Oracle® VM Server for SPARC OpenStack Nova Driver and Utilities 1.0 Administration Guide

Part No: E79549
May 2017

ORACLE

Oracle VM Server for SPARC OpenStack Nova Driver and Utilities 1.0 Administration Guide

Part No: E79549

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Référence: E79549

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.

Contents

Using This Documentation

- **Overview** – Provides cloud administrators with detailed information and procedures that describe the installation and configuration of an OpenStack Nova compute node with the Oracle VM Server for SPARC software.
- **Audience** – Cloud administrators who manage cloud services on SPARC servers.
- **Required knowledge** – Cloud administrators on these servers must have a working knowledge of UNIX systems, the Oracle Solaris operating system (Oracle Solaris OS), and OpenStack.

Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Overview

This chapter covers the following topics:

About OpenStack

OpenStack is an integrated component of the Oracle Solaris 11.3 OS that provides the tools that you need to create an IaaS (Infrastructure as a Service) private, multi-tenant cloud within your own data center.

The Solaris OpenStack offering includes the following core components that are required to build an OpenStack cloud:

- "Nova" (compute and virtualization)
- "Neutron" (networking)
- "Cinder" (volume and block storage)
- "Horizon" (web-based management dashboard)
- "Glance" (image management)
- "Keystone" (identity service)

You can consolidate various OpenStack services onto a single system, which is referred to as a single-node controller. Typically in a production environment, these services are spread across multiple systems. Depending on your environment, you can use a multi-node controller configuration to separate services to improve performance, provide increased availability, or both.

Version 1.0 of the Oracle VM Server for SPARC Nova driver provides the capability to use SPARC-based systems as *compute nodes* that are part of an OpenStack cloud.

Oracle VM Server for SPARC OpenStack Nova Driver 1.0 Features and Capabilities

The Oracle VM Server for SPARC OpenStack Nova Driver 1.0 includes the following features and capabilities:

- Logical domain instances
  - Supports Oracle Solaris 10, Oracle Solaris 11, and Linux for SPARC 1.0 (experimental) as guest domains
  - Includes an historical console log
  - Supports VNC and an serial console (experimental)
  - Supports live migration and cross-CPU live migration for Oracle Solaris OS guest domains
  - Includes unique Oracle VM Server for SPARC capabilities such as whole-core and max-core constraints
  - Supports Nova evacuation on shared storage only
  - Supports Solaris WAN boot
  - Supports parallel guest domain deployment
- Networking
  - Supports VLAN and flat networks
  - Supports alternate MAC addresses and optionally permits for extra VNICs to be created within a logical domain
  - Supports multi-homed network connectivity to guest domains
  - Supports multiple virtual switches for physical separation of network traffic
  - Supports varied MTU frame sizes
  - Supports the dynamic attachment and detachment of network ports for Solaris guest domains

    Note that only the Neutron ML2 core plugin can be used with Oracle VM Server for SPARC. The Oracle Solaris OS elastic virtual switch (EVS) is not supported for use with this Nova driver.
- Storage
  - Supports Cinder volumes that use Fibre Channel and iSCSI
  - Supports locally managed storage for NFS, local file systems, and ZFS volumes
  - Supports multiple virtual disks
  - Supports dynamic volume attachment and detachment for Oracle Solaris OS guest domains

- Supports the automatic resizing of disk labels, slices, and file systems for Oracle Solaris OS guest domains (EFI and VTOC) and for Linux for SPARC 1.0 (VTOC only)

♦ ♦ ♦ CHAPTER 2

Oracle VM Server for SPARC OpenStack Compute Node Prerequisites

This chapter describes the prerequisites to meet before you can install and configure an Oracle VM Server for SPARC OpenStack compute node.

Note - OpenStack and the Oracle Solaris OS requires reliable storage and network connectivity. In particular, configure redundant power, redundant network connectivity, and redundant storage connectivity on all systems that are part of an OpenStack cloud.

This chapter covers the following topics:

OpenStack Cloud Controller Prerequisites

- Run the Oracle Solaris 11.3 SRU 12 OS with OpenStack (Kilo) on the following servers:
  - x86 (bare metal, kernel zone)
  - SPARC (bare metal, logical domain, or kernel zone)
- Install the `openstack-ldoms-controller` package that is included the with the Nova driver download. See "Obtaining the Oracle VM Server for SPARC OpenStack Compute Node Software" on page 19.

Note - For non-production environments (such as development, testing, trial and demonstration environments), use the `create-demo-controller.sh` script in the `openstack-ldoms-controller` package to quickly get started with OpenStack and the Oracle VM Server for SPARC Nova driver.

Oracle VM Server for SPARC OpenStack Compute Node Prerequisites

Hardware Requirements

- **Cloud Controller:**
  - **CPU –** Has an x86 or SPARC server or a virtual machine that has at least a 2 Ghz processor.
  - **RAM –** Has at least 16 Gbytes of RAM.
  - **Disk –** Has at least one Tbyte of disk space if the cloud controller also acts as a Cinder volume storage provider, else you can use 146-300 Gbytes of local storage.
- **Compute Node:**
  - **CPU –** A `sun4v` platform that runs at least an UltraSPARC T2 server is required. It is best to use at least a SPARC T4 series server.
  - **RAM –** Has at least 128 GBytes of RAM. Assign at least eight Gbytes to the control domain or 16 Gbytes when using local ZFS volumes.
  - **Disk –** Has at least 300 Gbytes of disk space plus local storage for virtual machines if you use local files or ZFS volumes.

    Note - Compute nodes use local storage to cache VM images.

  - **Network –** Configures the network by using the `DefaultFixed` network configuration profile on the control domain.
  - **Nodes –** Requires at least two compute nodes for live migration. If the compute nodes are different platform types, specify the appropriate `cpu-arch` value for the virtual machines that you want to migrate. See “Creating Oracle VM Server for SPARC OpenStack Compute Node Flavors” on page 38.

Software Requirements

- **Cloud controller and compute nodes:**
  - Runs at least the Oracle Solaris 11.3 SRU 12 OS.

  - The cloud controller OS must supply at least the same OpenStack release as is being used on the compute nodes. For this release, it is the OpenStack Kilo release.
- **Guest domains:**
  - Runs at least the Oracle Solaris 10 1/13 OS, the Oracle Solaris 11 OS, or the Linux for SPARC 1.0 OS.
  - The Oracle Solaris 10 1/13 OS, Oracle Solaris 11.2 OS, Oracle Solaris 11.3 SRU 12 OS, and the Linux for SPARC 1.0 OS have been validated.

Distributed Lock Management Prerequisites

The distributed lock manager is optional. However, if you use DLM, it has the following prerequisites:

- Requires a highly available NFSv4 server that is external to the compute nodes.
- Requires a single NFSv4 share that has at least one Gbyte of free space.
- Requires that delegation is disabled on the NFSv4 share.
- Requires that the NFSv4 share is read-write for both the `root` user (UID 0) and the `nova` user (UID 85, GID 85).

♦ ♦ ♦ CHAPTER 3

Installing an Oracle VM Server for SPARC OpenStack Compute Node

This chapter describes how to install and configure a basic Oracle VM Server for SPARC OpenStack compute node by using the `setup.sh` script.

For information about configuring a more advanced compute node, see Chapter 4, "Oracle VM Server for SPARC OpenStack Compute Node Configuration Reference".

This chapter covers the following topics:

Obtaining the Oracle VM Server for SPARC OpenStack Compute Node Software

▼ How to Obtain the Oracle VM Server for SPARC OpenStack Compute Node Software

1. **Obtain the Oracle VM Server for SPARC OpenStack Nova driver and utilities 1.0 package from MOS at `https://updates.oracle.com/Orion/PatchDetails/process_form?patch_num=24654094`.**

2. **Obtain the distributed lock manager (DLM) package from MOS at `https://updates.oracle.com/Orion/PatchDetails/process_form?patch_num=22902518`.**

3. **Extract the contents of the DLM patch.**

```
# unzip p22902518_30_SOLARIS64.zip
Archive:  p22902518_30_SOLARIS64.zip
  inflating: README.html
  inflating: ovs-ldoms-3.4.1-b1350.tar.gz
  inflating: ovs-dlm-3.4.1-b1350.p5p
```

Caution - The `ovs-dlm-3.4.1-b1350.p5p` package in the `p22902518_30_SOLARIS64.zip` file is installed optionally on compute nodes to provide DLM. You must not install any other components from this `.zip` file. In particular, remove the `ovs-ldoms-3.4.1-b1350.tar.gz` file as its packages are incompatible with the Oracle VM Server for SPARC OpenStack Nova driver.

4. **Extract the contents of the Oracle VM Server for SPARC OpenStack Nova driver patch.**

```
# unzip openstack-ldoms-1.0.zip
Archive:  openstack-ldoms-1.0.zip
   creating: openstack-ldoms/
  inflating: openstack-ldoms/openstack-ldoms-nova-1.0.p5p
   creating: openstack-ldoms/simple-init/
  inflating: openstack-ldoms/simple-init/simple-init-1.0_s11.p5p
  inflating: openstack-ldoms/simple-init/simple-init-1.0.iso
  inflating: openstack-ldoms/simple-init/simple-init-1.0_s10.pkg
  inflating: openstack-ldoms/openstack-ldoms-controller-1.0.p5p
  inflating: README.txt
```

This `.zip` file includes the following contents:

- `openstack-ldoms/openstack-ldoms-nova-1.0.p5p` – Oracle VM Server for SPARC OpenStack Nova Compute Driver 1.0 package
- `openstack-ldoms/openstack-ldoms-controller-1.0.p5p` – Oracle VM Server for SPARC OpenStack Controller Utilities 1.0 package
- `openstack-ldoms/simple-init/simple-init-1.0.iso` – `simple-init` ISO image used when creating golden images
- `openstack-ldoms/simple-init/simple-init-1.0_s10.pkg` – Oracle Solaris 10 guest initialization package for use with Oracle Solaris 10 WAN boot guest installations
- `openstack-ldoms/simple-init/simple-init-1.0_s11.p5p` – Oracle Solaris 11 guest initialization package for use with Oracle Solaris 11 WAN boot guest installations

Configuring a Demonstration Version of the OpenStack Cloud Controller

The Oracle VM Server for SPARC OpenStack controller package includes a script to create a demonstration cloud controller for non-production use.

▼ How to Install the Demonstration OpenStack Cloud Controller Package

Before You Begin Install at least the Oracle Solaris 11.3 SRU 12 OS on an x86 or SPARC bare metal system, a SPARC logical domain, or an x86 or SPARC kernel zone.

1. **Create a new boot environment (BE) before you continue.**

   For example, create and activate the `solaris11.3_openstack` BE:

   ```
   cctrl# beadm create solaris11.3_openstack
   cctrl# beadm activate solaris11.3_openstack
   cctrl# reboot
   ```

2. **Install the `openstack-ldoms-controller` package.**

   Perform this step in the new BE.

   ```
   cctrl#pkg install -g openstack-ldoms-controller-1.0.p5p openstack-ldoms-controller
   ```

▼ How to Configure a Demonstration Single-Node OpenStack Cloud Controller (Optional)

1. **Create a copy of the `/opt/openstack-ldoms/etc/controller_setup.conf.example` file to an alternate location.**

   For example:

   ```
   cctrl# cp /opt/openstack-ldoms/etc/controller_setup.conf.example /var/tmp/
   controller_setup.conf
   ```

2. **Assign values to the variables in your copy of the `controller_setup.conf` file that pertain to your environment.**

Ensure that you assign passwords to the `MY_ADMIN_PASSWORD` and `MY_SERVICE_PASSWORD` variables.

```
########################################
# Which NIC for OpenStack management network?
MNGT_NET=net0

# What is the IP of the OpenStack controller?
CONTROLLER_IP=`ipadm|grep ${MNGT_NET}/v4|awk '{print $5}'|cut -f1 -d/`

# What is the controller's shortname?
CONTROLLER_SHORTNAME=`hostname`

# Password for the admin user
export MY_ADMIN_PASSWORD=password

# Password for the service accounts and mysql
export MY_SERVICE_PASSWORD=password

# Zpool to be used for iSCSI LUNs
export CINDER_ZPOOL=rpool

# VLAN range to be allowed
export VLAN_RANGE="1-4000"
########################################
```

3. **Create the demonstration cloud controller.**

```
cctrl# /opt/openstack-ldoms/bin/create-demo-controller.sh /var/tmp/controller_setup.conf
```

Installing an Oracle VM Server for SPARC OpenStack Compute Node

Caution - The `setup.sh` script configures a basic compute node for use with a single-node OpenStack cloud controller.

If the OpenStack services in your environment are spread across multiple nodes, you might need to perform additional steps to configure compute nodes. For more information, see "Oracle VM Server for SPARC OpenStack Compute Node Configuration Properties" on page 27 and the OpenStack (Kilo) configuration file reference guides (`http://docs.openstack.org`).

Note that the cloud controller and the compute nodes must run the “Kilo” OpenStack release.

▼ How to Prepare the Oracle VM Server for SPARC OpenStack Compute Node (Optional)

Optionally use this task if the system you want to use as a compute node must be installed from bare metal.

1. **Install at least the Oracle Solaris 11.3 SRU 12 OS.**

2. **Ensure that the system is in `factory-default`.**

```
# ldm list-spconfig
```

If the system is not in `factory-default`, perform the following command:

```
# ldm set-spconfig factory-default
```

3. **(Optional) If system was not in `factory-default`, power off the server.**

Use either of the following commands:

```
# shutdown -i 5 -g0 -y
```

Or:

```
# poweroff
```

4. **Create a boot environment (BE).**

```
# BE=BE-name
# beadm create ${BE}
```

5. **Boot into the BE.**

```
# beadm activate ${BE}
```

6. **Reboot the BE.**

```
# shutdown -i6 -g0 -y
```

How to Install an Oracle VM Server for SPARC OpenStack Compute Node

1. **Obtain and download the packages.**

   See "Obtaining the Oracle VM Server for SPARC OpenStack Compute Node Software" on page 19.

2. **Install the packages.**

   ```
   # pkg install -g openstack-ldoms-nova-1.0.p5p nova-ldoms
   # pkg install -g ovs-dlm-3.4.x-bxxxx.p5p dlm
   ```

3. **Create the installation configuration file based on the sample file.**

   ```
   # cp /opt/openstack-ldoms/etc/setup.conf.example /var/tmp/setup.conf
   ```

4. **Modify the `setup.conf` configuration file.**

   At a minimum, specify values for the `CONTROLLER_IP`, `NOVA_SERVICE_PASSWORD`, and `NEUTRON_SERVICE_PASSWORD` properties. Validate that the remaining default values set by the other properties are appropriate for your environment before you proceed.

   - `LDOMS_VSW_NET=`*NIC* specifies the NIC for the primary virtual switch, `primary-vsw0`. For example, the value might be `net0`.
   - `LDOMS_HOST_IP=`` `ipadm|grep ${LDOMS_VSW_NET}/v4|awk '{print $5}'|cut -f1 -d/` `` specifies the IP address that is used for communications with the OpenStack cloud controller.
   - `LDOMS_HOSTNAME=`` `hostname` `` specifies the host name of the system.
   - `CONTROLLER_IP=`*IP-address* specifies the IP address of the OpenStack cloud controller that runs Keystone, Neutron, Cinder, and the Nova conductor and scheduler services.
   - `NOVA_SERVICE_PASSWORD=`*services-password* specifies the services password for the central cloud controller. Note that this password must match the Nova administrative service account as configured in Keystone.
   - `NEUTRON_SERVICE_PASSWORD=`*services-password* specifies the services password for the central cloud controller. Note that this password must match the Neutron administrative service account as configured in Keystone.
   - `ZVOL_LOCATION=`*ZVOL-location* specifies the local ZFS dataset to be used for virtual machines (when using ZFS volume support). The default value is `rpool/vm_disks`.
   - `DLM_NFS_SERVER=`*IP-address* optionally specifies the IP address of the NFS server for DLM if DLM is to be used. The NFS server must be NFSv4 and have `delegation=off`. To disable DLM, set this property to an empty value.

- `DLM_NFS_SHARE=`*share-name* specifies the NFS share on the NFS server that is used for DLM.
- `CDOM_CORES=`*num-of-cores* specifies the number of cores that are allocated to the control domain. The default value is `1`.
- `CDOM_RAM=`*amount-of-RAM* specifies the amount of RAM in gigabytes that are allocated to the control domain. The default value is `16`.

5. **Run the `setup.sh` installer script.**

```
# /opt/openstack-ldoms/bin/setup.sh /var/tmp/setup.conf
```

6. **(Optional) Enable enable Fibre Channel multipathing.**

```
# stmsboot -D fp -e
```

7. **Reboot the system.**

```
# shutdown -i6 -g0 -y
```

How to Verify the Compute Node Configuration

Use this procedure to verify the Oracle VM Server for SPARC OpenStack compute node configuration.

Upon reboot, the server starts the `nova-compute` service automatically. If DLM is enabled, the Nova driver configures and joins the DLM cluster within its availability zone automatically. Note that DLM is enabled if the `dlm_nfs_server` property is set to an IP address in the `nova.conf` file.

1. **Ensure that the `nova-compute` service starts properly.**

When the `nova-compute` service and driver start or restart, you should see lines in the log that look similar to the following:

```
nova# tail -f `svcs -L nova-compute`
2016-07-07 15:20:14.011 1098 DEBUG nova.service [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] Creating RPC server for service compute
 start /usr/lib/python2.7/vendor-packages/nova/service.py:188
2016-07-07 15:20:14.013 1098 INFO oslo_messaging._drivers.impl_rabbit [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] Connecting to AMQP server on 10.0.68.21:5672
2016-07-07 15:20:14.026 1098 INFO oslo_messaging._drivers.impl_rabbit [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] Connected to AMQP server on 10.0.68.21:5672
```

```
2016-07-07 15:20:14.033 1098 DEBUG nova.service [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] Join ServiceGroup membership for this service
 compute start /usr/lib/python2.7/vendor-packages/nova/service.py:206
2016-07-07 15:20:14.033 1098 DEBUG nova.servicegroup.drivers.db [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] DB_Driver: join new ServiceGroup member
 10.0.68.22 to the compute group, service = <nova.service.Service object at 0xea967cf0>
 join /usr/lib/python2.7/vendor-packages/nova/servicegroup/drivers/db.py:59
```

If you do not see a connection to AMQP and the `DB_Driver: join new ServiceGroup member` message after five minutes, ensure that the `nova-compute` service is running and validate the compute node's configuration. See Chapter 4, “Oracle VM Server for SPARC OpenStack Compute Node Configuration Reference”.

If the problem is a configuration issue, you should see an exception that starts with `EXCEPTION` or `ERROR` early in the startup log.

If the problem is not a configuration issue, ensure that the MTU for the management network is the same everywhere, that NTP is configured, and that name resolution is operating properly.

2. **(Optional) Confirm that the cluster is `READY`.**

   Only perform this check if the DLM package is installed and if DLM is enabled.

   ```
   nova# dlmcli STATUS
   ```

   The cluster status can be one of the following values:

   - `UNCONFIGURED`
   - `CONFIGURING`
   - `READY`

   If the cluster status is not `READY`, wait a few minutes and re-check the status. If a re-check shows that the cluster status is not `READY`, the cluster might not be properly configured. See “Understanding Log Output From the Nova Driver” on page 48.

♦ ♦ ♦ CHAPTER 4

Oracle VM Server for SPARC OpenStack Compute Node Configuration Reference

This chapter covers the following topics:

Oracle VM Server for SPARC OpenStack Compute Node Configuration Properties

This reference section covers standard OpenStack Nova configuration information that pertains to the Oracle VM Server for SPARC OpenStack Nova driver as well as configuration information that is specific to the Oracle VM Server for SPARC OpenStack Nova driver.

Oracle VM Server for SPARC OpenStack Compute Node

This section provides reference information about the Oracle VM Server for SPARC OpenStack compute node `/etc/nova/nova.conf` configuration file and the properties that are specific to the Oracle VM Server for SPARC product.

Compute Node `/etc/nova/nova.conf` Configuration File

The following sample `/etc/nova/nova.conf` configuration file is based on a cloud controller with an IP address of `10.0.68.51` and a host name of `controller`, a compute node with an

IP address of `10.0.68.50` and a host name of `compute0`, and a service password of *services-password* for all service accounts. Each line that has been updated with the cloud controller and compute node information is highlighted.

```
[DEFAULT]
rabbit_host=10.0.68.51
my_ip=10.0.68.50
host=compute0
block_device_allocate_retries=360
compute_driver=ldoms.LDomsDriver
novncproxy_base_url=http://10.0.68.51:6080/vnc_auto.html
vncserver_listen=$my_ip
vncserver_proxyclient_address=$my_ip
vnc_enabled=true

[database]
connection = mysql://nova:services-password@controller.domain-name/nova

[glance]
host=10.0.68.51

[keystone_authtoken]
auth_uri = http://10.0.68.51:5000/v2.0/
signing_dir=$state_path/keystone-signing
identity_uri = http://10.0.68.51:35357/
admin_user = nova
admin_password = services-password
admin_tenant_name = service

[neutron]
url=http://10.0.68.51:9696
admin_username=neutron
admin_password=services-password
admin_tenant_name=service
admin_auth_url=http://10.0.68.51:5000/v2.0
proxyclient_address=$my_ip

[serial_console]
serialproxy_host=10.0.68.51
serialproxy_port=6083
enabled=true
base_url=ws://10.0.68.51:6083/
listen=$my_ip
proxyclient_address=$my_ip

[ldoms]
```

```
set_admin_passwd_on_guest=false
permit_root_logins=true
admin_user=root
force_volume_build=true
create_zvols=true
zvol_base=rpool/vm_disks
configure_all_routes=false
vds_name=primary-vds0
min_free_space_pct=10
dlm_nfs_server=10.0.68.51
dlm_nfs_share=/saspool/nfspool
dlm_nfs_mountpoint=/mnt/dlm
dlm_port=4999
```

Oracle VM Server for SPARC OpenStack Compute Node Configuration File Properties

The following section lists the properties that are specific to the Oracle VM Server for SPARC product.

- `admin_user=root` specifies the account name of the administration user in the guest domain. The default value is `root`.
- `create_zvols=true | false` specifies whether to store Oracle VM Server for SPARC root disks on top of ZFS volumes. This property also depends on the value of the `force_volume_build` property.

  When this property value is `false` and `force_volume_build=false`, Oracle VM Server for SPARC root disks are raw files stored in the location specified by `instance_path`, which defaults to `/var/lib/nova/instances`.

  When this property value is `true` and `force_volume_build=false`, Oracle VM Server for SPARC root disks are stored on top of ZFS volumes. The initial root ZFS volume stores the Oracle VM Server for SPARC image, and each VM is a clone of the base ZFS volume that is created for this image.

  The default value is `true`.
- `dlm_nfs_mountpoint=`*mount-point* specifies the local mount point. The default mount point is `/mnt/dlm` and should be used as-is.
- `dlm_nfs_server=`*IP-address* specifies an IP address to an NFSv4 server that has `delegation=off`. This property must be set to enable distributed lock management (DLM), and the Nova evacuation and rebuild capabilities. If this property is not set, or is set to no value, DLM is disabled, and Nova evacuations fail.
- `dlm_nfs_share=/export/dlm` specifies the NFSv4 share on the NFSv4 server for DLM. The share requires at least one Gbyte of disk space and must be reliably accessible. If this

server is inaccessible for as long as ten minutes, any compute nodes that cannot access the share panic.

- `dlm_port=`*port-number* specifies the port that DLM uses to communicate with other nodes. The default value is `4999` and should be left as-is unless this port is already used.

Note - This port must be set to the same port on all compute nodes.

- `force_volume_build=true | false` specifies whether to force the use of Cinder volumes instead of local storage even when a cloud user requests a VM that uses Boot from Image that typically uses local storage (virtual disk files or ZFS volumes). When the value is `true`, mimic Oracle Solaris OS behavior which uses Cinder volumes to create Boot from Image instances. When the value is `false`, mimic OpenStack behavior which uses Nova-managed local storage. The default value is `false`.
- `configure_all_routes=true | false` specifies whether to configure all default routes. When set to `true`, all default routes are configured if your environment has multiple networks per VM and more than one network has a default route.

Note - When you configure a logical domain with multiple networks, multiple default routers might be provided to the guest domain. By default, the guest domain configures the first default router presented to it, based on the order in which the networks are attached to the domain. If you want to set all default routers that are provided, set `configure_all_routes=true` in the `/etc/nova/nova.conf` configuration file. This property is set to `false` by default, as using multiple default routers without careful consideration can cause network issues.

When set to `false`, the VM uses the default route only from the first network configured with a default route on this VM. The default value is `false`.

- `min_free_space_pct=`*percentage* specifies the minimum percentage of space that must be free to create new instances when using local file-based storage. If creating a new VM image will leave less than the specified percentage, the creation fails. Valid values are from `0` to `100`. The default value is `10` percent.
- `permit_root_logins=true | false` specifies whether to permit direct SSH logins by the `root` user. The default value is `false`.
- `set_admin_passwd_on_guest=true | false` specifies whether to set the administration user password on the guest domain. When the value is set to `true` and you use Horizon to create a VM without using a password, OpenStack generates a password for the VM without informing you of the password. Therefore, the default value is `false`.

- `vds_name=primary-vds0` specifies the Oracle VM Server for SPARC virtual disk server in the control domain. The default value is `primary-vds0`.
- `zvol_base=rpool/vm_disks` specifies the ZFS dataset location in which to store the Oracle VM Server for SPARC ZFS volumes. This property applies only when `create_zvols=true`.

Network Configuration Properties

This section provides reference information about the Neutron network configuration files.

Note - Version 1.0 of the Oracle VM Server for SPARC OpenStack Nova driver supports only the Neutron ML2 core plugin with the `ldoms` mechanism driver from the `openstack-ldoms-controller` package.

The Oracle Solaris zones OpenStack Nova driver uses the Neutron elastic virtual switch (EVS) core plugin at this time, which is not compatible with the Oracle VM Server for SPARC OpenStack Nova driver. Thus, an existing OpenStack controller environment that manages Oracle Solaris zones is not compatible with the Oracle VM Server for SPARC OpenStack Nova driver at this time.

Neutron for the Cloud Controller

This section provides reference information about Neutron configuration properties for the OpenStack cloud controller `/etc/neutron/neutron.conf` configuration file.

Neutron for the Cloud Controller `/etc/neutron/neutron.conf` Configuration File

The following sample `/etc/neutron/neutron.conf` configuration file fragment shows the properties for Neutron:

```
[DEFAULT]
core_plugin = ml2
service_plugins = router
api_workers = 16
nova_url=http://IP-address:8774/v2
nova_admin_username = nova
```

```
nova_admin_tenant_name = service
nova_admin_password = services-password
nova_admin_auth_url = http://IP-address:5000/v2.0
rabbit_host=IP-address

[matchmaker_redis]
host=IP-address

[keystone_authtoken]
auth_uri = http://IP-address:5000/v2.0
identity_uri = http://IP-address:35357
admin_tenant_name = service
admin_user = neutron
admin_password = services-password
signing_dir = $state_path/keystone-signing

[database]
connection = mysql://neutron:services-password@FQDN-controller-hostname/neutron
```

Neutron for the Cloud Controller `/etc/neutron/api-paste.ini` File

The following sample `/etc/neutron/api-paste.ini` file shows the required properties for Neutron:

```
[filter:authtoken]
paste.filter_factory = keystonemiddleware.auth_token:filter_factory
admin_tenant_name = service
admin_user = neutron
admin_password = services-password
auth_uri = http://IP-address:5000/v2.0
identity_uri = http://IP-address:35357
```

Neutron for the Cloud Controller `/etc/neutron/plugins/ml2/ml2_conf.ini` File

You might be able to use the default `ml2_conf.ini` file as-is. On the cloud controller, ensure that the `[ml2]` stanza of the `ml2_conf.ini` includes at least the following lines:

```
[ml2]
type_drivers = flat,vlan
tenant_network_types = flat,vlan
```

```
mechanism_drivers = ldoms

[ml2_type_flat]
flat_networks = physnet1,physnet2,physnet3,physnet4

[ml2_type_vlan]
network_vlan_ranges = physnet1:1:4096,physnet2:1:4096,physnet3:1:4096,physnet4:1:4096
```

ML2 for the Oracle VM Server for SPARC OpenStack Compute Node

This section provides reference information about ML2 networking configuration properties for the Oracle VM Server for SPARC OpenStack compute node `/etc/neutron/plugins/ml2/ml2_conf.ini` configuration file.

ML2 Configuration File for Compute Nodes

The following sample `/etc/neutron/plugins/ml2/ml2_conf.ini` configuration file fragment shows the properties for ML2 networking and the Oracle VM Server for SPARC virtual switch.

```
[ml2]
path_mtu = 1500
physical_network_mtus = physnet1:MTU-value1,physnet2:MTU-value2

[ldomsvsw]
physical_vsw_mapping=physnet1:primary-vsw0, physnet2:primary-vsw1
netboot_segments=1
default_vsw=primary-vsw0
```

The following are the configuration properties for the `[ml2]` and `[ldomsvsw]` stanzas of the `/etc/neutron/plugins/ml2/ml2_conf.ini` file:

- `default_vsw=primary-vsw0` specifies the default Oracle VM Server for SPARC virtual switch name.
- `netboot_segments=1,2,5:10,30:50` is a comma-separated list of netboot segments (individual, or ranges) by VLAN ID. These values specify the segments on which you set up a netboot server that the VMs can use to boot from the network. Range values are separated by a colon character (`:`).
- `path_mtu=1500` specifies the default MTU value for a physical network device. The default value is `1500`.

- `physical_network_mtus=physnet1:1500,physnet2:9000` maps an ML2 physical network name to its associated MTU. Use commas to separate mappings.
- `physical_vsw_mapping=physnet1:`*primary-vsw1*`,physnet2:`*primary-vsw2* maps an ML2 physical network name to its associated Oracle VM Server for SPARC virtual network switch name. Use commas to separate mappings.

If you do not map a physical network name to an Oracle VM Server for SPARC OpenStack virtual switch name, the physical network falls back to using the virtual switch that is specified by the `default_vsw` property. If the `default_vsw` property is not specified, it falls back to using the default virtual switch, which is `primary-vsw0`.

♦ ♦ ♦ CHAPTER 5

Administering the Oracle VM Server for SPARC OpenStack Compute Node

This chapter covers the following topics:

Creating Networks

To create a Neutron network on a physical network, perform the following steps on a Neutron controller node:

```
cctrl# TENANT=`keystone tenant-get demo|grep id|awk '{print $4}'`
cctrl# VLAN_ID=1
cctrl# CIDR=IP-address/22
cctrl# GATEWAY=IP-address
cctrl# SUBNET_NAME=subnet-name
cctrl# NETWORK_NAME=network-name
cctrl# START_IP=start-IP-address
cctrl# END_IP=end-IP-address
cctrl# ML2_PHYSNET=physnetN
cctrl# export OS_USERNAME=neutron
cctrl# neutron net-create --provider:network_type=vlan \
--provider:segmentation_id=${VLAN_ID} --provider:physical_network=${ML2_PHYSNET} \
--tenant-id ${TENANT} ${NETWORK_NAME}
cctrl# neutron subnet-create --disable-dhcp --gateway ${GATEWAY} \
--name ${SUBNET_NAME} --allocation-pool start=${START_IP},end=${END_IP} \
```

```
--tenant-id ${TENANT} ${NETWORK_NAME} ${CIDR}
```

The following example shows how to create a Neutron network on a physical network called `physnet1`.

Note - The physical network name you use must be a name that aligns to the physical networks that are specified in your `ml2_conf.ini` files on both the cloud controller and compute nodes. The term `default` for the physical network will fail when attempting to create networks.

The following example sets up a simple 192.168.0.0/24 network, with a gateway (default router) of 192.168.0.1. IP addresses 192.168.0.100 - 192.168.0.200 are available for logical domain guests. The VLAN ID is 500 and assumes the `demo` tenant.

Note - OpenStack uses CIDR notation to define subnets.

```
cctrl# TENANT=`keystone tenant-get demo|grep id|awk '{print $4}'`
cctrl# VLAN_ID=500
cctrl# CIDR=192.168.0.0/24
cctrl# GATEWAY=10.0.68.1
cctrl# SUBNET_NAME=private_subnet
cctrl# NETWORK_NAME=private_network
cctrl# START_IP=192.168.0.100
cctrl# END_IP=192.168.0.200
cctrl# ML2_PHYSNET=physnet1
cctrl# export OS_USERNAME=neutron
cctrl# export OS_PASSWORD=services-password
cctrl# export OS_TENANT_NAME=service
cctrl# export OS_AUTH_URL=http://cloud-controller-IP-address:5000/v2.0
cctrl# neutron net-create --provider:network_type=vlan \
--provider:segmentation_id=${VLAN_ID} \
--provider:physical_network=${ML2_PHYSNET} --tenant-id ${TENANT} ${NETWORK_NAME}

cctrl# neutron subnet-create --disable-dhcp --gateway ${GATEWAY} \
--name ${SUBNET_NAME} --allocation-pool start=${START_IP},end=${END_IP} \
--tenant-id ${TENANT} ${NETWORK_NAME} ${CIDR}
```

Configuring Distributed Lock Management

The Distributed Lock Manager (DLM) implementation prevents the same VM from running on multiple compute nodes simultaneously. This situation might occur when a VM is rebuilt on another system during a temporary hardware failure or hardware maintenance and the initial

system that houses that VM is restored to service. DLM places a lock on each VM when it runs on a compute node, which prevents other compute nodes from running that VM. The Nova evacuate capability is not supported when DML is not installed and configured.

Configuring DLM on the Oracle VM Server for SPARC OpenStack compute node requires a specifically configured NFS server and changes to the `/etc/nova/nova.conf` file. See "Compute Node `/etc/nova/nova.conf` Configuration File" on page 27.

Note - The NFS clients are the compute nodes that participate in the cluster.

Configuring an NFSv4 Server on the Oracle Solaris OS for Use With Distributed Lock Management

The following example shows how to configure a single Solaris host as the NFSv4 server. The ZFS file system is on `saspool`, the share is called `nfspool`, and the share name is `/saspool/nfspool`. The IP addresses of your NFS clients are 10.0.68.20 and 10.0.68.22. This example also configures the NFS domain on all hosts.

Note - The following commands show how to use a single NFSv4 server for DLM. This configuration is not intended for use in a production environment. It is best to use a ZFS appliance in a dual-node, high-availability configuration or a similar highly available NFSv4 server.

```
cctrl# sharectl set -p server_versmin=4 nfs
cctrl# sharectl set -p server_delegation=off nfs
cctrl# zfs create -o quota=1g saspool/nfspool
cctrl# zfs set share=name=nfspool,path=/saspool/nfspool,prot=nfs,sec=sys,rw=@10.0.68.20
/32:@10.0.68.22/32,root=@10.0.68.20/32:@10.0.68.22/32,anon=0 saspool/nfspool
cctrl# zfs share.nfs=on saspool/nfspool
cctrl# chown 85:85 /saspool/nfspool # Nova user's UID
cctrl# sharectl set -p nfsmapid_domain=domain-name nfs
nova# sharectl set -p nfsmapid_domain=domain-name nfs
```

Note - You must specify the `nfsmapid_domain` property on both the NFSv4 server and on all the compute nodes.

Taking the Distributed Lock Management NFSv4 Server Offline for Maintenance

If you need to take the DLM's NFSv4 server down for an extended amount of time, first disable DLM on each of your compute nodes.

Edit the `/etc/nova/nova.conf` configuration file to comment out the `dlm_nfs_server` entry. For example:

```
#dlm_nfs_server=10.10.68.61
```

Restart the `nova-compute` service on all your compute nodes.

```
nova# svcadm restart nova-compute
```

DLM exits the cluster on all nodes and no longer attempts to fence the nodes on loss of connectivity to the NFS share.

When the NFS server is ready to come online, reverse the previous steps by removing the comment character (`#`) from the beginning of the `dlm_nfs_server` entry and restarting the `nova-compute` service.

Note that the Nova evacuate operation is not supported while the DLM cluster is offline.

Creating Oracle VM Server for SPARC OpenStack Compute Node Flavors

An OpenStack Nova compute node *flavor* is a preconfigured combination of settings for an OpenStack Oracle VM Server for SPARC VM. The flavors serve as a service catalog mechanism for OpenStack.

Configuring Flavors for Oracle VM Server for SPARC OpenStack Compute Nodes

Use the `nova flavor-create` command to create a flavor:

```
cctrl# nova flavor-create "flavor-name" flavor-ID memory-size disk-size vCPU-count
```

Where you must specify the operands in the following order:

- *flavor-name* is the name of the flavor such as `"LDom.medium"`
- *flavor-ID* is a unique ID number
- *memory-size* is the amount of RAM in Mbytes
- *disk-size* is the amount of disk space in Gbytes
- *vCPU-count* is the number of virtual CPUs (strands)

By using the `--ephemeral` *disk-size* option, you can optionally add a secondary ephemeral disk with *disk-size* Gbytes to a logical domain for the last flavor. This disk is a blank disk attached to the domain that uses local storage on the compute node. As such, the use of ephemeral storage blocks live migrations. To perform live migrations, use additional Cinder volumes instead of using the ephemeral storage flag on your flavors.

Customizing Oracle VM Server for SPARC OpenStack Compute Node Flavors

Version 1.0 of the Oracle VM Server for SPARC OpenStack Nova driver permits you to set “extra_specs” on flavors to specify how to configure logical domains.

You can set these extra_specs by using the `nova flavor-key` command.

```
cctrl# nova flavor-key flavor-ID set keyword=value [ keyword=value]...
```

Available extra_specs are as follows:

- `ldm:whole-core` is a Boolean value that specifies whether to use the whole-core constraint.
- `ldm:max-core` is a Boolean value that specifies whether to use the maximum number of cores constraint.
- `ldm:live-migrate` is a Boolean value that specifies whether to permit live migration for this VM.
- `ldm:cpu-arch` is a string value that specifies the migration class value of the `cpu-arch` property. See “Domain Migration Requirements for CPUs” in *Oracle VM Server for SPARC 3.4 Administration Guide*.
- `ldm:auto-alt-mac-addrs` is an integer value that specifies how many alternate MAC addresses to configure per virtual network for this VM. Use this if you want to create VNICs within a guest domain.

Note - The whole-core and max-core constraints require that the virtual CPUs allocated to the flavor align on a boundary that matches the underlying platform's strands per core. For example, a core on a SPARC T7 series server has eight strands (virtual CPUs). So, to ensure that these constraints work properly, the number of virtual CPUs for the flavor must be a multiple of eight.

Flavor Restrictions

- Avoid creating a pool of servers that includes both Fujitsu M10 servers and Oracle SPARC servers when using the whole-core constraint.

  Whole core and max core constraints depend on the number of virtual CPUs (or strands) provided by the underlying SPARC server.

  For example, a SPARC T7 series server has eight strands per core. So, setting the whole core constraint on a flavor requires that the number of virtual CPUs that you specify is a multiple of eight. While a Fujitsu M10 server has only two strands per core so the flavor requires that the number of virtual CPUs on a Fujitsu M10 server must be divisible by two.
- Avoid mixing whole-core and non-whole core configurations as partial CPU allocation might fragment the cores that remain available on a server over time.

  For example, a SPARC T7-2 series server might have 30 domains provisioned, each using two strands. If four domains are deleted, eight strands can be freed, however these strands might come from different physical cores. In this instance, a single whole core might be available. In such a case, the Nova driver and the underlying Oracle VM Server for SPARC hypervisor are unable to service a request for CPU resources. The `nova-scheduler` service might attempt to schedule this VM on another node.

Creating and Uploading Glance Images

This section describes the process to prepare a special-purpose logical domain for use as the source of a “golden OS image.” This process is deliberately destructive to the domain's configuration. This process unconfigures system components such as network addresses, routes, host names, DNS entries, and so on.

After you install the `simple-init` tool and complete the following task, relevant portions of the domain's unique configuration are removed so that the golden image can be captured and re-deployed as an OpenStack guest domain. The `simple-init` tool starts automatically on the initial boot and configures the guest domain. `simple-init` uses metadata provided by

OpenStack to configure the logical domain's host name, IP addresses, subnet masks, default gateways, DNS name servers and so on.

Caution - Do not use this process on live, active, or production systems. This process is destructive to OS configuration because it clears information such as IP addresses, routes, host names, and so on.

After you perform this process on a guest domain, you cannot use that guest domain for general purposes.

Golden Image Limitations

- It is best when the root disk is at least eight Gbytes in size. Smaller disks can produce small cylinder sizes and this situation might limit the expansion potential of an image to 18.75 Gbytes.
- Ensure that you use a smaller root disk than the root disk size specified by any flavor you will use to deliver the captured golden image. Namely, if you plan to deliver VM root disks that are a minimum of 16 Gbytes in size, use an 8 Gbyte to 15 Gbyte root for your golden image.
- The ability to expand the root zpool depends on the disk layout. Slice expansion is attempted only for the largest slice that is followed immediately (contiguously) by free space. The contents of the slice are not considered when selecting which slice to expand.
- Slice expansion might not be possible when using Linux for SPARC 1.0 if the image uses a GPT or EFI label.
- You might receive a `vfs_mountroot: cannot mount root` error message or a similar panic on re-deployment if a LUN that has previously been used for ZFS has been overwritten with an image that you plan to capture. To avoid this issue, use empty LUNs, ZFS volumes, or files. However, if you cannot create a image in this way, ensure that the first and last ten MBytes of the disk have been cleared before installing the Oracle Solaris OS onto a LUN that you plan to capture for use as a golden image.
- You can use only one disk when constructing a golden image.

▼ How to Create a Golden OS Image for Glance

Caution - Only perform Steps 3-6 of this procedure from the console of the special-purpose logical domain that will be the source of the golden OS image. This process removes the domain's network configuration and thus isolates it from the network and interrupts any session that relies on a direct network connection to this domain.

1. **Create a guest domain that uses only a single network interface such as `net0`, `vnet0`, or `eth0`. The network interface must have a static IP configuration.**

   See "How to Create and Start a Guest Domain" in *Oracle VM Server for SPARC 3.4 Administration Guide*.

2. **Attach the `simple-init` ISO image to your guest domain.**

```
nova# ldm add-vdsdev options=ro,slice /path-to-iso/simple-init-1.0.iso \
simple-init@primary-vds0
nova# ldm add-vdisk simple-init simple-init@primary-vds0 your-new-ldom
```

3. **Mount the image in the guest domain.**

   - **Oracle Solaris OS:**

```
golden# mount -F hsfs /dev/dsk/c1d1s0 /mnt    # or use c0d1s0 if Solaris 10
```

   - **Linux for SPARC 1.0:**

```
golden# mount -L simple-init /mnt
```

4. **Install the guest package.**

   This package mounts the config drive at boot and executes the driver's initialization instructions provided on the ConfigDrive by means of a payload file.

```
golden# cd /mnt; ./setup
```

5. **Remove the existing unique configuration from the special-purpose, logical domain that will be the source of a golden OS image.**

   - **Oracle Solaris 11:**

```
golden# ipadm delete-ip net0
golden# netadm enable -p ncp DefaultFixed
golden# rm /etc/defaultrouter
golden# route -p flush
golden# rm /etc/ssh/ssh_host_*
golden# nscfg unconfig svc:/network/dns/client:default
golden# svcadm refresh svc:/network/dns/client:default
golden# rm -f /etc/resolv.conf
golden# svcadm disable svc:/network/dns/client:default
golden# svccfg -s system/identity:node setprop config/nodename="openstack-build"
golden# svccfg -s system/identity:node refresh
```

   - **Oracle Solaris 10:**

```
golden# rm /etc/hostname.* /etc/dhcp.*
golden# rm /etc/defaultrouter /etc/resolv.conf
golden# route -p flush
golden# rm /etc/ssh/ssh_host_*
golden# echo "openstack-build" > /etc/nodename
```

- **Linux for SPARC 1.0:**

```
golden# rm -f /etc/sysconfig/network-scripts/ifcfg-eth*
golden# rm -f /etc/sysconfig/network-scripts/route-eth*
golden# rm -f /etc/resolv.conf
golden# rm -f /etc/ssh/ssh_host_*
golden# rm -f /etc/udev/rules.d/70-persistent-net.rules
golden# echo "HOSTNAME=openstack-build" > /etc/sysconfig/network
```

6. **Perform a clean shutdown of the guest domain.**

- **Oracle Solaris OS:**

```
golden# shutdown -i5 -g0 -y
```

- **Linux for SPARC 1.0:**

```
golden# shutdown -h now
```

7. **Find the guest domain disk's back end volume.**

For example, the disk volume is `myldom-vol10`.

```
nova# ldm list -o disk primary | grep myldom-vol0
myldom-vol0 /dev/zvol/dsk/ldompool/myldom-vol0
```

8. **Capture the disk image to a file.**

Even if the guest domain backend volume is a block device (`/dev/dsk`), the `gdd` command requires the corresponding character device (`/dev/rdsk`) for the image capture. Also use the appropriate whole disk device link that ends in d*N*`s2` (slice 2) for devices that have a VTOC label, and d*N* (the disk number) for devices that have an EFI label.

For example, the `myldom-vol10` disk volume is the input file and the output file is the `sol11_3s12_simp-init.img` image.

```
nova# gdd if=/dev/zvol/rdsk/ldompool/myldom-vol0 of=sol11_3s12_simp-init.img \
bs=1048576 oflag=nocache conv=sparse
```

How to Create a WAN Boot Configuration Image for Glance

The `ldoms` driver supports only WAN boot for network booting.

To use WAN boot to install a machine, you must create a WAN boot configuration image by using the `mkwanbootcfg` utility. This utility is included in the Nova driver package.

```
/opt/openstack-ldoms/bin/mkwanbootcfg [-h] --output-file filename --url WANboot-file
 [--client-id WANboot-client-ID] [--hostname WANboot-hostname]
 [--http-proxy WANboot-proxy] [--tftp-retries WANboot-TFTP-retries]
 [--overwrite] [--version]
```

Only the `--output-file` and `--output-file` options are required.

- **Create a WAN boot configuration image for Glance.**

```
nova# /opt/openstack-ldoms/bin/mkwanbootcfg --output-file /var/tmp/s11_wanboot.img \
--url http://10.0.241.223:5555/cgi-bin/wanboot-cgi
Wanboot configuration /var/tmp/s11_wanboot.img image now available to import into glance
```

How to Upload an Image to Glance on the Cloud Controller

1. **As superuser, source the `.profile` file.**

```
cctrl# . ~/.profile
```

2. **Upload the golden image.**

```
cctrl# export OS_USERNAME=glance
cctrl# export OS_PASSWORD=services-password
cctrl# export OS_TENANT_NAME=service
cctrl# export OS_AUTH_URL=http://localhost:5000/v2.0
cctrl# glance image-create --container-format bare --disk-format raw --is-public true \
--property architecture=sparc64 --property hypervisor_type=ldoms \
--property vm_mode=ldoms --name "image-description" < /var/tmp/Oracle Solaris OS-version-name.img
```

For example, the following command uploads an Oracle Solaris 11.3 image:

```
cctrl# glance image-create --container-format bare --disk-format raw --is-public true \
--property architecture=sparc64 --property hypervisor_type=ldoms \
--property vm_mode=ldoms --name "LDom: Solaris 11.3" < /var/tmp/solaris11.3.img
```

Configuring a Serial Console

How to Switch from VNC to a Serial Console

Ensure that you have enabled the serial console on the cloud controller where the `nova-serialproxy` service is running before you enable the serial console on the compute node.

These steps are performed by default when you run the `setup.sh` script except for the value of the `enabled` keyword, which you must set manually.

1. **Modify the `[serial_console]` stanza of the `/etc/nova/nova.conf` configuration file on each compute node.**

   IP-address is the IP address of the cloud controller.

   ```
   [serial_console]
   serialproxy_host=IP-address
   serialproxy_port=6083
   enabled=true
   base_url=ws://IP-address:6083/
   listen=$my_ip
   proxyclient_address=$my_ip
   ```

2. **Restart the `nova-compute` service.**

   ```
   nova# svcadm restart nova-compute
   ```

How to Enable or Disable the Experimental Serial Console

- **Enable or disable the serial console on the cloud controller.**
  - **Enable the serial console.**

```
cctrl# /opt/openstack-ldoms/bin/experimental_serial.sh enable
```

- **Disable the serial console.**

```
cctrl# /opt/openstack-ldoms/bin/experimental_serial.sh disable
```

♦ ♦ ♦ CHAPTER 6

Troubleshooting an Oracle VM Server for SPARC OpenStack Compute Node

This chapter describes troubleshooting an Oracle VM Server for SPARC OpenStack compute node.

Troubleshooting VM Deployment Issues

Error: `No valid host was found`

You might see the `no valid host was found` error message when a VM deployment fails. This problem might be a partial deployment failure even though a node should be able to satisfy the request based on available resources and hypervisor type required to deploy this image.

To determine the root cause, ensure that the compute node is in debug mode by setting `debug=true` and `verbose=true` in the `/etc/nova/nova.conf` file.

If debug mode is disabled, add the following lines to the `/etc/nova/nova.conf` file:

```
debug=true
verbose=true
```

Restart the `nova-compute` service:

```
nova# svcadm restart nova-compute
```

For each compute node, search for `DEBUG: run_method: spawn()` in the compute node's log to determine whether the compute node received the request. Run this command on your active compute nodes.

```
nova# tail -f `svcs -L nova-compute` | grep "DEBUG: run_method: spawn()"
```

If you see a line like the following, the request is reaching the compute node and you have identified the likely location of the problem:

```
2016-07-07 13:48:58.319 29186 DEBUG nova.virt.ldoms.driver [req-1440679a-771d-4e21-
aca7-7b42f6a35648 d225a5a7434f4685a9f47326a2e5ff9f 3255d9556a354e8589b9a0a8475d7c0e - -
 -] DEBUG: run_method: spawn() spawn /usr/lib/python2.7/vendor-packages/nova/virt/ldoms/
driver.py:954
```

If a `spawn()` occurs, continue to debug the problem. See "Troubleshooting Nova Compute Service Issues" on page 48.

If you receive the `No valid host found` error but never see a `spawn()` line, the problem is probably on the cloud controller. To continue debugging this failure, see "Troubleshooting Other OpenStack Issues" on page 50.

Troubleshooting Nova Compute Service Issues

This section describes how to troubleshoot problems with the Nova driver.

Understanding Log Output From the Nova Driver

The Oracle VM Server for SPARC OpenStack Nova driver provides different levels of detail based on the `debug` and `verbose` property values in `nova.conf`.

It is best to have `verbose=true`. You can also set `debug=true` to assist in troubleshooting an issue.

- When in debug mode, the Oracle VM Server for SPARC OpenStack Nova driver provides trace messages to help identify precisely which method is being run at any given time. These messages can help you identify the cause of an issue that you encounter. To find a method entry, search for `method_run:` in the `nova-compute` service (SMF) log. To find a method return, search for `method_return:` in the `nova-compute` service log.
- View other log messages from the driver. The driver also provides other log output, which begins with `DEBUG:`, `WARNING:`, `ERROR:`, `EXCEPTION:`, or `INFO:`.

  Note that the Nova debug driver log entries start with a line such as the following:

  ```
  2016-07-07 15:14:51.404 29186 DEBUG nova.virt.ldoms.driver
  ```
- The `nova-compute` service itself provides `TRACE` messages if an unhandled exception has occurred. Search for `TRACE` messages to identify the root cause of a problem.

- Ensure that the driver started as expected. When you start the `nova-compute` service, ensure that it starts properly. When the driver starts or restarts, you should see lines in the log that look similar to the following:

```
2016-07-07 15:20:14.011 1098 DEBUG nova.service [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] Creating RPC server for service compute
 start /usr/lib/python2.7/vendor-packages/nova/service.py:188
2016-07-07 15:20:14.013 1098 INFO oslo_messaging._drivers.impl_rabbit [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] Connecting to AMQP server on 10.0.68.21:5672
2016-07-07 15:20:14.026 1098 INFO oslo_messaging._drivers.impl_rabbit [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] Connected to AMQP server on 10.0.68.21:5672
2016-07-07 15:20:14.033 1098 DEBUG nova.service [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] Join ServiceGroup membership for this service
 compute start /usr/lib/python2.7/vendor-packages/nova/service.py:206
2016-07-07 15:20:14.033 1098 DEBUG nova.servicegroup.drivers.db [req-d8973f20-
af93-4de6-9732-9e4a0ab6c61a - - - - -] DB_Driver: join new ServiceGroup member
 10.0.68.22 to the compute group, service = <nova.service.Service object at
 0xea967cf0> join /usr/lib/python2.7/vendor-packages/nova/servicegroup/drivers/
db.py:59
```

  If you do not see a connection to AMQP and the `DB_Driver: join new ServiceGroup member` message, ensure that the driver has not stopped because of a configuration issue. If the problem is a configuration issue, you will see an exception starting with `EXCEPTION` or `ERROR` in the early startup log.

  If the driver does not detect configuration issues, ensure that your MTU for the management network is the same everywhere, that NTP is configured, and name resolution is working.

- When in debug mode (`debug=true`), `DEBUG` messages from the driver are written to the Nova compute manager logs. The Nova driver has `run_method` and `method_return` traces. You can follow these traces for all methods that the driver runs. These traces help you to locate precisely where a problem occurs and shows all preceding steps that led to the problem.

  You can use the following commands to simplify the debugging process when in debug mode or with a debug log file:

```
nova# tail -f `svcs -L nova-compute` | egrep 'DEBUG: run_method:|TRACE|EXCEPTION'
```

  Or, run a trace that includes `method_return` with the return value:

```
nova# tail -f `svcs -L nova-compute` \
| egrep 'DEBUG: run_method:|DEBUG: method_return:|TRACE|EXCEPTION'
```

  The following command performs a trace and excludes DLM and other PERIODIC tasks:

```
nova# tail -f `svcs -L nova-compute` | egrep 'DEBUG: run_method:|TRACE|EXCEPTION' \
| egrep -v 'PERIODIC|dlm'
```

Troubleshooting Other OpenStack Issues

Cannot Log In To Horizon Because CSRF Verification Failed

After you perform a `pkg upgrade` command, your attempt to log in to the Horizon dashboard might fail with the following error message:

```
CSRF verification failed. Request aborted.
```

To work around this issue, run the following command on the cloud controller:

```
cctrl# gsed -i -e s@SECURE_PROXY_SSL_HEADER@#SECURE_PROXY_SSL_HEADER@ \
       -e s@CSRF_COOKIE_SECURE@#CSRF_COOKIE_SECURE@ \
       -e s@SESSION_COOKIE_SECURE@#SESSION_COOKIE_SECURE@ \
       /etc/openstack_dashboard/local_settings.py
```

Restart the `apache24` service.

```
cctrl# svcadm restart apache24
```

Clear your browser cache.

Access the Horizon dashboard at `http://`*your-cloud-controller*`/horizon`.

Important Things to Validate In Your Environment

- Ensure that all compute nodes use NTP. OpenStack depends on accurate time to function properly.
- Ensure that name resolution, forward and reverse, functions properly by using a standard `/etc/hosts` file that contains information about the compute nodes in your OpenStack environment. You can also have a fully functioning DNS with accurate records and appropriate search paths on hosts and can be used for production.
- Ensure that your system is running at least the Oracle Solaris 11.3 SRU 12 OS.
- Ensure that no services on the cloud controller or compute nodes have failed.

  Use the `svcs` command to view the status of all services. Run it on the cloud controller and on the compute nodes.

  ```
  # svcs -xv
  ```

♦ ♦ ♦ CHAPTER 7

Release Notes

This chapter includes known issues about this release of the Oracle VM Server for SPARC OpenStack compute node.

Known Issues

Using an “All-in-One” OpenStack Configuration

An OpenStack “all-in-one” configuration is a topology where the cloud controller and all other logical nodes, such as nova compute nodes, co-exist in the same control domain. You might encounter some problems when using such a topology.

Caution - Do not use an all-in-one configuration for important or critical environments, such as production environments.

If you must set up an all-in-one configuration, consider and address the following issues that might affect your environment:

- Resolve the port conflicts between the Oracle VM Server for SPARC virtual console concentrator (VCC) service and the OpenStack keystone service.

  An all-in-one configuration might cause a port conflict at one of the following times:

  - Upon deployment of the first guest domain on this machine
  - Immediately if a guest domain has already been deployed

  This conflict might occur because both the OpenStack keystone service and the Oracle VM Server for SPARC VCC service use port 5000 by default.

  To work around this conflict, specify a VCC port range that starts above port 5000 when you configure the control domain:

```
# ldm set-vcc port-range=5001-5100 vcc-name
```

Note - To determine the VCC name, run the `ldm list-services` command.

- Install and set up the all-in-one configuration in the following order to avoid common problems such as accidentally overwriting OpenStack configuration files.

  1. Install the `openstack-controller-ldoms` package.
  2. Install the `nova-ldoms` package.
  3. Run the `/opt/openstack-ldoms/bin/demo-controller-setup.sh` script.
  4. Run the `/opt/openstack-ldoms/bin/setup.sh` script.
  5. Customize the configuration, as needed.
  6. Reboot the system.
  7. Ensure that all enabled OpenStack services start correctly.

Cannot Type in the Console Window for a VM

There is an OpenStack console-focus issue, not specific to the Oracle VM Server for SPARC OpenStack `nova` driver.

To address this issue, click the blue bar at the top of the console window.

Cannot Deploy EFI Images to Older Hardware

Some older servers (such as UltraSPARC T2 servers) do not support EFI labels. As such, you must create a VTOC based on VM images to support old and new hardware. This issue also imposes disk size limitations.

Cannot Set `cpu-arch` Property Value After Deployment

If the `cpu-arch` property is set on a VM, the `nova` driver cannot change the `cpu-arch` property value later. This issue occurs because flavor migration is not yet supported by the Oracle VM Server for SPARC OpenStack Nova driver.

Oracle Solaris 10 Guest Domains: Automated Disk Expansion Is Supported Only With the ZFS Root

You must use a ZFS root to use the automated disk expansion capability. File system and volume managers such as UFS, SVM, and VxFS are not supported by this capability.

Linux for SPARC Does Not Support All Oracle VM Server for SPARC Features

The following Oracle VM Server for SPARC features do not work with a guest domain that is running Linux for SPARC 1.0:

- Dynamic volume attachment and detachment
- Dynamic network attachment and detachment
- Live migration

Console Logs Are Not available After a Live Migration

The `vntsd` console log is not migrated with the guest domain. As a result, these console logs are no longer available and only recent log entries appear.

Mismatched MTUs on the Management Network Can Be Problematic

You might experience problems with the message queue or other OpenStack services if your controller and compute nodes have mismatched MTUs on their management interfaces. These management interfaces are used for OpenStack management communications. A mismatched MTU configuration might have a compute node management network of 9000 bytes and controller node of 1500 bytes. Ensure that all hosts are aligned on their management network in terms of the MTU.

Avoid Inline Comments in OpenStack Configuration Files

You might experience problems if you add a comment (`#` to the end of a configuration line in an OpenStack configuration file. OpenStack interprets inline comments as part of the value.

Ensure that you place comments on lines by themselves and that the comment line begins with a comment symbol (`#`).

For example, the `admin_password=welcome1 #my password` configuration line is interpreted as specifying the password as `welcome1 #my password`.

Use the following line to check a configuration file for inline comments:

```
# cat /etc/service/service.conf |egrep -v '^#' | grep '#'
```

`nova-compute` Service Hangs at `Mounting NFS share` Stage

Ensure that the NFS server settings are correct. If you choose the wrong server, the `nova-compute` service appears to hang at boot while attempting to mount the NFS share.

To work around this problem, disable the `nova-compute` service and issue a `kill` to the mount that is being attempted to the wrong share. The driver might make an additional attempt to mount the share, so ensure that all attempts the driver makes to mount the wrong share are killed after the `nova-compute` service is disabled. Then, correct the `nova.conf` file and enable the `nova-compute` service.

“Rebuild” Does Not Actually Rebuild the VM

The rebuild operation is not yet supported by the Oracle VM Server for SPARC OpenStack Nova driver. Only the Nova evacuate operation is supported. If a user attempts to perform a rebuild operation, the VM's existing disk might be recycled and not “re-imaged.”

The `nova-compute` Service Times Out Waiting For Cinder to Create a LUN When You Run `create new volume`

If a Cinder volume is being created with an OS image on it, the OS image copy might take a long. Nova can time out waiting for Cinder to complete its task. The `nova-compute` service (outside of the Oracle VM Server for SPARC Nova driver) simply polls for a period of time and waits to see whether Cinder created the volume.

Consider increasing the following value if you experience these “hangs” in your environment:

```
block_device_allocate_retries=360
```

Then, restart the `nova-compute` service.

Compute Node Panics Because of DLM Fencing

If you experience problems accessing the NFSv4 share, a compute node might panic. If the NFSv4 share becomes unavailable, lags, or has other connectivity issues for ten minutes or more, the compute nodes fence themselves off by issuing a panic to the control domain. If you experience this problem frequently, disable DLM by commenting out the `dlm_nfs_server` entry while you identify the root cause of the issue.

Ensure that your NFSv4 storage is highly available and resilient. Also ensure that delegation is disabled.

After Installing the Controller Package, `neutron-server` Service Goes In To Maintenance Mode

This problem arises if the `neutron-server` service is configured for EVS rather than from ML2 and if the profile attempts to bring the `neutron-server` service online before it is properly configured.

To correct this issue, restart the `manifest-import` service and disable the `neutron-server` service by running these commands:

```
cctrl# svcadm restart manifest-import
cctrl# svcadm disable neutron-server
```

If you are configuring your cloud controller services manually, you must complete configuring your `/etc/neutron/neutron.conf` and `/etc/neutron/api-paste.ini` cloud controller files before you re-enable the `neutron-server` service.

`vfs_mountroot` Panic Occurs on the First Boot of a Guest Domain

If a `vfs_mountroot` panic occurs on the first boot of a guest domain, see "Golden Image Limitations" on page 41.

Serial Console Immediately Closes Its Connection

The serial console validates that the URL in the browser matches the configuration. If the serial console closes the connection immediately, you might need to change the `base_url` property value to match the host name of the controller node that you use to access the console. This name is likely to be the domain name of the system or another front end such as a load balancer or reverse proxy.

The `base_url` property is specified on compute nodes in the `/etc/nova/nova.conf`.

The following example `nova.conf` file shows the `base_url` property changed from an IP address to a host name that matches the controller name, *cloud-controller-hostname*:

```
[serial_console]
enabled=true
#base_url=ws://10.0.68.51:6083/
base_url=ws://cloud-controller-hostname:6083/
listen=$my_ip
proxyclient_address=$my_ip
serialproxy_host=10.0.68.51
```

After you make changes to the `nova.conf` file, restart the `nova-compute` service.

```
nova# svcadm restart nova-compute
```

Index