Observability For ONAP

Amar Kapadia, Aarna Networks

https://www.linkedin.com/in/amarkapadia/

Agenda

- Metrics
  - Deployment Architecture
  - Prometheus configuration and service discovery
  - Grafana Dashboards
  - Persistent Storage TSDB
  - Alert Notification
- Logging
  - Deployment Architecture
  - Fluentd deployment and configuration
  - Kibana Dashboard
  - Alerts and Events

Subset of ONAP Projects Used

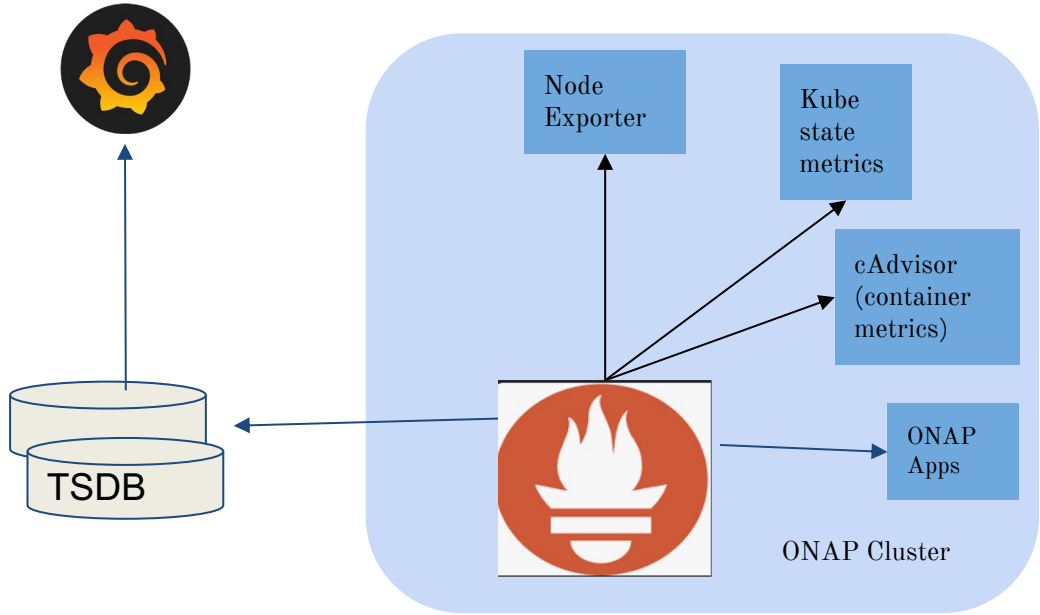
For this setup and demo we used the below ONAP projects.

- CDS
- Camunda
- Mariadb

Metrics

Metrics - Deployment Architecture

- Prometheus deployed in ONAP cluster to scrape:
  - Infrastructure Metrics
    - Node Exporter for node level metrics
    - cAdvisor for container level metrics
  - Application level Metrics
- Prometheus remote write to Distributed TSDB
- Grafana queries TSDB for plotting dashboards

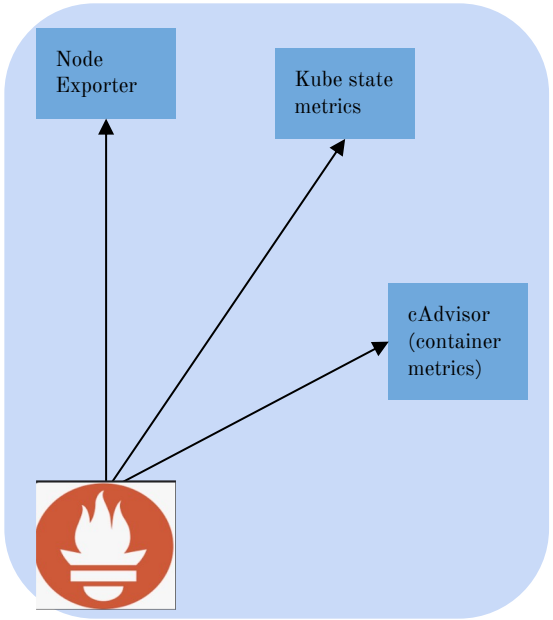


Scrape Config and Service Discovery

- Deploy Node exporter, kube-state-metric server from AMCOP (Aarna's ONAP distribution)
- Edit Prometheus server configmap and add scrape job

```
- job_name: 'node-exporter'
kubernetes_sd_configs:
  - role: endpoints
relabel_configs:
- source_labels: [__meta_kubernetes_endpoints_name]
  regex: 'node-exporter'
  action: keep
- job_name: 'kube-state-metrics'
static_configs:
  - targets: ['kube-state-metrics.kube-system.svc.cluster.local:8080']
```

- This should make Prometheus start scraping the following:
  - CPU, Memory and Network stats of the Node
  - CPU, Memory and Network stats of all containers
  - Kubernetes resources metrics i.e number of deployment, pods, statefulsets etc.



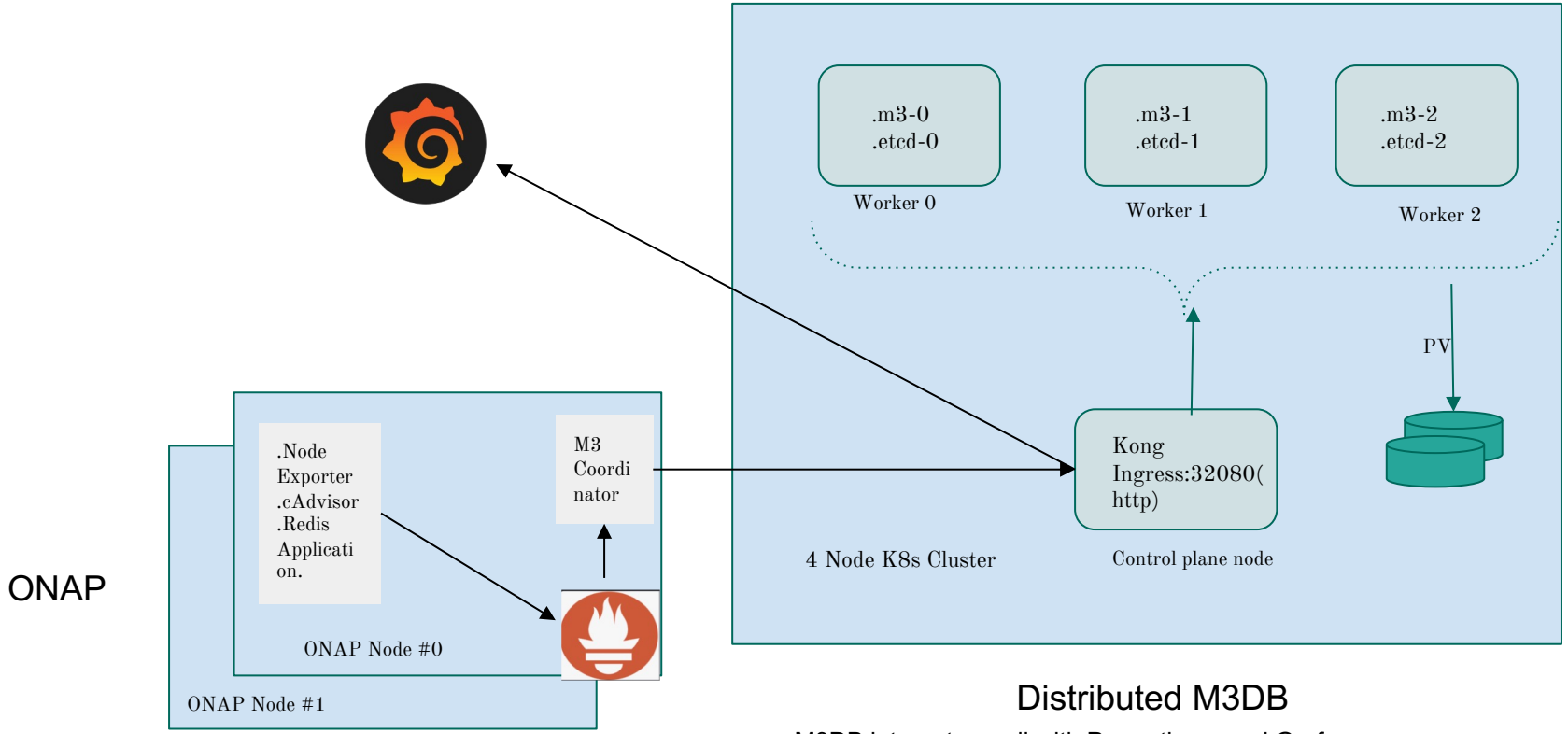
Grafana Dashboards

LFN Developer & Testing Forum

Example:
Dashboard for cAdvisor related metrics

https://grafana.com/grafana/dashboards/14282-cadvisor-exporter/

Persistent Storage for Prometheus



Distributed M3DB

M3DB integrates well with Prometheus and Grafana.

Alert Notification

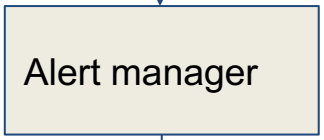
- Example rules.yaml in Prometheus,

```
rules:
  - alert: HighRequestLatency
    expr:
job:request_latency_seconds:mean5m{job="myjob"} >
0.5
for: 10m
```

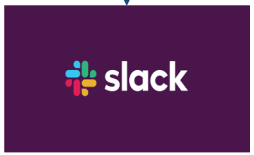
- Example configuration of alertmanager.yaml

```
global:
  resolve_timeout: 1m
  slack_api_url: 'https://hooks.slack.com/
route:
  receiver: 'slack-notifications'
receivers:
- name: 'slack-notifications'
  slack_configs:
  - channel: '#monitoring-instances'
    send_resolved: true
```

1.Configure alerting rules in Prometheus, with target as alertmanager.



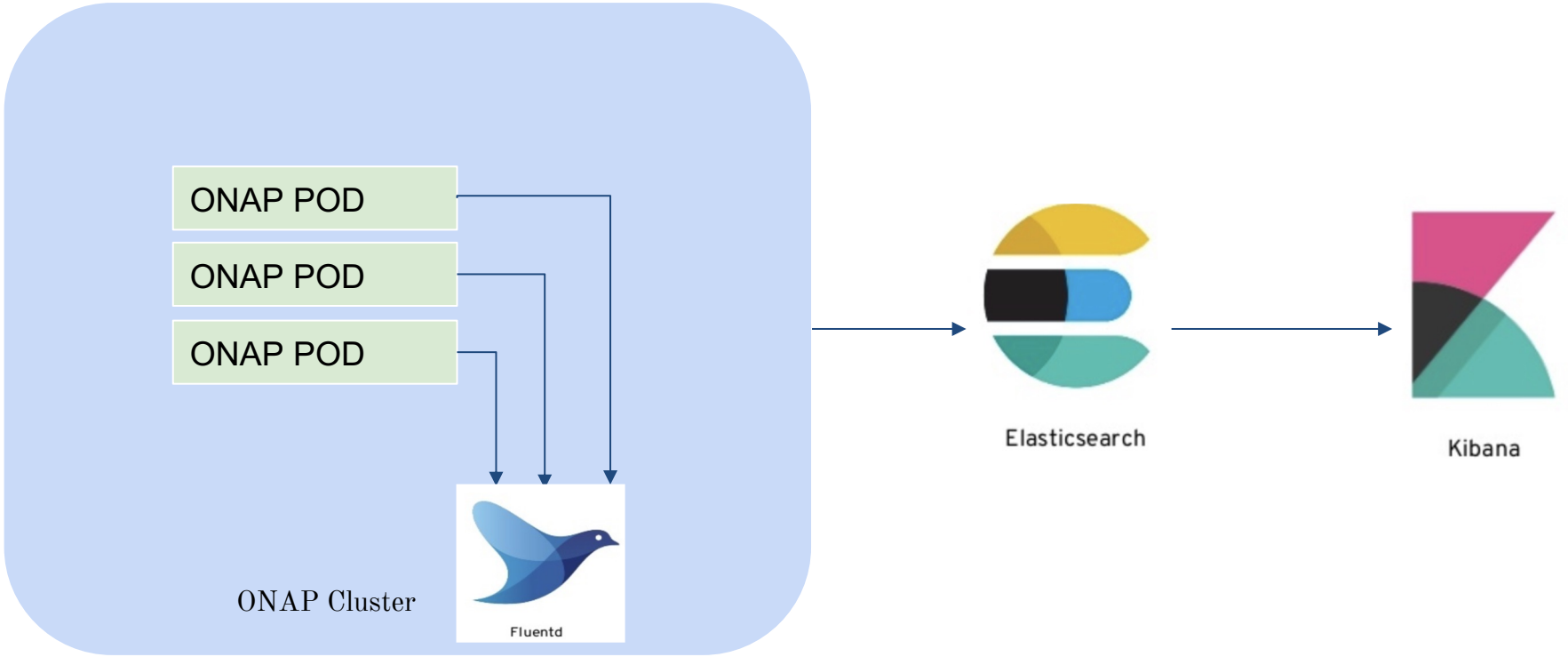
2.Configure prometheus alert manager to send alerts to slack.



Alert Message Notification

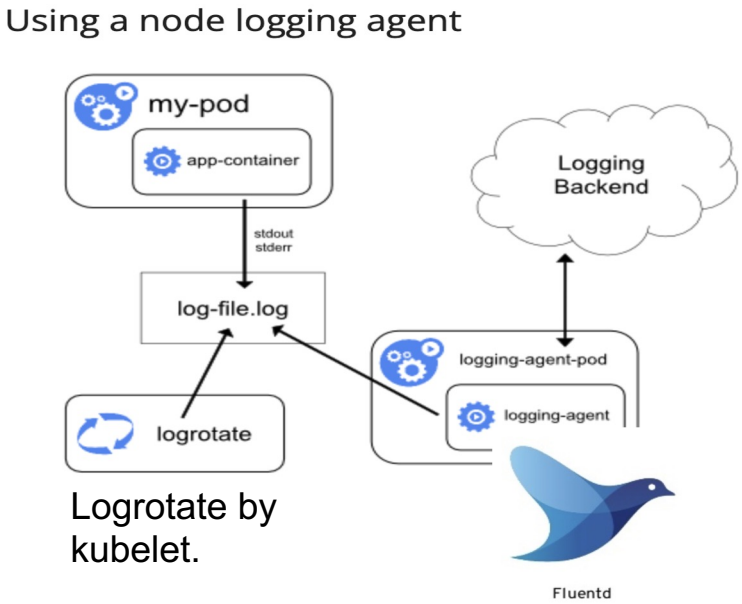
Logging

Deployment Architecture



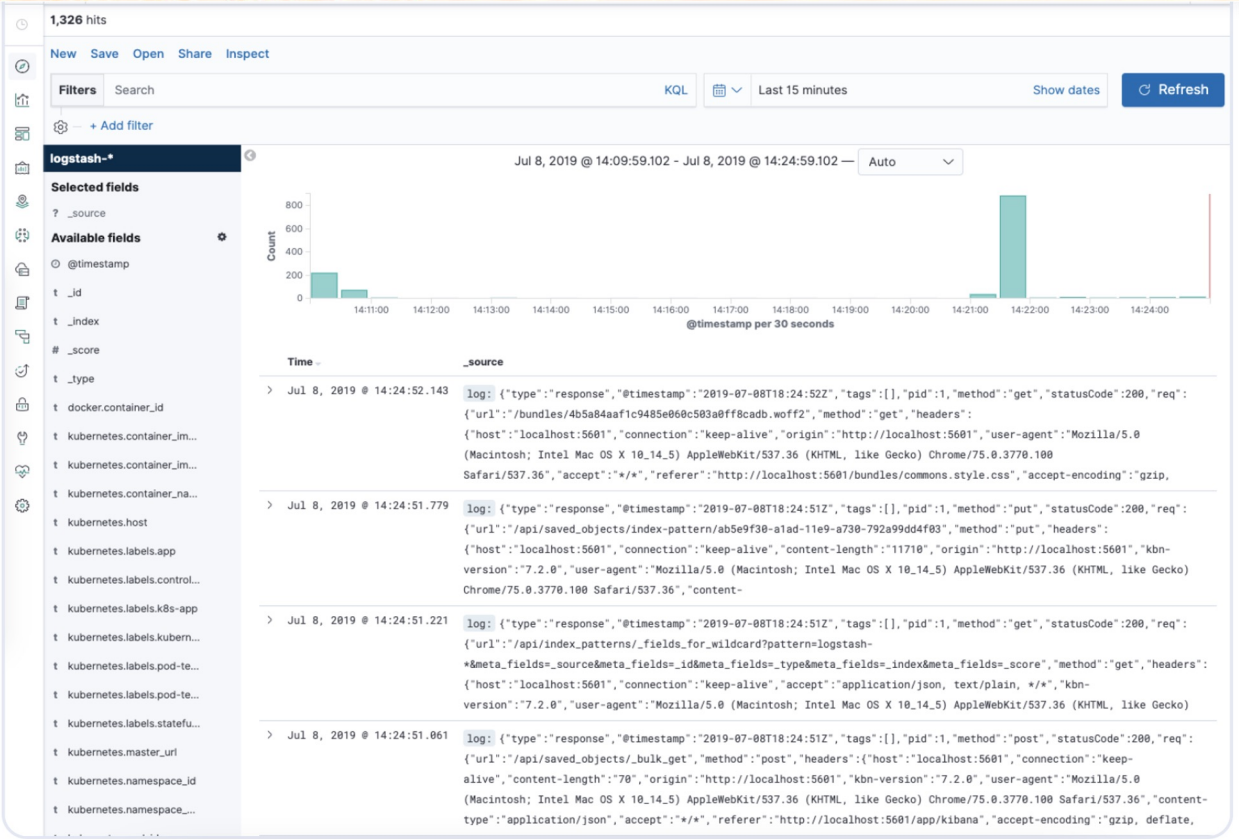
FluentD Deployment and Configuration

- Fluentd deployed as daemonset, node level logging agent in kubernetes
- Fluentd log capture
  - Stdout and stderr log streams of containerised applications.
  - K8s system logs from kubelet, kube-proxy etc.
- Configuration
  - Filters for the logs.
  - Configuration for transforming the logs
  - **`FLUENT_ELASTICSEARCH_HOST`**: Configure the elasticsearch host for storing the logs



Kibana Dashboard

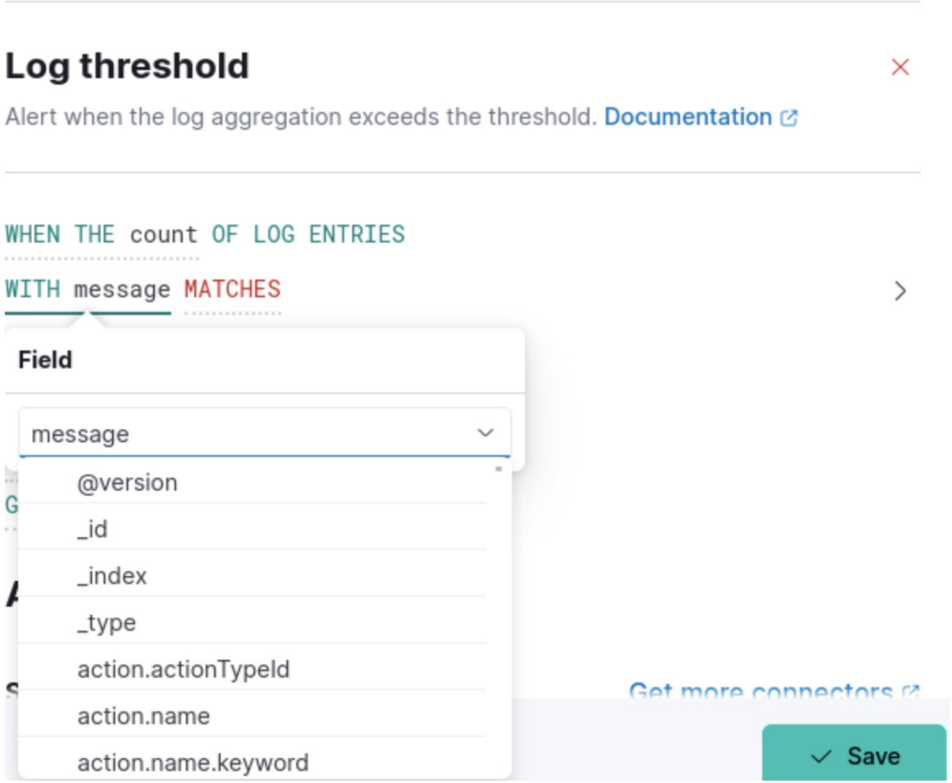
- Login to the Kibana dashboard and create intex patterns and rules to view the filtered logs



Alerts and Events

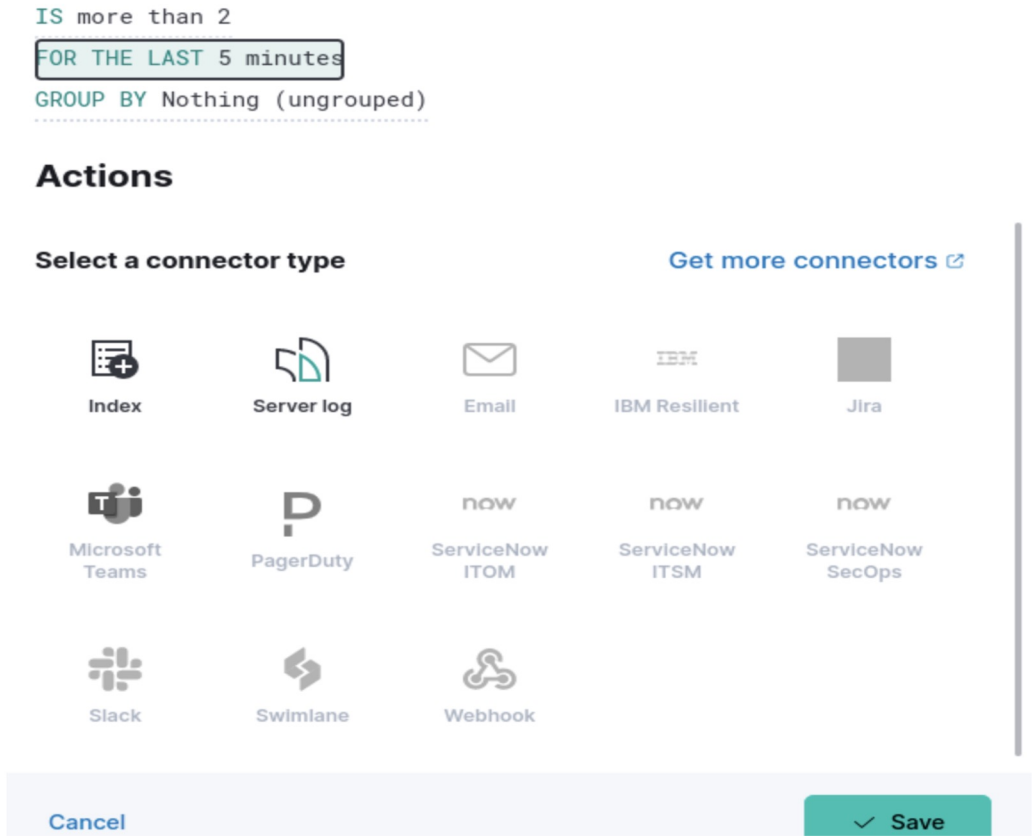
Configure rules and connectors for alerts

- Go to Stack management under management
- Click on Rules and Connectors inside Alert and insights
- Enter required fields, choose Log threshold and select the required field from the drop down below



Alerts and Events

Choose and configure the required connector to get the notification from the available options and save Example: Slack, Email, etc.



Future work

1. Bring in distributed tracing to the Observability Solution.
2. Have single pane of glass for metrics, logs and traces - which will be Grafana.
    a. Datasources on Grafana will be,
        i. Cortex - metrics
        ii. Loki - Logs
        iii. Tempo - Traces
3. Simplify the management of the storage Infrastructure by moving to solutions like cortex(prometheus) and loki(log aggregator) which use S3 as backend storage.
4. Correlation between logs and metrics,
    a. Use Promtail to fetch logs. Promtail implements service discovery, indexing and storing metadata in storage similar to how Prometheus does for metrics.
    b. With same labels/metadata in Loki and Cortex, any anomaly in the metrics can be easily correlated to corresponding logs.

Thank You!