



Avi Plugins in Kolla-Ansible Based Deployment

Avi Technical Reference (v20.1)

Copyright © 2020

Avi Plugins in Kolla-Ansible Based Deployment

[view online](#)

Overview

This guide explains the steps to be followed to install Avi OpenStack plugins in Kolla-Ansible based OpenStack deployments.

Environment

Following are the environment details:

- OpenStack version: Queens and Rocky
- Deployment: Kolla-Ansible based deployment
- Base image used for building images: CentOS
- Ansible Inventory: all-in-one
- Avi Heat plugin version to be installed: 17.2.7

Deploying Avi Plugins in Kolla-Ansible

Kolla-Ansible provides a way to deploy OpenStack services and customize them. It uses ansible and docker to run the services on hosts and uses Jinja templates for customization. Kolla-Ansible is usually installed on an ansible control host, and Kolla-Ansible commands are executed from there. Kolla builds are also carried out on the same host using Kolla-build command. Kolla-build builds the OpenStack service container images and provides options to customize the image to include third-party plugins.

Avi Vantage has following plugins: * [Avi Heat Plugin](#)

Customizations are done using combination of Jinja templates and config files. Using templates, you can install additional packages like third-party plugins in image. Additional configuration is provided to the plugin/driver using the custom config files.

Customization to existing templates are given in a Jinja-based template which extends the parent template. Parent templates are the templates from upstream for OpenStack services and it defines the contents of image.

Custom configurations are provided from node_custom_config directory usually located in /etc/kolla/config directory. You can change this in Kolla-Ansible's globals.yml file. For instance, if you need to provide extra config to heat service, create a file in /etc/kolla/config/heat.conf directory and provide additional configuration in it. Kolla-build will merge this config file with the upstream config file and provide it in heat container images.

Avi Heat Plugin

Avi Heat Plugin is a heat resource module for a heat-engine. This plugin is loaded by heat-engine. Avi Heat Plugin provides Avi resources and handlers for them. On a Kolla-based deployment, you need to install Avi Heat Plugin in heat-engine image.

The following are the steps to deploy Avi Heat Plugin:

Step 1: On Kolla build host, create a Jinja template as follows to add Avi heat plugin to heat-engine image, template-override.j2.

```
{ extends parent_template }
{ block heat_engine_footer }
RUN yum install -y epel-release && yum install -y python-pip \
    && git clone https://github.com/avinetworks/avi-heat.git \
    && cd avi-heat && git checkout -b 17.2.7 origin/17.2.7 \
    && pip --no-cache-dir install . && cd -
{ endblock }
```

Note: This format will be maintained for both source-build types and binary-build types.

You can use the following template to directly use the RPM/DEB packages, if available:

```
{ extends parent_template }
{ block heat_engine_footer }
RUN yum install -y wget \
    && wget https://github.com/avinetworks/avi-heat/releases/download/v17.2.7/aviheat-17.2.7b201812211855-1.noarch.rpm
    && yum install -y aviheat-17.2.7b201812211855-1.noarch.rpm
{ endblock }
```

The link to RPM/DEB packages for a particular release of Avi-Heat Plugin is available in <https://github.com/avinetworks/avi-heat/releases>

Step 2: Run the following command to build heat-engine image:

```
python tools/build.py --template-override template-overrides.j2 --tag 'queens' heat-engine
```

OR,

if you are using Kolla-build, run the following command:

```
kolla-build --template-override template-overrides.j2 --tag 'queens' heat-engine
```

This will build a heat-engine image as follows with Avi plugin installed in it:

```
kolla/centos-binary-heat-engine queens e1a090136e5a About an hour ago 940.5 MB
```

Note: The image name can change based on Linux distribution being used.

Step 3: Add the Avi Heat Plugin specific configuration to Kolla custom config. Usually Kolla custom config is located in /etc /kolla/config directory:

```
[root@kolla ~]# cat /etc/kolla/config/heat.conf
[DEFAULT]
plugin_dirs=/usr/lib/python2.7/site-packages/avi/heat
avi_controller=Avi Controller VIP
```

For Ubuntu based distro, the `plugin_dirs` changes to:

```
/usr/local/lib/python2.7/dist-packages/avi/heat
```

Note: The custom config directory is configurable and you need to check it in `/etc/kolla/globals.yaml` directory, with variable name: `node_custom_config`.

Step 4: Once you have the custom-image and config in place, you need to reconfigure the services. Kolla-Ansible should pick this newly built custom image, for which, you need to add the below mentioned code in `/etc/koll/globals.yml` directory:

```
Custom image of heat heat_engine_image: centos-binary-heat-engine
```

However, you can also manually tag the image and give that image name here.

```
docker tag img-id kolla/heat-avi-plugin:queens
```

Step 5: Run Kolla-Ansible reconfigure as follows:

```
kolla-ansible -i INVENTORY reconfigure
```

This will deploy the newly built heat engine container.

Step 6: Verify if Avi Heat Plugin is installed by using `heat resource-type-list` command. Avi Heat Resources should be available in output.

```
(avi-dev-venv) ~ $> heat resource-type-list | grep Avi
WARNING (shell) "heat resource-type-list" is deprecated, please use "openstack orchestration resource type list" instead
| Avi::LBaaS::ActionGroupConfig |
| Avi::LBaaS::AlertConfig |
| Avi::LBaaS::AlertEmailConfig |
| Avi::LBaaS::AlertObjectList |
| Avi::LBaaS::AlertScriptConfig |
| Avi::LBaaS::AlertSyslogConfig |
| Avi::LBaaS::AnalyticsProfile |
| Avi::LBaaS::Application |
| Avi::LBaaS::ApplicationPersistenceProfile |
| Avi::LBaaS::ApplicationProfile |
| Avi::LBaaS::AuthProfile |
| Avi::LBaaS::AutoScaleLaunchConfig |
| Avi::LBaaS::BackupConfiguration |
| Avi::LBaaS::CertificateManagementProfile |
| Avi::LBaaS::Cloud |
| Avi::LBaaS::CloudConnectorUser |
| Avi::LBaaS::CloudProperties |
| Avi::LBaaS::Cluster |
| Avi::LBaaS::ClusterCloudDetails |
| Avi::LBaaS::ControllerLicense |
| Avi::LBaaS::ControllerProperties |
| Avi::LBaaS::CustomIpamDnsProfile |
| Avi::LBaaS::DebugController |
| Avi::LBaaS::DebugServiceEngine |
```

```
| Avi::LBaaS::DebugVirtualService  
| Avi::LBaaS::DnsPolicy  
| Avi::LBaaS::ErrorPageBody  
| Avi::LBaaS::ErrorPageProfile  
| Avi::LBaaS::Gslb  
| Avi::LBaaS::GslbGeoDbProfile  
| Avi::LBaaS::GslbService  
| Avi::LBaaS::GslbSite  
| Avi::LBaaS::GslbThirdPartySite  
| Avi::LBaaS::HTTTPolicySet  
| Avi::LBaaS::HardwareSecurityModuleGroup  
| Avi::LBaaS::HealthMonitor  
| Avi::LBaaS::IpAddrGroup  
| Avi::LBaaS::IpAddrGroup::Addr  
| Avi::LBaaS::IpamDnsProviderProfile  
| Avi::LBaaS::L4PolicySet  
| Avi::LBaaS::MicroService  
| Avi::LBaaS::MicroServiceGroup  
| Avi::LBaaS::Network  
| Avi::LBaaS::NetworkProfile  
| Avi::LBaaS::NetworkSecurityPolicy  
| Avi::LBaaS::PKIProfile  
| Avi::LBaaS::Pool  
| Avi::LBaaS::Pool::Server  
| Avi::LBaaS::PoolGroup  
| Avi::LBaaS::PoolGroupDeploymentPolicy  
| Avi::LBaaS::PriorityLabels  
| Avi::LBaaS::Role  
| Avi::LBaaS::SSLKeyAndCertificate  
| Avi::LBaaS::SSLProfile  
| Avi::LBaaS::Scheduler  
| Avi::LBaaS::SeProperties  
| Avi::LBaaS::ServerAutoScalePolicy  
| Avi::LBaaS::ServiceEngine  
| Avi::LBaaS::ServiceEngineGroup  
| Avi::LBaaS::SnmpTrapProfile  
| Avi::LBaaS::StringGroup  
| Avi::LBaaS::SystemConfiguration  
| Avi::LBaaS::TrafficCloneProfile  
| Avi::LBaaS::UserAccountProfile  
| Avi::LBaaS::VIMgrHostRuntime  
| Avi::LBaaS::VIMgrVcenterRuntime  
| Avi::LBaaS::VSDataScriptSet  
| Avi::LBaaS::VirtualService  
| Avi::LBaaS::VrfContext  
| Avi::LBaaS::VsApicExtension  
| Avi::LBaaS::VsVip  
| Avi::LBaaS::WafCRS  
| Avi::LBaaS::WafPolicy  
| Avi::LBaaS::WafProfile  
| Avi::LBaaS::Webhook
```

```
(avi-dev-venv) ~ $>
```