



# Installing Avi Vantage with Mesosphere DC/OS (16.2)

Avi Technical Reference (v17.2)

# Installing Avi Vantage with Mesosphere DC/OS (16.2)

[view online](#)

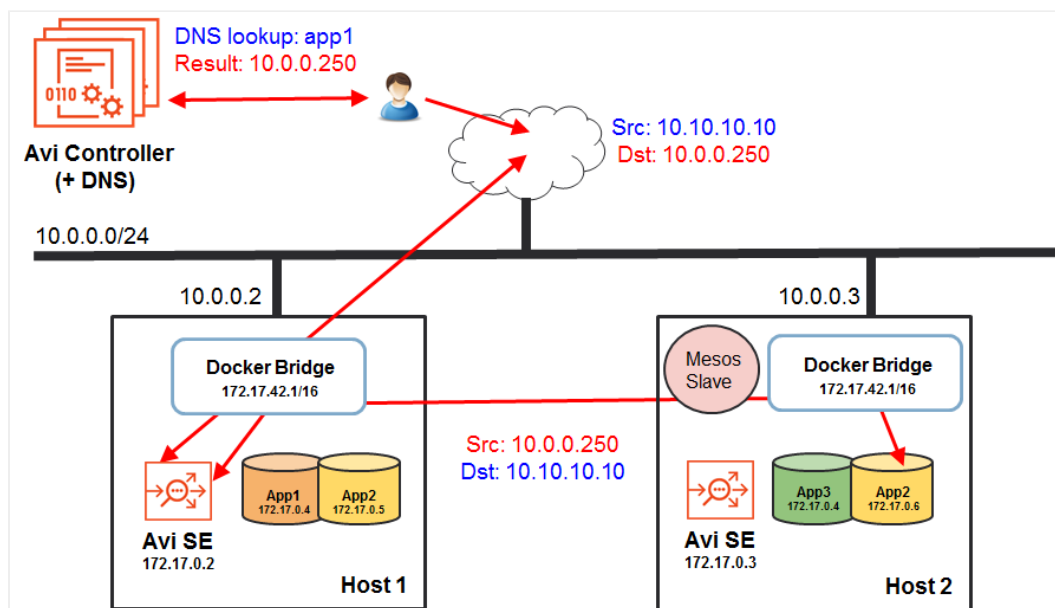
This guide describes how to install Avi Vantage into an on-premises (private) Mesos cloud. The instructions in this guide can be used for installing Avi Vantage 16.2.

Avi Vantage integrates with Mesos 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 within an on-premises cloud. If you are deploying into a Mesos cluster within Amazon Web Services (AWS), go [here](#) instead.

Here is how Avi Vantage integrates into a Mesos cloud:



As shown here, Avi Vantage consists of the Avi Controller cluster and multiple Service Engines (SEs). The Avi Controller analyzes traffic and can request spin-up/spin-down of SEs to load balance traffic. In a Mesos deployment, the Avi Controller works with Marathon to spin up/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 Controller cluster is recommended, with 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](#).

## System Time (NTP) Requirement

The system time on all nodes must be synchronized. Use of a Network Time Protocol (NTP) server is recommended.

## 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/down:
  - Fleet: Optional cluster management service for Mesos. If Fleet is installed, the Avi Controller can use it to schedule spin-up and spin?-down of SEs based on service health.
  - SSH: The Avi Controller uses password-less sudo SSH to access all the Mesos nodes in the cluster and create SEs on those nodes. The SSH user must have password-less sudo access to all the Mesos nodes in the Avi Vantage cluster. The SSH method requires a public-private key pair. You can import an existing private key onto the Avi Controller or generate a new key pair. In either case, the public key must be in the `~/home/ssh_user/.ssh/authorized_keys` file, where `ssh_user` is the SSH username on all Mesos nodes. The Avi Controller setup wizard automatically stores the private key on the Avi Controller node when you import or generate the key.

## 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/docker run --name=avicontroller --privileged=true -p 5054:5054 -p 8053:8053
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 (8 in this example).
- NUM\_MEMG: Sets the memory allocation (24 GB in this example).
- DISK\_GB: Sets the disk allocation (80 GB in this example).
- *interface*: Name of the default Ethernet interface (examples: eth0, ens0, en1671368, etc.) with access to the cluster. Tag value of the Avi Vantage image in the Docker repository. For example, 16.2-5000-20160212.235510.

### 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

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.

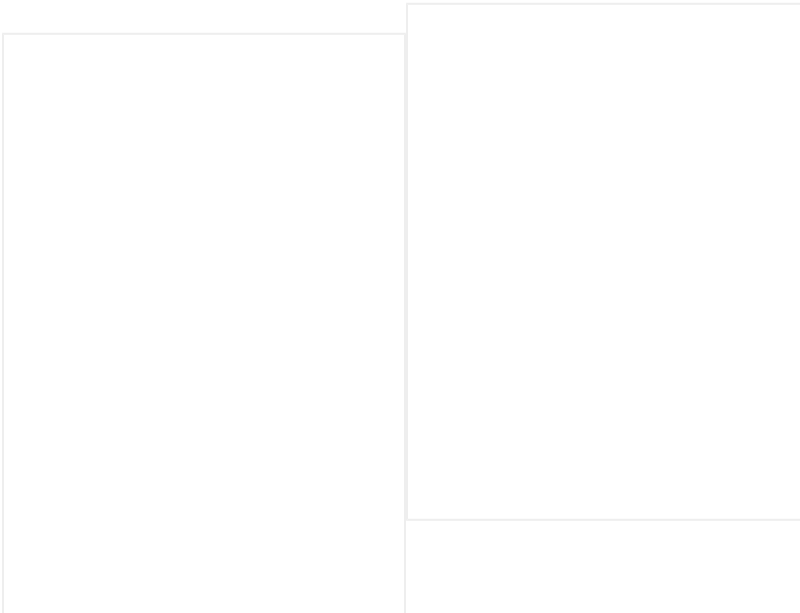
### Setup Parameters

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.
  - Marathon username and password.
4. SE creation settings: Fleet or SSH. (See [Software Infrastructure Requirements](#).)
5. 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

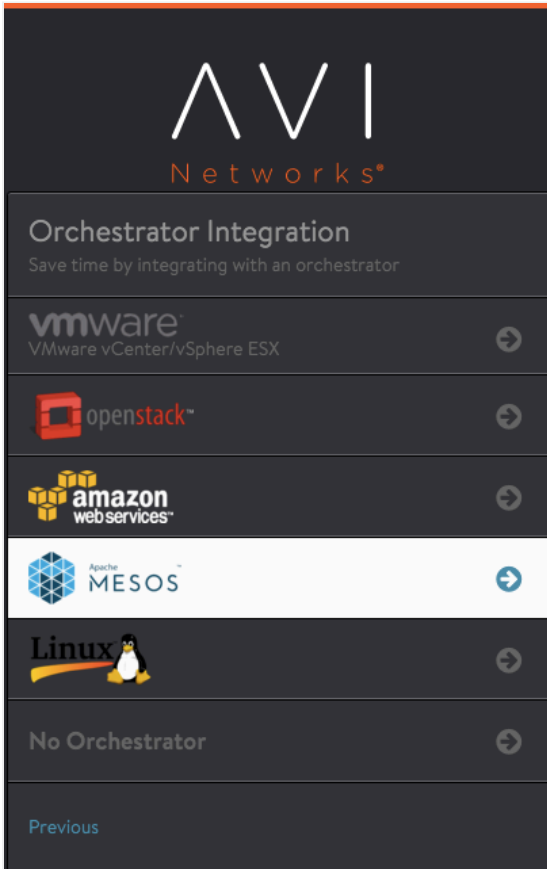
To start, use a browser to navigate to the Avi Controller.

1. Configure basic system settings:
  - Administrator account
  - DNS and NTP server information
  - Email and SMTP information





2. Select Mesos as the infrastructure type:



3. Configure Mesos infrastructure settings:

- Mesos URL: IP address or hostname of the Mesos Master
- Marathon URL: IP address or hostname of the Marathon instance in DC/OS
- Marathon username and password

4. Configure settings for SE deployment.

- If using Fleet:

The screenshot shows the 'Service Engine Deployment' screen in Fleet mode. The 'Service to Create Service Engine' section has 'Fleet' selected. The 'Fleet Remote Endpoint URL' field contains 'http://10.136.1.2:4444'. There are 'Previous' and 'Next' buttons at the bottom.

- If using SSH:

The screenshot shows the 'Service Engine Deployment' screen in SSH mode. The 'Service to Create Service Engine' section has 'SSH' selected. The 'SSH Username' field contains 'avuser'. The 'Generate SSH Key Value Pair' option is selected. There are 'Load File' and 'Remove Key' buttons. There is also a 'Passphrase' field with a 'Passphrase' label and a '\*' character. There are 'Previous' and 'Next' buttons at the bottom.

**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

##### 5. Configure Docker registry settings:

After you click Next, your Mesos deployment and initial Avi Controller configuration are complete.

## Creating and Testing Applications

This section provides steps for creating and testing virtual services for the following deployment scenarios:

- East-west: VIP and client are both in the same Mesos cluster.
- North-south with inside VIP: VIP is in the Mesos host network but client is outside the network.
- North-south with outside VIP: VIP and client are both outside the Mesos host network.

For each deployment scenario, the process is the same:

1. Create a Docker image file for the application.
2. Start the application.
3. Generate traffic.

The details for each deployment differ slightly, mainly in regard to network address information.

### East-west Application

Use these steps if the VIP and the client are both in the same Mesos cluster.

1. Create a Docker image file such as the following for the application:

```
{
  "id": "webappew1",
  "cpus": 0.5,
  "mem": 64.0,
  "instances": 2,
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "avinetworks/server",
      "network": "BRIDGE",
      "portMappings": [
        { "containerPort": 80, "hostPort": 0, "servicePort": 0, "protocol": "tcp" }
      ]
    }
  }
}
```



```

    }
  },
  "labels": {
    "avi_proxy": "{\"virtualservice\": {\"analytics_policy\": {\"metrics_realtime_update\": {\"duration\"
  },
  "healthChecks": [
    {
      "protocol": "HTTP",
      "portIndex": 0,
      "path": "/",
      "gracePeriodSeconds": 5,
      "intervalSeconds": 20,
      "maxConsecutiveFailures": 3
    }
  ]
}

```

2. Start the application to create a virtual service for it in Avi Vantage:

```
curl -H "Content-Type: application/json" -X POST -d@docker.json http://marathon-ip-or-hostname:8080/v2/apps
```

The H and X options are required. The H option inserts a Content-Type header for the Avi SE application. The X option changes the HTML method of the request from GET (the default) to POST. Replace marathon-ip-or-hostname with the IP address or hostname of Marathon.

3. Start a client container on the VM:

```
sudo docker run -d --name=aviclient avinetworks/server
```

4. Connect to the client container that you just started:

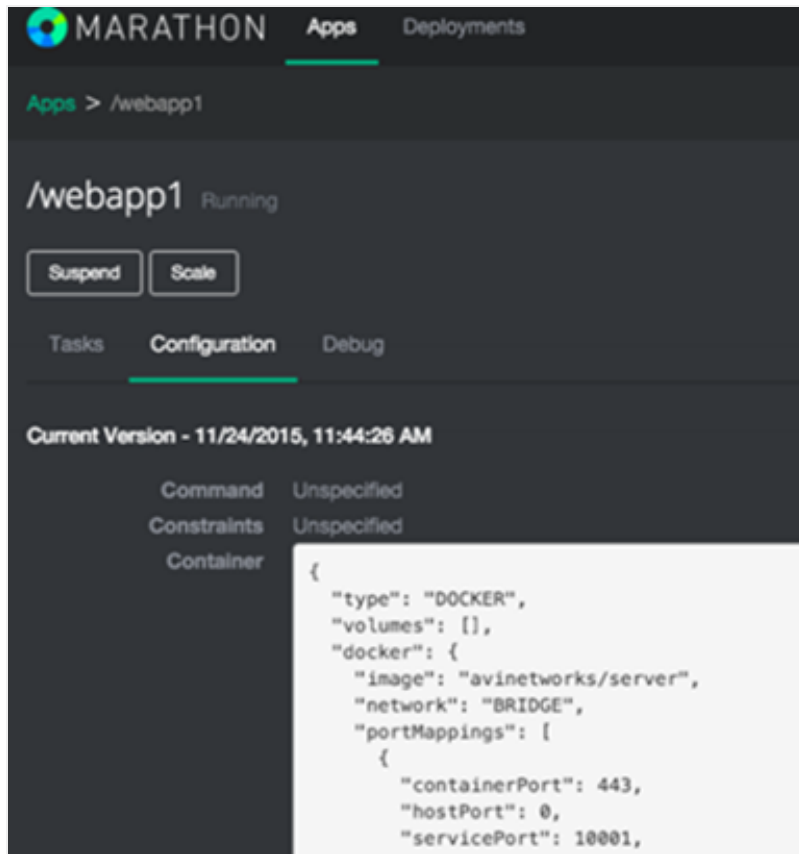
```
sudo docker exec -it aviclient bash
```

5. Generate test traffic:

```
ab -n 100 http://172.17.0.1:10001/100kb.txt
```

This command sends 100 requests for the specified file to the virtual service. The port number (10001 in this example) is the service port number assigned to the virtual service.

To view the service port number for the application, select the Configuration tab in the Marathon web interface. The virtual service service port number is specified in the Docker file by the servicePort parameter:



## North-south Application with Inside VIP

Use these steps if the VIP is in the same Mesos cluster as the application but the client is outside the cluster. In this example, the Mesos host and the VIP (10.10.10.100) are both in subnet 10.10.10.0/24.

1. Create a Docker image file such as the following for the application. In the file, edit the FE-Proxy?VIP to match the address of your VIP.

```
{
  "id": "webapp",
  "cpus": 0.5,
  "mem": 64.0,
  "instances": 2,
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "avinetworks/server",
      "network": "BRIDGE",
      "portMappings": [
        { "containerPort": 80, "hostPort": 0, "servicePort": 0, "protocol": "tcp" }
      ]
    }
  }
},
```

```

"labels": {
  "FE-Proxy": "yes",
  "avi_proxy": "{\"virtualservice\": {\"ip_address\": {\"addr\": \"10.90.28.250\", \"type\": \"V4\"}, \"servi
},
"healthChecks": [
  {
    "protocol": "HTTP",
    "portIndex": 0,
    "path": "/",
    "gracePeriodSeconds": 5,
    "intervalSeconds": 20,
    "maxConsecutiveFailures": 3
  }
]
}

```

2. Start the application to create a virtual service for it in Avi Vantage:

```
curl -H "Content-Type: application/json" -X POST -d@docker.json http://marathon-ip-or-hostname:8080/v2/apps
```

The H and X options are required. The H option inserts a Content-Type header for the Avi SE application. The X option changes the HTML method of the request from GET (the default) to POST. Replace marathon-ip-or-hostname with the IP address or hostname of Marathon.

3. To generate traffic to the application, open an SSH ssh connection to another VM that is located in the same network as the host, and enter the following command:

```
ab -n 100 http://10.10.10.100/100kb.txt
```

This command sends 100 requests for the specified file to the application VIP (10.10.10.100 in this example).

## North-south Application with Outside VIP

Use these steps if neither the VIP nor the client is in the same Mesos cluster as the application. In this example, the VIP is 20.20.20.20.

1. Create a Docker image file such as the following for the application. In the file, edit the FE-Proxy?VIP to match the address of your VIP.

```

{
  "id": "webapp",
  "cpus": 0.5,
  "mem": 64.0,
  "instances": 2,
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "avinetworks/server",
      "network": "BRIDGE",
      "portMappings": [

```

```

    { "containerPort": 80, "hostPort": 0, "servicePort": 0, "protocol": "tcp" }
  ]
}
},
"labels": {
  "FE-Proxy": "yes",
  "avi_proxy": "{\"virtualservice\": {\"ip_address\": {\"addr\": \"20.20.20.20\", \"type\": \"V4\"}, \"subnet\"
},
"healthChecks": [
  {
    "protocol": "HTTP",
    "portIndex": 0,
    "path": "/",
    "gracePeriodSeconds": 5,
    "intervalSeconds": 20,
    "maxConsecutiveFailures": 3
  }
]
}
}

```

## 2. Start the application to create a virtual service for it in Avi Vantage:

```
curl -H "Content-Type: application/json" -X POST -d@docker.json http://marathon-ip-or-?hostname:8080/v2/apps
```

The H and X options are required. The H option inserts a Content-Type header for the Avi SE application. The X option changes the HTML method of the request from GET (the default) to POST. Replace marathon-?ip-or-hostname with the IP address or hostname of Marathon.

## 3. To generate traffic to the application:

1. Open an SSH ssh connection to another VM that is located in the same network as the host.
2. Add a static host route that forwards traffic addressed to the VIP to the VM IP address:

```
ip route add 20.20.20.20/32 via

<vm-ip></vm-ip>
```

## 3. Generate traffic:

```
ab -n 100 http://20.20.20.20/100kb.txt
```

This command sends 100 requests for the specified file to the application VIP (20.20.20.20 in this example).

## Creating Virtual Services (applications)

After Avi Vantage is installed, virtual services (applications) can be created. For information:

- [Creating Applications with Mesos / Marathon Labels](#)
- [DNS-based Service Discovery for Mesos](#)

## Related Articles:

- [Upgrading Avi Vantage Software](#)
- [Upgrades in an Avi GSLB Environment](#)