



Installing Avi in AWS with Mesos Using the CLI

Avi Technical Reference (v16.3)

Copyright © 2018

Installing Avi in AWS with Mesos Using the CLI

[view online](#)

This guide describes how to install Avi Vantage into an Apache Mesos (Mesos) or Mesosphere DC/OS (DC/OS) cloud hosted by Amazon Web Services (AWS).

Avi Vantage integrates with Mesos or DC/OS and Marathon to provide the following services:

- Full-featured service proxy (distributed load balancing)
- Service discovery
- Autoscaling
- Application map and visibility

Note: This guide applies to installation using the CLI. To install using the AWS web interface and a CloudFormation template from Avi Networks instead, go [here](#).

If you are installing into an on-premises Mesos or DC/OS cloud, go [here](#) instead.

Avi Vantage consists of the Avi Controller and multiple Service Engines (SEs). The Avi Controller analyzes traffic and can request spin-up/spin-down of SEs to load balance traffic. In an Mesos or DC/OS deployment, the Avi Controller works with Marathon to spin-up/spin-down SEs.

Deployment Prerequisites

Physical Node Requirements

The main components of the Avi Vantage solution, Avi Controllers and Service Engines (SEs), run as containers on Mesos nodes. For production deployment, a 3-node Avi cluster is recommended, with the each of the Avi Controller nodes running on separate nodes. Each Avi SE is deployed as a container on one of the 3 Mesos nodes. Each Mesos node can run 1 SE container. The node on which the Avi Controller runs must meet at least the minimum system requirements, which are specified in [this article](#).

Software Infrastructure Requirements

For deployment of SEs, the following system-level software is required:

- Each node host OS must be a Linux distribution running systemd.
- One of the following is required for SE spin-up/spin-down:
 - CoreOS Fleet: Optional cluster management service for Apache Mesos. If Fleet is installed, the Avi Controller can use it to schedule Avi SE container starts/stops based on service health.
 - SSH: The Avi Controller requires passwordless sudo access on Mesos nodes to start/stop an Avi SE container. A SSH keypair can be configured in one of two ways.
 - Option 1: Import the private key of the AWS key pair used for DC/OS cluster creation. Click Import Private Key to import the private key into the Avi Controller.
 - Option 2: To instead generate a new key pair for SE creation, select Generate SSH Key Pair. Click Copy to clipboard, then copy the public key generated into the following file on each of the Mesos nodes: `/home/ssh_user/.ssh/authorized_keys`. `ssh_user` is the username of the AWS user.

Installing the Avi Controller

To install the Avi Controller:

1. Copy the .tgz package onto the Mesos node that will host the Avi Controller:

```
scp controller_docker.tgz username@remotehost.com:/some/local/directory
```

Note: Replace *username@remotehost.com* with your write-access username and password and the IP address or hostname for the host node.

2. Log onto the Mesos node:

```
ssh username@remotehost.com
```

3. Unzip the Avi Controller image:

```
gunzip controller.tgz
```

4. Load the Avi Controller image into the host's local docker repository:

```
sudo docker load -i controller_docker.tar
```

5. As a best practice, clean up any data that may be lingering from a previous run

```
sudo rm -rf /var/lib/controller/*
```

6. Use the vi editor to create a new file for spawning the Avi Controller service:

```
sudo vi /etc/systemd/system/avicontroller.service
```

7. Copy the following lines into the file:

```
[Unit]
Description=AviController
After=docker.service
Requires=docker.service

[Service]
Restart=always
RestartSec=0
TimeoutStartSec=0
TimeoutStopSec=120
StartLimitInterval=0
ExecStartPre=/usr/bin/docker kill avicontroller
ExecStartPre=/usr/bin/docker rm avicontroller
ExecStartPre=/usr/bin/bash -c "/usr/bin/docker run --name=avicontroller --privileged=true -p 5054:5054 -p 8053:53/udp -
ExecStart=/usr/bin/docker logs -f avicontroller
ExecStop=/usr/bin/docker stop avicontroller
```

```
[Install]
WantedBy=multi-user.target
```

8. Edit the following values in the file:

- NUM_CPU: Sets the number of CPU cores/threads used by the Controller (4 in this example).
- NUM_MEMG: Sets the memory allocation (12 GB in this example).
- DISK_GB: Sets the disk allocation (80 GB in this example).
- *interface*: Name of the default Ethernet interface (examples: `eth0`, `ens03`, `ens1616163`) with cluster access. `?` tag: Tag value of the Avi Vantage image in the Docker repository. For example, `?16.1-5000-20160212.235510?`.

9. Save and close the file.

Starting the Avi Controller Service

To start the Avi Controller, enter the following command at the OS shell prompt on the node where you installed the Avi Controller service:

```
sudo systemctl enable avicontroller && sudo systemctl start avicontroller
```

Initial startup and full system initialization takes around 5 minutes.

Accessing the Avi Controller Web Interface

Note: Avi Controller UI listens on ports 9080 & 9443. Allow 9080 & 9443 ports in the security group of the AWS instance where Avi Controller is running.

To access the Avi Controller web interface, navigate to the following URL:

```
https://mesos-ip-or-hostname:9443
```

The following section provides steps for initial configuration of the Avi Controller.

Setting Up the Avi Controller

This section shows how to perform initial configuration of the Avi Controller using its deployment wizard. You will configure the following settings.

Setup Parameters

1. Administrator account: Initial configuration of the Avi Controller begins with creation of an administrator account.
2. DNS and NTP servers:
 - DNS: If your deployment uses Mesos DNS, set the Avi Controller to use the Mesos Master as its DNS server. Otherwise, set the Avi Controller to use your network's DNS.
 - NTP: Can use the default or your local NTP server. The Avi Controller does not require use of the Mesos Master for network time.
3. Infrastructure settings for the Mesos cloud, and configuration settings for SEs:
 - Mesos URL: IP address or hostname of the Mesos Master.
 - Marathon URL: IP address or hostname of the Marathon instance in DCOS.
 - Service port range: Must match the service port range of Marathon. (The default is 10000-20000.)

- SE creation settings: Fleet or SSH. (See [Software Infrastructure Requirements](#).)
- Docker registry for SE creation: Access information for the Docker Registry. Enter the IP address or hostname, and the access credentials. You can use either Docker Hub or your cloud's private Docker Registry. You can change or customize settings following initial deployment using the Avi Controller's web interface.

Setup Procedure

To start, use a browser to navigate to the Avi Controller.

- Configure basic system settings:
 - Administrator account
 - DNS and NTP server information

The image displays two screenshots of the Avi Networks web interface. The left screenshot shows the 'Create Administrator Account' form. It features the Avi Networks logo at the top, followed by the title 'Avi Vantage' and 'Create Administrator Account'. The form includes four input fields: 'Username' (containing 'admin'), 'Password', 'Confirm Password', and 'Email Address (optional)' (containing 'john@doe.com'). A 'Create Account' button is at the bottom, and the version 'Version: 16.1' is displayed at the very bottom. The right screenshot shows the 'System Settings' form. It also features the Avi Networks logo and the title 'System Settings'. Below the title is a subtitle: 'Let's get started with some basic setup questions.' The form contains three input fields: 'DNS Address' (10.136.0.2), 'DNS Search Domain' (demo.avi.local), and 'NTP Address' (0.us.pool.ntp.org). There is a checked checkbox for 'Auto upload diagnostic information to Avi Networks, Inc.' and a green 'Next' button at the bottom right.

- Configure Mesos infrastructure settings:
 - Proxy Service Port Range:** The Avi Controller automatically creates a proxy Virtual Service (SE) for each Marathon application with a service_port in this range.
 - Non-Proxy Service Port Range:** The Avi Controller does not create a virtual service for any Marathon applications with a service_port in this range. These Marathon applications are bypassed during automatic virtual service creation.

Note: The Proxy Service Port Range and Non-Proxy Service Port Range fields specify the Marathon service_port ranges to use or ignore for automatic creation of proxy virtual services.

The screenshot shows the 'Mesos Infrastructure' configuration page. At the top is the Avi Networks logo. Below it, the 'Mesos Infrastructure' section contains the following fields:

- Mesos URL:** A text input field containing 'http://10.136.0.2:5050'.
- Marathon Framework:** A sub-section header.
- Marathon URL:** A text input field containing 'http://10.136.0.2:8080'.
- Username:** A text input field with the placeholder text 'Username'.
- Password:** A text input field with the placeholder text 'Password'.
- Proxy Service Port Range:** A range selector with 'min' and 'max' buttons. The 'min' value is '15000' and the 'max' value is '20000'.
- Non Proxy Service Port Range:** A range selector with 'min' and 'max' buttons, currently empty.

At the bottom left is a 'Previous' link, and at the bottom right is a green 'Next' button.

3. Configure settings for SE deployment.

- If using Fleet:

The screenshot shows the 'Service Engine Deployment' configuration page. At the top is the Avi Networks logo. Below it, the 'Service Engine Deployment' section contains the following fields:

- Service to Create Service Engine:** A radio button selection with 'Fleet' selected and 'SSH' unselected.
- Fleet Remote Endpoint URL:** A text input field containing 'http://10.136.1.2:4444'.

At the bottom left is a 'Previous' link, and at the bottom right is a green 'Next' button.

- If using SSH:

Note: The Avi Controller requires root access to the OS on the SE node to start the SE process on the node.

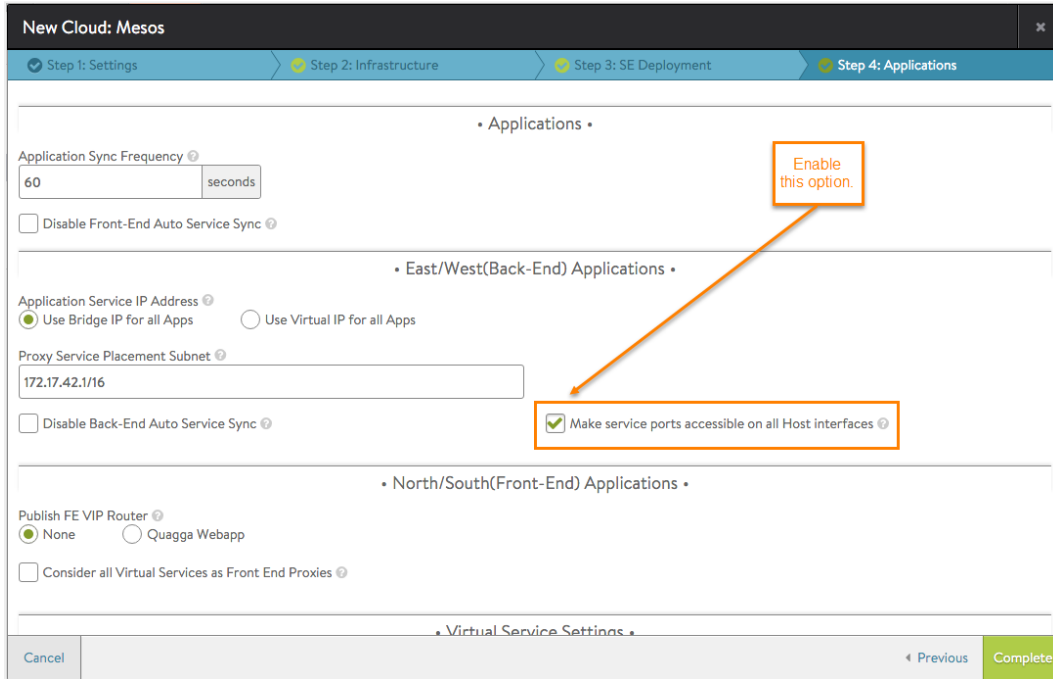
- If the key pair already exists, use Import Private Key to import the private key for each SE node into the Avi Controller.
- To instead generate a new key pair for SE creation, select Generate SSH Key Pair. Click Copy to clipboard, then copy the public key generated by the option into the following file on each of the SE nodes: `/home/ssh_user/.ssh/authorized_keys`

4. Configure Docker registry settings:

Make Service Ports Accessible on All Host Interfaces

After initial setup is completed using the wizard, the option to make service ports accessible on all host interfaces must be enabled. This is required for deployment into AWS.

1. Log onto the Controller using a browser.
2. Navigate to Applications > Infrastructure.
3. Select Clouds on the menu bar.
4. Click the edit icon in the row for the cloud.
5. Click the Applications tab.
6. Check (enable) the following option: Make service ports accessible on all Host interfaces



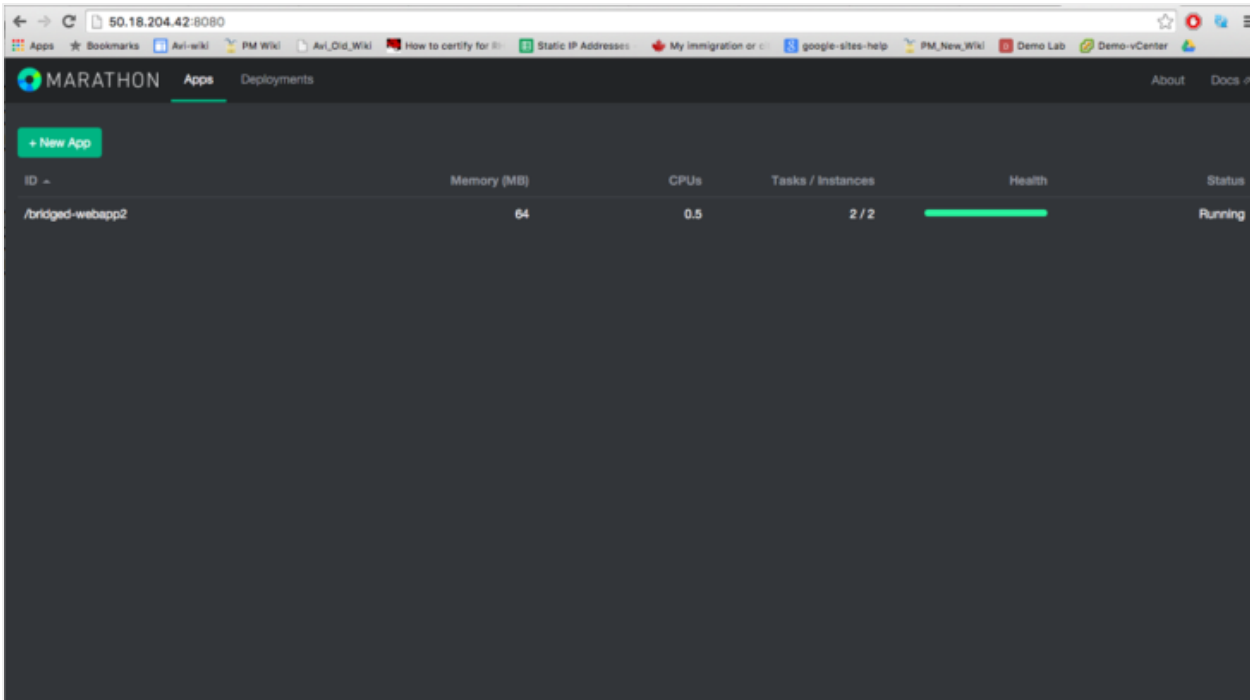
APPLICATION CREATION IN MARATHON

1. Download Docker.json from <https://s3-us-west-1.amazonaws.com/avi-tm/Docker.json>. Save the file in Downloads.
2. Open an SSH session to the Mesos master's IP address, and post the application:

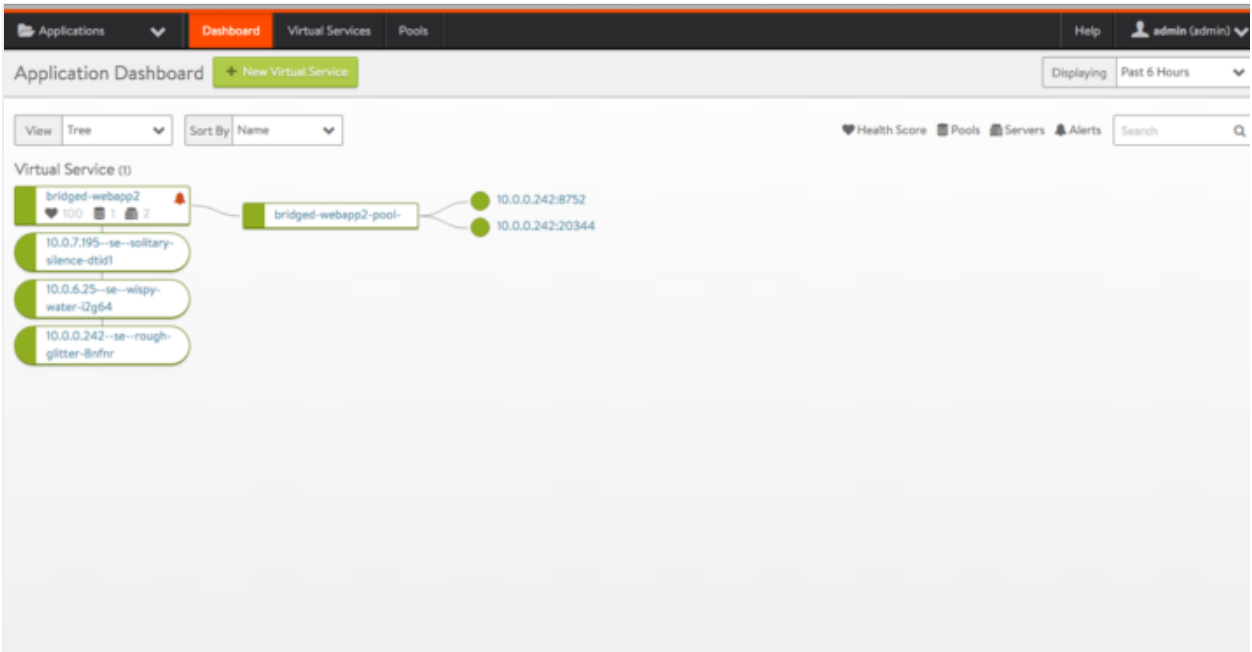
```
curl -H <span class="token string">"Content-Type: application/json"</span> -X POST -d@Downloads/Docker.json http
```

APPLICATION VERIFICATION

1. Log in to Marathon: <https://marathon-ip:8080>



2. Log in to the Avi Controller to verify that the application is getting load balanced.

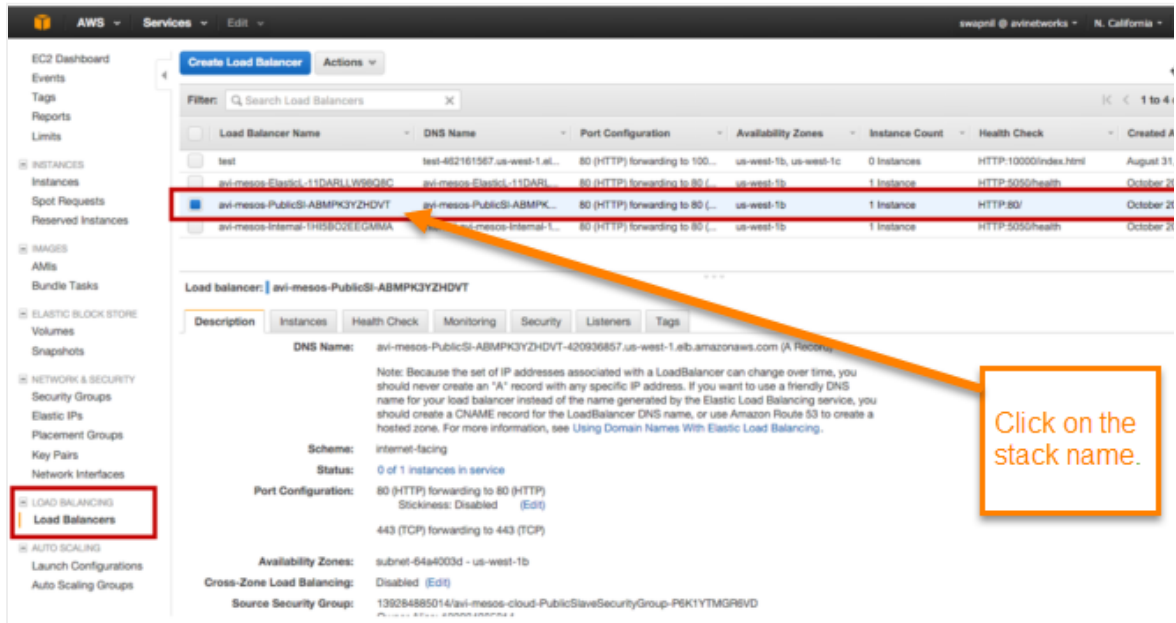


ACCESS WEB SERVICE THROUGH AVI VANTAGE

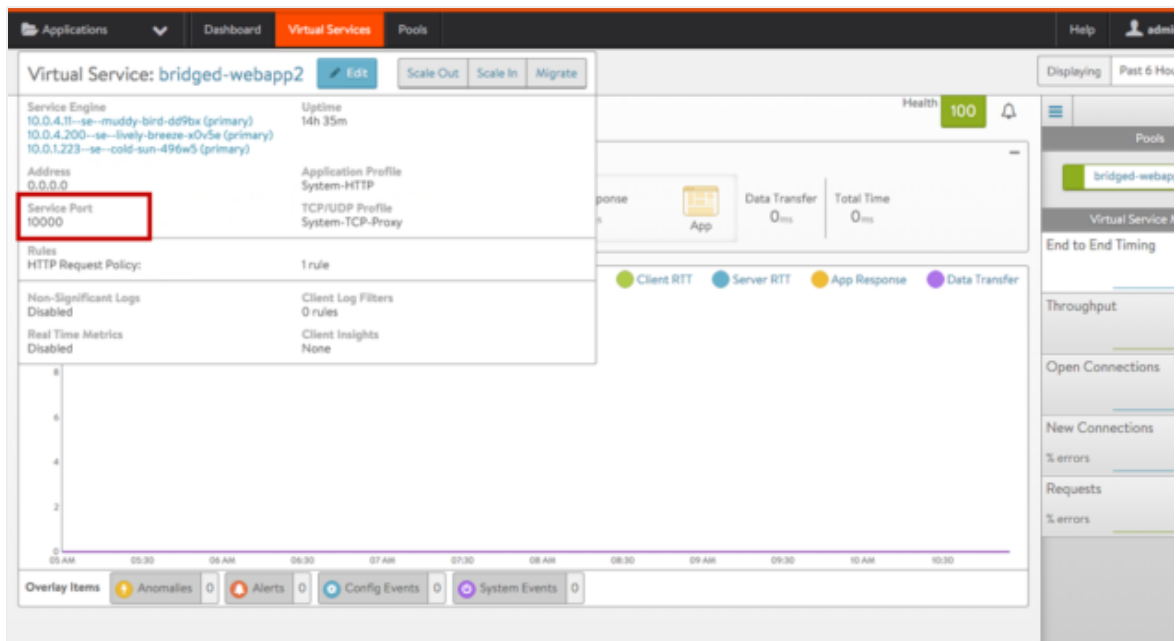
1. Browse to EC2 Load Balancing instance.
 1. Click on



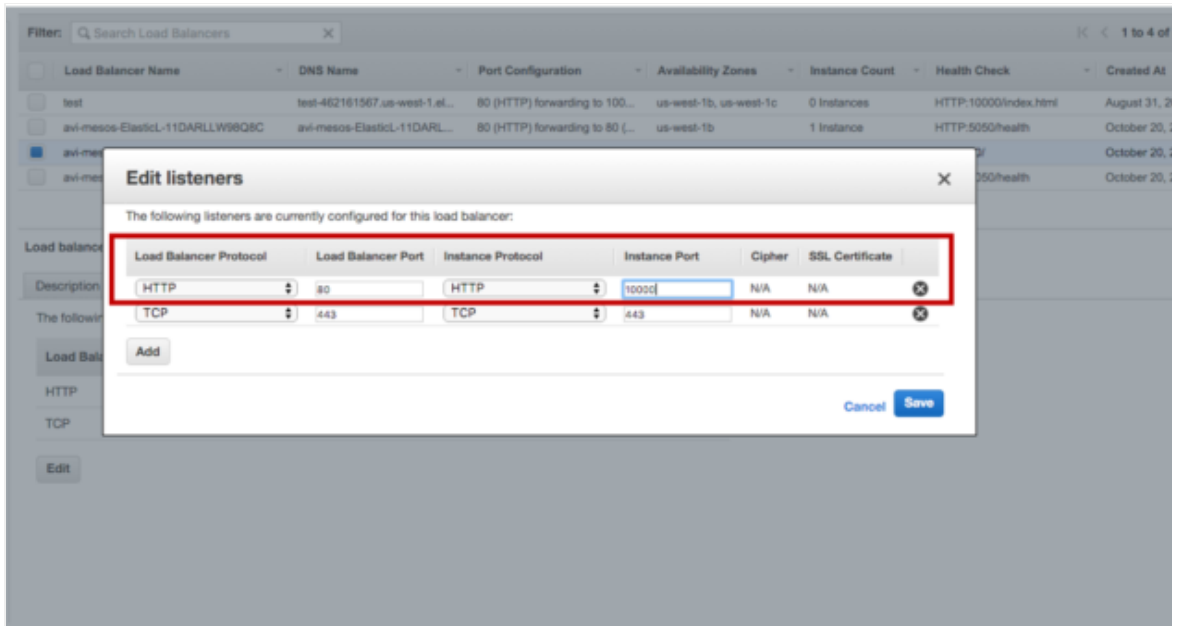
- to go to the AWS homepage.
- 2. Select EC2 and click on Load Balancing in the left pane.
- 3. Click on the ?PublicSlaveL? instance in the Load Balancer field (?avi-mesos-PublicSlaveL? in this example.)



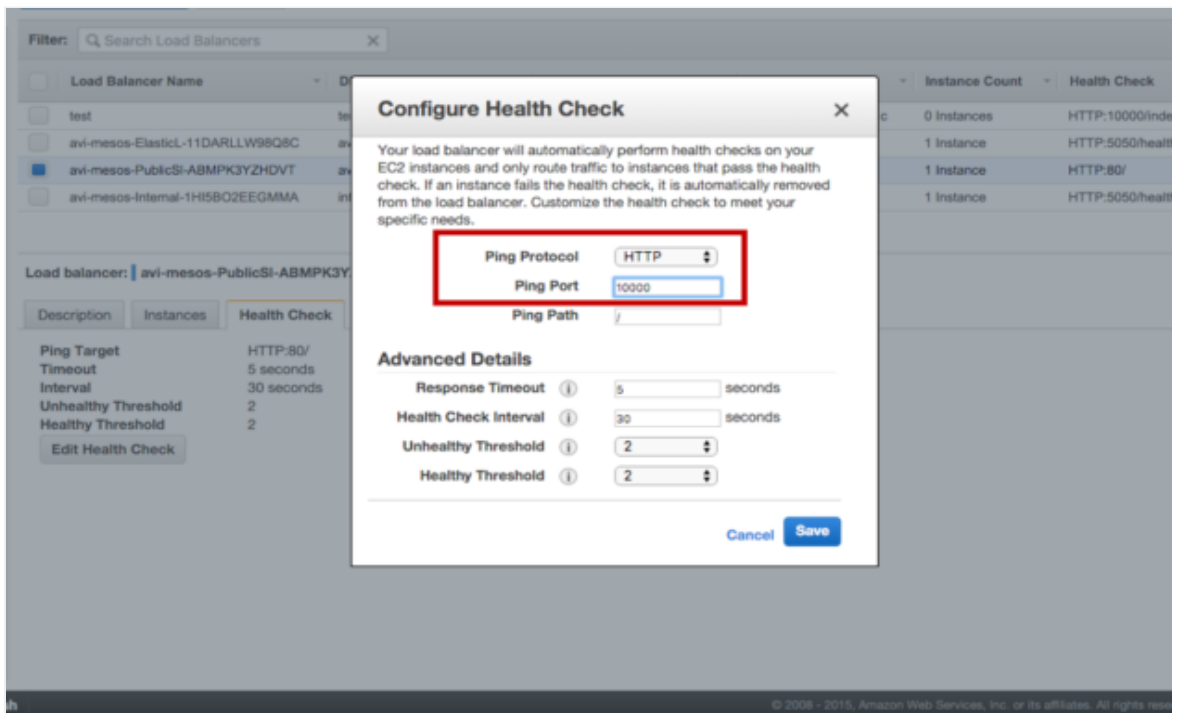
- 2. Edit the ELB instance to forward traffic to the application in the Mesos environment.
 - 1. Verify the application port from the Avi Controller web interface. Select the application and hover near the name.



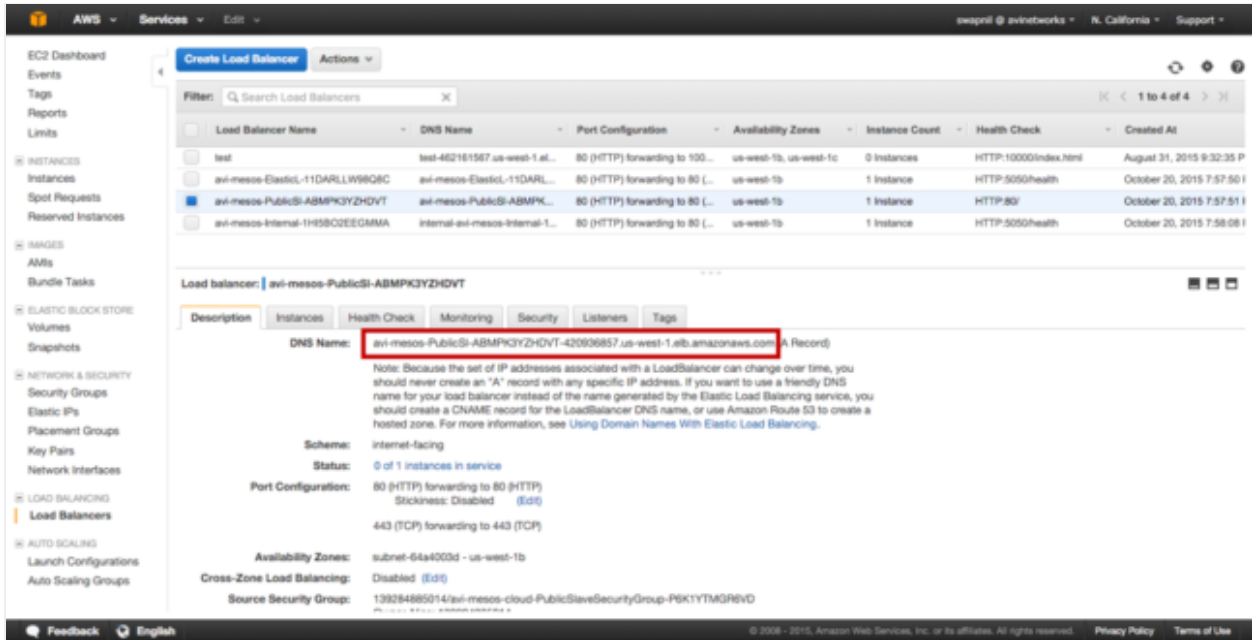
- 2. Click on Listener and edit ELB Listener to change the HTTP listener to the service (application) port number.



3. Click on Health Check and edit ELB health Check to the application port number.



3. Click on the Description tab for the same ELB instance. Copy the DNS name and paste it into a new browser tab.



1. Paste the DNS name into a new browser window.



2. Click on WebApp1 to verify that live traffic is flowing through.

