

---

# **Mock Aerohive Documentation**

*Release 0.0.2*

**Ryan Leonard**

**Jun 28, 2018**



<b>1</b>	<b>PyUnit Fixture</b>	<b>3</b>
<b>2</b>	<b>Versioning</b>	<b>5</b>
2.1	Installation . . . . .	5
2.1.1	Via pip . . . . .	5
2.2	API Documentation . . . . .	5
2.2.1	Module . . . . .	6
2.2.2	Utilities . . . . .	6
2.3	Indices and tables . . . . .	6



A mock SSH server emulating Aerohive devices.

Install:

```
pip install mock-aerohive
```

Basic usage:

```
from mock_aerohive import MockAerohive

aerohive = MockAerohive()
# You must add at least 1 user before starting the server! (Library limitation)
aerohive.addUser("admin", "aerohive")

port = aerohive.run("127.0.0.1")
# Or provide a port: aerohive.run("127.0.0.1", 2222)

# Now you can SSH in:
# ssh admin@127.0.0.1 -p 2222

aerohive.stop() # Stop a single server.

aerohive.stopAll() # Terminate the background thread running all SSH servers.
↳ (otherwise the process will hang)
# Once you stop the background thread, you may not start another
↳ server (with 'run') -
# another library limitation.
```

For an example of a `pytest` fixture that automates starting and stopping servers (which cleans up servers at the end of the testing session, but allows multiple servers to be run), see `test/util/MockAerohiveFixture.py`, and `test/integration/auth/test_addUser_and_login.py` for an example.

Some Aerohive commands have been created, for instance, `hostname`:

```
ssh admin@127.0.0.1 -p 2222
admin@127.0.0.1's password:
Aerohive Networks Inc.
Copyright (C) 2006-2012
AH-2A0B00#hostname example-1
example-1#example-1#hostname example-2 invalid-extra-argument
^-- unknown keyword or invalid input
example-1#exit
Connection to 127.0.0.1 closed.
```



# CHAPTER 1

---

## PyUnit Fixture

---

If you are using PyUnit, this package ships with a test fixture to automatically create and stop mock servers for each test.

First, import the fixture into your test file:

```
from mock_aerohive import MockAerohiveFixture as MockAerohive
```

(I like to rename the imported fixture `MockAerohive`, but that's a personal preference.)

Then, add a parameter to your PyUnit test:

```
def test_some_feature(MockAerohive):
    aerohive = MockAerohive()
    aerohive.addUser("admin", "aerohive")
    port = aerohive.run("127.0.0.1")

    # Now you can SSH into the server
    # ...

    aerohive.stop()
```

Calling `aerohive.stop()` at the end of your test is optional. The fixture will automatically clean up all servers once all tests are complete, but you can stop each server if they might interfere with future tests.





This package uses semantic versioning.

## 2.1 Installation

### 2.1.1 Via pip

Install the package directly:

```
pip install mock-aerohive
```

Or add it to `requirements.txt`:

```
echo "mock-aerohive" >> requirements.txt
```

## 2.2 API Documentation

Mock Aerohive provides a Python API to setup and control SSH servers.

## 2.2.1 Module

`MockAerohive`

## 2.2.2 Utilities

`MockAerohiveFixture`

## 2.3 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)