

Team B6 Chimera

Aihemaiti (gulsum) - UI design



Sidebar Menu

- Added the sidebar menu
- Organized the page into sections: Overview, Metrics, Documentation, and Team Members

-

Dez

Reformat website

Darkmode

Implement sliding scroll

Emails

- meeting setup

Chimera Performance Metrics

[Overview](#) [Metrics](#) [Documentation](#)

Project Overview

This project is about collecting and showing Chimera user's friendly dashboard.

Metrics

This is where all the metrics will be on the website

Documentation

This is where all documentation will go including testing

Stephanie H

- Prometheus documentation
 - Node-exporter, expose endpoint on Chimera cluster
 - <https://notes.kodekloud.com/docs/Kubernetes-and-Cloud-Native-Associate-KCNA/Cloud-Native-Observability/Prometheus-Node-Exporter/page>
 - Tested node_exporter on disk2/test-step
 - Run: ./node_exporter on default port 9100.
- OPTIMIZATION: Potential docker image?
Potential system file for the node-exporter.
- Set up prometheus onto babbage
 - Tested on @babbage:/disk2/test-step
 - Run ./prometheus on default port 9090.
 - Server works but cannot access through browser.
 - Tested locally and works, port can be opened to prometheus dashboard.

```
drwxr-xr-x  8 stepnie ugrad 4096 Mar 11 22:18 .
drwxrwxrwx 21 root    root  4096 Mar 10 22:08 ..
drwxr-xr-x  2 stepnie ugrad 4096 Mar 10 15:10 bin
drwxr-xr-x  3 stepnie ugrad 4096 Mar 10 15:35 chimera-dashboard
drwxr-xr-x  2 stepnie ugrad 4096 Mar 10 15:10 include
drwxr-xr-x  3 stepnie ugrad 4096 Mar 10 15:10 lib
lrwxrwxrwx  1 stepnie ugrad   3 Mar 10 15:10 lib64 -> lib
drwxr-xr-x  2 stepnie ugrad 4096 Oct 25 16:10 node_exporter-1.10.2.linux-amd64
drwxr-xr-x  2 stepnie ugrad 4096 Feb 25 16:46 prometheus-3.10.0.linux-amd64
-rw-r--r--  1 stepnie ugrad   71 Mar 10 15:10 pyenv.cfg
stepnie@babbage:/disk2/test-step$
```

```
time=2026-03-11T22:09:05.079-04:00 level=INFO source=node_exporter.go:141 msg=xfs
time=2026-03-11T22:09:05.079-04:00 level=INFO source=node_exporter.go:141 msg=zfs
time=2026-03-11T22:09:05.080-04:00 level=INFO source=tls_config.go:346 msg="Listening on" address=[:]:9100
time=2026-03-11T22:09:05.080-04:00 level=INFO source=tls_config.go:349 msg="TLS is disabled." http2=false address=[:]:9100
```

```
stepnie@babbage:/disk2/test-step/prometheus-3.10.0.linux-amd64$ curl http://158.121.95.41:9090/
<a href="/query">Found</a>.
```

Curl

<http://localhost:9100/metrics>

Dhruv.D - basic website

- Work on the basic overview.
- Set basic requirements for the project.

Chimera Performance Metrics

Project Overview

This project is about collecting and showing Chimera clustr which should be user-friendly.

Team Goals

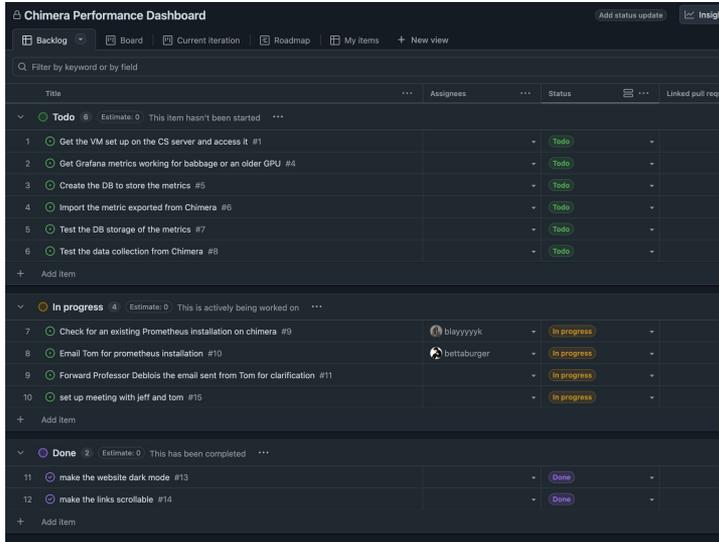
- Set up the VM
- Use Grafana
- Collect CPU and GPU data
- Store data e

Coming Soon

Live dashboard stay tuned.

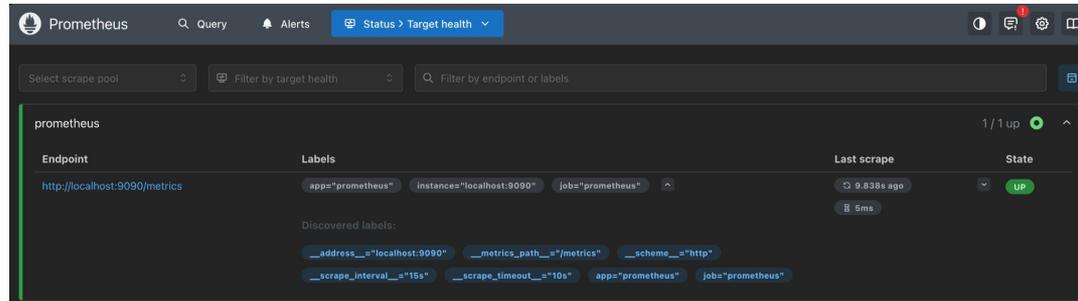
Blake M

- Set up Github Project page and backlog tickets
- Basic set up of Prometheus web service



The screenshot shows a Jira backlog for a project named 'Chimera Performance Dashboard'. The backlog is organized into three columns: 'Todo', 'In progress', and 'Done'. Each item in the backlog includes a title, an estimate, and a status. The 'Todo' column contains 8 items, the 'In progress' column contains 4 items, and the 'Done' column contains 2 items. The 'In progress' items are assigned to users 'blayyyyyk' and 'botaburger'.

Category	Item #	Title	Assignee	Status	
Todo	1	Get the VM set up on the CS server and access it		Todo	
	2	Get Grafana metrics working for baggage or an older GPU		Todo	
	3	Create the DB to store the metrics		Todo	
	4	Import the metric exported from Chimera		Todo	
	5	Test the DB storage of the metrics		Todo	
	6	Test the data collection from Chimera		Todo	
	In progress	7	Check for an existing Prometheus installation on chimera	blayyyyyk	In progress
		8	Email Tom for prometheus installation	botaburger	In progress
9		Forward Professor Deblols the email sent from Tom for clarification		In progress	
10		set up meeting with jeff and tom		In progress	
Done	11	make the website dark mode		Done	
	12	make the links scrollable		Done	



The screenshot shows the Prometheus web interface. The top navigation bar includes 'Prometheus', 'Query', 'Alerts', and 'Status > Target health'. The main content area displays a table of scrape targets for the 'prometheus' job. The table has columns for 'Endpoint', 'Labels', 'Last scrape', and 'State'. The 'Endpoint' column shows 'http://localhost:9090/metrics'. The 'Labels' column shows 'app="prometheus"', 'instance="localhost:9090"', and 'job="prometheus"'. The 'Last scrape' column shows '9.838s ago' and '5ms'. The 'State' column shows 'UP'.

Endpoint	Labels	Last scrape	State
http://localhost:9090/metrics	app="prometheus" instance="localhost:9090" job="prometheus"	9.838s ago 5ms	UP