**To make SONARQUBE up**

sh sonar.sh start/stop -> to start and stop

sudo netstat –nlput ->to check processed and server

sudo kill -9 <process id> ->to kill the process

**To make DevOps Enabler Platform up**

cd /home/automation/tomcat\_webapplications/devops/apache-tomcat-7.0.57/bin -> go to this path

sudo nohup bash startup.sh & ->to up DevOps

sudo netstat -nlput | grep 8088 ->to check for DevOps that is it up or not

**To check memory consumption by process id use any one of the following command**

* ps -o rss,sz,vsz <process id>
* ps -p <process id> -o %mem

**To check RAM and SWAP memory**

* free –m ->to check in mb
* free –g ->to check in gb

**To check hard disk memory**

* df –h path ->path can be any path (eg. /home or /home/automation)

To check processes running on server/port/user

Either server or port else user at a time.

* ps –ef | grep server/port/user (eg. tomcat/8088/ma834862)

**For firewall removal**

sudo iptables -I INPUT -p tcp -m tcp --dport 8092 -j ACCEPT

**Command To Copy From Remote to local**

sudo scp [ma834862@3.209.30.155:/home/ma834862/predixmachinesdk-16.2.0.tar.gz](mailto:ma834862@3.209.30.155:/home/ma834862/predixmachinesdk-16.2.0.tar.gz) /predix/Desktop

here 3.209.30.155 is remote

/predix/Desktop is local

Jenkins @155 path:- /home/automation/jenkins/jenkins\_production

To start command :- sudo nohup bash jenkins-8080.sh &

public void updateAnalytics(String url) {

String strUrl = url;

String strAction = "GET";

HttpUriRequest request = getHttpRequest(strUrl, strAction);

int proxy\_port = 8080;

HttpClientBuilder hcBuilder = HttpClients.custom();

HttpHost proxy = new HttpHost("proxy-src.research.ge.com", proxy\_port,"http");

DefaultProxyRoutePlanner routePlanner = new DefaultProxyRoutePlanner(proxy);

hcBuilder.setRoutePlanner(routePlanner);

CloseableHttpClient client = hcBuilder.build();

HttpResponse response;

try {

response = client.execute(request);

if((response.getStatusLine().getStatusCode()==200) || (response.getStatusLine().getStatusCode()==204)|| (response.getStatusLine().getStatusCode()==206)){

if((response.getEntity()!=null)&&(response.getEntity().getContent()!=null)){

BufferedReader rd = new BufferedReader(

new InputStreamReader(response.getEntity().getContent()));

System.out.println("Response "+response.getEntity().getContent());

StringBuffer result = new StringBuffer();

String line = "";

while ((line = rd.readLine()) != null) {

result.append(line);

}

if(request.getMethod().equals("GET"))

{

JSONObject jsonObj = new JSONObject(result.toString());

System.out.println(jsonObj.toString());

}

}

}

} catch (ClientProtocolException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

private HttpUriRequest getHttpRequest(String strUrl, String strAction){

HttpUriRequest request = null;

String action = strAction.toString();

if(strAction.equalsIgnoreCase("GET")){

request = new HttpGet(strUrl);

}

return request;

}