11-ElastiSearch and MongoDB and bash scripting basic brushup.

12-Python script to index all documents from MongoDB collection, Reading about dynamic and explicit mappings and including mappings (explicit) in above script, reading about Tailing cursors.

13-Python script to index only the updated documents (assuming no deletions and updations to Mongo collection). creating trigger functionality in MongoDB using oplog (Master-Slave).

16,17-As instructed by Hitesh, reading and (implementing) trying out aggregations (Metrics,Buckets and pipelining) and APIs (Indices APIs, Cluster APIs, Query DSL) and filters.

**18-05-2016:** Python scripts for CRUD operations in Elasticsearch. (Index from JSON input, Search by id and by query, Update by id, Delete by id and by query)

**19-05-2016**: Python script to update indexed documents in ES by id, Python script to index only the newly added documents in MongoDB collection(using count and assuming only insertions), Python script to index both newly added and updated documents in Mongo collection, by setting up and monitoring Oplog.

**20-05-2016**: Research about why indexing is much slower in ES version 2.x compared to 1.7 and found a way to make Indexing faster (x30-x40 times) in 2.x. Made a report regarding the same.

**23-05-2016&24-05-2016**: Python scripts to sanitize, score, rename sponsors by reading from database and updating these sanitized, scored, renamed values to the same.

**25-05-2016**: Researched and submitted report about configuring SHIELD for ES and was able to successfully setup the same with different user roles and wrote an improved version of Python script to index using bulk api (this script works 4-5 times faster than the last one).

**26-05-2016**: Researched about clusters in Elasticsearch and successfully tried setting up 4 nodes and forming a cluster with them, researching about how master is selected etc and prepared a report regarding the same.

**27-05-2016 & 28-05-2016**: Wrote a login page in PHP (MySQL as database) and implemented two factor authentication using Gmail servers with a Python script and using sessions in PHP.

**29-05-2016 to 01-06-2016**: Wrote Python scripts to monitor any number of connected servers from a master computer and shut down any server continuously not being used for predefined amount of time and power it on anytime a user wants to connect to it by SSH or a when a user pings it (Tested this on 3 computers with one of them being master and 2 being servers).

Wrote Python scripts to monitor error log (of ngnix server) and mail to system admin (Suyash) whenever an error level log appears with log file attached.

Wrote Python scripts to monitor data usage and uptimes of servers and mail it to system admin.

**02-06-2016:** Perfected the Python script to monitor data usage of servers, started working on Python script to grammatically correct strings and completed the python script to monitor bandwidth usages of hosts connected to a server and mailing sysadmin if bandwidth usage crosses limit, and did some more research on ES. Researched about logs in Elasticsearch and prepared a report.

**03-06-2016 & 06-06-2016**: Perfected the server controlling scripts written, added some features, cleared out bugs and wrote setup scripts for the same, so that deploying them on server becomes easy. Made some progress on text-correction script. Modified bulk indexing script to add shield authentication. Added adding cronjobs to the setup scripts.

**07-06-2016:** Python scripts (with spell correction, without spell correction but correcting from list of wanted symbols, without spell correction but correcting from list of unwanted symbols) for text-correction completed. (Need some test data to improve the script’s correction capabilities). Setted up a graylog server and configuring it to monitor elasticsearch’c log.

**08-06-2016:** Worked on the text-correction script a bit, Setted up a ES cluster with master and slave nodes on cloud, modified the bulk indexing script to accommodate to the fact that max size of http request is 2gb and started reading about and working on graph databases (neo4j) and mongodb, neo4j and neo4j, ES connections

**09-06-2016:** Setted up neo4j server and did all the basic queries and read about graphdb, used mongo connector to automatically insert the documents inserted into mongodb into neo4j and replicate the same for deletions and updations using oplog. Modified the bulk indexing script to start indexing from the position till which it last time indexed and also to recursively start indexing from the last error point to accomodate ‘read timeout’ errors. Improved upon the text correction script but need data to actually use it

**10-06-2016:** Read some more about elasticsearch, took part in a session between Touchpoint team members about elasticsearch. Researched about connecting mongodb, neo4j, elasticsearch. Started working on pdf crawler.

**13-06-2016:** Completed the Python script to get pdfs and extract data from them and inserting it into mongodb and also included data sanitization into it. Read about graphdbs and elasticsearch integration and researched a bit about graphdbs

**14-06-2016:** Completed Python script to index data to ES from neo4j, read about neo4j, graylog2. Read about Cypher queries, mongo queries. Went through my previous Python scripts and commented, improved some of them.

**15-06-2016**:Made complete reports about data monitor and speed monitor, OTP and Auto Poweroff and wrote improved versions of the same. Read about static mappings in ES, improved upon the bulk indexing script and started researching about sentry and some other log monitoring tools like logstash.

**16-06-2016, 17-06-2016:** Worked on some elasticsearh queries and graylog configuring.

**21-06-2016**: Setted up Graylog server and started working on monitoring elasticsearch logs with graylog.

21-06-2016 - 01-07-2016: Completed the python script to send elasticsearch logs to graylog

Worked on neo4j, debugged Yayati’s mongo to neo4j script and improved it using sets and checking inthe script and eliminating merge statements (made the script to work in linear time) improved it’s performance by almost 330 times, (took down the time from 6 days for 90000 mongodb documents to 1568 seconds for 90000 mongodb documents), worked on various elasticsearch queries