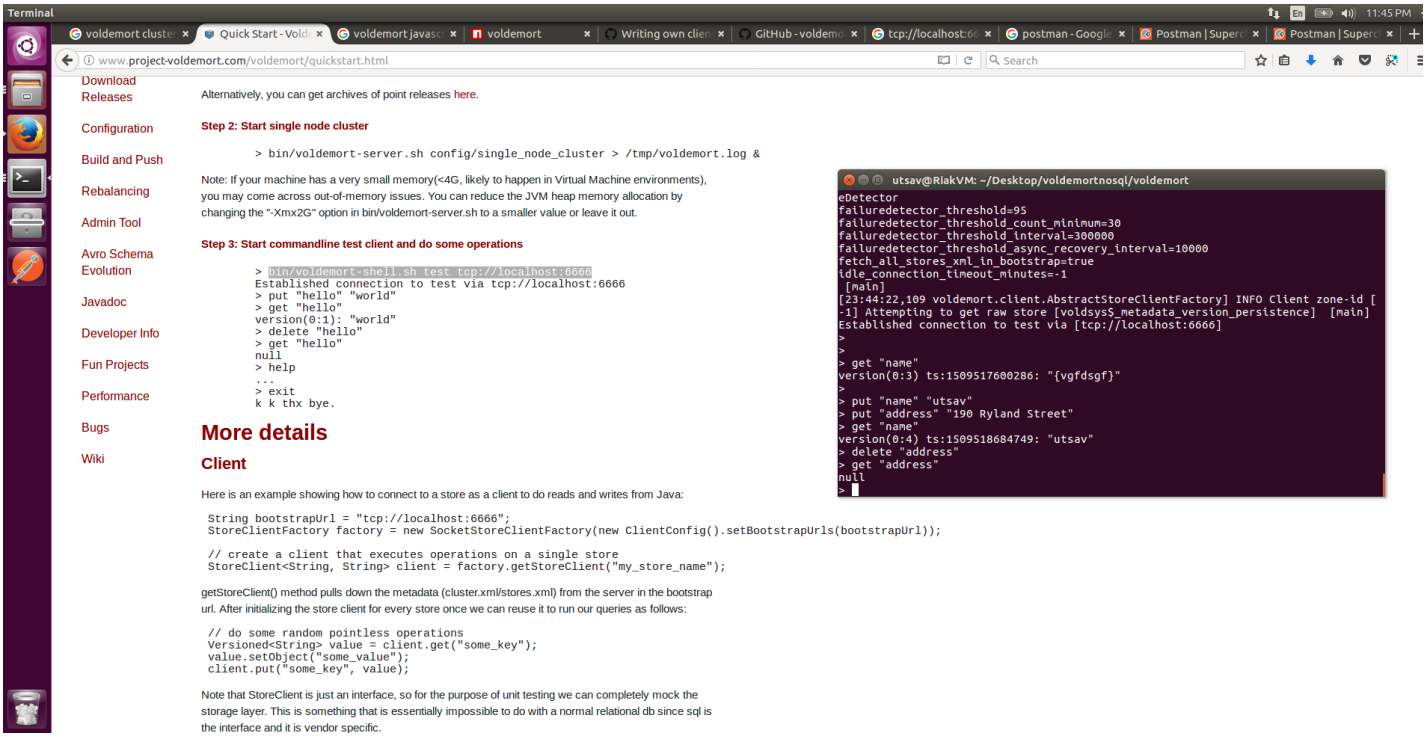
Terminal
11:45 PM
www.project-voldemort.com/voldemort/quickstart.html
Search
Download
Releases
Configuration
Build and Push
Rebalancing
Admin Tool
Avro Schema
Evolution
Javadoc
Developer Info
Fun Projects
Performance
Bugs
Wiki
Alternatively, you can get archives of point releases here.
Step 2: Start single node cluster
> bin/voldemort-server.sh config/single_node_cluster > /tmp/voldemort.log &
Note: If your machine has a very small memory(<4G, likely to happen in Virtual Machine environments), you may come across out-of-memory issues. You can reduce the JVM heap memory allocation by changing the "-Xmx2G" option in bin/voldemort-server.sh to a smaller value or leave it out.
Step 3: Start commandline test client and do some operations
> bin/voldemort-shell.sh test tcp://localhost:6666
Established connection to test via tcp://localhost:6666
> put "hello" "world"
> get "hello"
version(0:1): "world"
> delete "hello"
> get "hello"
null
> help
...
> exit
k k thx bye.
utsav@RiakVM: ~/Desktop/voldemortnosql/voldemort
eDetector
failuredetector_threshold=95
failuredetector_threshold_count_minimum=30
failuredetector_threshold_interval=300000
failuredetector_threshold_async_recovery_interval=10000
fetch_all_stores_xml_in_bootstrap=true
idle_connection_timeout_minutes=-1
[main]
[23:44:22,109 voldemort.client.AbstractStoreClientFactory] INFO Client zone-id [-1] Attempting to get raw store [voldsys$_metadata_version_persistence] [main]
Established connection to test via [tcp://localhost:6666]
>
>
> get "name"
version(0:3) ts:1509517600286: "{vgfdsgf}"
>
> put "name" "utsav"
> put "address" "190 Ryland Street"
> get "name"
version(0:4) ts:1509518684749: "utsav"
> delete "address"
> get "address"
null
>
More details
Client
Here is an example showing how to connect to a store as a client to do reads and writes from Java:
String bootstrapUrl = "tcp://localhost:6666";
StoreClientFactory factory = new SocketStoreClientFactory(new ClientConfig().setBootstrapUrls(bootstrapUrl));
// create a client that executes operations on a single store
StoreClient<String, String> client = factory.getStoreClient("my_store_name");
getStoreClient() method pulls down the metadata (cluster.xml/stores.xml) from the server in the bootstrap url. After initializing the store client for every store once we can reuse it to run our queries as follows:
// do some random pointless operations
Versioned<String> value = client.get("some_key");
value.setObject("some_value");
client.put("some_key", value);
Note that StoreClient is just an interface, so for the purpose of unit testing we can completely mock the storage layer. This is something that is essentially impossible to do with a normal relational db since sql is the interface and it is vendor specific.