Big data notes:

<https://cuboulder.zoom.us/j/3272586015>

Shreanvan shetty

Keval shah. Software eneginer splack

Manjunath rao

Archana anand

Music maps disturbed systems

Song preview player

MusicMap Demo (Team Disturbed system)

Select song, sunflower

Music player populartiy, detaills, release date

Artist song pakyer

Review

Live Tweets

Tweets cometing from

Back end

API: PredictHD, Spotify, Twitter, **Flask**

Spark + NLTK, Flask

Docker, elasticsearch, cassandra, google cloud

\\\

Big mood

Sentiment based on what's trending in the world

Trending topics by tweet volume

Regions

REST API

MongoDB

Information flow

340 Commits UI API Sentiment analysis

\\\\  
Travel Guide

Neethi shetty

Reshma moramupudi

|  |  |
| --- | --- |
| Priyanka sundaram | Splunk |

Tweets

Instagram

Tripadvisor

World weather online

Opencage data

MongoDB

Spark

Kurbernetes

Express JS

Bootstrap

jQuery

D3,js

Google maps

\\\\\

Get start early

Do the part you like

Focus on too much on backend, should one people in frontend UI

Do not spending too much time on data

Teleform

D3 Experts

Jira board

Pivotal

Kanban

Workfront

Trello

Agile will be tested

Agile Project Lifecycle

Pivotal tracker

• Provide experiential learning using real world tools

○Skills I see missing in graduates….

• Bring in Guest Lecturers that are experts in their domain

• Create a map of technology you can use to solve problems

• Toolset

○Cloud computing (GCP or AWS)

○Git/GitHub (Source Control)

○Pivotal (Agile Project Mgmt)

○Google Apps (Lectures/Presentations)

○Twitter (Announcements)

• Definition of Big Data

○Brief history of data growth and technology advancements

○Datafication of our world, Correlation vs. Causality, Value

• Storage

○Relational vs. NoSQL

○Origins - Big Table (Google) / Dynamo (Amazon)

○Survey of NoSQL stores & how to choose

○Cloud options

• Processing

○Batch vs. Streaming

○Concurrency - going wide

○Open Source frameworks (e.g. Hadoop, Spark, Flnk, etc.)

○Distributed Architecture (Queueing, Search, Web Services, etc.)

• Analytics

○Machine Learning, Statistical Methods,, NLP, etc. (Tensor Flow)

○Performance

• Visualization / Data Science

• Prepare you for the real world

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals.

Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly (retrospectives)

Big Mood: twitter trend sentiment analysis

REST API

CRON Job

AWS S3

Vue.js

D3.js

Travis CI

Pivotal

Kafka: Message Queue

OpenStreetMap: GPS Coordinates

MongoDB

Redis

Distributed Logging: elasticsearch; logstash; kibana

Framework: Vue.js

Wordcloud: D3.js

World map, trends bar chart - fusion charts

City maps - google charts

Deployment - S3 Bucket

Golang, typescript

Angular, React, Node.js, SCSS

Apache Kafka, redis, mongoDB

AWS, Jnenkins, JIRA, Docker, Kubernetes, GCP, Travis

Music maps distributed system

Spotify apu

PredictHQ api

Tweeter api

Flask

Elasticsearch

Cassandra

Google cloud

Apache spark

NLTK python

React

AMCHARTS

Redux

Redux-Saga

Travel guide:

Apache kafka

Kubernetes, node.js

Apache spark

Cloud dataproc

Express.js

Bootstrap

Jquery

D3.JS

C++ when you care about performace

Run cafca in terrform, infrusture

Application level in ansible

Python,Ruby

Collecting data. API

Process

|  |  |  |
| --- | --- | --- |
| Cafca (queue) | Rabbit mq | Pulsar |

P,P,P, S3, Mongodatabase

Map reduce

UI

…..

AWS/GCP server

Cloud server three ways:

AWS, GCP, Azure, digital ocean, IBM, Oracle

GCP: create VM, boot disk ubuntu 20.04,   
command line utilities: vim setup.sh, ls, gcloud compute instances list, ./run.sh gobuffs, uname -a, sudo su - , sudo update,

Automate with code:

Terraform: terraform apply/destroy

Redis

Ruby

PHP

Code - Manage

Optimination for satellite

NP-Hard

database <- meta, scheduler, redis run\_id,

Runner berlin, API will access the redis cache and database

Spark, Tensorflow,

Tuple, attribute, relation

ORM is not used in Tweets, write objects

Object Oriented Database?

Micro service, REST API, same parameter then the customer do not care.

75,000 for twitter each year, Network

1…\* one and many

Cassandra: node cluster, self-healing, one node dead, another node coming out,

Tunable consistency: all node be updated.