamer QuickSpecs, https://www.hpe.com/h20195/v2/GetPDF.aspx/c04815217.pdf

HPE Synergy Composer QuickSpecs, https://www.hpe.com/h20195/v2/GetPDF.aspx/c04815139.pdf

HPE Synergy 12000 Frame QuickSpecs, https://www.hpe.com/h20195/v2/GetPDF.aspx/c04815113.pdf

HPE Synergy 480 Gen9 Compute Module QuickSpecs, https://www.hpe.com/h20195/v2/GetPDF.aspx/c04815134.pdf

Oracle Database Online Documentation 12c Release 1, https://docs.oracle.com/database/121/nav/portal_11.htm

HPE Synergy for Oracle 12c Technical Brief, https://www.hpe.com/h20195/v2/GetPDF.aspx/4AA6-6554ENN.pdf

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

Learn more at, hpe.com/info/synergy

Sign up for updates

a00001298enw, February 2017