Preinstallation Tasks

Overview

Cisco Policy Suite offers a carrier-grade, high capacity, high performance, virtualized software solution, capable of running on VMware, OpenStack/KVM hypervisors or cloud infrastructures. To meet the stringent performance, capacity, and availability demands, the Cisco software requires that all allocated hardware system resources be 100% available when needed, and not oversubscribed or shared across separate VM's.

The following steps outline the basic process for a new installation of CPS:

Chapter 1:

1. Review virtual machine requirements
2. Orchestration Requirements
3. Install OpenStack
4. CPU Pinning
5. Configure OpenStack Users and Networks
6. Define Availability Zones
7. Download the required CPS images
8. Import images to Glance

9. Create Cinder Volumes
10. Verify or updated Default Quotas
11. Create Flavors
12. Set up Access and Security

Chapter 2:

1. Create CPS VMs using Nova Boot Commands
2. Create CPS VMs using Heat
3. Deploy CPS
4. Validate CPS Deployment
5. SR-IOV Support
6. Enable Custom Puppet to Configure Deployment
7. HTTPS Support for Orchestration API

Chapter 3:

1. Installation APIs
2. Upgrade APIs
    1. Unmount ISO
    2. Mount ISO
    3. Upgrade CPS
    4. Upgrade Status
3. System Configuration APIs

Virtual Machine Requirements

For customers operating a cloud infrastructure, the infrastructure must be configured to guarantee CPU, memory, network, and I/O availability for each CPS VM. Oversubscription of system resources will reduce the performance and capacity of the platform, and may compromise availability and response times. CPU core requirements are listed as pCPUs (physical cores) not vCPU's (hyper-threaded virtual cores).

In addition, the CPS carrier-grade platform requires:

- RAM reservation is enabled for all memory allocated to the CPS VM.
- CPU Hyperthreading must be ENABLED. To prevent over-subscription of CPU cores, CPU pinning should be ENABLED.
- CPU benchmark of at least 13,000 rating per chip and 1,365 rating per thread.
- The total number of VM CPU cores allocated should be 2 less than the total number of CPU cores per blade.

- Monitor the CPU STEAL statistic. This statistic should not cross 2% for more than 1 minute.

Note A high CPU STEAL value indicates the application is waiting for CPU, and is usually the result of CPU over allocation or no CPU pinning. CPS performance cannot be guaranteed in an environment with high CPU STEAL.

- CPU must be a high performance Intel x86 64-bit chipset.

Note BIOS settings should be set to high-performance values, rather than energy saving, hibernating, or speed stepping (contact hardware vendor for specific values).

- For deployments which cannot scale by adding more VM's, Cisco will support the allocation of additional CPU's above the recommendation to a single VM, but does not guarantee a linear performance increase.
- Cisco will not support performance SLA's for CPS implementations with less than the recommended CPU allocation.
- Cisco will not support performance SLA's for CPS implementations with CPU over-allocation (assigning more vCPU than are available on the blade, or sharing CPU's).
- Scaling and higher performance can be achieved by adding more VM's, not by adding more system resources to VM's.
- RAM latency should be lower than 15 nanosecond.
- RAM should be error-correcting ECC memory.
- Disk storage performance should be less than 2 millisecond average latency.
- Disk storage performance needs to support greater than 5000 input/output operations per second (IOPS) per CPS VM.
- Disk storage must provide redundancy and speed, such as RAID 0+1.
- Hardware and hardware design must be configured for better than 99.999% availability.
- For HA deployments, Cisco requires the customer designs comply with the Cisco CPS HA design guidelines.
  - At least two of each CPS VM type must be deployed: Policy Server (qns), Policy Director (lb), OAM (pcrfclient), Session Manager (sessionmgr).
  - Each CPS VM type must not share common HW zone with the same CPS VM type.
- The number of CPU cores, memory, NICs, and storage allocated per CPS VM must meet or exceed the requirements.

The following table provides information related to vCPU requirements based on:

- Hyper-threading: Enabled (Default)
- CPU Pinning: Enabled

- CPU Reservation: Yes (if allowed by hypervisor)
- Memory Reservation: Yes (if allowed)
- Hard Disk (in GB): 100

Table 1: HA Virtual Machine Requirements - Chassis Architecture

Physical Cores / Blade	VM Type	Memory (in GB)	Hard Disk (in GB)	vCPU	Configuration
Blade with 16 CPUs	Policy Server VMs (QNS)	16	100	12	Threading = 200 Mongo per host = 10 Criss-cross Mongo for Session Cache = 2 on each VM
Blade with 16 CPUs	Session Manager VMs	128	100	6	
Blade with 16 CPUs	Control Center (OAM) VMs	16	100	6	
Blade with 16 CPUs	Policy Director VMs (LB)	32	100	12	
Blade with 24 CPUs	Policy Server VMs (QNS)	16	100	10	Threading = 100 Mongo per host = 10 Criss-cross Mongo for Session Cache = 2 on each VM Hyper-threading = Default (Enable)
Blade with 24 CPUs	Session Manager VMs	80	100	8	
Blade with 24 CPUs	Control Center (OAM) VMs	16	100	8	
Blade with 24 CPUs	Policy Director VMs (LB)	32	100	12	

Table 2: HA Virtual Machine Requirements - Cloud Architecture

Physical Cores / Blade	VM Type	Memory (in GB)	Hard Disk (in GB)	vCPU	Configuration
Blade with 16 CPUs	Policy Server VMs (QNS)	16	100	12+	Threading = 200 Mongo per host = 10 Criss-cross Mongo for Session Cache = 2 on each VM
Blade with 16 CPUs	Session Manager VMs	128	100	6+	
Blade with 16 CPUs	Control Center (OAM) VMs	16	100	6+	
Blade with 16 CPUs	Policy Director VMs (LB)	32	100	8+	

Physical Cores / Blade	VM Type	Memory (in GB)	Hard Disk (in GB)	vCPU	Configuration
Blade with 24 CPUs	Policy Server VMs (QNS)	16	100	10+	Threading = 100 Mongo per host = 10 Criss-cross Mongo for Session Cache = 2 on each VM
Blade with 24 CPUs	Session Manager VMs	80	100	8+	
Blade with 24 CPUs	Control Center (OAM) VMs	16	100	8+	
Blade with 24 CPUs	Policy Director VMs (LB)	32	100	12+	

Note For large scale deployments having Policy Server (qns) VMs more than 35, Session Manager (sessionmgr) VMs more than 20, Policy Director (lb) VMs more than 2, recommended RAM for OAM (pcrfclient) VMs is 64GB.

Orchestration Requirements

The following orchestration capabilities are required to administer a CPS deployment in OpenStack:

- Ability to independently create/delete/re-create the Cluster Manager VM.
- Ability to snapshot Cluster Manager VM and restore the Cluster Manager VM from snapshot.
- Ability to attach and detach the ISO cinder volume to/from the Cluster Manager VM.
- CPS recommends that the CPS software ISO be mapped to a cinder volume. In deployments where this recommendation is used, prior to installation or upgrade or migration, the ISO cinder volume must be attached to the Cluster Manager so that the ISO can be mounted inside the Cluster Manager. The sample HEAT template provided in this document demonstrates how to automate mounting the ISO inside Cluster Manager. In deployments where this recommendation is not used, the CPS software ISO must be made available inside Cluster Manager VM and mounted using the method implemented by the customer.
- The Config drive must be used to pass in files such as userdata and the Config drive must be mounted to CPS VM in the 'iso9660' format.
- Any cinder volume required by the product code must be attached first to the VM and any customer environment specific cinder volumes should be attached after. One exception is the ISO cinder volume attached to Cluster Manager VM. In cases where ISO cinder volume is attached in a different order, the API to mount the ISO needs to be supplied with the right device name in the API payload.
- eth0 needs to be on the 'internal' network for inter-VM communication.
- On all CPS VMs, the Cluster Manager IP needs to be injected in /etc/hosts to ensure connectivity between each host and the Cluster Manager.
- CPS VM's role needs to be injected in `/etc/broadhop/.profile`, for example:

  ```
  NODE_TYPE=pcrfclient01
  ```

- For upgrades/migration and rollbacks, the orchestrator must have the ability to independently create/delete/re-create half/all of the following CPS VMs:

- Policy Server (qns)
- Policy Director (lb and iomanager)
- OAM (pcrfclient)
- Session Manager (sessionmgr)

  During a rollback, half of SM VMs must be deleted during Rollback procedure. As a result, the replica sets must be configured such that not all members of the replica set are in one half or the other. Replica set members must be distributed across the two upgrade sets so that the replica set is not entirely deleted. Refer to the *CPS Migration and Upgrade Guide* for more details.

- For scaling, the orchestrator must have the ability to independently create/delete/re-create half/all of the following CPS VMs in each scaling unit:
  - Policy Server (qns)
  - Session Manager (sessionmgr)

Install OpenStack

CPS is supported on OpenStack Liberty or Newton or Queens.

CPS can also be installed on Cisco distributed platforms: Ultra B1.0 or Mercury 2.2.8

For more information about installing OpenStack and Cisco distributed platforms, refer to:

- OpenStack Liberty: http://docs.openstack.org/liberty/
- OpenStack Newton: https://docs.openstack.org/newton/
- OpenStack Queens: https://docs.openstack.org/queens/
- Ultra B1.0: https://www.cisco.com/c/en/us/solutions/service-provider/virtualized-packet-core/index.html
- Mercury 2.2.8:

  https://www.cisco.com/c/en/us/support/cloud-systems-management/nfv-infrastructure/tsd-products-support-series-home.html

After you install OpenStack, you must perform some prerequisite tasks. The following sections describe these prerequisite tasks.

Note The example commands in the following sections are related to OpenStack Liberty. For commands related to other supported platforms, refer to the corresponding platform documentation.

CPU Pinning

CPU pinning is supported and recommended in OpenStack deployments where hyperthreading is enabled. This enables CPS VMs to be pinned to dedicated physical CPU cores.

Prerequisites

- OpenStack Liberty (OSP 7.2) or OpenStack Newton or OpenStack Queens or Ultra B1.0 or Mercury 2.2.8
- Numactl must be installed on control and compute nodes.

Refer to the following link for general instructions to enable CPU pinning for guest VMs:
http://redhatstackblog.redhat.com/2015/05/05/cpu-pinning-and-numa-topology-awareness-in-openstack-compute/

Install numactl

The numactl package provides a command to examine the NUMA layout of the blades. Install this package on compute nodes to help determine which CPUs to set aside for pinning.

Run the following command to install numactl:

```
yum install numactl
```

Identify the Physical CPUs to Use for Pinning

Step 1 Run the following command on the compute nodes where you want to set aside physical CPUs for pinning.

```
numactl --hardware
```

```
[root@os6-compute-1 ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 8 9 10 11
node 0 size: 98245 MB
node 0 free: 87252 MB
node 1 cpus: 4 5 6 7 12 13 14 15
node 1 size: 98304 MB
node 1 free: 95850 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
```

Step 2 Determine the pool of CPUs you want to set aside for pinning.

At least 2 CPUs should be set aside for the Hypervisor in each node if it is a compute only blade. If the blade is operating as both a control and compute node, set aside more CPUs for OpenStack services.

Select the remaining CPUs for pinning.

In the above example, the following CPUs could be selected for pinning: **2, 3, 6, 7, 8-11, 12-15**.

Prevent Hypervisor from Using CPUs Set Aside for Pinning

To configure the hypervisor so that it will not use the CPUs identified for CPU Pinning:

Step 1 Open the KVM console for the Compute node.

Step 2 Execute the following command to update the kernal boot parameters:

```
grubby --update-kernel=ALL --args="isolcpus=2,3,6,7,8,9,10,11,12,13,14,15"
```

Step 3 Edit the `/etc/nova/nova.conf` file on that blade and set the `vcpu_pin_set` value to a list or range of physical CPU cores to reserve for virtual machine processes. For example:

```
vcpu_pin_set=2,3,6,7,8,9,10,11,12,13,14,15
```

Step 4 Reset the blade.

Step 5 After Linux has finished the boot process, enter the following command to verify the above Kernel boot options:

```
cat /proc/cmdline
```

The iso1cpus options you defined will be displayed, for example:

```
BOOT_IMAGE=/vmlinuz-3.10.0-327.el7.x86_64 root=/dev/mapper/cinder--volumes-slash ro rd.lvm.lv=rhel/swap
 crashkernel=auto rd.lvm.lv=cinder-volumes/slash rhgb quiet LANG=en_US.UTF-8
isolcpus=2,3,6,7,8,9,10,11,12,13,14,15
```

Configure Host Aggregates and Update Flavors

Step 1 Follow the instructions in the post (refer to Prerequisites, on page 7) to create host-aggregate, add compute hosts to the aggregate and set the CPU pinning metadata.

Step 2 Update Flavors which are NOT used for CPU pinning (non-CPS VM flavors) with the following command:

```
nova flavor-key <id> set "aggregate_instance_extra_specs:pinned"="false"
```

Step 3 Update Flavors which are CPS VM flavors (or a sub-set, if you are planning to bring up only certain VMs in CPS with CPU pinning) with the following command:

```
nova flavor-key <id> set hw:cpu_policy=dedicated
nova flavor-key <id> set aggregate_instance_extra_specs:pinned=true
```

Step 4 Launch a CPS VM with the performance enhanced Flavor. Note the host on which the instance is created and the instance name.

`nova show <id>` will show the host on which the VM is created in the field: `OS-EXT-SRV-ATTR:host`

`nova show <id>` will show the virsh Instance name in the field: `OS-EXT-SRV-ATTR:instance_name`

Step 5 To verify that vCPUs were pinned to the reserved physical CPUs, log in to the Compute node on which the VM is created and run the following command:

```
virsh dumpxml <instance_name>
```

The following section will show the physical CPUs in the field `cpuset` from the list of CPUs that were set aside earlier. For example:

```
<vcpu placement='static'>4</vcpu>
  <cputune>
```

```
  <shares>4096</shares>
  <vcpupin vcpu='0' cpuset='11'/>
  <vcpupin vcpu='1' cpuset='3'/>
  <vcpupin vcpu='2' cpuset='2'/>
  <vcpupin vcpu='3' cpuset='10'/>
  <emulatorpin cpuset='2-3,10-11'/>
</cputune>
```

Configure OpenStack Users and Networks

For more information about keystone commands, refer to the keystone command reference at: http://docs.openstack.org/cli-reference/index.html

Step 1 Create an OpenStack tenant with the name **core** (under which you can install the VMs) as shown in the following command:

```
source /root/keystonerc_admin

openstack project create --description "PCRF Admin" core
```

Step 2 For the above tenant, create an OpenStack user with the name **core** as shown in the following command:

```
source /root/keystonerc_admin

openstack user create --password "Core123" --email "core@cisco.com" --project core core
```

Step 3 The tenant must have access to the following three provider VLANs. In this guide, the names of the VLANs are:

- Internal - Used for inter-VM communication of CPS VMs.
- Gx - Used by CPS to access the PCRF
- Management - Used to access the management functions of the CPS

Note Hosts in the CPS cluster can be configured to have IPv4 or IPv6 addresses. Currently, IPv6 is supported only for external interfaces. All alphabet characters used in virtual IPv6 addresses configured in csv files must be in small case letters.

You can specify any names.

Set up the VLANs as shown in the following table:

Table 3: VLANs

Name	Subnet	VLAN-ID	Allocation Pool	Purpose
Internal	Specific to environment, for example:172.xx.xx.0/24	Specific to environment, for example: 20xx	Specific to environment, for example:172.xx.xx.16 to 172.xx.xx.220	The neutron network used by Cisco Policy Suite VMs for internal / private communication.

Name	Subnet	VLAN-ID	Allocation Pool	Purpose
Management	Specific to environment, for example: 10.xx.xx.0/24	Specific to environment, for example: 20xx	Specific to environment, for example: 10.xx.xx.100 to 10.xx.xx.120	The neutron network used by Cisco Policy Suite VMs to connect on Management network.
Gx	Specific to environment, for example:192.xx.xx.0/24	Specific to environment, for example: 20xx	Specific to environment, for example: 192.xx.xx.16 to 192.xx.xx.220	The neutron network used by Cisco Policy Suite VMs to connect on Gx network.

The following example illustrates how to create networks and subnets:

```
source /root/keystonerc_admin
# $1 network name
# $2 vlan_id
# $3 gw ip
# $4 pool start
# $5 pool end
# $6 subnet x.x.x.x./y
# $OSTACKTENANT  core
#### Networks
echo "openstack network create --share --project $OSTACKTENANT --provider-network-type "vlan"
--provider-physical-network "physnet1" --provider-segment $2 "$1""
openstack network create --share --project $OSTACKTENANT --provider-network-type "vlan"
--provider-physical-network "physnet1" --provider-segment $2  "$1"
echo "openstack subnet create --no-dhcp --project $OSTACKTENANT --gateway $3 --subnet-range $6
--network $1 --allocation-pool start=$4,end=$5 $2""
openstack subnet create --no-dhcp --project $OSTACKTENANT --gateway $3 --subnet-range $6 --network
$1 --allocation-pool start=$4,end=$5 $2"
```

All of these networks must be either shared or accessible by Cisco Policy Suite OpenStack network.

Define Availability Zones

Step 1 A core user must have administrator role to launch VMs on specific hosts in OpenStack. Add the administrator role to the core user in the core tenant as shown in the following command:

```
openstack role add --project core --user core admin
```

Note The administrator role for the core user is not required if you do not intend to launch the VM on a specific host and if you prefer to allow nova to select the host for the VM.

Step 2 You must define at least one availability zone in OpenStack. Nova hypervisors list will show list of available hypervisors. For example:

```
[root@os24-control]# nova hypervisor-list
+----+-------------------------+
| ID | Hypervisor hostname     |
+----+-------------------------+
| 1  | os24-compute-2.cisco.com |
```

```
| 2  | os24-compute-1.cisco.com |
+----+--------------------------+
```

Step 3 Create availability zones specific to your deployment. The following commands provide an example of how to create the availability zones:

```
nova aggregate-create osXX-compute-1 az-1

nova aggregate-add-host osXX-compute-1 osXX-compute-1.cisco.com

nova aggregate-create osXX-compute-2 az-2

nova aggregate-add-host osXX-compute-2 osXX-compute-2.cisco.com
```

Note The above command creates two availability zones az-1 and az-2. You need to specify the zones az-1 or az-2 using Nova boot commands (see Create CPS VMs using Nova Boot Commands), or in the Heat environment files (see Create CPS VMs using Heat). You can also put more than one compute node in an availability zone. You could create az-1 with both blades, or in a 6-blade system, put three blades in each and then use `az-1:osXX-compute-2.cisco.com` to lock that VM onto that blade.

Availability zone for svn01 volume should be the same as that of pcrfclient01, svn02 volume as that of pcrfclient02, similarly for mongo01 and sessionmgr01, mongo02 and sessionmgr02. The same concept is applicable to cluman – the ISO volume and Cluster Manager (cluman) should be in same zone.

Step 4 Configure the compute nodes to create volumes on availability zones: Edit the `/etc/cinder/cinder.conf` file to add the `storage_availability_zone` parameter below the `[DEFAULT]` line. For example:

```
ssh root@os24-compute-1.cisco.com

 [DEFAULT]
storage_availability_zone=az-1:os24-compute-1.cisco.com
```

After adding the storage availability zone lines in `cinder.conf` file, restart the cinder volume with following command:

```
systemctl restart openstack-cinder-volume
```

Repeat Step 4, on page 11 for other compute nodes.

Download the ISO Image

Download the CPS ISO image file (CPS_x.x.x.release.iso) for the release from software.cisco.com and load it on the OpenStack control node.

Download the Base Image

CPS supports the QCOW2 image format for OpenStack installations. The QCOW2 base image is available to download as a separate file, and is not packaged inside the ISO.

Download the CPS QCOW2 base image file and extract it as shown in the following command:

```
tar -zxvf CPS_x.x.x_Base.qcow2.release.tar.gz
```

Locate the base image that is the root disk used by Cisco Policy Suite VM.

Import Images to Glance

Note The commands mentioned in this section are specific to OpenStack Liberty. For other OpenStack release specific commands, refer to https://releases.openstack.org/.

Import the Cisco Policy Suite base QCOW2 or VMDK image into the OpenStack glance repository.

To import the QCOW2 image, enter the following:

```
source /root/keystonerc_admin
```

```
glance image-create --name "<base vm name>" --visibility "<visibility>" --disk-format "qcow2"
--container "bare" --file <path of base qcow2>
```

To import the VMDK image, enter the following:

```
source /root/keystonerc_admin
```

```
glance image-create --name " <base vm name> " --visibility "<visibility>" --disk-format
"vmdk" --container "bare" --file <path of base vmdk>
```

Import the ISO image by running the following command:

```
source /root/keystonerc_admin
```

```
glance image-create --name "CPS_x.x.x.release.iso" --visibility "public" --disk-format "iso"
--container "bare" --file <path to iso file>
```

For more information on glance commands, refer to http://docs.openstack.org/cli-reference/glance.html.

Create Cinder Volumes

Create a cinder volume to map the glance image to the volume. This ensures that the cinder volume (and also the ISO that you imported to glance) can be automatically attached to the Cluster Manager VM when it is launched.

In the core tenant, create and format the following cinder volumes to be attached to various VMs:

- svn01
- svn02
- mongo01
- mongo02
- CPS_*x.x.x*.release.iso

It is recommended you work with Cisco AS to determine the size of each volume.

Note For mongo01 and mongo02, the minimum recommended size is 60 GB.

The following commands illustrate how to create the cinder volumes:

```
source /root/keystonerc_user
cinder create --metadata fstype=ext4 fslabel=newfs dio=yes --display-name svn01
--availability-zone az-1:os24-compute-1.cisco.com 2
cinder create --metadata fstype=ext4 fslabel=newfs dio=yes --display-name svn02
--availability-zone az-2:os24-compute-2.cisco.com 2
cinder create --metadata fstype=ext4 fslabel=newfs dio=yes --display-name mongo01
--availability-zone az-1:os24-compute-1.cisco.com 60
cinder create --metadata fstype=ext4 fslabel=newfs dio=yes --display-name mongo02
--availability-zone az-2:os24-compute-2.cisco.com 60
cps_iso_id=$(glance image-list | grep $cps_iso_name | awk ' {print $2}')
cinder create --display-name $cps_iso_name --image-id $cps_iso_id --availability-zone
az-1:os24-compute-1.cisco.com 3
```

Note

- Replace *$cps_iso_name* with the ISO filename. For example: `CPS_9.0.0.release.iso`
- If any host in the availability zone may be used, then only the zone needs to be specified. Currently, the recommendation only specifies `zone:host`

Verify or Update Default Quotas

OpenStack must have enough Default Quotas (that is, size of RAM, number of vCPUs, number of instances) to spin up all the VMs.

Update the Default Quotas in the following page of the OpenStack dashboard: **Admin** > **Defaults** > **Update Defaults**.

For example:

- instances: 20
- ram: 1024000
- cores: 100

Create Flavors

OpenStack flavors define the virtual hardware templates defining sizes for RAM, disk, vCPUs, and so on.

To create the flavors for your CPS deployment, run the following commands, replacing the appropriate values for your CPS deployment.

```
source /root/keystonerc_admin
   nova flavor-create --ephemeral 0 pcrfclient01 auto 16384 0 2
   nova flavor-create --ephemeral 0 pcrfclient02 auto 16384 0 2
   nova flavor-create --ephemeral 0 cluman auto 8192 0 4
   nova flavor-create --ephemeral 0 qps auto 10240 0 4
   nova flavor-create --ephemeral 0 sm auto 16384 0 4
   nova flavor-create --ephemeral 0 lb01 auto 8192 0 6
   nova flavor-create --ephemeral 0 lb02 auto 8192 0 6
```

Set up Access and Security

Allow access of the following TCP and UDP ports from the OpenStack dashboard **Project** > **Access & Security** > **default / Manage Rules** or from the CLI as shown in the following example:

```
source /root/keystonerc_user
openstack security group rule create default --protocol icmp --remote-ip 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 22:22 --remote-ip
0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 53:53 --remote-ip
0.0.0.0/0
openstack security group rule create default --protocol udp --dst-port 53:53 --remote-ip
0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 80:80 --remote-ip
0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 443:443 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 7443:7443 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 8443:8443 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 9443:9443 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 5540:5540 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 1553:1553 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 3868:3868 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 9160:9160 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 27717:27720 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 5432:5432 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 61616:61616 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 9443:9450 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 8280:8290 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 7070:7070 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 8080:8080 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 8090:8090 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 7611:7611 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 7711:7711 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol udp --dst-port 694:694 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 10080:10080 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 11211:11211 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 111:111 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 27717:27720 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 2049:2049 --remote-ip
```

```
 0.0.0.0/0
openstack security group rule create default --protocol udp --dst-port 2049:2049 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 32767:32767 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol udp --dst-port 32767:32767 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 2049:2049 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 2049:2049 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 8161:8161 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 12712:12712 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 9200:9200 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 2049:2049 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol udp --dst-port 5060:5060 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol tcp --dst-port 8548:8548 --remote-ip
 0.0.0.0/0
openstack security group rule create default --protocol udp --dst-port 8548:8548 --remote-ip
 0.0.0.0/0
```

where: default is the name of the security group.