Solace Systems®

Installation Runbook for Solace Systems Virtual Message Router

Application Version	Solace Virtual Message Router (VMR) OpenStack (Cloud Edition) Version 7.1.1.327
MOS Version	7.0
OpenStack Version	Kilo
Application Type	Message Broker

Content

Document History

Version	Revision Date	Description
0.1	31-08-2015	Initial Version
0.2	24-09-2015	Feedback from Mirantis
0.3	17-02-2016	Updated for VMR Cloud Edition and MOS 7.0

1 Introduction

This document is to serve as a detailed Deployment Guide for the Virtual Message Router (VMR). Solace Systems offers the VMR (message broker solution) as an application level infrastructure to support VNF-to-VNF communications in a Network Function Virtualization (NFV) environment, VNF to OSS/BSS integration, as well as traditional enterprise messaging queueing, data ingestion-as-a-service for Big Data, and Web, Mobile, or IoT communications over the public Internet. This document describes the reference architecture, installation steps for certified MOS+VMR, limitations and testing procedures.

1.1 Target Audience

The audience that would use Solace VMR would include application developers familiar with coding directly to APIs for Publish/Subscribe (Pub/Sub), Request/Reply (REST), Message Queueing (MQ), Remote Procedure Calls (RPC), or Inter-Process Communications (IPC) middleware.

Another set of potential users of Solace VMR would be infrastructure operations managers that are deploying and configuring platform services for higher level applications and middleware to use as transport. This could include deploying Solace VMR as the underlying transport for an ESB, CEP Engine, Change Data Capture system, Database Replication, In-Memory Data Grid, Big Data Repository, Streaming Analytics Engine, IoT Device Server, M2M/SCADA Gateway, etc.

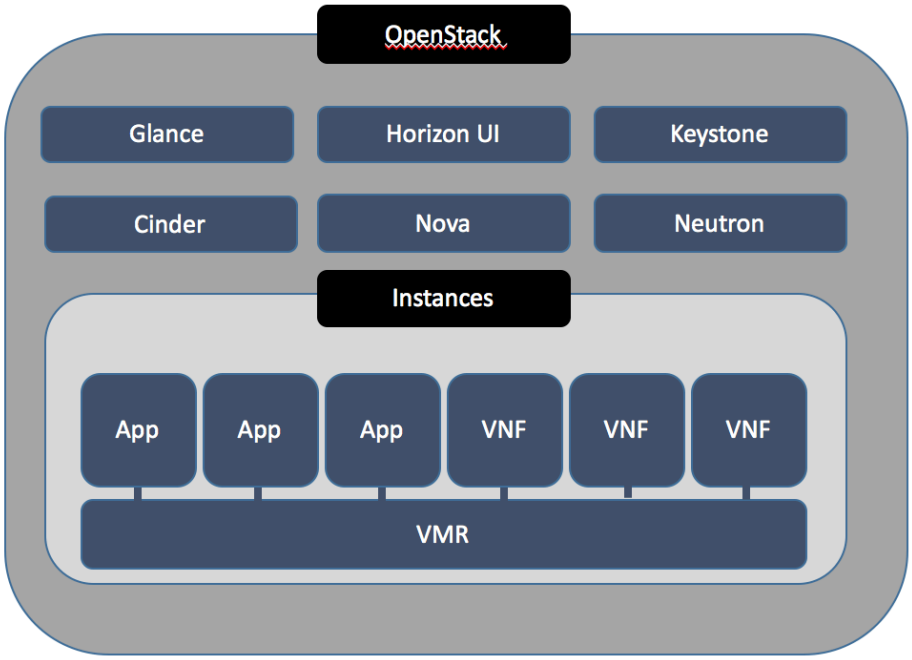
Although there is considerable overlap between the functionality of the Solace VMR and AMQP brokers such as RabbitMQ and Qpid, this runbook does not attempt to document the use of Solace VMR as a replacement for AMQP as used for RPC by Fuel, Nova, or any other component of the Mirantis OpenStack distribution. This runbook is strictly designed to enable application level execution of the Solace VMR for Application to Application or Device to Datacenter messaging.

2 Application overview

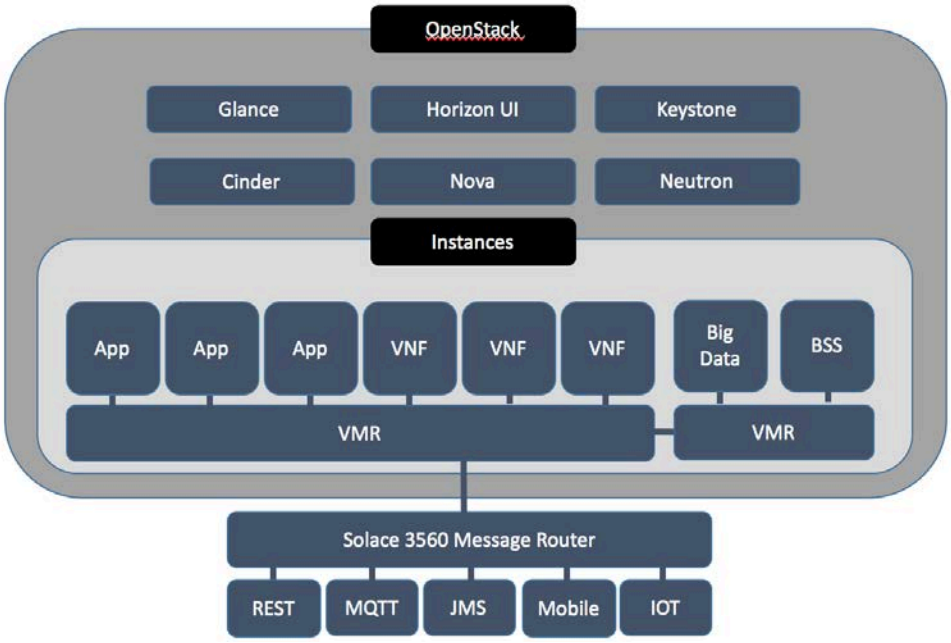
The Solace VMR is a completely self contained and turnkey carrier-class message broker that comes bundled with it's own SolOS Operating System (SolOS being a special hardened flavor of Linux with messaging functionality built-in). The VMR can run in either a stand alone or clustered environment. Clustering of VMR instances can be achieved using Solace Multi-Node Routing (MNR) or Solace Message VPN Bridging, but neither of these mechanisms requires any special setup within OpenStack and is configured entirely via the Solace Element Management Protocol (SEMP), CLI, or SolAdmin GUI.

3 Joint Reference Architecture

Solace VMR can run as a stand-alone message broker (hub and spoke) or as a networked cluster of of to 64 VMR instances in a single domain. Further scalability some from linking multiple domains using Solace Remote VPN bridge capability. Application and/or VNFs running as tenant applications within Nova can connect to the Solace VMR over a number of supported interfaces and protocols including REST, JMS, MQTT, WebSocket, HTTP, and the Solace Messaging Format (SMF). All connections are via TCP socket connections over Neutron provisioned virtual IP interfaces (both public and private). The Solace VMR will translate between all these supported transport protocols so for example, a publisher can send a REST message (HTTP POST) and a subscriber might listen to a JMS topic to receive the data.



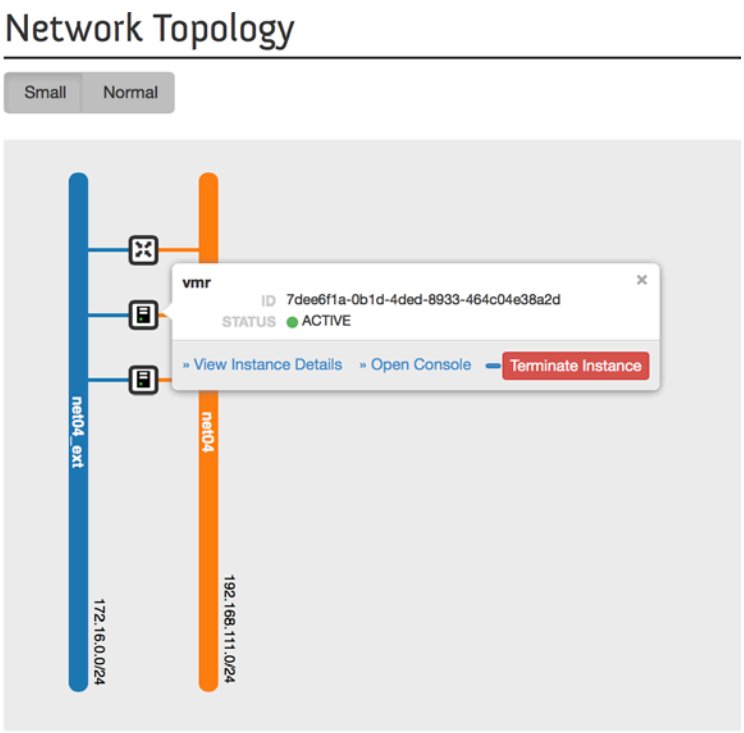
Solace VMR instances can also link with one or more VMRs running in another datacenter or outside the OpenStack environment. The VMR can also connect with Solace hardware appliances (such as the highest performance 3560) to form and seamless multi-node messaging fabric.



4 Physical & Logical Network Topology

The Solace VMR can run in a number of network topologies depending on the use case. For IOT, Web Streaming, or any other internet facing client connections, it important to have the VMR connected to a publically accessible external subnet. If HTTP or TLS termination is done by another VNF or proxy in front of the Solace VMR then it might not be required to have a public interface and security rules that allow external client connections.

In order to make the Solace VMR work easily in the POC setup, both an external and private interface was created as show in the following Network Topology diagram from the Horizon UI.



5 Installation & Configuration

5.1 Overview of MOS installation steps

Solace VMR requires only a minimum of 1 compute node to run a stand-alone VMR instance, and 1 storage node in which to store the VMR image(s) in a format compatible with the available hypervisor(s).

The high-level description of MOS installation steps are as follows:

- Download Mirantis 7.0 ISO from Mirantis website.
- Follow the installation instructions as specified in Mirantis OpenStack User Guide.
- Download the Solace Virtual Message Router
- Create a volume from the SolOS/VMR image
- Configure a Security Group which will allow external clients to connect into SolOS/VMR using the desired messaging wire protocols
- Launch a SolOS/VMR instance

- Connect into the running VMR using SSH, or SolAdmin GUI to continue setup as specified in the Solace Feature Provisioning Guide.
- Test that the Solace VMR is running by publishing and subscribing to a test topic using cURL, or any Open Source MQTT client tools such those provided by the Eclipse Paho Project (http://www.eclipse.org/paho/) or using the Solace-provided sdkperf benchmarking tools and sample client applications.
- (optional) start multiple instances of the Solace VMR and configure Solace Multi-Node-Routing or Remote VPN Bridging between the available VMR nodes to form a Solace messaging cluster/network.

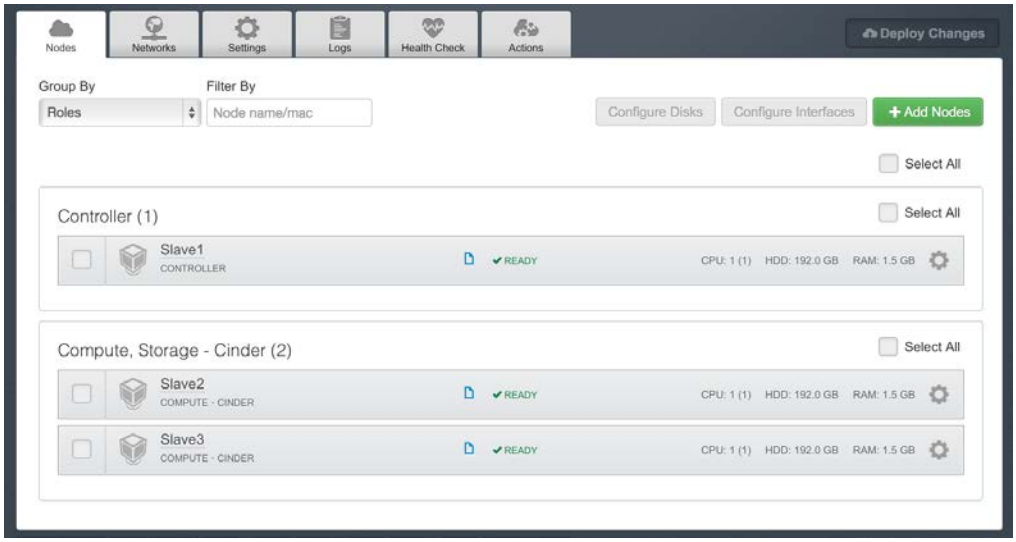
5.2 MOS installation in details

5.2.1 Creation of OpenStack environment

Solace VMR runs in the default environment which can be created as documented in the Mirantis OpenStack documentation (Create a new OpenStack environment section).

5.2.2 MOS Deployment

VMR requires only at least a single Controller node and two Compute nodes with enough CPU and Memory to run a standard m1.small flavor instance. For best performance Solace recommends 4 CPUs and 4GB of RAM for each instance of the VMR. 2 CPUs and 4 GB RAM is the minimum required to run.



OS	Mode	HV	Network	Storage
			VLAN	Ceph
UbuntuCentOS	HAHA	AnyAny	AnyAny	optionaloptional
Ubuntu	HA	Any	Any	optional

Comment [1]: MOS 7.0 has not Centos OS for slaves, please remove it

5.2.3 Health Check Results

No additional health checks are required beyond the basic network verification tests in Fuel.

5.3 Solace VMR installation steps

System Requirements

- 22 GB of disk space, 4 GB of RAM, and two CPUs minimum (4 CPUs recommended)

Download the Solace VMR and associated tools

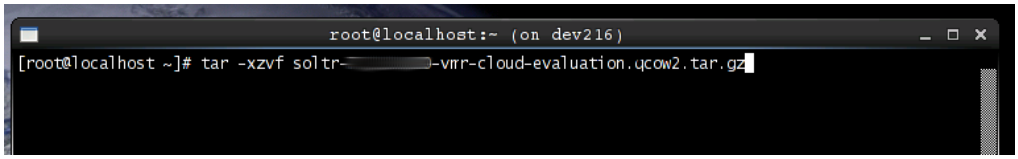
- On the Downloads page of dev.solacesystems.com, in the Products — SolAdmin section, click SolAdmin, then click the SolAdmin distribution for your administrative workstation's OS environment. This is optional, and only required if you want to use the thick client SolAdmin GUI in place of the command line interface (CLI). SolAdmin can run on any workstation or laptop and does not need to run in the OpenStack environment.

- On the Downloads page of dev.solacesystems.com, in the Tools section, click SDKperf, then click the SDKperf distribution for your client application OS environment. Use of SDKperf is optional and only required for benchmarking or otherwise generating pub/sub traffic without programming. SDKperf is a command line tool and does require a GUI environment to run. SDKperf can run on any workstation or laptop and does not need to run in the OpenStack environment.

- On the Downloads page of dev.solacesystems.com, in the Products — Cloud Images section, click Virtual Message Router, then click the Compressed QCOW2 Cloud Package for your environment.

- After you read and agree to the licence agreement, a compressed QCOW2 file called soltr-<version>-vmr-cloud-<type>.qcow2.tar.gz will be downloaded.

- Once you have the compressed QCOW2 package, you need to extract it so that you are left with the desired QCOW2 image. The compressed QCOW2 can be extracted using the tar command.



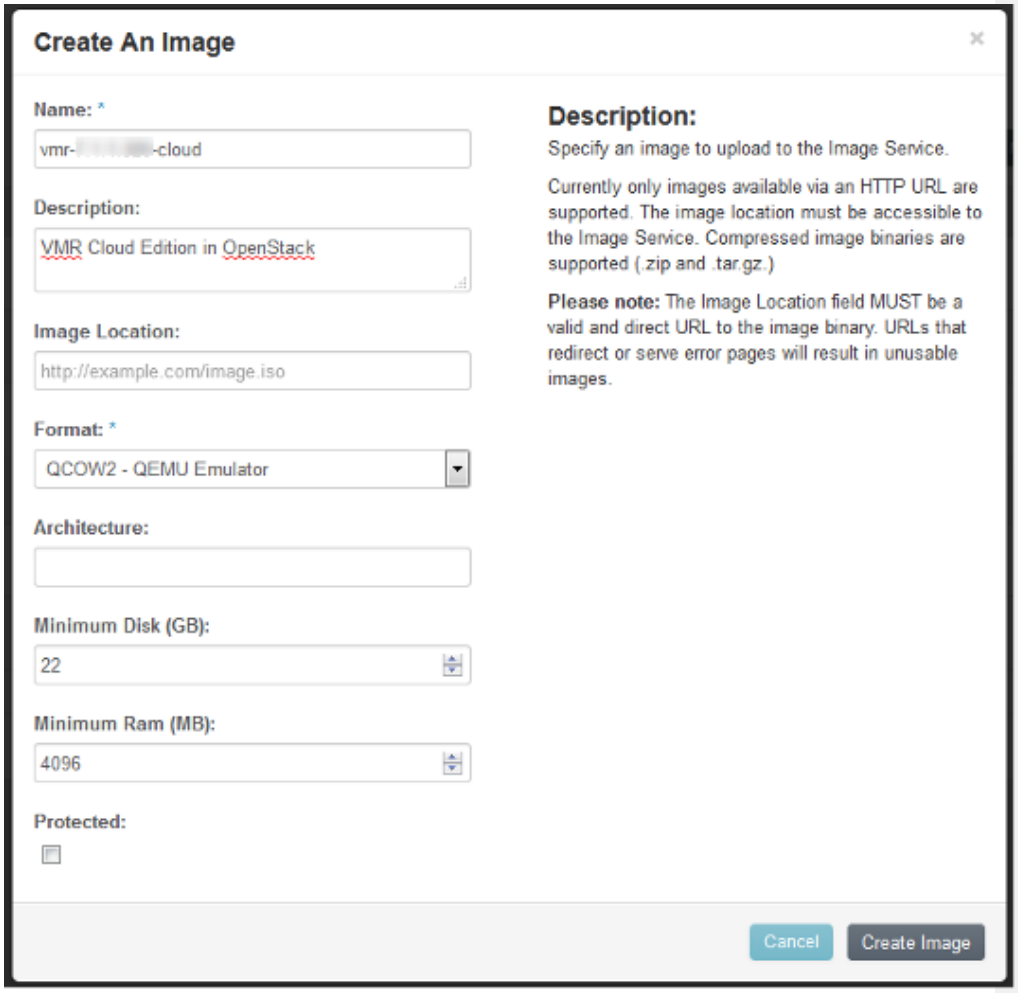
Then you are left with a file called soltr-<version>-vmr-cloud-<type>.qcow2. This is the file you will use going forward with OpenStack.

Note: Some OpenStack implementations claim to support direct qcow2.tar.gz file imports. However, Solace strongly recommends importing a decompressed image.

Import the QCOW2 into OpenStack

Note: Your OpenStack GUI may be slightly different than shown below. The following procedure may need to be adjusted to work in your OpenStack environment.

In the OpenStack GUI under **Images**, click **Create Image**.

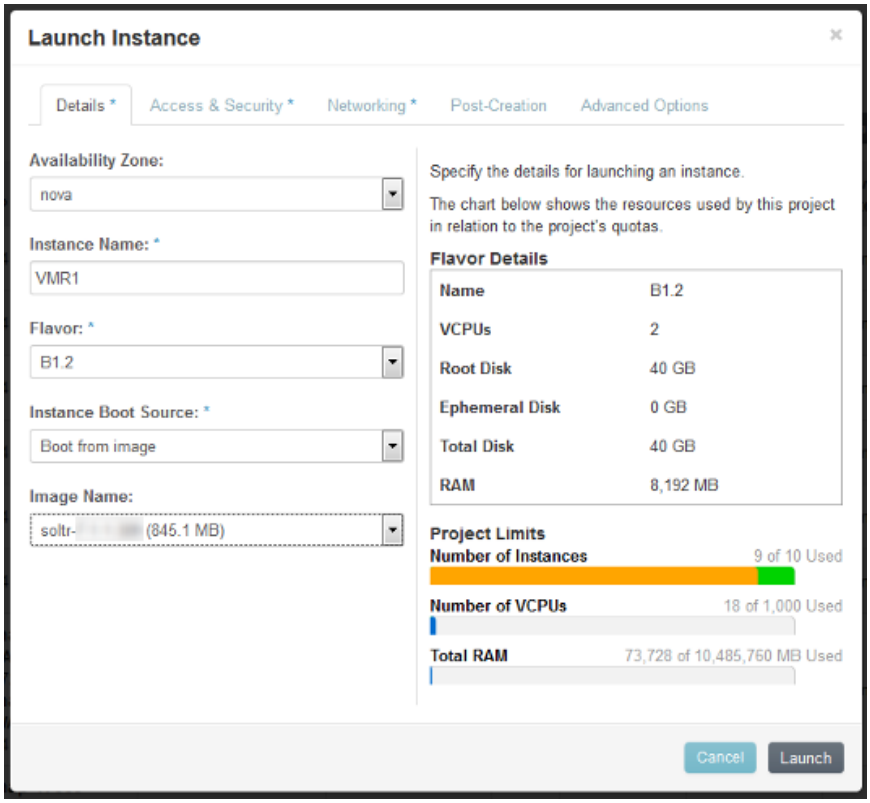


2. Enter details for your VMR Image, then click **Create Image**. The creation of the image may take several minutes to complete.

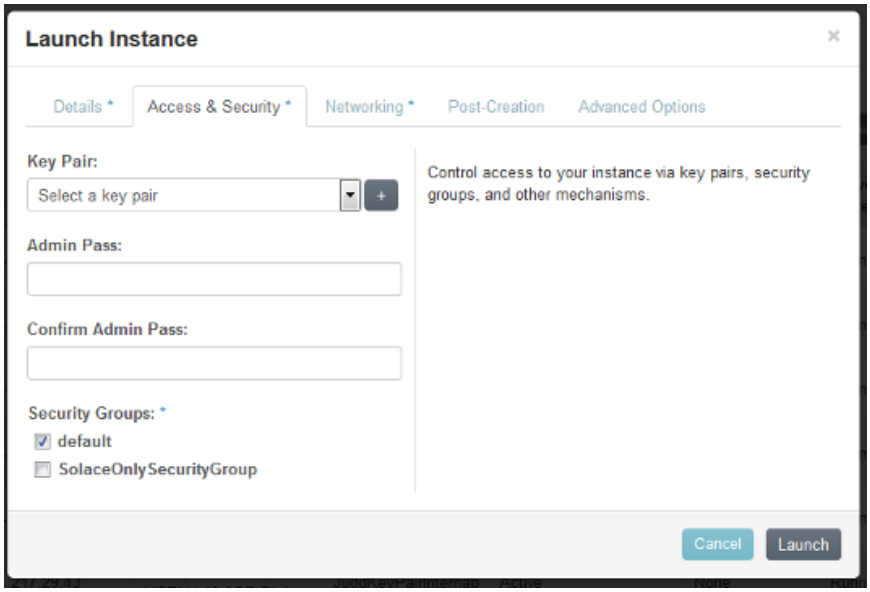
Starting the VMR Instance in OpenStack

1. Once the image is available for use, go to **Images**, and click **Launch Instance**.
2. On the **Details** tab, specify an Instance Name and Flavor; under **Instance Boot Source**, select Boot from **Image**; and under **Image Name**, select the VMR image that was created in the previous section.

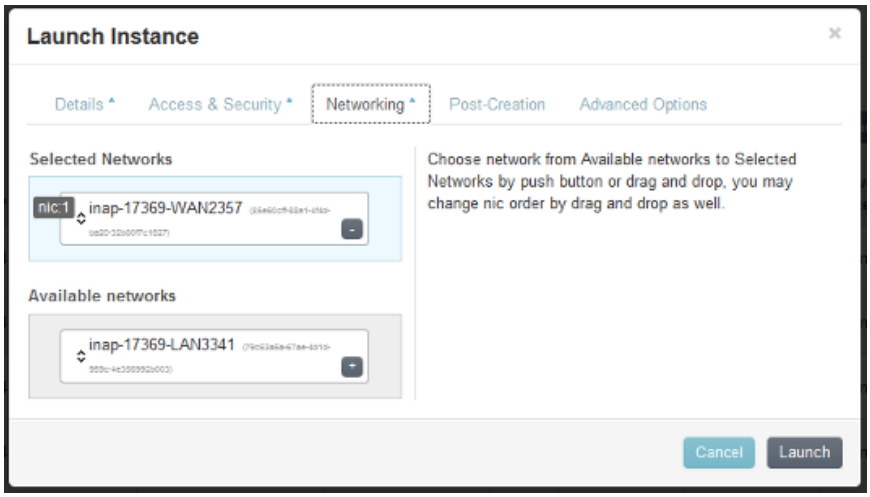
   Note: The VMR will inherit the Instance Name as its hostname.



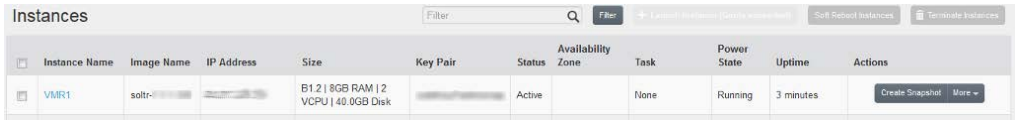
3. On the **Access & Security** tab, select or create a Key Pair, and assign a security group.



4. On the **Networking** tab, assign one and only one network interface to the VMR.



5. Click the **Launch** button. The VMR will launch and the OpenStack dashboard will display the running instance. Here you can find the IP address of the instance.



6. To log into SolOS CLI, enter the following command:

    ssh -i <auth_key> admin@<~~external_~~ip_address>

    Notes:

    - <auth_key> is the key pair you registered when launching the image
    - <ip_address> will need to be a floating ip if you are connecting from a shell hosted outside of openstack

Managing the VMR Configuration

Unlike a Solace messaging appliance, a VMR starts with a basic configuration that enables most common services. This basic configuration can be modified as required. For more information, refer to VMR Configuration Defaults.

Configuring Users

After initially booting a cloud instance of a VMR, only key-based login for the admin account is allowed. For more information on login and user configuration, refer to Configuring Users in the Cloud.

Configuring Security Groups

Open up the default “Security Groups” or create a new “Solace” Security Group to allow TCP connections on SSH (22), HTTP/WebSockets (80), MQTT (1883), SEMP (8080), REST (9000), and JMS (55555) as shown in the screen image below.

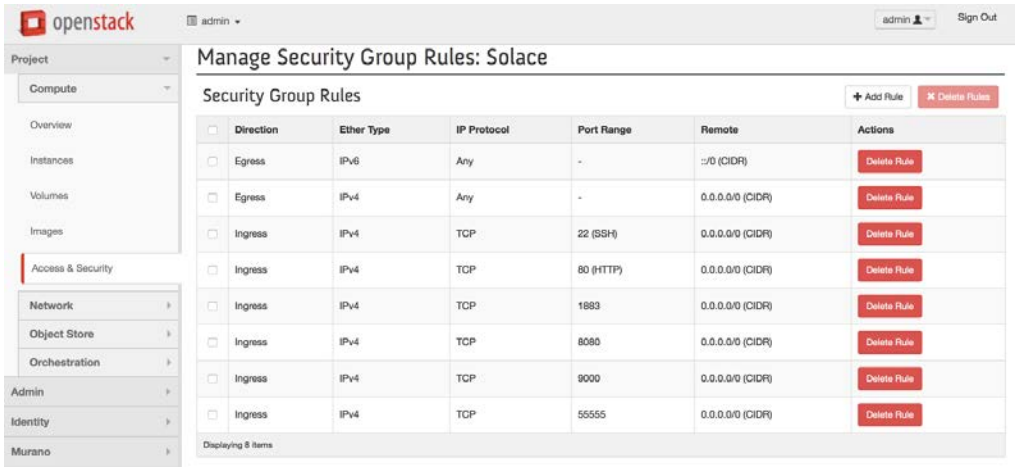
Comment [2]: From where?

Comment [SR3]: *Reply to Unknown Author (02/25/2016, 16:13): "..."*
I mean: Please provide more info. Please specify, that if user want to connect to an instance not from openstack nodes, a floating ip should be used

Formatted: Indent: Left: 0"

Formatted: List Paragraph, Outline numbered + Level: 1 + Numbering Style: Bullet + Aligned at: 0.75" + Indent at: 0.5"

Formatted: Indent: Left: 0"



5.4 Testing

5.4.1 Target use case(s)

OpenStack provides the environment in which to easily deploy and horizontally scale instances of the Solace VMR for carrier-class pub/sub and message queueing. Whether it be scaling of IOT device connections or guaranteed JMS message delivery, Mirantis Openstack plus Solace VMR can provide the solution within an easy to operate environment that takes full advantage of any NFV specific performance enhancement (if required) such as CPU pinning, Hypervisor bypass, service chaining, etc.

5.4.2 Test cases

Test cases include:

- Being able to SSH to the administrative CLI
- Being able to start SolAdmin GUI and connect to the SEMP management port
- Publishing using REST
- Publishing and subscribing using MQTT
- Publishing and subscribing using SMF (or JMS)

5.4.2 Test Tools

Recommended testing tools include:

- ssh
- curl

- SolAdmin (download from Solace, see section 5.3 "Solace VMR installation steps")
- SDKperf (download from Solace, see section 5.3 "Solace VMR installation steps")
- ~~(download from Solace)~~
- Mosquitto (or any other open source MQTT client from https://eclipse.org/paho/)

5.4.3 Test Results

5.4.3.1 SSH Port (22)

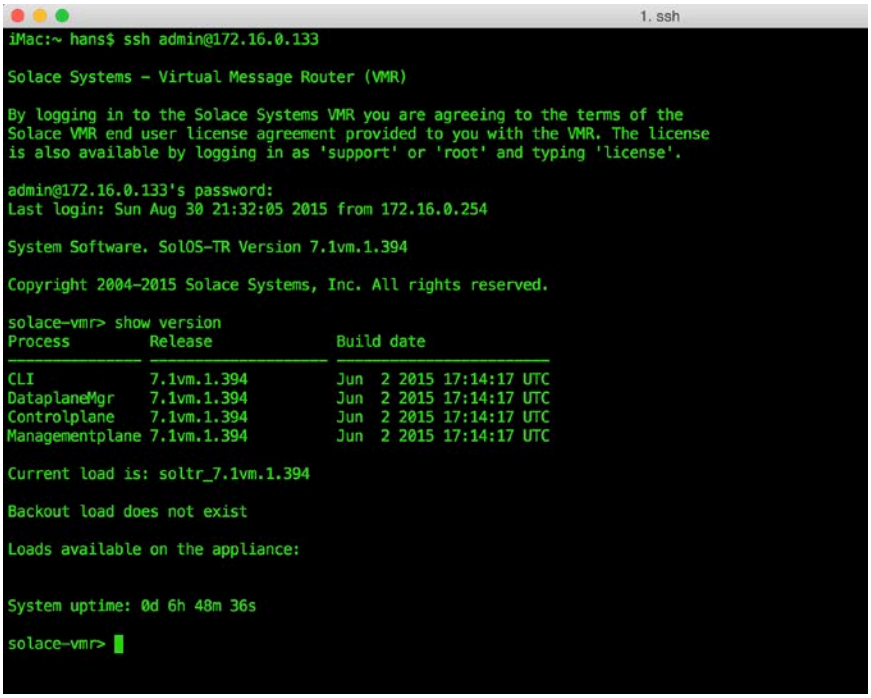
Test that you can SSH into the VMR administrative CLI from an external host and execute a "show version" command.

```
$ ssh -i <auth_key> admin@<floating~~external~~_ip>
```

Comment [SR4]: Do you mean floating IP (in the openstack terminology)? If yes, please clarify that.

Try a few show commands like "show version" or "show service".
Complete CLI help is online with the "help" command.
The full tree of commands can be listed with "tree"
Privileged mode (enables configuration commands) can be enabled with the "enable" command and is indicated with the "#" prompt in place of the ">" prompt.

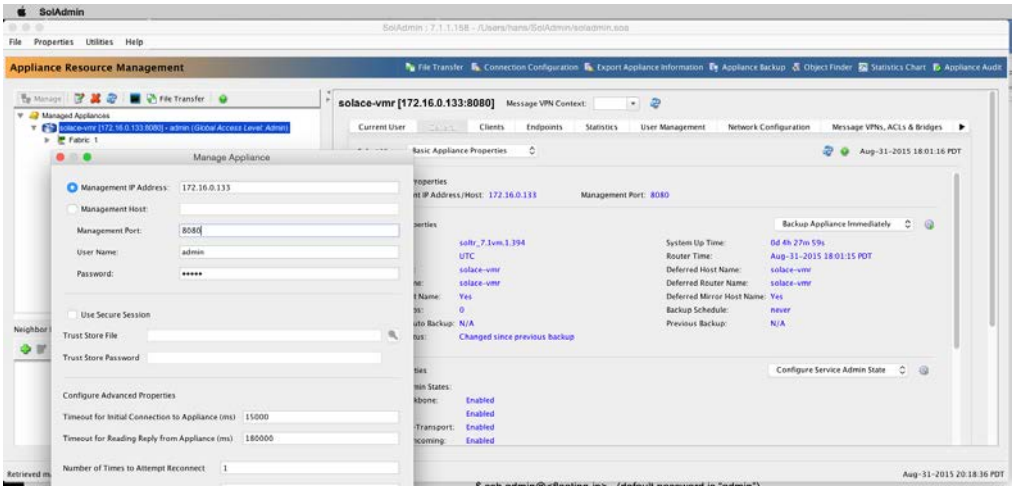
See the Solace Feature Provisioning Guide for specific example command to enable the key features of the Solace VMR.



5.4.3.2 SEMP Port (8080)

Test that you can connect to the SEMP admin port on 8080 by installing and connecting using the SolAdmin GUI (download from Solace, see section 5.3 "Solace VMR installation steps")

Comment [SR5]: Where can we get this software?



5.4.3.3 REST Port (9000)

Test that you can connect to the Solace REST API port using cURL. In the following command, replace 172.16.0.137 with the floating ip address of the VMR instance you wish to test.

Formatted: Font: Arial

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

```
$ curl -v -X POST -d "hello REST"
http://172.16.0.137:~~172.16.0.137:~~9000/TOPIC/mytopic
*   Trying 172.16.0.137...
* Connected to 172.16.0.137 (172.16.0.137) port 9000 (#0)
> POST /TOPIC/mytopic HTTP/1.1
> Host: 172.16.0.137:9000
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Length: 10
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 10 out of 10 bytes
< HTTP/1.1 200 OK
< Cache-Control: no-cache
< Content-Length: 0
< Server: Solace_Simulation/7.1vm.1.394
< Set-Cookie: TSID=000000002c100012; Path=/
```

Comment [SR6]: Please notify that this is the instance's floating ip in the current case and it may be differ.

```
< Solace-Client-Name: #rest/client/172.16.0.254/52229/000000002c100012
<
* Connection #0 to host 172.16.0.137 left intact
$
```

5.4.3.4 MQTT Port (1883)

Test that you can connect to the Solace MQTT API port using the command line tools from http://mosquitto.org/download/ or any other MQTT client application. In the following command, replace 172.16.0.137 with the floating ip address of the VMR instance you wish to test.

```
$ mosquitto_sub -h 172.16.0.137 -t mytopic

Then in a separate window start an MQTT publisher

$ mosquitto_pub -h 172.16.0.137 -t mytopic -l
hello MQTT
hello again MQTT
^C
```

5.4.3.5 SMF/JMS Port (55555)

Test that you can connect to the Solace Messaging Format (SMF) port using either the native Solace API or an open API such as JMS. Solace provides sample applications with each language supported as well as separate JMS examples for use in stand alone or POJO JMS apps or inside J2EE App Servers such as JBoss, WebLogic, and WebSphere using JCA and Message Driven Beans.

Example sdkperf command to start an SMF subscriber (download from Solace, see section 5.3 "Solace VMR installation steps"). In the following commands, replace 172.16.0.137 with the floating ip address of the VMR instance you wish to test.

```
$ sdkperf_java.sh -cip 172.16.0.137 –stl mytopic
```

Comment [SR7]: Where can user get this utility?

Example command line to publish 1000 messages of 32 bytes in size, at a rate of 100 messages per second, with the “direct” non-persistent message transport.

```
$ sdkperf_java.sh -cip 172.16.0.137 -ptl mytopic -mt direct -mn 1000 -mr 100
-msa 32
```

5.4.3 Known Issues and Workarounds

The table below lists the known issues in Solace VMR Enterprise Edition version 7.1.1.327 and provides workarounds for correcting them, where available.

Reference Number	Description
53435	CSPFNeighbor stats will always show zero.
53726	The VMR cannot participate in a network of Solace appliances, if that network contains more than 500,000 total subscriptions
54308	When logging in as root or support user, the following message will ap on the terminal: "Error sending status request (Operation not permitte
55637	Configuration changes made between executing backup 'for- upgrade' a non- upgrade VMR restart will be lost. Workaround: Do not make any configuration changes until you have completed the upgrade. If configuration changes have been made in th window, re- apply them to the VMR after the restart has completed.
55863	The power- down command will not stop the VM instance when runnin the cloud. The VMR will restart automatically after issuing the power- d command Workaround: Use cloud management tools to stop the VMR instance instead.
55915	The 'backup for- upgrade' command may fail due to lack of disk space when using the default message spool partition. Workaround: Migrate the message spool to a larger external block dev following the procedure in the VMR Setup Guide, and then retry the

Comment [SR8]: Please add a troubleshooting section.

Formatted: Outline numbered + Level: 3 + Numbering Style: 1, 2, 3, … + Start at: 2 + Alignment: Left + Aligned at: 0" + Indent at: 0.5"

Formatted: List Paragraph, Indent: Left: 0.39"

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Line spacing: At least 17 pt

Formatted: Font: (Default) Arial, 11 pt, Not Bold, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Not Bold, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted Table

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Not Bold, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Not Bold, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Font: (Default) Arial, 11 pt, Not Bold, Font color: Black

'backup for- upgrade' command.

Formatted: Font: (Default) Arial, 11 pt, Font color: Black

Formatted: Space After: 12 pt, Line spacing: At least 17 pt, No widow/orphan control, Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers