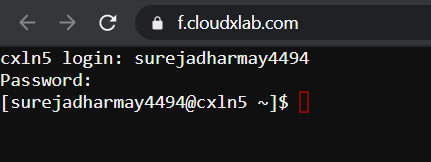
Practical-11

**Perform ZOOKEEPER Command Line Interface ( CLI ) in CLoudXLab .**

ZooKeeper Command Line Interface (CLI) is used to interact with the ZooKeeper ensemble which lets you perform simple, file-like operations. It is useful for debugging purposes. To perform ZooKeeper CLI operations, first start your ZooKeeper server and Client

**STEPS to run the Zookeeper on webconsole(in cloudxlab)**

* First logged in webconsole using user name and password.



* Type the following command

**zookeeper-client**

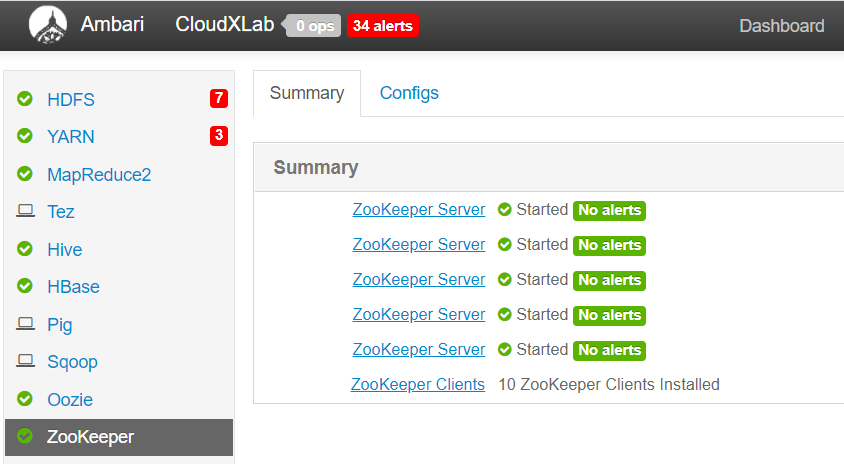
****

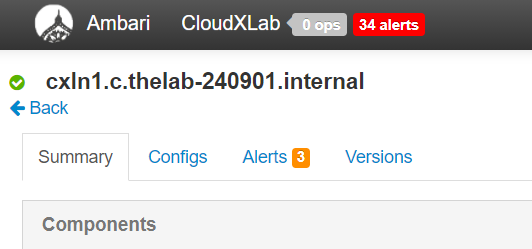
* If you get any error then run the following command:

1. First, you need to open the Ambari in cloudxlab then go to zookeeper

and click on **“zookeeper Server under summery”** and copy

**“cxln3.c.thelab-240901.internal”**





1. Run the following command on Webconsole.

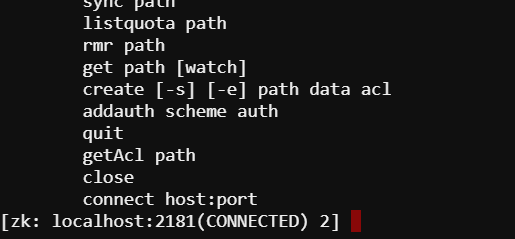
**zookeeper-client-server** **cxln1.c.thelab-240901.internal**



This would bring the zookeeper prompt . this is where we are going to type

Zookeeper commands.

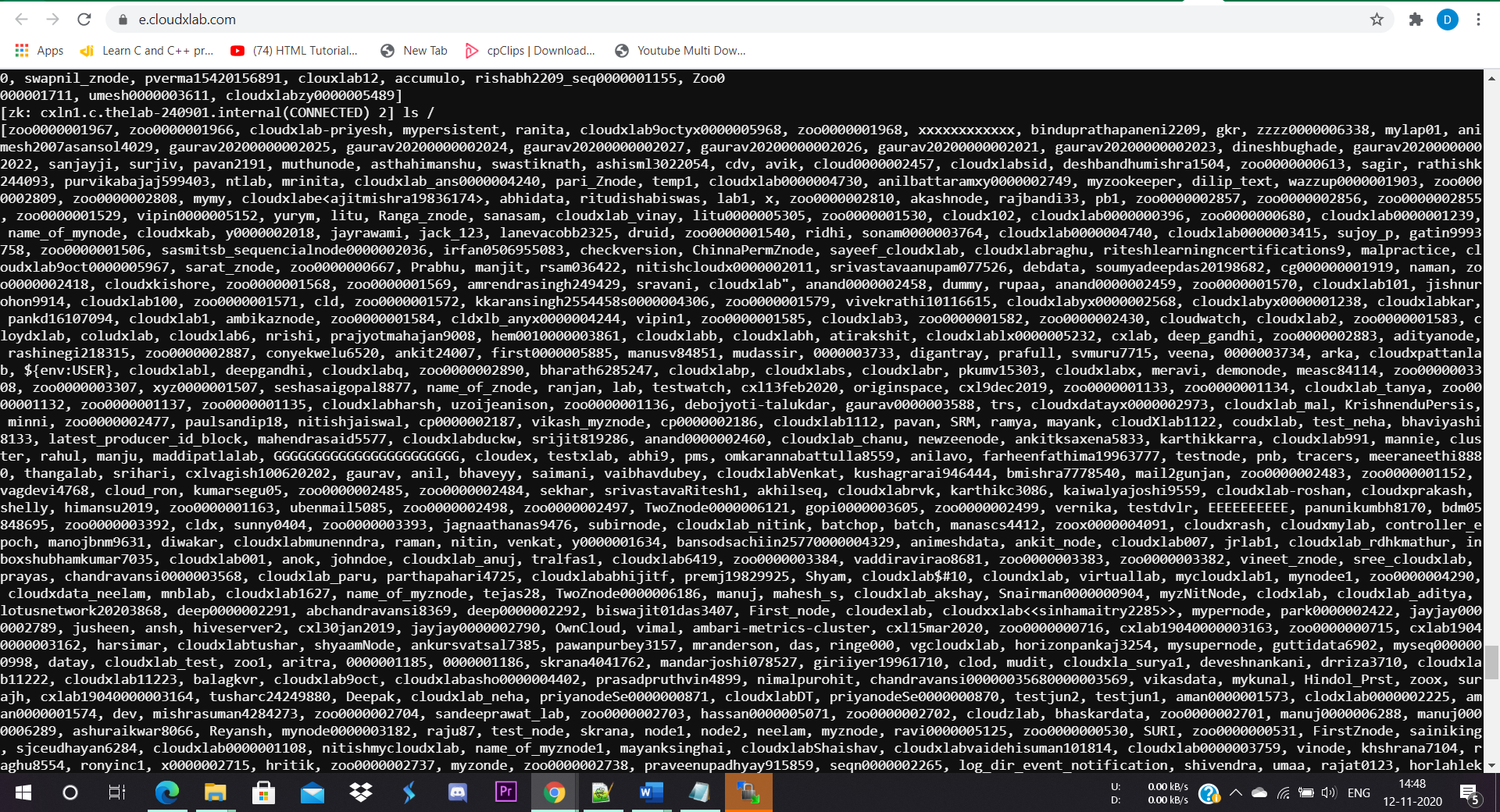
**Output:**



**Perform following Commands**

1. See the list of ZNodes at the top level

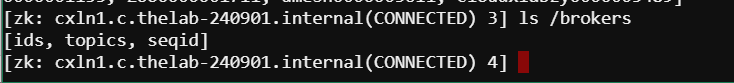
**ls /**



1. See the children of ZNodes under a node called “**brokers**” using the command .

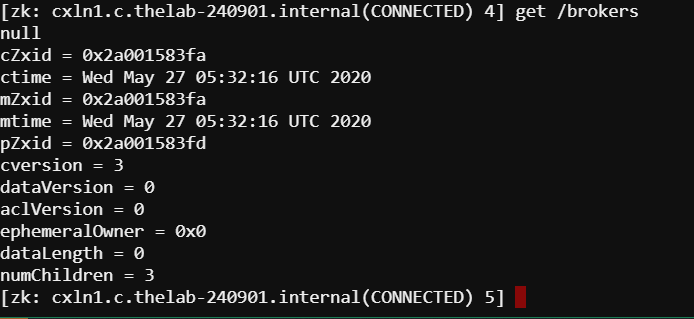
( You can find the name of znode from the above command )

**ls /brokers**



1. See the data inside ZNode “**brokers**”

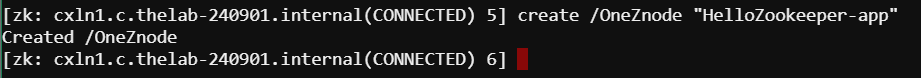
**get /brokers**



1. Create ZNodes

**Syntax : create /path /data**

**Create /OneZnode “HelloZookeeper-app”**

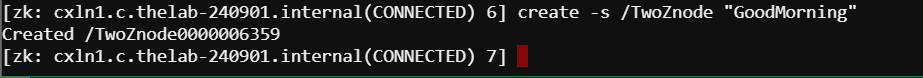


1. create a Sequential znode, add -s flag

**Syntax:** **create -s /path /data**

**create -s /TwoZnode “GoodMorning”**

**NOTE:** Sequential znodes guaranty that the znode path will be unique. ZooKeeper ensemble will add sequence number along with 10 digit padding to the znode path. For example, the znode path /myapp will be converted to /myapp0000000001 and the next sequence number will be /myapp0000000002. If no flags are specified, then the znode is considered as persistent.

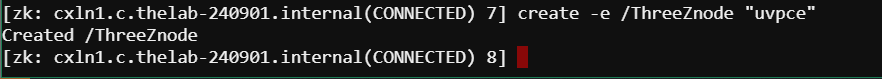


1. Create an Ephemeral Znode, add -e flag as shown below.

**Syntax:** **create -e /path /data**

**create -e /ThreeZnode “uvpce”**

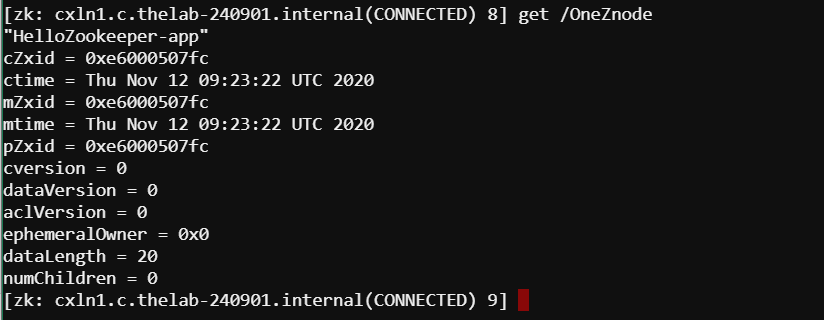
**NOTE:** Ephemeral znodes (flag: e) will be automatically deleted when a session expires or when the client disconnects. Remember when a client connection is lost, the ephemeral znode will be deleted. You can try it by quitting the ZooKeeper Client and then re-opening the Client.



1. Get Data

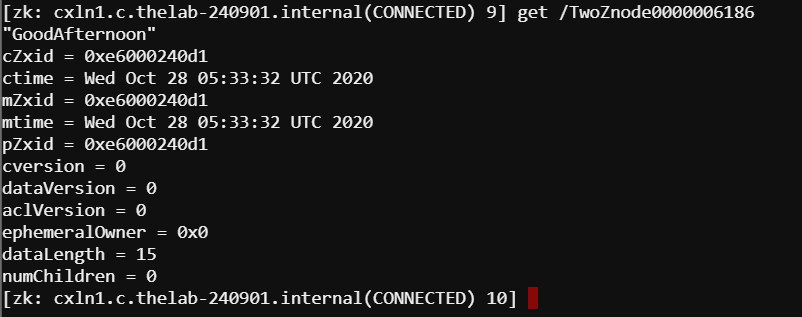
**Syntax: get /path**

**get** **/OneZnode**



**get /TwoZnode0000000023** (To access a sequential znode, you must enter the full path of the znode.)

**Output:**



**NOTE:** It returns the associated data of the znode and metadata of the specified znode. You will get information such as when the data was last modified, where it was modified, and information about the data. This CLI is also used to assign watches to show notification about the data.

**NOTE:** Below show the output of “get” command and description.

**my\_data :**This line of text is the data that we stored in the znode.

**cZxid = 0x8 :**The zxid (ZooKeeper Transaction Id) of the change that caused this znode to be created.

**ctime = Mon Nov 30 18:41:06 IST 2015 :**The time when this znode was created.

**mZxid = 0x8 :**The zxid of the change that last modified this znode.

**mtime = Mon Nov 30 18:41:06 IST 2015 :**The time when this znode was last modified.

**pZxid = 0x8 :**The zxid of the change that last modified children of this znode.

**cversion = 0 :**The number of changes to the children of this znode.

**dataVersion = 0 :**The number of changes to the data of this znode.

**aclVersion = 0 :**The number of changes to the ACL of this znode.

**ephemeralOwner = 0x0:** The session id of the owner of this znode if the znode is an ephemeral node. If it is not an ephemeral node, it will be zero.

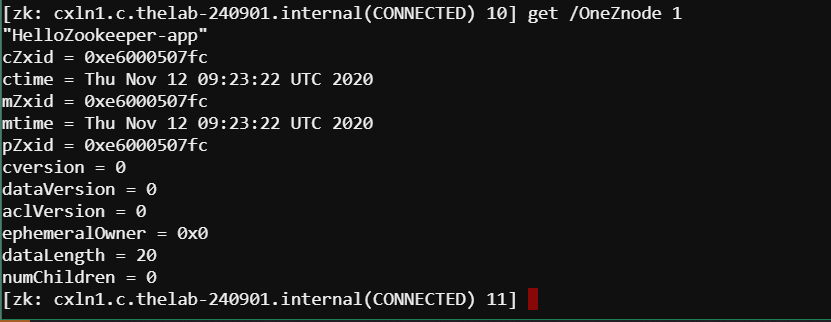
**dataLength = 7 :**The length of the data field of this znode.

**numChildren = 0 :**The number of children of this znode

1. Watch

**Syntax : get /path [watch] 1**

**Get /OneZnode1**



**Note:** Watches show a notification when the specified znode or znode’s children data changes. You

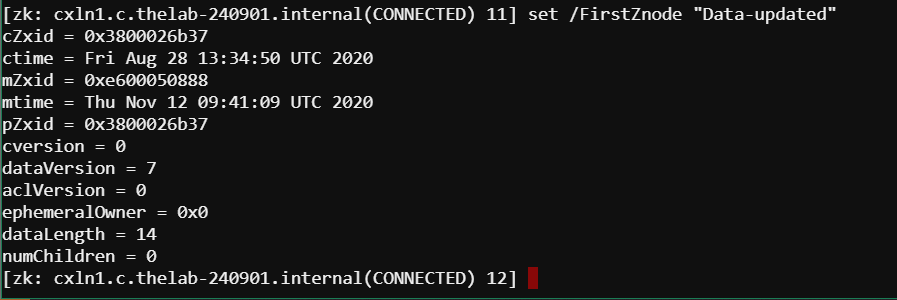
can set a watch only in get command. The output is similar to normal get command, but it will

wait for znode changes in the background.

1. Set Data

**Syntax: set /path /data**

**set /FirstZnode “Data-updated”**

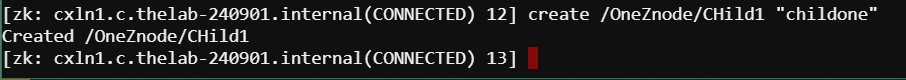


**NOTE:** Set the data of the specified znode. Once you finish this set operation, you can check the data using the get CLI command.

1. Create Children / Sub-znode

**Syntax: create /parent/path/subnode/path /data**

**create /OneZnode/Child1 “childone”**

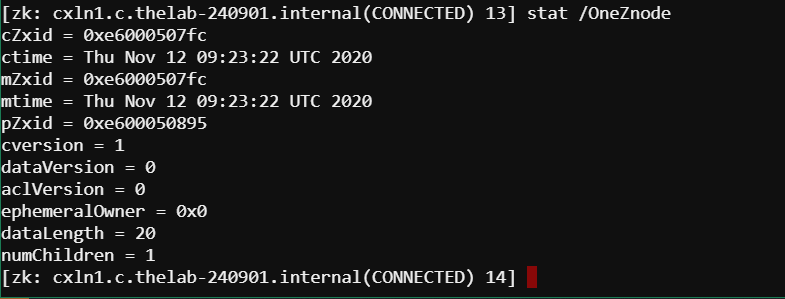


**Note:** Creating children is similar to creating new nodes. The only difference is that the path of the child znode will have the parent path as well.

1. Check Status

**Syntax: stat /path**

**stat /OneZnode**

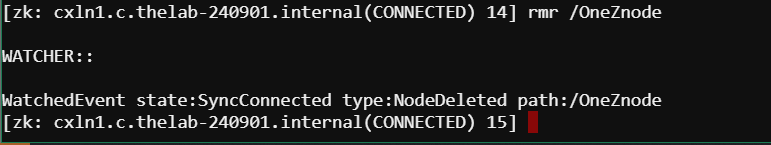


**Note:** Status describes the metadata of a specified znode. It contains details such as Timestamp, Version number, ACL, Data length, and Children znode.

1. Remove a Znode

**Syntax:** **rmr /path**

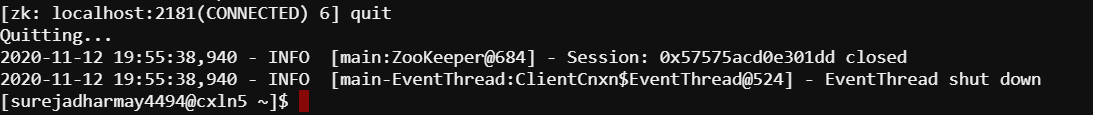
**rmr /OneZnode**



**Note:** Delete (delete /path) command is similar to remove command, except the fact that it works only on znodes with no children.

1. **Quit**

**quit**

****