

Eyevery - An Application Monitor

Project Test Plan

CS 387 - Spring 2022

Group Members

Name	Roll Number	Contact
Ankit Kumar Misra	190050020	ankitkmisra@cse.iitb.ac.in
Richeek Das	190260036	richeek@cse.iitb.ac.in
Rahul Prajapat	190050095	rahulprajapat@cse.iitb.ac.in
Raja Gond	190050096	rajagond@cse.iitb.ac.in

Roles & Privileges

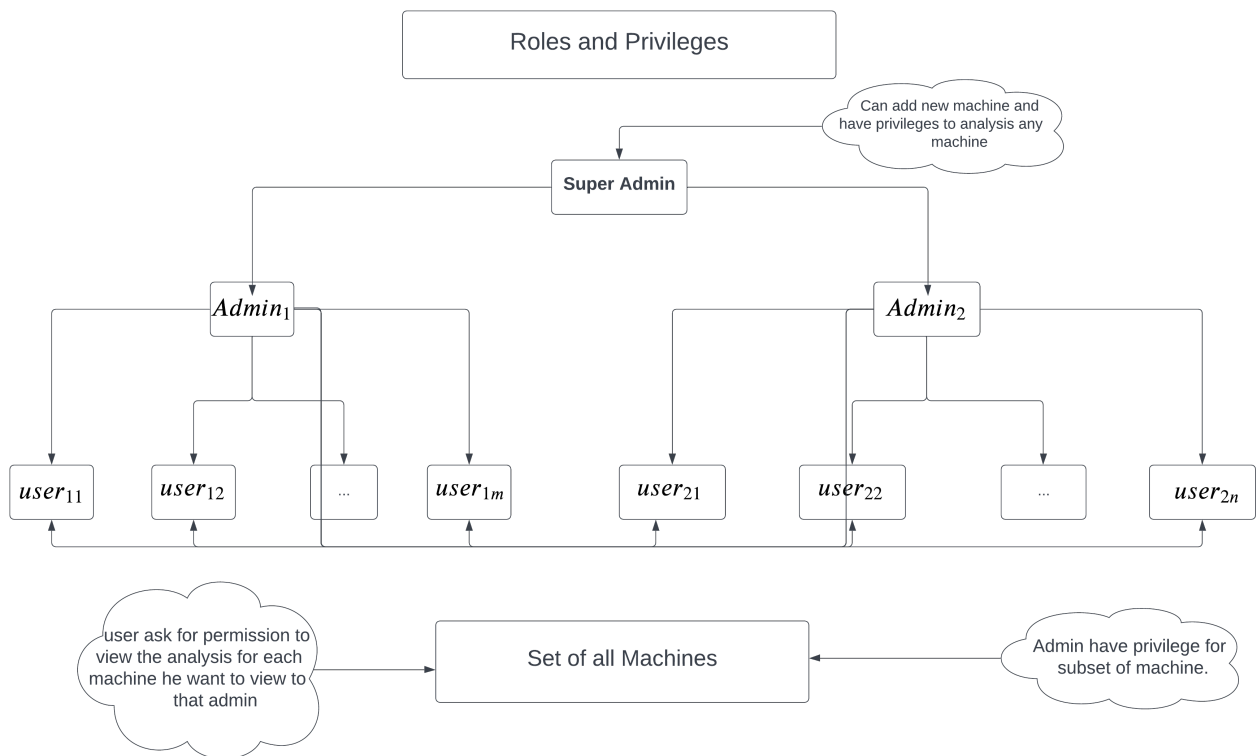


Figure 1: Roles & Privileges

1 Functional Testing

S. No.	Use Case & Requirements	Positive Test	Negative Test
1	Log in to an account (Super Admin, Admin, or User) using the appropriate credentials.	Correct username and password, for each type of account.	Non-existent user or incorrect password, for each type of account.
2	Performing accurate real-time cluster analysis and rendering a dashboard for the data received from machines available to an account.	Dynamic updates of metrics and graphs displayed on the dashboard for each active machine.	The dashboard stops displaying the status of a machine when it stops sending data.
3	Implement failsafes by allowing the user to set appropriate thresholds on various metrics for the machines visible to that user, and sending alerts when thresholds are exceeded for these machines.	Appropriate alerts are displayed without delay when a threshold for some metric gets exceeded at a machine.	No alerts when resource thresholds are respected. Therefore, no negative tests.
4	Predict crashes in the possible future along with the probable time of crash. <i>Example:</i> Host 1 with IP 172.14.5.xx will cross the recommended Memory usage at 11:22PM UTC+05:30.	Simulate data resembling linear growth in resource usage. A naive prediction algorithm should be able to predict the time of crash reasonably well.	No predicted crash times if no future crashes seem probable. This case can be simulated by data with erratic changes but with all values lower than the recommended threshold.
5	Users will have certain privileges to change/finetune the failsafe thresholds for different hard and soft resources	Test that only a logged-in user can change the thresholds. Check on edge cases with simulated data if the failsafe threshold changes are reflected correctly.	User not logged in should not be able to change thresholds. Extreme changes in thresholds will show a warning about the recommended resource threshold.
6	A live query panel, which will provide a low level view of the underlying Influx DB. Users will be able to directly write queries using Flux and get their desired results (with options to export the results into CSV)	Passes some certain test cases and edge cases on a stock DB to ensure correctness of the overall query pipeline	A wrong Flux query displays appropriate error message in the logs and query panel.
7	New Machine added to analysis by Super Admin	Correct permission to integrate new machine with controller, correct configuration of the telegraf of the new machine	Permission not granted to integrate(i.e any random machine should not able to integrate) and any wrong configuration of the telegraf

8	Super Admin can grant viewing and granting privileges to an admin for the machines requested by that admin.	Admin gets privileges to grant viewing to a user.	User cannot ask for privileges from the Super Admin.
9	Admin can grant viewing to a user from the subset of requested machines.	User successfully obtains viewing rights for the requested cluster of machines.	User cannot grant rights to another user, Admin cannot grant privileges of any machine not available to it, and Admin cannot grant rights to another Admin.

2 Load Testing

Simulating Load

- Processes
- DB Connections
- Sockets
- Descriptors
- Threads

As all of these are entities, the more the number the more load per instance. We can have many combinations of varying these loads on the available machines (Like Machine1 having many Processes, while Machine2 having many Sockets, etc.)

Measurement Metrics

- Response Time: For dashboard, the data has timestamps associated with it, so we can compare timestamp and the current time to get response time. This will be our sole factor for judging load.
- Controller Machine Stats (by running telegraf on that machine) such as CPU Usage, RAM Usage, etc.

Many Active Users and Admins

As we are planning to test using only a limited number of machines, we will have limited users, admins. So we won't be doing load testing for that.

Testing Methodology

We are planning to test our application monitor using two methodologies.

Pre-loaded Data

We get data beforehand by running **Telegraf** on different machines and stored it in **InfluxDB**. Since all the data gets loaded into the database beforehand, we would be able to observe the upper bounds of response delays and display update latencies immediately. This is thus useful in load testing. These higher loads will be useful in checking and removing potential crashes and bottlenecks in our application monitor.

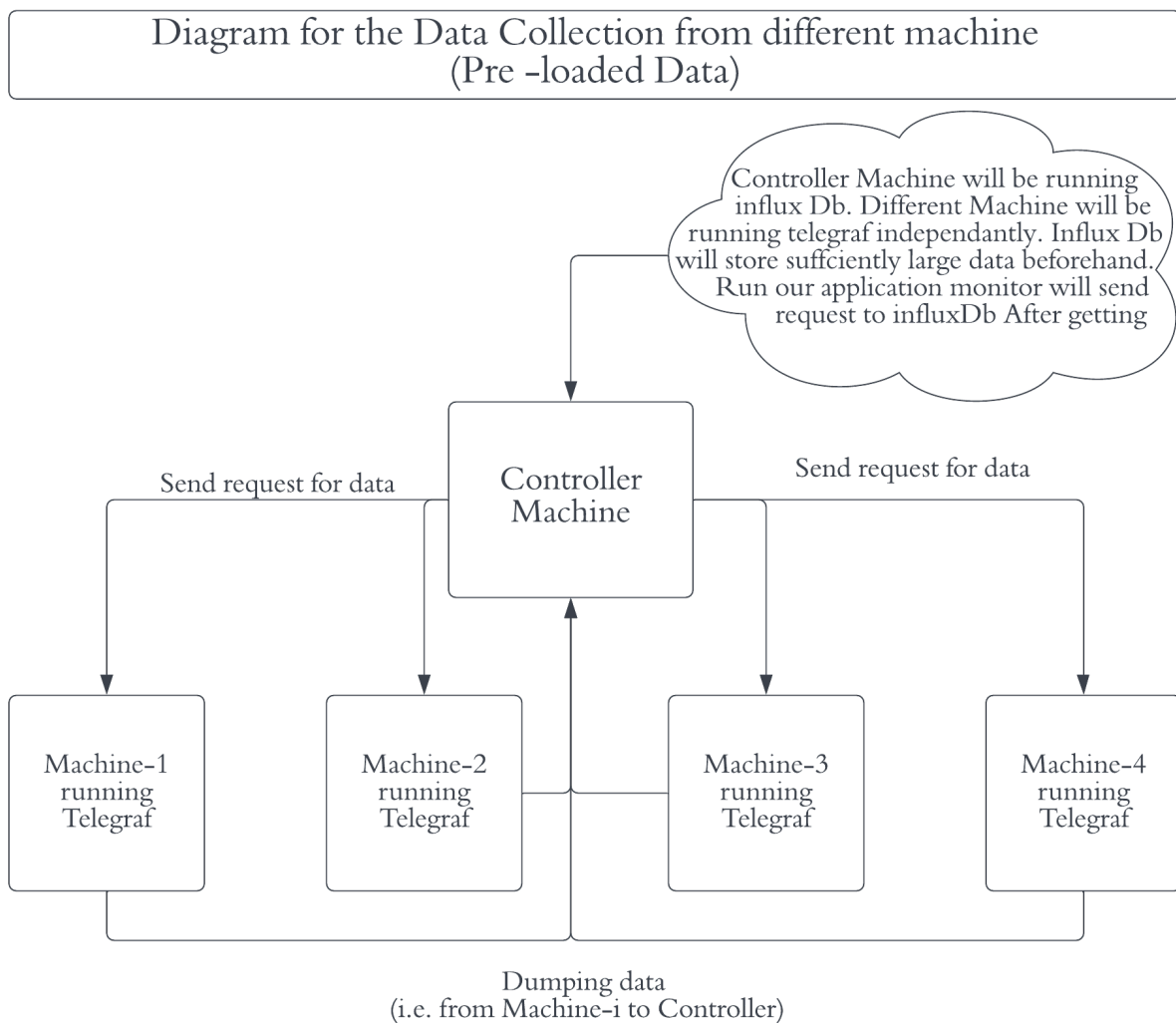


Figure 2: Pre-Loaded Data

Live Insertion Data

Effects are displayed on the dashboard while updates are made to the database simultaneously in real time. As the size of the database increases over time, we should be able to observe an increase in response times for queries, and in general slower updates to the data visualization. We would also measure the delays in generating predictions and alerts.

Diagram for the Data send from different machine
(Live - insertion Data)

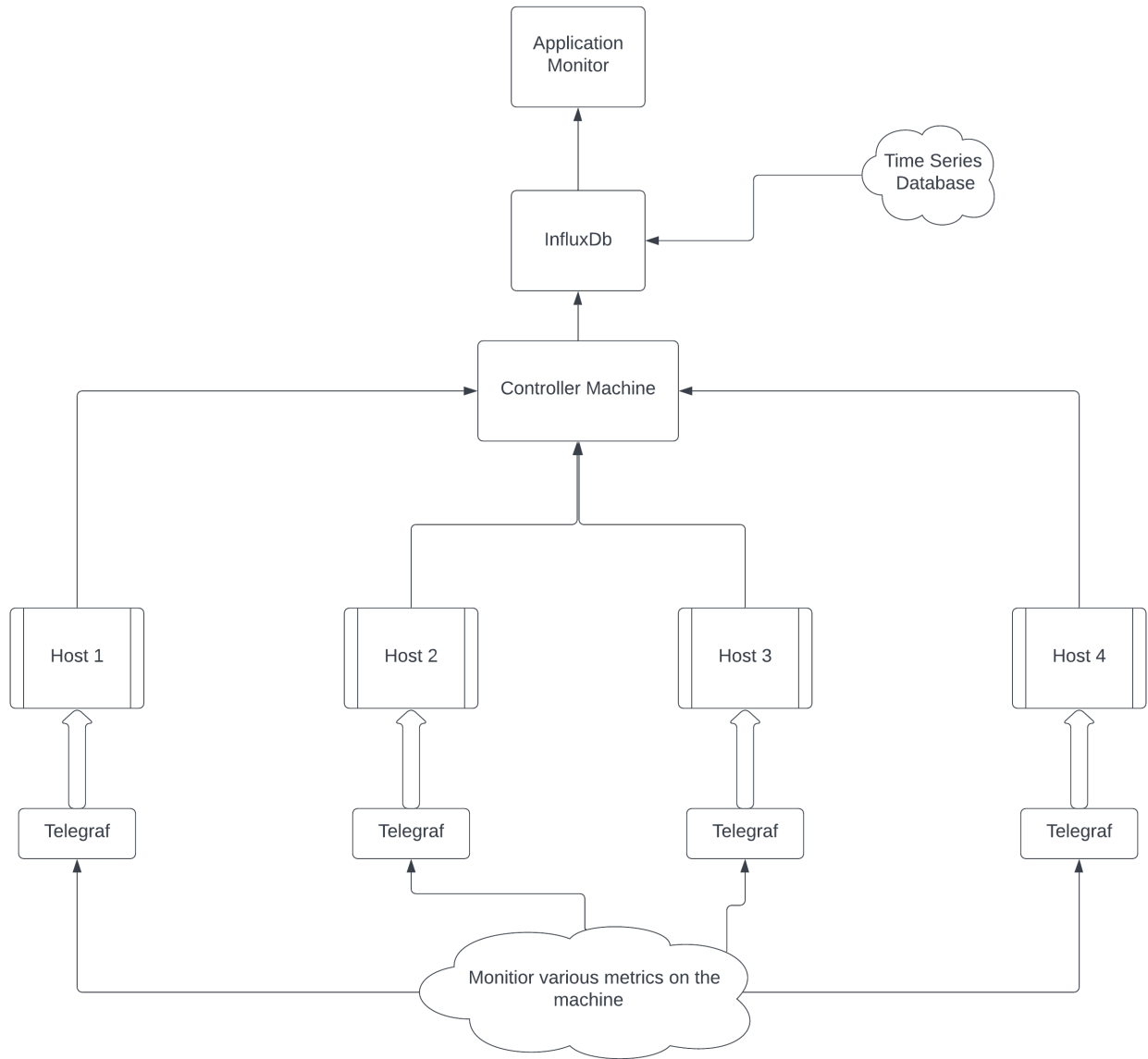


Figure 3: Live-Insertion Data