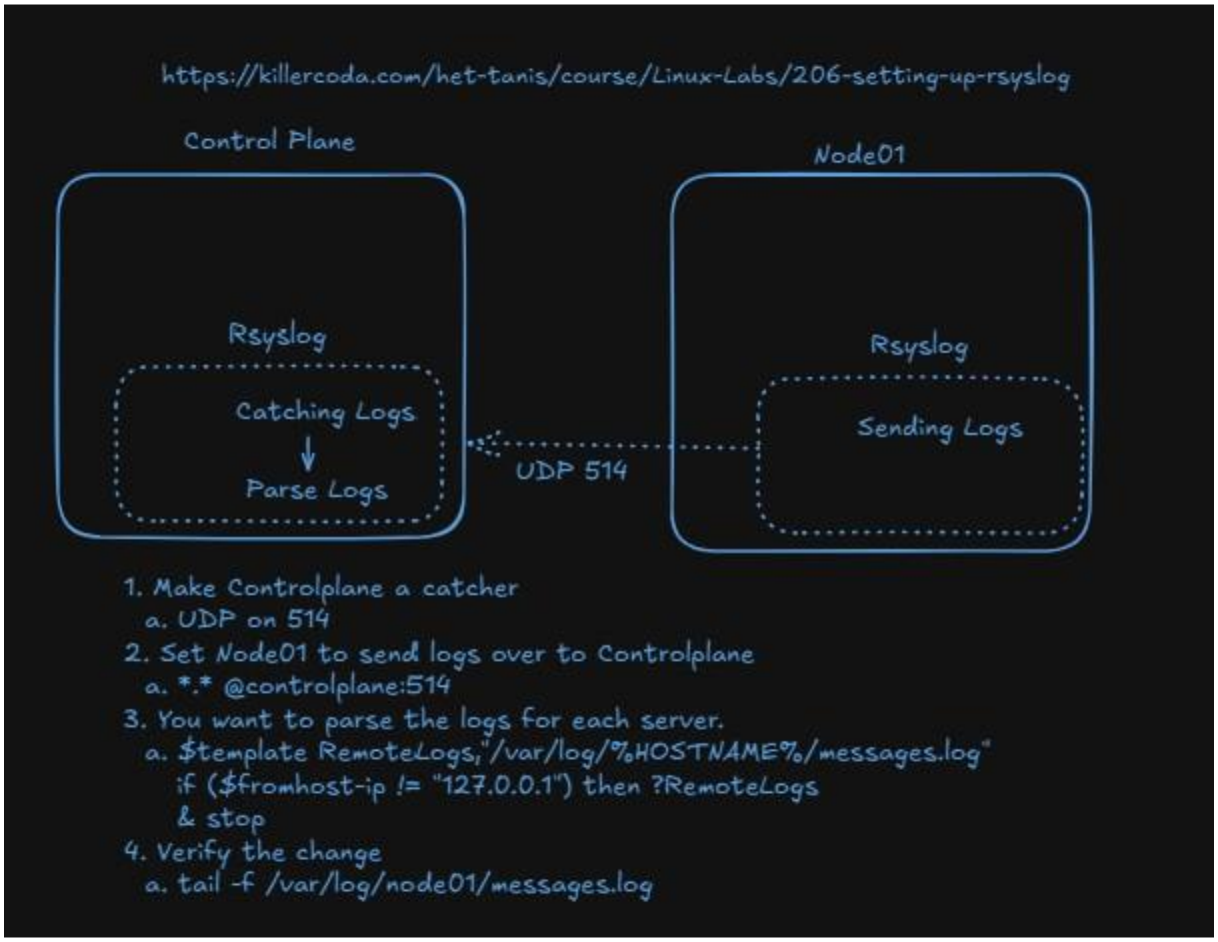
Unit 6 Lab - Monitoring and Parsing Logs

<u>Rsyslog forwarding and collection 1 of 2</u>

1. Consider this architecture

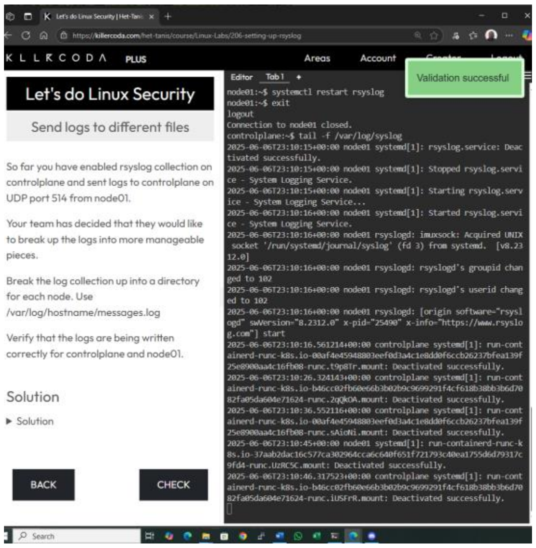


https://killercoda.com/het-tanis/course/Linux-Labs/206-setting-up-rsyslog

1. Make Controlplane a catcher
   a. UDP on 514
2. Set Node01 to send logs over to Controlplane
   a. *.* @controlplane:514
3. You want to parse the logs for each server.
   a. $template RemoteLogs,"/var/log/%HOSTNAME%/messages.log"
      if ($fromhost-ip != "127.0.0.1") then ?RemoteLogs
      & stop
4. Verify the change
   a. tail -f /var/log/node01/messages.log

Rsyslog forwarding and collection 2 of 2

2 Complete the lab: https://killercoda.com/het-tanis/course/Linux-Labs/206-setting-up-rsyslog


Let's do Linux Security | Het-Tanis
https://killercoda.com/het-tanis/course/Linux-Labs/206-setting-up-rsyslog
KILLERCODA PLUS
Areas
Account
Validation successful
Editor
Tab 1
BACK
CHECK
Search

Let's do Linux Security

Send logs to different files

So far you have enabled rsyslog collection on controlplane and sent logs to controlplane on UDP port 514 from node01.

Your team has decided that they would like to break up the logs into more manageable pieces.

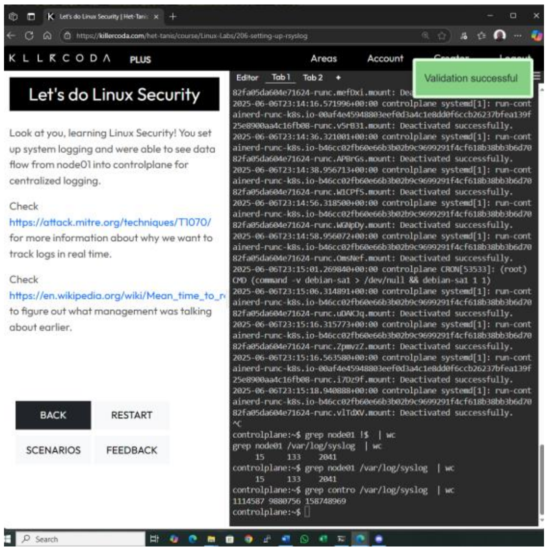
Break the log collection up into a directory for each node. Use /var/log/hostname/messages.log

Verify that the logs are being written correctly for controlplane and node01.

Solution

▶ Solution

```
node01:~$ systemctl restart rsyslog
node01:~$ exit
logout
Connection to node01 closed.
controlplane:~$ tail -f /var/log/syslog
2025-06-06T23:10:15+00:00 node01 systemd[1]: rsyslog.service: Deac
tivated successfully.
2025-06-06T23:10:15+00:00 node01 systemd[1]: Stopped rsyslog.servi
ce - System Logging Service.
2025-06-06T23:10:15+00:00 node01 systemd[1]: Starting rsyslog.serv
ice - System Logging Service...
2025-06-06T23:10:16+00:00 node01 systemd[1]: Started rsyslog.servi
ce - System Logging Service.
2025-06-06T23:10:16+00:00 node01 rsyslogd: imuxsock: Acquired UNIX
 socket '/run/systemd/journal/syslog' (fd 3) from systemd.  [v8.23
12.0]
2025-06-06T23:10:16+00:00 node01 rsyslogd: rsyslogd's groupid chan
ged to 102
2025-06-06T23:10:16+00:00 node01 rsyslogd: rsyslogd's userid chang
ed to 102
2025-06-06T23:10:16+00:00 node01 rsyslogd: [origin software="rsysl
ogd" swVersion="8.2312.0" x-pid="25490" x-info="https://www.rsyslo
g.com"] start
2025-06-06T23:10:16.561214+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-00af4e45948803eef0d3a4c1e8dd0f6ccb26237bfea139f
25e8900aa4c16fb08-runc.t9p8Tr.mount: Deactivated successfully.
2025-06-06T23:10:26.324143+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.2qQkOA.mount: Deactivated successfully.
2025-06-06T23:10:36.552116+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-00af4e45948803eef0d3a4c1e8dd0f6ccb26237bfea139f
25e8900aa4c16fb08-runc.sAioNi.mount: Deactivated successfully.
2025-06-06T23:10:45+00:00 node01 systemd[1]: run-containerd-runc-k
8s.io-37aab2dac16c577ca302964cca6c640f651f721793c40ea1755d6d79317c
9fd4-runc.UzRC5C.mount: Deactivated successfully.
2025-06-06T23:10:46.317523+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.iUSFrR.mount: Deactivated successfully.
```


Let's do Linux Security | Het-Tanis
https://killercoda.com/het-tanis/course/Linux-Labs/206-setting-up-rsyslog
KILLERCODA PLUS
Areas
Account
Validation successful
Editor
Tab 1
Tab 2
BACK
RESTART
SCENARIOS
FEEDBACK
Search

Let's do Linux Security

Look at you, learning Linux Security! You set up system logging and were able to see data flow from node01 into controlplane for centralized logging.

Check https://attack.mitre.org/techniques/T1070/ for more information about why we want to track logs in real time.

Check https://en.wikipedia.org/wiki/Mean_time_to_r to figure out what management was talking about earlier.

```
82fa05da604e71624-runc.mefDxi.mount: Dea
2025-06-06T23:14:16.571996+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-00af4e45948803eef0d3a4c1e8dd0f6ccb26237bfea139f
25e8900aa4c16fb08-runc.v5rB31.mount: Deactivated successfully.
2025-06-06T23:14:36.321001+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.APBrGs.mount: Deactivated successfully.
2025-06-06T23:14:38.956713+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.W1CPf5.mount: Deactivated successfully.
2025-06-06T23:14:56.318500+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.WGNpDy.mount: Deactivated successfully.
2025-06-06T23:14:58.956072+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.OmsNef.mount: Deactivated successfully.
2025-06-06T23:15:01.269840+00:00 controlplane CRON[53533]: (root)
CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2025-06-06T23:15:06.314891+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.uDAKJq.mount: Deactivated successfully.
2025-06-06T23:15:16.315773+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.ZpmvzZ.mount: Deactivated successfully.
2025-06-06T23:15:16.563580+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-00af4e45948803eef0d3a4c1e8dd0f6ccb26237bfea139f
25e8900aa4c16fb08-runc.i7Dz9f.mount: Deactivated successfully.
2025-06-06T23:15:18.940888+00:00 controlplane systemd[1]: run-cont
ainerd-runc-k8s.io-b46cc02fb60e66b3b02b9c9699291f4cf618b38bb3b6d70
82fa05da604e71624-runc.vlTdXV.mount: Deactivated successfully.
^C
controlplane:~$ grep node01 !$  | wc
grep node01 /var/log/syslog  | wc
     15     133    2041
controlplane:~$ grep node01 /var/log/syslog  | wc
     15     133    2041
controlplane:~$ grep contro /var/log/syslog  | wc
1114587 9880756 158748969
controlplane:~$
```

Why do we split out the logs in this lab?

So that we can properly manage the files for each machines … we don’t want one huge file

Why don't we just aggregate them to one place?

We don’t want one huge file … it just creates another single point of failure

What do we split them out by?

By hostname

How does that template configuration work?

tbd

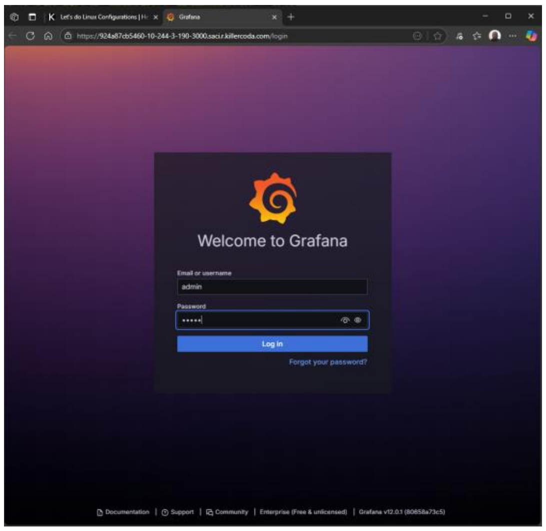
Are we securing this communication in any way, or do we still need to configure that?

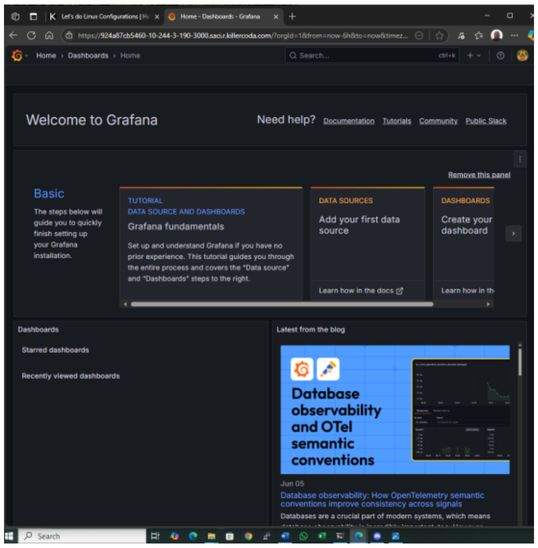
tbd

<u>Agents forward to a centralized platform 1 of 4</u>

1. Review the base architecture here: https://grafana.com/docs/loki/latest/get-started/architecture/

2. Complete the lab here: https://killercoda.com/het-tanis/course/Linux-Labs/102-monitoring-linux-logs


Grafana
Welcome to Grafana
Email or username
admin
Password
Log in
Forgot your password?
Documentation | Support | Community | Enterprise (Free & unlicensed) | Grafana v12.0.1 (80658a73c5)



Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Remove this panel

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

DATA SOURCES
Add your first data source

Learn how in the docs

DASHBOARDS
Create your dashboard

Dashboards

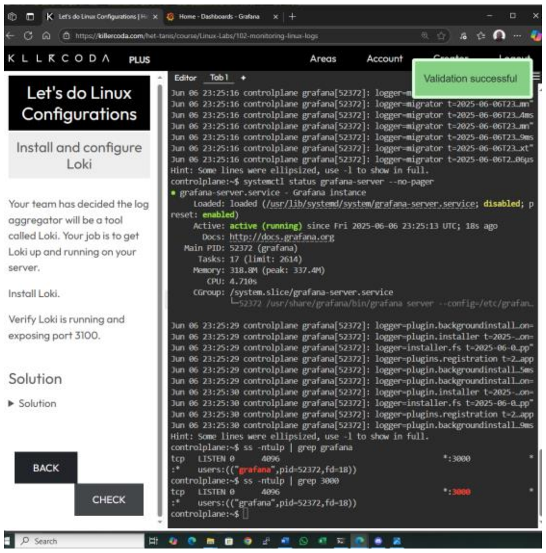
Starred dashboards

Recently viewed dashboards

Latest from the blog

Jun 05
Database observability: How OpenTelemetry semantic conventions improve consistency across signals

Databases are a crucial part of modern systems, which means



Let's do Linux Configurations

Install and configure Loki

Your team has decided the log aggregator will be a tool called Loki. Your job is to get Loki up and running on your server.

Install Loki.

Verify Loki is running and exposing port 3100.

Solution

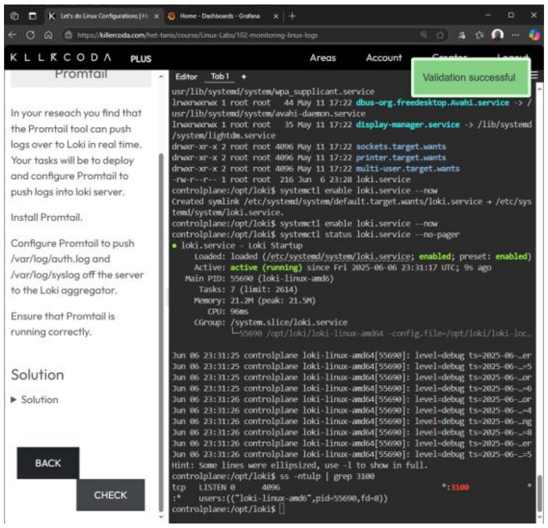
▸ Solution

BACK

CHECK

Validation successful

```
Jun 06 23:25:16 controlplane grafana[52372]: logger=m
Jun 06 23:25:16 controlplane grafana[52372]: logger=migrator t=2025-06-06T23…mn"
Jun 06 23:25:16 controlplane grafana[52372]: logger=migrator t=2025-06-06T23…4ms
Jun 06 23:25:16 controlplane grafana[52372]: logger=migrator t=2025-06-06T23…mn"
Jun 06 23:25:16 controlplane grafana[52372]: logger=migrator t=2025-06-06T23…9ms
Jun 06 23:25:16 controlplane grafana[52372]: logger=migrator t=2025-06-06T23…xt"
Jun 06 23:25:16 controlplane grafana[52372]: logger=migrator t=2025-06-06T2…06µs
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:~$ systemctl status grafana-server --no-pager
● grafana-server.service - Grafana instance
     Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; p
reset: enabled)
     Active: active (running) since Fri 2025-06-06 23:25:13 UTC; 18s ago
       Docs: http://docs.grafana.org
   Main PID: 52372 (grafana)
      Tasks: 17 (limit: 2614)
     Memory: 318.8M (peak: 337.4M)
        CPU: 4.710s
     CGroup: /system.slice/grafana-server.service
             └─52372 /usr/share/grafana/bin/grafana server --config=/etc/grafan…

Jun 06 23:25:29 controlplane grafana[52372]: logger=plugin.backgroundinstall…on=
Jun 06 23:25:29 controlplane grafana[52372]: logger=plugin.installer t=2025-…on=
Jun 06 23:25:29 controlplane grafana[52372]: logger=installer.fs t=2025-06-0…pp"
Jun 06 23:25:29 controlplane grafana[52372]: logger=plugins.registration t=2…app
Jun 06 23:25:29 controlplane grafana[52372]: logger=plugin.backgroundinstall…5ms
Jun 06 23:25:29 controlplane grafana[52372]: logger=plugin.backgroundinstall…on=
Jun 06 23:25:30 controlplane grafana[52372]: logger=plugin.installer t=2025-…on=
Jun 06 23:25:30 controlplane grafana[52372]: logger=installer.fs t=2025-06-0…pp"
Jun 06 23:25:30 controlplane grafana[52372]: logger=plugins.registration t=2…app
Jun 06 23:25:30 controlplane grafana[52372]: logger=plugin.backgroundinstall…9ms
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:~$ ss -ntulp | grep grafana
tcp   LISTEN 0      4096                                  *:3000            *
:*    users:(("grafana",pid=52372,fd=18))
controlplane:~$ ss -ntulp | grep 3000
tcp   LISTEN 0      4096                                  *:3000            *
:*    users:(("grafana",pid=52372,fd=18))
controlplane:~$
```



Promtail

In your reseach you find that the Promtail tool can push logs over to Loki in real time. Your tasks will be to deploy and configure Promtail to push logs into loki server.

Install Promtail.

Configure Promtail to push /var/log/auth.log and /var/log/syslog off the server to the Loki aggregator.

Ensure that Promtail is running correctly.

Solution

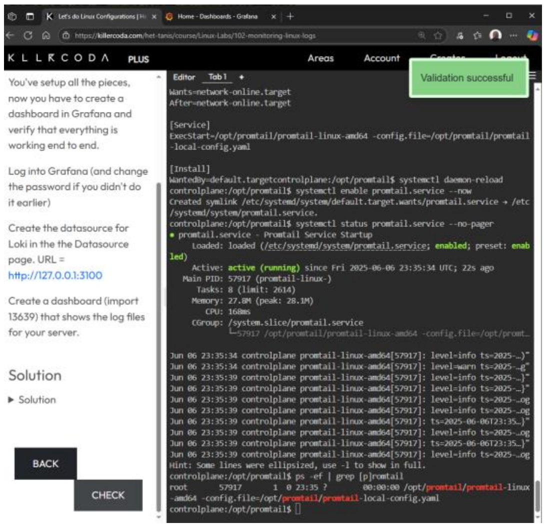
▸ Solution

BACK

CHECK

Validation successful

```
usr/lib/systemd/system/wpa_supplicant.service
lrwxrwxrwx 1 root root   44 May 11 17:22 dbus-org.freedesktop.Avahi.service -> /
usr/lib/systemd/system/avahi-daemon.service
lrwxrwxrwx 1 root root   35 May 11 17:22 display-manager.service -> /lib/systemd
/system/lightdm.service
drwxr-xr-x 2 root root 4096 May 11 17:22 sockets.target.wants
drwxr-xr-x 2 root root 4096 May 11 17:22 printer.target.wants
drwxr-xr-x 2 root root 4096 May 11 17:22 multi-user.target.wants
-rw-r--r-- 1 root root  216 Jun  6 23:28 loki.service
controlplane:/opt/loki$ systemctl enable loki.service --now
Created symlink /etc/systemd/system/default.target.wants/loki.service → /etc/sys
temd/system/loki.service.
controlplane:/opt/loki$ systemctl enable loki.service --now
controlplane:/opt/loki$ systemctl status loki.service --no-pager
● loki.service - Loki Startup
     Loaded: loaded (/etc/systemd/system/loki.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-06-06 23:31:17 UTC; 9s ago
   Main PID: 55690 (loki-linux-amd6)
      Tasks: 7 (limit: 2614)
     Memory: 21.2M (peak: 21.5M)
        CPU: 96ms
     CGroup: /system.slice/loki.service
             └─55690 /opt/loki/loki-linux-amd64 -config.file=/opt/loki/loki-loc…

Jun 06 23:31:25 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…er
Jun 06 23:31:25 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…=5
Jun 06 23:31:25 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…or
Jun 06 23:31:25 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…=6
Jun 06 23:31:26 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…or
Jun 06 23:31:26 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…=4
Jun 06 23:31:26 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…ng
Jun 06 23:31:26 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…=8
Jun 06 23:31:26 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…er
Jun 06 23:31:26 controlplane loki-linux-amd64[55690]: level=debug ts=2025-06-…=5
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:/opt/loki$ ss -ntulp | grep 3100
tcp   LISTEN 0      4096                                  *:3100            *
:*    users:(("loki-linux-amd6",pid=55690,fd=8))
controlplane:/opt/loki$
```


Let's do Linux Configurations | H
Home - Dashboards - Grafana
https://killercoda.com/het-tanis/course/Linux-Labs/102-monitoring-linux-logs
KILLERCODA PLUS
Areas
Account
Validation successful
Editor
Tab 1

You've setup all the pieces, now you have to create a dashboard in Grafana and verify that everything is working end to end.

Log into Grafana (and change the password if you didn't do it earlier)

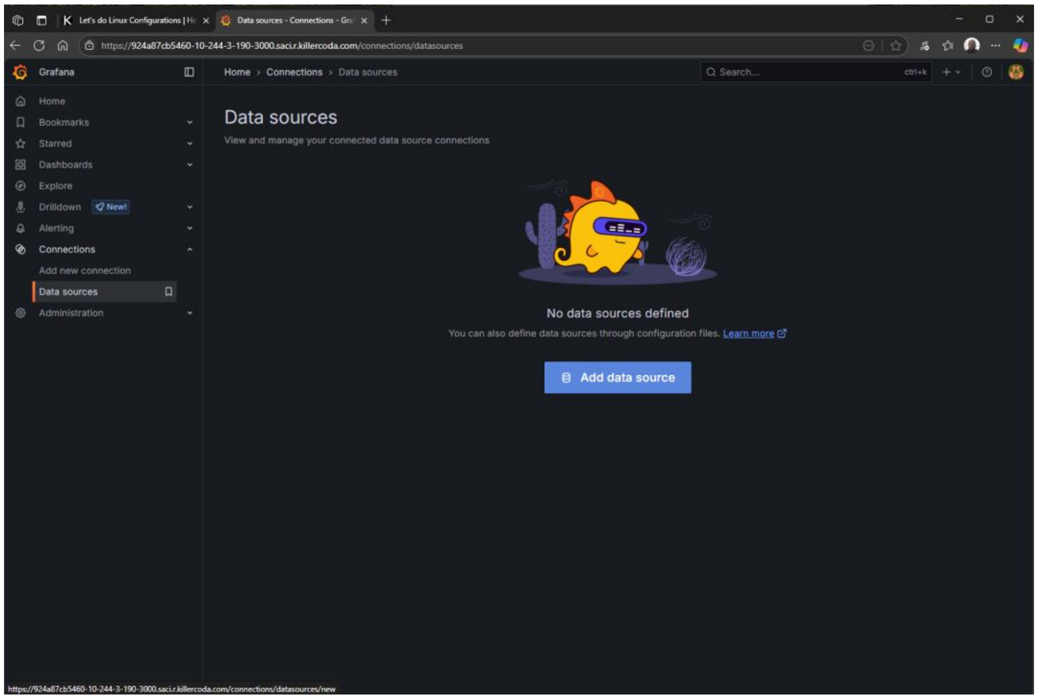
Create the datasource for Loki in the the Datasource page. URL = http://127.0.0.1:3100

Create a dashboard (import 13639) that shows the log files for your server.

Solution

▶ Solution

```
Wants=network-online.target
After=network-online.target

[Service]
ExecStart=/opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/promtail
-local-config.yaml

[Install]
WantedBy=default.targetcontrolplane:/opt/promtail$ systemctl daemon-reload
controlplane:/opt/promtail$ systemctl enable promtail.service --now
Created symlink /etc/systemd/system/default.target.wants/promtail.service → /etc
/systemd/system/promtail.service.
controlplane:/opt/promtail$ systemctl status promtail.service --no-pager
● promtail.service - Promtail Service Startup
     Loaded: loaded (/etc/systemd/system/promtail.service; enabled; preset: enab
led)
     Active: active (running) since Fri 2025-06-06 23:35:34 UTC; 22s ago
   Main PID: 57917 (promtail-linux-)
      Tasks: 8 (limit: 2614)
     Memory: 27.8M (peak: 28.1M)
        CPU: 168ms
     CGroup: /system.slice/promtail.service
             └─57917 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promt…

Jun 06 23:35:34 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…)"
Jun 06 23:35:34 controlplane promtail-linux-amd64[57917]: level=warn ts=2025-…g"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: ts=2025-06-06T23:35…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: ts=2025-06-06T23:35…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:/opt/promtail$ ps -ef | grep [p]romtail
root       57917       1  0 23:35 ?        00:00:00 /opt/promtail/promtail-linux
-amd64 -config.file=/opt/promtail/promtail-local-config.yaml
controlplane:/opt/promtail$ 
```


Let's do Linux Configurations | H
Data sources - Connections - Gra
https://924a87cb5460-10-244-3-190-3000.saci.r.killercoda.com/connections/datasources
Grafana
Home › Connections › Data sources
Search...
ctrl+k
Home
Bookmarks
Starred
Dashboards
Explore
Drilldown
New!
Alerting
Connections
Add new connection
Data sources
Administration

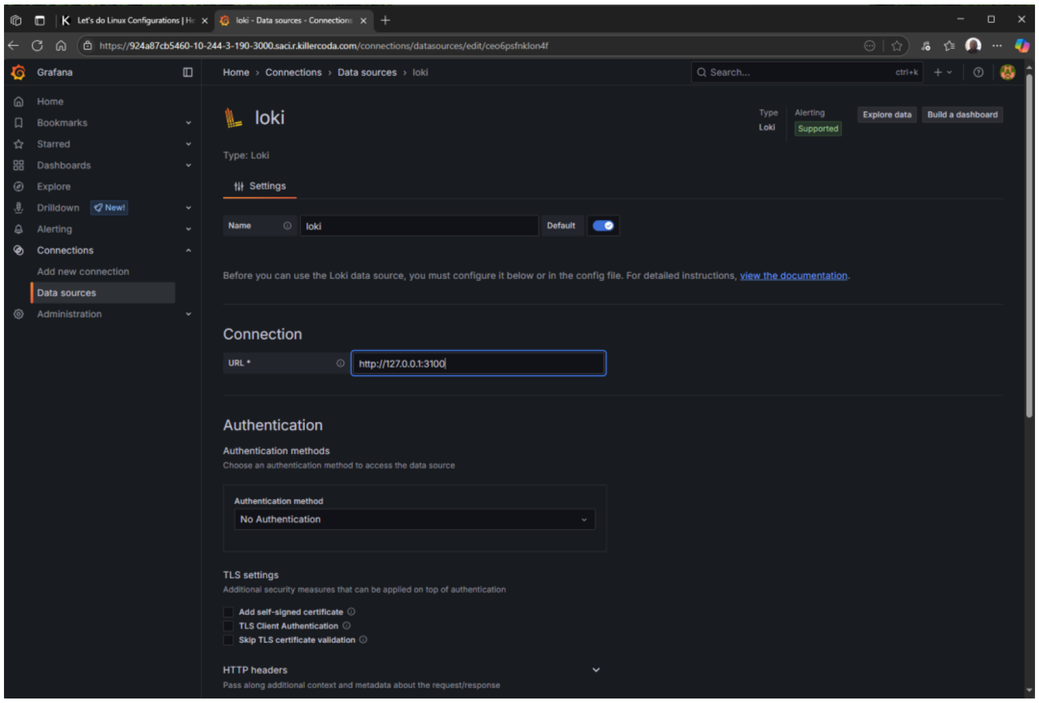
Data sources

View and manage your connected data source connections

No data sources defined

You can also define data sources through configuration files. Learn more

Add data source
https://924a87cb5460-10-244-3-190-3000.saci.r.killercoda.com/connections/datasources/new



https://924a87cb5460-10-244-3-190-3000.saci.r.killercoda.com/connections/datasources/edit/ceo6psfnklon4f

Home › Connections › Data sources › loki

loki

Type: Loki

Before you can use the Loki data source, you must configure it below or in the config file. For detailed instructions, view the documentation.

Name: loki   Default: on

Connection

URL *: http://127.0.0.1:3100

Authentication

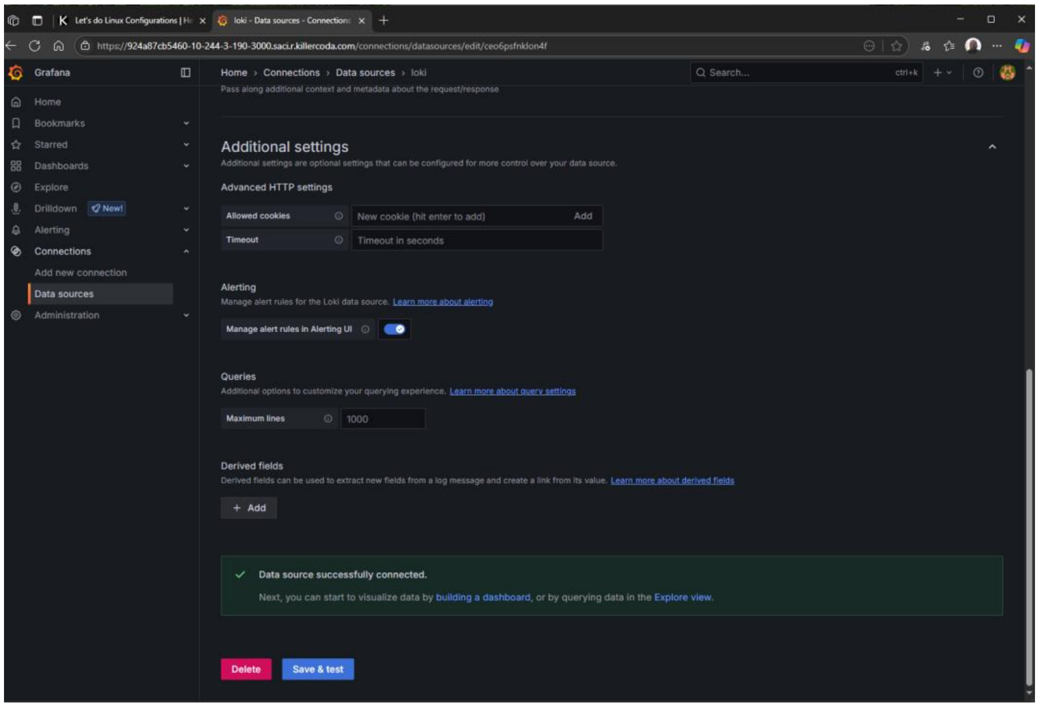
Authentication methods
Choose an authentication method to access the data source

Authentication method: No Authentication

TLS settings
Additional security measures that can be applied on top of authentication

- Add self-signed certificate
- TLS Client Authentication
- Skip TLS certificate validation

HTTP headers
Pass along additional context and metadata about the request/response



Additional settings
Additional settings are optional settings that can be configured for more control over your data source.

Advanced HTTP settings

Allowed cookies: New cookie (hit enter to add)   Add
Timeout: Timeout in seconds

Alerting
Manage alert rules for the Loki data source. Learn more about alerting

Manage alert rules in Alerting UI: on

Queries
Additional options to customize your querying experience. Learn more about query settings

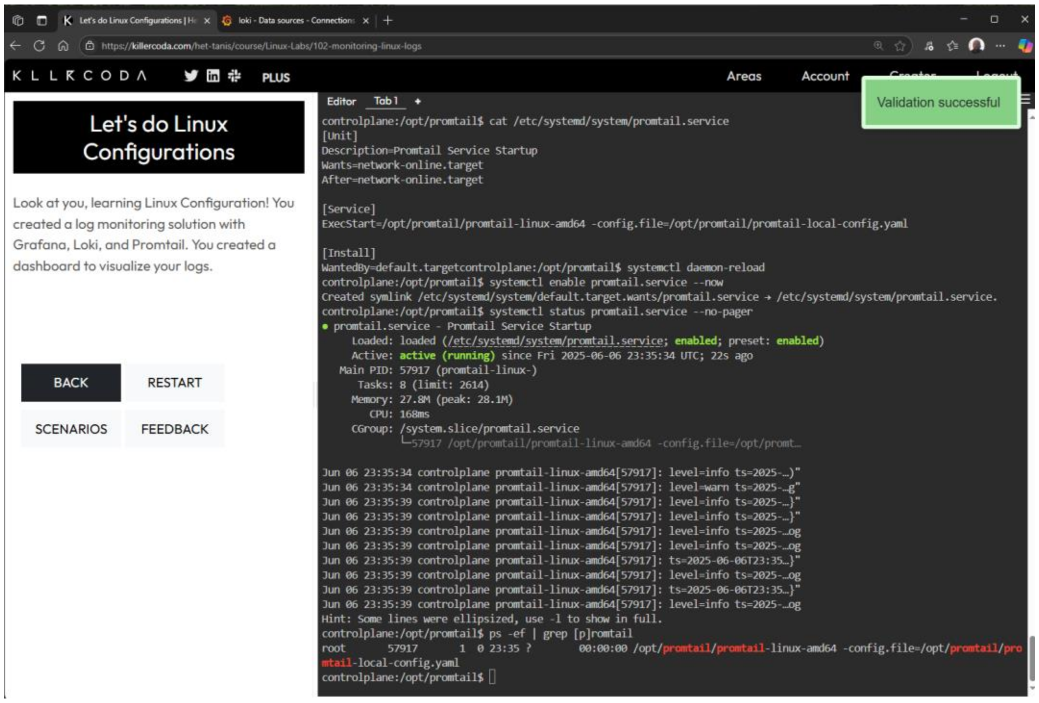
Maximum lines: 1000

Derived fields
Derived fields can be used to extract new fields from a log message and create a link from its value. Learn more about derived fields

+ Add

✓ Data source successfully connected.
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.

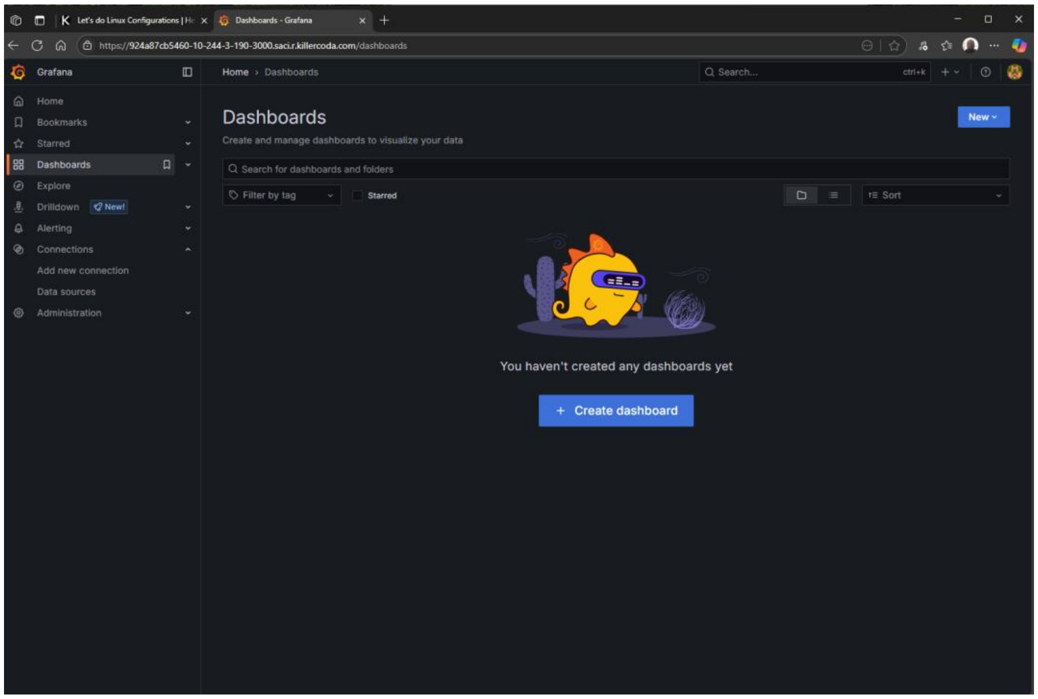
Delete   Save & test



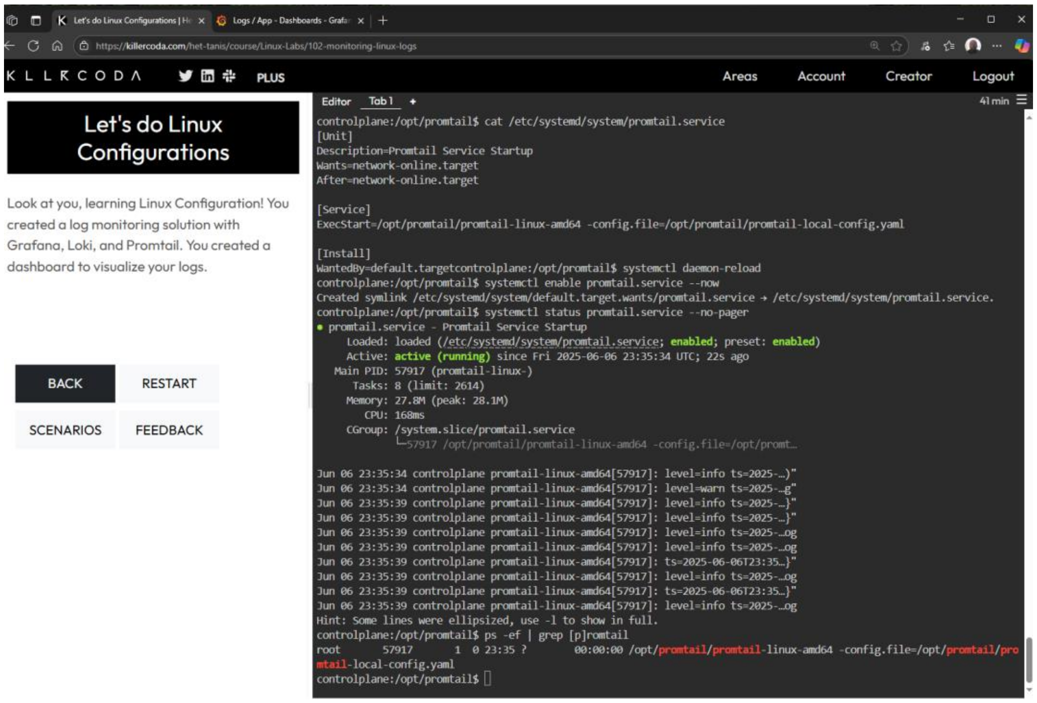
Let's do Linux Configurations | Ha
loki - Data sources - Connections
https://killercoda.com/het-tanis/course/Linux-Labs/102-monitoring-linux-logs
KILLERCODA
PLUS
Areas
Account
Validation successful
Let's do Linux Configurations
BACK
RESTART
SCENARIOS
FEEDBACK
Editor
Tab 1

Look at you, learning Linux Configuration! You created a log monitoring solution with Grafana, Loki, and Promtail. You created a dashboard to visualize your logs.

```
controlplane:/opt/promtail$ cat /etc/systemd/system/promtail.service
[Unit]
Description=Promtail Service Startup
Wants=network-online.target
After=network-online.target

[Service]
ExecStart=/opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/promtail-local-config.yaml

[Install]
WantedBy=default.targetcontrolplane:/opt/promtail$ systemctl daemon-reload
controlplane:/opt/promtail$ systemctl enable promtail.service --now
Created symlink /etc/systemd/system/default.target.wants/promtail.service → /etc/systemd/system/promtail.service.
controlplane:/opt/promtail$ systemctl status promtail.service --no-pager
● promtail.service - Promtail Service Startup
     Loaded: loaded (/etc/systemd/system/promtail.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-06-06 23:35:34 UTC; 22s ago
   Main PID: 57917 (promtail-linux-)
      Tasks: 8 (limit: 2614)
     Memory: 27.8M (peak: 28.1M)
        CPU: 168ms
     CGroup: /system.slice/promtail.service
             └─57917 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promt…

Jun 06 23:35:34 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…)"
Jun 06 23:35:34 controlplane promtail-linux-amd64[57917]: level=warn ts=2025-…g"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: ts=2025-06-06T23:35…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: ts=2025-06-06T23:35…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:/opt/promtail$ ps -ef | grep [p]romtail
root       57917       1  0 23:35 ?        00:00:00 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/pro
mtail-local-config.yaml
controlplane:/opt/promtail$
```



Let's do Linux Configurations | Ha
Dashboards - Grafana
https://924a87cb5460-10-244-3-190-3000.saci.r.killercoda.com/dashboards
Grafana
Home › Dashboards
Search...
ctrl+k
Home
Bookmarks
Starred
Dashboards
Explore
Drilldown
New!
Alerting
Connections
Add new connection
Data sources
Administration
Dashboards
Create and manage dashboards to visualize your data
New
Search for dashboards and folders
Filter by tag
Starred
Sort
You haven't created any dashboards yet
+ Create dashboard


Let's do Linux Configurations
Logs / App - Dashboards - Grafana
https://killercoda.com/het-tanis/course/Linux-Labs/102-monitoring-linux-logs
KILLERCODA
PLUS
Areas
Account
Creator
Logout
Editor
Tab 1
41 min
BACK
RESTART
SCENARIOS
FEEDBACK

Let's do Linux Configurations

Look at you, learning Linux Configuration! You created a log monitoring solution with Grafana, Loki, and Promtail. You created a dashboard to visualize your logs.

```
controlplane:/opt/promtail$ cat /etc/systemd/system/promtail.service
[Unit]
Description=Promtail Service Startup
Wants=network-online.target
After=network-online.target

[Service]
ExecStart=/opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/promtail-local-config.yaml

[Install]
WantedBy=default.targetcontrolplane:/opt/promtail$ systemctl daemon-reload
controlplane:/opt/promtail$ systemctl enable promtail.service --now
Created symlink /etc/systemd/system/default.target.wants/promtail.service → /etc/systemd/system/promtail.service.
controlplane:/opt/promtail$ systemctl status promtail.service --no-pager
● promtail.service - Promtail Service Startup
     Loaded: loaded (/etc/systemd/system/promtail.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-06-06 23:35:34 UTC; 22s ago
   Main PID: 57917 (promtail-linux-)
      Tasks: 8 (limit: 2614)
     Memory: 27.8M (peak: 28.1M)
        CPU: 168ms
     CGroup: /system.slice/promtail.service
             └─57917 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promt…

Jun 06 23:35:34 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…)"
Jun 06 23:35:34 controlplane promtail-linux-amd64[57917]: level=warn ts=2025-…g"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: ts=2025-06-06T23:35…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: ts=2025-06-06T23:35…}"
Jun 06 23:35:39 controlplane promtail-linux-amd64[57917]: level=info ts=2025-…og
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:/opt/promtail$ ps -ef | grep [p]romtail
root       57917       1  0 23:35 ?        00:00:00 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/pro
mtail-local-config.yaml
controlplane:/opt/promtail$ 
```

Agents forward to a centralized platform 2 of 4

Does the lab work correctly, and do you understand the data flow?

Yes

While still in the lab

cd /answers

python3 loki-write.py #Do this a few times

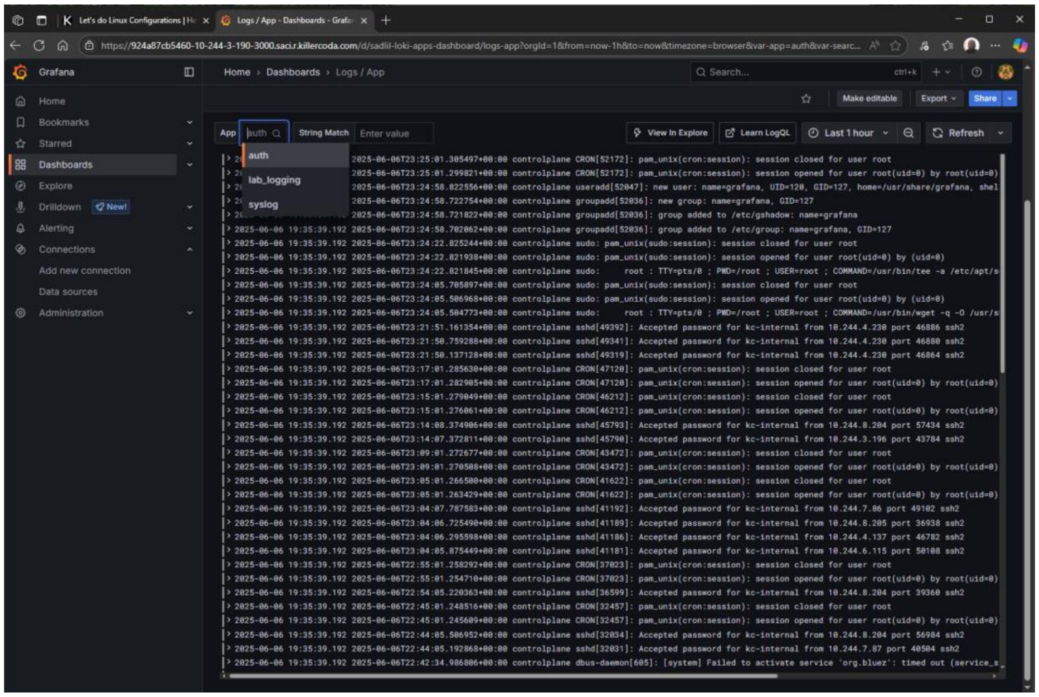
Refresh your Grafana and change the app to lab_logging

Can you see it in your Grafana?

At first I did not see it with Microsoft edge

-but then realized I needed to scroll the page to the top

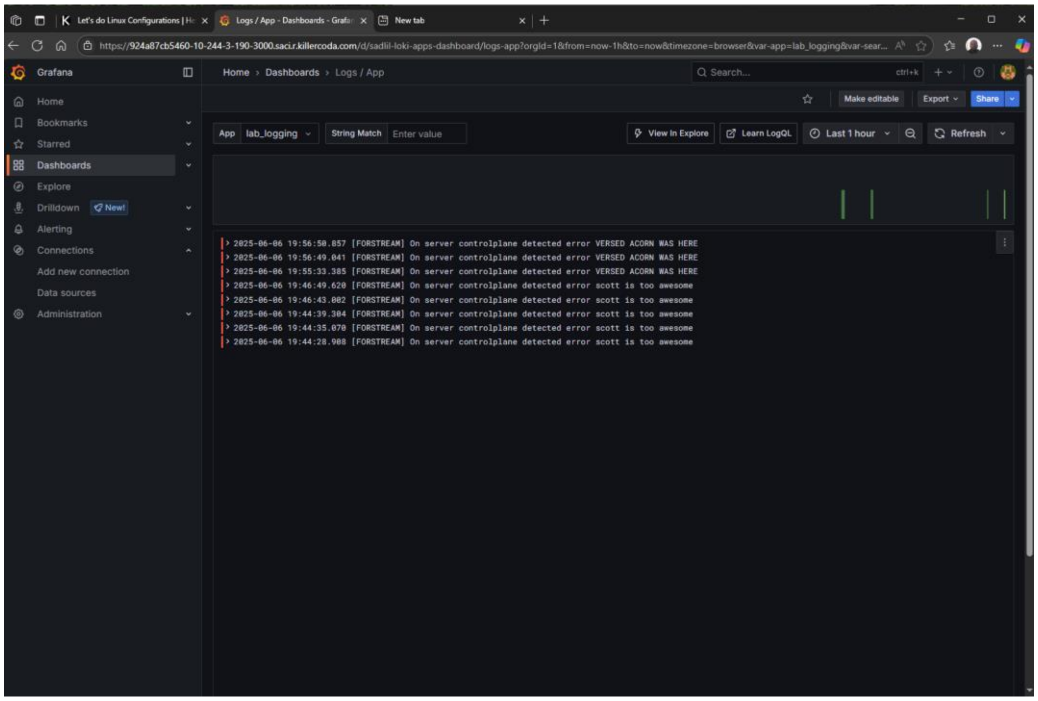
And I see it with google chrome



https://924a87cb5460-10-244-3-190-3000.saci.r.killercoda.com/d/sadlil-loki-apps-dashboard/logs-app?orgId=1&from=now-1h&to=now&timezone=browser&var-app=auth&var-searc...

App: auth | String Match: Enter value

```
2025-06-06T23:25:01.305497+00:00 controlplane CRON[52172]: pam_unix(cron:session): session closed for user root
2025-06-06T23:25:01.299821+00:00 controlplane CRON[52172]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-06T23:24:58.822556+00:00 controlplane useradd[52047]: new user: name=grafana, UID=120, GID=127, home=/usr/share/grafana, shel
2025-06-06T23:24:58.722754+00:00 controlplane groupadd[52036]: new group: name=grafana, GID=127
2025-06-06T23:24:58.721822+00:00 controlplane groupadd[52036]: group added to /etc/gshadow: name=grafana
> 2025-06-06 19:35:39.192 2025-06-06T23:24:58.702062+00:00 controlplane groupadd[52036]: group added to /etc/group: name=grafana, GID=127
> 2025-06-06 19:35:39.192 2025-06-06T23:24:22.825244+00:00 controlplane sudo: pam_unix(sudo:session): session closed for user root
> 2025-06-06 19:35:39.192 2025-06-06T23:24:22.821938+00:00 controlplane sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=0)
> 2025-06-06 19:35:39.192 2025-06-06T23:24:22.821845+00:00 controlplane sudo:     root : TTY=pts/0 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/tee -a /etc/apt/s
> 2025-06-06 19:35:39.192 2025-06-06T23:24:05.705897+00:00 controlplane sudo: pam_unix(sudo:session): session closed for user root
> 2025-06-06 19:35:39.192 2025-06-06T23:24:05.506968+00:00 controlplane sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=0)
> 2025-06-06 19:35:39.192 2025-06-06T23:24:05.504773+00:00 controlplane sudo:     root : TTY=pts/0 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/wget -q -O /usr/s
> 2025-06-06 19:35:39.192 2025-06-06T23:21:51.161354+00:00 controlplane sshd[49392]: Accepted password for kc-internal from 10.244.4.230 port 46886 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T23:21:50.759288+00:00 controlplane sshd[49341]: Accepted password for kc-internal from 10.244.4.230 port 46880 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T23:21:50.137128+00:00 controlplane sshd[49319]: Accepted password for kc-internal from 10.244.4.230 port 46864 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T23:17:01.285630+00:00 controlplane CRON[47120]: pam_unix(cron:session): session closed for user root
> 2025-06-06 19:35:39.192 2025-06-06T23:17:01.282905+00:00 controlplane CRON[47120]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
> 2025-06-06 19:35:39.192 2025-06-06T23:15:01.279049+00:00 controlplane CRON[46212]: pam_unix(cron:session): session closed for user root
> 2025-06-06 19:35:39.192 2025-06-06T23:15:01.276061+00:00 controlplane CRON[46212]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
> 2025-06-06 19:35:39.192 2025-06-06T23:14:08.374906+00:00 controlplane sshd[45793]: Accepted password for kc-internal from 10.244.8.204 port 57434 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T23:14:07.372811+00:00 controlplane sshd[45790]: Accepted password for kc-internal from 10.244.3.196 port 43784 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T23:09:01.272677+00:00 controlplane CRON[43472]: pam_unix(cron:session): session closed for user root
> 2025-06-06 19:35:39.192 2025-06-06T23:09:01.270508+00:00 controlplane CRON[43472]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
> 2025-06-06 19:35:39.192 2025-06-06T23:05:01.266500+00:00 controlplane CRON[41622]: pam_unix(cron:session): session closed for user root
> 2025-06-06 19:35:39.192 2025-06-06T23:05:01.263429+00:00 controlplane CRON[41622]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
> 2025-06-06 19:35:39.192 2025-06-06T23:04:07.787583+00:00 controlplane sshd[41192]: Accepted password for kc-internal from 10.244.7.86 port 49102 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T23:04:06.725490+00:00 controlplane sshd[41189]: Accepted password for kc-internal from 10.244.8.205 port 36938 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T23:04:06.295598+00:00 controlplane sshd[41186]: Accepted password for kc-internal from 10.244.4.137 port 46782 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T23:04:05.875449+00:00 controlplane sshd[41181]: Accepted password for kc-internal from 10.244.6.115 port 50108 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T22:55:01.258292+00:00 controlplane CRON[37023]: pam_unix(cron:session): session closed for user root
> 2025-06-06 19:35:39.192 2025-06-06T22:55:01.254710+00:00 controlplane CRON[37023]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
> 2025-06-06 19:35:39.192 2025-06-06T22:54:05.220363+00:00 controlplane sshd[36599]: Accepted password for kc-internal from 10.244.8.204 port 39360 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T22:45:01.248516+00:00 controlplane CRON[32457]: pam_unix(cron:session): session closed for user root
> 2025-06-06 19:35:39.192 2025-06-06T22:45:01.245609+00:00 controlplane CRON[32457]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
> 2025-06-06 19:35:39.192 2025-06-06T22:44:05.506952+00:00 controlplane sshd[32034]: Accepted password for kc-internal from 10.244.8.204 port 56984 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T22:44:05.192868+00:00 controlplane sshd[32031]: Accepted password for kc-internal from 10.244.7.87 port 40504 ssh2
> 2025-06-06 19:35:39.192 2025-06-06T22:42:34.986806+00:00 controlplane dbus-daemon[605]: [system] Failed to activate service 'org.bluez': timed out (service_s
```



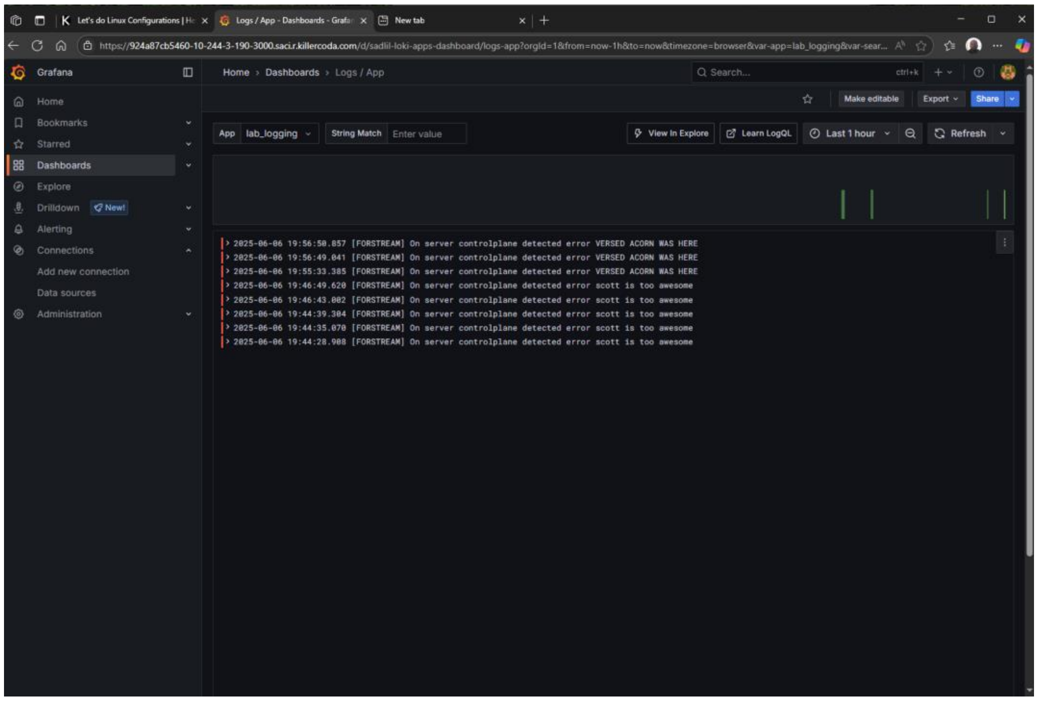
https://924a87cb5460-10-244-3-190-3000.saci.r.killercoda.com/d/sadlil-loki-apps-dashboard/logs-app?orgId=1&from=now-1h&to=now&timezone=browser&var-app=lab_logging&var-sear...

App: lab_logging | String Match: Enter value

```
> 2025-06-06 19:56:50.857 [FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE
> 2025-06-06 19:56:49.041 [FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE
> 2025-06-06 19:55:33.385 [FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE
> 2025-06-06 19:46:49.620 [FORSTREAM] On server controlplane detected error scott is too awesome
> 2025-06-06 19:46:43.002 [FORSTREAM] On server controlplane detected error scott is too awesome
> 2025-06-06 19:44:39.304 [FORSTREAM] On server controlplane detected error scott is too awesome
> 2025-06-06 19:44:35.070 [FORSTREAM] On server controlplane detected error scott is too awesome
> 2025-06-06 19:44:28.908 [FORSTREAM] On server controlplane detected error scott is too awesome
```

<u>Agents forward to a centralized platform 3 of 4</u>

Can you modify the file loki-write.py to say something related to your name?



```
2025-06-06 19:56:50.857 [FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE
2025-06-06 19:56:49.041 [FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE
2025-06-06 19:55:33.385 [FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE
2025-06-06 19:46:49.620 [FORSTREAM] On server controlplane detected error scott is too awesome
2025-06-06 19:46:43.002 [FORSTREAM] On server controlplane detected error scott is too awesome
2025-06-06 19:44:39.304 [FORSTREAM] On server controlplane detected error scott is too awesome
2025-06-06 19:44:35.070 [FORSTREAM] On server controlplane detected error scott is too awesome
2025-06-06 19:44:28.908 [FORSTREAM] On server controlplane detected error scott is too awesome
```

<u>Agents forward to a centralized platform 4 of 4</u>

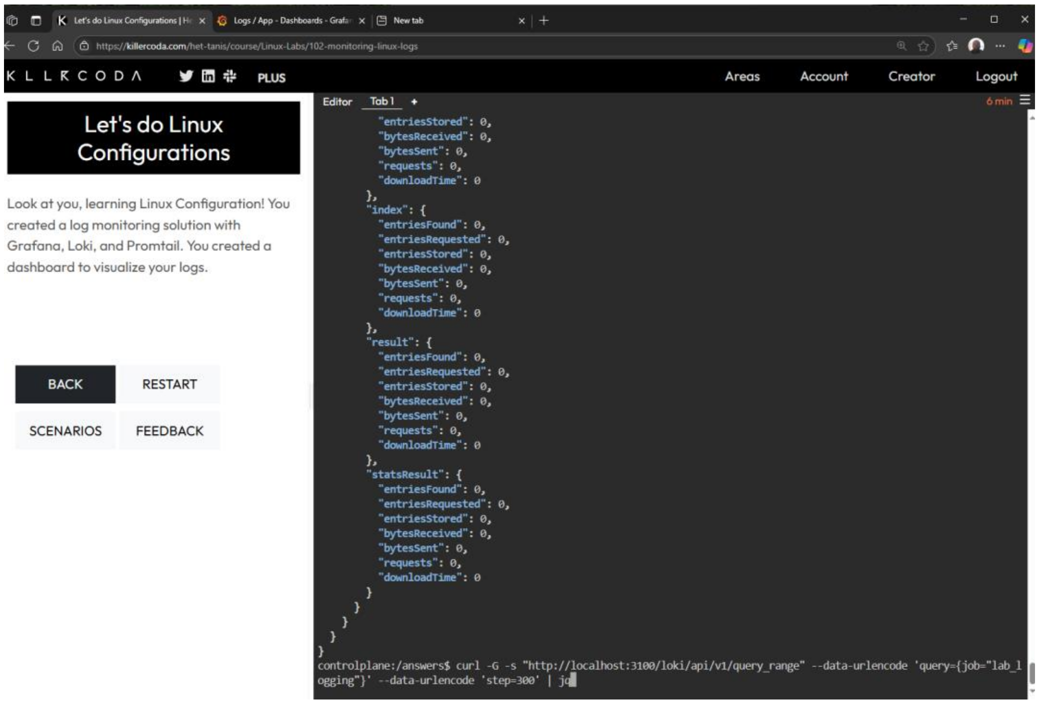
Run this bash snippet and see if you can see your loki-writes

```
curl -G -s "http://localhost:3100/loki/api/v1/query_range" \
--data-urlencode 'query=sum(rate({job="lab_logging"}[10m])) by (level)' \
--data-urlencode 'step=300' | jq
```

Can you modify that to see the actual entries?
https://grafana.com/docs/loki/latest/reference/loki-http-api/#query-logs-within-a-range-of-time

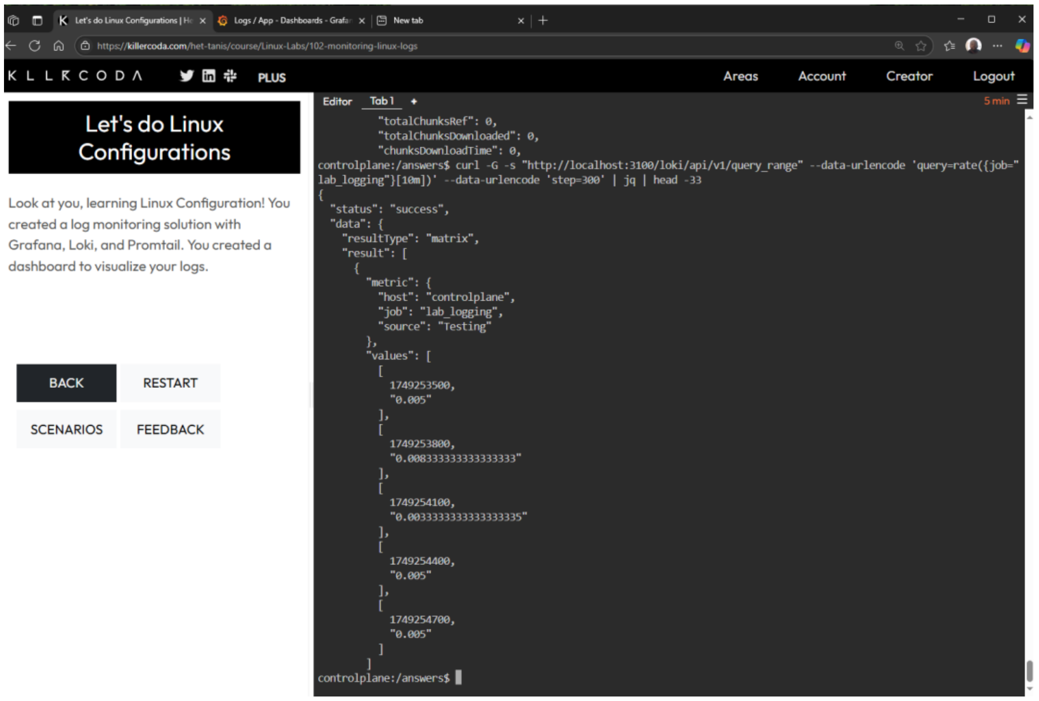
```
controlplane:/answers$ curl -G -s "http://localhost:3100/loki/api/v1/query_range" --data-urlencode 'query={job="lab_l
ogging"}' --data-urlencode 'step=300' | jq
{
 "status": "success",
 "data": {
  "resultType": "streams",
  "result": [
   {
    "stream": {
     "host": "controlplane",
     "job": "lab_logging",
     "source": "Testing"
    },
    "values": [
     [
      "1749254210857868000",
      "[FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE"
     ],
     [
      "1749254209041840000",
      "[FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE"
     ],
     [
      "1749254133385143000",
      "[FORSTREAM] On server controlplane detected error VERSED ACORN WAS HERE"
     ],
     [
      "1749253609620244000",
      "[FORSTREAM] On server controlplane detected error scott is too awesome"
     ],
     [
      "1749253603002031000",
      "[FORSTREAM] On server controlplane detected error scott is too awesome"
     ],
     [
      "1749253479304208000",
      "[FORSTREAM] On server controlplane detected error scott is too awesome"
     ],
     [
      "1749253475070708000",
      "[FORSTREAM] On server controlplane detected error scott is too awesome"
     ],
     [
      "1749253468908918000",
      "[FORSTREAM] On server controlplane detected error scott is too awesome"
     ]
    ]
   }
  ],
  "stats": {
   "summary": {
    "bytesProcessedPerSecond": 139323,
    "linesProcessedPerSecond": 1979,
    "totalBytesProcessed": 563,
    "totalLinesProcessed": 8,
    "execTime": 0.004041,
    "queueTime": 0.000075,
    "subqueries": 0,
    "totalEntriesReturned": 8,
    "splits": 3,
    "shards": 0,
    "totalPostFilterLines": 8,
    "totalStructuredMetadataBytesProcessed": 0
   },
   "querier": {
    "store": {
     "totalChunksRef": 0,
     "totalChunksDownloaded": 0,
     "chunksDownloadTime": 0,
     "chunk": {
      "headChunkBytes": 0,
      "headChunkLines": 0,
      "decompressedBytes": 0,
      "decompressedLines": 0,
      "compressedBytes": 0,
      "totalDuplicates": 0,
      "postFilterLines": 0,
      "headChunkStructuredMetadataBytes": 0,
      "decompressedStructuredMetadataBytes": 0
```

```
        },
        "chunkRefsFetchTime": 0
      }
    },
    "ingester": {
      "totalReached": 3,
      "totalChunksMatched": 1,
      "totalBatches": 4,
      "totalLinesSent": 8,
      "store": {
        "totalChunksRef": 0,
        "totalChunksDownloaded": 0,
        "chunksDownloadTime": 0,
        "chunk": {
          "headChunkBytes": 563,
          "headChunkLines": 8,
          "decompressedBytes": 0,
          "decompressedLines": 0,
          "compressedBytes": 0,
          "totalDuplicates": 0,
          "postFilterLines": 8,
          "headChunkStructuredMetadataBytes": 0,
          "decompressedStructuredMetadataBytes": 0
        },
        "chunkRefsFetchTime": 439303
      }
    },
    "cache": {
      "chunk": {
        "entriesFound": 0,
        "entriesRequested": 0,
        "entriesStored": 0,
        "bytesReceived": 0,
        "bytesSent": 0,
        "requests": 0,
        "downloadTime": 0
      },
      "index": {
        "entriesFound": 0,
        "entriesRequested": 0,
        "entriesStored": 0,
        "bytesReceived": 0,
        "bytesSent": 0,
        "requests": 0,
        "downloadTime": 0
      },
      "result": {
        "entriesFound": 0,
        "entriesRequested": 0,
        "entriesStored": 0,
        "bytesReceived": 0,
        "bytesSent": 0,
        "requests": 0,
        "downloadTime": 0
      },
      "statsResult": {
        "entriesFound": 0,
        "entriesRequested": 0,
        "entriesStored": 0,
        "bytesReceived": 0,
        "bytesSent": 0,
        "requests": 0,
        "downloadTime": 0
      }
    }
  }
 }
}
controlplane:/answers$
```



Look at you, learning Linux Configuration! You created a log monitoring solution with Grafana, Loki, and Promtail. You created a dashboard to visualize your logs.

```
          "entriesStored": 0,
          "bytesReceived": 0,
          "bytesSent": 0,
          "requests": 0,
          "downloadTime": 0
        },
        "index": {
          "entriesFound": 0,
          "entriesRequested": 0,
          "entriesStored": 0,
          "bytesReceived": 0,
          "bytesSent": 0,
          "requests": 0,
          "downloadTime": 0
        },
        "result": {
          "entriesFound": 0,
          "entriesRequested": 0,
          "entriesStored": 0,
          "bytesReceived": 0,
          "bytesSent": 0,
          "requests": 0,
          "downloadTime": 0
        },
        "statsResult": {
          "entriesFound": 0,
          "entriesRequested": 0,
          "entriesStored": 0,
          "bytesReceived": 0,
          "bytesSent": 0,
          "requests": 0,
          "downloadTime": 0
        }
      }
    }
  }
}
controlplane:/answers$ curl -G -s "http://localhost:3100/loki/api/v1/query_range" --data-urlencode 'query={job="lab_l
ogging"}' --data-urlencode 'step=300' | jq
```



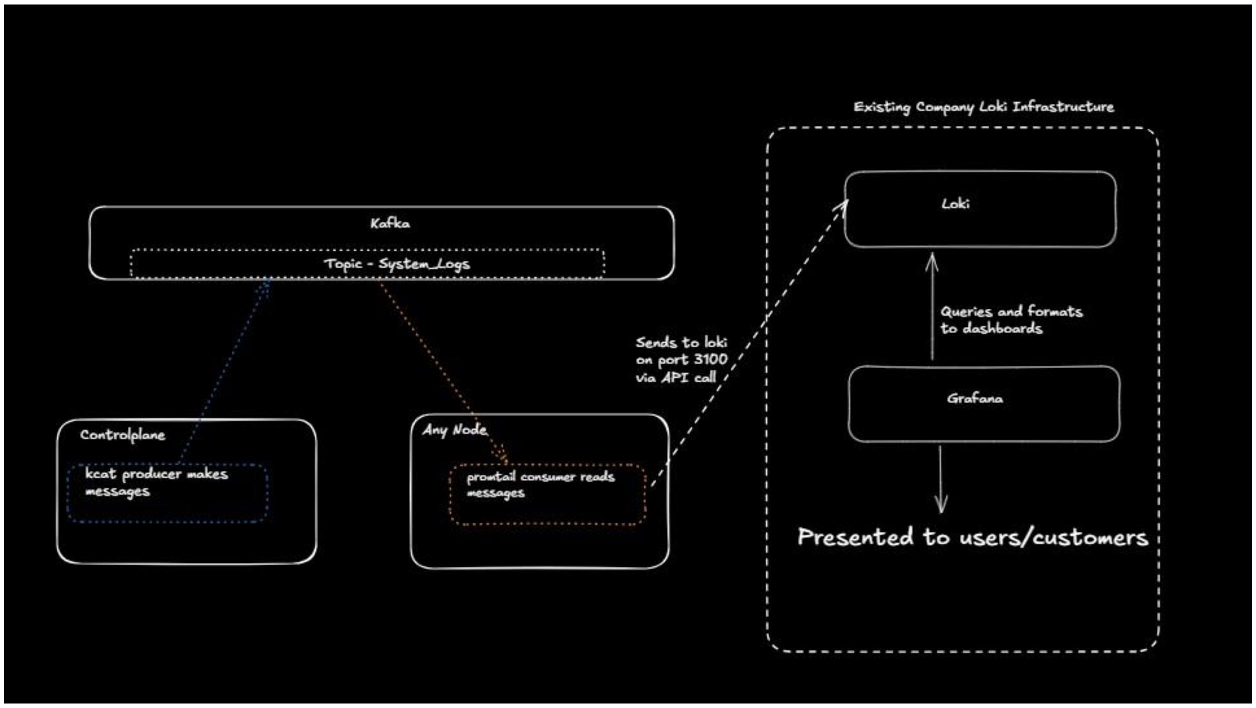
Look at you, learning Linux Configuration! You created a log monitoring solution with Grafana, Loki, and Promtail. You created a dashboard to visualize your logs.

```
          "totalChunksRef": 0,
          "totalChunksDownloaded": 0,
          "chunksDownloadTime": 0,
controlplane:/answers$ curl -G -s "http://localhost:3100/loki/api/v1/query_range" --data-urlencode 'query=rate({job="
lab_logging"}[10m])' --data-urlencode 'step=300' | jq | head -33
{
  "status": "success",
  "data": {
    "resultType": "matrix",
    "result": [
      {
        "metric": {
          "host": "controlplane",
          "job": "lab_logging",
          "source": "Testing"
        },
        "values": [
          [
            1749253500,
            "0.005"
          ],
          [
            1749253800,
            "0.008333333333333333"
          ],
          [
            1749254100,
            "0.0033333333333333335"
          ],
          [
            1749254400,
            "0.005"
          ],
          [
            1749254700,
            "0.005"
          ]
        ]
controlplane:/answers$
```

Message Queues (Event Bus) for log aggregation and propagation

1. Apache Kafka is not the only message queue, but it is extremely popular (found in 80% for Fortune 100 companies… or 80 of them). Read about the use cases here: https://kafka.apache.org/uses

2. Review our diagram here. Maybe we're testing kafka and want to integrate it to the existing infrastructure. Maybe we have a remote location that we need to reliably catch logs in real time and then move them remote. There are many reasons to use this.

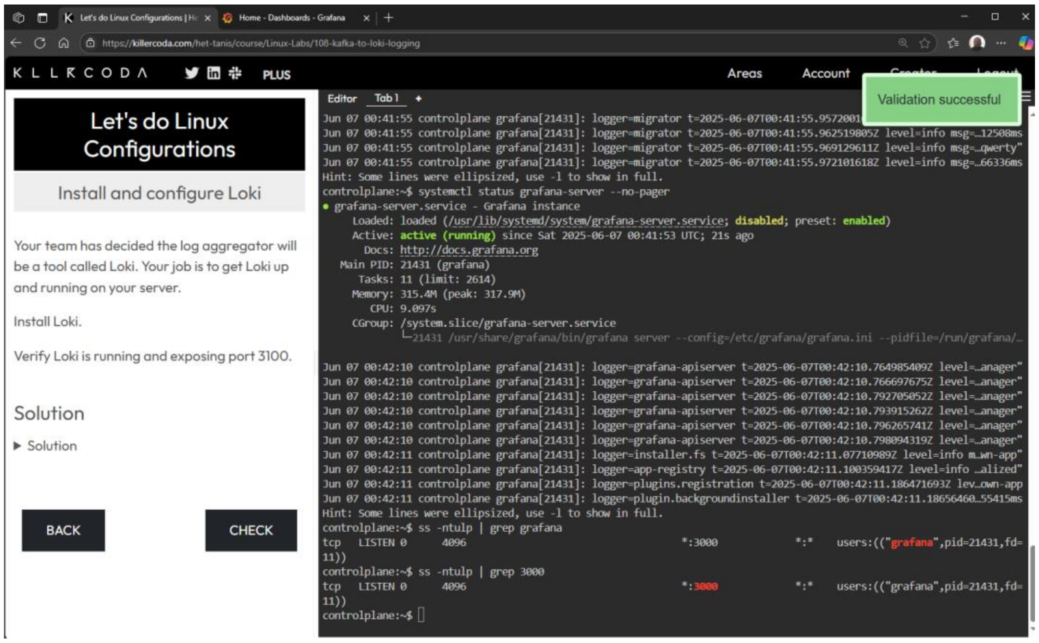

Existing Company Loki Infrastructure
Kafka
Topic - System_Logs
Loki
Sends to loki on port 3100 via API call
Queries and formats to dashboards
Grafana
Controlplane
kcat producer makes messages
Any Node
promtail consumer reads messages
Presented to users/customers

Message Queues (Event Bus) for log aggregation and propagation

3. Complete the killercoda lab found here: https://killercoda.com/het-tanis/course/Linux-Labs/108-kafka-to-loki-logging

Did you get it all to work?

Yes



Let's do Linux Configurations

Install and configure Loki

Your team has decided the log aggregator will be a tool called Loki. Your job is to get Loki up and running on your server.

Install Loki.

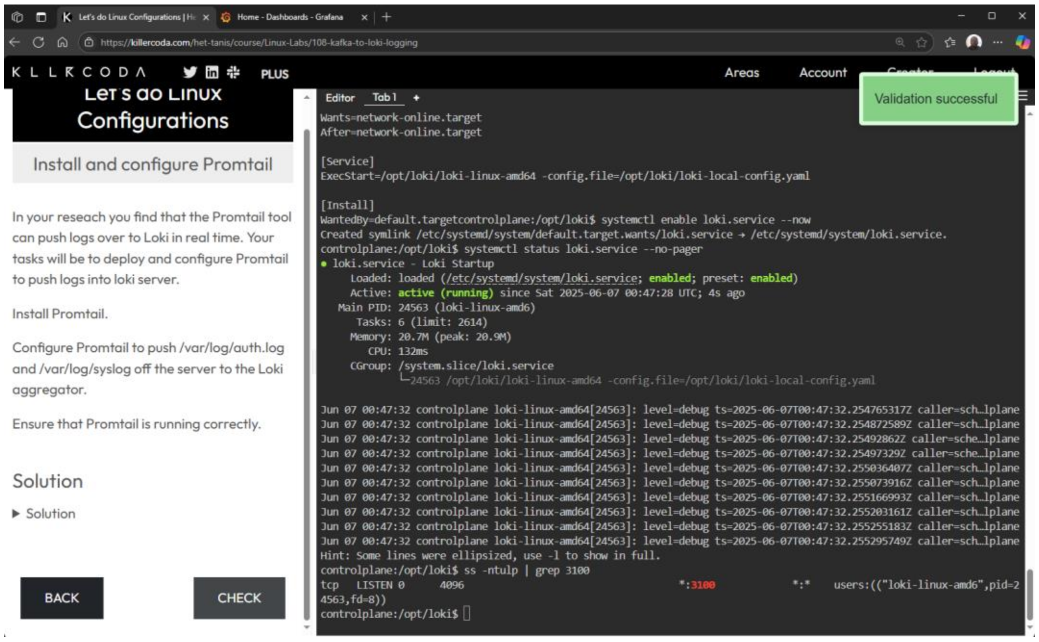
Verify Loki is running and exposing port 3100.

Solution

▶ Solution

Validation successful

```
Jun 07 00:41:55 controlplane grafana[21431]: logger=migrator t=2025-06-07T00:41:55.9572001
Jun 07 00:41:55 controlplane grafana[21431]: logger=migrator t=2025-06-07T00:41:55.962519805Z level=info msg=…12508ms
Jun 07 00:41:55 controlplane grafana[21431]: logger=migrator t=2025-06-07T00:41:55.969129611Z level=info msg=…qwerty"
Jun 07 00:41:55 controlplane grafana[21431]: logger=migrator t=2025-06-07T00:41:55.972101618Z level=info msg=…66336ms
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:~$ systemctl status grafana-server --no-pager
● grafana-server.service - Grafana instance
     Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; preset: enabled)
     Active: active (running) since Sat 2025-06-07 00:41:53 UTC; 21s ago
       Docs: http://docs.grafana.org
   Main PID: 21431 (grafana)
      Tasks: 11 (limit: 2614)
     Memory: 315.4M (peak: 317.9M)
        CPU: 9.097s
     CGroup: /system.slice/grafana-server.service
             └─21431 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/…

Jun 07 00:42:10 controlplane grafana[21431]: logger=grafana-apiserver t=2025-06-07T00:42:10.764985409Z level=…anager"
Jun 07 00:42:10 controlplane grafana[21431]: logger=grafana-apiserver t=2025-06-07T00:42:10.766697675Z level=…anager"
Jun 07 00:42:10 controlplane grafana[21431]: logger=grafana-apiserver t=2025-06-07T00:42:10.792705052Z level=…anager"
Jun 07 00:42:10 controlplane grafana[21431]: logger=grafana-apiserver t=2025-06-07T00:42:10.793915262Z level=…anager"
Jun 07 00:42:10 controlplane grafana[21431]: logger=grafana-apiserver t=2025-06-07T00:42:10.796265741Z level=…anager"
Jun 07 00:42:10 controlplane grafana[21431]: logger=grafana-apiserver t=2025-06-07T00:42:10.798094319Z level=…anager"
Jun 07 00:42:11 controlplane grafana[21431]: logger=installer.fs t=2025-06-07T00:42:11.07710989Z level=info m…wn-app"
Jun 07 00:42:11 controlplane grafana[21431]: logger=app-registry t=2025-06-07T00:42:11.100359417Z level=info …alized"
Jun 07 00:42:11 controlplane grafana[21431]: logger=plugins.registration t=2025-06-07T00:42:11.186471693Z lev…own-app
Jun 07 00:42:11 controlplane grafana[21431]: logger=plugin.backgroundinstaller t=2025-06-07T00:42:11.18656460…55415ms
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:~$ ss -ntulp | grep grafana
tcp   LISTEN 0      4096                            *:3000            *:*    users:(("grafana",pid=21431,fd=
11))
controlplane:~$ ss -ntulp | grep 3000
tcp   LISTEN 0      4096                            *:3000            *:*    users:(("grafana",pid=21431,fd=
11))
controlplane:~$
```



Let's do Linux Configurations

Install and configure Promtail

In your reseach you find that the Promtail tool can push logs over to Loki in real time. Your tasks will be to deploy and configure Promtail to push logs into loki server.

Install Promtail.

Configure Promtail to push /var/log/auth.log and /var/log/syslog off the server to the Loki aggregator.

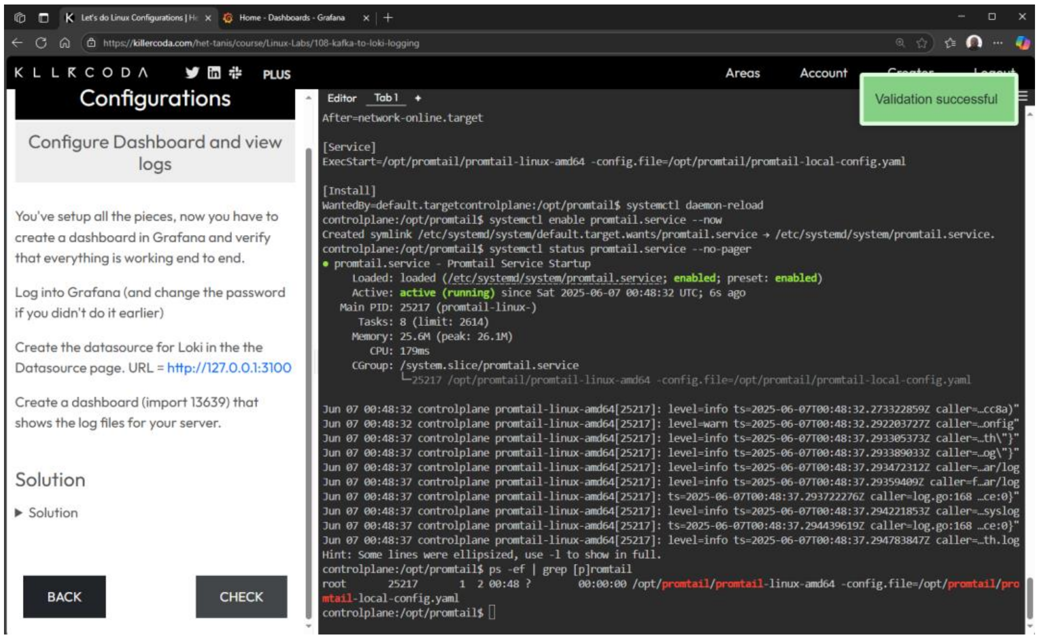
Ensure that Promtail is running correctly.

Solution

▶ Solution

Validation successful

```
Wants=network-online.target
After=network-online.target

[Service]
ExecStart=/opt/loki/loki-linux-amd64 -config.file=/opt/loki/loki-local-config.yaml

[Install]
WantedBy=default.targetcontrolplane:/opt/loki$ systemctl enable loki.service --now
Created symlink /etc/systemd/system/default.target.wants/loki.service → /etc/systemd/system/loki.service.
controlplane:/opt/loki$ systemctl status loki.service --no-pager
● loki.service - Loki Startup
     Loaded: loaded (/etc/systemd/system/loki.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-06-07 00:47:28 UTC; 4s ago
   Main PID: 24563 (loki-linux-amd6)
      Tasks: 6 (limit: 2614)
     Memory: 20.7M (peak: 20.9M)
        CPU: 132ms
     CGroup: /system.slice/loki.service
             └─24563 /opt/loki/loki-linux-amd64 -config.file=/opt/loki/loki-local-config.yaml

Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.254765317Z caller=sch…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.254872589Z caller=sch…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.25492862Z caller=sche…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.25497329Z caller=sche…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.255036407Z caller=sch…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.255073916Z caller=sch…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.255166993Z caller=sch…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.255203161Z caller=sch…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.255255183Z caller=sch…lplane
Jun 07 00:47:32 controlplane loki-linux-amd64[24563]: level=debug ts=2025-06-07T00:47:32.255295749Z caller=sch…lplane
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:/opt/loki$ ss -ntulp | grep 3100
tcp   LISTEN 0      4096                            *:3100            *:*    users:(("loki-linux-amd6",pid=2
4563,fd=8))
controlplane:/opt/loki$
```



Configurations

Configure Dashboard and view logs

You've setup all the pieces, now you have to create a dashboard in Grafana and verify that everything is working end to end.

Log into Grafana (and change the password if you didn't do it earlier)

Create the datasource for Loki in the the Datasource page. URL = http://127.0.0.1:3100

Create a dashboard (import 13639) that shows the log files for your server.

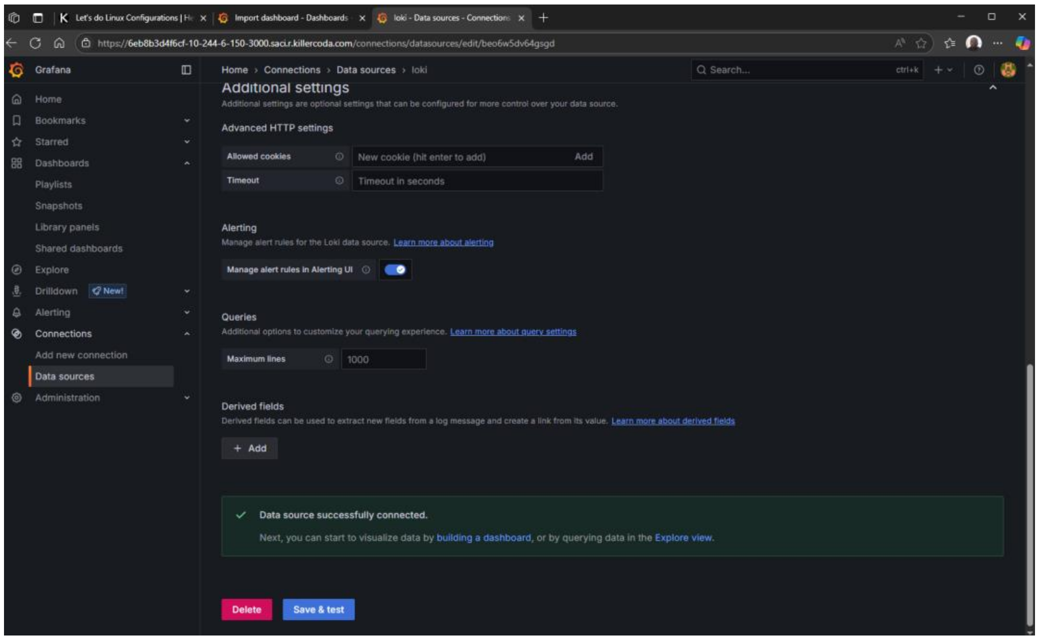
Solution

▶ Solution

Validation successful

```
After=network-online.target

[Service]
ExecStart=/opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/promtail-local-config.yaml

[Install]
WantedBy=default.targetcontrolplane:/opt/promtail$ systemctl daemon-reload
controlplane:/opt/promtail$ systemctl enable promtail.service --now
Created symlink /etc/systemd/system/default.target.wants/promtail.service → /etc/systemd/system/promtail.service.
controlplane:/opt/promtail$ systemctl status promtail.service --no-pager
● promtail.service - Promtail Service Startup
     Loaded: loaded (/etc/systemd/system/promtail.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-06-07 00:48:32 UTC; 6s ago
   Main PID: 25217 (promtail-linux-)
      Tasks: 8 (limit: 2614)
     Memory: 25.6M (peak: 26.1M)
        CPU: 179ms
     CGroup: /system.slice/promtail.service
             └─25217 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/promtail-local-config.yaml

Jun 07 00:48:32 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:32.273322859Z caller=…cc8a)"
Jun 07 00:48:32 controlplane promtail-linux-amd64[25217]: level=warn ts=2025-06-07T00:48:32.292203727Z caller=…onfig"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.293305373Z caller=…th\"}"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.293389033Z caller=…og\"}"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.293472312Z caller=…ar/log
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.29359409Z caller=f…ar/log
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: ts=2025-06-07T00:48:37.293722276Z caller=log.go:168 …ce:0}"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.294221853Z caller=…syslog
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: ts=2025-06-07T00:48:37.294439619Z caller=log.go:168 …ce:0}"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.294783847Z caller=…th.log
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:/opt/promtail$ ps -ef | grep [p]romtail
root       25217       1  2 00:48 ?        00:00:00 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/pro
mtail-local-config.yaml
controlplane:/opt/promtail$
```

Let's do Linux Configurations | He
Home - Dashboards - Grafana
https://killercoda.com/het-tanis/course/Linux-Labs/108-kafka-to-loki-logging
KILLERCODA
PLUS
Areas
Account
Editor
Tab 1
BACK
CHECK



Additional settings

Additional settings are optional settings that can be configured for more control over your data source.

Advanced HTTP settings

Allowed cookies	New cookie (hit enter to add)
Timeout	Timeout in seconds

Alerting

Manage alert rules for the Loki data source. Learn more about alerting

Manage alert rules in Alerting UI: on

Queries

Additional options to customize your querying experience. Learn more about query settings

Maximum lines: 1000

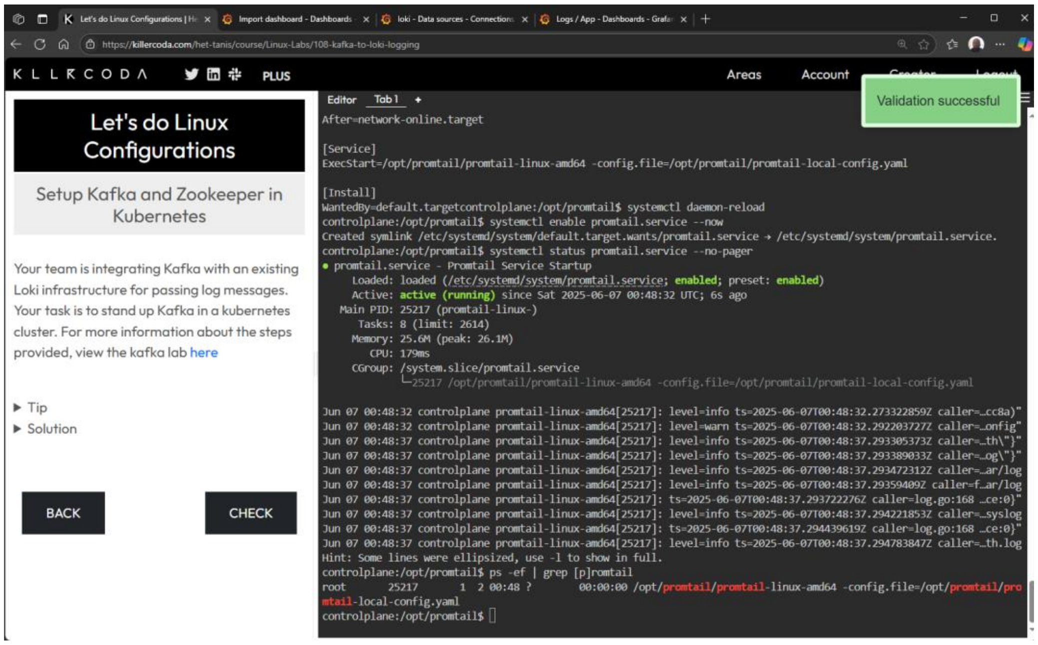
Derived fields

Derived fields can be used to extract new fields from a log message and create a link from its value. Learn more about derived fields

✓ Data source successfully connected.

Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.

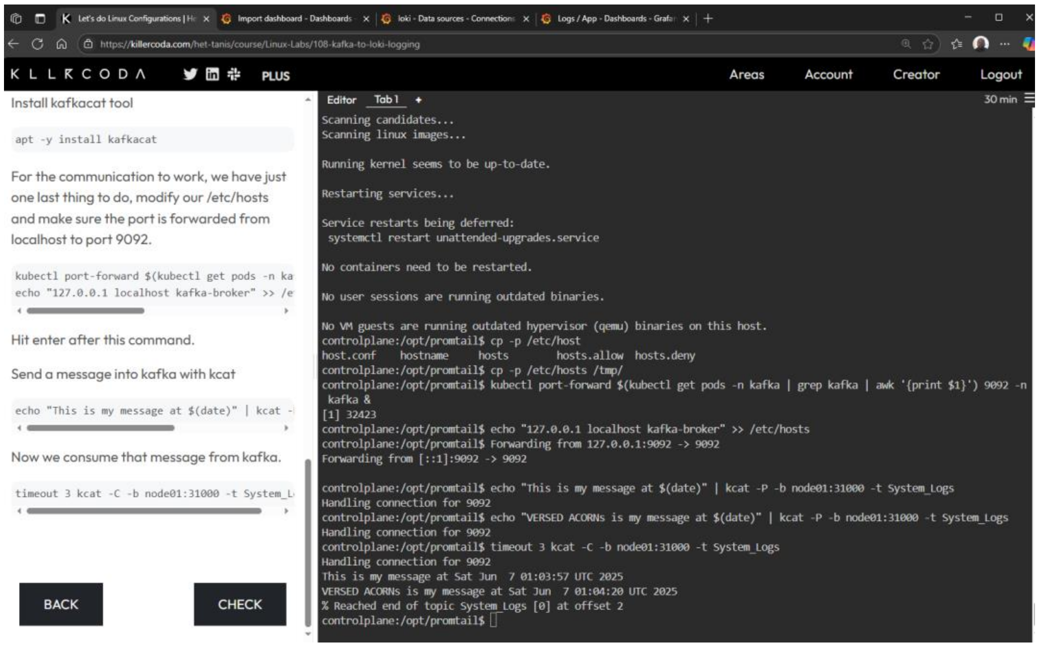
Let's do Linux Configurations | He
Import dashboard - Dashboards
loki - Data sources - Connections
https://6eb8b3d4f6cf-10-244-6-150-3000.saci.r.killercoda.com/connections/datasources/edit/beo6w5dv64gsgd
Grafana
Home › Connections › Data sources › loki
Search...
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
New!
Alerting
Connections
Add new connection
Data sources
Administration
Add
+ Add
Delete
Save & test



Let's do Linux Configurations
Setup Kafka and Zookeeper in Kubernetes
Tip
Solution
BACK
CHECK
Validation successful

Your team is integrating Kafka with an existing Loki infrastructure for passing log messages. Your task is to stand up Kafka in a kubernetes cluster. For more information about the steps provided, view the kafka lab here

```
After=network-online.target

[Service]
ExecStart=/opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/promtail-local-config.yaml

[Install]
WantedBy=default.targetcontrolplane:/opt/promtail$ systemctl daemon-reload
controlplane:/opt/promtail$ systemctl enable promtail.service --now
Created symlink /etc/systemd/system/default.target.wants/promtail.service → /etc/systemd/system/promtail.service.
controlplane:/opt/promtail$ systemctl status promtail.service --no-pager
● promtail.service - Promtail Service Startup
     Loaded: loaded (/etc/systemd/system/promtail.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-06-07 00:48:32 UTC; 6s ago
   Main PID: 25217 (promtail-linux-)
      Tasks: 8 (limit: 2614)
     Memory: 25.6M (peak: 26.1M)
        CPU: 179ms
     CGroup: /system.slice/promtail.service
             └─25217 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/promtail-local-config.yaml

Jun 07 00:48:32 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:32.273322859Z caller=…cc8a)"
Jun 07 00:48:32 controlplane promtail-linux-amd64[25217]: level=warn ts=2025-06-07T00:48:32.292203727Z caller=…onfig"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.293305373Z caller=…th\"}"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.293389033Z caller=…og\"}"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.293472312Z caller=…ar/log
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.29359409Z caller=f…ar/log
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: ts=2025-06-07T00:48:37.293722276Z caller=log.go:168 …ce:0}"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.294221853Z caller=…syslog
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: ts=2025-06-07T00:48:37.294439619Z caller=log.go:168 …ce:0}"
Jun 07 00:48:37 controlplane promtail-linux-amd64[25217]: level=info ts=2025-06-07T00:48:37.294783847Z caller=…th.log
Hint: Some lines were ellipsized, use -l to show in full.
controlplane:/opt/promtail$ ps -ef | grep [p]romtail
root       25217       1  2 00:48 ?        00:00:00 /opt/promtail/promtail-linux-amd64 -config.file=/opt/promtail/pro
mtail-local-config.yaml
controlplane:/opt/promtail$ 
```



Install kafkacat tool

```
apt -y install kafkacat
```

For the communication to work, we have just one last thing to do, modify our /etc/hosts and make sure the port is forwarded from localhost to port 9092.

```
kubectl port-forward $(kubectl get pods -n ka
echo "127.0.0.1 localhost kafka-broker" >> /e
```

Hit enter after this command.

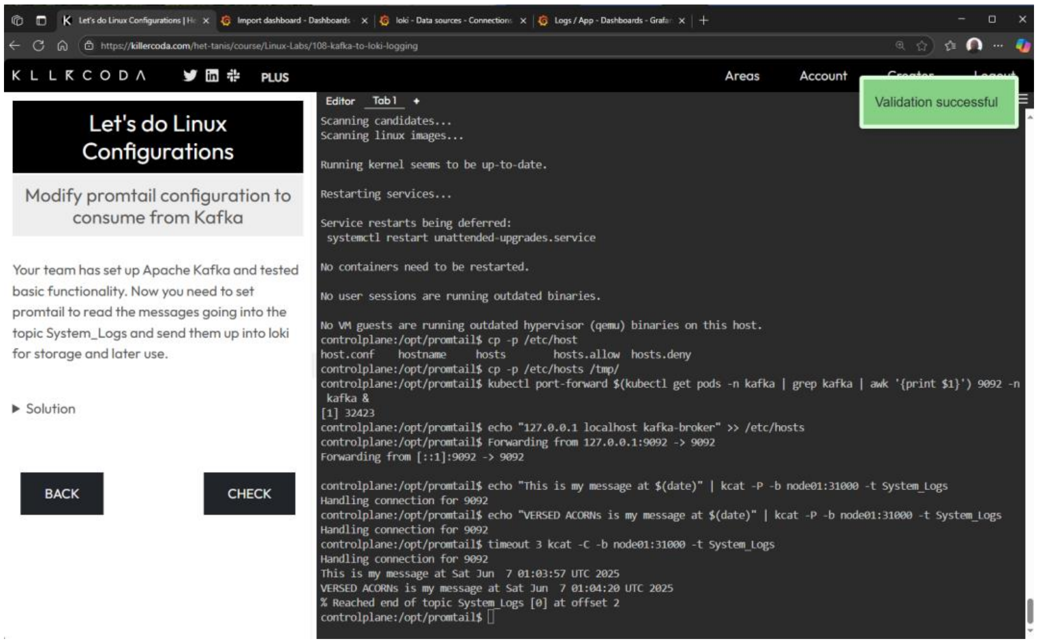
Send a message into kafka with kcat

```
echo "This is my message at $(date)" | kcat -
```

Now we consume that message from kafka.

```
timeout 3 kcat -C -b node01:31000 -t System_L
```

```
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
controlplane:/opt/promtail$ cp -p /etc/host
host.conf    hostname     hosts        hosts.allow  hosts.deny
controlplane:/opt/promtail$ cp -p /etc/hosts /tmp/
controlplane:/opt/promtail$ kubectl port-forward $(kubectl get pods -n kafka | grep kafka | awk '{print $1}') 9092 -n
 kafka &
[1] 32423
controlplane:/opt/promtail$ echo "127.0.0.1 localhost kafka-broker" >> /etc/hosts
controlplane:/opt/promtail$ Forwarding from 127.0.0.1:9092 -> 9092
Forwarding from [::1]:9092 -> 9092

controlplane:/opt/promtail$ echo "This is my message at $(date)" | kcat -P -b node01:31000 -t System_Logs
Handling connection for 9092
controlplane:/opt/promtail$ echo "VERSED ACORNs is my message at $(date)" | kcat -P -b node01:31000 -t System_Logs
Handling connection for 9092
controlplane:/opt/promtail$ timeout 3 kcat -C -b node01:31000 -t System_Logs
Handling connection for 9092
This is my message at Sat Jun  7 01:03:57 UTC 2025
VERSED ACORNs is my message at Sat Jun  7 01:04:20 UTC 2025
% Reached end of topic System_Logs [0] at offset 2
controlplane:/opt/promtail$ 
```



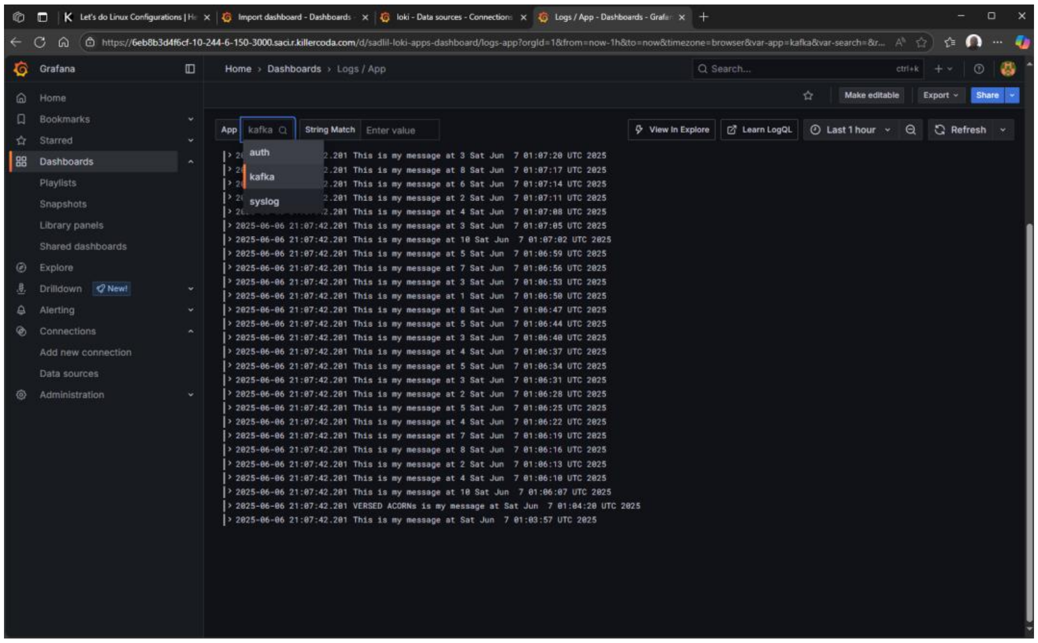
Let's do Linux Configurations

Modify promtail configuration to consume from Kafka

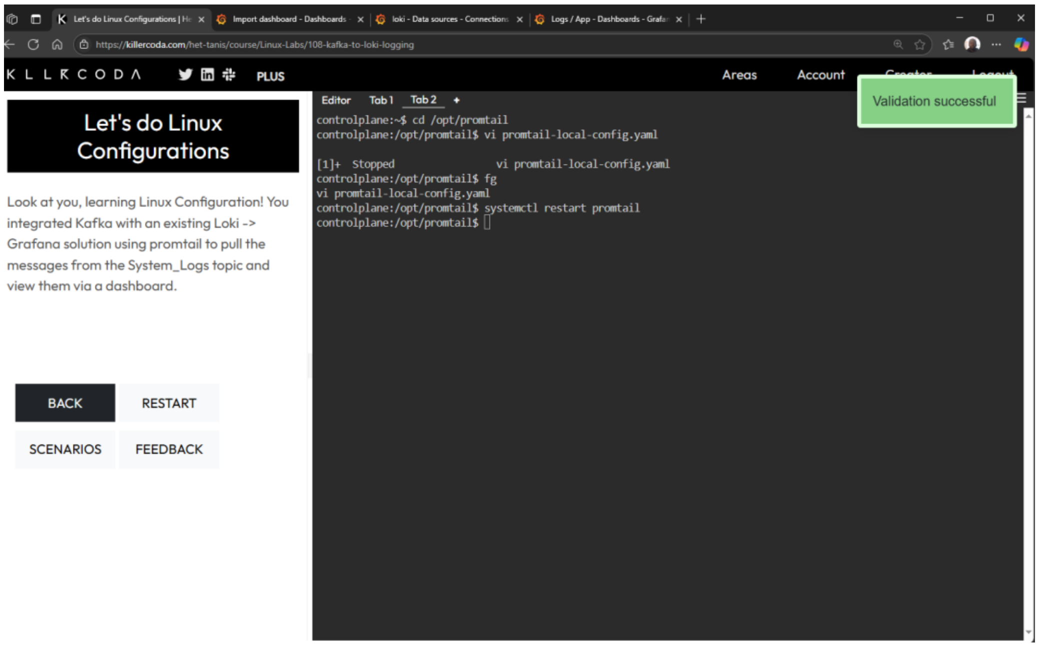
Your team has set up Apache Kafka and tested basic functionality. Now you need to set promtail to read the messages going into the topic System_Logs and send them up into loki for storage and later use.

▶ Solution

```
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
controlplane:/opt/promtail$ cp -p /etc/host
host.conf    hostname     hosts        hosts.allow  hosts.deny
controlplane:/opt/promtail$ cp -p /etc/hosts /tmp/
controlplane:/opt/promtail$ kubectl port-forward $(kubectl get pods -n kafka | grep kafka | awk '{print $1}') 9092 -n
 kafka &
[1] 32423
controlplane:/opt/promtail$ echo "127.0.0.1 localhost kafka-broker" >> /etc/hosts
controlplane:/opt/promtail$ Forwarding from 127.0.0.1:9092 -> 9092
Forwarding from [::1]:9092 -> 9092

controlplane:/opt/promtail$ echo "This is my message at $(date)" | kcat -P -b node01:31000 -t System_Logs
Handling connection for 9092
controlplane:/opt/promtail$ echo "VERSED ACORNs is my message at $(date)" | kcat -P -b node01:31000 -t System_Logs
Handling connection for 9092
controlplane:/opt/promtail$ timeout 3 kcat -C -b node01:31000 -t System_Logs
Handling connection for 9092
This is my message at Sat Jun  7 01:03:57 UTC 2025
VERSED ACORNs is my message at Sat Jun  7 01:04:20 UTC 2025
% Reached end of topic System_Logs [0] at offset 2
controlplane:/opt/promtail$
```



```
> 2(          2.201 This is my message at 3 Sat Jun  7 01:07:20 UTC 2025
> 2(          2.201 This is my message at 8 Sat Jun  7 01:07:17 UTC 2025
> 2(          2.201 This is my message at 6 Sat Jun  7 01:07:14 UTC 2025
> 2(          2.201 This is my message at 2 Sat Jun  7 01:07:11 UTC 2025
> 2(          2.201 This is my message at 4 Sat Jun  7 01:07:08 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 3 Sat Jun  7 01:07:05 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 10 Sat Jun  7 01:07:02 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 5 Sat Jun  7 01:06:59 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 7 Sat Jun  7 01:06:56 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 3 Sat Jun  7 01:06:53 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 1 Sat Jun  7 01:06:50 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 8 Sat Jun  7 01:06:47 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 5 Sat Jun  7 01:06:44 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 3 Sat Jun  7 01:06:40 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 4 Sat Jun  7 01:06:37 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 5 Sat Jun  7 01:06:34 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 3 Sat Jun  7 01:06:31 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 2 Sat Jun  7 01:06:28 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 5 Sat Jun  7 01:06:25 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 4 Sat Jun  7 01:06:22 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 7 Sat Jun  7 01:06:19 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 8 Sat Jun  7 01:06:16 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 2 Sat Jun  7 01:06:13 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 4 Sat Jun  7 01:06:10 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at 10 Sat Jun  7 01:06:07 UTC 2025
> 2025-06-06 21:07:42.201 VERSED ACORNs is my message at Sat Jun  7 01:04:20 UTC 2025
> 2025-06-06 21:07:42.201 This is my message at Sat Jun  7 01:03:57 UTC 2025
```


Let's do Linux Configurations | He
Import dashboard - Dashboards
loki - Data sources - Connections
Logs / App - Dashboards - Grafan
https://killercoda.com/het-tanis/course/Linux-Labs/108-kafka-to-loki-logging
KILLERCODA
PLUS
Areas
Account
Validation successful
Editor
Tab 1
Tab 2
BACK
RESTART
SCENARIOS
FEEDBACK

Let's do Linux Configurations

Look at you, learning Linux Configuration! You integrated Kafka with an existing Loki -> Grafana solution using promtail to pull the messages from the System_Logs topic and view them via a dashboard.

```
controlplane:~$ cd /opt/promtail
controlplane:/opt/promtail$ vi promtail-local-config.yaml

[1]+  Stopped                 vi promtail-local-config.yaml
controlplane:/opt/promtail$ fg
vi promtail-local-config.yaml
controlplane:/opt/promtail$ systemctl restart promtail
controlplane:/opt/promtail$ 
```

Does the flow make sense in the context of this diagram?

Yes

Can you find any configurations or blogs that describe why you might want to use this architecture or how it has been used in the industry?

tbd