Datacenter Infrastructure Management System

Siddhartha Mishra
Divya Spoorthy
Raaghav R.

Problem Statement:

The purpose of this product is to provide a unified interface for visualisation, monitoring and alert management of IITH's data center..

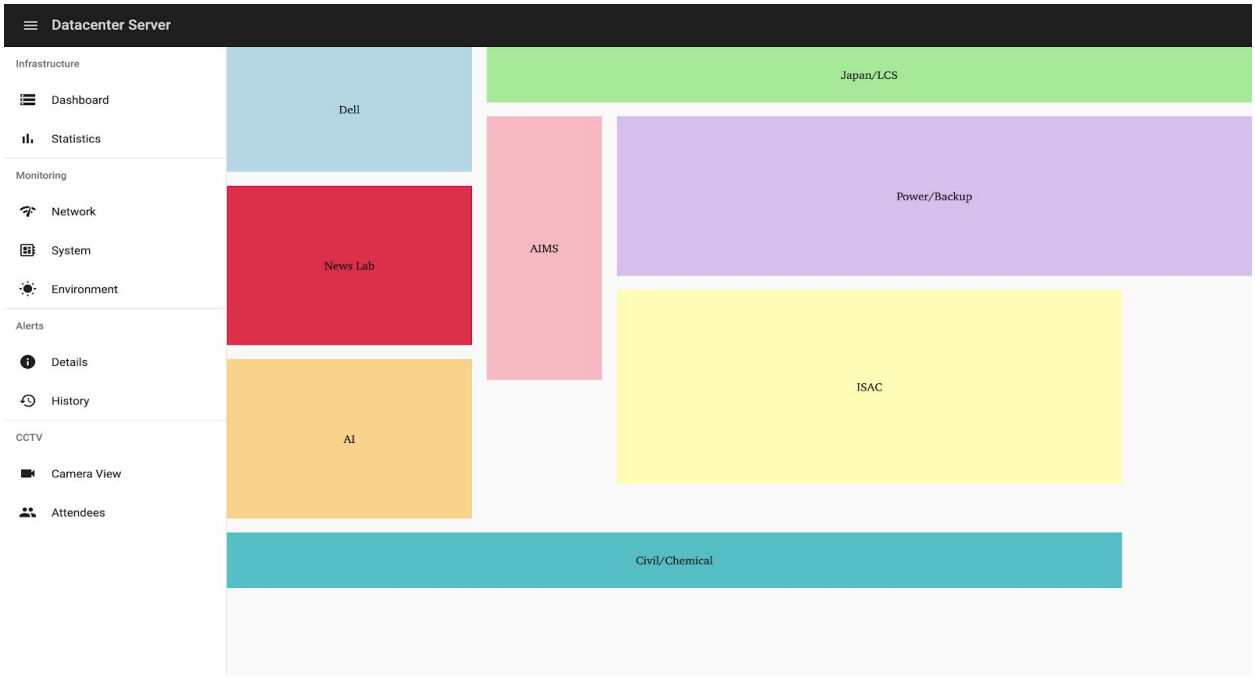
Github Link (Please report in issues section of github if you face any problem and we'll resolve it right away)
Client: https://github.com/Siddhartha1234/DataCenter-client
Server: https://github.com/DivyaSpoorthy/Datacenter-Server

System Design:
The major components of this product include

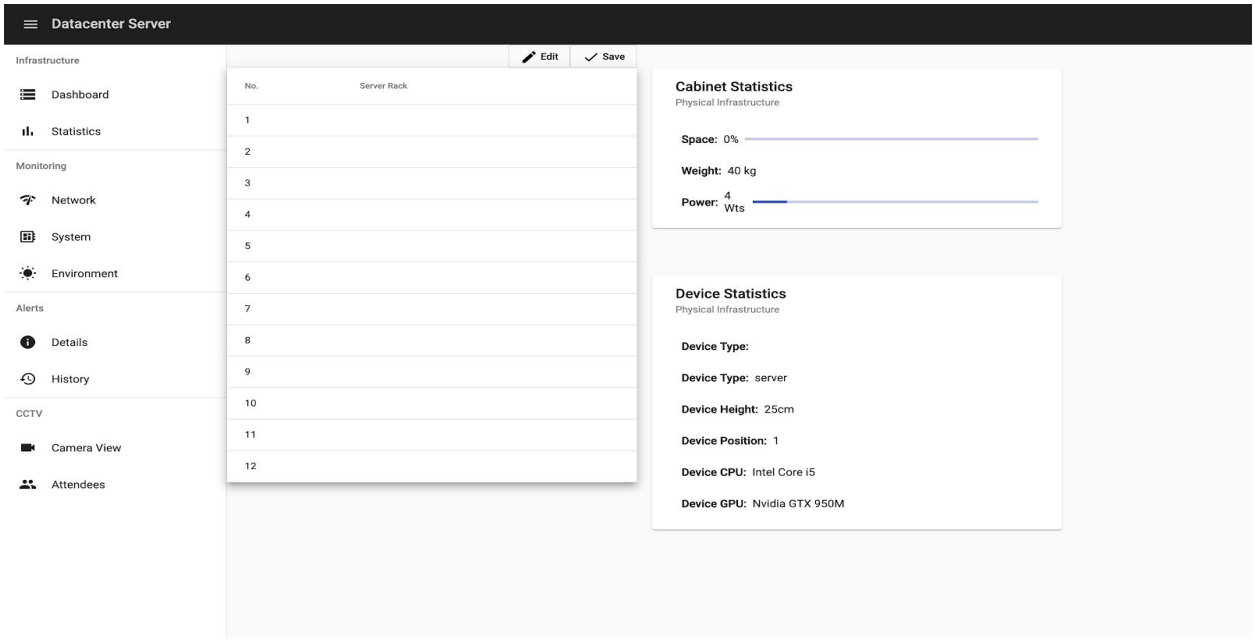
- Client view for monitoring and alert viewing for the staff/students using the data center resources.
- Admin view for the data center staff for global resource and infrastructure management

Components of Server View (Used by data center system admin) :

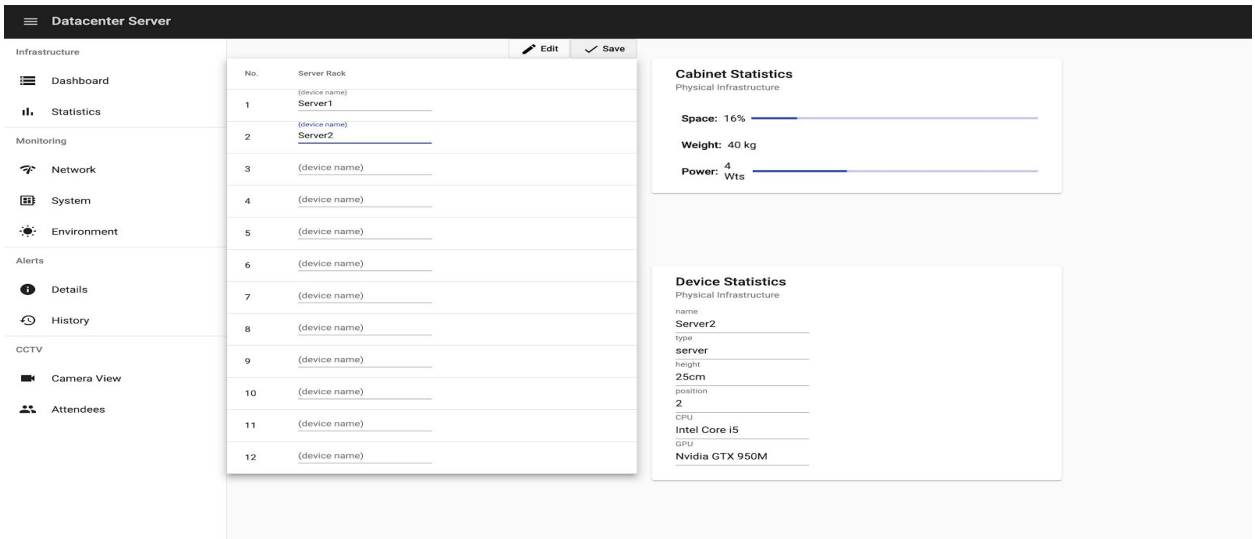
- **Dashboard**



- Dashboard consists of the top view of the server racks, individually placed as the servers are placed in the DataCenter
- **Statistics**

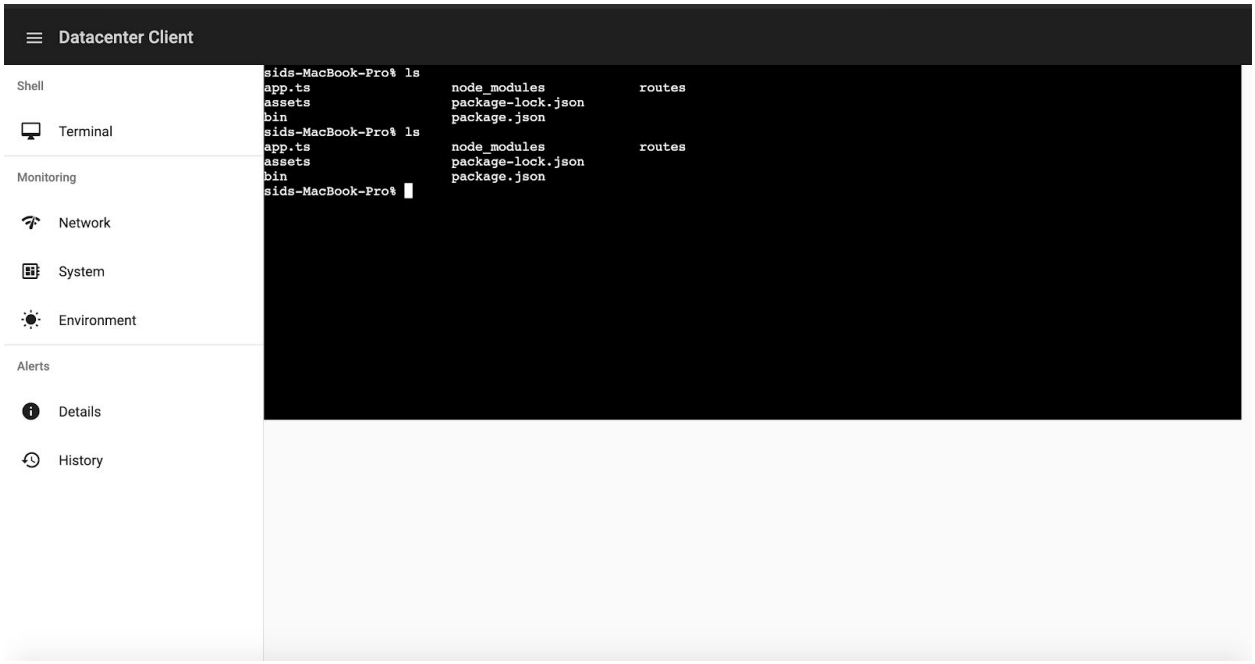


  - Statistics consist of the overall infra management component view. It consists of space for the space racks and displays the space used up, weight and power consumed.
  - It also displays individual rack's statistics.
  - We can use the edit and save option to add/ remove the space racks from the server and the data updates accordingly.

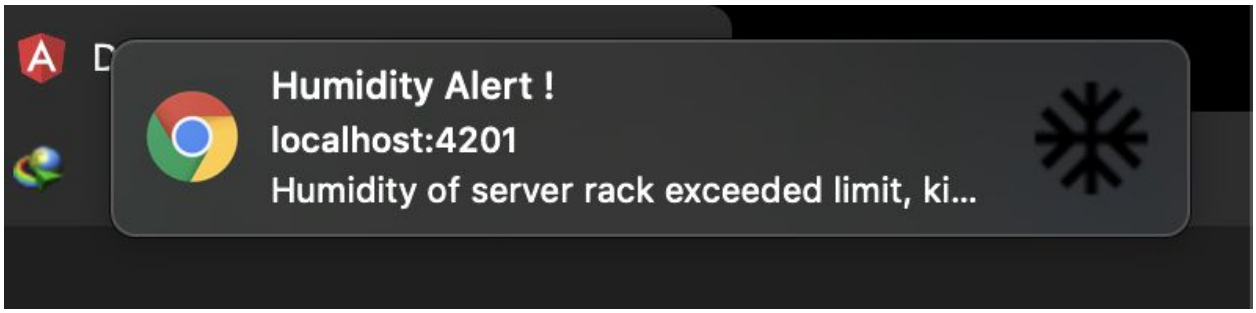


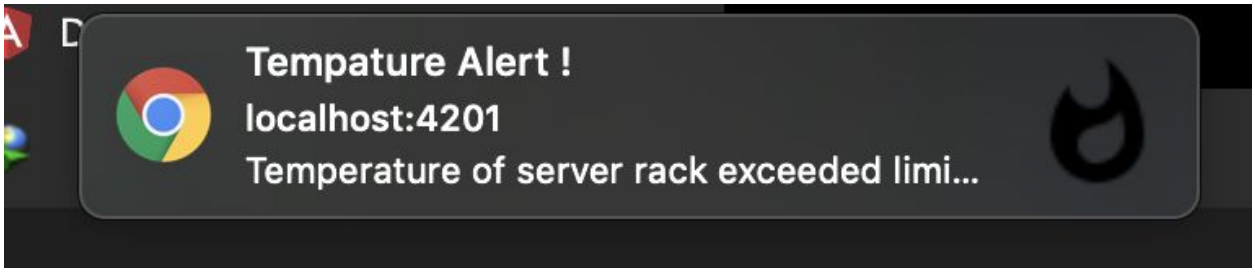
Client view : To be used by professors and data center clients

- **Embedded Terminal:** For usage for cluster connected to data center

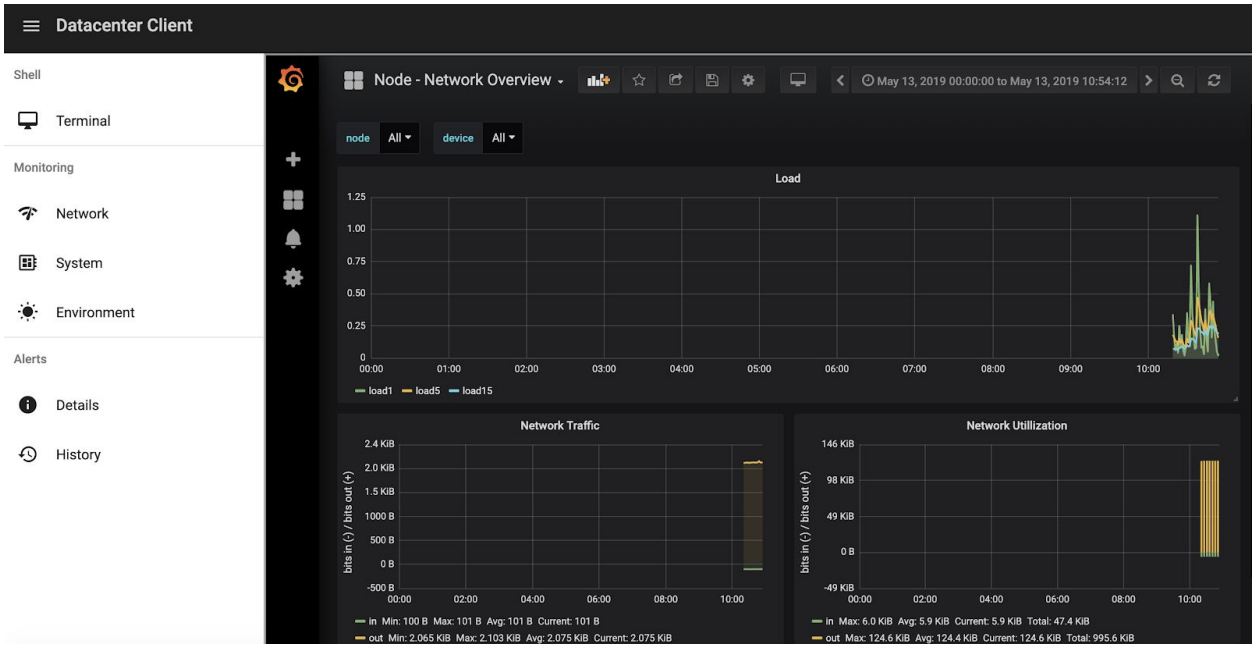


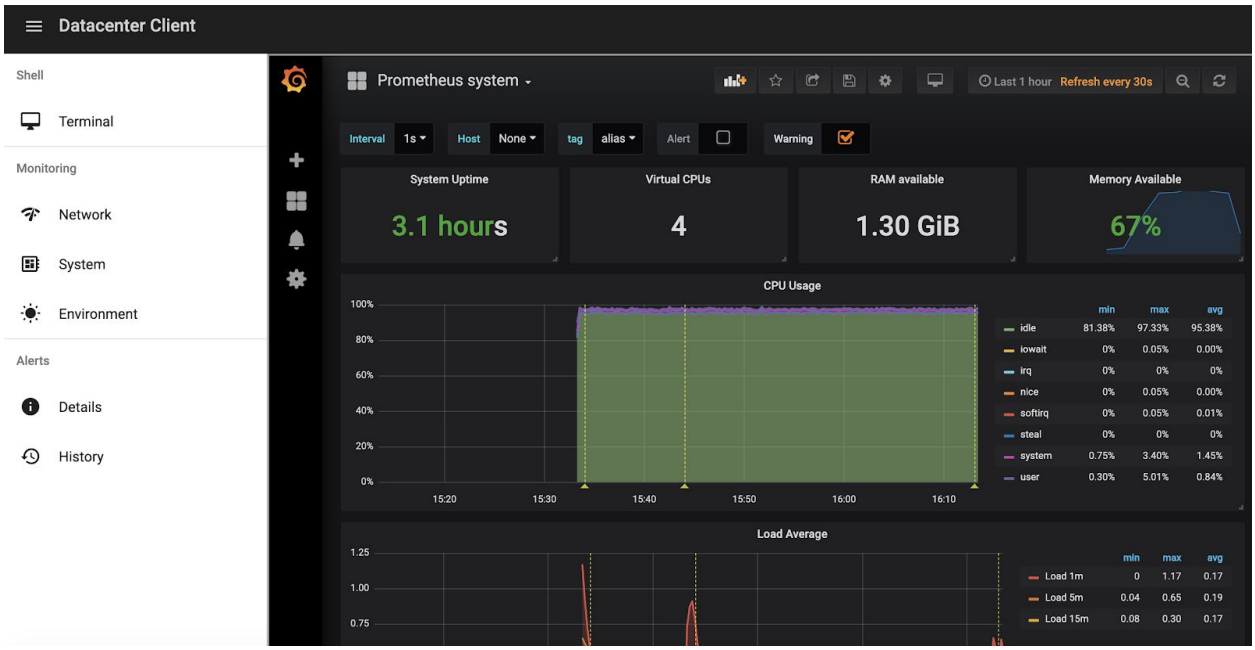
- **Push notifications on alerts can be seen on top end, press on it to take action..**





- **Prometheus / Graphana Monitoring :** Use it for dynamic visualization and customize graphical visualizer as you demand.





Deploy Instructions :

Prerequisite: *Install node/npm in the deploying machine.*

1. Prometheus docker deploy
   - * Enable host docker daemon *
   - cd prometheus-docker
   - docker stack deploy -c docker-stack.yml prom
   - prom_cadvisor, prom_grafana, prom_prometheus, prom_node-exporter, prom_alertmanager services are hence enabled
2. Client app :
   - `Deploy docker using (1)`
   - `npm install -g http-server`
   - `cd $[ROOT_SERVER]`
   - `npm install`
   - `ng build --prod`
   - `http-server -p $[CLIENT_APP_PORT] -c-1 dist/Datacenter-client --proxy http://localhost:$[CLIENT_NODE_PORT]`
   - `cd server`
   - `npm install`
   - `node app.ts (app.listen is on $[CLIENT_NODE_PORT])`
   - `Client accessible at localhost:$[CLIENT_APP_PORT]`

3. Server app :
   - Deploy docker using (1)
   - `npm install -g http-server`
   - `cd $[ROOT_SERVER]`
   - `npm install`
   - `ng build --prod`
   - `http-server -p $[SERVER_APP_PORT] -c-1 dist/Datacenter-client`
   - Server app accessible at localhost:$[SERVER_APP_PORT]