

Bigtable

[Google Cloud Platform](#) provides hosting of Google's Bigtable database, the original inspiration of HBase and many NoSQL storage systems. Because HBase is so similar to Bigtable, running OpenTSDB 2.3 and later with Google's backend is simple. Indeed, the schemas (see [HBase Schema](#)) are exactly the same so all you have to do is create your Bigtable instance, create your TSDB tables using the Bigtable HBase shell, and fire up the TSDs.

Note

The clients for Bigtable are in beta and undergoing a number of changes. Performance should improve as we adjust the code and uncover new tuning parameters. Please help us out on the mailing list or by modifying the code in GitHub.

Setup

1. Setup your Google Cloud Platform account.
2. Follow the steps in [Creating a Cloud Bigtable Cluster](#).
3. Follow the steps in [HBase Shell Quickstart](#), paying attention to where you download your JSON key file.
4. Set the `HBASE_HOME` environment variable to your Bigtable shell directory, make sure the `HBASE_CLASSPATH`, `JAVA_HOME`, and `GOOGLE_APPLICATION_CREDENTIALS` environment variables have been set according to the values in [Creating a Cloud Bigtable Cluster](#) document, and run the `src/create_table.sh` script. If the script fails to launch the shell, try running the shell manually and execute the `create` statements substituting the proper values.
5. Build TSDB by executing `sh build-bigtable.sh` (or if you prefer Maven, `sh build-bigtable.sh pom.xml`)
6. Prepare the `opentsdb.conf` file with the required and/or optional configuration parameters below.
7. Run the TSD via `build/tsdb tsd --config=<path>/opentsdb.conf`

Configuration

The following is a table of required and optional parameters to run OpenTSDB with Bigtable. These are in addition to the standard TSD configuration parameters from [Configuration](#).

Property	Type	Required	Description	Default
<code>google.bigtable.project.id</code>	String	Required	The project ID hosting your Bigtable cluster.	
<code>google.bigtable.cluster.name</code>	String	Required	The cluster ID you assigned to your Bigtable cluster at creation.	
<code>google.bigtable.zone.name</code>	String	Required	The zone where your Bigtable cluster is operating; chosen at creation.	
<code>hbase.client.connection.impl</code>	String	Required	The class that will be used to implement the HBase API AsyncBigtable will use as a shim between the Bigtable client and OpenTSDB. Set this to <code>com.google.cloud.bigtable.hbase1_0.BigtableConnection</code>	
<code>google.bigtable.auth.service.account.enable</code>	Boolean	Required	Whether or not to use a Google cloud service account to connect. Set this to <code>true</code>	false
<code>google.bigtable.auth.json.keyfile</code>	String	Required	The full path to the JSON formatted key file associated with the service account you want to use for Bigtable access. Download this from your cloud console.	
<code>google.bigtable.grpc.channel.count</code>	Integer	Optional	The number of sockets opened to the Bigtable API for handling RPCs. For higher throughput consider increasing the channel count.	4

Note

Google's Bigtable client communicates with their servers over HTTP2 with TLS using ALPN. As Java 7 and 8 (dunno about 9) lack native ALPN support, a [library](#) must be loaded at JVM start to modify the JVM's bytecode. The build script for OpenTSDB will attempt to detect your JDK version and download the proper version of ALPN but if you have a custom JVM or something other than Hotspot or OpenJDK you may run into issues. Try different versions of the `alpn-boot` JAR to see what works for you.