# facebookincubator/beringei:

Beringei is a high performance, in-memory storage engine for time series data.

https://github.com/facebookincubator/beringei

# Beringei

A high performance, in memory time series storage engine

[](https://github.com/facebookincubator/beringei/blob/master/beringei_logo_clear.png)

In the fall of 2015, we published the [paper “Gorilla: A Fast, Scalable, In-Memory Time Series Database”](http://www.vldb.org/pvldb/vol8/p1816-teller.pdf) at VLDB 2015. Beringei is the open source representation of the ideas presented in this paper.

Beringei is a high performance time series storage engine. Time series are commonly used as a representation of statistics, gauges, and counters for monitoring performance and health of a system.

Features

Beringei has the following features:

* Support for very fast, in-memory storage, backed by disk for persistence. Queries to the storage engine are always served out of memory for extremely fast query performance, but backed to disk so the process can be restarted or migrated with very little down time and no data loss.
* Extremely efficient streaming compression algorithm. Our streaming compression algorithm is able to compress real world time series data by over 90%. The delta of delta compression algorithm used by Beringei is also fast - we see that a single machine is able to compress more than 1.5 million datapoints/second.
* Reference sharded service implementation, including a client implementation.
* Reference http service implementation that enables direct Grafana integration.

How can I use Beringei?

Beringei can be used in one of two ways.

1. We have created a simple, sharded service, and reference client implementation, that can store and serve time series query requests.
2. You can use Beringei as an embedded library to handle the low-level details of efficiently storing time series data. Using Beringei in this way is similar to [RocksDB](https://rocksdb.org/) - the Beringei library can be the high performance storage system underlying your performance monitoring solution.

Requirements

Beringei is tested and working on:

* Ubuntu 16.10

We also depend on these open source projects:

* [fbthrift](https://github.com/facebook/fbthrift)
* [folly](https://github.com/facebook/folly)
* [wangle](https://github.com/facebook/wangle)
* [proxygen](https://github.com/facebook/proxygen)
* [gtest](https://github.com/google/googletest)
* [gflags](https://github.com/gflags/gflags)

Building Beringei

Our instructions are for Ubuntu 16.10 - but you will probably be able to modify the install scripts and directions to work with other linux distros.

* Run sudo ./setup\_ubuntu.sh.
* Build beringei.

mkdir build && cd build && cmake .. && make

* Generate a beringei configuration file.

./beringei/tools/beringei\_configuration\_generator --host\_names $(hostname) --file\_path /tmp/beringei.json

* Start beringei.

./beringei/service/beringei\_main \

-beringei\_configuration\_path /tmp/beringei.json \

-create\_directories \

-sleep\_between\_bucket\_finalization\_secs 60 \

-allowed\_timestamp\_behind 300 \

-bucket\_size 600 \

-buckets $((86400/600)) \

-logtostderr \

-v=2

* Send data.

while [[ 1 ]]; do

./beringei/tools/beringei\_put \

-beringei\_configuration\_path /tmp/beringei.json \

testkey ${RANDOM} \

-logtostderr -v 3

sleep 30

done

* Read the data back.

./beringei/tools/beringei\_get \

-beringei\_configuration\_path /tmp/beringei.json \

testkey \

-logtostderr -v 3

License

Beringei is BSD-licensed. We also provide an additional patent grant.