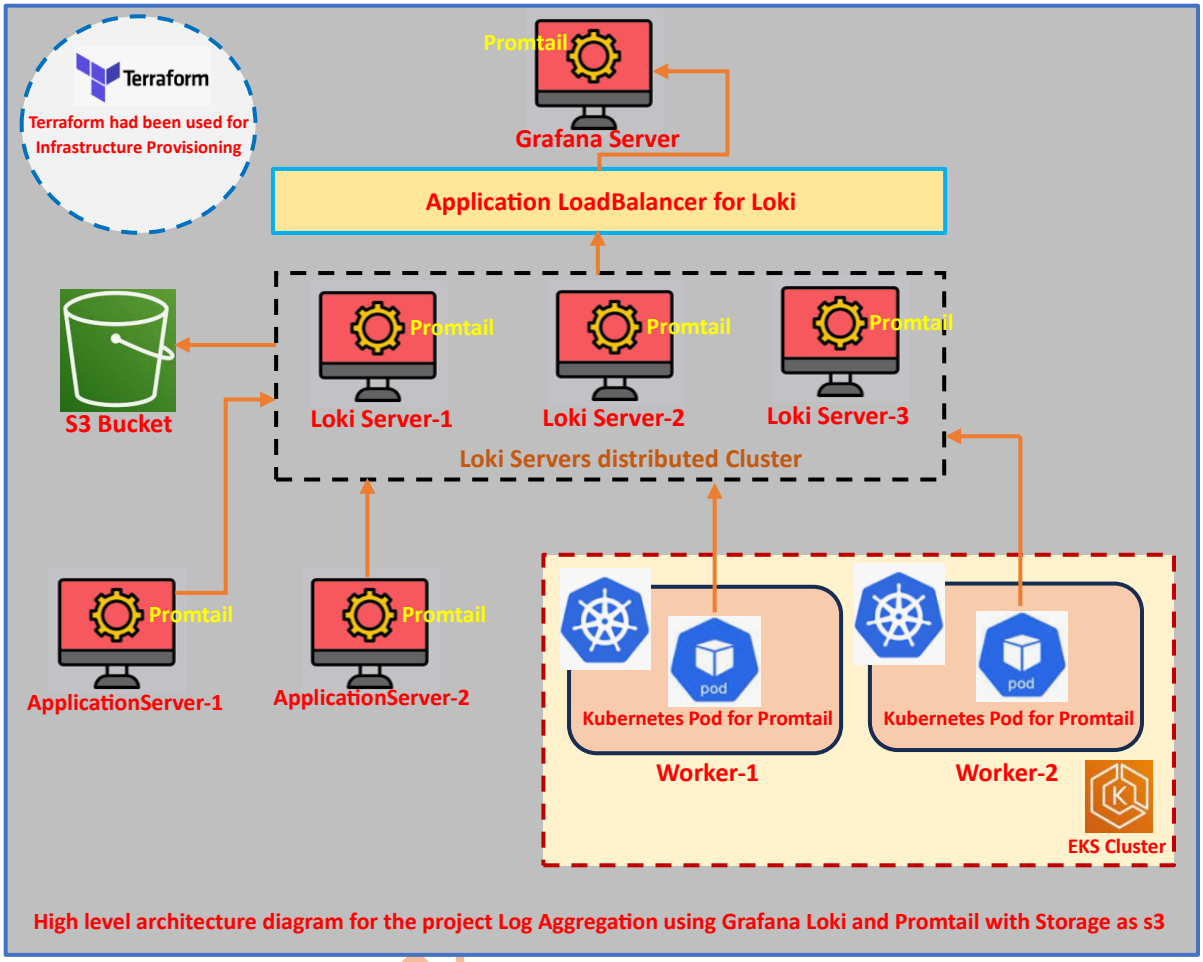
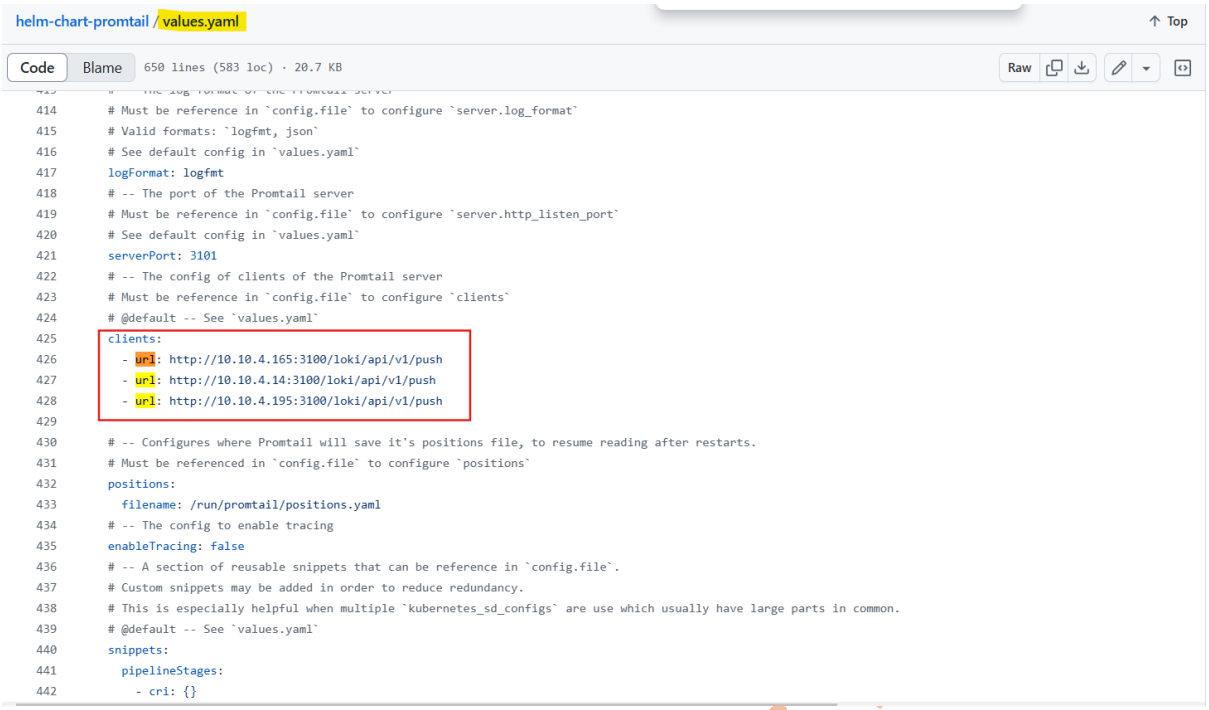
Log Aggregation using Grafana Loki and Promtail with Storage as s3



In this project there are two application servers on which NodeJS Applications are running. Promtail had been installed on the two Application Servers to extract the logs from these two Application Servers and on the three distributed Loki Servers and on Grafana Servers to extract the Logs. Promtail pods was created on EKS Cluster as a part of daemonset using the helm chart. The helm chart is present in GitHub Repository https://github.com/singhritesh85/helm-chart-promtail.git. In the values.yaml file of this helm chart change the url as shown in the screenshot attached below to send the extracted logs to Loki Servers distributed cluster.



helm-chart-promtail / values.yaml

Top

Code | Blame | 650 lines (583 loc) · 20.7 KB | Raw

```
414   # Must be reference in `config.file` to configure `server.log_format`
415   # Valid formats: `logfmt, json`
416   # See default config in `values.yaml`
417   logFormat: logfmt
418   # -- The port of the Promtail server
419   # Must be reference in `config.file` to configure `server.http_listen_port`
420   # See default config in `values.yaml`
421   serverPort: 3101
422   # -- The config of clients of the Promtail server
423   # Must be reference in `config.file` to configure `clients`
424   # @default -- See `values.yaml`
425   clients:
426     - url: http://10.10.4.165:3100/loki/api/v1/push
427     - url: http://10.10.4.14:3100/loki/api/v1/push
428     - url: http://10.10.4.195:3100/loki/api/v1/push
429
430   # -- Configures where Promtail will save it's positions file, to resume reading after restarts.
431   # Must be referenced in `config.file` to configure `positions`
432   positions:
433     filename: /run/promtail/positions.yaml
434   # -- The config to enable tracing
435   enableTracing: false
436   # -- A section of reusable snippets that can be reference in `config.file`.
437   # Custom snippets may be added in order to reduce redundancy.
438   # This is especially helpful when multiple `kubernetes_sd_configs` are use which usually have large parts in common.
439   # @default -- See `values.yaml`
440   snippets:
441     pipelineStages:
442       - cri: {}
```

Here s3 bucket acts as the storage for Loki servers distributed cluster. The Loki Servers distributed cluster will send the Logs to Grafana Server through the Application LoadBalancer for Loki as shown in the diagram above. The three distributed Loki Servers are the part of Target Group which health check path and health check port is **/ready** and **3100**. This target group is attached to the Application LoadBalancer for Loki. A Promtail pod was created on each node of the EKS Cluster using the daemonset and whenever a new node will be created in future for this EKS Cluster a promtail pod will also be created on that newly created node. The Loki servers distributed cluster will be integrated to the Grafana Server using the DNS name of the Application LoadBalancer of Loki.

Creation of Loki Servers distributed cluster

I had created Loki distributed cluster with three Loki Servers, to achieve this the configuration file for the three Loki Servers had been changed as shown in the screenshot shown below with the aim to use s3 bucket as storage for Loki logs. Promtail was also installed on the three Loki servers, two Application Servers, Grafana Server and on EKS Cluster.

```
[root@[redacted] ~]# cat /opt/loki-local-config.yaml
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096
  log_level: debug
  grpc_server_max_concurrent_streams: 1000

common:
  instance_addr: 10.10.4.251
  path_prefix: /tmp/loki
  storage:
    s3:
      bucketnames: s3bucketforlokilogs-dev
      region: us-east-2
#       access_key_id: Key             ### Provided RBAC to the Loki EC2 Instances to send the Logs to S3 Bucket.
#       secret_access_key: Secret      ### Provided RBAC to the Loki EC2 Instances to send the Logs to S3 Bucket.
#       s3forcepathstyle: false        ### Default value is false.
#   storage:
#     filesystem:
#       chunks_directory: /tmp/loki/chunks
#       rules_directory: /tmp/loki/rules
  replication_factor: 3
  ring:
    kvstore:
      store: memberlist

memberlist:
  join_members:
    - 10.10.4.251:7946
    - 10.10.4.221:7946
    - 10.10.4.146:7946

query_range:
  results_cache:
    cache:
      embedded_cache:
        enabled: true
        max_size_mb: 100

schema_config:
  configs:
    - from: 2020-10-24
      store: tsdb
      object_store: s3    ###filesystem
      schema: v13
      index:
        prefix: index_
        period: 24h

pattern_ingester:
  enabled: true
  metric_aggregation:
    enabled: true
    loki_address: 10.10.4.251:3100

ruler:
  alertmanager_url: http://10.10.4.251:9093

frontend:
  encoding: protobuf

# By default, Loki will send anonymous, but uniquely-identifiable usage and configuration
# analytics to Grafana Labs. These statistics are sent to https://stats.grafana.org/
#
# Statistics help us better understand how Loki is used, and they show us performance
# levels for most users. This helps us prioritize features and documentation.
# For more information on what's sent, look at
# https://github.com/grafana/loki/blob/main/pkg/analytics/stats.go
# Refer to the buildReport method to see what goes into a report.
#
# If you would like to disable reporting, uncomment the following lines:
#analytics:
#  reporting_enabled: false
```

```
cat /opt/loki-local-config.yaml

auth_enabled: false


server:

 http_listen_port: 3100

 grpc_listen_port: 9096

 log_level: debug

 grpc_server_max_concurrent_streams: 1000


common:

 instance_addr: 10.10.4.251

 path_prefix: /tmp/loki

 storage:

  s3:

    bucketnames: s3bucketforlokilogs-dev

    region: us-east-2

#     access_key_id: Key          ### Provided RBAC to the Loki EC2 Instances to send the Logs to S3 Bucket.

#     secret_access_key: Secret     ### Provided RBAC to the Loki EC2 Instances to send the Logs to S3 Bucket.

#     s3forcepathstyle: false      ### Default value is false.

#  storage:

#    filesystem:

#      chunks_directory: /tmp/loki/chunks

#      rules_directory: /tmp/loki/rules

 replication_factor: 3

 ring:

  kvstore:

    store: memberlist
```

```
memberlist:
  join_members:
    - 10.10.4.251:7946
    - 10.10.4.221:7946
    - 10.10.4.146:7946

query_range:
  results_cache:
    cache:
      embedded_cache:
        enabled: true
        max_size_mb: 100

schema_config:
  configs:
    - from: 2020-10-24
      store: tsdb
      object_store: s3     ###filesystem
      schema: v13
      index:
        prefix: index_
        period: 24h

pattern_ingester:
  enabled: true
  metric_aggregation:
    enabled: true
    loki_address: 10.10.4.251:3100

ruler:
  alertmanager_url: http://10.10.4.251:9093

frontend:
  encoding: protobuf
```

```
[root@[redacted] ~]# systemctl start loki.service
[root@[redacted] ~]# systemctl status loki.service
```

```
[root@[redacted] ~]# systemctl enable loki.service
```

```
[root@[redacted] ~]# systemctl start promtail.service
[root@[redacted] ~]# systemctl status promtail.service
```

```
[root@[redacted] ~]# systemctl enable promtail.service
```

The Service for Promtail and Loki will be started, checked its status, and stated from the boot time on all the three Loki Servers as shown in the screenshot attached above.

For the three Loki Servers the Promtail configuration file is as shown in the screenshot attached below.

```
[root@[redacted] ~]# cat /opt/promtail-local-config.yaml
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://10.10.4.251:3100/loki/api/v1/push
  - url: http://10.10.4.221:3100/loki/api/v1/push
  - url: http://10.10.4.146:3100/loki/api/v1/push

scrape_configs:
- job_name: system
  static_configs:
  - targets:
      - localhost
    labels:
      job: varlogs
      __path__: /var/log/*log
      stream: stdout
```

```
cat /opt/promtail-local-config.yaml
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://10.10.4.251:3100/loki/api/v1/push
  - url: http://10.10.4.221:3100/loki/api/v1/push
  - url: http://10.10.4.146:3100/loki/api/v1/push

scrape_configs:
- job_name: system
  static_configs:
  - targets:
      - localhost
    labels:
      job: varlogs
      __path__: /var/log/*log
      stream: stdout
```


The Configuration file for the Promtail on the two Application Servers is as shown in the screenshot attached below. I had created a job with the name **dexter** and label **dexter-application-logs** to capture for the logs from the specified path as shown in the screenshot attached below.

```
[root@[redacted] simple-nodejs-app]# cat /opt/promtail-local-config.yaml
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://10.10.4.251:3100/loki/api/v1/push
  - url: http://10.10.4.221:3100/loki/api/v1/push
  - url: http://10.10.4.146:3100/loki/api/v1/push

scrape_configs:
- job_name: system
  static_configs:
  - targets:
      - localhost
    labels:
      job: varlogs
      __path__: /var/log/*log
      stream: stdout
- job_name: dexter
  static_configs:
  - targets:
      - localhost
    labels:
      job: dexter-application-logs
      __path__: /root/simple-nodejs-app/*log
      stream: stdout
```

```
cat /opt/promtail-local-config.yaml
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://10.10.4.251:3100/loki/api/v1/push
  - url: http://10.10.4.221:3100/loki/api/v1/push
  - url: http://10.10.4.146:3100/loki/api/v1/push

scrape_configs:
- job_name: system
  static_configs:
  - targets:
      - localhost
    labels:
      job: varlogs
      __path__: /var/log/*log
      stream: stdout
- job_name: dexter
  static_configs:
  - targets:
      - localhost
    labels:
      job: dexter-application-logs
      __path__: /root/simple-nodejs-app/*log
      stream: stdout
```

Cloned the NodeJS Application from GitHub Repo and started that. Then started Promtail service, checked status and started from the boot time as shown in the screenshot attached below.

```
[root@[redacted] ~]# git clone https://github.com/singhritesh85/simple-nodejs-app.git

[root@[redacted] ~]# cd simple-nodejs-app/
[root@[redacted] simple-nodejs-app]# npm install

[root@[redacted] simple-nodejs-app]# nohup npm start >> app.log &

[root@[redacted] ~]# systemctl start promtail.service
[root@[redacted] ~]# systemctl status promtail.service

[root@[redacted] ~]# systemctl enable promtail.service
```

Here for EKS Cluster I am aggregating logs from the EKS cluster by installing the promtail on EKS using helm chart as shown in the screenshot attached below.

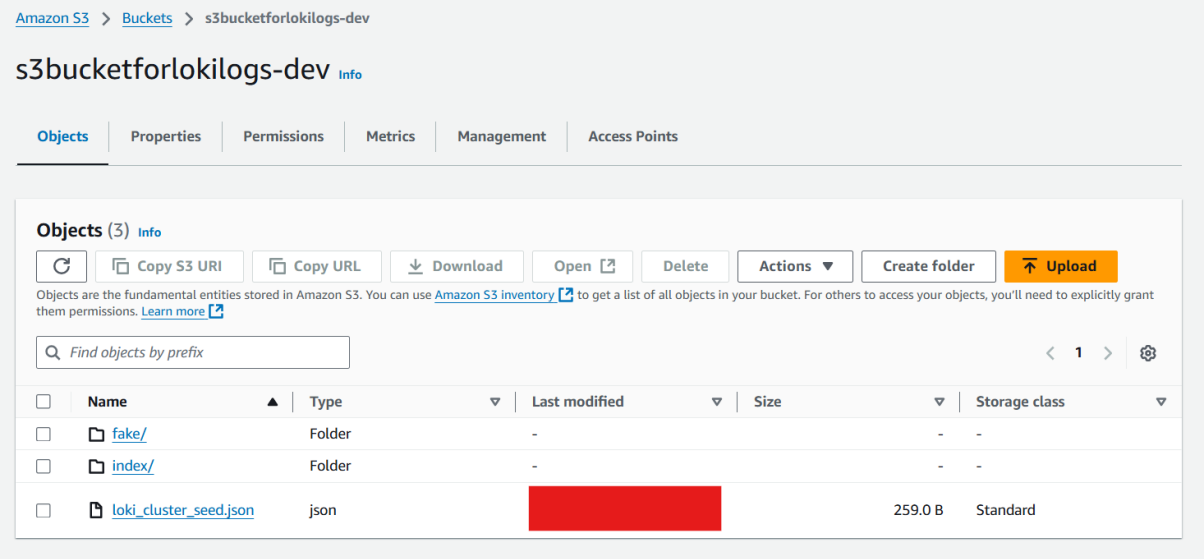
```
[root@[redacted] ~]# git clone https://github.com/singhritesh85/helm-chart-promtail.git

[root@[redacted] ~]# helm upgrade --values helm-chart-promtail/values.yaml --install promtail ./helm-chart-promtail -n promtail
Release "promtail" does not exist. Installing it now.
NAME: promtail
LAST DEPLOYED: [redacted]
NAMESPACE: promtail
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
**********************************************************************
 Welcome to Grafana Promtail
 Chart version: 6.16.6
 Promtail version: 3.0.0
**********************************************************************

Verify the application is working by running these commands:
* kubectl --namespace promtail port-forward daemonset/promtail 3101
* curl http://127.0.0.1:3101/metrics
[root@[redacted] ~]# kubectl get all -n promtail
NAME                 READY   STATUS    RESTARTS   AGE
pod/promtail-[redacted]   1/1     Running   0          12s
pod/promtail-[redacted]   1/1     Running   0          11s

NAME                      DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/promtail   2         2         2       2            2           <none>          12s
```

The promtail pods had been created on each node of the EKS cluster and scraped the logs and send to the Loki which was created as a distributed cluster as explain earlier.

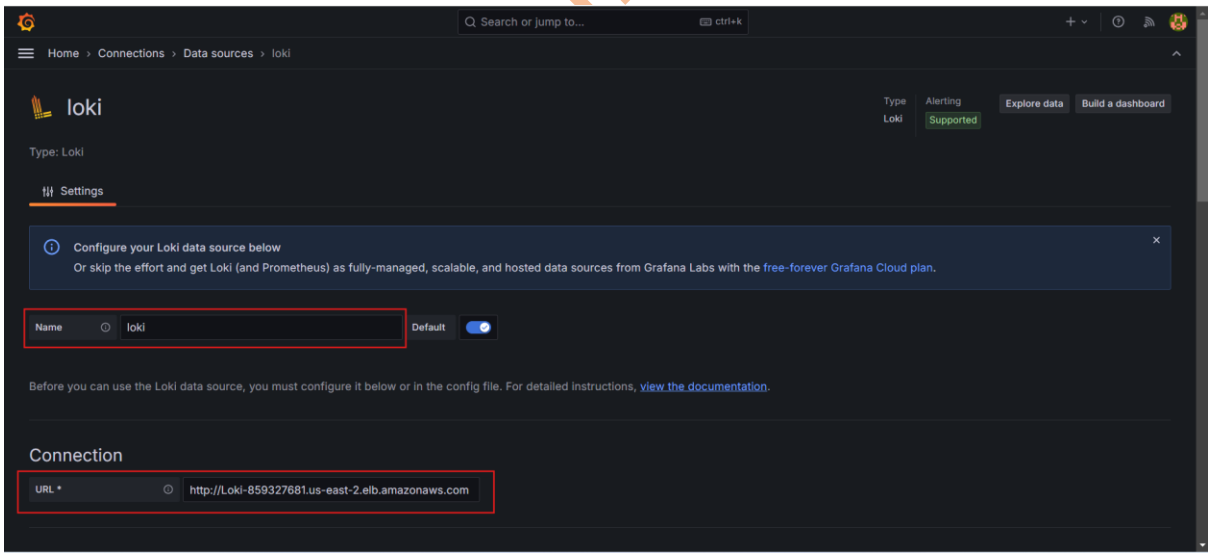
```
[root@[redacted] ~]# kubectl get nodes
NAME                                        STATUS   ROLES    AGE     VERSION
ip-[redacted].us-east-2.compute.internal    Ready    <none>   3h54m   v1.27.9-eks-5e0fdde
ip-[redacted].us-east-2.compute.internal    Ready    <none>   3h54m   v1.27.9-eks-5e0fdde
[root@[redacted] ~]# kubectl get pods -n promtail -o wide
NAME             READY   STATUS    RESTARTS   AGE   IP             NODE                                        NOMINATED NODE   READINESS GATES
promtail-[redacted]   1/1     Running   0          [redacted]m   [redacted]   ip-[redacted].us-east-2.compute.internal    <none>           <none>
promtail-[redacted]   1/1     Running   0          [redacted]m   [redacted]   ip-[redacted].us-east-2.compute.internal    <none>           <none>
```

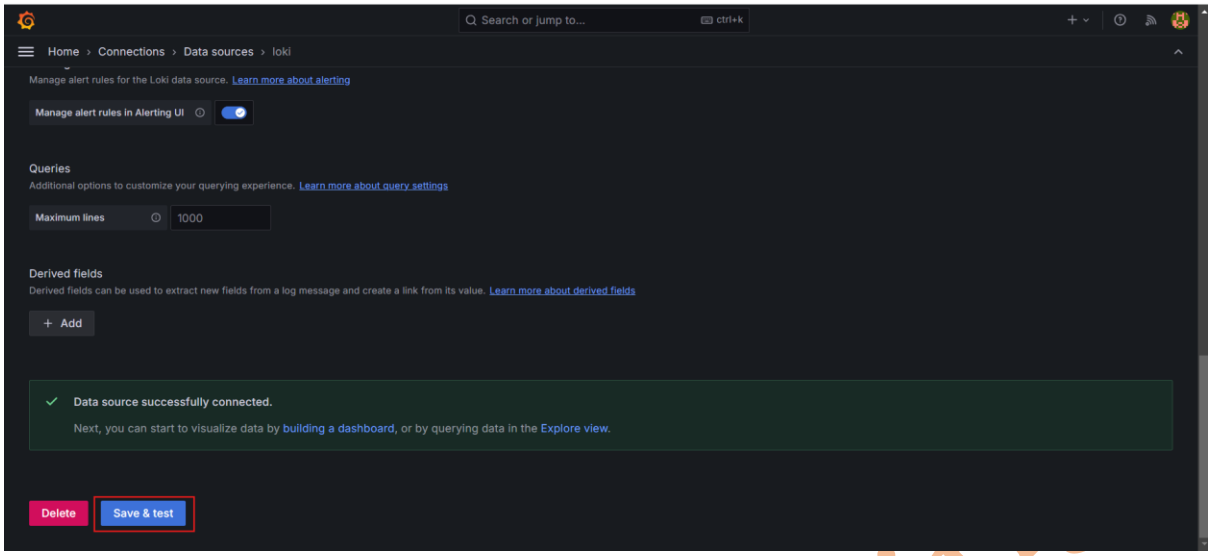
Before installing promtail pods using helm chart make sure you provided the correct information regarding Loki distributed cluster as explained earlier on page 1 and on page 2 with the attached screenshot.

The s3 bucket started capturing the Loki Logs as shown in the screenshot attached below.

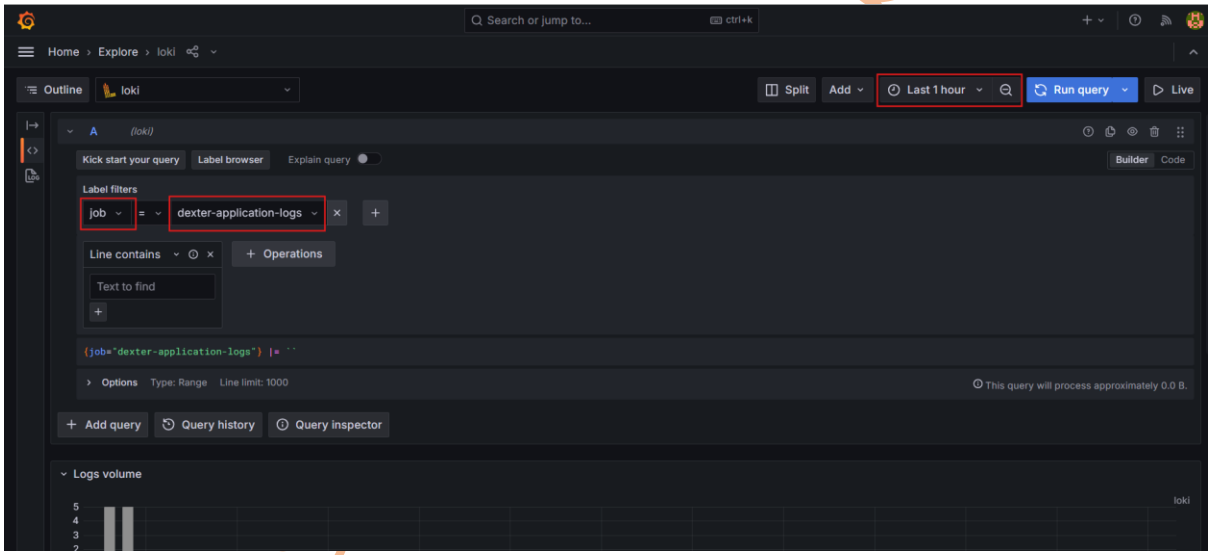


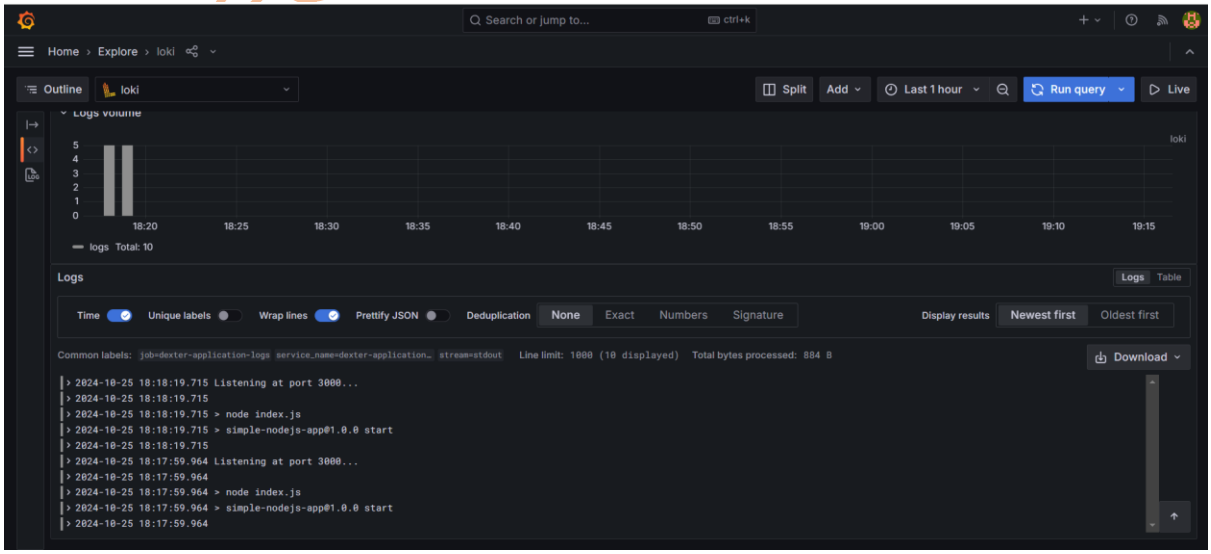
To integrate the Loki with Grafana the connection URL is provided with DNS name of the Application LoadBalancer of Loki as shown in the screenshot attached below and tested the connection which showed it was connected successfully.

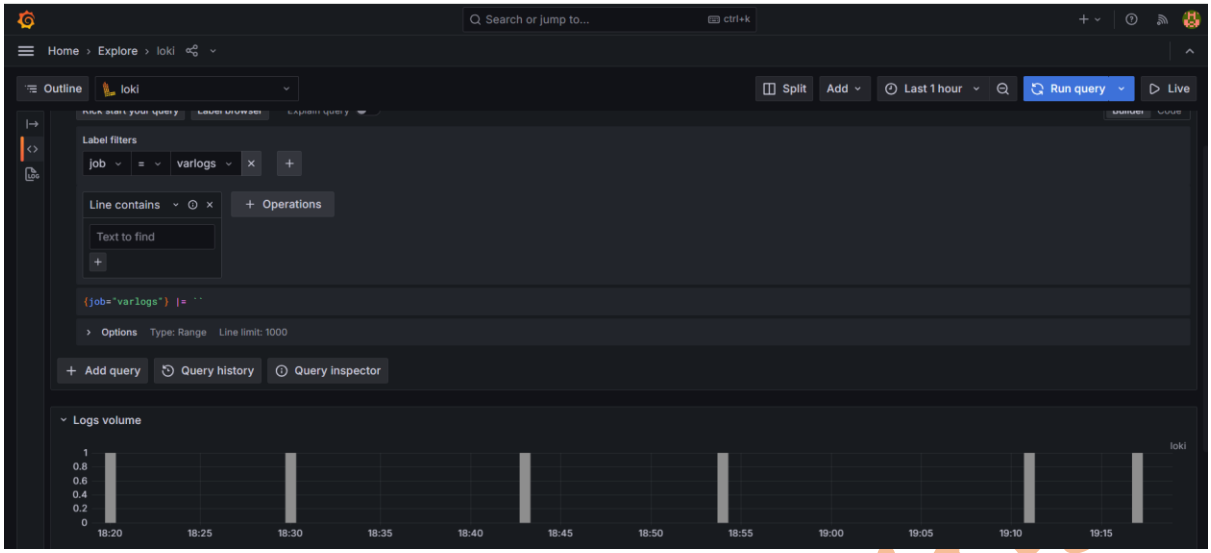




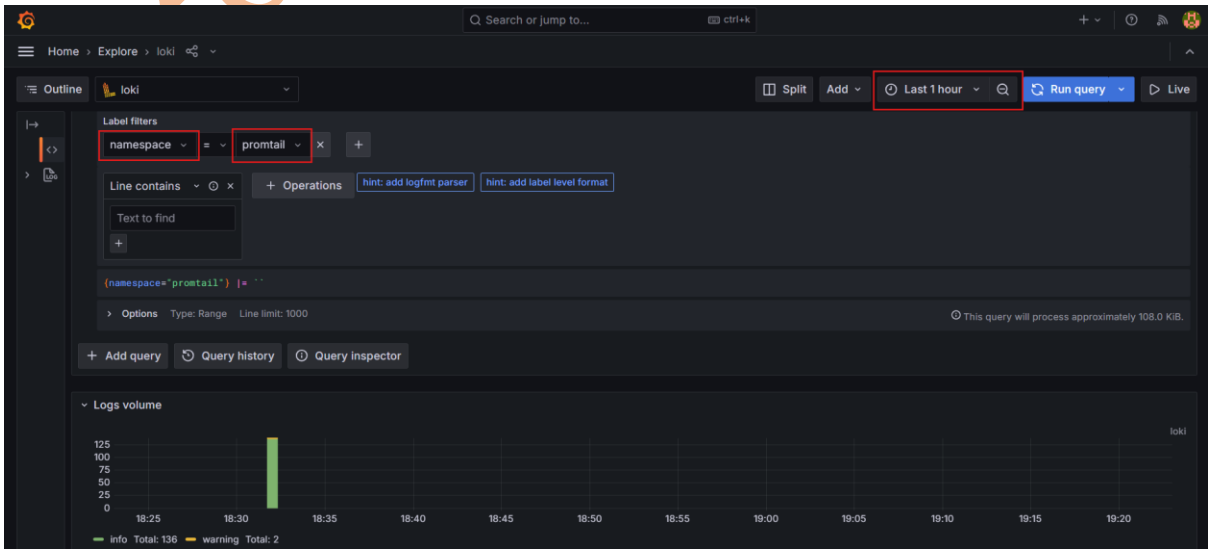
Finally start filtering the required Logs using the labels as shown in screenshot attached below.







You can filer the specific line using the Line Contains filer for Logs as shown in the screenshot attached below.



The entry for Route53 to create the record set is as shown in the screenshot attached below.