



# Streamlining Software Release Process and Resource Management for Microservice based Architecture on Multi-Cloud

ReleaseX Solution

23-193

# OUR TEAM



**Dr. Nuwan Kodagoda**  
**Supervisor**



**Mr. Udara Samarakunge**  
**Co-Supervisor**



**Herath H.M.I.P.**  
**IT20125516**



**Jayawardena R.D.S.H.**  
**IT20074968**



**Fadhil M.R.A.**  
**IT20784720**

# Problems



Available software release process systems are not fully automated and optimized.



Cannot predict the necessary resource request and resource limit for VPAs.

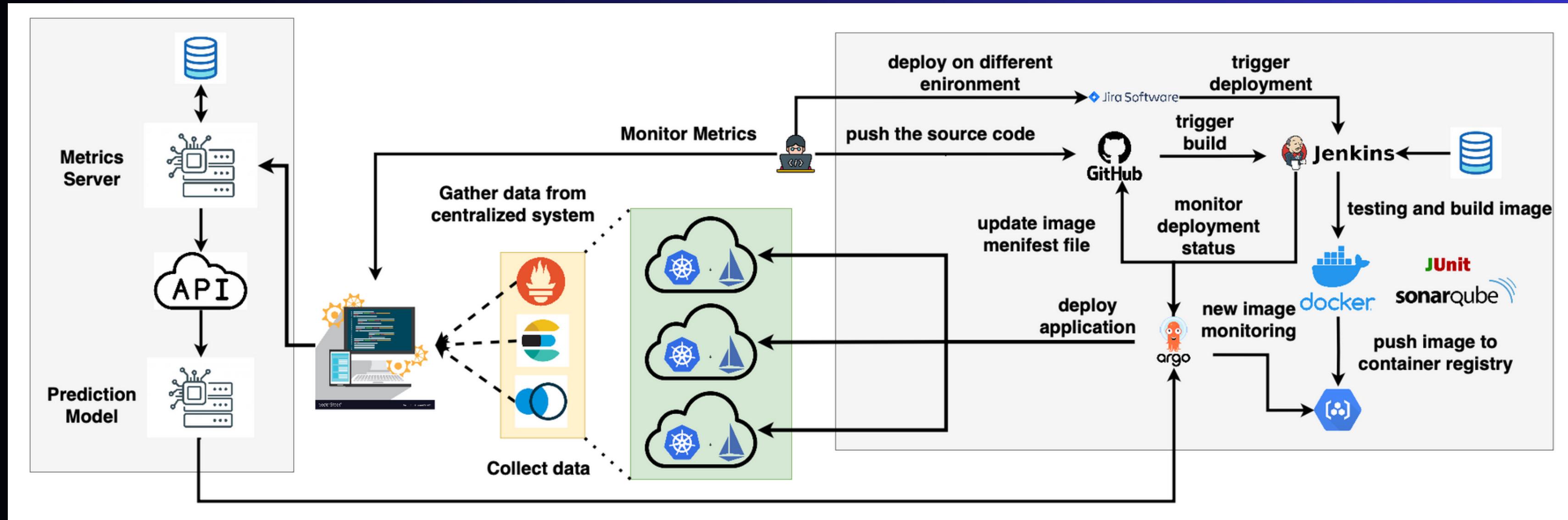


Available technologies are not focused to deploy applications in multiple cloud providers.

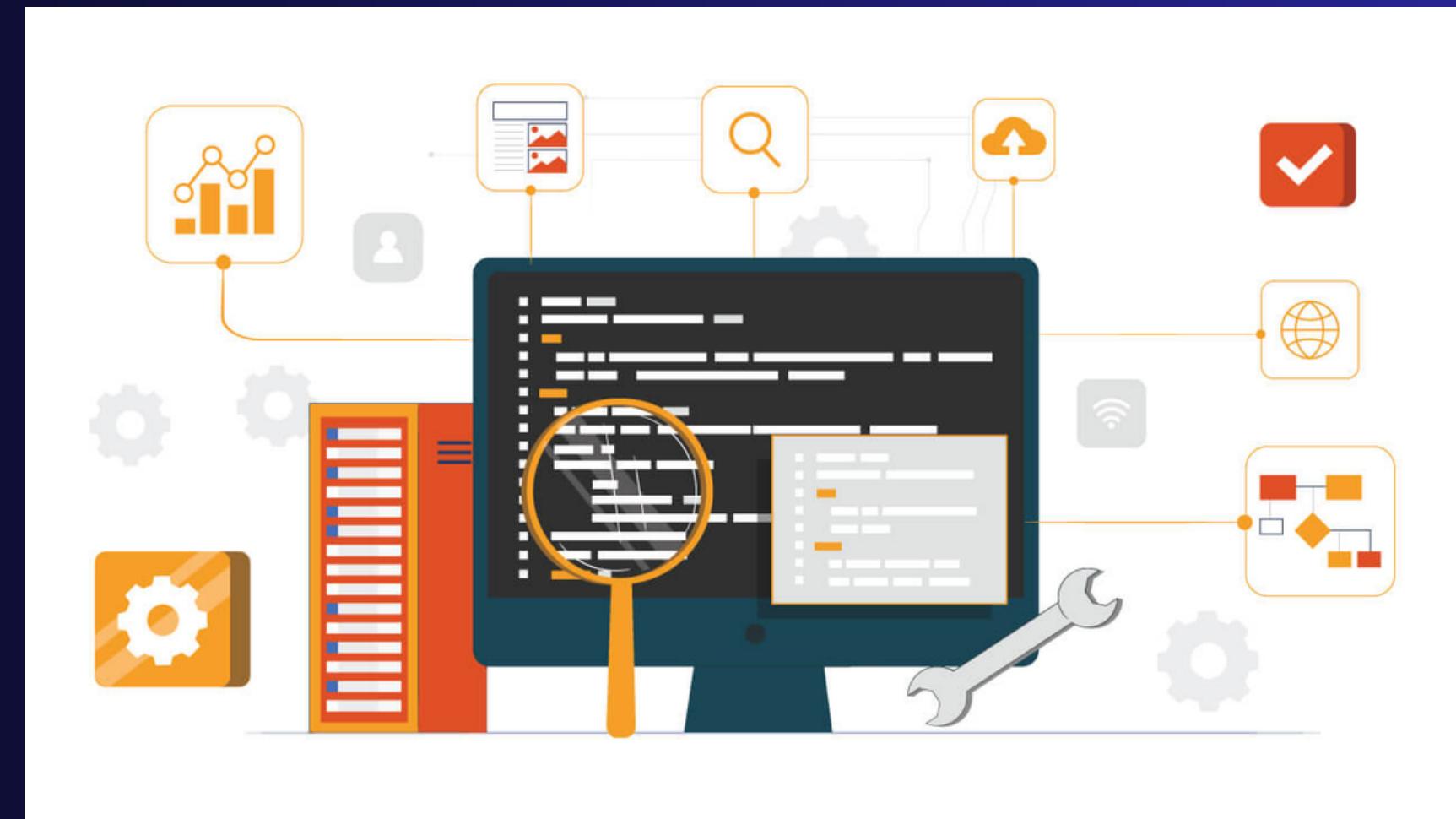
# Main Objective

The overall objective of the research is to improve software development and deployment processes by addressing limitations in software release process automation, monitoring tool integration, resource allocation, and multi-cloud deployment.

# Proposed Solution



# OPTIMIZING AND AUTOMATING SOFTWARE RELEASE PROCESS



**Herath H.M.I.P. - IT20125516**  
**BSc. Software Engineering (SLIIT)**

# Target Problem

The majority of software release process solutions now in use rely on manual execution, which has a number of drawbacks including poor architectural design, a higher chance of human error, excessive time consumption, and a sensitivity to errors.

# Existing Solutions

- GitHub Actions are one of the most popular solution but it cannot run heavy-compute jobs. Also take more time to execute.
- Circle CI is one another platform with hard to debug issues with builds and unable to find help, customizations not supported and Outages impacted.
- There is no any solution for automated software releasing as a optimized solution.

# Problem Definition

The current market does not offer an enhanced software solution to accelerate the deployment of container applications to the Kubernetes cluster while automating the creation, storage, and transfer of container images.

# Proposed Solution

Rearchitect a software release process by enhancing it in accordance with architectural characteristics like performance, security, extendibility, reliability, and modifiability and fully automating the suggested software release process architecture. This software release process can then be used frequently for the majority of software release processes.

# Objectives

## Functional Properties

- Build container image and push to the container registry by automatically triggered pipeline.
- Reduce the build time and image publish time.
- Increase the security, reliability, modifiability and extendability of the software release process.
- Fast deployment by fully automated sync process with GitOps.
- Multi-Cloud deployment feasibility with the software release process.

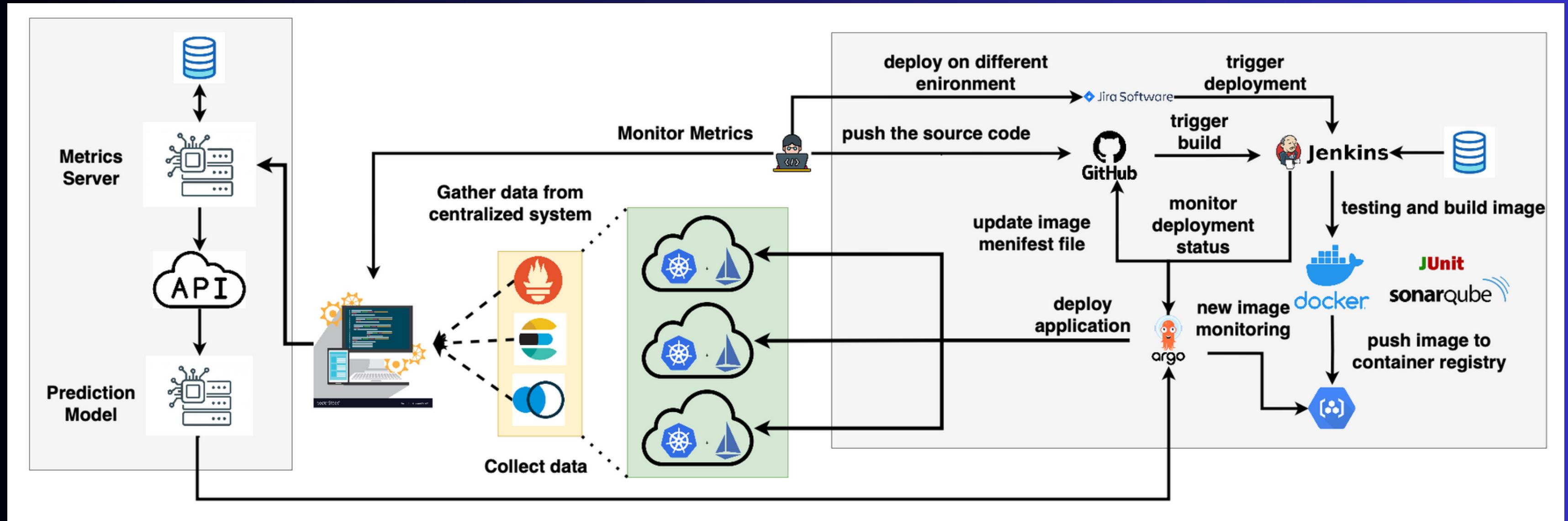
# Objectives

## Non-functional Properties

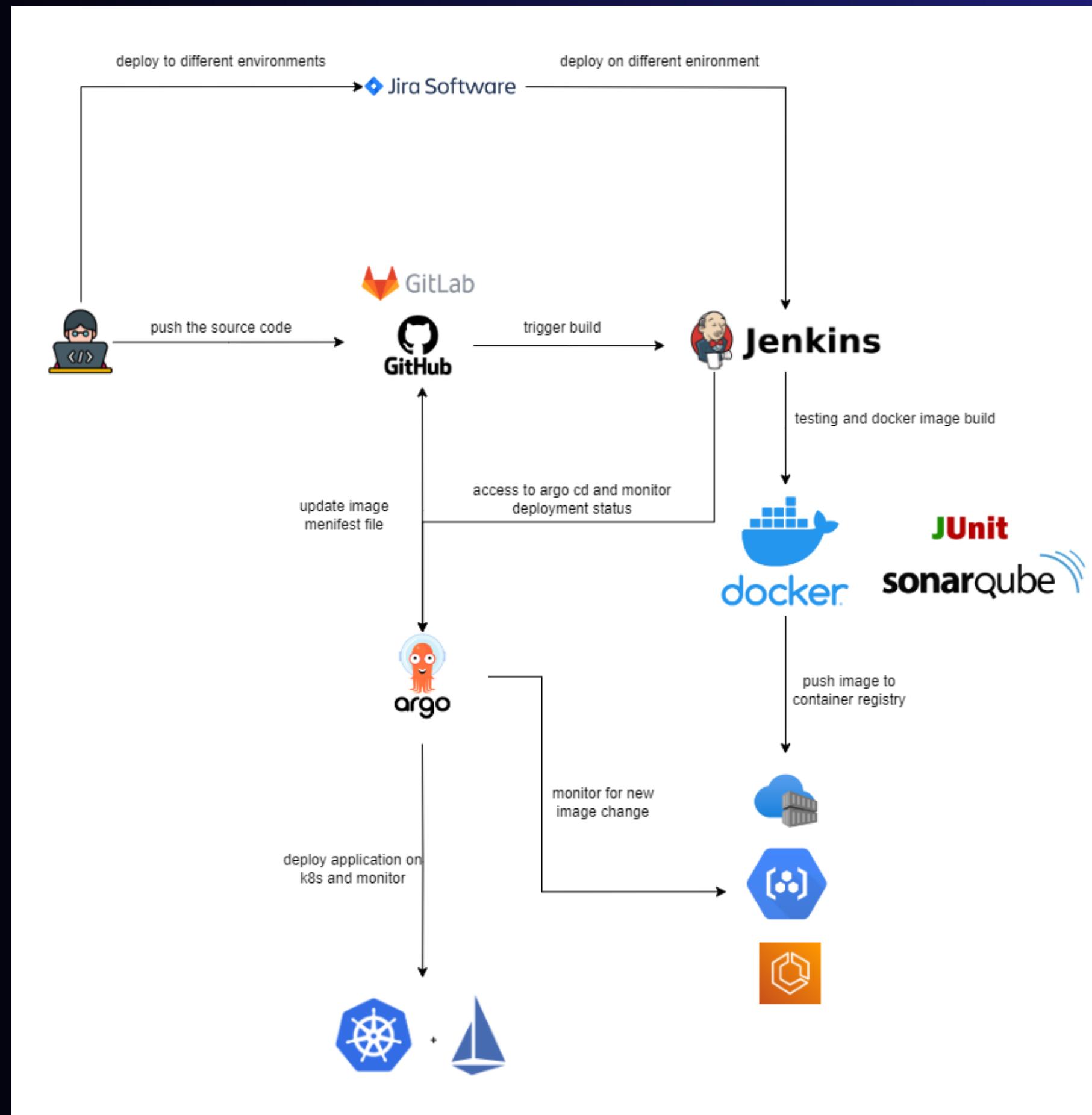
- Track the development life cycle improvement
- Server availability and the fast response of the system
- Increase the performance of deployed software
- Track the deployment speed and the success rate

# Best Practices for Security Enhancement

- All the secrets are saved as Jenkins credentials.
- The system is sync with the git repository to increase the availability and the performance.
- The downloaded files are automatically deleted as junk files once the pipelines are completed.
- The pipelines can be locked with a manual password or the proposed Jira architecture can be used to manage the permission of the deployments.
- The webhooks are used to trigger from Jira and GitHub.
- The GitHub authentication or other SAML is recommended as a security improvement of the system.

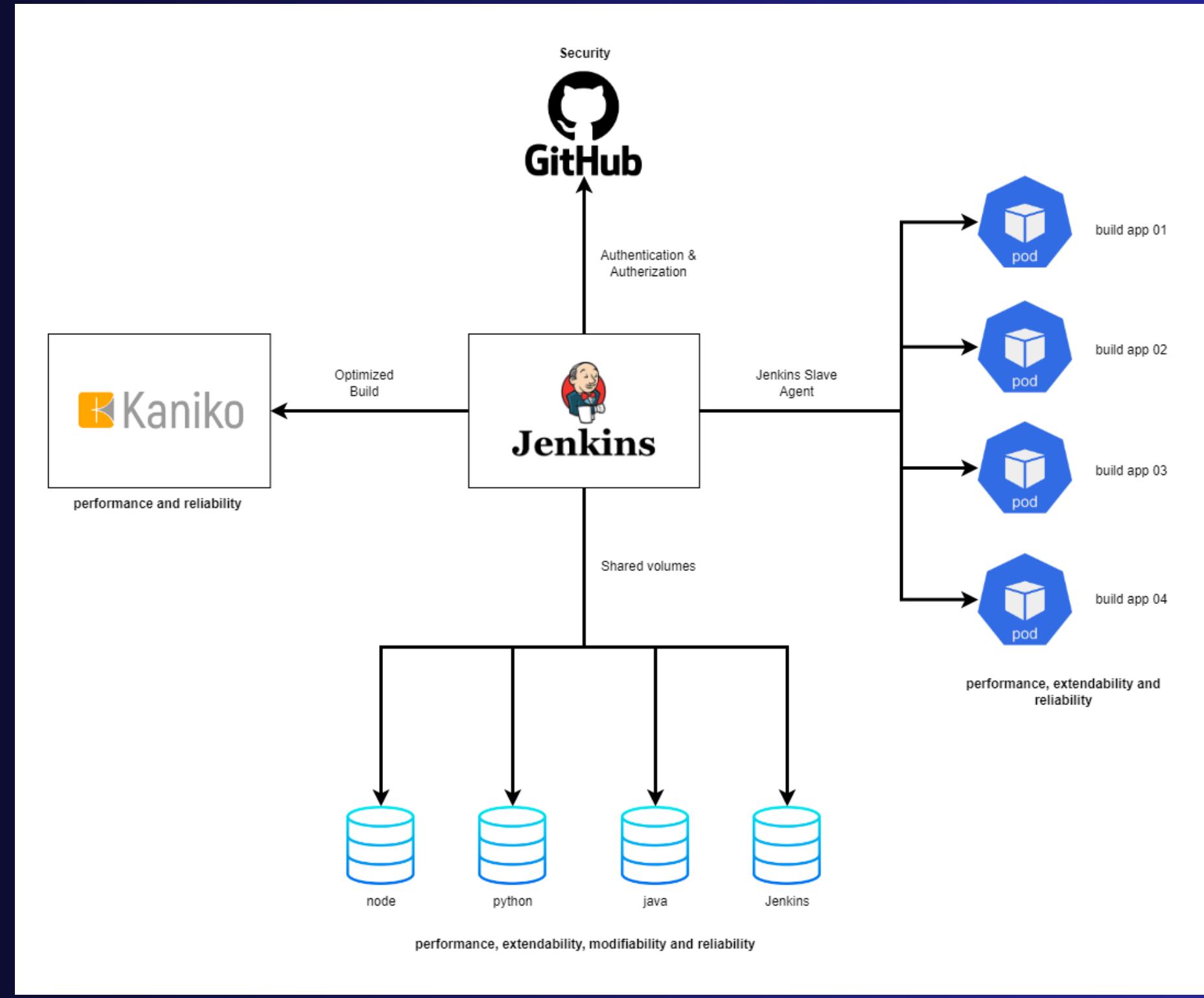


## proposed Automated CI/CD Architecture



# proposed Automated CI/CD Architecture

# Optimization of Pipeline Architecture



# Test & Results - CI

Tested the proposed system in LOLC FusionX Project with adding all of the components related to the CI/CD Arhitecture

Programming Language	Average Time to build in previous system	Average Time to build with proposed system	Average builds per day	Performance Improvment
Java (Spring Boot)	10 - 15 min	5 - 10 min	30 +	33.33 %
JavaScript (React)	15 - 25 min	5 - 15 min	10 +	40 %

# JavaScript Project Builds

Jenkins

This screenshot shows the Jenkins activity log. It lists 18 entries, each with a status icon, run number, commit hash, branch, message, duration, and completion time. The messages include chore tasks, pull request updates, and Jenkinsfile configuration changes.

STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	34	5348b6c	main	chore: version bump 1.15.0 [no ci] <small>3 commits</small>	6m 50s	3 days ago
✗	2	7e1a634	PR-51	Pull request #51 updated	4m 16s	3 days ago
✓	14	7e1a634	review	chore: version bump 1.15.0 [no ci]	5m 0s	3 days ago
✓	1	5927162	PR-51	Pull request #51 opened	4m 32s	3 days ago
✓	13	5927162	review	build(Jenkinsfile): updated node ve... <small>351 commits</small>	5m 8s	3 days ago
✓	33	da3560a	main	Update Jenkinsfile <small>350 commits</small>	4m 37s	3 days ago
✗	1	ef11963	PR-50	Pull request #50 opened	55s	3 days ago
✓	1	ef11963	br0sive-patch-1	Push event to branch br0sive-patch-1	4m 30s	3 days ago
✓	2	0d0f268	PR-43	v1.14.0 bump <small>1033 commits</small>	5m 3s	8 days ago
✓	32	90cd41f	main	v1.14.0 bump <small>1026 commits</small>	4m 50s	8 days ago
✓	2	c0dd420	PR-49	v1.14.0 bump	4m 6s	8 days ago
✓	12	c0dd420	review	v1.14.0 bump	4m 11s	8 days ago
✓	1	bb5fd59	PR-49	Pull request #49 opened	4m 41s	8 days ago

Jenkins

This screenshot shows the Jenkins activity log. It lists 15 entries, each with a status icon, run number, commit hash, branch, message, duration, and completion time. The messages include pull request opens, pushes to branches, and chore tasks.

STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	1	81dcbb5	feat/cheque-bbok-issu...	Push event to branch feat/cheque-bbok-issuanc...	10m 40s	an hour ago
✓	1	f59f8b8	PR-1685	Pull request #1685 opened	10m 7s	6 hours ago
✓	3	f59f8b8	2023/8/31	fix(account-approve): FT2-247 <small>21 commits</small>	11m 4s	6 hours ago
✓	339	7b09178	main	chore: version bump 2.26.0 [no ci] <small>224 commits</small>	10m 16s	7 hours ago
✓	2	43fb82b	PR-1684	chore: version bump 2.26.0 [no ci]	14m 48s	8 hours ago
✓	1082	43fb82b	review	chore: version bump 2.26.0 [no ci]	11m 49s	8 hours ago
✓	1	7db6986	PR-1684	Pull request #1684 opened	9m 21s	8 hours ago
✓	1081	7db6986	review	Push event to branch review	10m 16s	8 hours ago
✓	1080	7db6986	review	feat(Reports): FT3-296 <small>5 commits</small>	14m 0s	8 hours ago
✓	3	2604199	fix/casa/ft3-296	fix(acc-opening): FT1-217 <small>11 commits</small>	10m 14s	8 hours ago
✓	1079	2ed2cf3	review	fix(CASA Account Approve): FT2-244 <small>2 commits</small>	10m 24s	8 hours ago
✓	1078	572f967	review	fix(acc-opening): FT1-217 <small>3 commits</small>	11m 1s	8 hours ago
✓	1077	1fd00f5	review	feat(report): FT2-238 <small>78 commits</small>	14m 21s	8 hours ago

# JavaScript Project Builds

The image shows two Jenkins project dashboards side-by-side.

**Project 1: comn-react-module-auth**

STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	13	cca8ffd	main	chore: version bump 1.9.0 [no ci] 5 commits	4m 10s	3 days ago
✓	3	a38ba95	PR-84	Pull request #84 updated	3m 39s	3 days ago
✓	9	a38ba95	review	chore: version bump 1.9.0 [no ci]	4m 2s	3 days ago
	3b41d84	PR-84		Pull request #84 updated	4m 14s	3 days ago
	3b41d84	review		feat(github/workflows): FT1-64	4m 6s	3 days ago
	3354e95	PR-84		Pull request #84 opened	7m 38s	3 days ago
	3354e95	review		build(Jenkinsfile): updated node v...	4m 31s	3 days ago
	21377f7	main		Update Jenkinsfile	4m 3s	3 days ago
	e42e9b1	PR-83		Pull request #83 opened	4m 11s	3 days ago
	e42e9b1	br0sive-patch-1		Push event to branch br0sive-patch-1	5m 13s	3 days ago
	b5f02d9	main		v1.8.1 bump	5m 11s	a month ago
	9693d36	PR-82		Pull request #82 opened	5m 23s	a month ago
	9693d36	review		v1.8.1 bump	4m 0s	a month ago

**Project 2: comn-react-module-cob**

STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	272	2f73709	review	refactor(package.json): engines upda... 9 commits	5m 7s	3 days ago
✓	149	b935145	main	chore: version bump 1.26.0 [no ci] 5 commits	5m 24s	3 days ago
✓	2	1f909ea	PR-403	Pull request #403 updated	5m 17s	3 days ago
✓	271	1f909ea	review	chore: version bump 1.26.0 [no ci] 3 commits	4m 32s	3 days ago
✓	1	4d1aab5	PR-403	Pull request #403 opened	5m 8s	3 days ago
✓	270	4d1aab5	review	build(Jenkinsfile): updated node ver... 37 commits	4m 43s	3 days ago
✓	148	758387a	main	Update Jenkinsfile	4m 30s	3 days ago
✓	1	80f769d	PR-402	Pull request #402 opened	5m 6s	3 days ago
✓	1	80f769d	br0sive-patch-1	Push event to branch br0sive-patch-1	5m 14s	3 days ago
✓	147	f0a6389	main	chore: version change v1.25.4 130 commits	5m 9s	3 days ago
✓	1	b494804	PR-401	Pull request #401 opened	5m 0s	3 days ago
✓	269	b494804	review	chore: version change v1.25.4 129 commits	4m 55s	3 days ago
✗	1	dd2bc5f	PR-400	Pull request #400 opened	53s	3 days ago

# Java Spring Boot Project Builds

The image displays two Jenkins project dashboards side-by-side. The left dashboard is for the 'comn-customer' project, and the right one is for the 'lending-product' project. Both dashboards show a table of build history with columns for Status, Run, Commit, Branch, Message, Duration, and a link to the build details.

**comn-customer Build History:**

Status	Run	Commit	Branch	Message	Duration
✓	304	52a5a43	main	FT3-302: Fixing issue	5m 26s
✓	1	c08d167	PR-466	Pull request #466 opened	4m 58s
✓	5	c08d167	FXN-4048	FT3-302: Fixing issue	5m 9s
✓	4	8c676e0	FXN-4048	build: automatic update of fusionx...	6m 9s
✓	303	8c676e0	main	FT1-84: Get API changes.	6m 29s
✓	1	1368d03	PR-465	Pull request #465 opened	5m 58s
✓	3	1368d03	FXN-4048	FT1-84: Get API changes.	5m 17s
✓	302	c41e444	main	FT1-84: Trade status changes.	5m 17s
✓	1	b1dd433	PR-464	Pull request #464 opened	5m 27s
✓	2	b1dd433	FXN-4048	Push event to branch FXN-4048	6m 17s
✗	1	5a524c7	FXN-4048	Push event to branch FXN-4048	<1s
✓	301	5a524c7	main	updated for data migration.	7m 51s
✗	1	f65311b	PR-463	Pull request #463 opened	1m 11s

**lending-product Build History:**

Status	Run	Commit	Branch	Message	Duration	Completed
✓	1	e851fe0	PR-611	Pull request #611 opened	7m 11s	3 days ago
✓	69	e851fe0	FXN-1520	FXL-19481	5m 46s	3 days ago
✗	23	89da2ac	PR-588	kenya	11m 41s	4 days ago
✓	552	89da2ac	main	bug fix	6m 28s	4 days ago
✓	1	85635bc	PR-610	Pull request #610 opened	6m 27s	4 days ago
✓	3	85635bc	fxn-4023	bug fix	5m 40s	4 days ago
✗	22	7d65e46	PR-588	build: automatic update of fusionx-sl...	10m 17s	7 days ago
✓	551	7d65e46	main	updated	9m 25s	7 days ago
✗	21	1ddf55e	PR-588	build: automatic update of fusionx-...	11m 1s	7 days ago
✓	550	1ddf55e	main	fix	8m 39s	7 days ago
✓	1	6080e59	PR-609	Pull request #609 opened	6m 30s	7 days ago
✓	58	6080e59	FXN-557	fix	9m 1s	7 days ago
✓	1	8e8bb9e	PR-609	Pull request #608 opened	7m 47s	7 days ago

# Java Spring Boot Project Builds

The image shows two Jenkins dashboards side-by-side, illustrating the build process for Java Spring Boot projects.

**Left Dashboard (leg-legal Project):**

- Project Name:** leg-legal
- Pipeline Status:** Active
- Build Log:** Shows 164 runs. The most recent run (Run 164) was successful, taking 4m 5s and involved 20 commits. Other runs include PR-128, FXN-3034, and various main branches.

**Right Dashboard (term-deposit Project):**

- Project Name:** term-deposit
- Pipeline Status:** Active
- Build Log:** Shows 538 runs. The most recent run (Run 538) was successful, taking 6m 28s and involved 27 commits. Other runs include PR-565, FXN-24, and various main branches.

STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	164	9f7457a	main	Fixed	20 commits 4m 5s	
✓	1	dd5b1e5	PR-128	Pull request #128 opened		4m 42s
✓	43	dd5b1e5	FXN-3034	Fixed	5 commits 3m 41s	
✓	1	6d61931	PR-127	Pull request #127 opened		6m 20s
✓	42	6d61931	FXN-3034	Fixed	16 commits 5m 39s	
✓	163	5e3cf20	main	Fixed	17 commits 4m 10s	
✓	2	06978aa	PR-126	build: automatic update of fusionx-sl-qa-td		4m 8s
✓	41	06978aa	FXN-3034	Fixed	2 commits 4m 38s	
✓	1	644c383	PR-126	Pull request #126 opened		4m 44s
✓	40	644c383	FXN-3034	Fixed	16 commits 4m 25s	
✓	162	f38aca3	main	Since there are no UI to add signat...	22 commits 4m 33s	
✓	1	e0a31e4	PR-125	Pull request #125 opened		6m 17s
✓	39	e0a31e4	FXN-3034	Since there are no UI to add signat...	20 commits 3m 41s	

STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	542	d7fd0ba	main	updated	31 commits 5m 26s	6 days ago
✓	1	6c1d4dd	PR-565	Pull request #565 opened		6m 26s
✓	460	6c1d4dd	FXN-24	updated	31 commits 5m 37s	6 days ago
✓	541	066dd97	main	updated	191 commits 5m 29s	7 days ago
✓	1	a613672	PR-564	Pull request #564 opened		5m 0s
✓	459	a613672	FXN-24	updated	191 commits 4m 53s	7 days ago
✓	540	5e39261	main	updated	38 commits 7m 31s	10 days ago
✓	1	a839d32	PR-563	Pull request #563 opened		4m 48s
✓	458	a839d32	FXN-24	updated	38 commits 5m 22s	10 days ago
✓	539	e1d3b1d	main	updated	19 commits 7m 46s	10 days ago
✗	1	14dd208	PR-562	Pull request #562 opened		2m 17s
✓	457	14dd208	FXN-24	updated	22 commits 9m 21s	10 days ago
✓	538	e28c321	main	updated	27 commits 6m 28s	10 days ago

# Test & Results - CD

Programming Language	Average Time to Deploy in previous system	Average Time to Deploy with the proposed system	Average builds per day	Performance Improvment
Java (Spring Boot)	10 - 15 min	4 - 5 min	50 +	66.66 %
JavaScript (React)	15 - 25 min	4 - 5 min	30 +	80 %

# Application Deployment

Jenkins

fusionx-argocd-autopilot ★ ⚙

Activity Branches Pull Requests

Status	Run	Commit	Branch	Message	Duration	Completed
✓	lending-oi	8e42568	main	Executed By: Sewandi Hansika Peter	3m 32s	7 minutes ago
✓	lending-oi	8e42568	main	Executed By: Sewandi Hansika Peter	4m 7s	6 minutes ago
✓	casa-trans	22a1af7	main	Executed By: Sanatha De Silva	2m 46s	14 minutes ago
✓	casa-trans	d549ce6	main	Executed By: Sanatha De Silva	5m 42s	26 minutes ago
✓	casa-trans	d549ce6	main	Executed By: Sanatha De Silva	3m 8s	33 minutes ago
✓	casa-trans	eefd306	main	Executed By: Sanatha De Silva	4m 8s	an hour ago
✓	casa-trans	eefd306	main	Executed By: Sanatha De Silva	2m 49s	an hour ago
✓	casa-trans	eefd306	main	Executed By: Sanatha De Silva	2m 57s	an hour ago
✓	lending-oi	8eb2ca5	main	Executed By: Sewandi Hansika Peter	6m 27s	2 hours ago
✓	lending-oi	8eb2ca5	main	Executed By: Sewandi Hansika Peter	2m 48s	2 hours ago
✓	lending-ac	8eb2ca5	main	Executed By: Sewandi Hansika Peter	2m 48s	2 hours ago
✓	lending-ac	8eb2ca5	main	Executed By: Sewandi Hansika Peter	4m 26s	2 hours ago
✓	lending-ac	8eb2ca5	main	Executed By: Sewandi Hansika Peter	2m 53s	2 hours ago

# Feedback from the User Experience

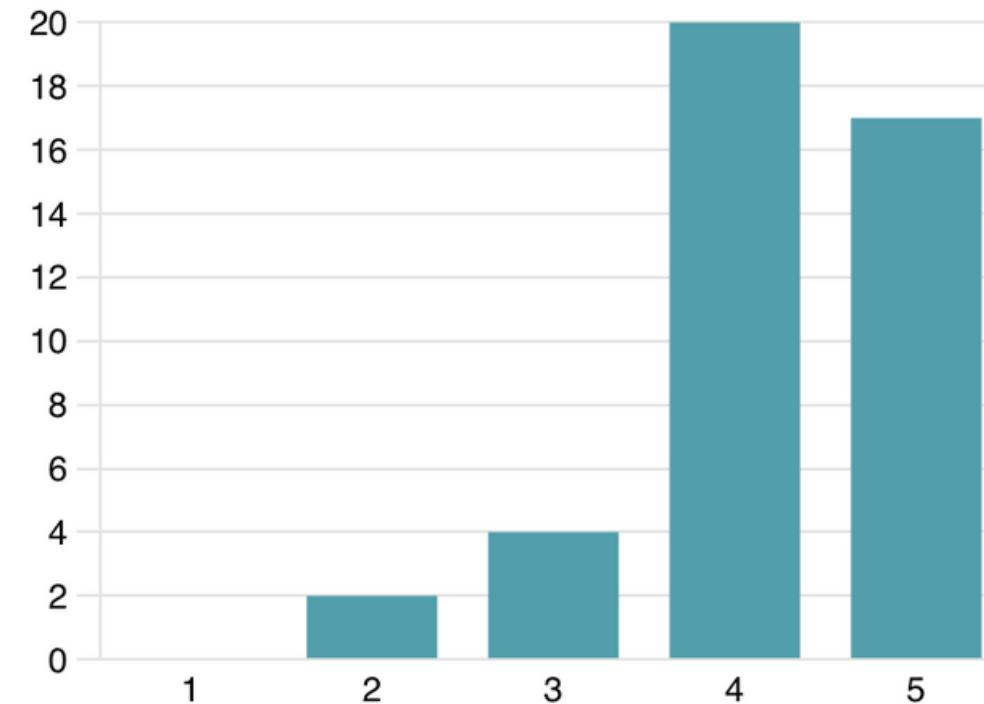
43	05:40	Active
Responses	Average time to complete	Status

10. Rate your overall satisfaction with the automated release process and its impact on your work.

[More Details](#)

 Insights

4.21  
Average Rating





# CENTRALIZED CLOUD VISUALIZATION TOOL USING OPEN-SOURCE MONITORING TOOLS.

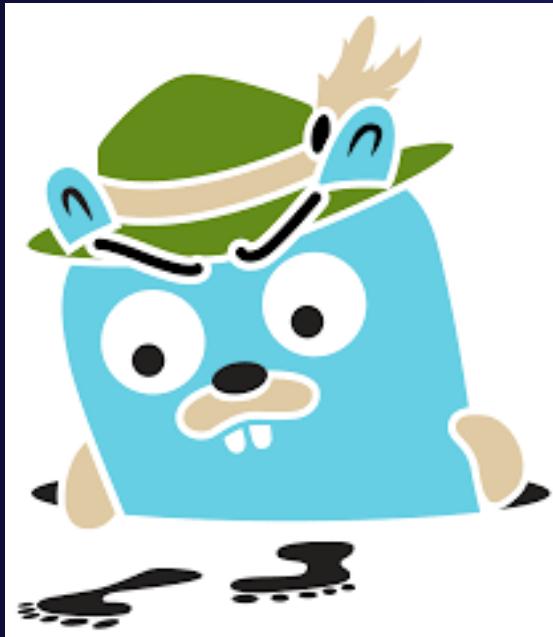


**Jayawardena R.D.S.H - IT20074968**  
**BSc Software Engineering (SLIIT)**

# Target Problem

Users switch between multiple monitoring tools for collecting and visualizing application data, such as metrics, logs, and Traces. This leads to inefficiencies and fragmentation in data analysis.

# Existing Solution



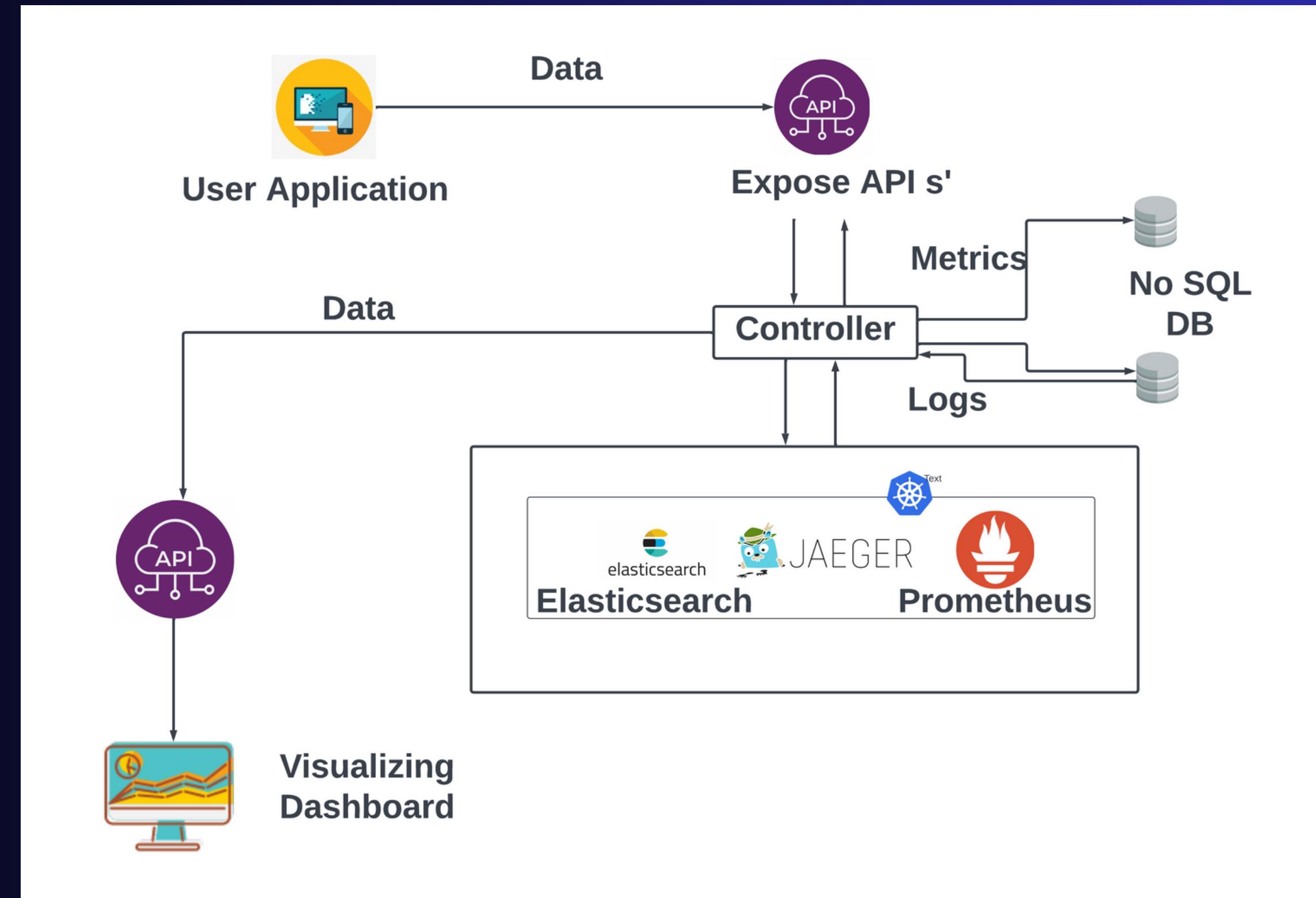
Current solutions such as Prometheus, Kiali, Jaeger, and Elasticsearch are specialized for specific data types, requiring users to switch between tools. We need a unified solution for diverse data types

# Proposed Solution

Develop an application that allows users to configure and collect various types of application data in a unified manner. Provide a consolidated visualization platform for comprehensive analysis and decision-making.



# SYSTEM ARCHITECTURE



# Objectives

## Functional Properties

- 1. Data Configuration:** Users can easily configure different types of application data sources, including logs, metrics, traces, and custom data.
- 2. Data Collection:** The application collects data from configured sources, ensuring data completeness and accuracy.
- 3. Unified Data Repository:** All collected data is stored in a single, unified repository for easy access and analysis.
- 4. Visualization:** The solution provides a consolidated visualization platform, allowing users to create custom dashboards, charts, and reports for comprehensive analysis.
- 5. Data Aggregation:** Users can aggregate and correlate data from multiple sources for a holistic view of application performance.

# Non-Functional Properties

1. **Usability:** The user interface should be intuitive and user-friendly to minimize the learning curve for users.
2. **Interoperability:** Support integration with other tools and systems commonly used in the organization's tech stack.
3. **Monitoring and Logging:** Implement comprehensive monitoring and logging for system health and troubleshooting.
4. **Cost-Efficiency:** Optimize resource utilization and costs associated with data storage and processing.
5. **Scalability:** The solution should scale horizontally to handle a growing volume of data and users.

# Testing & Results

## API Testing

MonitoringAPI 1.0 OAS3  
https://localhost:7024/swagger/v1/swagger.json

### DataSourceValidation

POST /api/DataSourceValidation/validate

### Metrics

POST /api/Metrics/log

POST /api/Metrics/metric

POST /api/Metrics/timeseries

GET /api/Metrics/metrics

POST /api/Metrics/traces

GET /api/Metrics/logs

GET /api/Metrics/

localhost:7024/swagger/index.html

GET /api/Metrics/customtraces

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7024/api/Metrics/customtraces' \
  -H 'accept: */*'
```

Request URL

https://localhost:7024/api/Metrics/customtraces

Server response

Code Details

200 Response body

```
{
  "traceId": "abc123",
  "spanId": "def456",
  "operationName": "SampleOperation",
  "startTime": "2023-09-04T10:32:29.945Z",
  "duration": "00:00:02.500000"
},
{
  "traceId": "abc123",
  "spanId": "def456",
  "operationName": "SampleOperation",
  "startTime": "2023-09-04T10:32:29.945Z",
  "duration": "00:00:02.500000"
}
```

The screenshot shows a web browser window with the address bar set to `localhost:7024/swagger/index.html`. The main content area displays the Swagger UI for a custom metrics API. At the top left, there is a "Curl" section containing a command-line interface (CLI) command:

```
curl -X 'GET' \
  'https://localhost:7024/api/Metrics/custommetrics' \
  -H 'accept: */*'
```

Below the "Curl" section is a "Request URL" field containing the URL `https://localhost:7024/api/Metrics/custommetrics`. Underneath these fields is a "Server response" section. It includes two tabs: "Code" and "Details". The "Code" tab is selected, showing the HTTP status code `200`. The "Details" tab is also visible. The response body is displayed as a JSON array:

```
[{"name": "SampleMetric", "value": 0, "description": "Sample metric description", "labels": {"additionalProp1": "value1", "additionalProp2": "value2", "additionalProp3": "value3"}, "timestamp": "2023-09-04T15:36:36.348Z"}, {"name": "ResponseTime", "value": 925, "description": "HTTP Response Time", "labels": {"http.status_code": "200", "endpoint": "/api/resource"}, "timestamp": "2023-09-04T15:36:36.217Z"}]
```

At the bottom right of the response body area, there are "Copy" and "Download" buttons. Below the response body, under the "Response headers" section, the value `content-type: application/json; charset=utf-8` is listed.

# Database

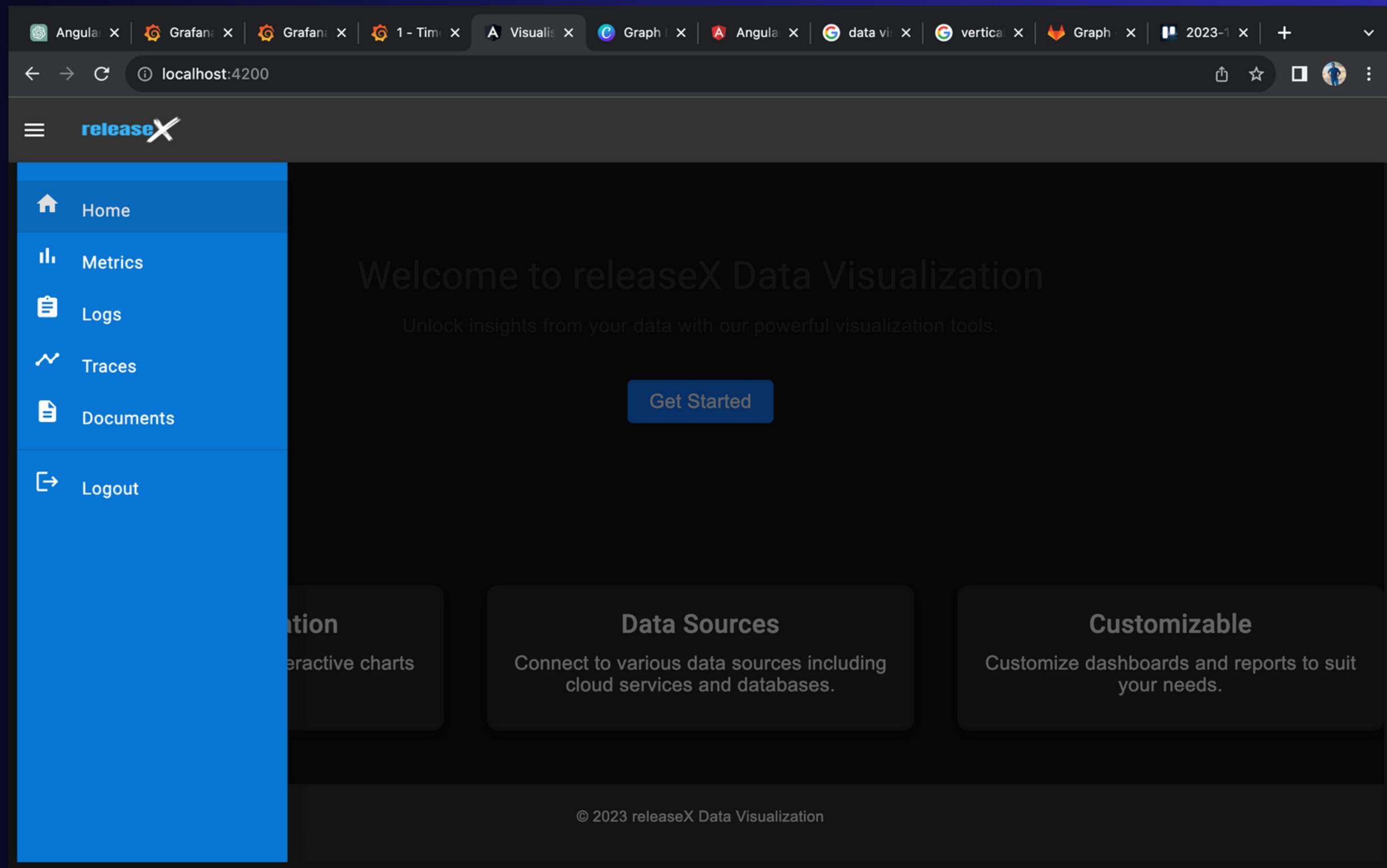
The screenshot shows the MongoDB Atlas Data Services interface. The left sidebar includes sections for MonitoringTool..., DEPLOYMENT, Database (selected), SERVICES, SECURITY, and a Goto button. The main area displays monitoringData with sub-options logs, metrics, timeSeries, and traces (selected). A search bar at the top right says "Search Namespaces". Below it, storage details are shown: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 15.68KB, TOTAL DOCUMENTS: 110, INDEXES TOTAL SIZE: 36KB. A "Find" tab is active, followed by Indexes, Schema Anti-Patterns (0), Aggregation, and Search Indexes. An "INSERT DOCUMENT" button is available. A "Filter" section allows querying with "Type a query: { field: 'value' }" and includes Reset, Apply, and More Options buttons. The results section shows "QUERY RESULTS: 1-20 OF MANY" with one document listed:

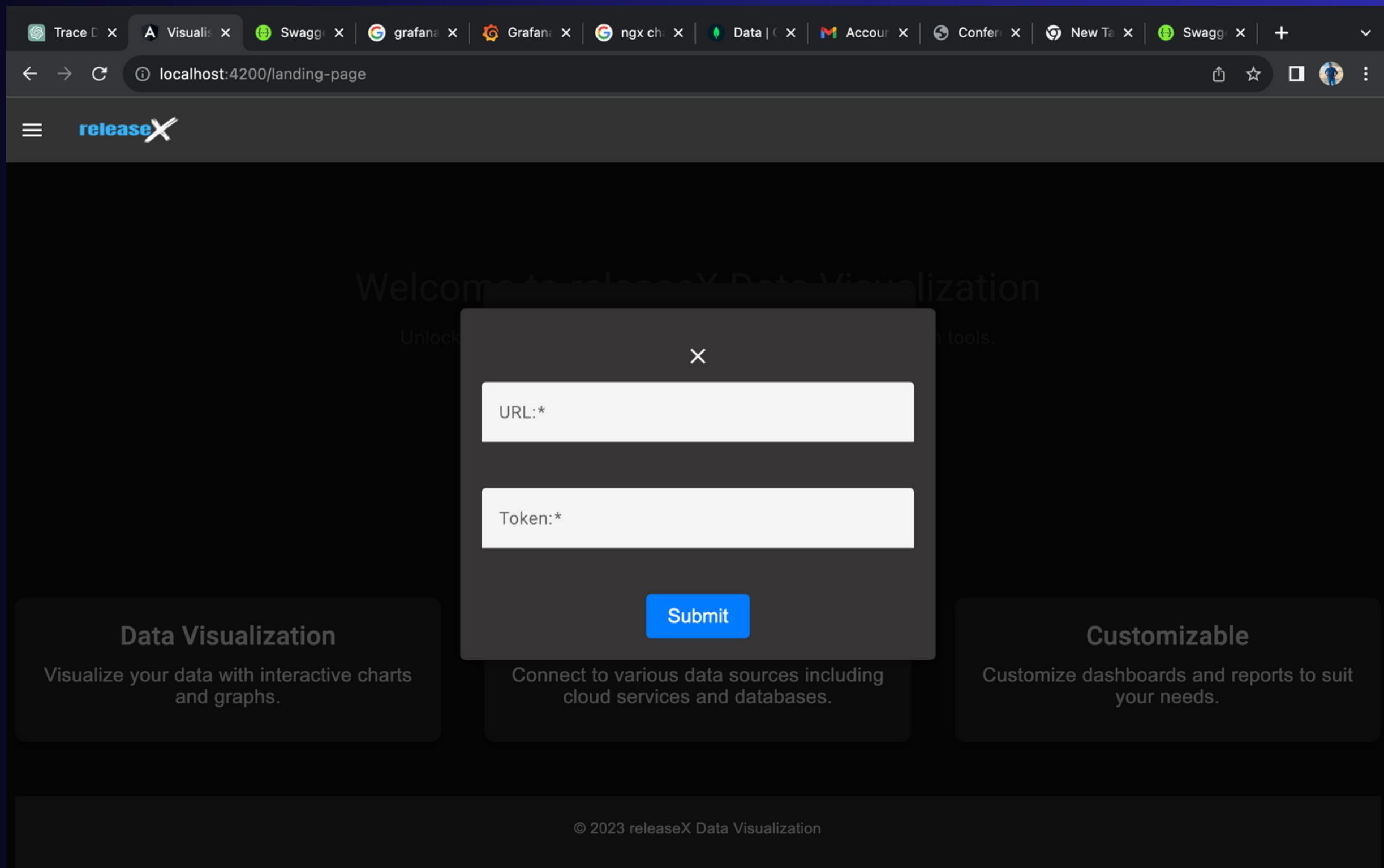
```
_id: ObjectId('64f322acd73e8ce460d088d5')
TraceId: "abc123"
SpanId: "def456"
OperationName: "SampleOperation"
StartTime: 2023-09-02T11:49:50.168+00:00
Duration: "00:00:02.5000000"
```

Navigation buttons for PREVIOUS, NEXT, and a message icon are at the bottom.

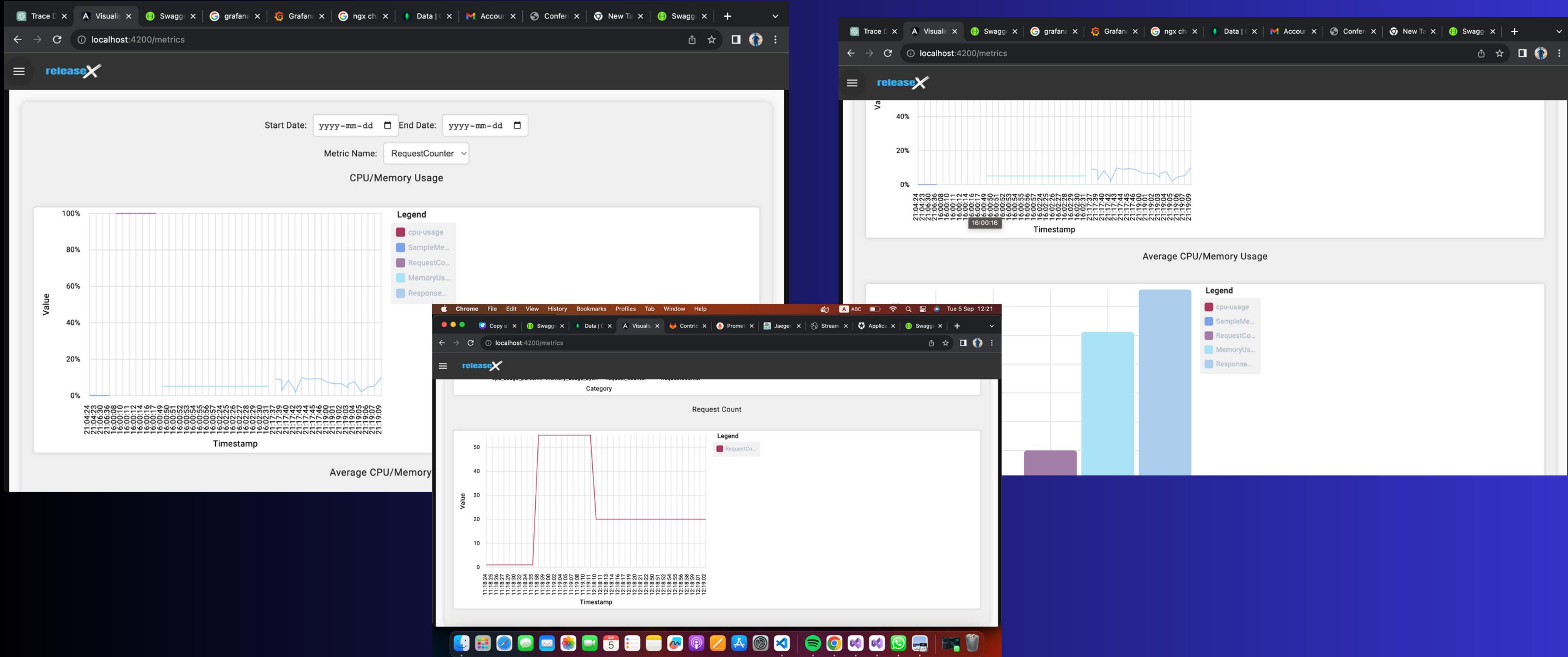
# Visualisation Tool

The screenshot shows a web browser window with the URL `localhost:4200` in the address bar. The title bar of the browser says "releaseX". The main content area is dark-themed with white text. At the top, it says "Welcome to releaseX Data Visualization" and "Unlock insights from your data with our powerful visualization tools.". A blue "Get Started" button is centered. Below this, there are three dark rectangular boxes with rounded corners: "Data Visualization" (with text "Visualize your data with interactive charts and graphs."), "Data Sources" (with text "Connect to various data sources including cloud services and databases."), and "Customizable" (with text "Customize dashboards and reports to suit your needs."). At the bottom center, it says "© 2023 releaseX Data Visualization".





# Metrics Dashboard

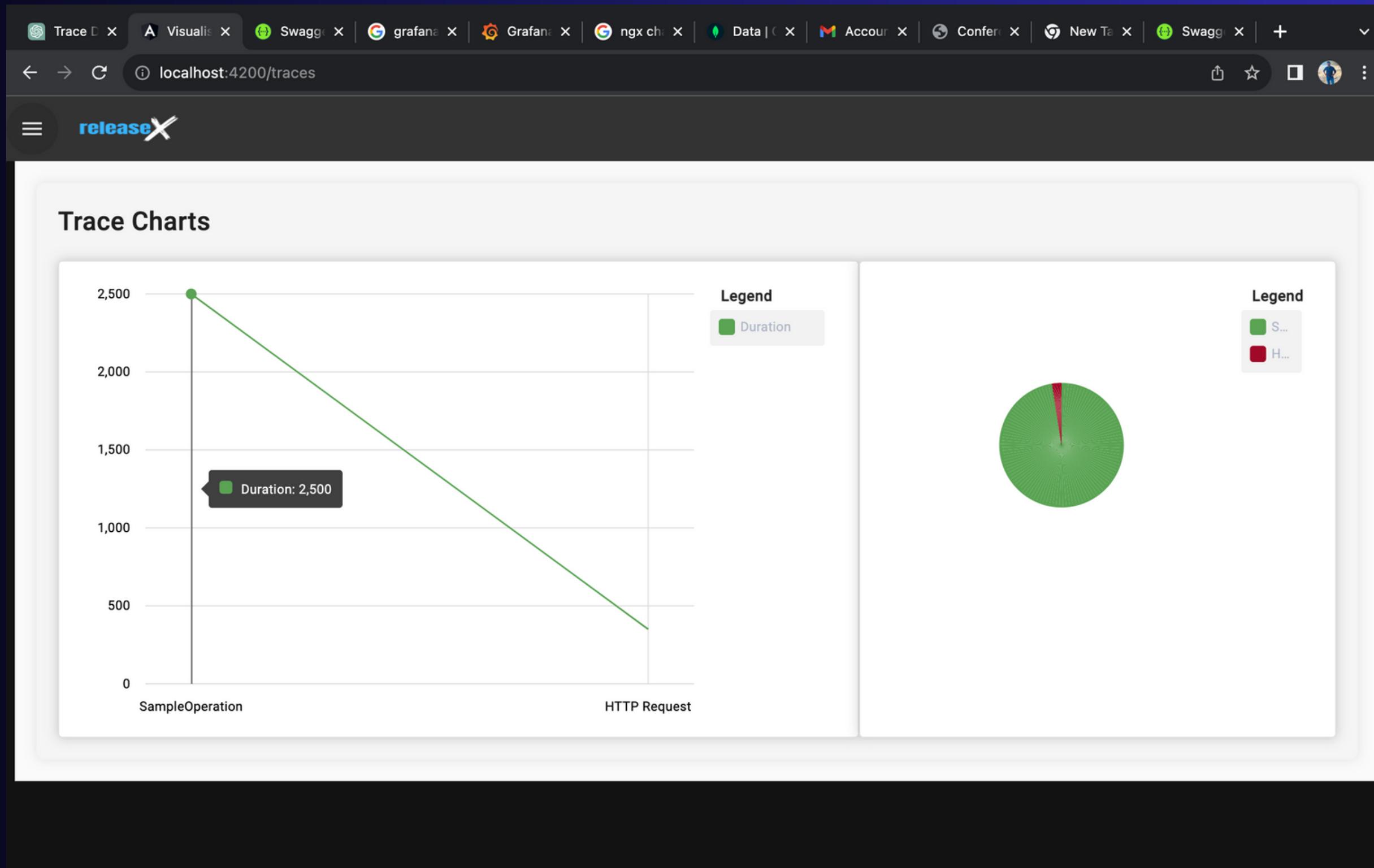


# Log Dashboard

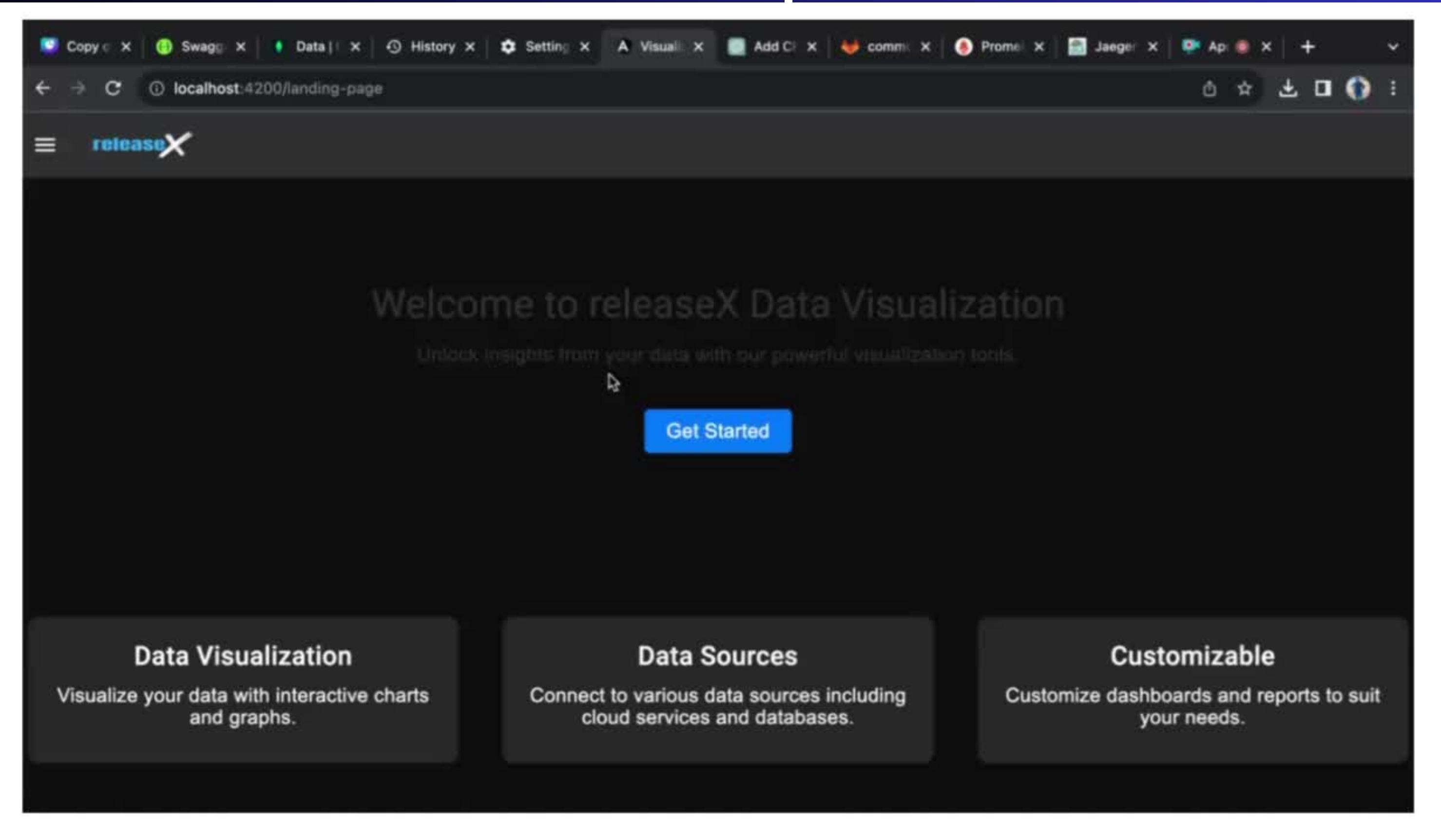
The screenshot shows a web browser window displaying a log dashboard. The title bar of the browser says "localhost:4200/logs". The dashboard has a header with a logo for "releaseX" and a "Log Entries" section. Below the header is a search bar with dropdowns for "Filter by Log Level" (set to "All"), "From Date", and "To Date". The main area is a table with three columns: "Timestamp", "Log Level", and "Message". The table contains ten rows of sample log entries, all with the timestamp "2023-09-03 17:03:11" and the log level "Information". Each entry has the message "Sample log message".

Timestamp	Log Level	Message
2023-09-03 17:03:11	Information	Sample log message
2023-09-03 17:03:13	Information	Sample log message
2023-09-03 17:03:15	Information	Sample log message
2023-09-03 17:03:16	Information	Sample log message
2023-09-03 17:03:17	Information	Sample log message
2023-09-03 17:03:19	Information	Sample log message
2023-09-03 17:03:20	Information	Sample log message
2023-09-03 17:03:21	Information	Sample log message
2023-09-03 17:03:21	Information	Sample log message

# Traces



# Test and Comparison



# Machine Learning workflow for Optimization of Kubernetes Deployment Performance.



**Fadhil M.R.A. - IT20784720**  
**BSc Software Engineering (SLIIT)**

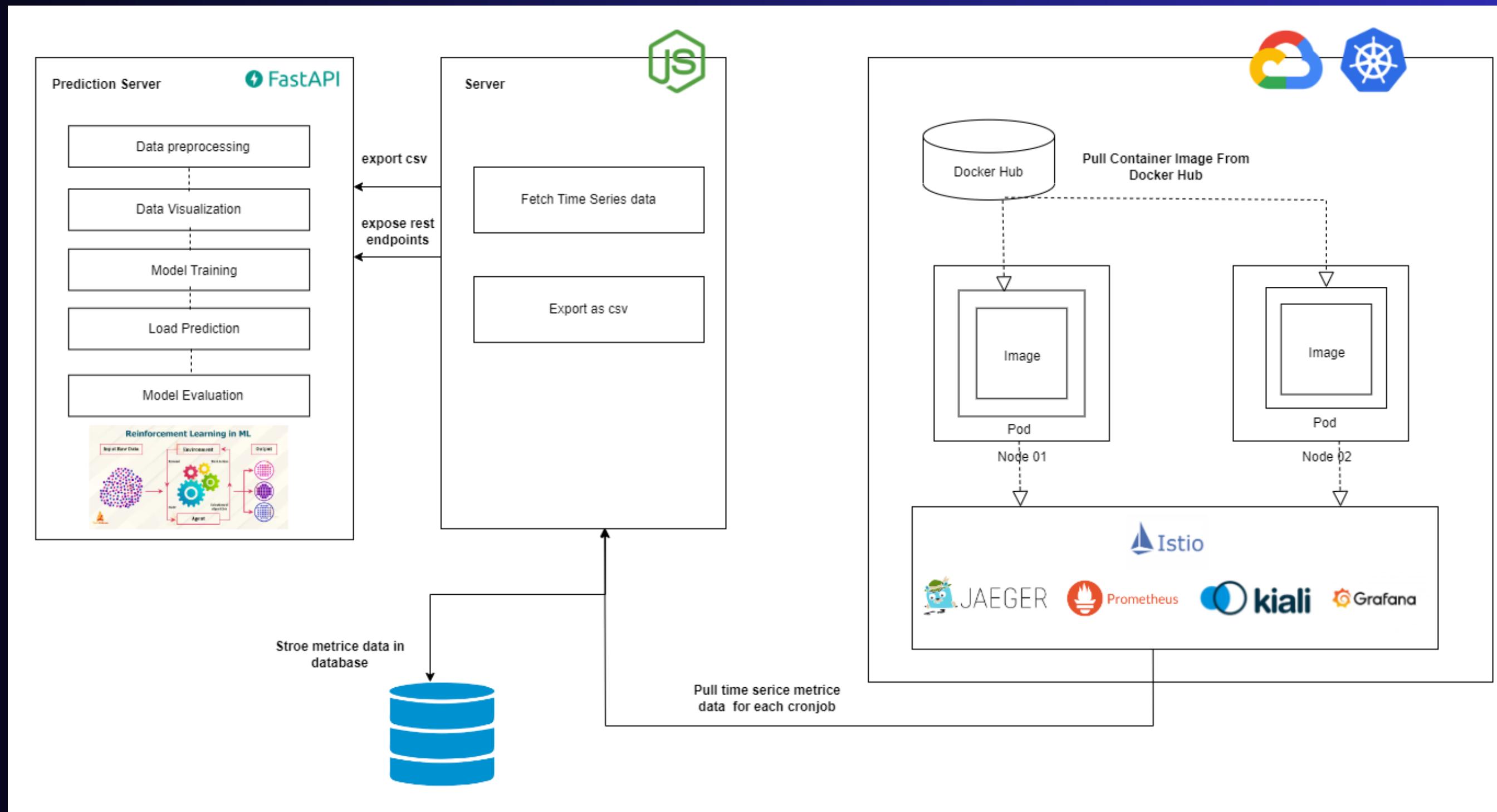
# Target Problem

there are few studies that focus on optimizing the resource workload of distributed applications using a machine learning model and both vertical and horizontal auto-scaling on Kubernetes, by gathering the pod level time series metrices using a Restapi and by creating an algorithm to scale the kubernets cluster according to the metrices workload.

# Proposed Solution

the proposed solution is to set up the Istio service mesh in Google Kubernetes Engine, and integrate prometheus , Graphana, Kiali , Jaeger and by deplying a sample microservice application on google kubernets engine and gether the time series metrice by creating a cron job , by using the reinforcement learning algorithm an optimized and fast prediction mechanism can be provide for scale the cluster.

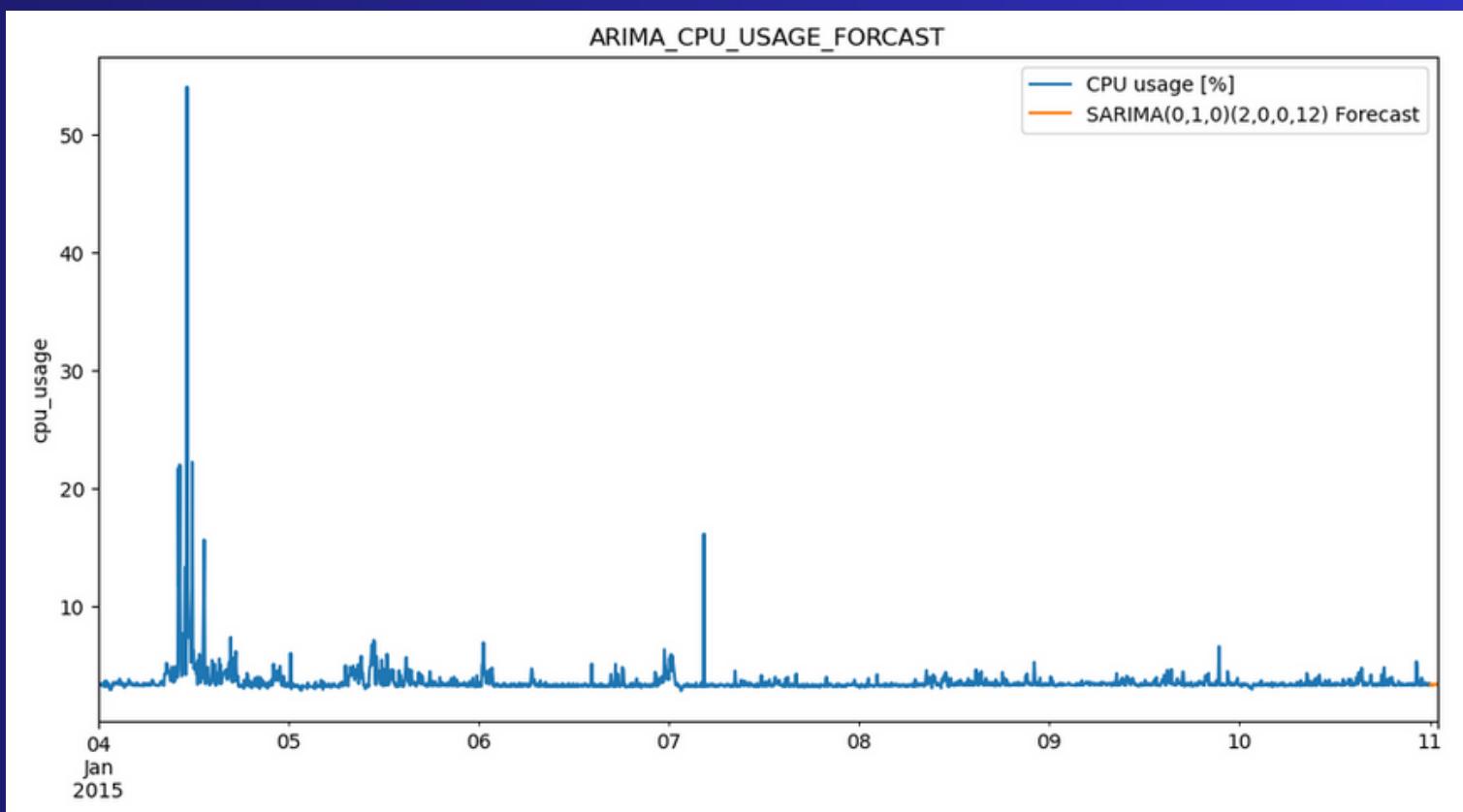
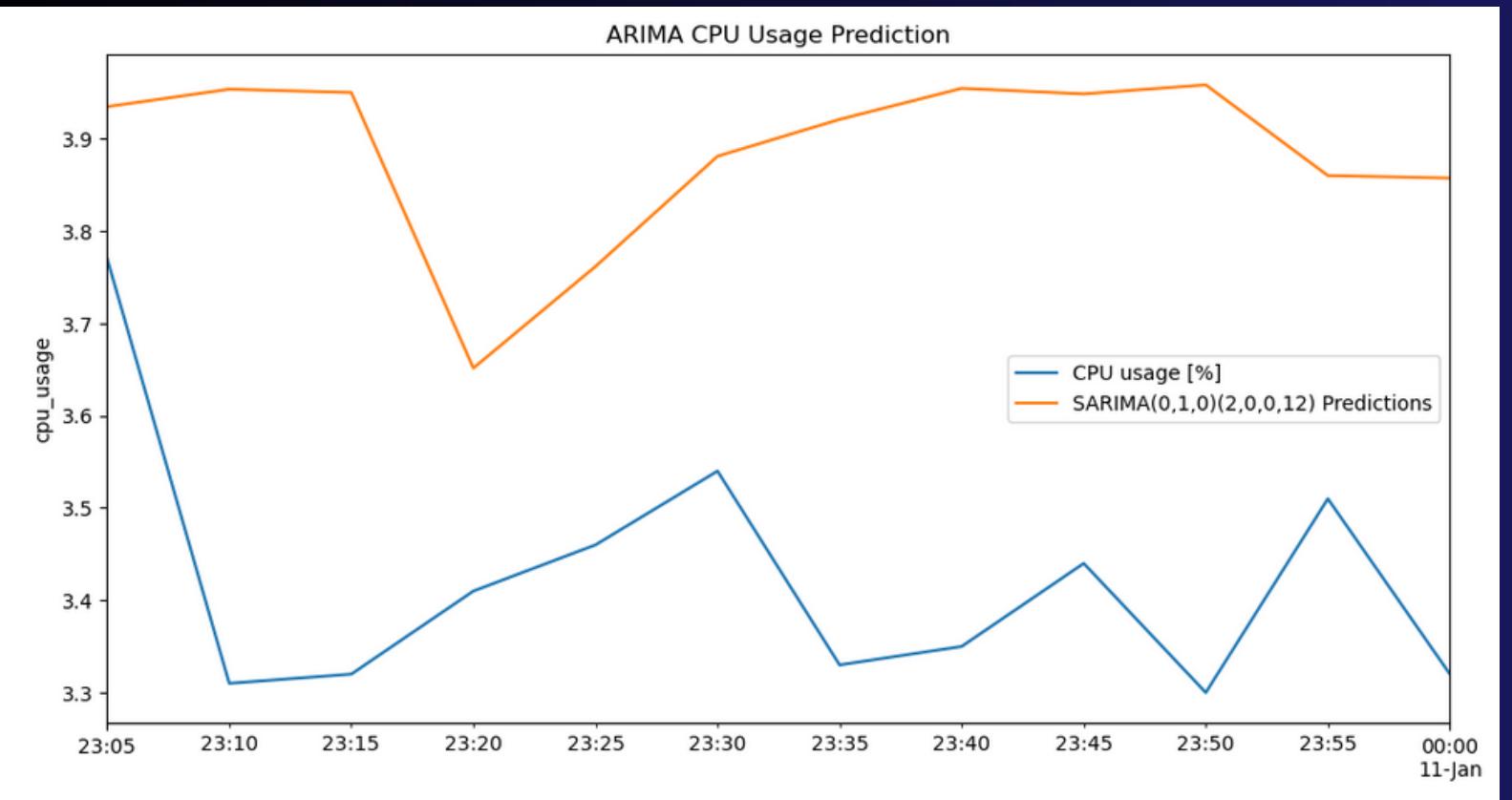
# SYSTEM ARCHITECTURE



# Functional Requirement

- The metrices server need to be available.
- The metrice must be able to fetch metrices data according to the cronjob with less time consuming.
- The prediction model need to be accurate.
- The prediction model must be able to dynamicaly predict the cpu workload.

# ARIMA



```
error1 = mean_squared_error(test['CPU usage [%]'], predictions)
error2 = rmse(test['CPU usage [%]'], predictions)
print(f'SARIMA(0,1,0)(2,0,0,12) MSE Error: {error1:11.10}')
print(f'SARIMA(0,1,0)(2,0,0,12) RMSE Error: {error2:11.10}'
```

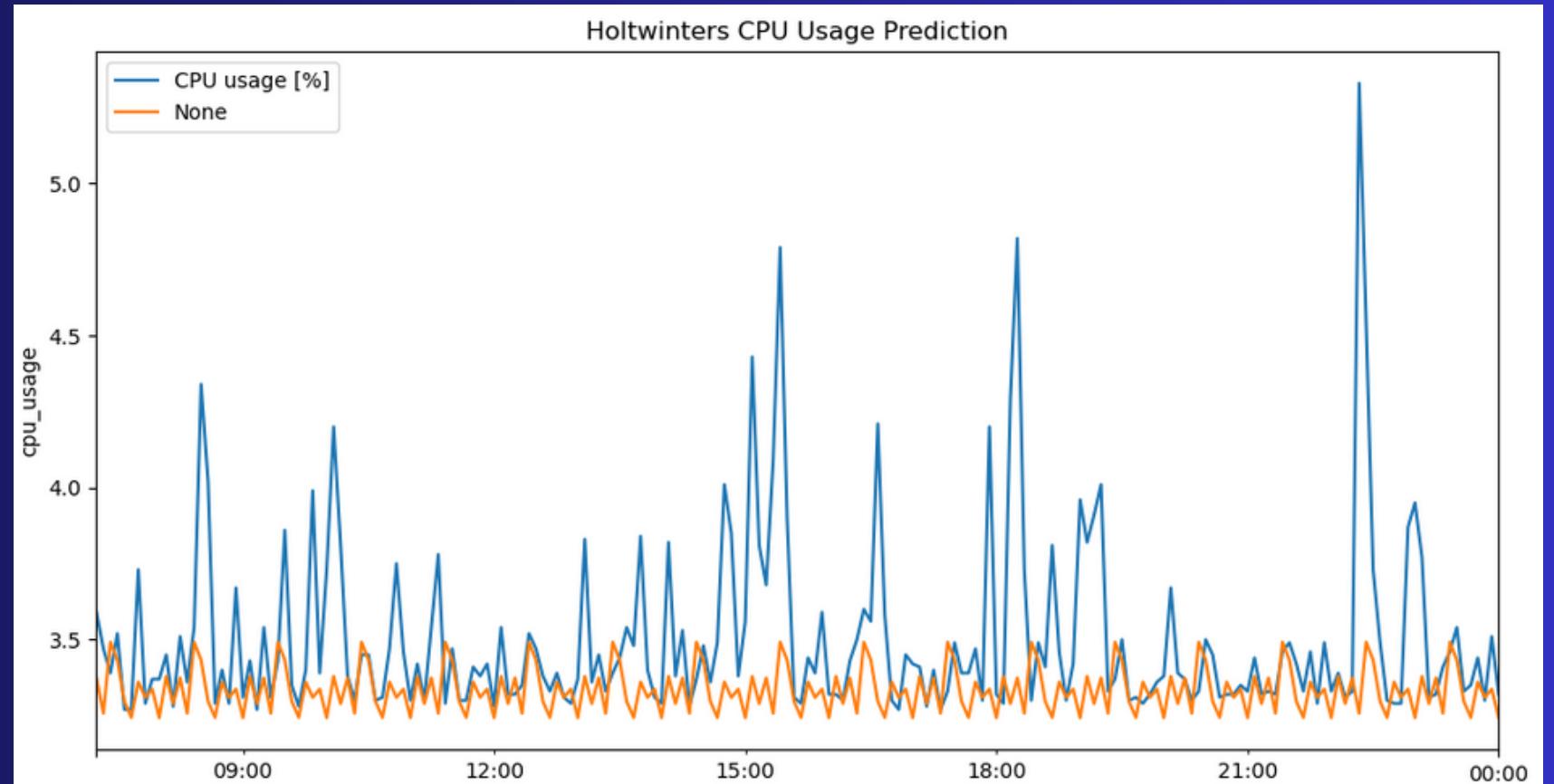
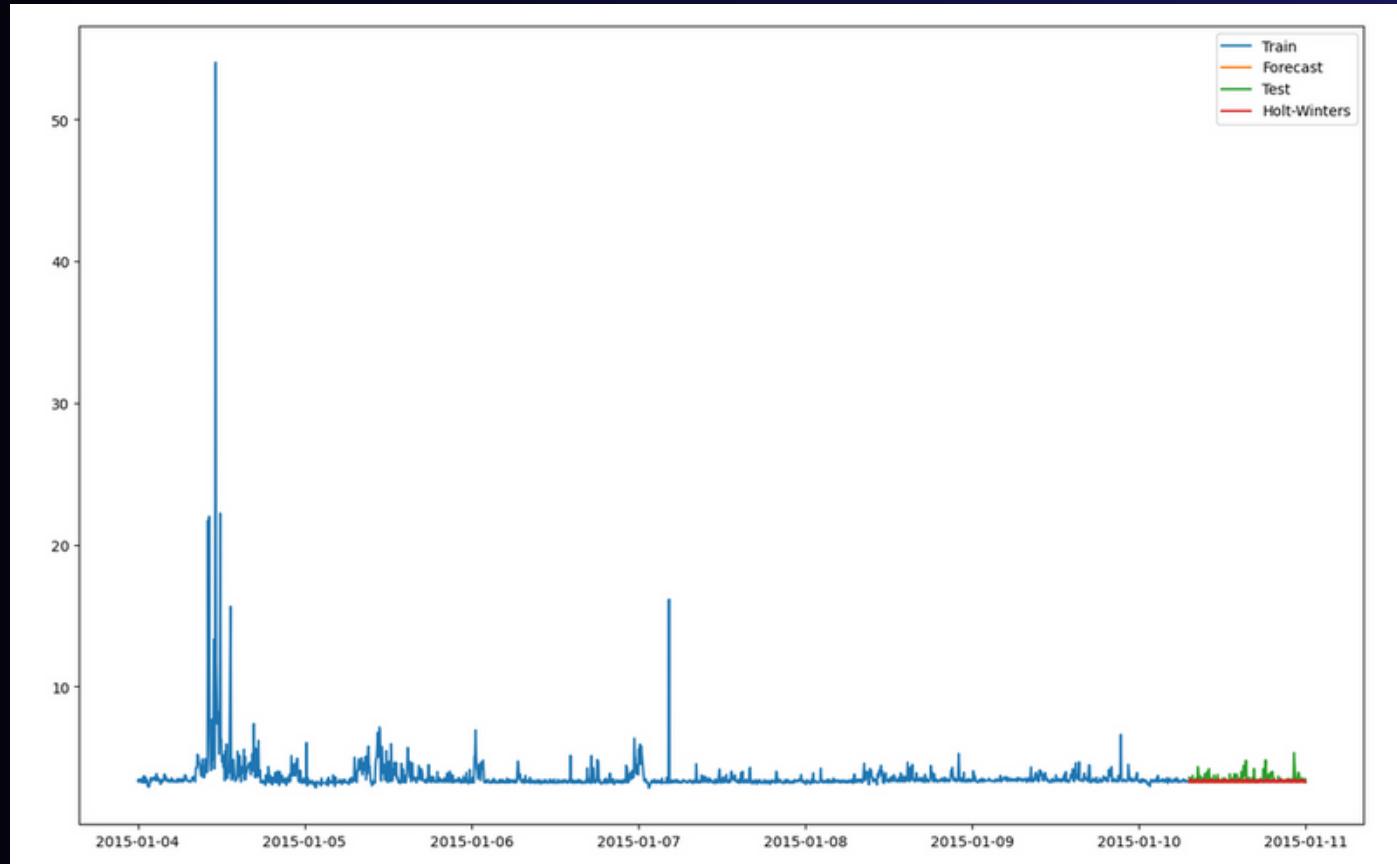
```
SARIMA(0,1,0)(2,0,0,12) MSE Error: 0.2434518722
SARIMA(0,1,0)(2,0,0,12) RMSE Error: 0.4934084233
```

```
from sklearn.metrics import mean_absolute_error
```

```
mae = mean_absolute_error(test['CPU usage [%]'], predictions)
print("MAE:", mae)
```

```
MAE: 0.4643183744944281
```

# HoltWinters



```
rmse = sqrt(mean_squared_error(test, pred))
print('Test RMSE: %.3f' % rmse)
```

Test RMSE: 0.346

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(test, pred)
print("MSE:", mse)
```

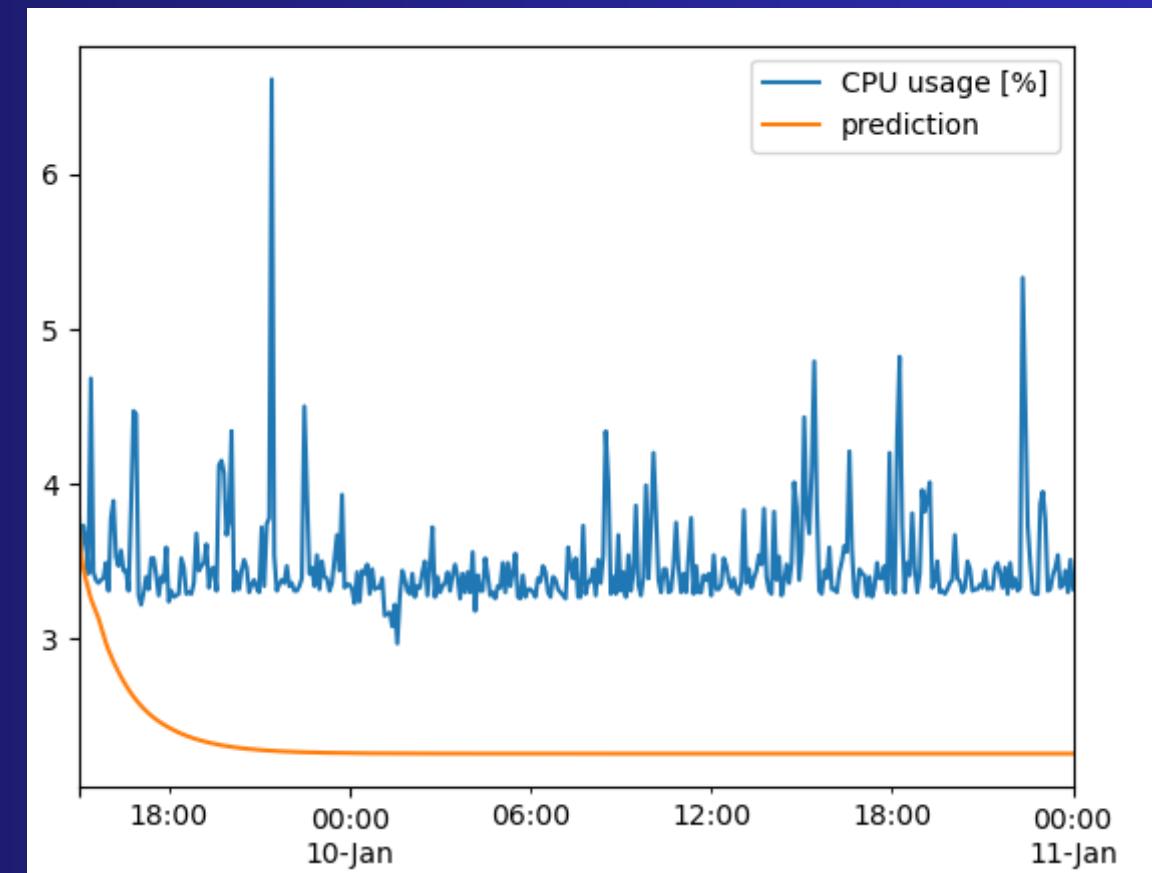
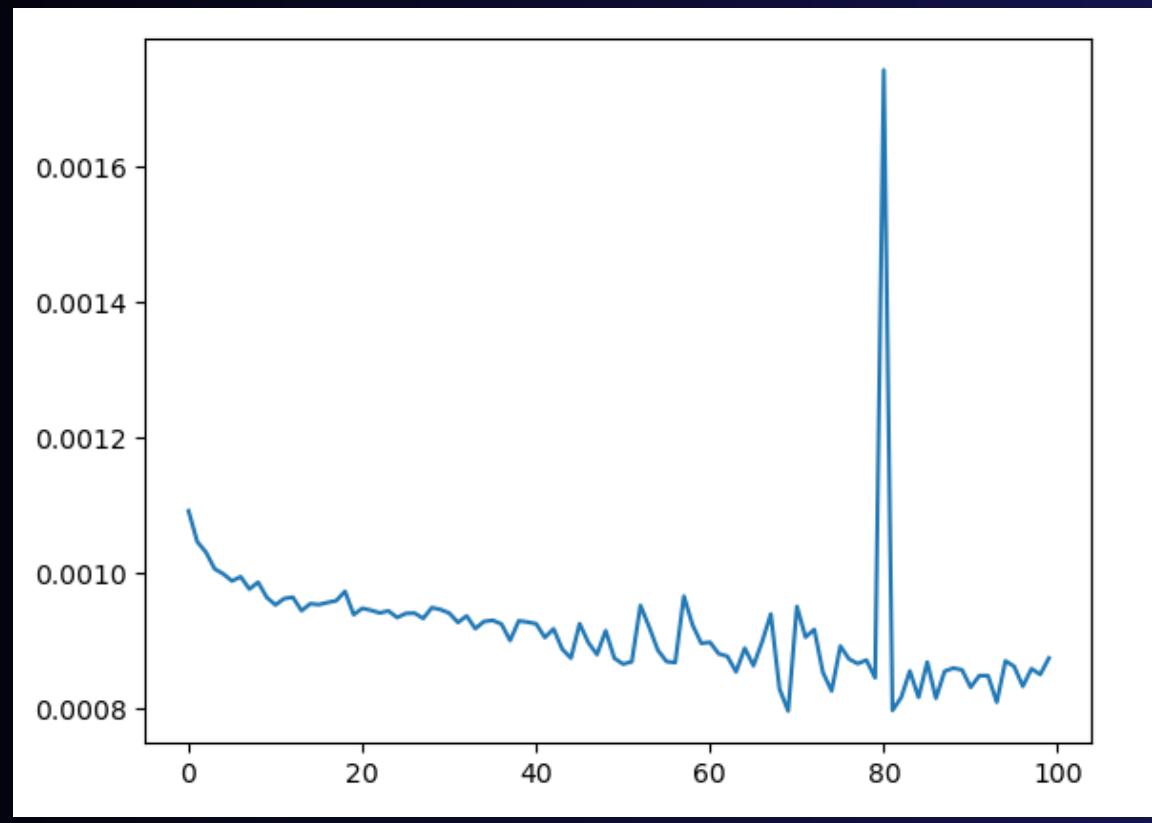
MSE: 0.11959645375081758

```
from sklearn.metrics import mean_absolute_error
```

```
mae = mean_absolute_error(test, pred)
print("MAE:", mae)
```

MAE: 0.19297471975207048

# LSTM



```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse = sqrt(mean_squared_error(test['CPU usage [%]'], test['prediction']))
print("RMSE:", rmse)

RMSE: 1.2144278099337076

mse = mean_squared_error(test['CPU usage [%]'], test['prediction'])
print("MSE:", mse)

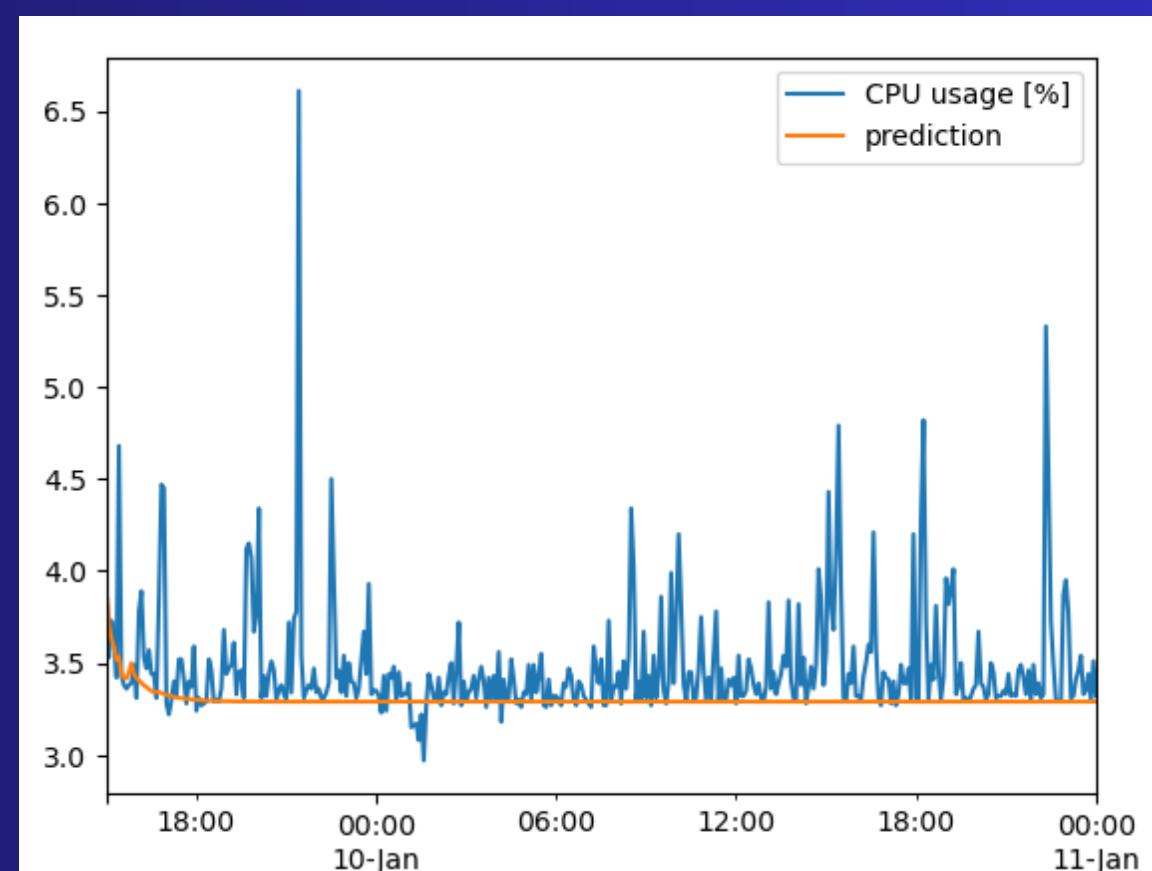
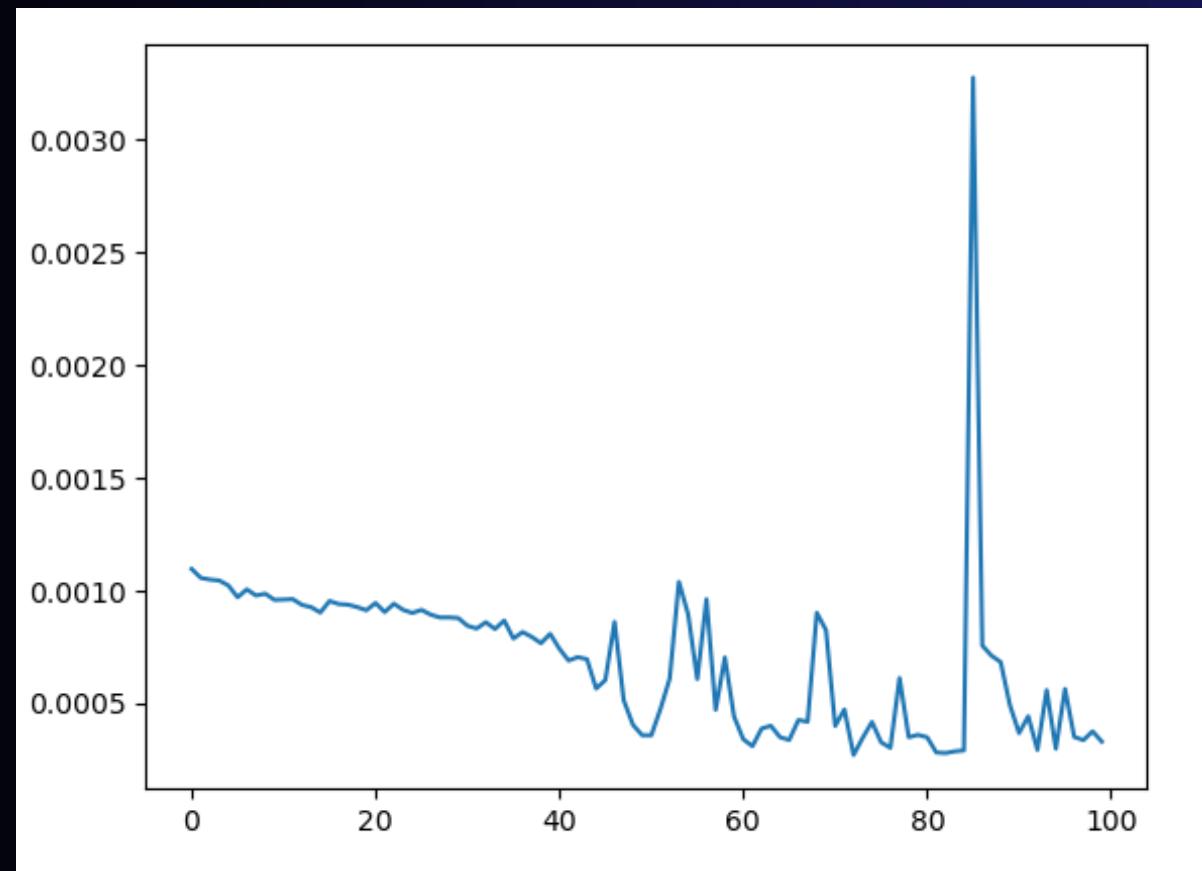
MSE: 1.4748349055403815

from sklearn.metrics import mean_absolute_error

mae = mean_absolute_error(test['CPU usage [%]'], test['prediction'])
print("MAE:", mae)

MAE: 1.160831789017024
```

# Bi-LSTM



```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse = sqrt(mean_squared_error(test['CPU usage [%]'], test['prediction']))
print("RMSE:", rmse)

RMSE: 0.364866943462763

mse = mean_squared_error(test['CPU usage [%]'], test['prediction'])
print("MSE:", mse)

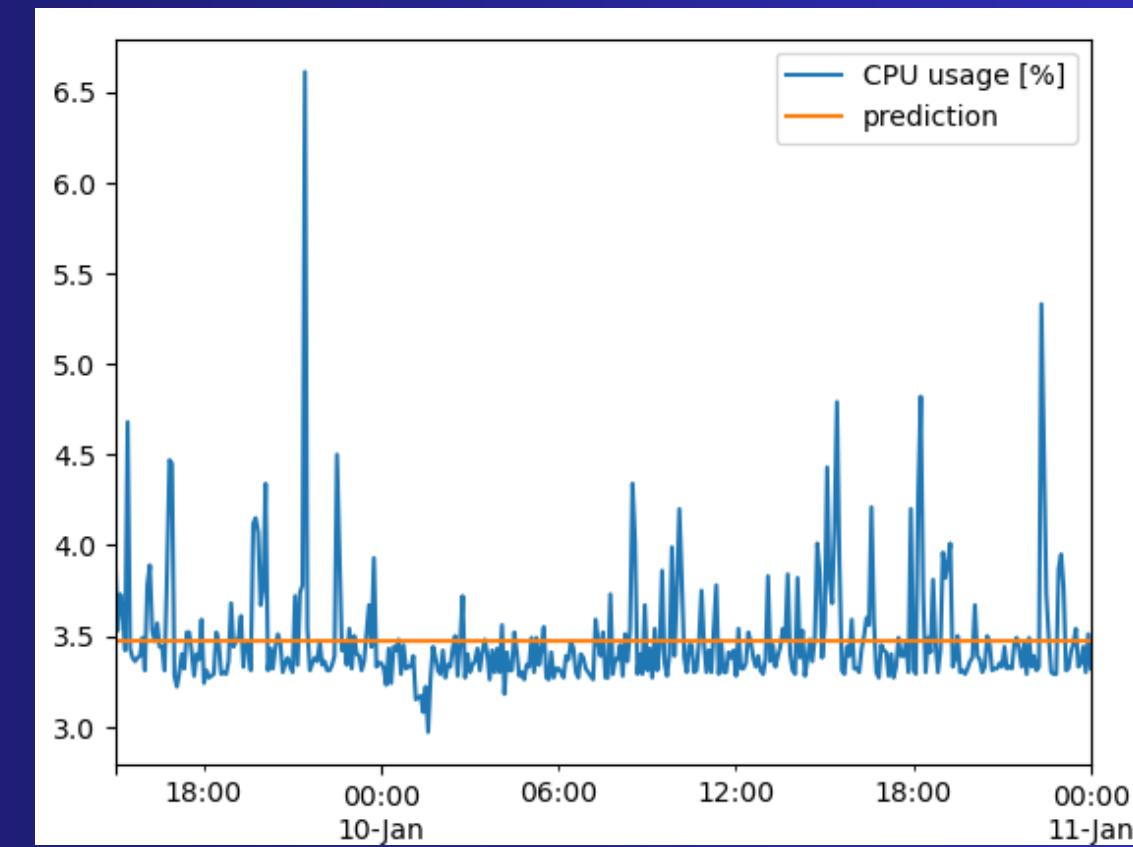
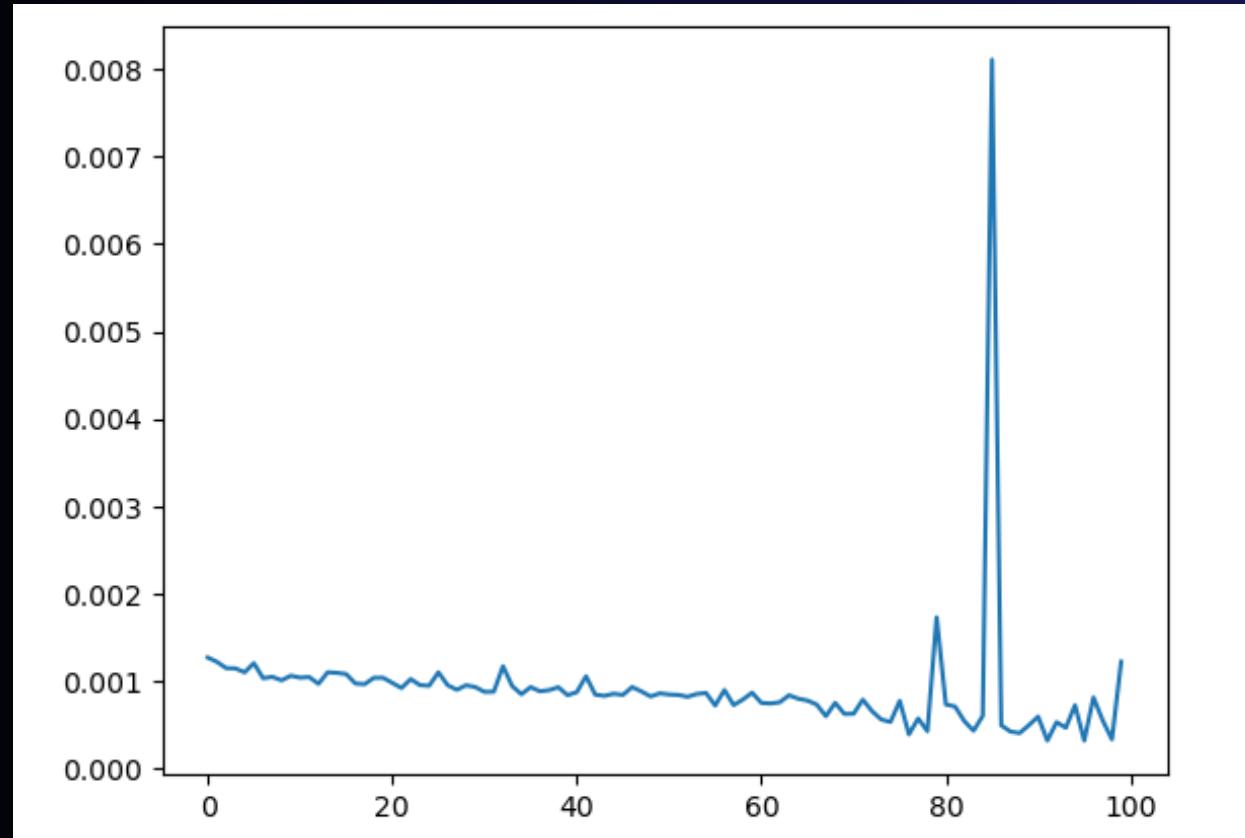
MSE: 0.13312788643185908

from sklearn.metrics import mean_absolute_error

mae = mean_absolute_error(test['CPU usage [%]'], test['prediction'])
print("MAE:", mae)

MAE: 0.194181981280033
```

# Bi-LSTM-CNN



```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse = sqrt(mean_squared_error(test['CPU usage [%]'], test['prediction']))
print("RMSE:", rmse)

RMSE: 0.3187168022722696

mse = mean_squared_error(test['CPU usage [%]'], test['prediction'])
print("MSE:", mse)

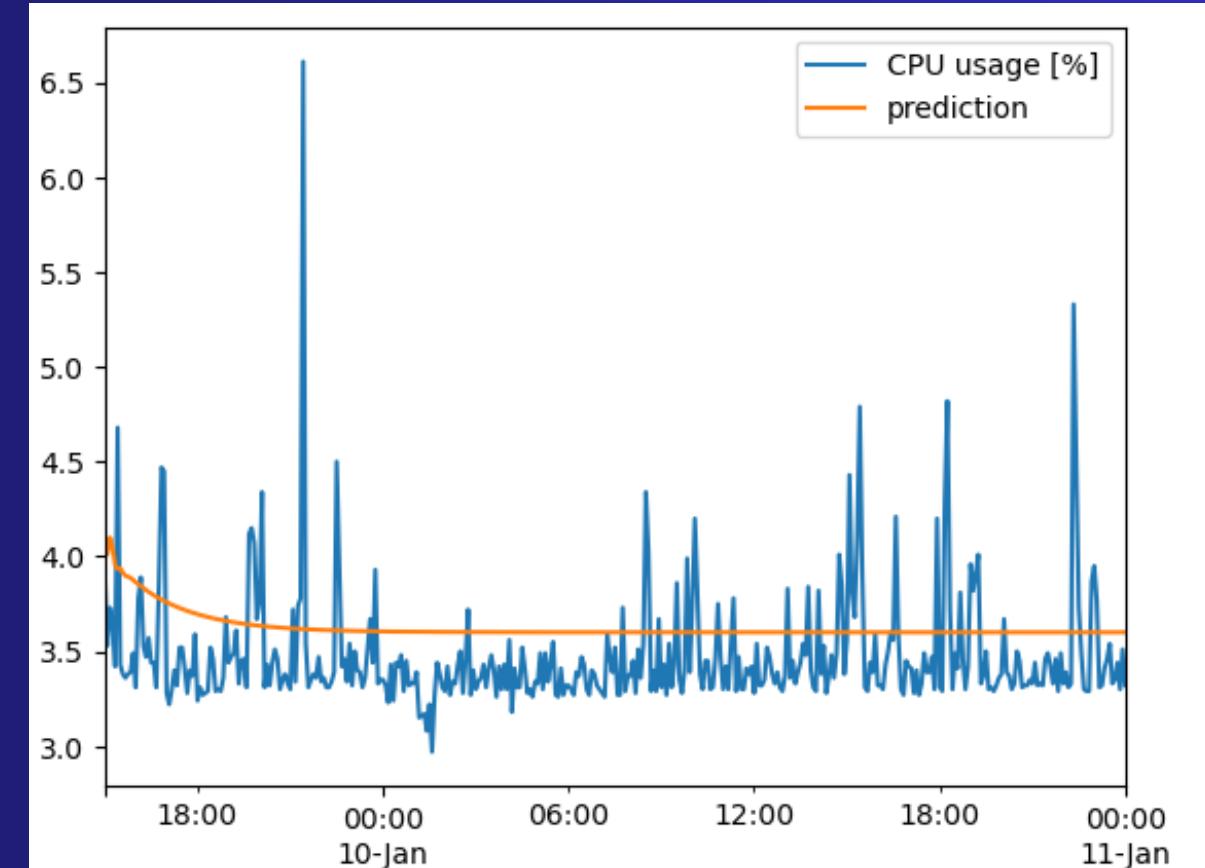
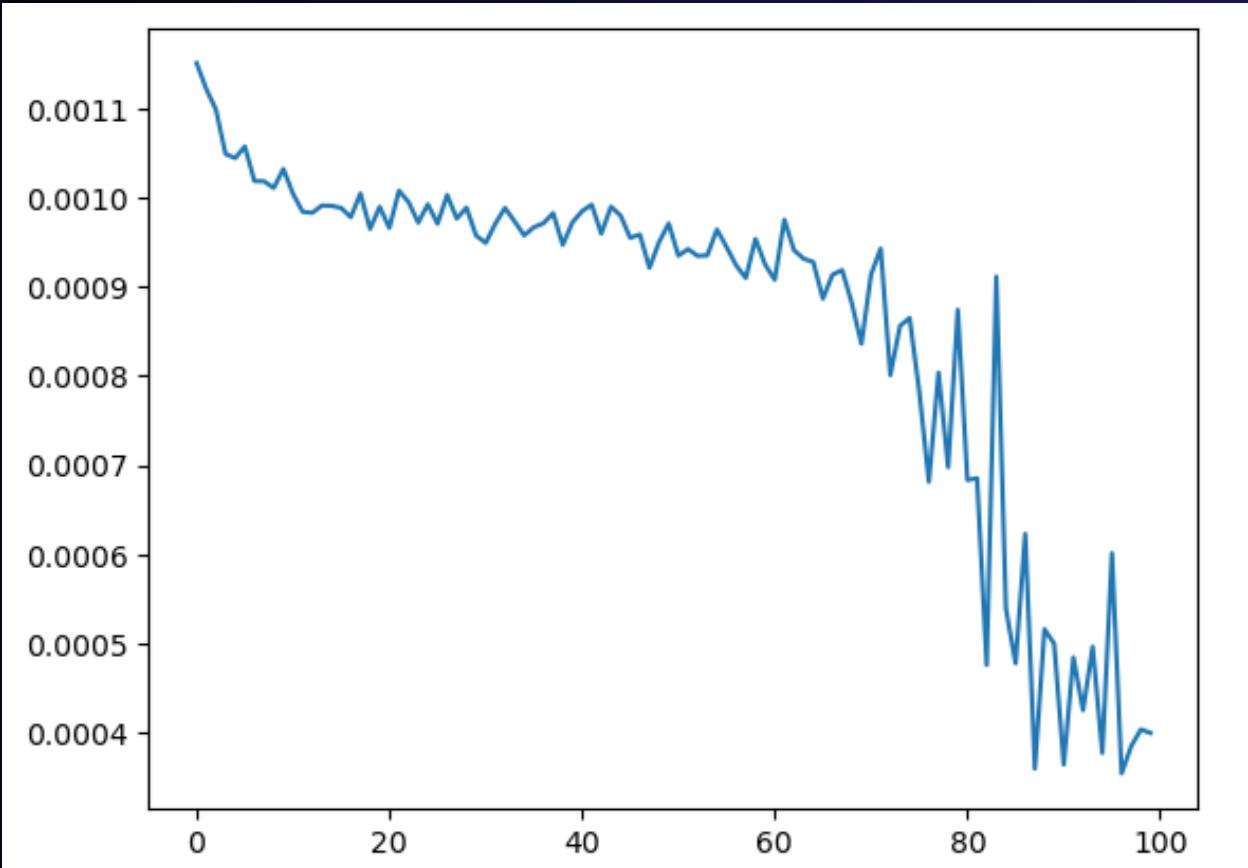
MSE: 0.10158040005066099

from sklearn.metrics import mean_absolute_error

mae = mean_absolute_error(test['CPU usage [%]'], test['prediction'])
print("MAE:", mae)

MAE: 0.1864492814053292
```

# GRU



```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse = sqrt(mean_squared_error(test['CPU usage [%]'], test['prediction']))
print("RMSE:", rmse)

RMSE: 0.35113275503945757

mse = mean_squared_error(test['CPU usage [%]'], test['prediction'])
print("MSE:", mse)

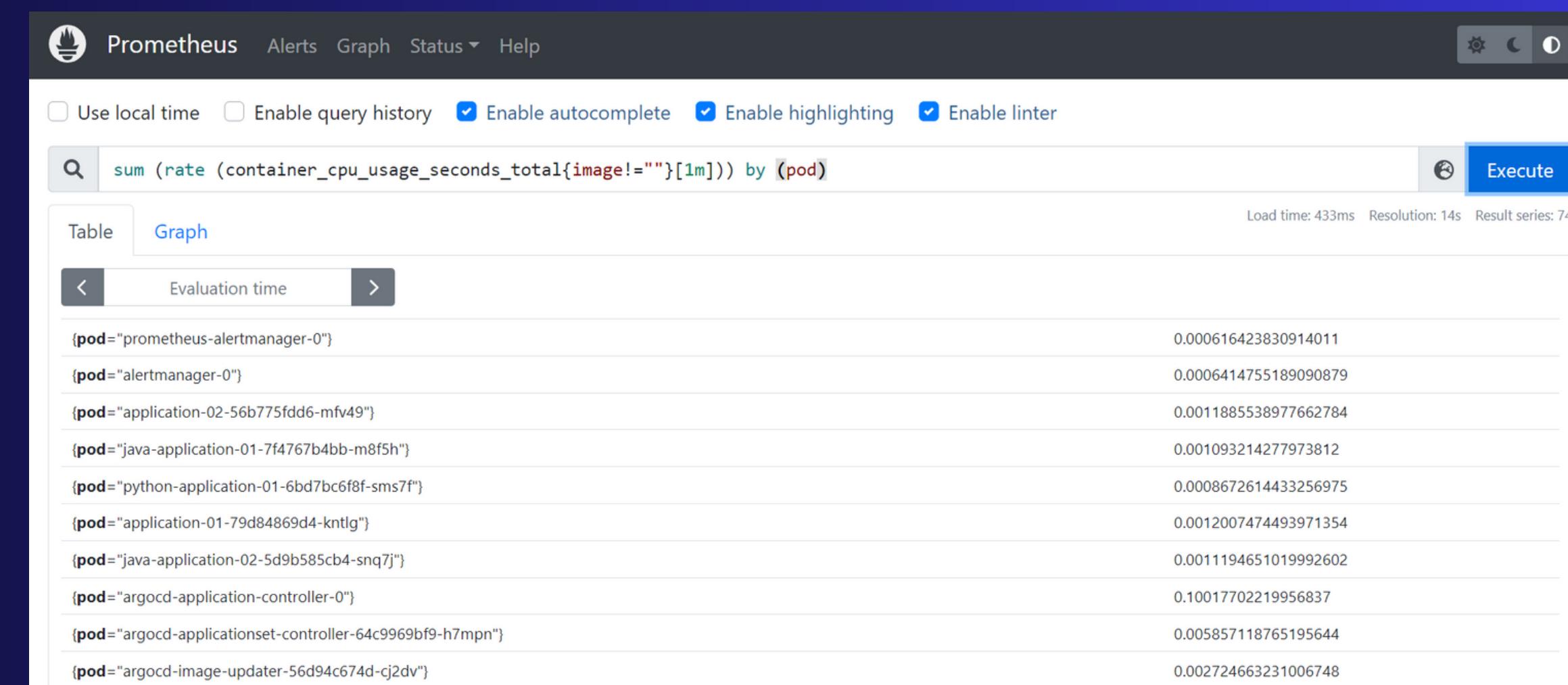
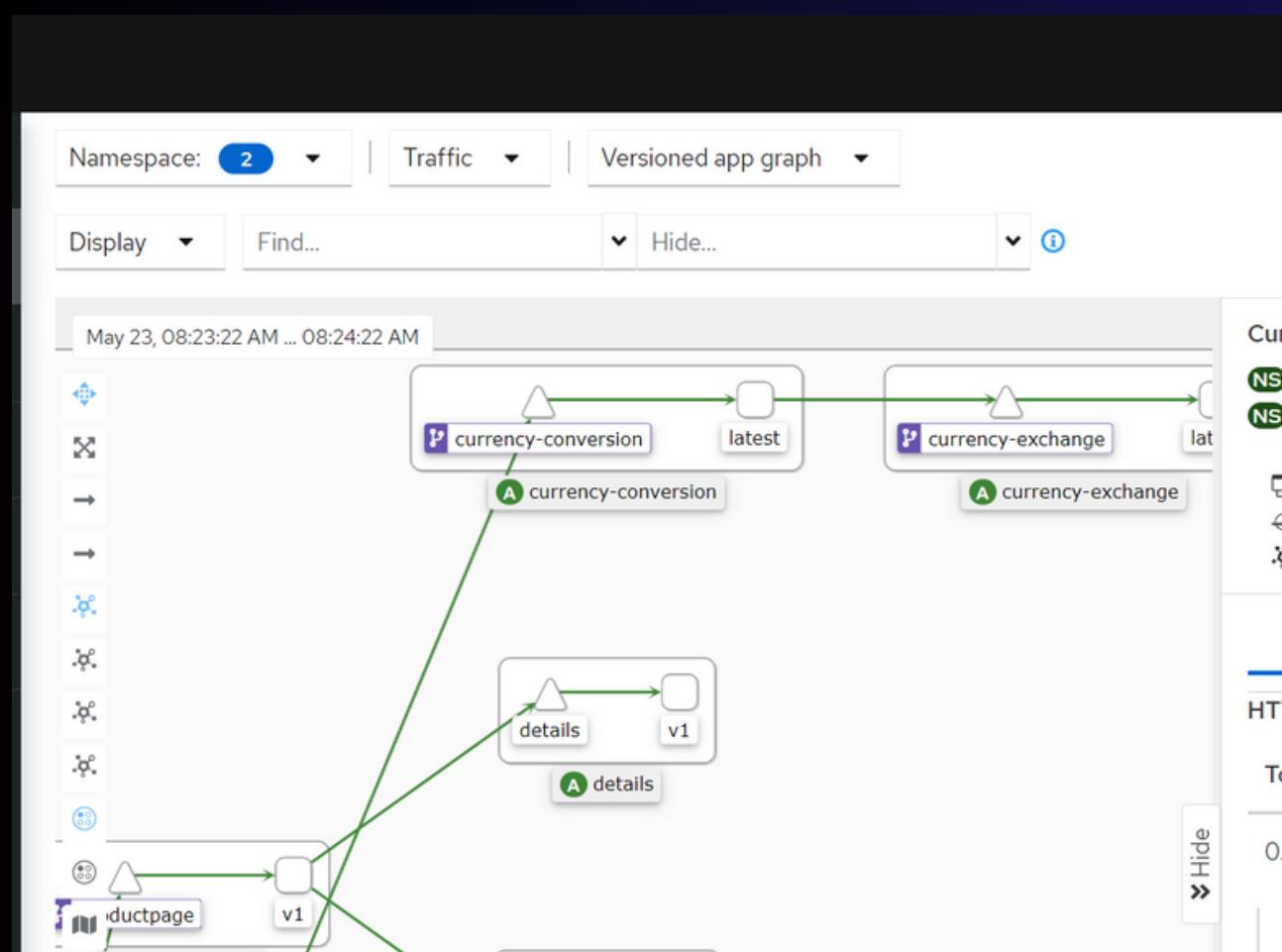
MSE: 0.12329421166159972

from sklearn.metrics import mean_absolute_error

mae = mean_absolute_error(test['CPU usage [%]'], test['prediction'])
print("MAE:", mae)

MAE: 0.27582480294526734
```

# PromQL



# MongoDb

The screenshot shows the MongoDB Cloud interface. On the left, there's a sidebar with sections for CSSE, DEPLOYMENT, Database (with 'PREVIEW' selected), SERVICES, and SECURITY. Under Database, the 'memories' collection is highlighted. The main panel displays the 'test.memories' collection details: STORAGE SIZE: 68KB, LOGICAL DATA SIZE: 283.94KB, TOTAL DOCUMENTS: 204, and INDEXES TOTAL SIZE: 24KB. Below this, there are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, Search Indexes, and Charts. A search bar at the top says 'Type a query: { field: 'value' }'. The results pane shows a list of document snippets, starting with:

```
_id: ObjectId('646c4eabca1847c49a772958')
__name__: "container_memory_usage_bytes"
beta_kubernetes_io_arch: "amd64"
beta_kubernetes_io_instance_type: "e2-medium"
beta_kubernetes_io_os: "linux"
cloud_google_com_gke_boot_disk: "pd-balanced"
cloud_google_com_gke_container_runtime: "containerd"
cloud_google_com_gke_cpu_scaling_level: "2"
cloud_google_com_gke_logging_variant: "DEFAULT"
cloud_google_com_gke_max_pods_per_node: "110"
cloud_google_com_gke_nodepool: "node-pool-01"
cloud_google_com_gke_os_distribution: "cos"
cloud_google_com_gke_provisioning: "standard"
```

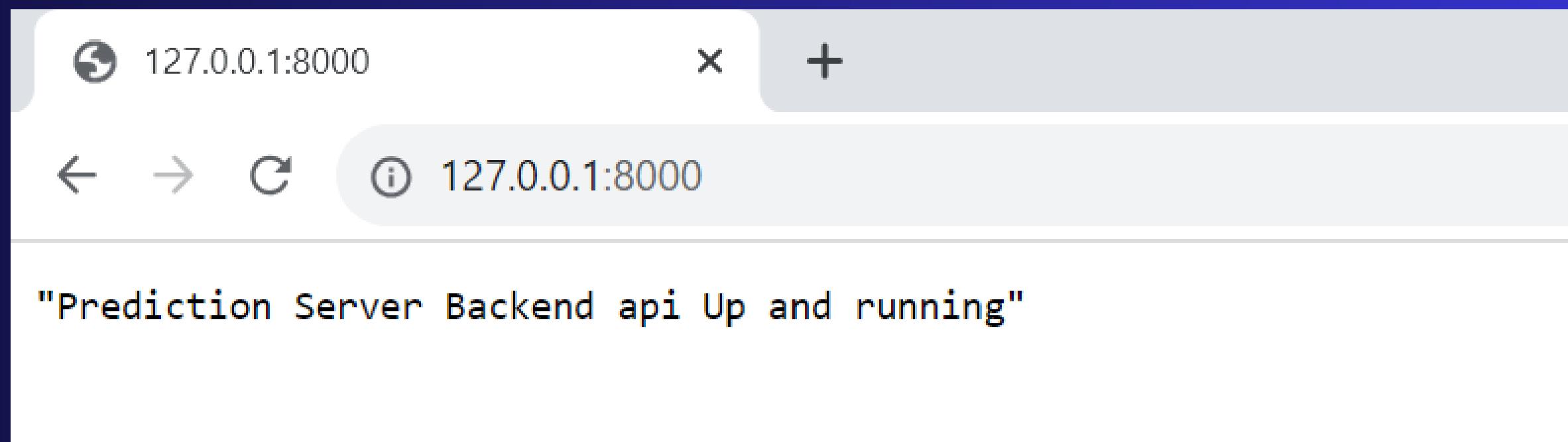
At the bottom, it says '1-20 of many results'.

# Prediction Server

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\fahamed\Documents\fastapi> uvicorn server:app --reload
←[32mINFO←[0m:     Will watch for changes in these directories: ['C:\\\\Users\\\\fahamed\\\\Documents\\\\fastapi']
←[32mINFO←[0m:     Uvicorn running on ←[1mhttp://127.0.0.1:8000←[0m (Press CTRL+C to quit)
←[32mINFO←[0m:     Started reloader process [←[36m←[1m23800←[0m] using ←[36m←[1mWatchFiles←[0m
←[32mINFO←[0m:     Started server process [←[36m25352←[0m]
←[32mINFO←[0m:     Waiting for application startup.
←[32mINFO←[0m:     Application startup complete.
```



# Commercialization

Pricing Table		Free	Silver	Gold	Platinum	Custom
Choose Your Best Plan		\$0	\$9.99	\$15.99	\$34.99	TBD
Fully Automated CI Architecture		✓	✓	✓	✓	
Integrated Monitoring Tool		✗	✓	✓	✓	
Automate Resource Allocation		✗	✗	✓	✓	
Multi Cloud Deployment		✗	✗	✗	✓	
Fully Automated and Optimised CD/CD Architecture		✗	✗	✗	✓	Contact to Discuss with ReleaseX

# THANK YOU

23-193