



Running prombench on EKS and update dashboards

Personal Information



Name: Patel Drumil
Github: [drumilpatel2000](https://github.com/drumilpatel2000)
Email: drumilpatel720@gmail.com
Phone No: +91-6351173421
Location: India 
Time Zone: India (UTC +5:30)
Riot: Flipbyte
LinkedIn: [Drumil Patel](#)

University Information



University: [Indian Institute of Technology, Roorkee](#)
Majors: Electrical Engineering
Current: IInd Year (batch 2022)
Degree: Bachelor of Technology (4 Year Program)

Contact and Working hours



Reachable anytime through **email**, **riot** or **contact number**

Typical working hours :-

- UTC 0430 - 0730 hrs (IST 1000 - 1300 hrs)
- UTC 0930 - 1230 hrs (IST 1500 - 1800 hrs)
- UTC 1530 - 2030 hrs (IST 2100 - 0200 hrs)

Coding Skills



Programming Languages:

- Proficient **PHP**, **Python**, **Go**, **Javascript**
- Intermediate Knowledge in **C++**, **Java**, **Typescript**
- Worked with **Django**, **React**, **Laravel**, **Yii 2.0**, **Symfony**, **Figma**
- Databases - **MySQL**, **PostgreSQL**, **MongoDB**
- Devops - **Prometheus**, **Kubernetes**, **Grafana**

Experience



Full Stack Web Developer and Devops Engineer, currently Hub-Coordinator at Information Management Group ([IMG](#)), IIT Roorkee. During my first year at college my workspace was confined to web development, during which I was selected for GSoC 2019 to work on [Automating configuration for Assignment uploads](#) project in [Submittity](#), in my second year my workspace expanded to Cloud Computing and some major projects listed in the following section.

- **CMS**
- **Forminator**
- **Lectures and Tutorials Portal(LecTut):**
- **Cortago**

Contributions



- **Prometheus**

- Prometheus([2 Commits](#))

- ❖ [6762](#) :- Formatting short tables for readability(**Merged**)
 - ❖ [6761](#) :- Add conditional rendering of Navlink for Consoles(**Merged**)

- Pushgateway([22 Commits](#))

- ❖ [327](#) :- APIs for metadata of metrics(**Merged**)
 - ❖ [328](#) :- New Revamped UI in React(**Closed**)
 - ❖ [329](#) :- Change httprouter to common/route(**Closed**)
 - ❖ [332](#) :- Add test for APIs(**Merged**)
 - ❖ [334](#) :- API documentation for v1(**Merged**)

- **Prometheus-community**

- Promql Langserver([3 Commits](#))

- ❖ [122](#) :- Add Support for new metadata api for hover(**Merged**)
 - ❖ [125](#) :- Use metadata api for completion(**Merged**)
 - ❖ [136](#) :- Use documented tpe for promql type on hover(**Merged**)

- **Submitty (GSoC 2019)**

- Submitty([35 Commits](#))

- ❖ **Automating Configuration For Assignments Uploads** is an upgrade to old assignment configuration. Previously all inputs and outputs file was to be provided at time of configuration of assignment this made input and output constant for every submission. After this project we can add random inputs and corresponding output at the time of submission. This is possible by providing an instructor solution and python script which can generate random inputs. We can also generate only outputs from predefined inputs at the time of building gradable. This can be used to remove dependencies on predefined output.
 - ❖ The main PRs which sumifies my whole GSoC period are:-
 - ❖ [4335](#) :- [Feature:Autograding] solution to generate expected output
 - ❖ [4280](#) :- [BUGFIX:Tests] Validate solution_containers in complete_config.json
 - ❖ [3882](#) :- [Feature:Autograding] Randomized input and generated output
 - ❖ [3744](#) :- new test_output and solution in TMP_WORK
 - ❖ Other PRs can be viewed [here](#).

- Submitty.github.io([2 Commits](#))

- ❖ [162](#) :- [GSoC] 2019 report from Drumil
 - ❖ [85](#) :- Incorrect Directory for autograding_workers.json
-

Project Overview



Prombench

Prombench is the automated E2E testing and benchmarking for prometheus. Prombench uses Kubernetes to benchmark promethues using various scalable configurations of fake web servers. Prombench uses Prometheus, Grafana and Loki for monitoring difference between versions.

Presently Google Kubernetes can only be used for prombench. Prombench can also run in different platforms such as :-

1. (AWS)Amazon web services
2. local machine

Deliverables



Update Grafana Dashboards[#328]

Monitoring changes is one of the main aims behind prombench. For this purpose prombench uses following tools :-

1. Prometheus (Meta)
2. Grafana Dashboard
3. Loki from Grafana

Presently, Grafana Dashboard consist of many useful graphs and useful visualization but there is a scope of improvement. Various issues which aims to improve visualizations for grafana are as follows :-

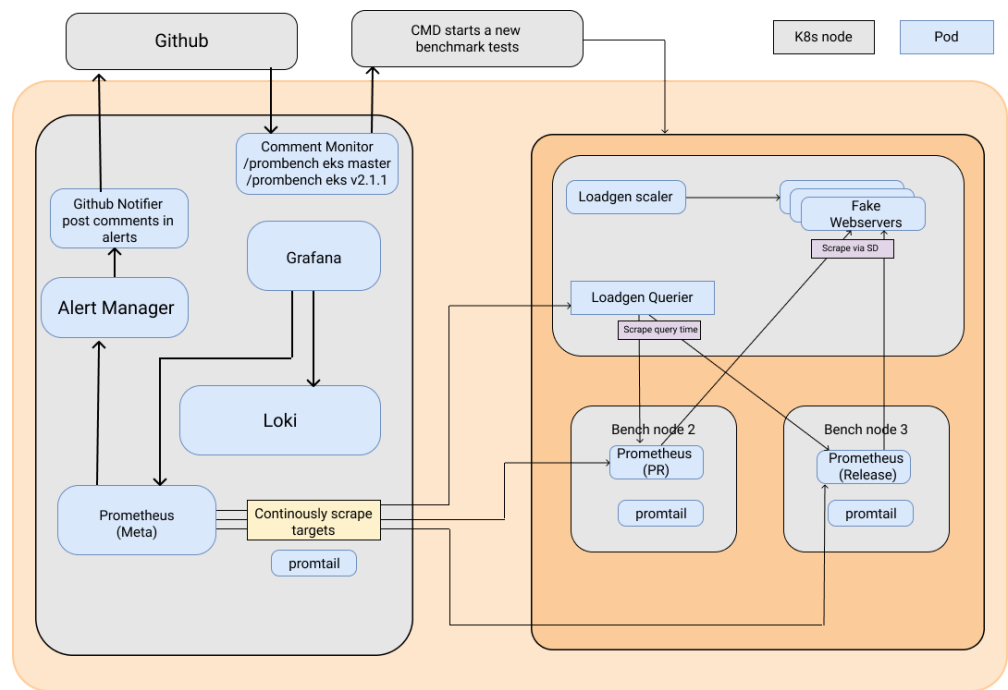
- **Refactor the grafana dashboards to show a more consistent memory usage(#324)**
After the release of Prometheus version 2.16 the RSS and allocations shows differences. This can be solved by treating all evictable memory and non evictable memory differently.
As mentioned in [this comment](#), we also need support for choosing irate and rate and rate intervals. This can be easily achieved using support for [variables](#) in grafana dashboard
- **Revist all current dashboards to add new and remove unneeded to make them more useful for catching bugs(#186)**
Add more panels to enhance visualization. Requirements of new panels should be discussed with maintainers.
- **Node exported does not get filesystem size metrics from SSD path /mnt/disks/ssd0(#159)**
- **Add explanations to all panels on the Prombench dashboard(#305)**
Enhance grouping of panels in dashboard and add explanation to explain correlation between panels
- **Loki not showing previous test prometheus instance logs(#322)**
Dive more into problem of loki not showing previous prombench logs.

Prombench on AWS(Amazon Web Service)[#341]

AWS is one of the most used cloud providers. While kubernetes was not natively supported by AWS, Amazon EKS(Elastic Kubernetes Service) was added later. EKS is most trusted way to serve kubernetes in amazon as it has a highly available control plane which removes a single point of failure.

EKS has its own [SDK](#) written in golang similar to [GKE SDK](#) which makes its implementation more streamlined in prombench. EKS will use the same configuration as GKE uses but the way to create clusters will be somewhat different due to differences in their SDKs.

Cluster Configuration



Cluster will consist of following nodes :-

1. Main Node
2. Scaler Node
3. Present Prometheus Configured Node
4. Released Prometheus Configured Node

Main node

The main node consist of all tools required to monitor benchmarking prometheus. It consist of :-

1. Alert Manager
2. Prometheus(Meta)
3. Grafana
4. Lok
5. Github Notifier
6. Comment Monitor

Scaler Node

The Scaler Node is main node which does all the work of benchmarking via different configuration of fake web servers and loadgen Querier.

More Generic Prombench

With possibility of prombench can be configured using both the cloud server. It becomes vital to be prombench more generic and make switching between cloud servers more seamless. A brief discussion is needed to achieve the same. This include following steps :-

1. **Finding Equivalents of machines in both EKS and GKE:-**

As discussed, prombench presently works on GKE. which have configurable machines depending memory and storage like

1. **n1-highmem-8**
2. **n1-highcpu-16**, etc.

We should find eks equivalent for such machine configurations

Additional Issues

As per now, there are plans to complete but I have not got into details of respective issues but love to complete in GSoC or post GSoC.

1. **Add additional rules or queries to catch regressions**
2. **Extend Prombench for scalability tests**
3. **When a prombench apply commands fails display the last error**

Project Timeline



Community Bonding	
4th May - 31st May	<ul style="list-style-type: none">- Get to know the Community more, and bond with mentors, admins and developers.- Get feedback if something in the project needs amendment.- Learn more about EKS and it's SDK
Phase - 1	
Week 1 - 2 1st June - 14th June	Implement Provider for EKS cluster Implement provider for EKS cluster similar to gke.go and k8s.go which will be helpful to provide direct interaction via CLI
Week 3 15th June - 21st June	Add Support for resource create and delete Add function for resource create and delete in EKS provider for adding support for prometheus(meta), grafana, loki, etc.
Week 4 22th June - 28th June	Add support for node creation and deletion in EKS Add function to add nodes and deletion of nodes in EKS for benchmarking prometheus (similar to nodepool create and delete)
Phase - 2	
Week 5 29th June - 5th July	Add Documentation for EKS Add documentation for creating adding resources and running benchmarks in EKS cluster
W-eek 6 6th July - 12th July	More Generic Prombench With possibility of prombench can be configured using both the cloud server. It becomes vital to be prombench more generic and make switching between cloud server more seamless
Week 7 13th July - 19th July	Refactor the grafana dashboards to show a more consistent memory usage. (#324)

Week 8 20th July - 26th July	Node exporter does not get filesystem size metrics from SSD path /mnt/disks/ssd0(#159)
Phase - 3	
Week 9 27th July - 2nd Aug	Revisit all current dashboards to add new and remove unneeded to make them more useful for catching bugs (#186)
Week 10 3rd Aug - 9th Aug	Add explanations to all panels on the prombench dashboard (#305)
Week 11 10th Aug - 16th Aug	Loki not showing previous test prometheus instance logs(#322)
Week 12 17th Aug - 23th Aug	1. Fix bugs and polish new features 2. Fulfil all remaining backlogs
Final Evaluation	

Activity Deliverables



-
- Write blog posts on medium every alternate week.
 - Write weekly report of accomplished tasks on the mailing list (or something more relevant)
 - Prepare and submit a presentation for this project.

Motivation



Google Summer of Code is an excellent platform to get acquainted with the open source community and their skillful mentors. It gives one a professional work experience in their college years where they collaboratively build a product for the welfare of the society. In this process, both the individual and the community improve and grow.

A couple of months have passed since I have been contributing to the Prometheus organisation, I have had a phenomenal learning experience throughout my journey with prometheus, one specific point that really makes me fond about this organisation is the way they appreciate their contributors for their work.

Resources



-
- [CNCF SOC Page](#)
 - [Test-Infra Repo](#)
 - [Google Kubernetes Engine SDK](#)
 - [Amazon Elastic Kubernetes Service SDK](#)
 - [Figma](#) for All Diagram Illustrations