

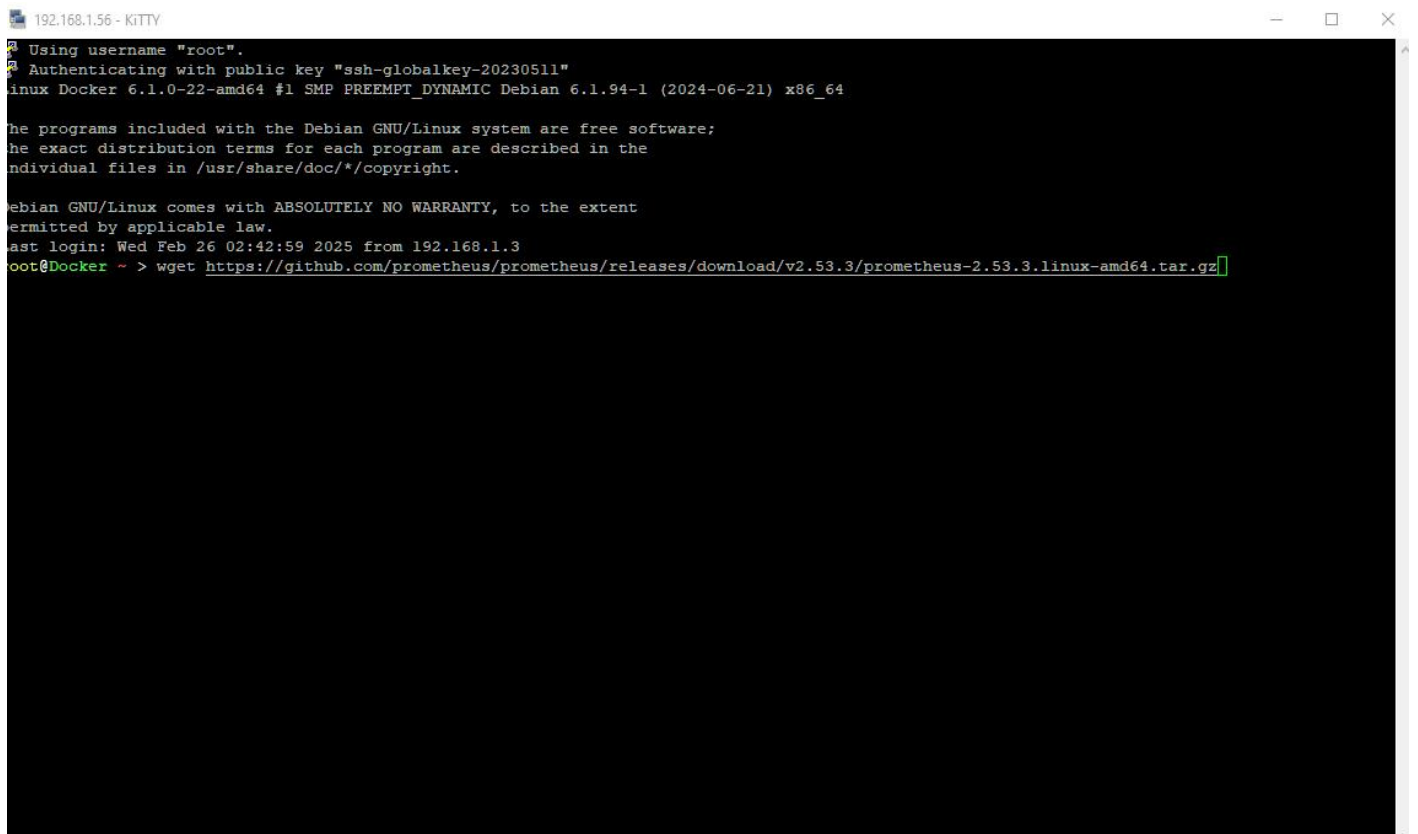
Prometheus installieren, Mailcow-Exporter installieren und die "Mailcow" in Grafana visuell darstellen

Also meine Grundlage ist eine VM auf Proxmox mit Docker und Docker-Compose. Wobei Letzteres nicht benötigt wird.

Ebenfalls empfehle ich immer ein Snapshot / Backup anzulegen! :)

Auf der Docker-VM habe ich erstmal Prometheus installiert:

```
wget https://github.com/prometheus/prometheus/releases/download/v2.53.3/prometheus-2.53.3.linux-amd64.tar.gz
```

A terminal window titled "192.168.1.56 - KITTY" showing an SSH session. The user logs in as root on a Debian 6.1.94-1 system. The terminal output includes the SSH banner, the Debian GNU/Linux welcome message, and the user's last login information. The user then runs the command to download Prometheus using wget.

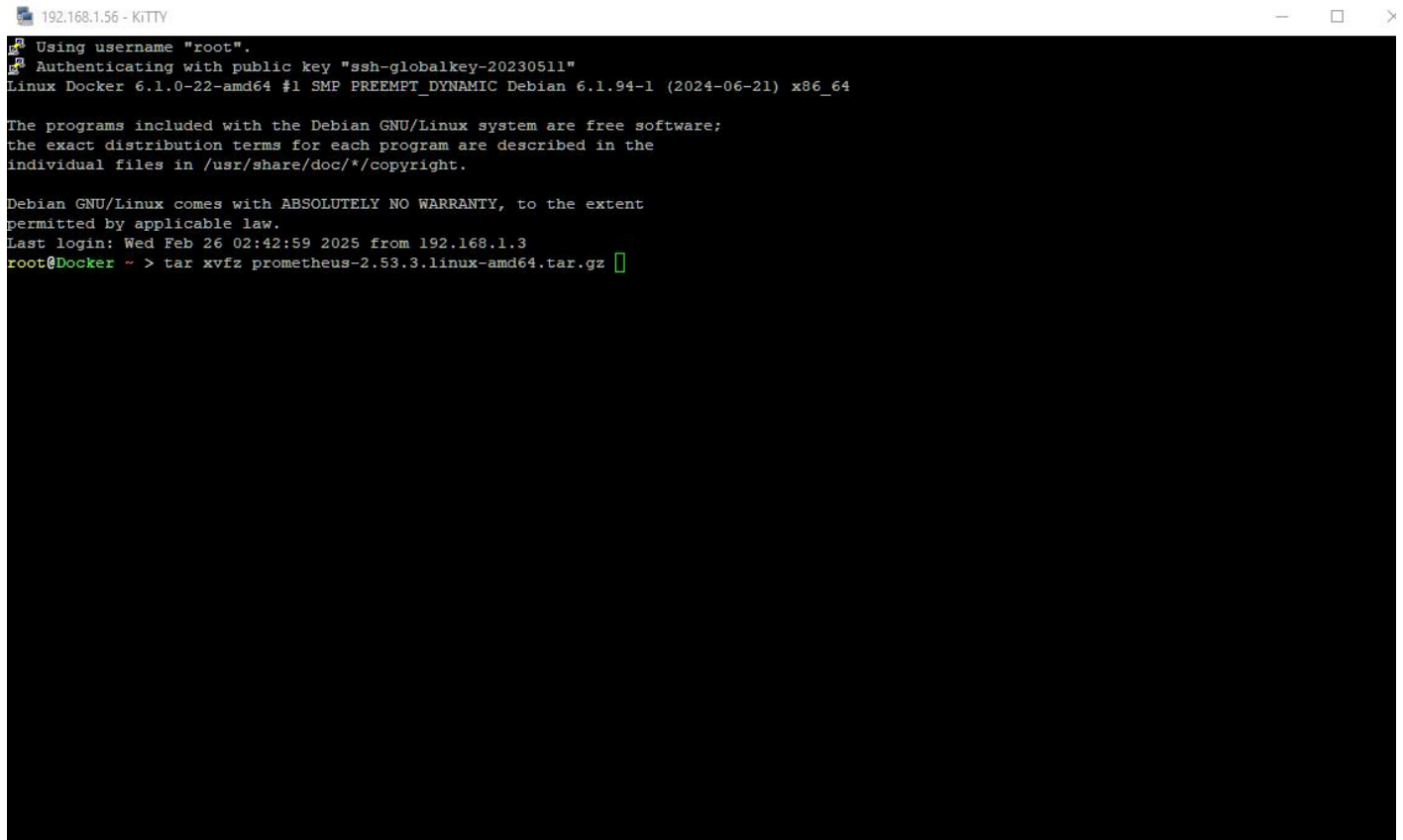
```
192.168.1.56 - KITTY
Using username "root".
Authenticating with public key "ssh-globalkey-20230511"
linux Docker 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
last login: Wed Feb 26 02:42:59 2025 from 192.168.1.3
root@Docker: ~ > wget https://github.com/prometheus/prometheus/releases/download/v2.53.3/prometheus-2.53.3.linux-amd64.tar.gz
```

Datei entpacken:

```
tar xvfz prometheus-2.53.3.linux-amd64.tar.gz
```

A terminal window titled '192.168.1.56 - KITTY' with standard window controls. The terminal output shows an SSH login process for the 'root' user. It displays the authentication method (public key), the system information (Linux Docker 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64), and the Debian GNU/Linux warranty notice. The last login is recorded as 'Wed Feb 26 02:42:59 2025 from 192.168.1.3'. The prompt is 'root@Docker ~'. The user has entered the command 'tar xvfz prometheus-2.53.3.linux-amd64.tar.gz' and the cursor is at the end of the line.

```
192.168.1.56 - KITTY
Using username "root".
Authenticating with public key "ssh-globalkey-20230511"
Linux Docker 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb 26 02:42:59 2025 from 192.168.1.3
root@Docker ~ > tar xvfz prometheus-2.53.3.linux-amd64.tar.gz
```

Dann ins Prometheus-Verzeichnis gewechselt:

```
cd prometheus-2.53.3.linux-amd64/
```

```
Using username "root".
Authenticating with public key "ssh-globalkey-20230511"
Linux Docker 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb 26 02:42:59 2025 from 192.168.1.3
root@Docker ~ > cd prometheus-2.53.3.linux-amd64/
```

Dort dann die "prometheus.yml" angepasst an die URL inkl. API-Key der Mailcow (s. 2. Screenshot):

nano prometheus.yml

```
Using username "root".
Authenticating with public key "ssh-globalkey-20230511"
Linux Docker 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb 26 02:42:59 2025 from 192.168.1.3
root@Docker ~ > nano prometheus.yml
```

```
192.168.1.56 - KTTY
GNU nano 7.2 prometheus.yml
my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "mailcow"
    static_configs:
      - targets: [ "mail.                .de:9099" ]
    params:
      host: [ "mail.                .de" ]
      apiKey: [ "                ~D513EA" ]

[ 36 Zeilen gelesen ]
^G Hilfe      ^C Speichern  ^W Wo ist     ^K Ausschneiden ^T Ausführen   ^C Position  ^M-U Rückgängig ^M-A Markierung ^M-] Zu Klammer
^X Beenden    ^S Datei öffnen ^\ Ersetzen   ^U Einfügen    ^J Ausrichten  ^/ Zu Zeile   ^M-E Wiederholen ^M-6 Kopieren   ^C Wo war
```

Den API-Key kann man im Mailcow Config-Menü ganz einfach generieren.

Lese-Schreib-Zugriff

IP-Adressen oder Netzwerke (CIDR Notation) für Zugriff auf API:

1 [redacted]

☒ IP-Check für API nicht ausführen

API-Key: [redacted]D513EA

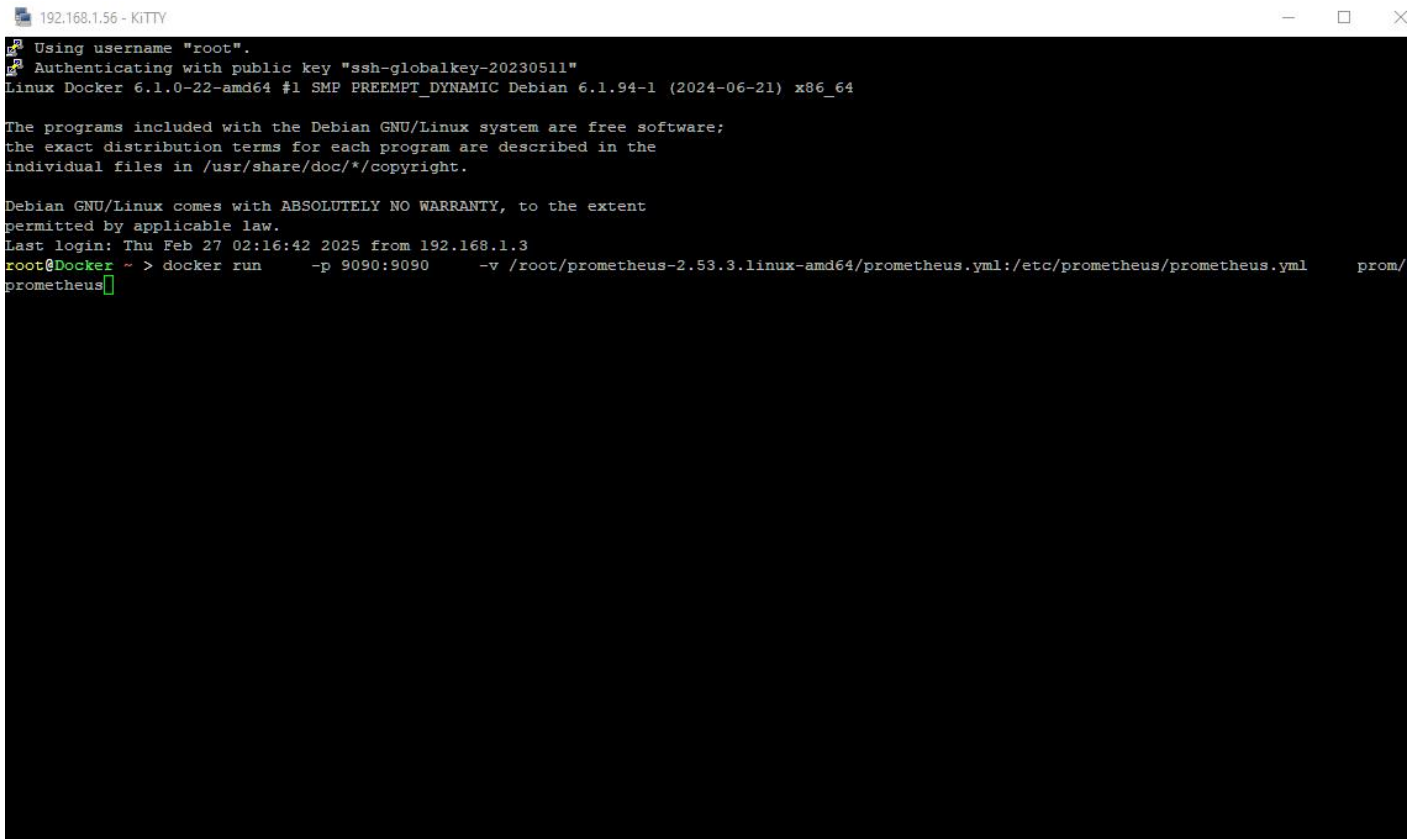
☒ API aktivieren

✓ Änderungen speichern

API-Key regenerieren

Ich habe dann einmal den Container manuell mit der Config-Datei gestartet:

```
docker run -p 9090:9090 -v /root/prometheus-2.53.3.linux-  
amd64/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus
```



A terminal window titled "192.168.1.56 - KITTY" showing an SSH session. The user logs in as root on a Linux Docker 6.1.0-22-amd64 system. The terminal displays the Debian GNU/Linux system information and the Docker command being executed: `docker run -p 9090:9090 -v /root/prometheus-2.53.3.linux-amd64/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus`. The prompt is `root@Docker ~`.

Um den Prometheus-Container immer automatisch zu starten, habe ich per Portainer den Prometheus-Container bei "Restart" auf "Always" gesetzt.
Alternativ geht das aber auch mit dem Befehl

```
docker run --restart always <image_name>
```

In meinem Fall heißt der Container "prom/prometheus:latest".

```
192.168.1.56 - KITTYY
root@Docker: ~/prometheus-2.53.3.linux-amd64 > docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  NAMES               CREATED        STATUS        PORTS
6745ddd3de68   prom/prometheus:latest             "/bin/prometheus --c..." 23 hours ago      Up 23 hours    0.0.0.0:9090->9090/tcp, :::9
090->9090/tcp   prometheus
b36dlf7f4cf1   grafana/grafana:latest             "/run.sh"                grafana             23 hours ago   Up 23 hours   0.0.0.0:3000->3000/tcp, :::3
000->3000/tcp   grafana
008e89a86fal   portainer/portainer-ee:latest       "/portainer"             portainer           28 hours ago   Up 28 hours   0.0.0.0:8000->8000/tcp, :::8
000->8000/tcp, 0.0.0.0:9443->9443/tcp, :::9443->9443/tcp, 9000/tcp portainer
680a01f1115d   checkmk/check-mk-rw:2.2.0-latest    "/docker-entrypoint..." 20 months ago     Up 5 days (healthy) 6557/tcp, 0.0.0.0:8080->5000
/tcp, :::8080->5000/tcp, 0.0.0.0:8001->8000/tcp, :::8001->8000/tcp monitoring
3205db621ca8   linuxserver/mariadb                "/init"                  bookstack_db        23 months ago   Up 5 days     3306/tcp

root@Docker: ~/prometheus-2.53.3.linux-amd64 > []
```

Am Besten ist es aber, über den Portainer diese Einstellung vorzunehmen, da per CLI ein neuer Container angelegt wird, warum auch immer.

Bezüglich Prometheus sind wir auf der Docker-VM fertig.

Auf der Mailcow habe ich ebenfalls Portatiner laufen. Hier müssen wir nicht so viel machen. Lediglich den Container "thej6s/mailcow-exporter" laden.

```
docker run -p '9099:9099' thej6s/mailcow-exporter
```

Auch hier habe ich per Portainer den "Restart" auf "Always" gestellt, damit dieser immer automatisch geladen wird.

```
⚠ Using username "root".
⚠ Authenticating with public key "Hetzner-Mailcow"
Linux mail. de 6.1.0-23-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.99-1 (2024-07-15) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb 26 02:22:42 2025 from
root@mail ~ > docker run -p '9099:9099' thej6s/mailcow-exporter
```

Das war's im Grunde schon.

Sobald auf der Mailcow der "thej6s/mailcow-exporter" läuft, werden die Daten schon zum Prometheus geschoben und wir können unser Dashboard im Grafana "basteln" bzw. eher gesagt "klauen". :)

Im Grafana dann noch die Datenquelle "Prometheus" hinzufügen:

Home > Connections > Data sources > Prometheus

Q Search or jump to...

Type: Prometheus

AlertingSupportedExplore dataBuild a dashboard

Type: Prometheus

SettingsDashboards

NamePrometheusDefault

Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).

Fields marked with * are required

Connection

Prometheus server URLhttp://192.168.1.56:9090

Authentication

Authentication methods

Choose an authentication method to access the data source

Authentication methodNo Authentication

TLS settings

Additional security measures that can be applied on top of authentication

Add self-signed certificateTLS Client authenticationSkip TLS certificate validation

HTTP headers

Pass along additional context and metadata about the request/response

Advanced settings

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

Advanced HTTP settings

Allowed cookiesNew cookie (hit enter to add)Add

TimeoutTimeout in seconds

Alerting

Manage alerts via Alerting UI

Interval behaviour

Scrape interval15s

Query timeout60s

Query editor

Default editorBuilder

Disable metrics lookup

Performance

Prometheus typePrometheus

Prometheus version2.19.x

Cache levelLow

Incremental querying (beta)

Disable recording rules (beta)

Other

Custom query parametersExample: max_source_resolution=5m&timeout

HTTP methodPOST

Use series endpoint

Exemplars

Internal link

URLhttps://example.com/\${L_value.raw}

URL LabelGo to example.com

Label nametraceID

Remove exemplar link

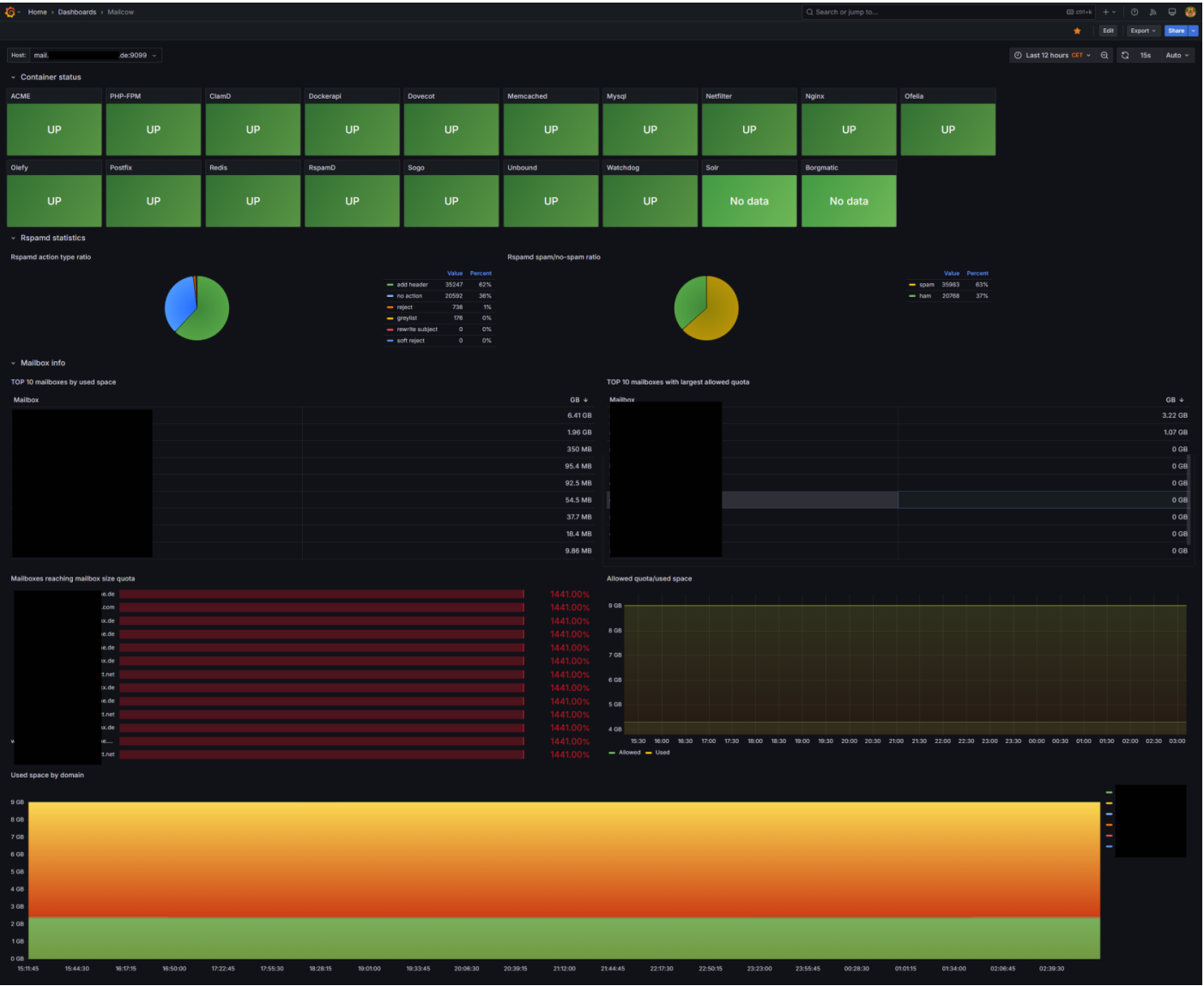
+ Add

DeleteSave & test

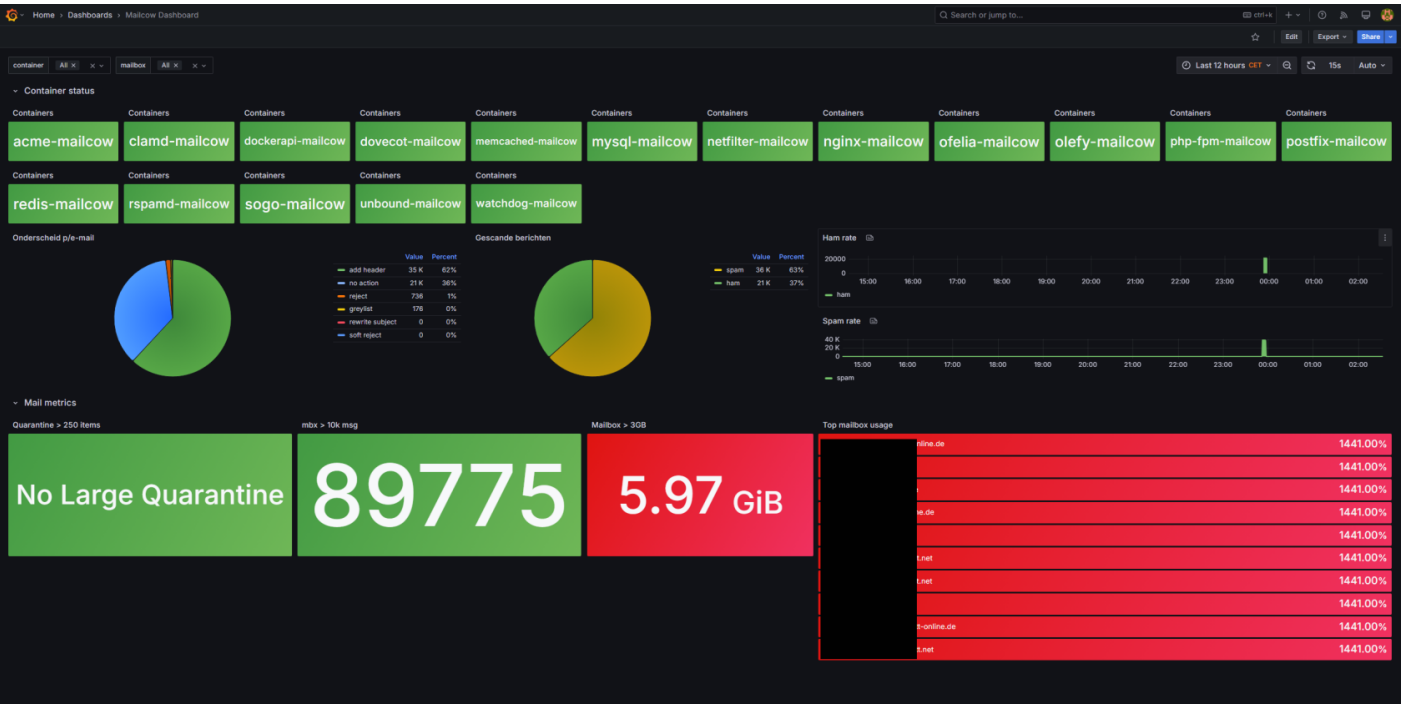
Danach können wir die Dashboards in Grafana importieren.

Ich habe diese beide Dashboards genommen (sind aber einfach nur importiert, da ist noch nichts angepasst. Daher die wilden Werte).:

<https://github.com/hagaram/mailcow-grafana-dashboard>



<https://github.com/remkolodder/mailcow-dashboard>



Fertig. :)

Revision #4

Created 27 February 2025 01:55:11 by Christian

Updated 5 December 2025 07:40:04 by Christian