**Voldemort Installation:**

**Pre-Requirements:** 1) Linux

2) Java  
 3) Ant

4) Lot of memory ( default 2G)

**Installation Steps:**

1. Update the packages using yum
   1. Yum install updates
2. Using apt-get
   1. Apt-get install updates

3)Get the Voldemort packages from github.

$ git clone git://github.com/voldemort/voldemort.git

1. Install java
   1. Yum install java
   2. Apt-get install java-package-name
2. Run the voldemort

bin/voldemort-server.sh config/single\_node\_cluster > /tmp/voldemort.log &

Test the connection:

* 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

Change voldemort to executable mode:

Chmod a+x /etc/init.d/voldemort

Allow http,socket and admin in linux iptables ( if enabled) using command.

Iptables –A input –p –tcp --dport portno –j Accept

Configuration files in Voldemort:

Cluster.xml – Holds information about all nodes. in cluster

E.g,

<cluster><!-- The name is just to help users identify this cluster from the gui -->

<name>mycluster</name>

<zone>

<zone-id>0</zone-id>

<proximity-list>1</proximity-list>

<zone>

<zone>

<zone-id>1</zone-id>

<proximity-list>0</proximity-list>

<zone>

<server>

<!-- The node id is a unique, sequential id beginning with 0 that identifies each server in the cluster-->

<id>0</id>

<host>vldmt1.prod.linkedin.com</host>

<http-port>8081</http-port>

<socket-port>6666</socket-port>

<admin-port>6667</admin-port>

<!-- A list of data partitions assigned to this server --> <partitions>0,1,2,3</partitions>

<zone-id>0</zone-id>

</server>

<server>

<id>1</id>

<host>vldmt2.prod.linkedin.com</host>

<http-port>8081</http-port>

<socket-port>6666</socket-port>

<admin-port>6667</admin-port>

<partitions>4,5,6,7</partitions>

<zone-id>1</zone-id>

</server>

</cluster>

Stores.xml – Holds Information about all stores in cluster

E.g,

<stores>

<store>

<name>test</name>

<replication-factor>2</replication-factor>

<preferred-reads>2</preferred-reads>

<required-reads>1</required-reads>

<preferred-writes>2</preferred-writes>

<required-writes>1</required-writes> <persistence>bdb</persistence>

<routing>client</routing>

<routing-strategy>consistent-routing</routing-strategy>

<key-serializer>

<type>string</type>

<schema-info>utf8</schema-info>

</key-serializer>

<value-serializer>

<type>json</type>

<schema-info version="1">[{"id":"int32", "name":"string"}]</schema-info>

<compression>

<type>gzip<type> </compression> </value-serializer> </store> </stores>

Server.Properties – This contains the tuning parameters that control a particular node. This includes the id of the local node so it knows which entry in cluster.xml corresponds to itself, also the threadpool size, as well as any configuration needed for the local persistence engine such as BDB or mysql. This file is different on each node.