

---

# Ankidb <sup>TM</sup> cloud v11 API testing document

document version 1.1

email <[support@yuktix.com](mailto:support@yuktix.com)<sup>1</sup>>

## 1. ankidb cloud machine on Linux

first step is to create an ankidb v11 cloud machine. To that end, we need a linux machine and install following software,

- influxdb (v1.2.x)
- redis (> 3.0)
- beanstalkd (latest version)
- mantiscore (for sphnix search engine)
- apache tomcat version 8.5.x
- apache maven
- git
- MySQL 5.7.x server
- Python 2.7.x (do not use Python 3.x)

The details of package install, list of packages and gotchas are documented in digital ocean install document.

Make sure all services are running

- mysql,
- influxdb,
- redis server
- beanstalkd
- tomcat 8.5.x

---

<sup>1</sup> <mailto:support@yuktix.com>

## 2. mac osx setup

most of the following software is installed via brew on macosx.

- `$mysql.server start`
- `$redis-server /usr/local/etc/redis.conf`
- `$influxd -config /usr/local/etc/influxdb.conf`
- `$beanstalkd -V` (add more V for more verbose output, e.g. `-VVVV`)
- `$HOME/sw/apache-tomcat-8.5.11/bin/startup.sh` - will start tomcat 8.x on 8080

start **JAVA** offline worker,

- `$cd <sensordb>/misc/offline-worker`
- `$copy latest worker jar` (see maven info on how to compile that)
- `cp <sensordb>/cloud/target/sensordb-worker.jar .`
- logging properties are in file `logging.properties`

`$/run.sh`

## 3. code and echo API testing

- checkout code from sensordb git repository
- use maven to create a war file
  - `cd /sensordb/cloud`
  - `mvn clean`
  - `mvn package`
- deploy the `.war` file on tomcat8 server copy from `cloud/target` folder
- verify that we can access tomcat using <http://localhost:8080/sensordb/v1/echo>



API access port can vary depending on the setup. If we want to test using apache tomcat directly, we should use port 8080. However on production testing machines, we turn off 8080 port and allow access using an nginx front end only.

## 4. MySQL and InfluxDB creation

- create database using sql script in <sensordb>/misc/mysql
- create influxdb database sensordb using influx client

mysql script files are

- sensordb/misc/newdb.sql
- sensordb/misc/schema.sql



Write down the api\_store keys values after schema creation. This same client\_key and secret\_key we have to use for pysdk. To do so,

```
insert into api_store(client_key, secret_key)
values (md5(LEFT(UUID(), 8)), SHA2(LEFT(UUID(), 16),256));

mysql/> select * from api_store

grant api permissions for global read and write. This key should be
deleted after creating an admin user.

mysql/> insert into api_permission(client_key, operation, permission)
values('client_key_from_above', 'read', '/resource/account');

mysql/> insert into api_permission(client_key, operation, permission)
values('client_key_from_above', 'write', '/resource/account');
```

To create sensordb database in influxdb,

```
$influx
> create databse sensordb ;
```

## 5. Setup pysdk

**environment variables**

- YUKTIX\_PY\_CONFIG and
- PYTHONPATH

Yuktix pysdk needs above environment variables.

- PYTHONPATH should be set to <pysdk>/lib folder
- YUKTIX\_PY\_CONFIG should be set to <pysdk>/conf/yuktix.conf file.

conf/ folder is not part of pysdk code repo and you should add it. yuktix.conf file contains configuration for pysdk.

### **configuration file.**

config file is a valid JSON file with keys and values. A server name or database name is just a key in this JSON file. An example is,

```
"cloudv1108june": {
  "_ds" : "table",
  "version" : 1 ,
  "user": "xxxx",
  "host": "127.0.0.1",
  "password": "xxxxxx",
  "port": "3306",
  "database": "cloudv1108june",
  "version" : 11
},

"eclipse": {
  "_docs" : "api_store.client_key and api_store.secret_key" ,
  "_ds" : "api",
  "endpoint": "http://localhost:8087/sensordb/v1",
  "admin_endpoint": "http://localhost:8087/sensordb/admin/v1",
  "monitor_endpoint": "http://localhost:8087/sensordb/monitor/v1",
  "client_key": "87a6bcb1785f9ca5e3f52f8f6e94eef6",
  "secret_key":
  "dd76bfbeceeb83be1840633f97093e6a09788a244d6543f8eba648a48c388571",
  "session_cookie" : "ce1b85b6-c7a2-4ed6-b020-474b45089538",
  "version" : 1
},
```

- endpoint and admin\_endpoint should point to REST API and admin service.
- client\_key and secret\_key should be copied from api\_store table.

## **6. seed data**

### **6.1. <pysdk>/tools/api/seed.py script**

\$tools/api/seed.py

This script will

- create an account
- add a user ([john.doe@yuktix.com](mailto:john.doe@yuktix.com)<sup>2</sup>)
- add customer admin role to the user
- create a device (test001)

## 6.2. account dump and load script

\$tools/data/account.py

Another way to load account and associated data is to use \$tools/data/account.py script. That script loads a JSON file that should be in prescribed format. you can use data tool to dump this file from an existing database or you can create this file by hand.

```
$ python tools/data/account.py --load seed.json --server eclipse
```



superadmins can only be created by hand.

## 7. push data points

\$<pysdk>tools/api/points.py

This script assumes that we already have an account, user and device. We should send data points to the device. The total number of points and the range of channels can be controlled via the script.

```
$ python tools/api/points.py --server eclipse --serial test001 --
database cloudv1108june --total 10
going to pump 10 data points for test001
POST point 1 of 10, response code is 200
POST point 2 of 10, response code is 200
POST point 3 of 10, response code is 200
POST point 4 of 10, response code is 200
```

<sup>2</sup> <mailto:john.doe@yuktix.com>

```
POST point 5 of 10, response code is 200
POST point 6 of 10, response code is 200
POST point 7 of 10, response code is 200
POST point 8 of 10, response code is 200
POST point 9 of 10, response code is 200
POST point 10 of 10, response code is 200
attempted total:10, fails: 0
pump data script run over. report bugs to rjha@yuktix.com
mbair-2:pysdk rjha$
```

## 7.1. what is a data point

A datapoint is a combination of

- serial number
- readings
- xattributes map
- timestamp
  - timestamp is unix timestamp in millis (13 digits)
  - timestamp can also be supplied as \_\_rtc attribute of xattributes map
  - \_\_rtc is in ddmmyy,hhmmss format
  - \_\_rtc takes precedence over timestamp
  - if neither \_\_rtc or timestamp is supplied, server will supply current time as datapoint timestamp.

### Reading.

A readings is an object having

- channel
- channel value
- channel type (float | int | string )