PSA
SYSTEMÜBERWACHUNG

Mit Prometheus & Grafana.

Agenda

Aufgabe

Anforderungen

Prometheus

Prometheus
Setup
Exporter

Services

Exporter
Unsere Systeme

Grafana

Visualisierung

Alerts

AlertManager
Slack Integration
Unsere Alerts

01

Aufgabe

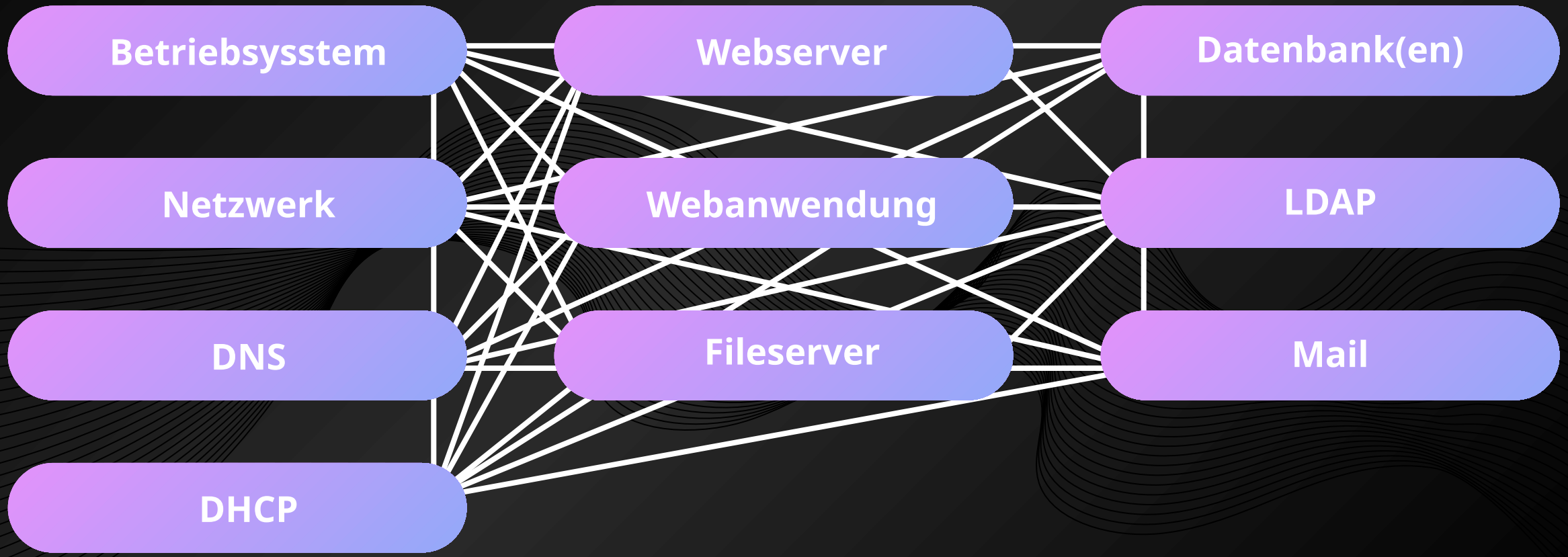
Systemüberwachung
Anforderungen

Läuft alles?

Komplexe System-Strukturen, verteilt auf viele Maschienen und Services:

Systemüberwachung, nur wie?

Was heißt “alles”?



Große Auswahl.

Nagios

1999
Vom Aufgabenblatt :)
Weit Verbreitet

Prometheus

2012
Industrie-genutzt
Großer Support
Einfache Einrichtung

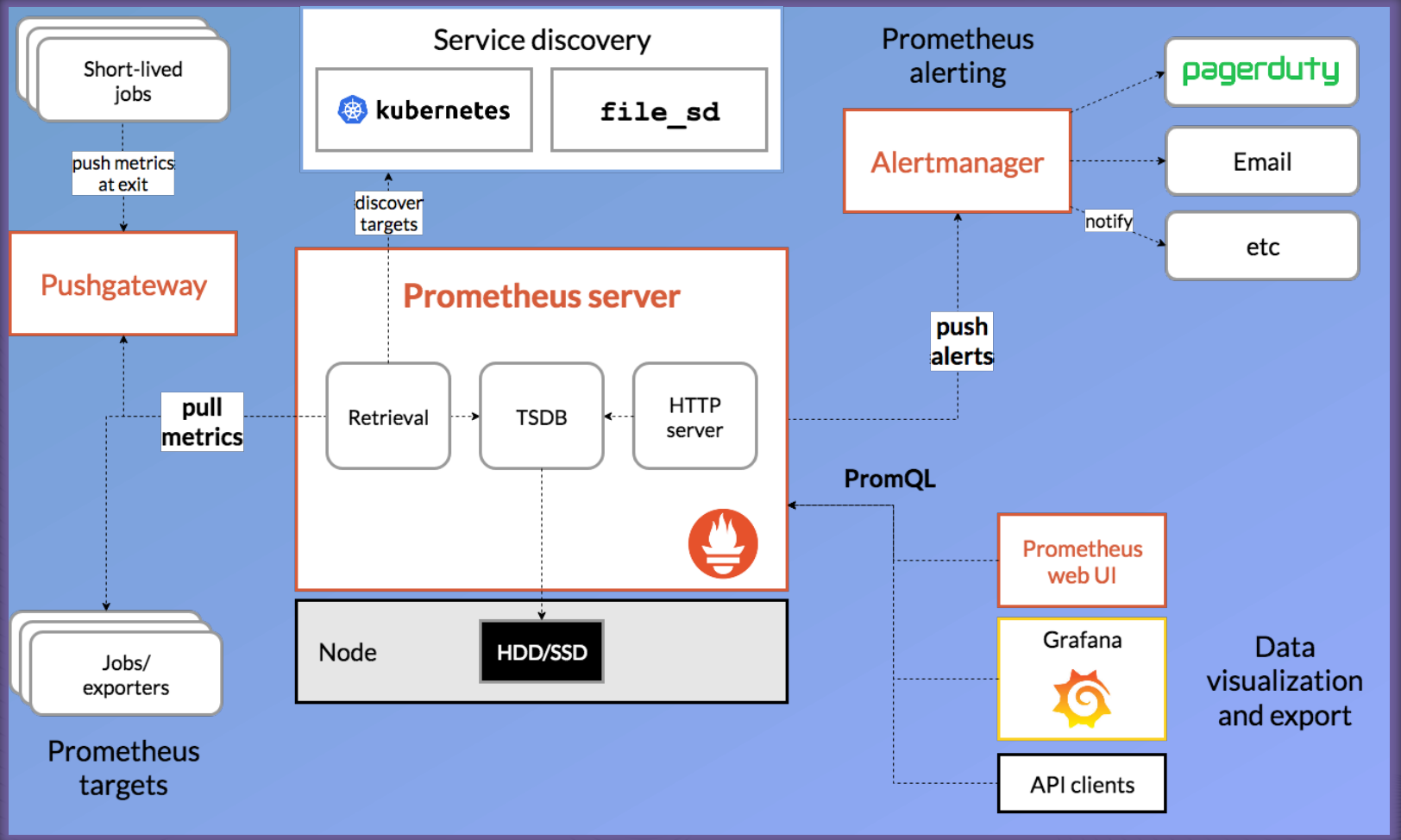
Viele mehr...

02

Prometheus

Architektur
Komponenten
Setup & Installation

Architektur.



Prometheus

Als zentraler Server

Jobs / Exporters

An den Systemen

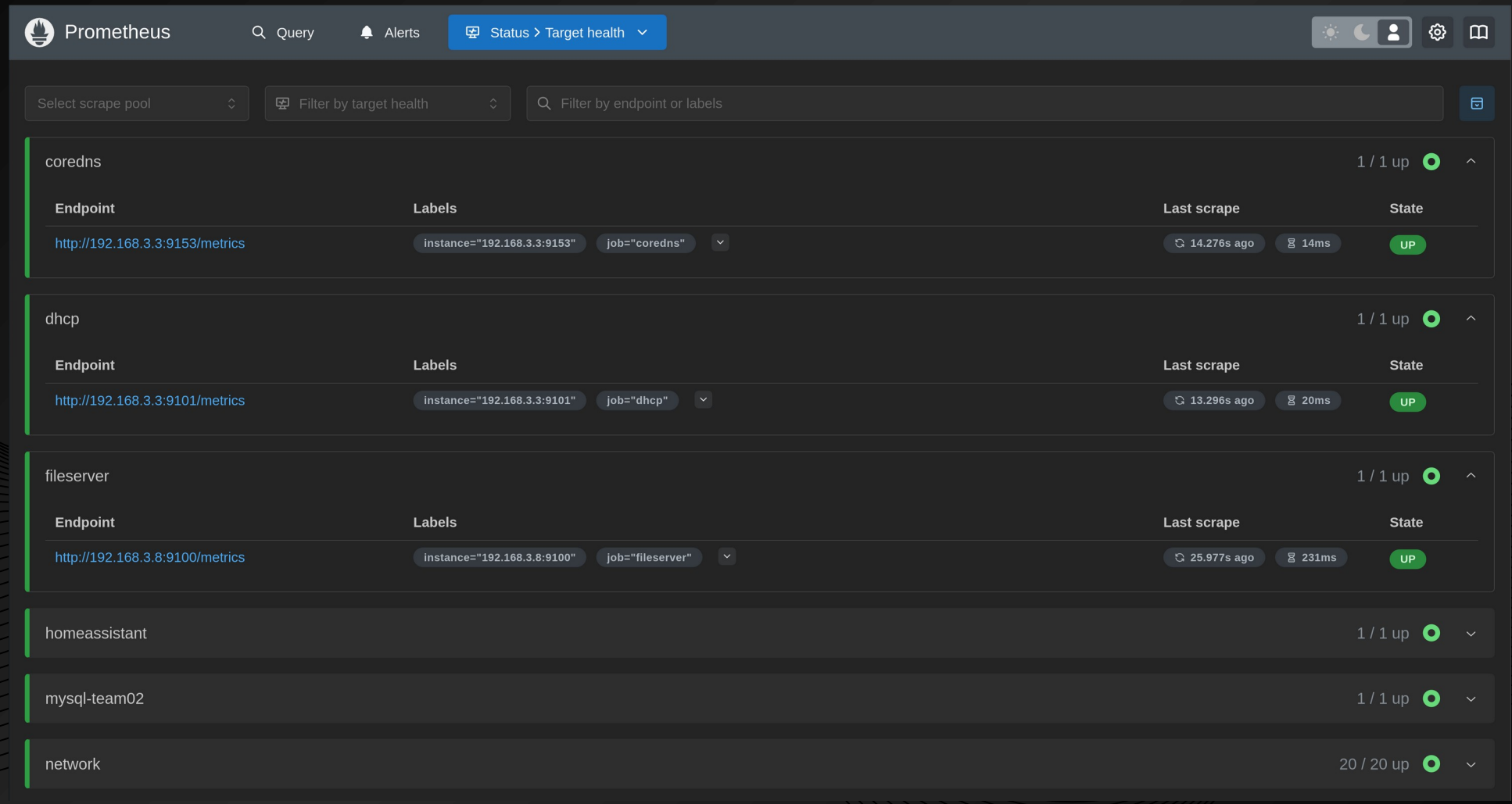
Grafana

Als visuelles Interface

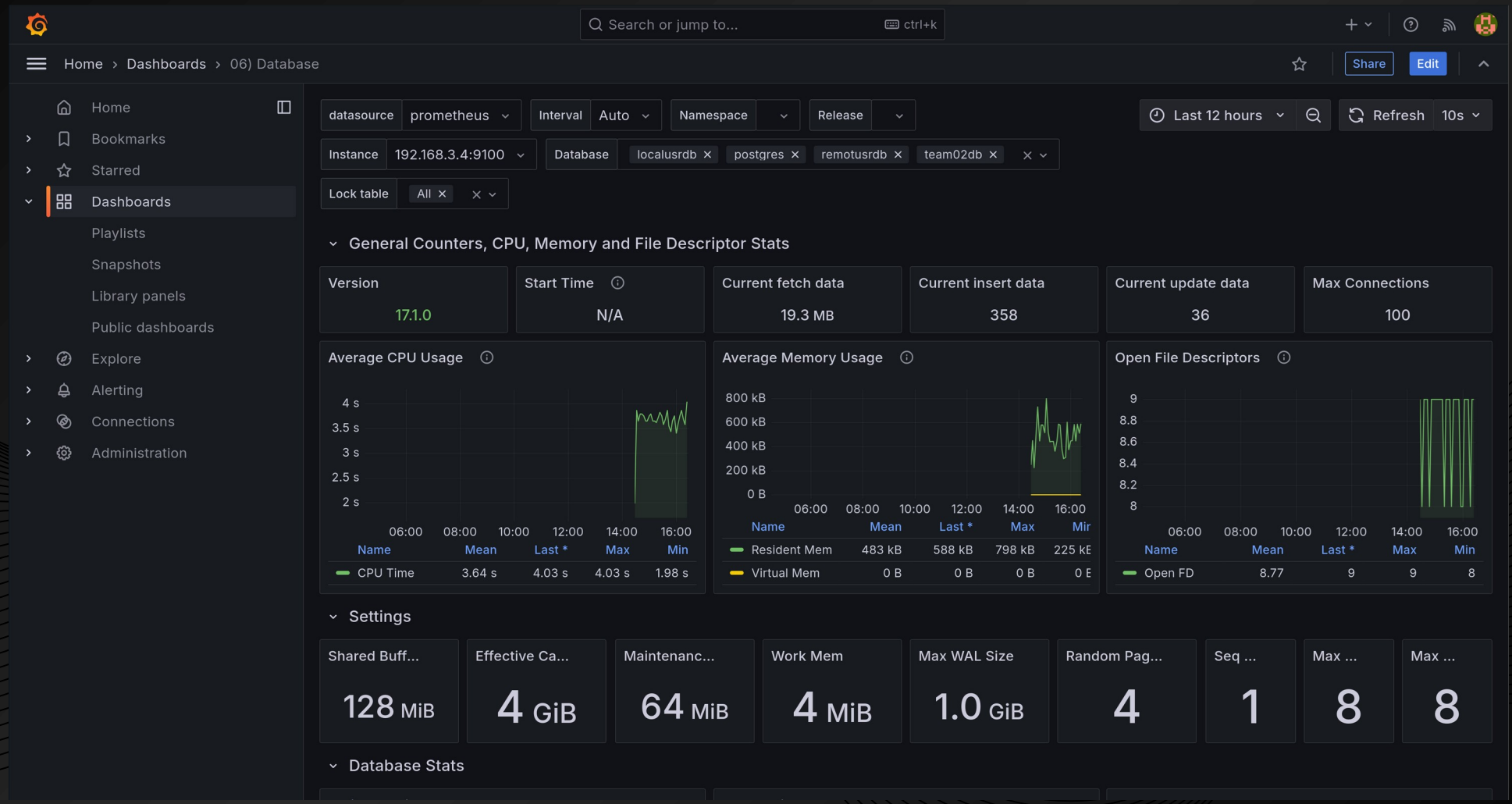
Alertmanager

Als Alert Manager :P

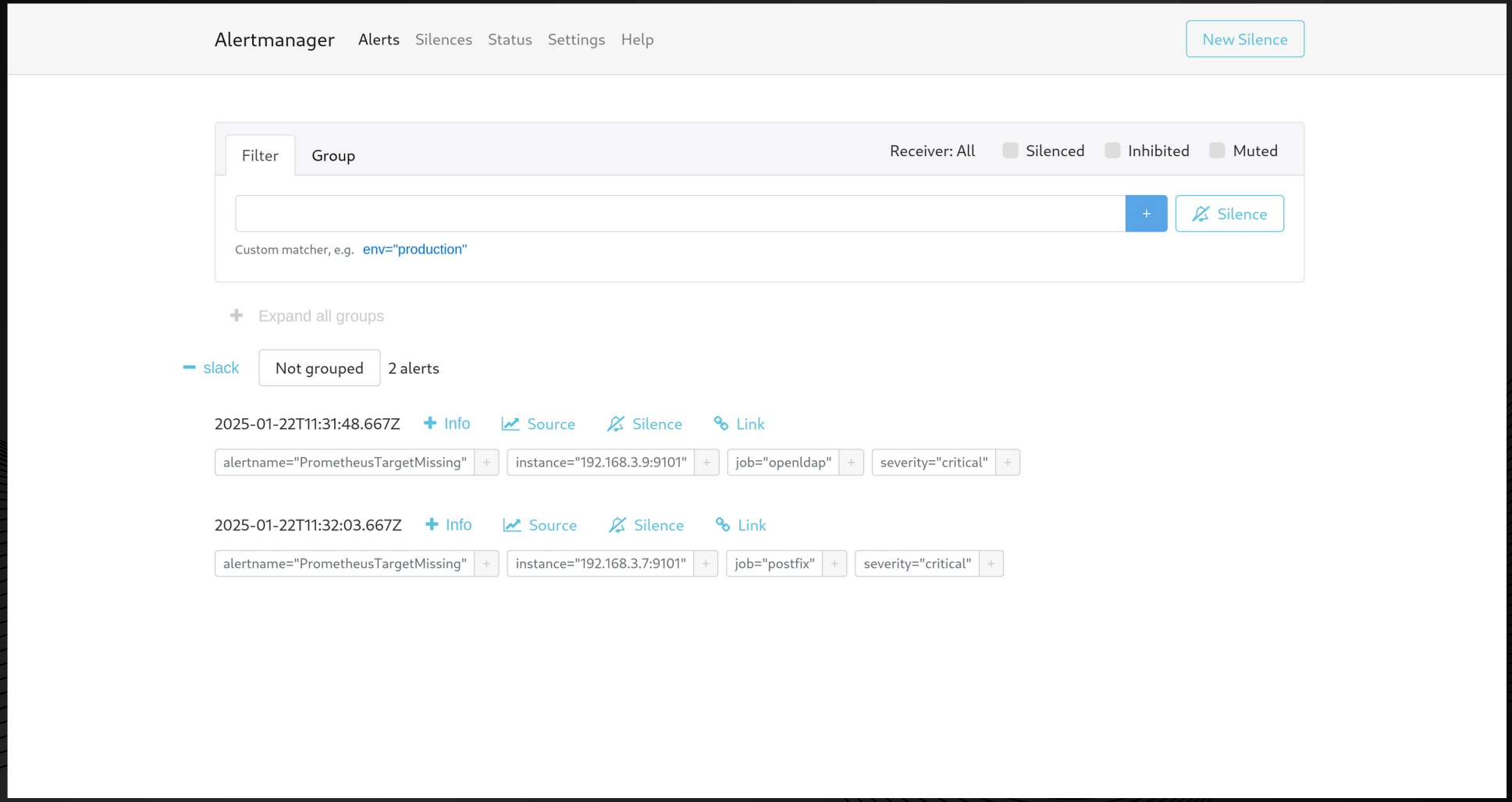
Prometheus.



Grafana.



Alertmanager.



Installation.

Docker

Firewall

Konfiguration

```
# monitoring-config.nix
{ config, lib, pkgs, ... }:
{
    virtualisation.docker.enable = true;
    users.extraGroups.docker.members = [ "root" ];
}
```

NixOS

Docker aktivieren, fertig.

Installation.

Docker | Firewall | Konfiguration

```
# Create the directory structure
mkdir -p /root/docker/alert-manager
mkdir -p /root/docker/grafana
mkdir -p /root/docker/prometheus
mkdir -p /root/docker/blackbox

# Create the config files
touch /root/docker/docker-compose.yaml
touch /root/docker/alert-manager/alertmanager.yml
touch /root/docker/grafana/grafana.ini
touch /root/docker/prometheus/alert-rules.yml
touch /root/docker/prometheus/prometheus.yml
touch /root/docker/blackbox/blackbox.yml
```

Neue Directories

Für jeden Service

Config-Dateien

In den neuen Pfaden

Installation.

Docker

```yaml
# docker-compose.yml
services:
  # Prometheus zur Metriken-Sammlung
  prometheus:
    image: prom/prometheus
    ports:
      - '9090:9090'
    volumes:
      - /root/docker/prometheus:/etc/prometheus
    command:
      - '--config.file=/etc/prometheus/prometheus.yml'
      - '--web.enable-lifecycle'
    restart: unless-stopped

  # Grafana zur Visualisierung
  grafana:
    image: grafana/grafana
    ports:
      - '3000:3000'
    depends_on:
      - prometheus
    restart: unless-stopped
    environment:
     - HTTP_PROXY=http://proxy.cit.tum.de:8080/
     - HTTPS_PROXY=http://proxy.cit.tum.de:8080/
     - NO_PROXY=localhost,127.0.0.1,prometheus
    volumes:
     - grafana-storage:/var/lib/grafana

  # Alertmanager zur Alarmierung bei Fehlern
  alertmanager:
    image: prom/alertmanager
    ports:
      - "9093:9093"
    volumes:
      - /root/docker/alert-manager:/config
    command: --config.file=/config/alertmanager.yml --log.level=debug

  # Blackbox Exporter zur Überwachung von Webseiten
  blackbox:
    image: prom/blackbox-exporter:latest
    ports:
     - 9115:9115
    volumes:
     - /root/docker/blackbox:/etc/blackbox
    command:
     - --config.file=/etc/blackbox/blackbox.yml
    depends_on:
      - prometheus
    restart: unless-stopped
    environment:
     - HTTP_PROXY=http://proxy.cit.tum.de:8080/
     - HTTPS_PROXY=http://proxy.cit.tum.de:8080/
     - NO_PROXY=localhost,127.0.0.1,blackbox

volumes:
  grafana-storage: {}
```

Prometheus

port 9090
config.file

Grafana

port 3000
abhängig von Prometheus
proxy Einstellungen

Alertmanager

port 9093
config.file

Blackbox

port 9115
config.file
abhängig von Prometheus

Installation.

Docker | Firewall | Konfiguration

```
docker compose up -d
```

Docker

Start nach Firewall & Konfiguration.

Installation.

Docker | Firewall | Konfiguration

```
iptables -A INPUT -p tcp --dport 9100 -j ACCEPT
iptables -A INPUT -p tcp --dport 9090 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 9100 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 9090 -j ACCEPT
...
```

Firewall

Erlauben aller nötigen Verbindungen

Installation.

Docker

Firewall

Konfiguration

```
# prometheus.yml
global:
  scrape_interval: 15s
  evaluation_interval: 15s
scrape_configs:
  # Prometheus selbst als Beispiel
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']
```

Prometheus

Update-Rate, etc.

Jobs

Alle angehängten Jobs, die beobachtet werden sollen.

03

Unsere Services

Exporter
Installation & Setup

Installation.

```nix
# os-expoerter.nix
{ config, lib, pkgs, ... }:
{
  services.prometheus.exporters.node = {
    enable = true;
    port = 9100;
    enabledCollectors = [
      "logind"
      "systemd"
    ];
    disabledCollectors = [
      "textfile"
    ];
    openFirewall = true;
  };
}
```

NixOS-Config

.nix

Default Exporter

enable
port

weitere Konfigurationen

Installation.

```
# prometheus.yml
...
scrape_configs:
  ...
```

NixOS-Config

.nix

```
  - job_name: 'os-status'
    static_configs:
      - targets:
          - '192.168.3.1:9100' #vm1
          - '192.168.3.2:9100' #vm2
          - '192.168.3.3:9100' #router
          # database, homeassistant up over own exporter
          - '192.168.3.6:9100' #webserver
          - '192.168.3.7:9100' #fileserver
```

Job

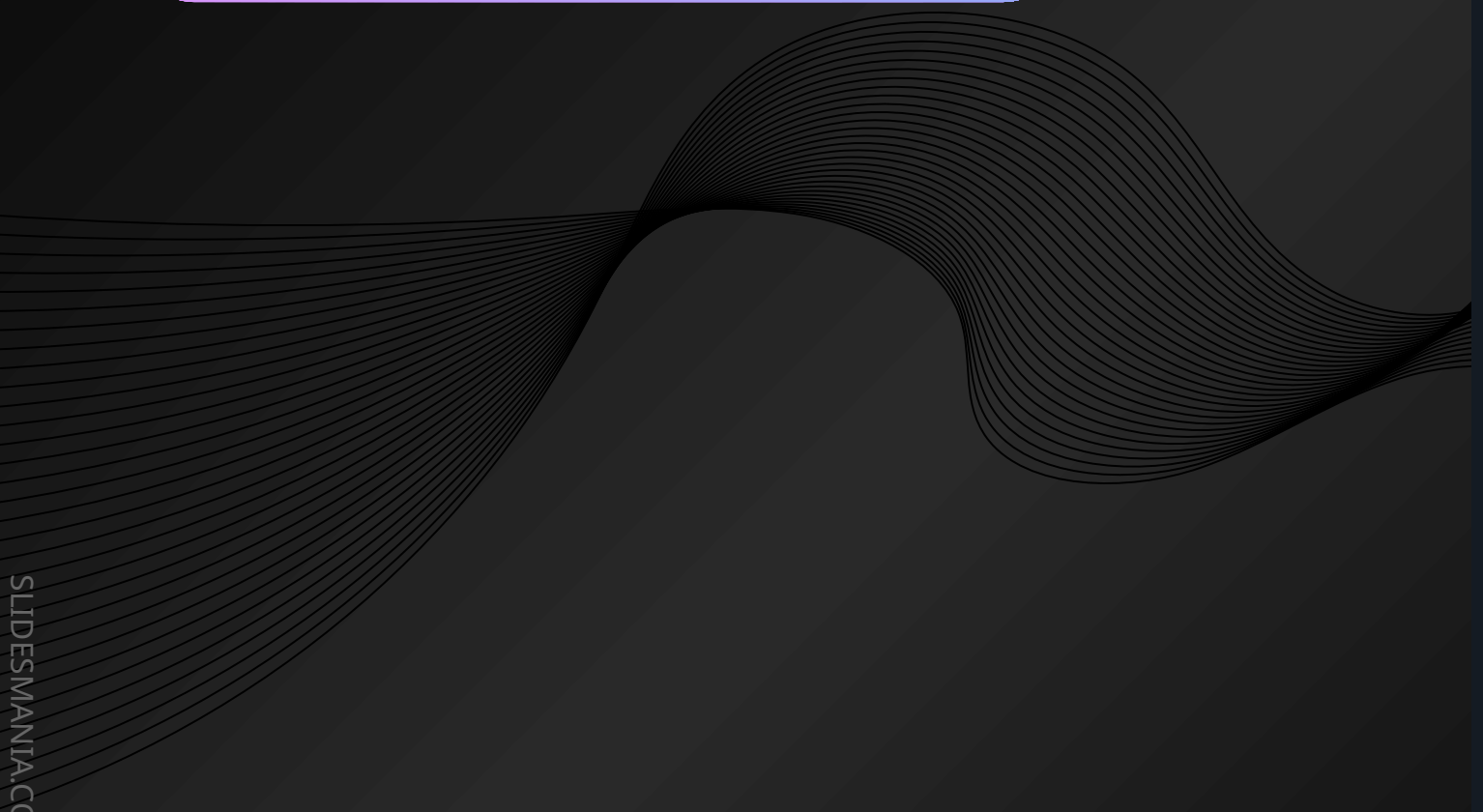
name
targets

Zum Überwachen.

- Betriebsyssteme
- Webserver
- Datenbank(en)
- Netzwerk
- Webanwendung
- LDAP
- DNS
- Fileserver
- Mail
- DHCP

Zum Überwachen.

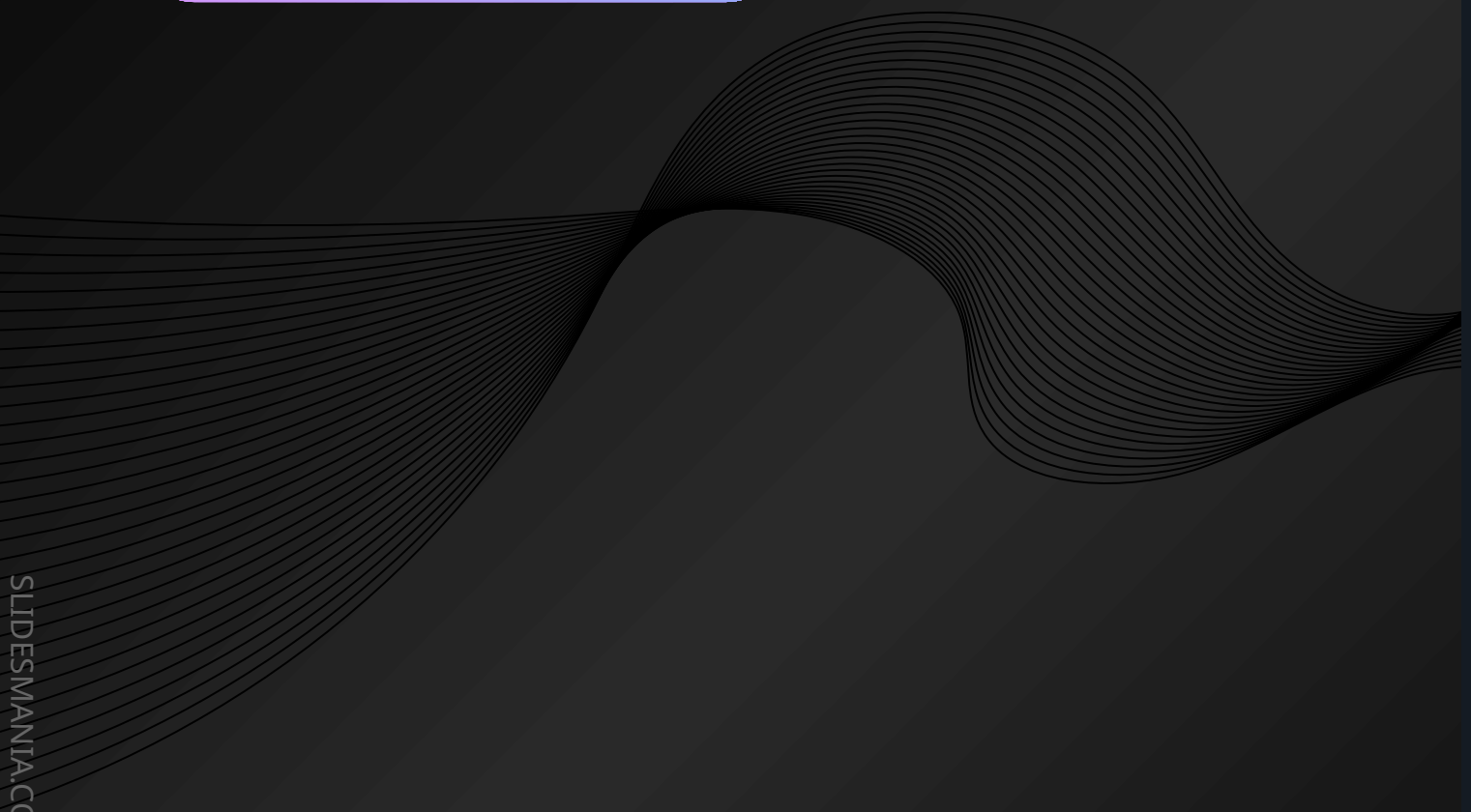
Betriebsyssteme & Fileserver



```
# os-expoerter.nix
{ config, lib, pkgs, ... }:
{
  services.prometheus.exporters.node = {
    enable = true;
    port = 9100;
    enabledCollectors = [
      "logind"
      "systemd"
    ];
    disabledCollectors = [
      "textfile"
    ];
    openFirewall = true;
  };
}
```

Zum Überwachen.

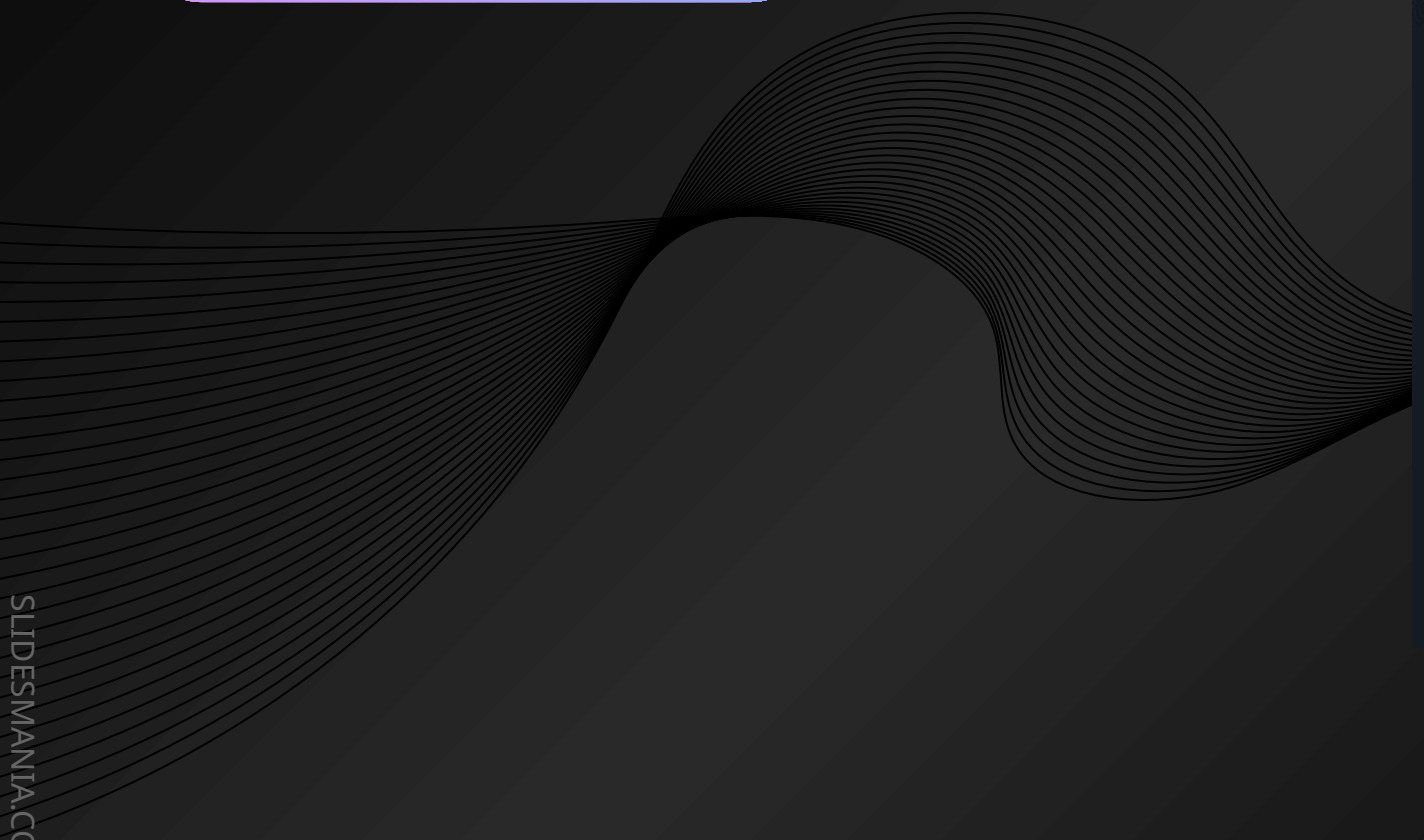
Netzwerk



```
# prometheus.yml
...
  - job_name: 'network'
    metrics_path: /probe
    params:
      module: [ping]  # Look for a HTTP 200 response.
    static_configs:
      - targets:
        - vm1.psa-team03.cit.tum.de
        - vm2.psa-team03.cit.tum.de
        - router.psa-team03.cit.tum.de
        - database.psa-team03.cit.tum.de
        - homeassistant.psa-team03.cit.tum.de
        - vm6.psa-team03.cit.tum.de
        - ldap.psa-team03.cit.tum.de
        - fileserver.psa-team03.cit.tum.de
        - mail.psa-team03.cit.tum.de
        - monitoring.psa-team03.cit.tum.de
        # team routers
        - vm001.psa-team01.cit.tum.de
        - vm1.psa-team02.cit.tum.de
        - vm4.psa-team02.cit.tum.de
        - vm04-02.psa-team04.cit.tum.de
        - vm01.psa-team05.cit.tum.de
        - shika.psa-team06.cit.tum.de
        - vm1.psa-team07.cit.tum.de
        - router.psa-team08.cit.tum.de
        - ns01.psa-team09.cit.tum.de
        - vm02.psa-team10.cit.tum.de
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: blackbox:9115  # muss blackbox:9115 sein
```

Zum Überwachen.

DNS



```
{ ... }:
{
  services.coredns = {
    enable = true;
    config = ''
        (default) {
            bind enp0s8
            root /etc/nixos/dns
            log
            prometheus :9153
        }
...
```

Zum Überwachen.

DHCP

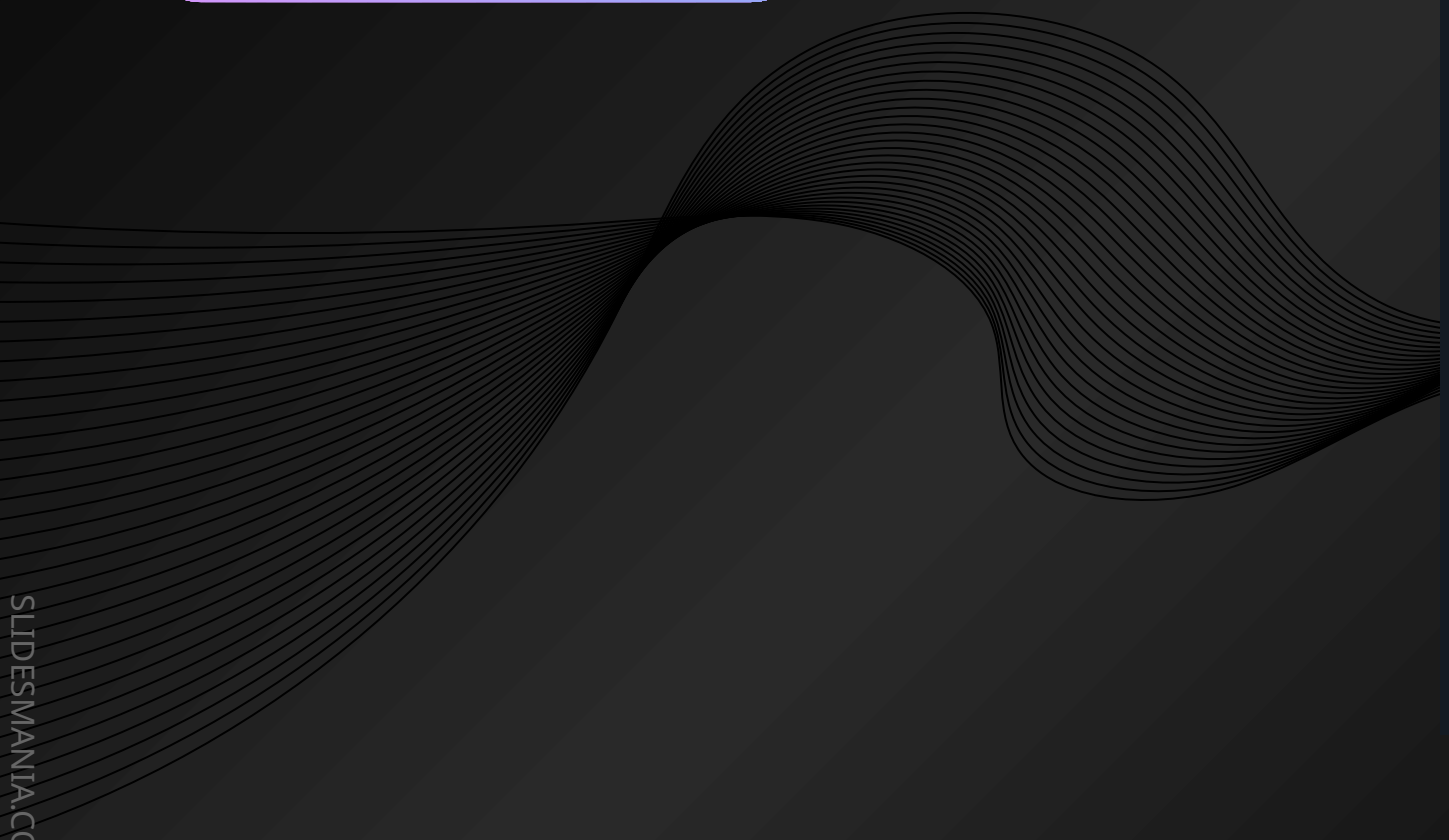
&

```
...
        "control-socket": {
            "socket-type": "unix",
            "socket-name": "/run/kea/kea-dhcp4.socket"
        },
...
```

```
{ config, lib, pkgs, ... }:
{
  services.prometheus.exporters.node = {
    enable = true;
    port = 9100;
    enabledCollectors = [
      "logind"
      "systemd"
    ];
    disabledCollectors = [
      "textfile"
    ];
    openFirewall = true;
  };
  services.prometheus.exporters.kea = {
    enable = true;
    targets = ["/run/kea/kea-dhcp4.socket"];
    port = 9101;
  };
}
```

Zum Überwachen.

Webanwendung



```
cadvisor:
  container_name: cadvisor
  image: gcr.io/cadvisor/cadvisor
  volumes:
    - /:/rootfs:ro
    - /var/run:/var/run:rw
    - /sys:/sys:ro
    - /var/lib/docker/:/var/lib/docker:ro
    - /dev/disk/:/dev/disk:ro
  ports:
    - "8080:8080"
  restart: unless-stopped
  devices:
    - /dev/kmsg
  privileged: true
```

Zum Überwachen.

Webserver

```yaml
# prometheus.yml
...
  - job_name: 'webserver'
    metrics_path: /probe
    params:
      module: [http_2xx]  # Look for a HTTP 200 response.
    static_configs:
      - targets:
        - http://web1.psa-team03.cit.tum.de
        - https://web1.psa-team03.cit.tum.de
        - https://web2.psa-team03.cit.tum.de
        - https://web3.psa-team03.cit.tum.de
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: 192.168.3.6:9102  # muss blackbox:9115 sein
```

```yaml
# blackbox.yml
modules:
  http_2xx:
    prober: http
    timeout: 5s
    http:
      valid_http_versions: ["HTTP/1.1", "HTTP/2.0"]
      valid_status_codes: []
      method: GET
      follow_redirects: true
      fail_if_ssl: false
      fail_if_not_ssl: false
      tls_config:
        insecure_skip_verify: true
      preferred_ip_protocol: "ip4" # defaults to "ip6"
      ip_protocol_fallback: false  # no fallback to "ip6"
  ping:
    prober: icmp
    timeout: 5s
    icmp:
      preferred_ip_protocol: "ip4"
```

Zum Überwachen.

Datenbank(en)

```
# database.nix & database-backup.nix
services.prometheus.exporters.postgres = {
  enable = true;
  port = 9100;
  runAsLocalSuperUser = true;
};
```

&

```
services.prometheus.exporters.script = {
  enable = true;
  port = 9100;
  settings.scripts = [
    { name = "db-check"; script = "nc -zv 192.168.4.5 3306"; }
  ];
};
```

Zum Überwachen.

LDAP

```
git clone https://github.com/jcollie/openldap_exporter.git
cd openldap_exporter
virtualenv --python=/usr/bin/python2 /opt/openldap_exporter
/opt/openldap_exporter/bin/pip install --requirement requirements.txt
cp openldap_exporter.py /opt/openldap_exporter
cp openldap_exporter.yml /opt/openldap_exporter
vi /opt/openldap_exporter/openldap_exporter.yml

# edit configuration file
cp openldap_exporter.service /etc/systemd/system
systemctl daemon-reload
systemctl enable openldap_exporter
systemctl start openldap_exporter

docker build . -t openldap_exporter
```

Zum Überwachen.

Mail

```
services.prometheus.exporters.postfix = {
  enable = true;
  port = 9154;
  ...
};
```

Zeit für eine

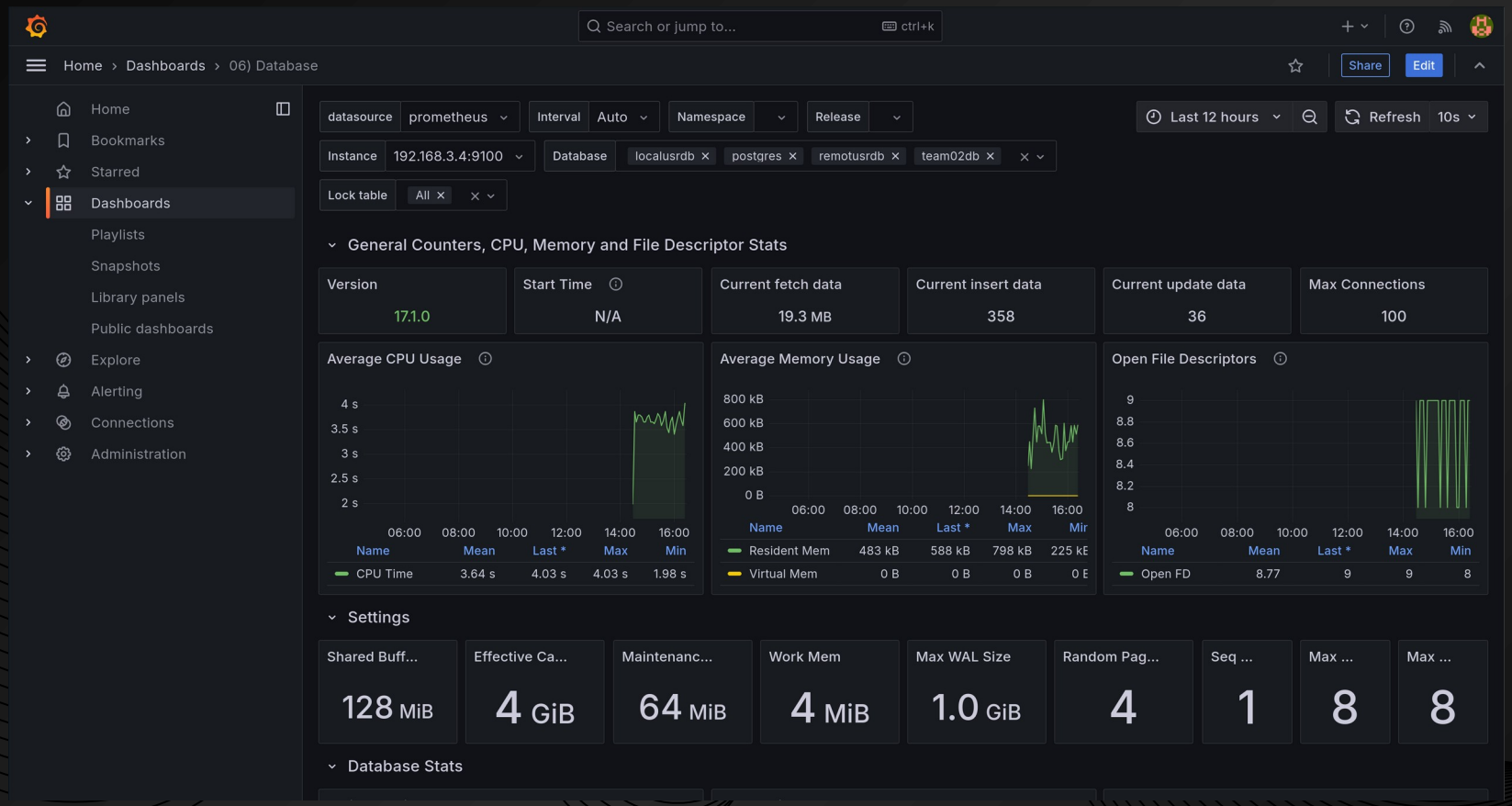
Live-Demo!

04

Grafana

Visualisierung

Grafana.



Plattformübergreifend

Nicht Prometheus spezifisch.
Bsp. MySQL, Graphite, ...

Grafische Darstellung

Von allerlei Daten.
Unterteilung in Dashboards.
Viele vorkonfigurierte Elemente.

Konfiguration.

Erster Start

Dashboards

Out-of-the-Docker

Web-Interface auf
http://131.159.74.56:60313/

Login

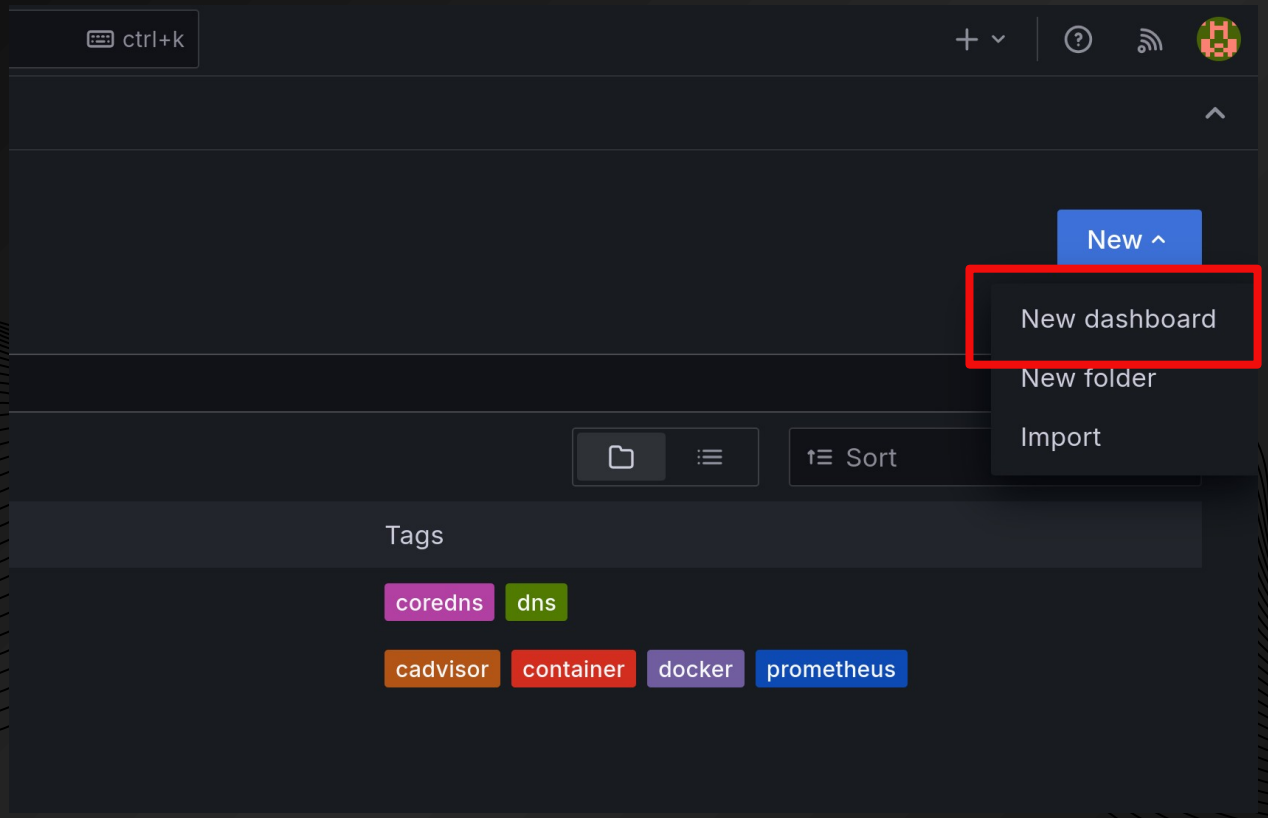
Erster Login:
Username: admin
Passwort: admin

Neues Passwort vergeben, fertig.

Konfiguration.

Erster Start

Dashboards



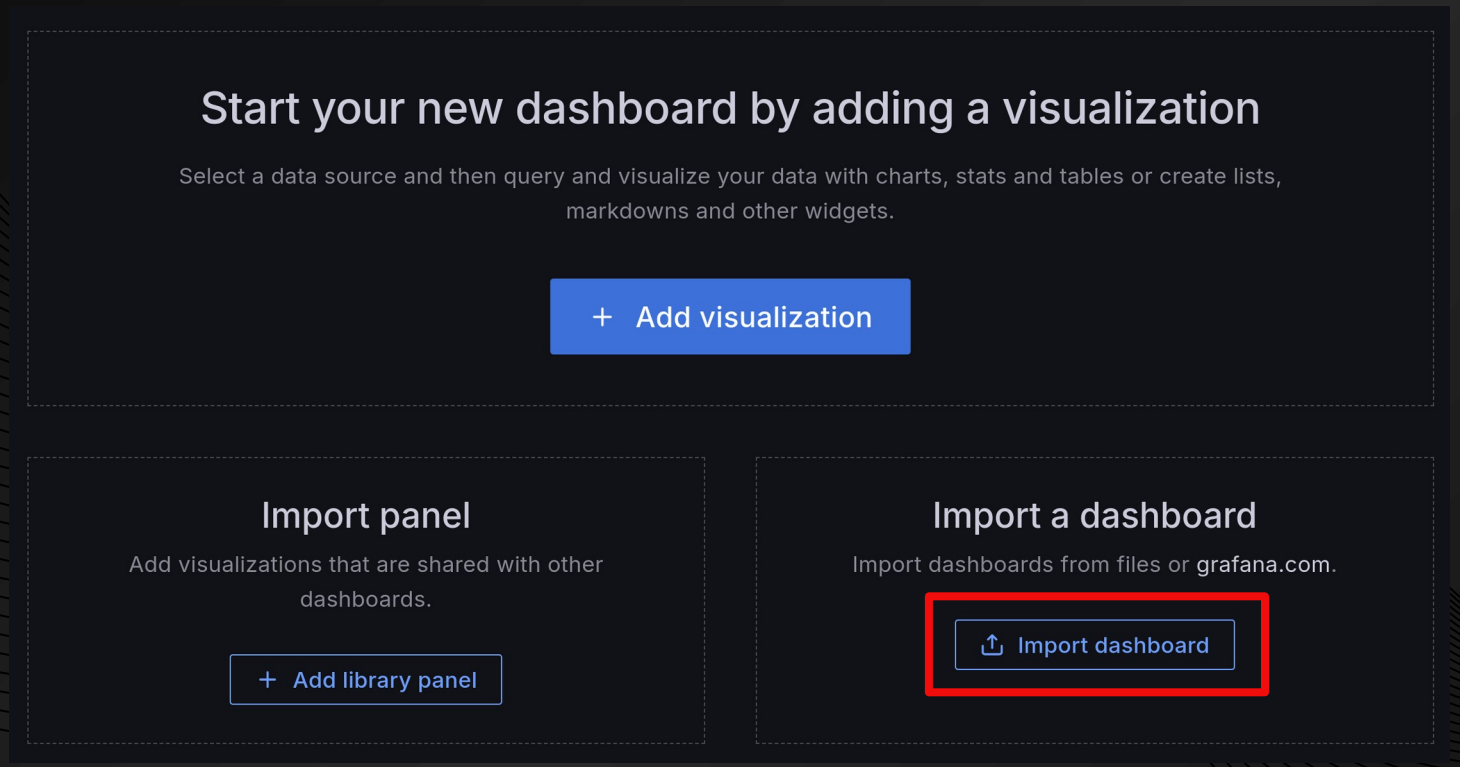
Neues Dashboard

Hinzufügen

Konfiguration.

Erster Start

Dashboards



Neues Dashboard

Hinzufügen

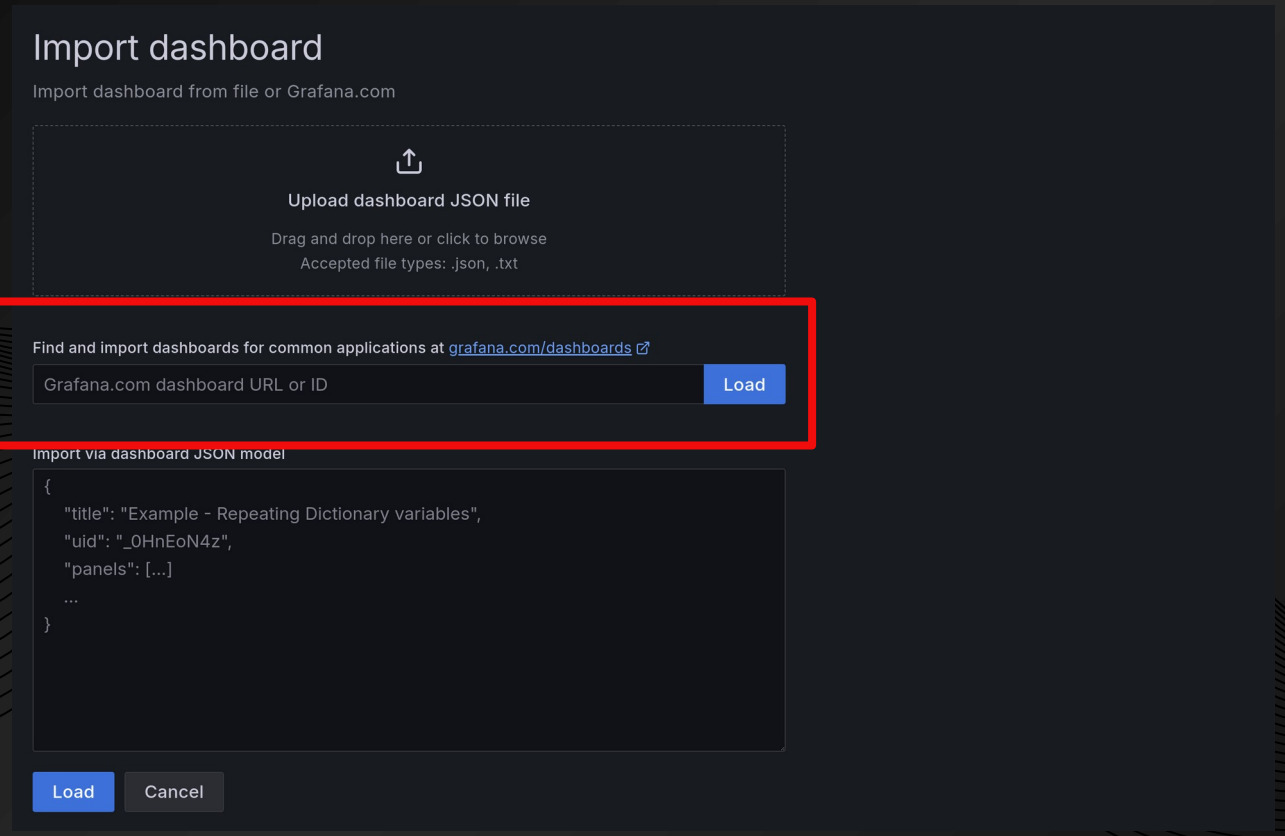
Import / Sebst Bauen

Dashboard Art Angeben
Importieren, oder selbst bauen

Konfiguration.

Erster Start

Dashboards



Neues Dashboard

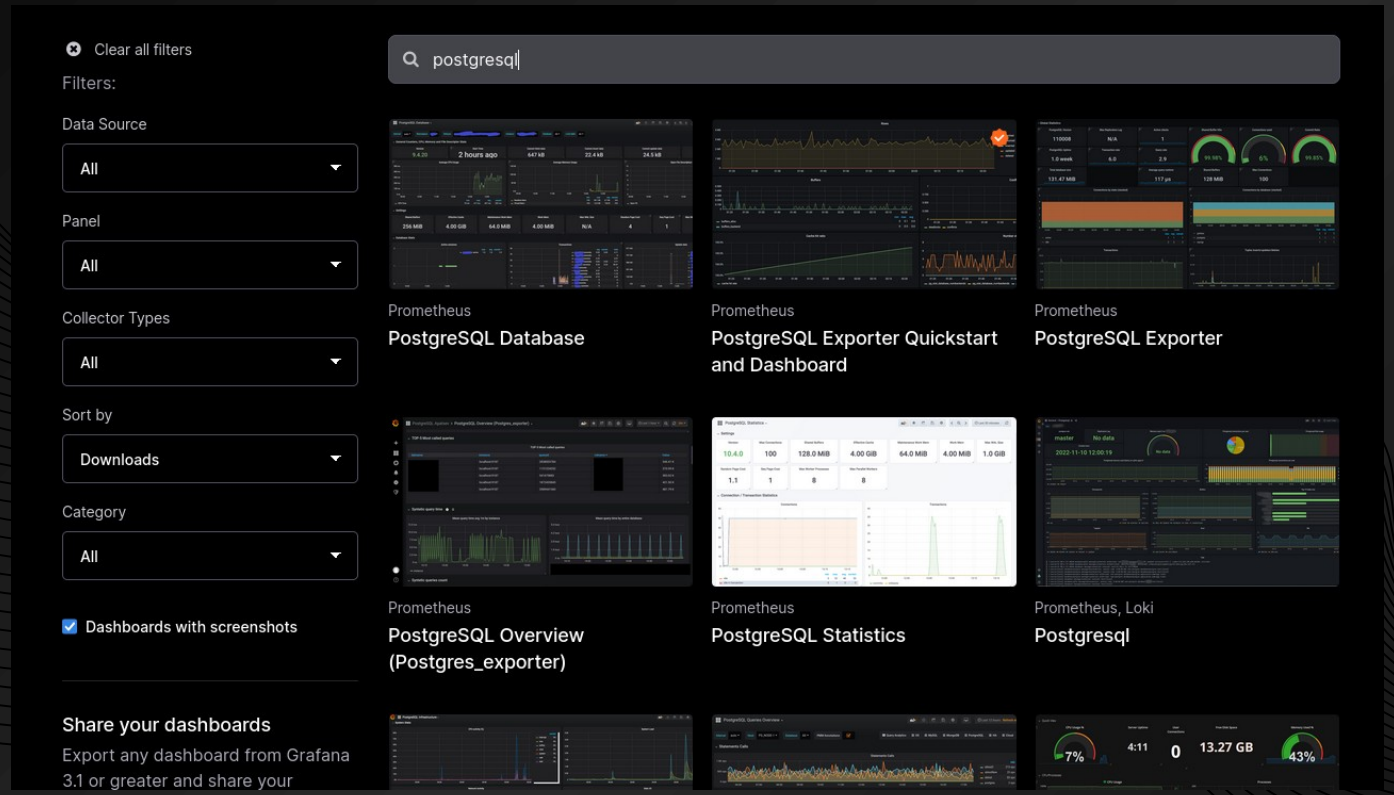
Hinzufügen

Import / Sebst Bauen

Dashboard Art Angeben
Importieren, oder selbst bauen

Konfiguration.

Erster Start | **Dashboards**



Neues Dashboard

Hinzufügen

Import / Sebst Bauen

Dashboard Art Angeben
Importieren, oder selbst bauen

Suche Dashboards

https://grafana.com/grafana/dashboards/

Zeit für eine

Live-Demo!

05

Alerts

AlertManager
Slack Integration
Unsere Alerts

Konfiguration.

alertmanager.yml | Slack | alert.rules.yml

Route

Wer? Welche Art?

Receiver

Receiver Konfiguration

```yaml
# alertmanager.yml
route:
  receiver: slack
  repeat_interval: 1m
receivers:
  - name: slack
    slack_configs:
      - channel: "#all-praktikum-system-administration-ws2425"
        send_resolved: true
        api_url: "https://hooks.slack.com/services/
        title: Alert
        text: >-
          {{ range .Alerts -}}
          *Alert:* {{ .Annotations.title }}{{ if .Labels.severity }} - `{{ .Labels.severity }}`{{ end }}
          *Description:* {{ .Annotations.description }}
          *Details:*
            {{ range .Labels.SortedPairs }} • *{{ .Name }}:* `{{ .Value }}`
            {{ end }}
```

Konfiguration.

alertmanager.yml

Slack

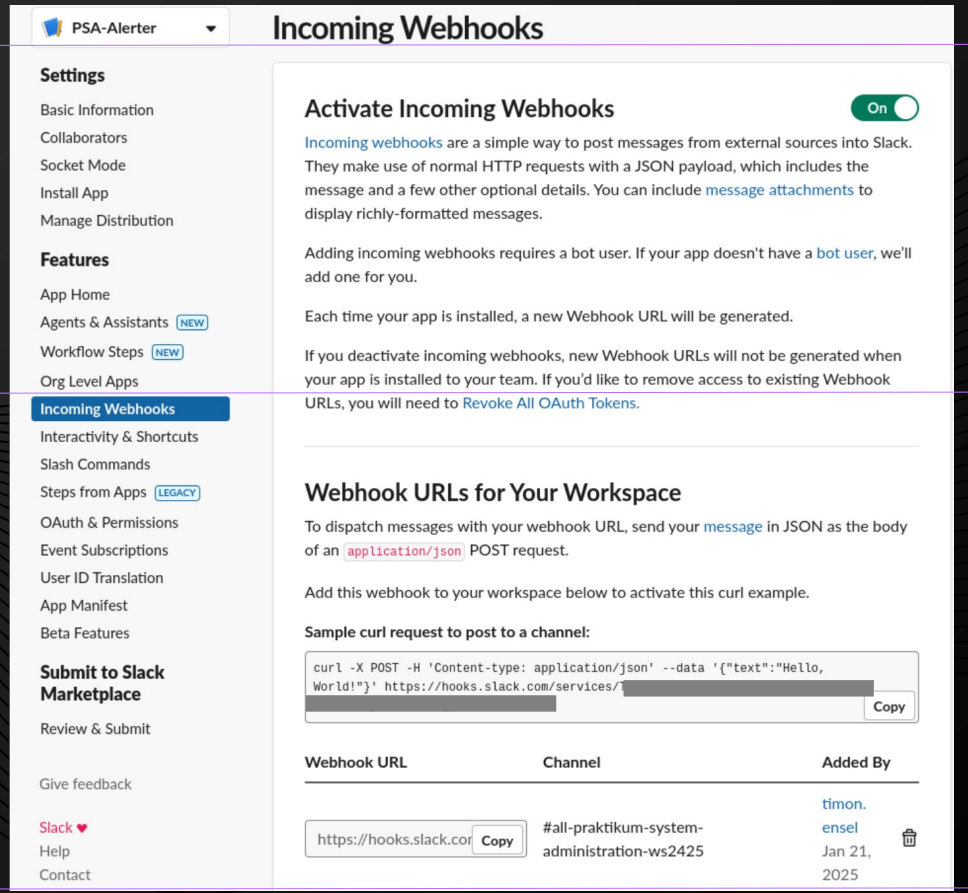
alert.rules.yml

Slack App

Neue Slack App

Webhook

Webhook einrichten



Konfiguration.

alertmanager.yml | Slack | alert.rules.yml

Alert Rule Datei

Wo?

Alertmanager

Target angeben.

```
# prometheus.yml
rule_files:
  - "alert.rules.yml"
alerting:
  alertmanagers:
    - static_configs:
        - targets: ["alertmanager:9093"]
```

Konfiguration.

alertmanager.yml | Slack | alert.rules.yml

Gruppe

Name.
Regeln.

Alert

Titel.
Beschreibung.
Level.

```yaml
# alert.rules.yml
groups:
  - name: <Gruppen-Name>
    rules:
      - alert: <Alert-Titel>
        expr: <Bedingung>
        for: <Dauer>
        labels:
          severity: <Level>
        annotations:
          summary: <Kurz-Beschreibung>
          description: <Beschreibung>
```

Konfiguration.

alertmanager.yml

Slack

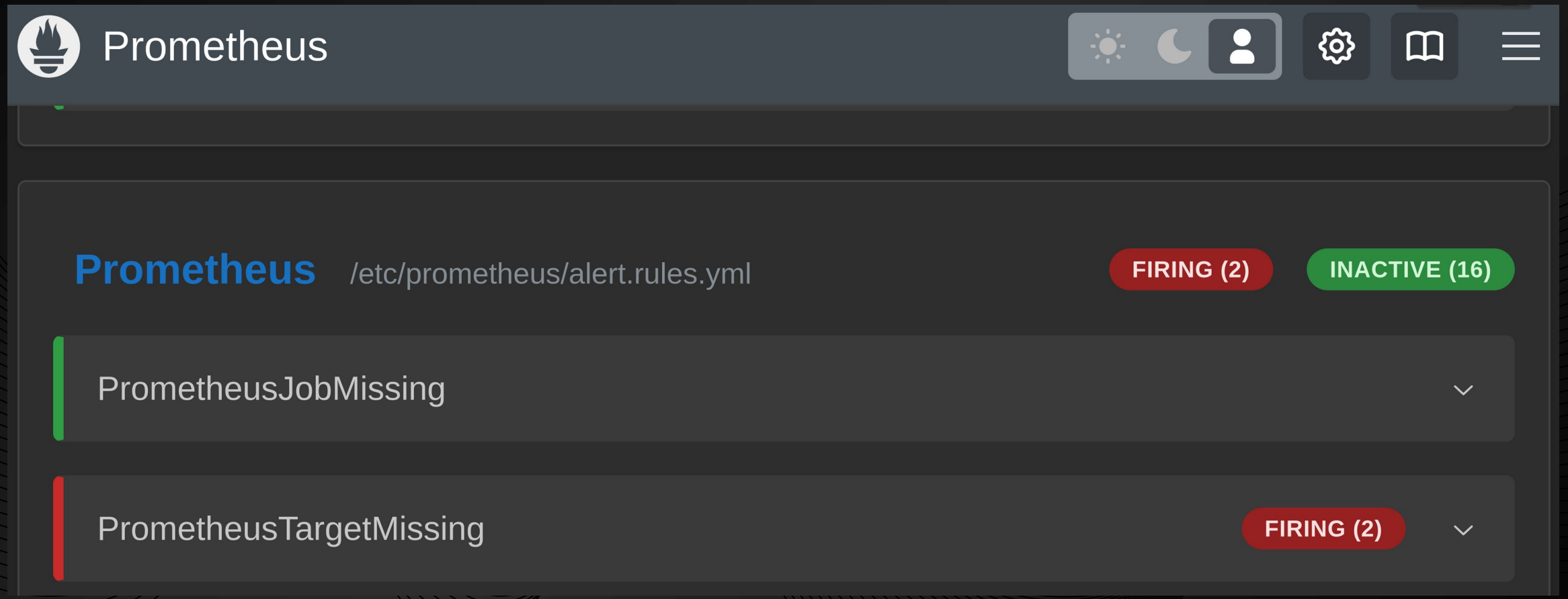
alert.rules.yml

Alert

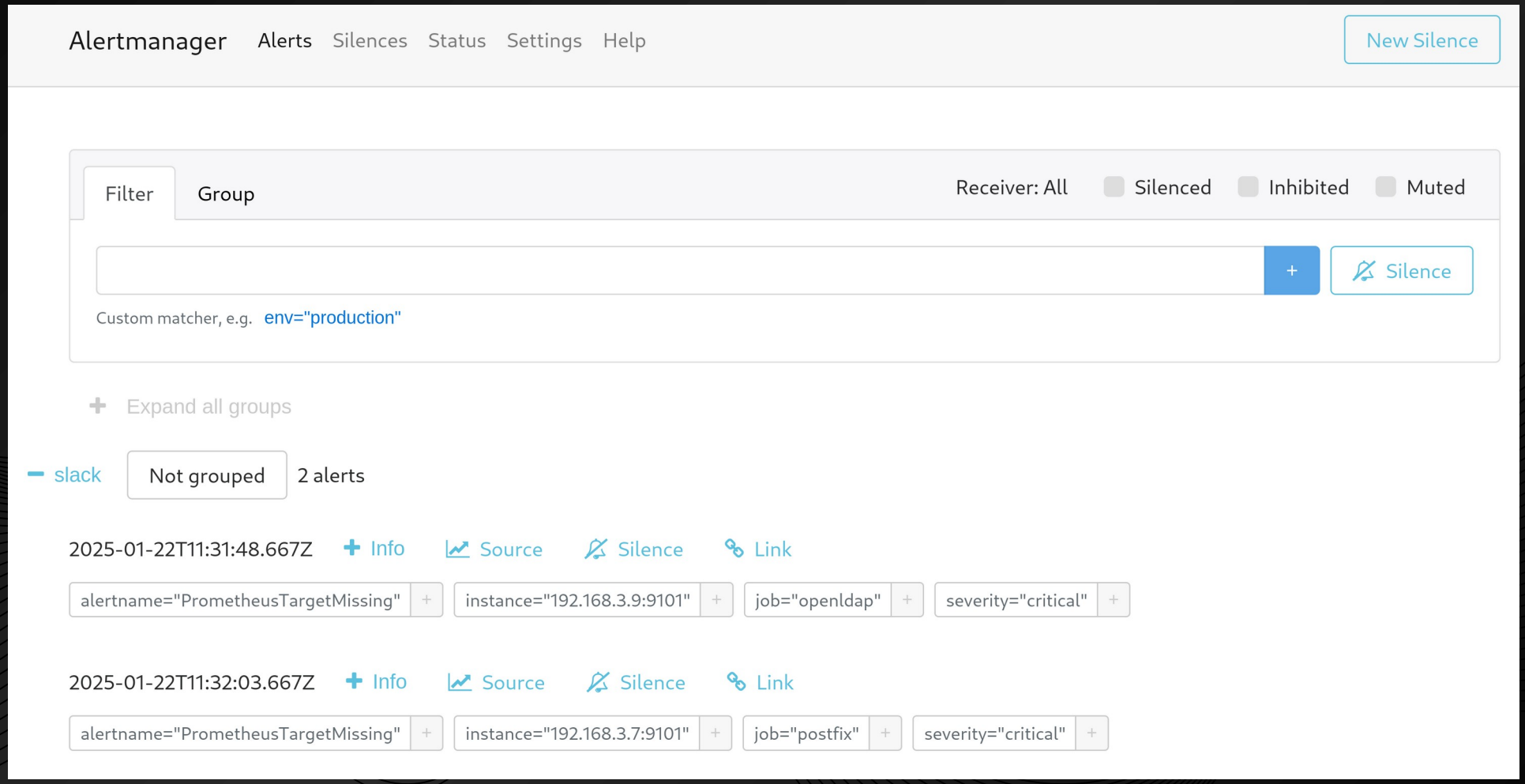
Name.
Expression (Regel).
Dauer.
Level.
Bescheibung.

```
# alert.rules.yml
      - alert: PostgresqlDown
        expr: pg_up == 0
        for: 0m
        labels:
          severity: critical
        annotations:
          summary: Postgresql down (instance {{ $labels.instance }})
          description: "Postgresql instance is down\n  VALUE = {{ $value }}\n  LA
```

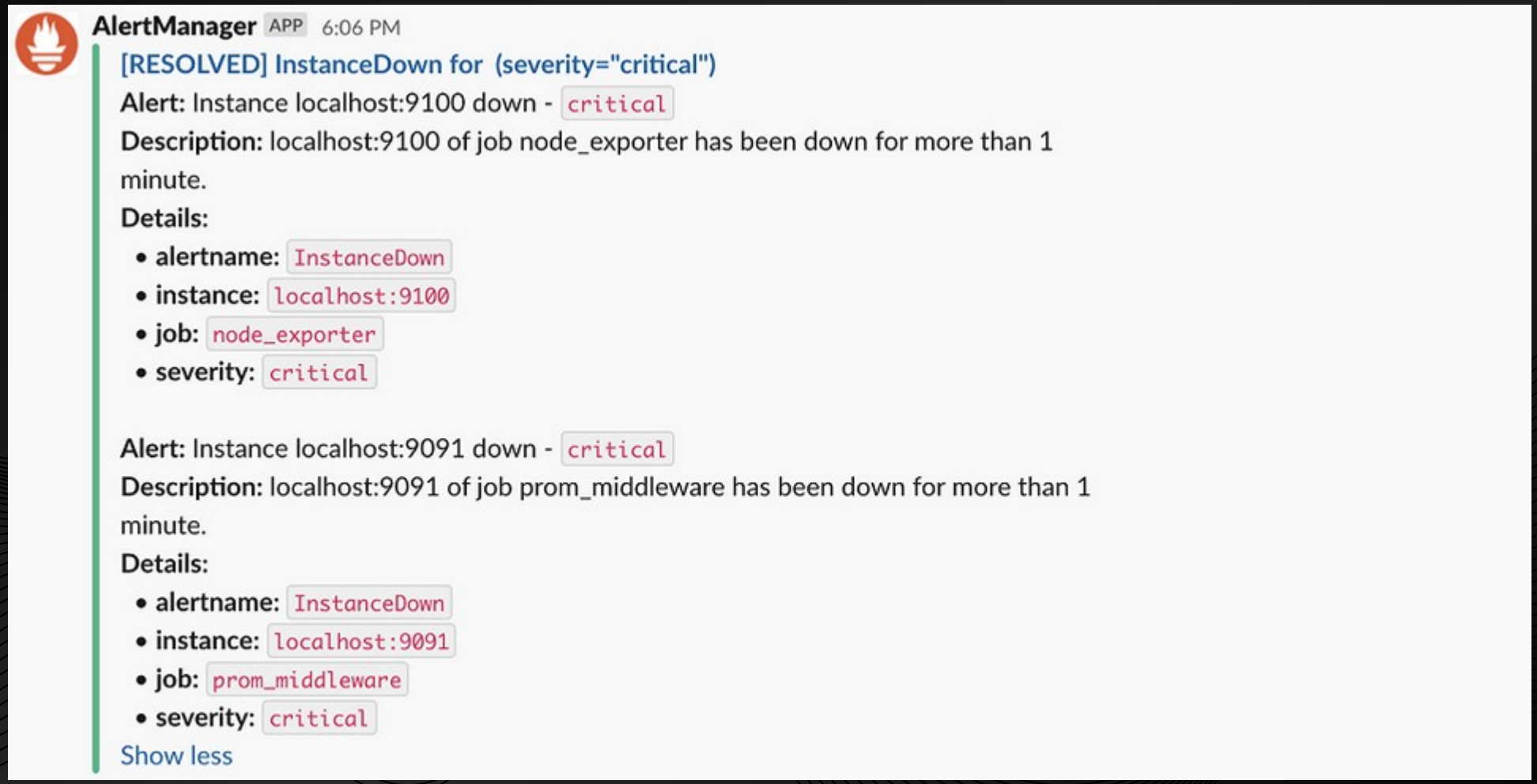
Alert! - Prometheus

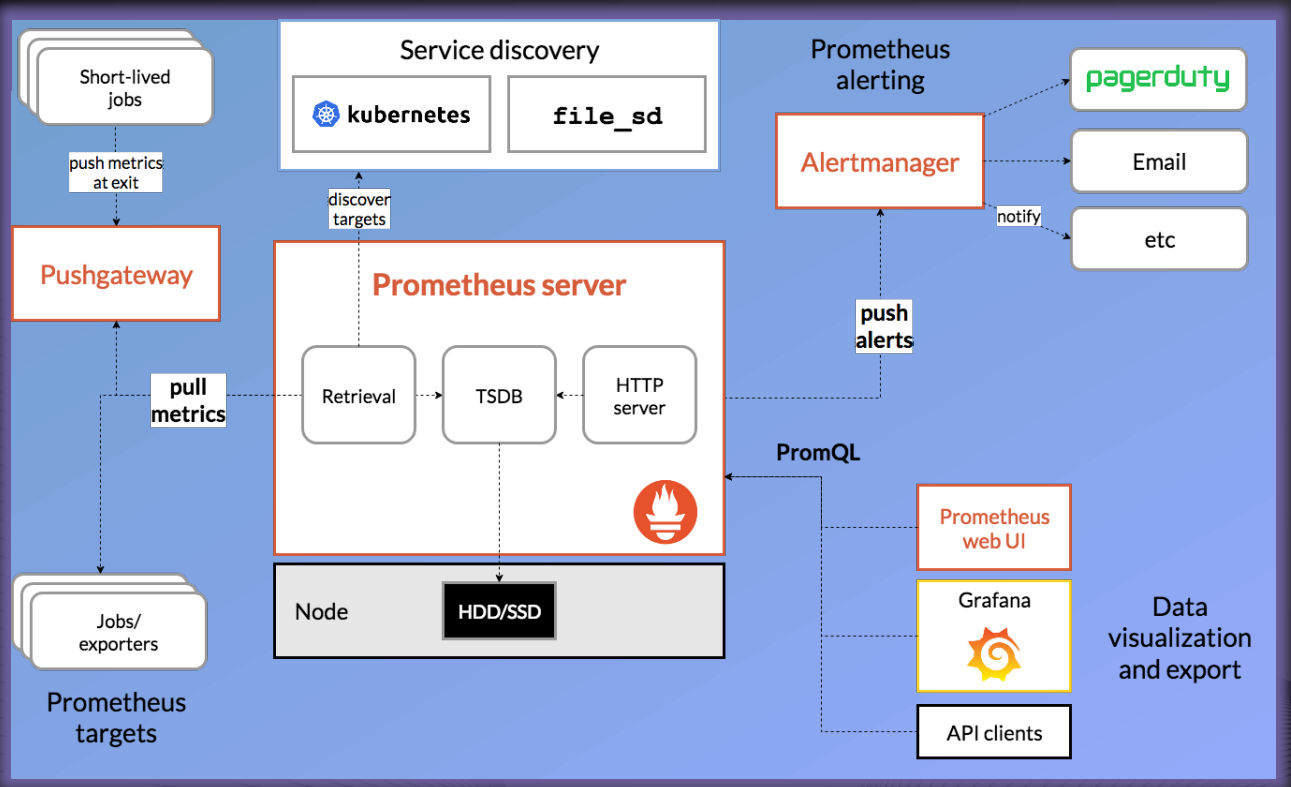


Alert! - Alertmanager



Alert! - Slack





Danke!

Fragen?

Benamin Liertz
Timon Ole Ensel

Credits.

Presentation Template: SlidesMania

Sample Images: Unsplash

Fonts used in this presentation: DM Sans