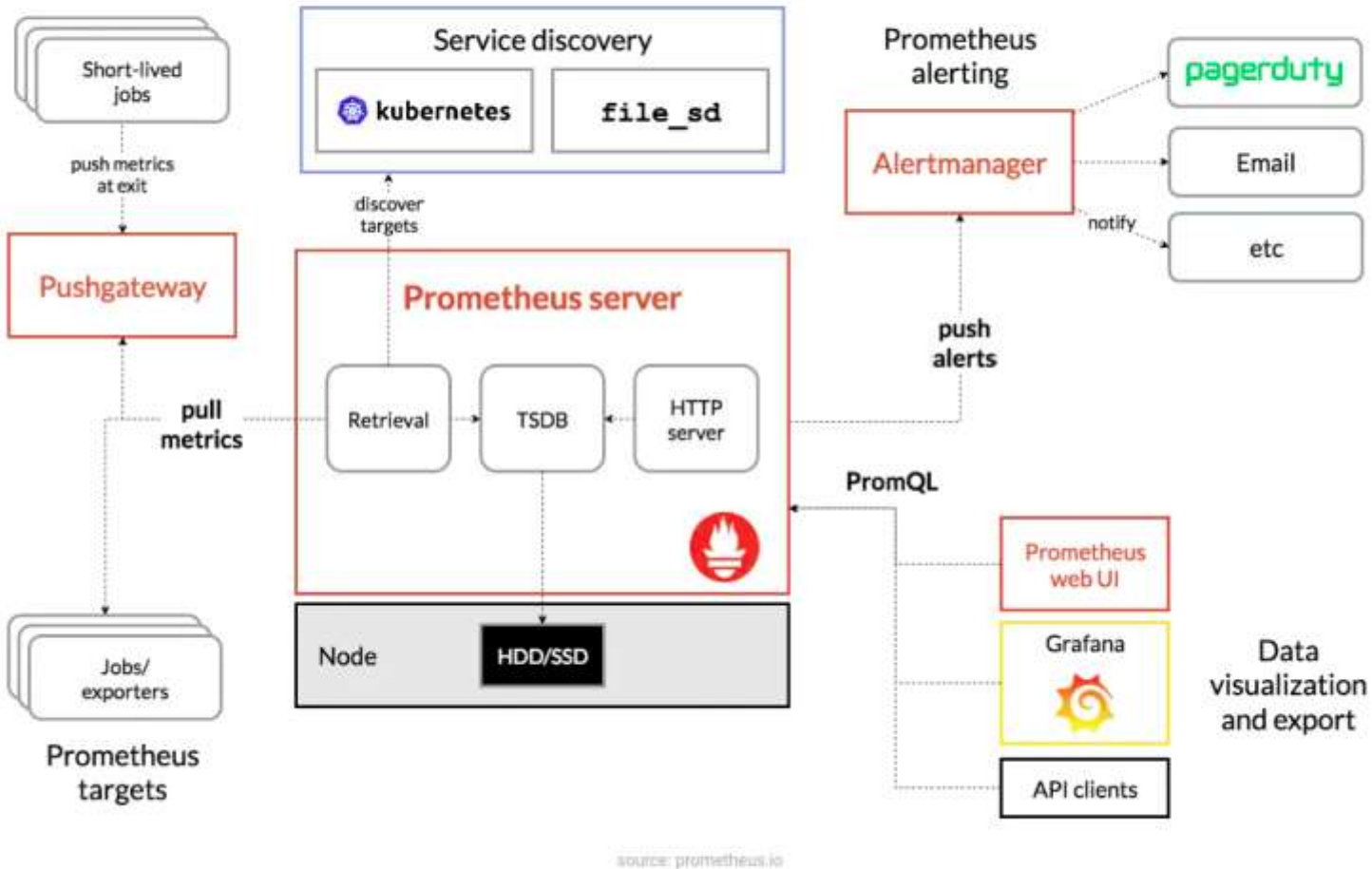
Monitoring with Prometheus part 2

Agenda

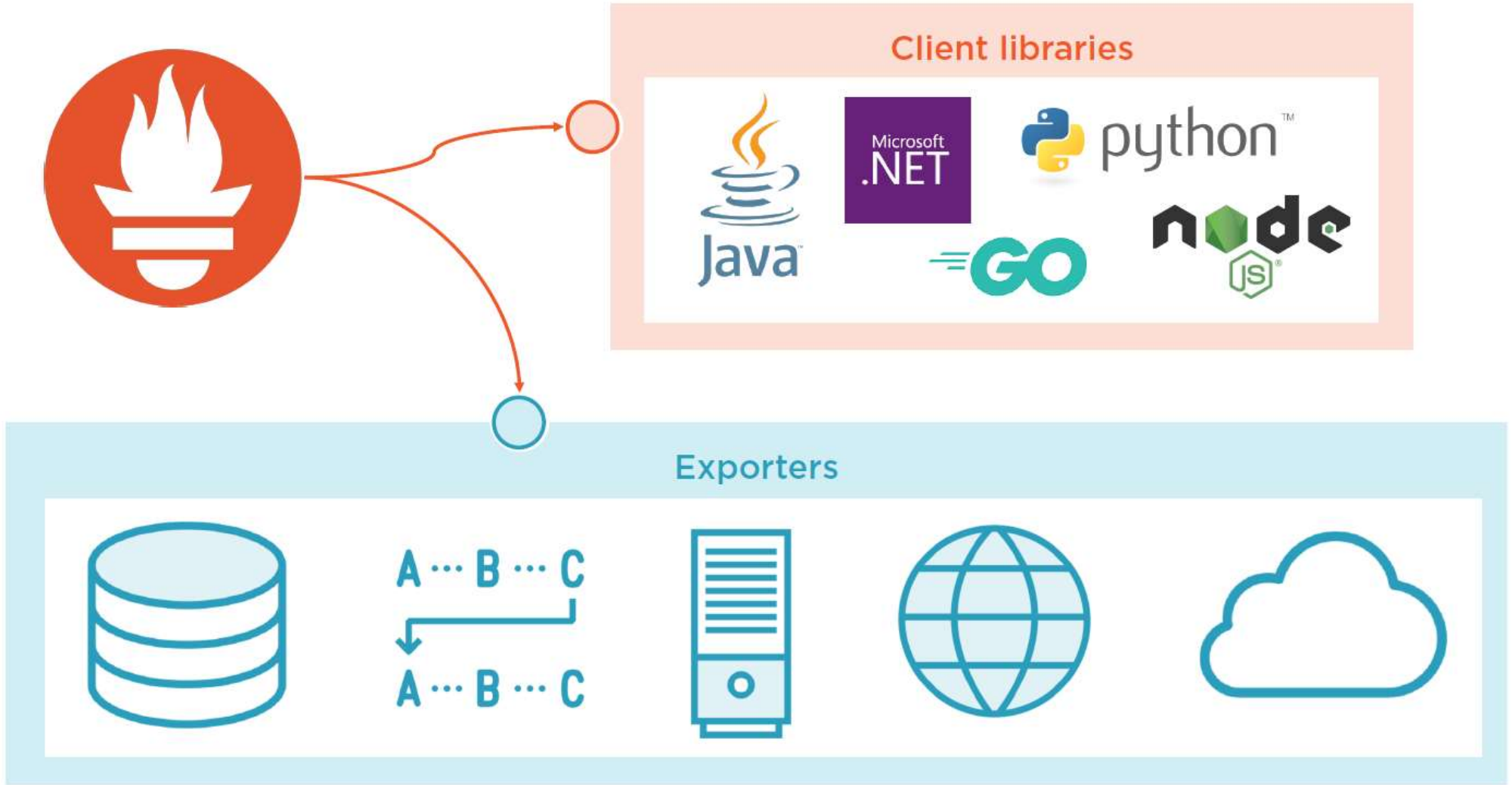
- exporter
- Service discovery
- Rules files
- Alert manger
- Intro to grafana
- PromeQl advanced

Prometheus Architecture

Here is the high-level **architecture of Prometheus.**



source: prometheus.io


Client libraries
Java
Microsoft .NET
python
GO
node
Exporters
A ··· B ··· C
A ··· B ··· C

Overview

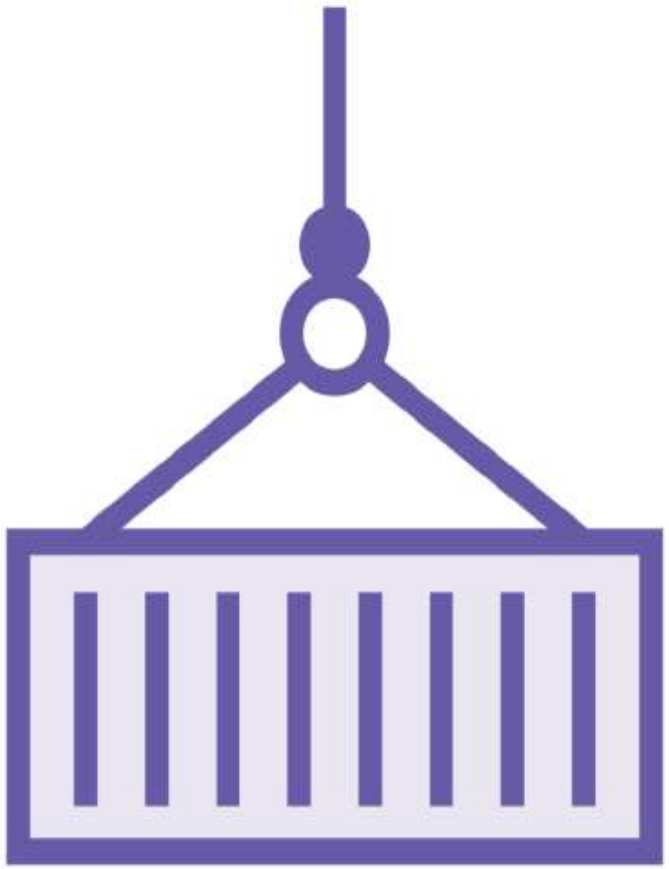
What are exporters?

What do they do?

Available exporters

Node exporter

Prometheus Exporters



Translate metrics to Prometheus format

Existing systems

- Linux
- MySQL

Some are built-in

https://prometheus.io/docs/instrumenting/exporters/

Using Exporters

Identify and install desired exporter on target system

Create Prometheus job to pull data from exporter

Prometheus will gather metrics and report on job status

You can set up multiple targets and multiple exporters

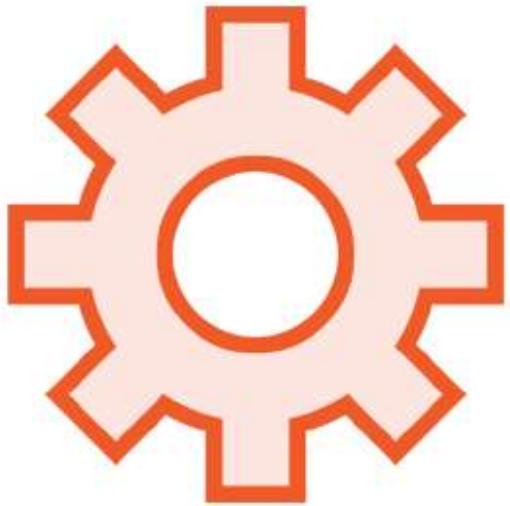
Prometheus Exporters

You can write your own exporter

Prometheus provides guidelines

https://prometheus.io/docs/instrumenting/writing_exporters/

Node Exporter



System Metrics

Linux based systems

CPU, memory, disk, network

Download

https://prometheus.io/download/
#node_exporter

```
wget https://github.com/
prometheus/node_exporter/releases/
download/v*/node_exporter-
*.*linux-amd64.tar.gz
```

◀ Get the exporter tar

```
tar xvfz node_exporter-*.*-
amd64.tar.gz
```

◀ Unpack the tar

```
cd node_exporter-*.*-amd64
```

◀ Move into folder

```
./node_exporter > node.out 2>&1
```

◀ Start the exporter

```
curl http://localhost:9100/metrics
```

◀ Check that metrics are publishin

Exporter Setup

Port 9100

Prometheus must be able to reach the target machine to pull metrics

Change default port

```
--web.listen-address=":9200"
```

List of parameters and defaults

```
node_exporter --help
```

```
scrape_configs:
  - job_name: 'node_exporter'
    static_configs:
    - targets: ['172.31.27.103:9100']
```

Prometheus Configuration for Exporter

Create a job to pull metrics from exporter

IP will depend on your network setup (internal vs. external IP)

Restart Prometheus to pick up the change

Demo

Node Exporter

- Install and configure on Linux machine

View results in Prometheus

Overview

Database exporters for Prometheus

MySQL server exporter

- Install and configure

Database Metrics

More than just system metrics

Database-specific metrics

- View into database performance

Prometheus database exporters

- https://prometheus.io/docs/instrumenting/exporters/#databases

```
CREATE USER 'mysqld_exporter'@'localhost' IDENTIFIED BY 'password';

GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO
'mysqld_exporter'@'localhost';

FLUSH PRIVILEGES;

export DATA_SOURCE_NAME='mysqld_exporter:password@(localhost:3306)/'
```

MySQL Database Configuration

Create a database user

- Grant permissions

Provide credentials to exporter

- Many different options depending on your system

```
wget https://github.com/
prometheus/mysqld_exporter/releases
/download/v*/mysqld_exporter-
*.*.linux-amd64.tar.gz

tar xvfz mysqld_exporter-*.*-
amd64.tar.gz


cd mysqld_exporter-*.*-amd64

./mysqld_exporter > mysqld.out 2>&1

curl http://localhost:9104/metrics
```

- Get the exporter tar
- Unpack the tar
- Move into folder
- Start the exporter
- Check that metrics are publishing

prometheus/prometheus.yml

```
scrape_configs:
  - job_name: 'mysqld_exporter'
    static_configs:
    - targets: ['172.31.18.7:9104']
```

Prometheus Configuration for Exporter

Create a job to pull metrics from exporter

IP will depend on your network setup (internal vs. external IP)

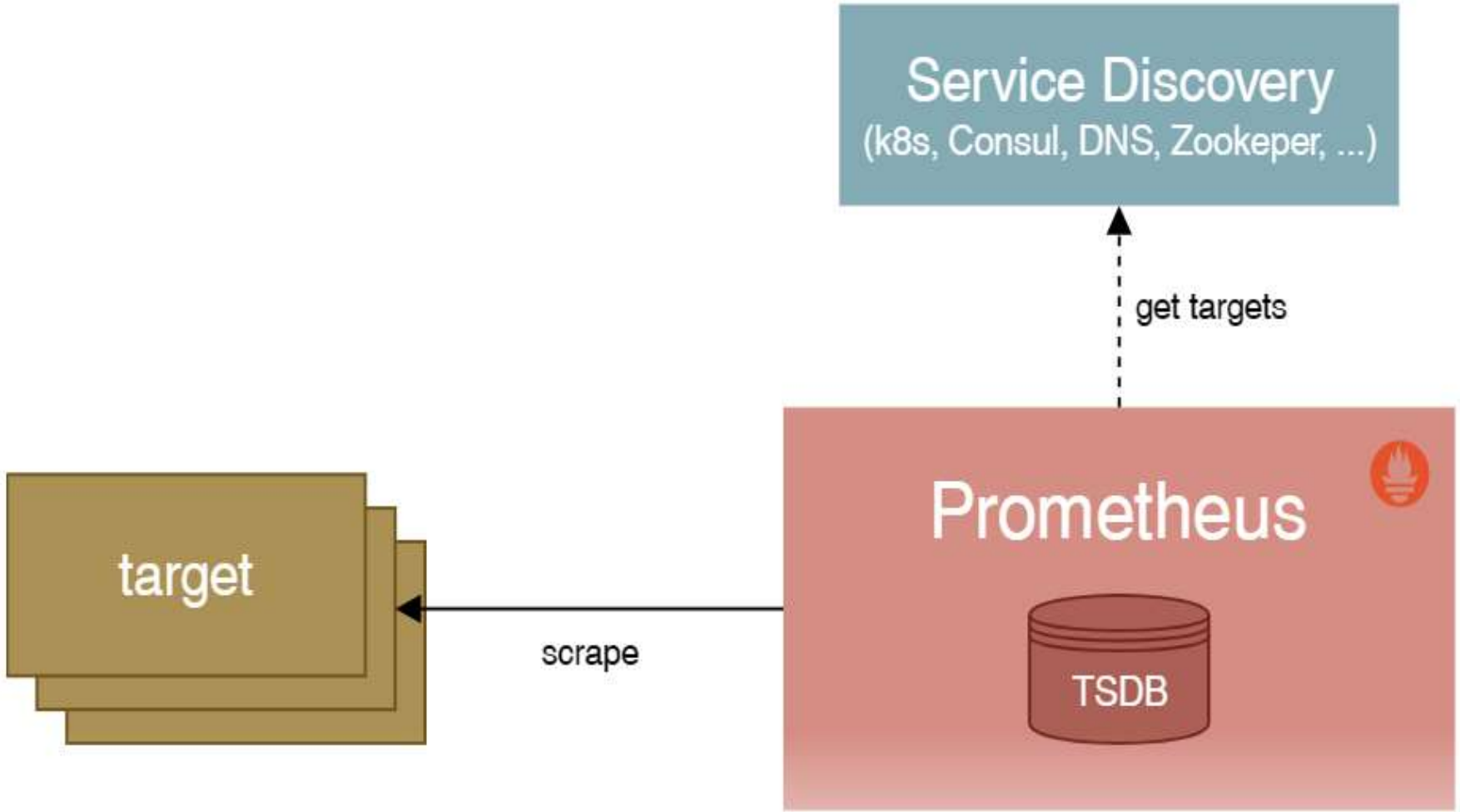
Restart Prometheus to pick up the change

Demo

MySQL Exporter

- Install and configure on Linux machine

View results in Prometheus


Service Discovery
(k8s, Consul, DNS, Zookeper, ...)
get targets
Prometheus
TSDB
target
scrape

Overview

Alerting

- What
- Why
- When

Alertmanager

- Connect to Prometheus

Alert

The state of being watchful for possible danger.

Quick to notice any unusual and potentially dangerous circumstances.

Warn of a danger, threat, or problem, typically with the intention of having it avoided or dealt with.

-Oxford Languages

Alertmanager

Handles alerts sent from Prometheus

Manage and filter alerts

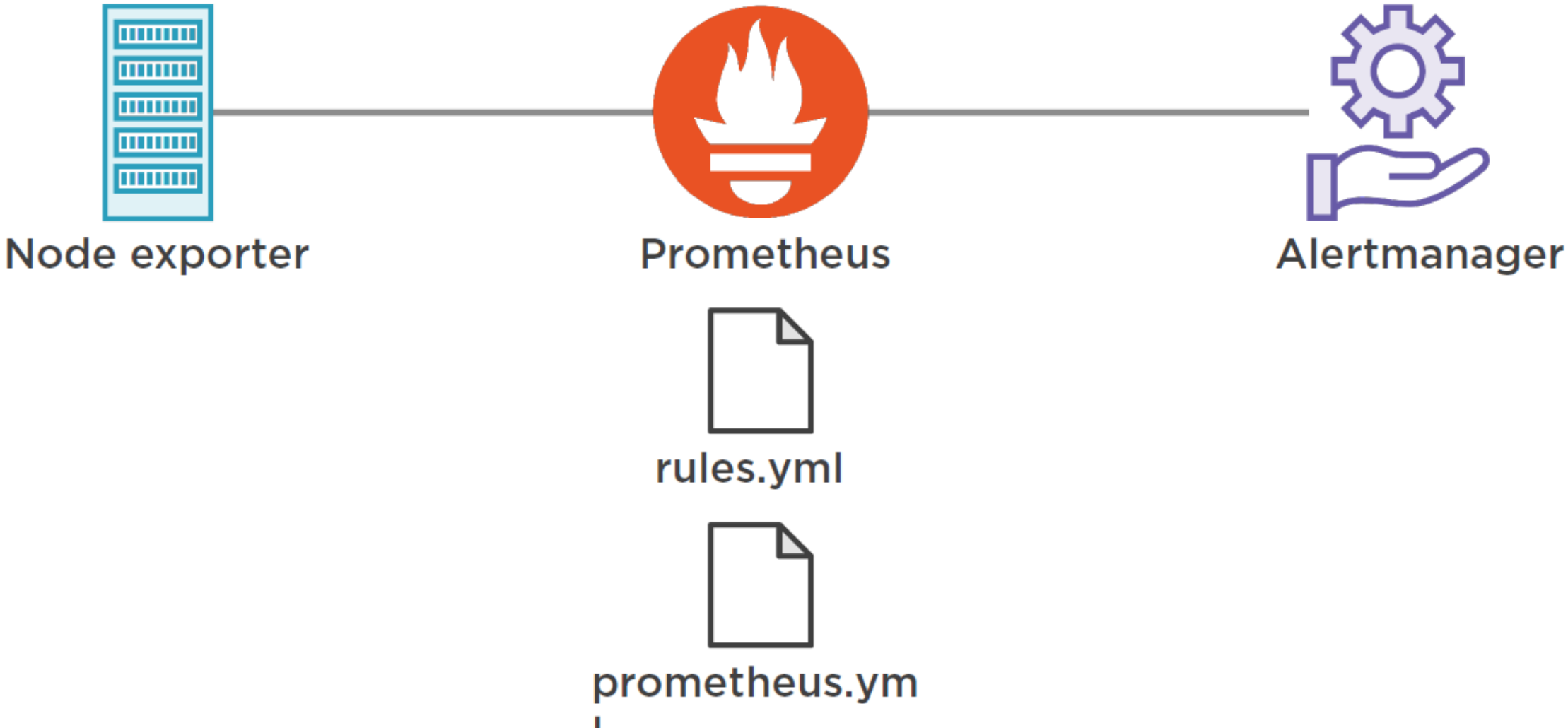
- Deduplicate
- Group
- Silence
- Inhibit

Notification routing

- Email, on-call systems, chat platforms

Download, install, and configure

Defining Rules



```
groups:
- name: example
  rules:
  - alert: InstanceDown
    expr: up == 0
    for: 1m
    labels:
      severity: critical
    annotations:
      summary: Instance is down
```

- rules.yml in directory with prometheus.yml

- Name of alert to display
- Expression to define alert
- Optional: duration needed for alert to fire

```
# Alertmanager configuration
alerting:
  alertmanagers:
  - static_configs:
    - targets:
      - localhost:9093

rule_files:
  - "rules.yml"
```

◀ prometheus.yml

◀ Alertmanager target

◀ list rules.yml as a rule file

Demo

Setup:

- Running Prometheus instance
- Node exporter on separate machine

Create rules.yml file

Modify prometheus.yml file

Install Alertmanager

See alert fire

Demo

Slack receiver

What you need from Slack

Modify alertmanager.yml file

See alerts in slack channel

```
global:
  slack_api_url: 'https://hooks.slack.com…'
route:
  receiver: 'slack-notifications'
receivers:
- name: 'slack-notifications'
  slack_configs:
  - channel: '#prometheus-alerts'
    send_resolved: true
```

- alertmanager.yml
- URL to your Slack workspace
- Name for receiver
- Reference name you used
- Add configs for receiver


Grafana

Summary

- Prometheus architecture revision
- Node exporter
- Mysql exporter
- Service discovery
- Rules file
- Aler manger
- PromQL
- Grafana

Q&A

Thank you