

Deploy the VM-Series Firewall with Service Chaining and Service Scaling

Service chaining is a Contrail feature that deploys a VM-Series firewall as a service instance in your OpenStack environment. A service chain is a set of service virtual machines, such as firewalls or load balancers, and each virtual machine in the service chain is a service instance. Service scaling allows you to dynamically deploy additional instances of the VM-Series firewall. Using CPU utilization or incoming bytes per second metrics gathered by Celiometer, OpenStack deploys or shuts down additional instances of the VM-Series firewall to meet the current needs of your network.


The VM-Series firewall in OpenStack solution leverages heat orchestration templates to configure and deploy the components required for service chaining and service scaling. The heat templates provided by Palo Alto networks create a service template, service instance, and service policy (to direct traffic to the VM-Series firewall) to deploy two Linux servers and the VM-Series firewall service instance between them.

- ▲ [Components of the VM-Series for OpenStack Solution](#)
- ▲ [Service Chaining and Service Scaling Environment File](#)
- ▲ [Install the VM-Series Firewall as a Service Chain](#)

Components of the VM-Series for OpenStack Solution

The following components are required for deploying the VM-Series firewall in an OpenStack environment.

Component	Description
Software	<ul style="list-style-type: none"> • Hypervisor: KVM/Ubuntu 14.04 • Networking: Contrail 3.0.2 • OpenStack Distro: Mirantis 8.0 (Liberty) • Orchestration: OpenStack Heat Templates (Version 2015-10-15 or higher) • VM-Series for KVM PAN-OS 8.0 or later
VM-Series Hardware Resources	<p>See VM-Series System Requirements for the minimum hardware requirements for your VM-Series model.</p> <p>In OpenStack, flavors define the CPU, memory, and storage capacity of a compute instance. When setting up your Heat template, choose the compute flavor that meets or exceeds the hardware requirements for the VM-Series model.</p>
Fuel Master	Fuel is a web UI-driven deployment and management tool for OpenStack.
OpenStack Controller	This node runs most of the shared OpenStack services, such as API and scheduling. Additionally, the Horizon UI runs on this node.

Component	Description
OpenStack Compute	<p>The compute node contains the virtual machines, including the VM-Series firewall, in the OpenStack deployment. The compute node that houses the VM-Series must meet the following criteria:</p> <ul style="list-style-type: none"> • Instance type OS::Nova::Server • Allow configuration of at least three interfaces • Accept the VM-Series qcow2 image • Accept the compute flavor parameter <p> Install the OpenStack compute node on a bare-metal server because the VM-Series firewall does not support nested virtualization.</p>
Contrail Controller	<p>The Contrail controller node is a software-defined networking controller used for management, control, and analytics for the virtualized network. It provides routing information to the compute and gateway nodes.</p> <p>Additionally, the Contrail controller provides the necessary support for service chaining.</p>
Contrail Gateway	<p>The Contrail gateway node provides IP connectivity to external networks from virtual networks. MPLS over GRE tunnels from the virtual machines terminate at the gateway node, where packets are decapsulated and sent to their destinations on IP networks.</p>
Celometer (OpenStack Telemetry)	<p>In the case of the VM-Series firewall for OpenStack, Celometer monitors CPU utilization for service scaling. When CPU utilization meets the defined thresholds, a new service instance of the VM-Series firewall is deployed or shut down.</p>
Heat Orchestration Template Files	<p>Palo Alto Networks provides a sample Heat template for deploying the VM-Series firewall. This template is made up of a main template and an environment template. These files instantiate one VM-Series instance with one management interface and two data interfaces. The management interface and one data interface attach to an untrust network. The other data interface connects to the trust network.</p> <p>Additionally, the template instantiates a Linux server with one interface. The interface of the server attaches to the private network created by the template.</p>
VM-Series Firewall Bootstrap Files	<p>The VM-Series firewall bootstrap files consist of a init-cfg.txt file, bootstrap.xml file, and VM-Series auth codes. Along with the Heat template files, Palo Alto Networks provides a sample init-cfg.txt and bootstrap.xml files. You must provide your own auth codes to license your VM-Series firewall and activate any subscriptions. See Bootstrap the VM-Series Firewall for more information about VM-Series bootstrap files.</p>

Palo Alto Networks provides a set of Heat Orchestration Templates that deploy a VM-Series firewall with three interfaces between two Linux virtual machines with one interface each. Additionally, these templates allow you to deploy the firewall in virtual wire mode or layer 3 mode. The various parameters of the environment template file are prepopulated and you must edit some fields to match your network environment.

Service Chaining and Service Scaling Environment File

The heat template environment file defines the parameters specific to the VM-Series firewall instance deployed through service chaining or service scaling. The parameters defined in the environment file are divided into sections described below. There are two versions of the heat templates for service chaining—vwire and L3— and one for service scaling.

Service chaining requires the heat template files and two bootstrap files to launch the VM-Series firewall service instance and two Linux servers in the left and right networks.

- **<file-name>.yaml**—This template defines the resources created to support the VM-Series firewall and two Linux servers, such as interfaces and IP addresses,
- **<file-name>_env.yaml**—This environment file defines the environment that the VM-Series firewall and Linux servers exist in. Many parameters in the template reference the parameters defined in this file, such as flavor for the VM-Series and the names of the Linux servers.
- **service_instance.yaml**—(Service Scaling only) This is a nested heat template that is reference by Service_Scaling_template.yaml to deploy the service instance. It provides the necessary information to deploy service instances for scaling events.
- **init-cfg.txt**—Provides the minimum information required to bootstrap a VM-Series firewall. The init-cfg.txt provided only includes the operational command to enable DHCP on the firewall management interface.
- **bootstrap.xml**—Provides basic configuration for the VM-Series firewall. The bootstrap.xml file configures the data interfaces and IP addresses. These values must match the corresponding values in the heat templates files.

For more information about the init-cfg.txt and bootstrap.xml files, see [Bootstrap Configuration Files in the VM-Series 8.0 Deployment Guide](#).

- ▲ [Virtual Network](#)
- ▲ [Virtual Machine](#)
- ▲ [Service Template](#)
- ▲ [Service Instance](#)
- ▲ [IPAM](#)
- ▲ [Service Policy](#)
- ▲ [Alarm](#)

Virtual Network

The virtual network configuration parameters in the heat template environment file define the virtual network that connects the VM-Series firewall and the two Linux servers deployed by the heat template.

Virtual Network (VN Config)	
management_network	The VM-Series firewall management interface attaches to the network specified in this parameter.
left_vn or left_network	Name of the left virtual network.

Virtual Network (VN Config)	
right_vn or right_network	Name of the right virtual network.
left_vn_fqdn	Fully qualified domain name of the left virtual network.
right_vn_fqdn	Fully qualified domain name of the right virtual network.
route_target	Edit this value so route target configuration matches that of your external gateway.

Virtual Machine

The virtual machine parameters define the left and right Linux servers. The name of the port tuple is defined here and referenced by the heat template. In Contrail, a port tuple is an ordered set of virtual network interfaces connected to the same virtual machine. With a port tuple, you can create ports and pass that information when creating a service instance. The heat template creates the left, right, and management ports and adds them to the port tuple. The port tuple is then linked to the service instance. When you launch the service instance using the heat templates, the port tuple maps the service virtual machine to the virtual machine deployed in OpenStack.

Virtual Machine (VM Config)	
flavor	The flavor of the left and right virtual machines. The default value is m1.small.
left_vm_image or right_vm_image or image	The name of the software image for the left and right virtual machines. Change this value to match the file name of the image you uploaded. The default is TestVM, which is a default image provided by OpenStack.
svm_name	The name applied to the VM-Series firewall.
left_vm_name and right_vm_name	The name of the left and right virtual machines.
port_tuple_name	The name of the port tuple used by the two Linux servers and the VM-Series firewall.
server_key	The server key is used for accessing virtual machines through ssh. The default value is server_key. You can change this value by enter a new server key in the environment file.

Service Template

The service template defines the parameters of the service instance, such as the software image, virtual machine flavor, service type, and interfaces. Service templates are configured within the scope of a domain and can be used on all projects within the specified domain.

Service Template (ST Config)	
S_Tmp_name	The name of the service template.

Service Template (ST Config)	
S_Tmp_version	The service template version. The default value is 2. Do not change this parameter because service template version 2 is required to support port tuples.
S_Tmp_service_mode	Service mode is the network mode used by the VM-Series firewall service instance. For the L3 network, the default value is in-network. For the virtual wire template, the default value is transparent.
S_Tmp_service_type	The type of service being deployed by the template. The default value is firewall and should not be changed when deploying the VM-Series firewall.
S_Tmp_image_name	This parameter specifies the VM-Series base image used by the Heat template when deploying the VM-Series firewall. Edit this parameter to match the name of the VM-Series firewall image uploaded to your OpenStack environment.
S_Tmp_flavor	This parameter defines the hardware resources allocated to the VM-Series firewall. The default value is m1.large.
S_Tmp_interface_type_mgmt S_Tmp_interface_type_left S_Tmp_interface_type_right	These parameters define the interface type for management, left, and right interfaces.
domain	The domain where this service template is tied to. The default value is default-domain.

Service Instance

The service instance portion of the heat template environment file provides the name of the individual instance deployed by the heat template and service template.

Service Instance (SI Config)	
S_Ins_name	The service instance name. This is the name of the VM-Series firewall instance in Contrail.
S_Ins_fq_name	The fully qualified name of the service instance.

IPAM

IP address management (IPAM) provides the IP address information for the interfaces of the service instance.

IPAM (IPAM Config)	
NetIPam_ip_prefix_mgmt	The IP prefix of the management interface on the VM-Series firewall. The default value is 172.20.0.
NetIPam_ip_prefix_len_mgmt	The IP prefix length of the management interface on the VM-Series firewall. The default value is /24.

IPAM (IPAM Config)	
NetIPam_ip_prefix_left	The IP prefix of the left interface on the VM-Series firewall. The default value is 10.10.1.0.
NetIPam_ip_prefix_len_left	The IP prefix length of the left interface on the VM-Series firewall. The default value is /24.
NetIPam_ip_prefix_right	The IP prefix of the right interface on the VM-Series firewall. The default value is 10.10.2.0.
NetIPam_ip_prefix_len_right	The IP prefix length of the right interface on the VM-Series firewall. The default value is /24.
NetIPam_addr_from_start_true	This parameter determines how IP addresses are assigned to VMs on the subnets described above. If true, any new VM takes the next available IP address. If false, any new VM is assigned an IP address at random. The default value is true.

Service Policy

The service policy defines the traffic redirection rules and policy that point traffic passing between the left and right virtual machines to the VM-Series firewall service instance.

Service Policy (Policy Config)	
policy_name	The name of the service policy in Contrail that redirects traffic through the VM-Series firewall. For the L3 template, the default value is PAN_SVM_policy-L3. For the virtual wire template, the default value is PAN_SVM_policy-vw.
policy_fq_name	The fully qualified name of the service policy.
simple_action	The default action Contrail applies to traffic going to the VM-Series firewall service instance. The default value is pass because the VM-Series firewall will apply its own security policy to the traffic.
protocol	The protocols allowed by Contrail to pass to the VM-Series firewall. The default value is any.
src_port_end and src_port_start	Use this parameter to specify source port(s) that should be associated with the policy rule. You can enter a single port, a list of ports separated with commas, or a range of ports in the form of <port>-<port>. The default value is -1 in the provided heat templates; meaning any source port.
direction	This parameter defines the direction of traffic that is allowed by Contrail to pass to the VM-Series firewall. The default value is <> or bidirectional traffic.

Service Policy (Policy Config)	
dst_port_end and dst_port_start	<p>Use this parameter to specify destination port(s) that should be associated with the policy rule. You can enter a single port, a list of ports separated with commas, or a range of ports in the form of <port>-<port>.</p> <p>The default value is -1 in the provided heat templates; meaning any destination port.</p>

Alarm

The alarm parameters are used in service scaling and are not included in the service chaining environment files. These parameters define the thresholds used by Contrail to determine when scaling should take place. This set of parameters is only used the service scaling heat template.

Alarm	
meter_name	The metric monitored by Celiometer and used by contrail to determine when an additional VM-Series firewall should be deployed or brought down. The heat template uses CPU utilization or bytes per second as metrics for service scaling.
cooldown_initial	The amount time Contrail waits before launching a additional service instance after the initial service instance is launched. The default is 1200 seconds.
cooldown_scaleup	The amount of time Contrail waits between launching additional service instance after the first scale up service instance launch. The default is 1200 seconds.
cooldown_scaledown	The amount of time Contrail waits between shutting down additional service instances after the first scale up service instance shut down. The default is 1200 seconds.
period_high	The interval during which the average CPU load is calculated as high before triggering an alarm. The default value is 300 seconds.
period_low	The interval during which the average CPU load is calculated as low before triggering an alarm. The default value is 300 seconds.
threshold_high	The value of CPU utilization in percentage or bytes per second that Contrail references before launching a scale up event. The default is 40% CPU utilization or 2800 bytes per second.
threshold_low	The value of CPU utilization in percentage or bytes per second that Contrail references before launching a scale down event. The default is 20% CPU utilization or 12000 bytes per second.

Install the VM-Series Firewall as a Service Chain

Complete the following steps to prepare the heat templates, bootstrap files, and software images needed to deploy the VM-Series firewall. After preparing the files, deploy the VM-Series firewall service and two Linux servers.

Install the VM-Series Firewall as a Service Chain	
Step 1 Download the Heat template and bootstrap files.	Download the Heat template package from the GitHub repository .
Step 2 Download the VM-Series base image.	<ol style="list-style-type: none"> 1. Login in to the Palo Alto Networks Customer Support Portal. 2. Select Software Updates and choose PAN-OS for VM-Series KVM Base Images from the Filter By drop-down. 3. Download PA-VM-KVM-8.0.0.qcow2.
Step 3 Download Ubuntu 14.04 and upload the image to the OpenStack controller. The Heat template needs an Ubuntu image for launching the Linux server.	<ol style="list-style-type: none"> 1. Download Ubuntu 14.04. 2. Log in to the Horizon UI. 3. Select Project > Compute > Images > Create Image. 4. Name the image Ubuntu 14.04 to match the parameter in the pan_basic_gw_env.yaml file. 5. Set Image Source to Image File. 6. Click Choose File and navigate to your Ubuntu image file. 7. Set the Format to match the file format of your Ubuntu image. 8. Click Create Image.
Step 4 Upload the VM-Series for KVM base image to the OpenStack controller.	<ol style="list-style-type: none"> 1. Log in to the Horizon UI. 2. Select Project > Compute > Images > Create Image. 3. Name the image PA-VM-8.0.0. 4. Set Image Source to Image File. 5. Click Choose File and navigate to your VM-Series image file. 6. Set the Format to QCOW2-QEMU Emulator. 7. Click Create Image.
Step 5 Upload the bootstrap files. The files must be uploaded to the folder structure described here. The heat template uses this folder structure to locate the bootstrap files.	<ol style="list-style-type: none"> 1. Login to your OpenStack controller. 2. Create the following folder structure /root/bootstrap/config/ /root/bootstrap/license/ 3. Using SCP or FTP, add the init-cfg.txt and bootstrap.xml files to the config folder and add your VM-Series auth codes to the license folder.

Install the VM-Series Firewall as a Service Chain

Step 6 Edit the template environment file to suit your environment. Verify that the image names in the environment file match the names you gave the files when you uploaded them.

```
parameters:
# VN config
  management_network: 'mgmt_net'
  left_vn: 'left_net'
  right_vn: 'right_net'
  left_vn_fqdn: 'default-domain:admin:left_net'
  right_vn_fqdn: 'default-domain:admin:right_net'
  route_target: "target:64512:20000"
# VM config
  flavor: 'm1.small'
  left_vm_image: 'TestVM'
  right_vm_image: 'TestVM'
  svm_name: 'PAN_SVM_L3'
  left_vm_name: 'Left_VM_L3'
  right_vm_name: 'Right_VM_L3'
  port_tuple_name: 'port_tuple_L3'
#ST Config
  S_Tmp_name: PAN_SVM_template_L3
  S_Tmp_version: 2
  S_Tmp_service_mode: 'in-network'
  S_Tmp_service_type: 'firewall'
  S_Tmp_image_name: 'PA-VM-8.0.0'
  S_Tmp_flavor: 'm1.large'
  S_Tmp_interface_type_mgmt: 'management'
  S_Tmp_interface_type_left: 'left'
  S_Tmp_interface_type_right: 'right'
  domain: 'default-domain'
# SI Config
  S_Ins_name: PAN_SVM_Instance_L3
  S_Ins_fq_name: 'default-domain:admin:PAN_SVM_Instance_L3'
#IPAM Config
  NetIPam_ip_prefix_mgmt: '172.2.0.0'
  NetIPam_ip_prefix_len_mgmt: 24
  NetIPam_ip_prefix_left: '10.10.1.0'
  NetIPam_ip_prefix_len_left: 24
  NetIPam_ip_prefix_right: '10.10.2.0'
  NetIPam_ip_prefix_len_right: 24
  NetIPam_addr_from_start_true: true
#Policy Config
  policy_name: 'PAN_SVM_policy-L3'
  policy_fq_name: 'default-domain:admin:PAN_SVM_policy-L3'
  simple_action: 'pass'
  protocol: 'any'
  src_port_end: -1
  src_port_start: -1
  direction: '<>'
  dst_port_end: -1
  dst_port_start: -1
```

Step 7 Upload the heat template files.

1. Login to your OpenStack Controller.
2. Use SCP or FTP to add the heat template file and environment file at the root level.

Install the VM-Series Firewall as a Service Chain	
<p>Step 8 Deploy the Heat template.</p> <ol style="list-style-type: none"> 1. Execute the command <code>source openrc</code> 2. Execute the command <code>heat stack-create <stack-name> -f <template> -e <env-template></code> 	
<p>Step 9 Verify that your VM-Series firewall is deployed successfully.</p>	<p>You can use the following commands to check the creation status of the stack.</p> <ul style="list-style-type: none"> • Check the stack status with <code>heat stack-list</code> • View a detailed list of events that occurred during stack creation with <code>heat event-list</code> • View details about your stack with <code>heat stack-show</code>
<p>Step 10 Verify that the VM-Series firewall is bidirectionally inspecting traffic between the Linux servers.</p>	<ol style="list-style-type: none"> 1. From an external network, execute the command <code>ssh -i <server-key>@<pan_untrust_floating_ip></code> 2. Log in to the firewall and select Monitor > Logs > Traffic to view the ssh session.