IAS Group Design Document

Technology stack to be used for project development

PREPARED FOR:
IAS Project : IIIT-Hyderabad

PREPARED BY:
Team 1:
Pratik Tiwari
Danish Zargar
Team 2:
Neeraj Barthwal
Abhinav Anand
Team 3:
Tushar Patil
Gaurav Chaudhari
Team 4:
Jay Krishna
Tirth Pandit
Team 5:
Smit Khanpara
Dharmesh Gusai

1. **Language: Python, Java (depending upon the component)**
2. **Servers/VMs: AWS EC2, Virtual box**
   - Platform will intialize new machines as per load, EC2 and virtual box can be used to fullfill resource requirement.
   - Each Algorithm will be run on seperate Virtual Machines, AWS EC2 / Virtual Box will be used for provising VMs.
3. **Database: NoSQL=MongoDB , SQL=Postgres**
   - For storing logs, Client Data, Algorithms can be stored in mentioned database as per requirement
   - Sensor meta data ¡distance,GeoLocation,Sensor Types¿ will be stored as a document in Mongodb.
   - Document will be identified uniquely using Mongodb inbuild ID specific to each sensor.
   - User Data and other Component topics can be stored in Postgres.
4. **Data link: Kafka**
   - Intercommunication among components can be done via kafka, also for sending live sensor data during algorithm execution.
   - Communication between IoT Data Handler and Sensor data can be done via Kafka Topic.
   - Specific Broker will be made for high volume data streaming with consumer offsets for each component.
5. **Automation scripts for VM: Chef/Puppet/Vagrant**
   - Automation scripts will be made for initializing VMs on EC2/Virtual box. Chef/Puppet/Vagrant can help in creating complete scripts for deployment service.
   - Virtual Machince can be configured/packages can be installed using automation scripts written using Chef/Puppet.

- Selecting desired configuration (RAM,HDD,CPU) of virtual machine will be done via predefined templates(Chef/Puppet)
- Containers can be deployed on each machine via dockerfile.

6. **Notification service**
   - To implementmessage queues we will use Kafka/RabbitMQ /AWS SNS
   - Messaging and Email service can be done AWS SNS
   - RabbitMQ will be used to make notification service asynchronous
7. **Logging service: FluentD (Needs more exploration)**
   - Try fluentd for unified logging service across all component
   - Predefined structure can be made for debugging components.
8. **Webservers: Tomcat/Nginx**
   - For deployment of application, also to serve request/reply for user.
   - Dashboard/API will be served via webservers.
9. **Dashboard / User Interface: Bootstrap Framework (for Web based)**