# AIC Valet Service Deployment Guide

Contents

[AIC Valet Service Deployment Guide 1](#_Toc452996367)

[1. Valet High Availability Design 2](#_Toc452996368)

[2. Configuring Clustered Valet 4](#_Toc452996369)

[2.1. Prerequisites 4](#_Toc452996370)

[2.2. Configuring Valet on Host Valet1 4](#_Toc452996371)

[2.2.1. HaValet 4](#_Toc452996372)

[2.2.2. Valet API 4](#_Toc452996373)

[2.2.3. Ostro 4](#_Toc452996374)

[2.2.4. Music 5](#_Toc452996375)

[2.2.5. Zookeeper 5](#_Toc452996376)

[2.2.6. Cassandra 5](#_Toc452996377)

[2.2.7. Supervisor (HaValet watchdog) 5](#_Toc452996378)

[2.2.8. Ostro Event Listener 5](#_Toc452996379)

[2.3. Configuring Valet on Host Valet2 5](#_Toc452996380)

[2.3.1. HaValet 5](#_Toc452996381)

[2.3.2. Valet API 5](#_Toc452996382)

[2.3.3. Ostro 6](#_Toc452996383)

[2.3.4. Music 6](#_Toc452996384)

[2.3.5. Zookeeper 6](#_Toc452996385)

[2.3.6. Cassandra 6](#_Toc452996386)

[2.3.7. Supervisor (HaValet watchdog) 6](#_Toc452996387)

[2.4. Configuring Valet on Host Valet3 7](#_Toc452996388)

[2.4.1. HaValet 7](#_Toc452996389)

[2.4.2. Valet API 7](#_Toc452996390)

[2.4.3. Ostro 7](#_Toc452996391)

[2.4.4. Music 7](#_Toc452996392)

[2.4.5. Zookeeper 7](#_Toc452996393)

[2.4.6. Cassandra 7](#_Toc452996394)

[2.4.7. Supervisor (HaValet watchdog) 7](#_Toc452996395)

[3. Adding Valet filter to OpenStack cluster - Nova, Heat. 8](#_Toc452996396)

[3.1. Nova 8](#_Toc452996397)

[3.2. Heat 9](#_Toc452996398)

[4. Appendix 9](#_Toc452996399)

[4.1. HAProxy configuration bigsite sample 9](#_Toc452996400)

[4.2. Silent SSH 10](#_Toc452996401)

# Valet High Availability Design

This document describes the distribution of the Valet components on 3 dedicated hosts and the different configuration of the various components on these hosts.

* Valet hosts should be instantiated from image found in : <http://valet.research.att.com/> (doc/aic)

Valet-openstack archive for AIC Valet 1.0: <http://valet.research.att.com/> (doc/aic)

* Valet hosts have 3 execution setups:
* Valet1 runs all Valet components (Ostro in primary mode).
* Valet2 runs all Valet components (Ostro in secondary mode, no Ostro events listener).
* Valet3 runs only Music components (Zookeeper, Cassandra and Tomcat).
* The different Valet flavors – configuration-wise – are described [below](#_Configuring_Clustered_Valet)
* HAProxy receives Valet REST API calls on port 8090 from the Valet Heat Stack Lifecycle plugin and Nova Filter Scheduler plugin (ValetFilter). HAProxy balances Valet API calls between Valet1 and Valet2 (Valet1:8090, Valet2:8090).

It is assumed HAProxy already installed in AIC env.

* HAValet is running only on Valet1 and Vale2 hosts.
* HAValet monitors the following:
* valet-api (local wsgi/pecan)
* Ostro – local and remote
* HAValet does not monitor Music components. Apache Tomcat, Zookeeper, and Cassandra are monitored via a cron job: /opt/valet/musicHealthCheck.sh.
* Ostro HA: HAValet gives precedence to the Ostro instance on the primary host.

HAValet queries Ostro for its priority. When two Ostro instances are found, HAValet shuts down the Ostro instance with the lower priority.

When the Ostro instance with the higher priority fails to start, HAValet starts the second instance.

* HAValet and the Ostro Event Listener (ostro-listener) are watched by the supervisor (Linux watch-dog).

Valet High Availability Architecture

HAProxy

Valet2 Valet1



Tomcat



HAValet

Valet API

Ostro

ZK



Cassandra



Supervisor

Supervisor completely

Valet API

HAValet

Ostro

s

Secondary

Primary

Events Listener

Music

Tomcat

ZK

Cassandra

Valet3



Tomcat

ZK

Cassandra

# Configuring HA Valet

## Prerequisites

* Define hosts ‘Valet1’, ‘Valet2’, ‘Valet3’ in /etc/hosts file on 3 Valet hosts.
* Verify in /etc/ssh/sshd\_config : *PasswordAuthentication* set to yes
* Verify silent ssh between following hosts ():
  1. Valet1 🡪 Valet1
  2. Valet2 🡪 Valet2
  3. Valet1 🡨🡪Valet2

For detailed silent ssh procedure please see [appendix](#_Silent_SSH)

## Configuring Valet on Host Valet1

## HaValet

In /opt/valet/allegro/havalet/ha\_valet.cfg – set following properties in ‘Ostro’ section:

stand\_by\_list=Valet1, Valet2

host=Valet1

priority=1

user=< VALET HOST USER NAME>

## Valet API

Set in /opt/valet/allegro/valet\_api/config.py *identity* parameters according your cloud configuration.

Set in /opt/valet/allegro/valet\_api/config.py *messaging* parameters according your RabbitMQ cloud configuration.

## Ostro

Verify following parameters in /opt/valet/ostro/code/ostro/ostro.cfg

priority=1

control\_loc=< Controller IP >

ip=< Valet1 IP >

db\_hosts = Valet1, Valet2, Valet3

Set in /opt/valet/ostro/code/ostro/ostro.auth authentication parameters according your cloud configuration.

## Music

Verify run permissions on /opt/valet/musicHealthCheck.sh

## Zookeeper

## Cassandra

## Supervisor (HaValet watchdog)

## Ostro Event Listener

In /etc/ostro-listener/ostro-listener.conf verify that the parameters values aligned with your cloud configuration

host - compute node on which rabbitmq is running,

username - rabbitmq user.

Refer to <http://valet.research.att.com/> (“ostro-listener”) for further reference.

## Configuring Valet on Host Valet2

## HaValet

In /opt/valet/allegro/havalet/ha\_valet.cfg – **set** following properties in ‘Ostro’ section:

stand\_by\_list=Valet1, Valet2

host=Valet2

priority=2

user=< VALET HOST USER NAME>

## Valet API

Set in /opt/valet/allegro/valet\_api/config.py – in ‘music’ section

host: Valet2

Set in /opt/valet/allegro/valet\_api/config.py identity parameters according your cloud configuration.

Set in /opt/valet/allegro/valet\_api/config.py messaging parameters according RabbitMQ your cloud configuration.

## Ostro

Set following parameters in /opt/valet/ostro/code/ostro/ostro.cfg

Priority=2

control\_loc=<Controller IP>

ip=<Valet2 IP>

replication\_factor=3

db\_hosts = Valet1, Valet2, Valet3

Set in /opt/valet/ostro/code/ostro/ostroa.auth authentication parameters according your cloud configuration.

## Music

Verify run permissions on /opt/valet/musicHealthCheck.sh

## Zookeeper

Set /var/zookeeper/myid – to contain – 2

## Cassandra

## Supervisor (HaValet watchdog)

Stop supervisor:

sudo service supervisor stop

Kill ostro\_daemon and wsgi.py process

pgrep –f wsgi.py | sudo xargs kill -9

cd /opt/valet/ostro/code/ostro\_server ; sudo python ./ostro\_daemon.py stop

Start supervisor:

sudo service supervisor start

## Ostro Event Listener

Disable ostro-listener service

sudo service ostro-listener stop

sudo update-rc.d ostro-listener disable

## Configuring Valet on Host Valet3

## Supervisor (HaValet watchdog)

Not running on Valet3. Stop supervisor service.

sudo service supervisor stop

sudo update-rc.d supervisor disable

sudo mv /etc/supervisor/conf.d/HAValet.conf /etc/supervisor/conf.d/ HAValet.down

## HaValet

Not running on Valet3. Stop the process.

pgrep –f havalet | sudo xargs kill -9

## Valet API

Not running on Valet3. Stop the process.

pgrep –f wsgi.py | sudo xargs kill -9

## Ostro

Not running on Valet3. Stop the process.

cd /opt/valet/ostro/code/ostro\_server ; sudo python ./ostro\_daemon.py stop

## Music

Verify run permissions on /opt/valet/musicHealthCheck.sh

## Zookeeper

Set /var/zookeeper/myid – to contain - 3

## Cassandra

## Ostro Event Listener

Disable ostro-listener service

sudo service ostro-listener stop

sudo update-rc.d ostro-listener disable

# Installing valet\_os - adding Valet filter to OpenStack cluster - Nova, Heat.

Installing Valet Filter (need to be done on the Nova and Heat hosts)

Download and unpack [VALET\_OPENSTACK\_1.0.tar.gz](https://tspace.web.att.com/files/app#/folder/54c30069-3570-4f74-8c6e-96f90ec8d06e)

$ cd ./VALET\_OPENSTACK\_1.0/valet\_os

$ sudo pip install .

Refer to <http://valet.research.att.com/> (“valet-openstack”) for further details (also available as README in valet\_os directory)

## Nova

Add the following lines to /etc/nova/nova.conf:

Under the [DEFAULT] section (note for not dropping already exists filters):

scheduler\_available\_filters = nova.scheduler.filters.all\_filters

scheduler\_available\_filters = valet\_os.nova.valet\_filter.ValetFilter

scheduler\_default\_filters = RetryFilter, AvailabilityZoneFilter, RamFilter, ComputeFilter, ComputeCapabilitiesFilter, ImagePropertiesFilter, ServerGroupAntiAffinityFilter, ServerGroupAffinityFilter, ValetFilter

Add [Valet] section:

[valet]

url = http://valet\_proxy:8090/v1

failure\_mode = reject

admin\_tenant\_name = < set tenant name >

admin\_username = < set tenant admin name >

admin\_password = < set tenant admin password >

admin\_auth\_url = http://<keystone-server>:35357/v2.0

Restart nova-scheduler using separate stop and start directives:

$ sudo service nova-scheduler stop

$ sudo service nova-scheduler start

## Heat

The following changes are made in /etc/heat/heat.conf:

Set the plugin\_dirs option in the [DEFAULT] section so that Heat can locate and use the Valet Stack Lifecycle Plugin. If plugin\_dirs is already present, separate entries by commas. The order of entries does not matter.

[DEFAULT]

plugin\_dirs = /home/stack/allegro/valet\_os/etc/valet\_os/heat

stack\_scheduler\_hints = True

Add a [valet] section. This will be used by the Valet Stack Lifecycle Plugin:

**Note**: *valet\_api\_proxy* requires load balancing solution (e.g. [haproxy](#_Valet_High_Availability))

[valet]

url = <http://valet_api_proxy:8090/v1>

Restart heat-engine using separate stop and start directives:

$ sudo service heat-engine stop

$ sudo service heat-engine start

# Appendix

Appendix items are illustrations only and AIC integration can configure below items as needed to support the AIC environment.

The below configurations were used in the BIG-SITE system supported by AT&T Labs, Research.

# HAProxy configuration bigsite sample

HAProxy service installation (in case HAProxy is not already installed):

apt-get update

apt-get upgrade

apt-get install haproxy

set ENABLED=1 in /etc/default/haproxy

Starting the service (see conf sample below):

service haproxy start

HAProxy configuration bigsite sample:

maxconn 2000

errorfile 400 /etc/haproxy/errors/400.http

errorfile 403 /etc/haproxy/errors/403.http

errorfile 408 /etc/haproxy/errors/408.http

errorfile 500 /etc/haproxy/errors/500.http

errorfile 502 /etc/haproxy/errors/502.http

errorfile 503 /etc/haproxy/errors/503.http

errorfile 504 /etc/haproxy/errors/504.http

listen valet\_api 0.0.0.0:8090

mode http

stats enable

stats uri /haproxy?stats

balance roundrobin

no option http-server-close

http-check expect status 200

option forwardfor

server valet1 <valet1 IP>:8090 check

server valet2 <valet2 IP>:8090 check

# Silent SSH

In case needed this procedure explains how to establish silent ssh between two linux hosts.

Generate rsa key-pair:

ssh-keygen -f /home/attcloud/.ssh/valet\_id\_rsa -q -N ""

This will create two files:

/home/attcloud/.ssh/valet\_id\_rsa, /home/ attcloud /.ssh/valet\_id\_rsa.pub

Copy the content of:

/home/attcloud/.ssh/valet\_id\_rsa.pub into /home/attcloud/.ssh/authorized\_keys on the destination host

As some of the valet scripts are executed via ‘sudo’, silent ssh is required for both valet and root user. There for perform this procedure both as root and attcloud user.

Run ssh between hosts and hit ‘enter’ when asked to accept the destination host fingerprint

(this will set the host fingerprints in known\_hosts file).