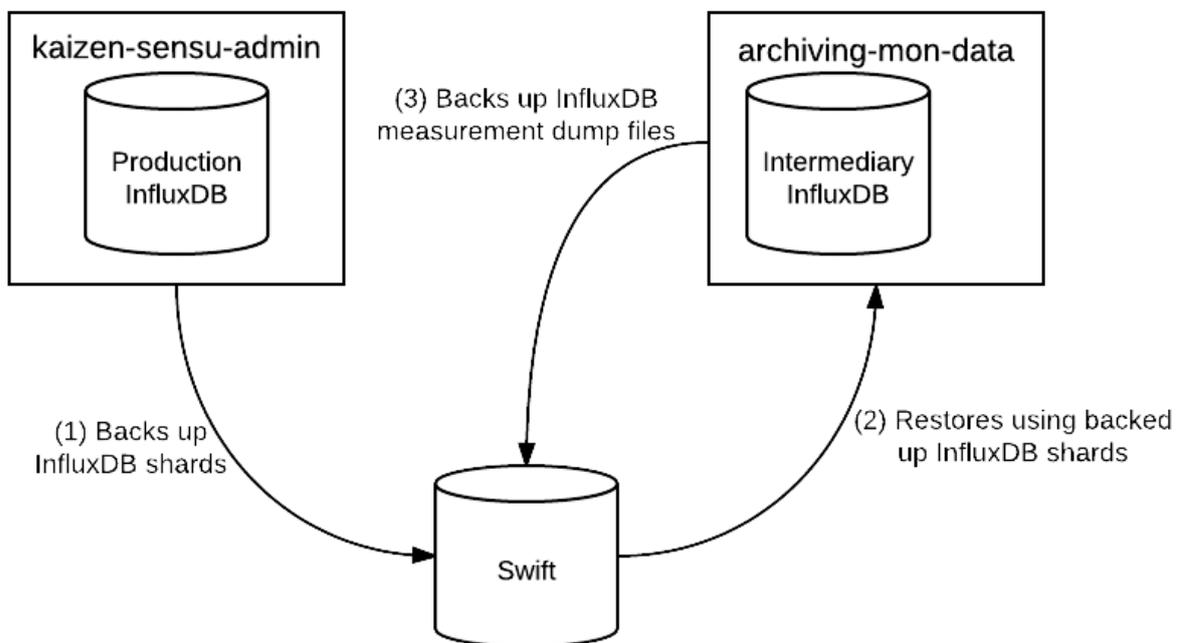


This document will explain how the backup infrastructure works and how to backup and restore InfluxDB. In addition, you will learn how to download JSON dumps of tables (a.k.a measurements) from Swift.

There are three parts to monitoring data backup infrastructure

1. The production InfluxDB database
2. The production Swift Object Storage
3. The intermediary InfluxDB database

Here is a picture of how these three components work together.



First, a cron job starts everyday at 11:55 pm on the kaizen-sensu-admin vm. This backup job executes the following steps in order.

1. Determines the Swift container and object names based on today's date
2. Backs up InfluxDB shards
3. Creates the Swift container if it doesn't exist already
4. Uploads the InfluxDB shards
5. Verifies that the InfluxDB shards have been uploaded successfully
5. Deletes local copy of the InfluxDB shards that were created before uploading

Now that today's InfluxDB shards are uploaded to Swift, the other two cron jobs can start. The

other two cron jobs use the backed up InfluxDB shards.

The second cron job starts everyday at 12:15 am (20 minutes after the first backup job) on the archiving-monitoring-data vm. This restore job executes the following steps in order.

1. Drops the Sensu database if it exists.
2. Determines the container to download yesterday's InfluxDB shards, which is yesterday's date
3. Downloads yesterday's InfluxDB shards from Swift
4. Restores InfluxDB using yesterday's shards
5. Restarts InfluxDB
6. Checks to see if the Sensu database exists
7. Create the directory for the InfluxDB measurement dump files if doesn't already exist

Now that the intermediary InfluxDB is now restored. We can dump InfluxDB measurements.

The next cron jobs can start at any time after the restore job has finished. The cron job needs to be passed the measurement name in order to create a dump file of it. At the moment the `Aggregated_metrics` measurement is the only measurement that has it's dump files being backed up. Currently the job that does this starts at 12:30 am. This job executes the following steps in order:

1. Formats the name of the dump file
2. Repeatedly, writes all of the records for the measurement to the dump file in 1000 record increments
3. Uploads the dump file to the correct container, if the dump file isn't empty

Provided that all of the different cron jobs discussed have executed successfully, let's take a look at what's inside one these containers.

Each container is named based on the date it was *backed up*. This document will use the `2016-06-11` container as an example, which means it was backed up on 2016-06-11. The naming convention for the containers is YYYY:MM:DD.

There are two directories in this container, which are called `shards` and `influxdb`. The `shards` directory contains all of the InfluxDB shards that were backed up on 2016-06-11. The `influxdb` directory contains all of the JSON dump files for the measurements. Here is an example of the object name for one of these dump file:

```
2016-06-11/influxdb/Aggregated_metrics_2016-06-11-00:00:00_2016-06-12-00:00:00.json
```

`2016-06-11` is the name of the container `influxdb` is the name of a directory in the container

Aggregated\_metrics\_2016-06-11-00:00:00\_2016-06-12-00:00:00.json is the JSON dump file name.

The JSON dump filename starts with Aggregated\_metrics, which means it only contains data from the Aggregated\_metrics measurement. The JSON dump filename has 2016-06-11-00:00:00\_2016-06-12-00:00:00, which means it only has data that was created from 2016-06-11 at midnight to 2016-06-12 at midnight.

## Backup

If you want the back up our current InfluxDB, you can run the following command on the kaizen-sensu-admin VM:

```
influxd backup -database sensu_db /tmp/backup
```

This tells InfluxDB that you want to backup the Sensu database to the /tmp/backup directory.

If the backup is working, the output should look like this:

```
2016/06/13 15:59:30 backing up db=sensu_db since 0001-01-01 00:00:00 +0000 UTC
2016/06/13 15:59:30 backing up metastore to /tmp/backup/meta.00
2016/06/13 15:59:30 backing up db=sensu_db rp=default shard=5 to
/tmp/backup/sensu_db.default.00005.00 since 0001-01-01 00:00:00 +0000 UTC
2016/06/13 15:59:30 backing up db=sensu_db rp=default shard=2 to
/tmp/backup/sensu_db.default.00002.00 since 0001-01-01 00:00:00 +0000 UTC
2016/06/13 15:59:30 backing up db=sensu_db rp=default shard=11 to
/tmp/backup/sensu_db.default.00011.00 since 0001-01-01 00:00:00 +0000 UTC
...
```

## Restore

If the production InfluxDB loses all of its data for some reason, here are the steps to restore it from a backup.

First, you may want to check if the most recent shards you want to download are in Swift. To do this, run the following command:

```
swift list 2016-06-11 -p 2016-06-11/shards/
```

2016-06-11 is the container name and 2016-06-11/shards/ is prefix that all of the shards from 2016-06-11 will have. By running this command, you are listing all of the shards that were backed up on 2016-06-11.

If the shards are in swift, you should see the following output:

```
2016-06-11/shards/meta.00
2016-06-11/shards/sensu_db.default.00002.00
2016-06-11/shards/sensu_db.default.00005.00
2016-06-11/shards/sensu_db.default.00011.00
...
2016-06-11/shards/sensu_db.four_weeks.01866.00
2016-06-11/shards/sensu_db.four_weeks.01872.00
2016-06-11/shards/sensu_db.four_weeks.01883.00
```

Now that you know the shards you want are in Swift, you can download them. To do this, run the following command.

```
swift download 2016-06-11 -p 2016-06-11/shards/
```

Again, `2016-06-11` is the container name and `2016-06-11/shards/` is prefix that all of the shards from `2016-06-11` will have. By running this command, you are downloading all of the shards that were backed up on `2016-06-11`.

Now that you have the shards from Swift, you can restore InfluxDB by running the following command:

```
influxd restore -metadir /var/lib/influxdb/meta -database sensu_db -datadir /var/lib/influxdb/data 2016-06-11/shards
```

This tells influxdb to use the backed up shards from the `2016-06-11/shards` directory and unpack them so the metadata is stored in the `/var/lib/influxdb/meta` directory and the data is stored at `/var/lib/influxdb/data` directory. The metadata and data directories must match the directories InfluxDB is currently using, which are configured in `/etc/influxdb/influxdb.conf`. The metadata and data directories use the `dir` variable under the `[meta]` and `[data]` subheadings respectively.

If the restore is working the output should look like this.

```
Using metastore snapshot: 2016-06-11/shards/meta.00
Restoring from backup /tmp/backup/sensu_db.*
unpacking /var/lib/influxdb/data/sensu_db/default/2/000000001-000000001.tsm
unpacking /var/lib/influxdb/data/sensu_db/default/5/000000001-000000001.tsm
unpacking /var/lib/influxdb/data/sensu_db/default/11/000000001-000000001.tsm
unpacking /var/lib/influxdb/data/sensu_db/default/19/000000001-000000001.tsm
unpacking /var/lib/influxdb/data/sensu_db/default/27/000000001-000000001.tsm
...
```

If you restart the InfluxDB instance, then InfluxDB will only use the shards that have been unpacked to the metadata and data directories. To restart InfluxDB run the following command:

```
sudo systemctl restart influxdb
```

## Downloading the dump files

If you want to do some analysis with some of the data from InfluxDB, you can download all of the dump files for a certain date by running the following command:

```
swift download 2016-06-11 -p 2016-06-11/influxdb/
```

All of the dump files will be located in `2016-06-11/influxdb/`

If you want to download all of the dump files for a number of days, you can write a couple of lines of bash to accomplish this like so:

```
mkdir dump_files; for i in $(swift list | grep 2016-06); do swift download $i -p $i/influxdb; mv $i/influxdb/* dump_files; done
```

This creates the `dump_files` directory and gets all of the dump files for June 2016 and puts them all in a directory called `dump_files`

If you want to adjust the span when you want to dump files from you can modify the `2016-06` in the `swift list | grep 2016-06` to make sure you're getting all of the containers, you want the dump files from. You can use regular expressions to isolate the containers you want dump files from.