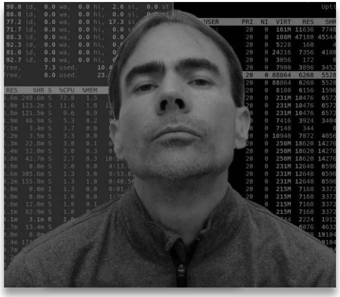
Using Prometheus to Monitor Linux and Kubernetes

Repository: https://github.com/daveprowse/prom-live



Dave Prowse

Author and Trainer,
Technology Consultant

Rev F

Pearson

AGENDA	
Segment 1	Introduction
Segment 2	Prometheus Installation
Segment 3	Basic Querying
Segment 4	Dashboarding
Segment 5	Monitoring Metrics
Segment 6	Instrumenting Code
Segment 7	Monitoring Linux Systems
Segment 8	Monitoring Kubernetes

Repository: https://github.com/daveprowse/prom-live

Audience Poll

What platform will you be installing Prometheus to?

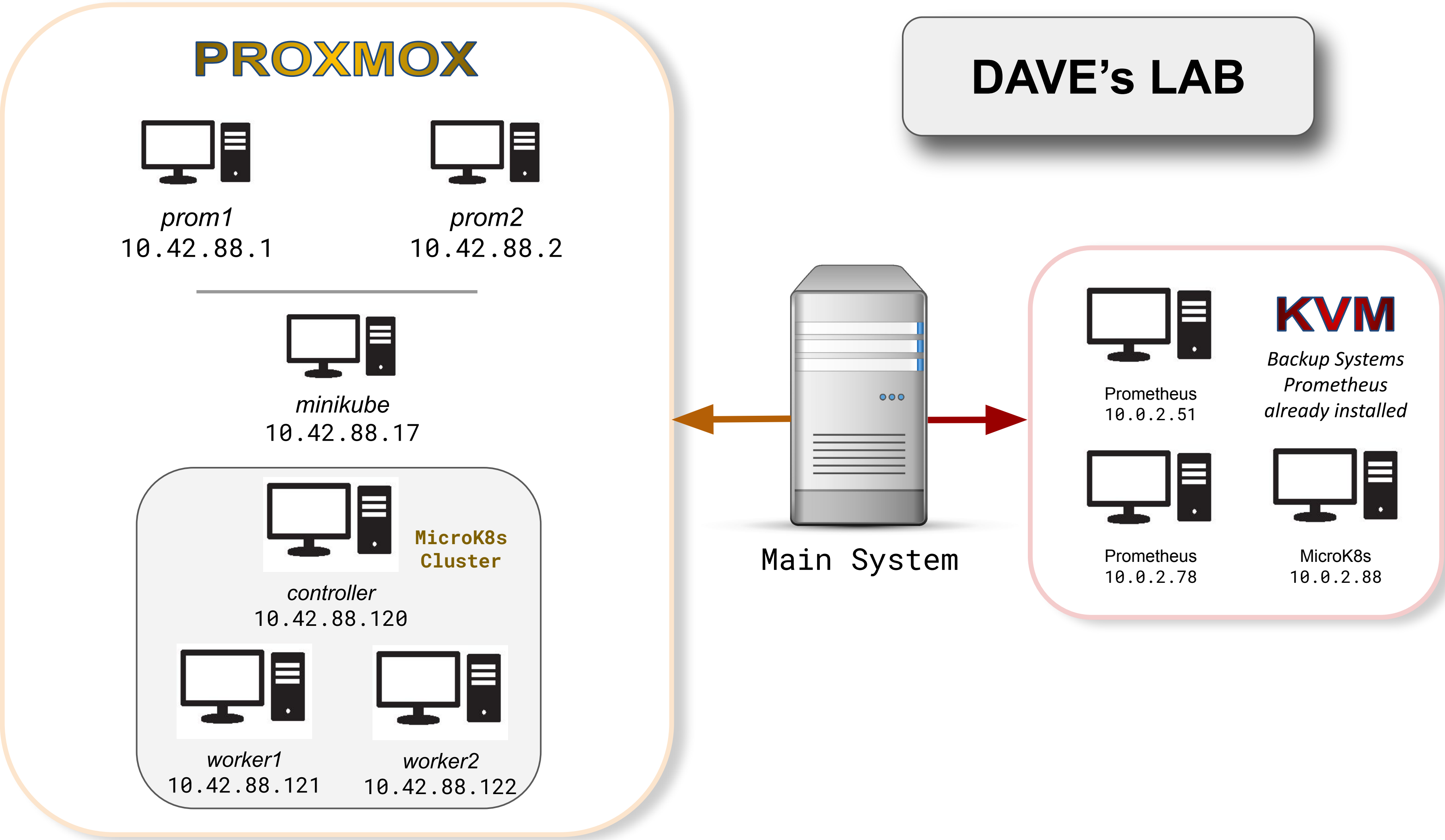
A. Ubuntu Server (recommended)
B. Debian Server (recommended)
C. CentOS / RHEL / Fedora
D. macOS
E. Windows

Audience Poll

What Prometheus concepts do you want to learn the most?

A. Basic Prometheus Usage
B. Monitoring Linux servers
C. Monitoring on-premises Kubernetes
D. Monitoring cloud-based Kubernetes
E. Prometheus Certification (PCA)
F. Other (shout it out in the group chat!)

DAVE’s LAB



PROXMOX

prom1
10.42.88.1

prom2
10.42.88.2

minikube
10.42.88.17

MicroK8s Cluster

controller
10.42.88.120

worker1
10.42.88.121

worker2
10.42.88.122

Main System

KVM

Backup Systems
Prometheus
already installed

Prometheus
10.0.2.51

Prometheus
10.0.2.78

MicroK8s
10.0.2.88

Introduction to Prometheus

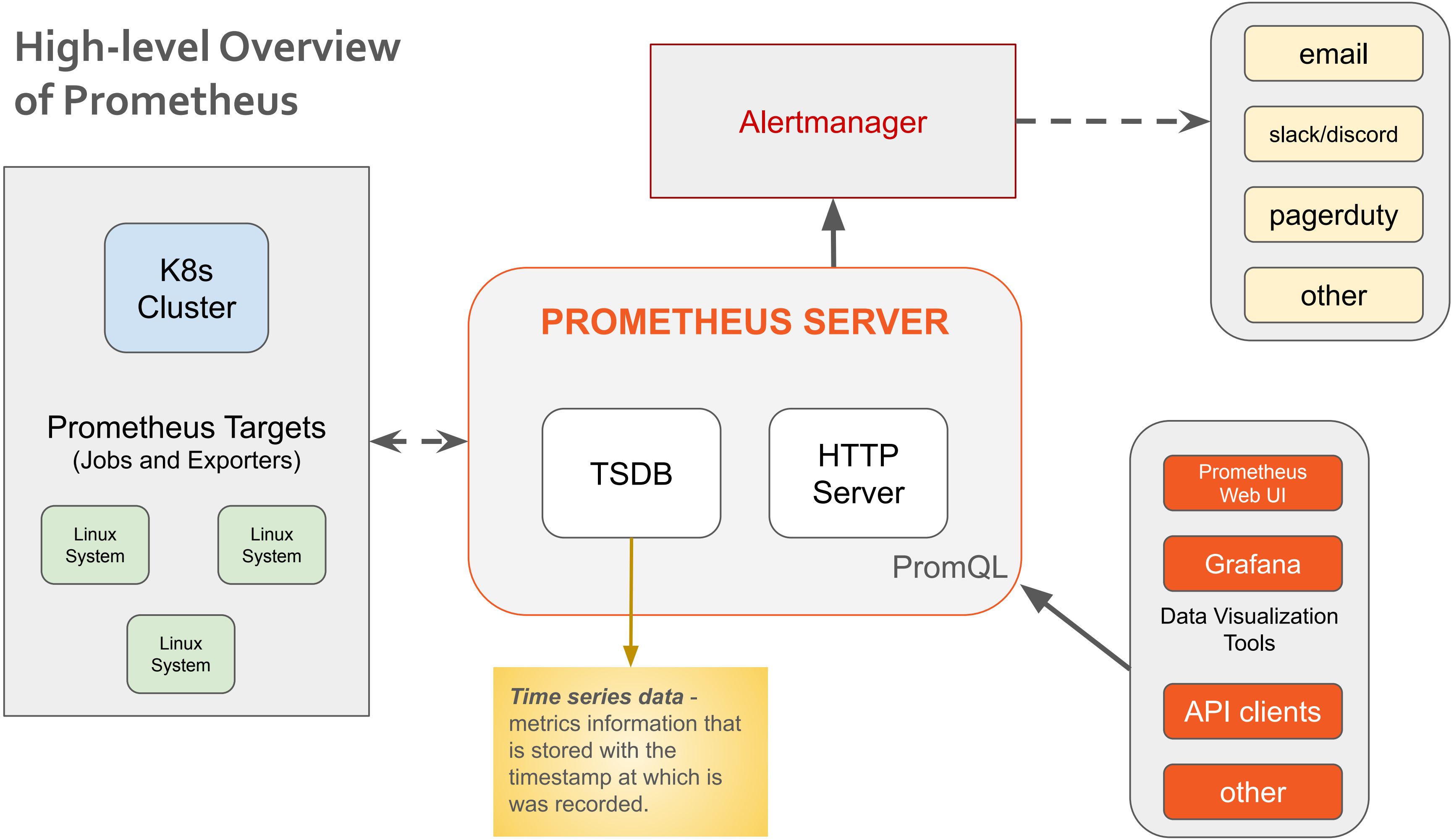
What is Prometheus?

Prometheus is:

- ➢ A monitoring and alerting toolkit.
- ➢ Open source.
- ➢ Over a decade old.
- ➢ Actively developed with over 50K stars on Github.
- ➢ Extensible with tools such as node_exporter and alertmanager.

Quickly discover performance, availability, and security issues!

High-level Overview of Prometheus



Time series data - metrics information that is stored with the timestamp at which is was recorded.

Image based on diagram at https://prometheus.io - © Prometheus Authors 2014-2024

Prometheus Installation

‼ DISCLAIMER and WARNING ‼

The information contained in this webinar is for educational purposes only. The author, O’Reilly, Pearson, or any other entity is not responsible for any misuse of the information.

Practice the concepts provided in this training course on systems that you own, or have authorization to use.

Do not attempt any of the procedures on live work systems or production systems.

Lab 01 - Prometheus Installation

You can install Prometheus in a variety of ways. For example:

- From source
- Package manager
- Docker container
- Scripted install

* **Recommended installation:** Use the provided script!

- Script is located in the repository: labs/lab-01
- Set the script permissions to executable
- Run with sudo
- Ubuntu and Debian are recommended

Or: Ubuntu or Debian package manager

- Ubuntu: `apt install prometheus`
- Debian: `apt install prometheus openipmi-`

Note: CentOS DNF is not recommended. Requires repo config (see lab doc)

Or download the binary from here: https://prometheus.io/download/

Let's check out Lab 01!

Repository: https://github.com/daveprowse/prom-live

Basic Querying

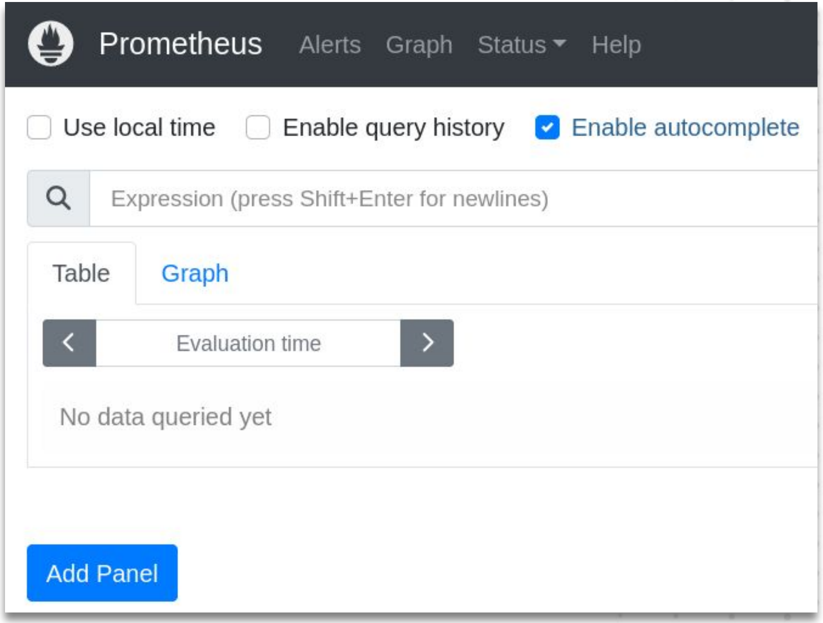
Basic Querying

To run PromQL queries you can use:

- The **Expression Browser** (web UI)
- A dashboard such as Grafana
- Connection to the API

The Expression Browser is helpful for:

- Running basic queries
- Creating PromQL expressions
- Debugging

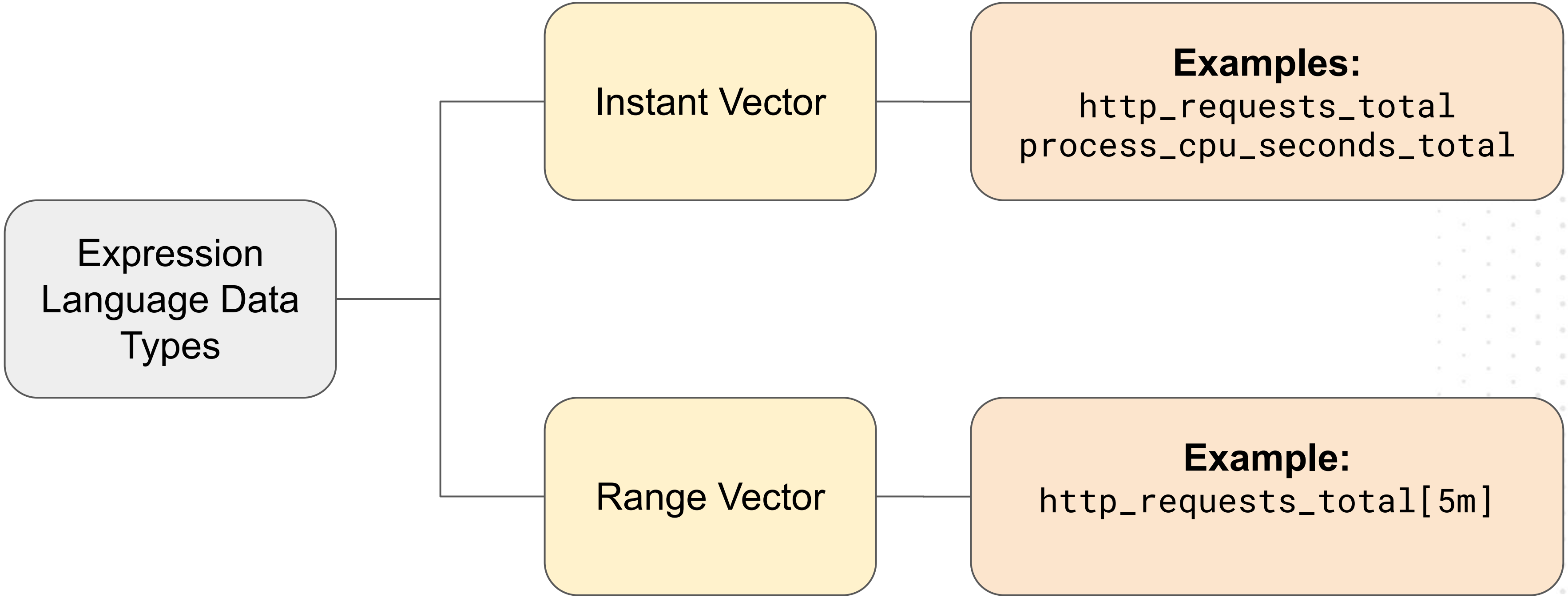


□ Use local time □ Enable query history ☑ Enable autocomplete

Expression (press Shift+Enter for newlines)

No data queried yet

Basic Querying



Expression Language Data Types

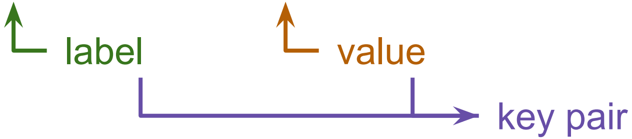
Instant Vector

Examples:
`http_requests_total`
`process_cpu_seconds_total`

Range Vector

Example:
`http_requests_total[5m]`

Example expression: `up{instance="localhost:9090",job="prometheus"}`



Lab 02 - Web UI

In this lab we'll:

- Access the Prometheus expression browser (web UI).
- Examine its components.
- Run some basic queries.

The Web UI can be accessed:

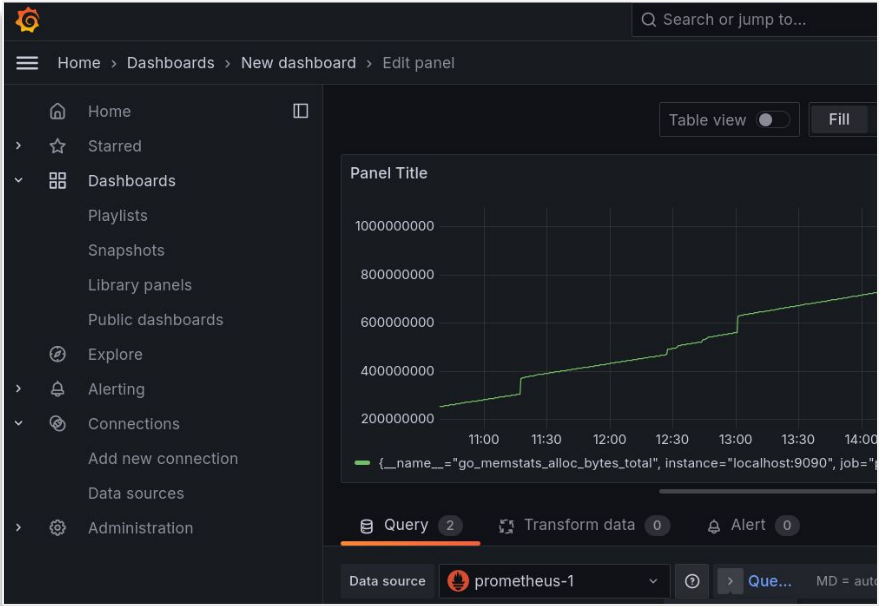
- Locally: `http://localhost:9090`
- Remotely: `http://<ip_address>:9090`
  - *Make sure that port 9090 is open!*

Let's check out Lab 02!

Dashboarding

Dashboarding

- ➢ A dashboard is a visual tool for data analysis, often running in a web browser.
- ➢ Dashboards include:
  - Graphs
  - Counters
  - Other visualizations
  - The ability to easily save your work.
- ➢ Dashboards can be used to:
  - Check the speed and availability of services.
  - Analyze how much traffic is headed inbound and outbound.
  - Easily set up thresholds and alerts.
  - and much more…



Data source prometheus-1

Prometheus

Lab 03 - Dashboarding

In this lab we'll:

- Install Grafana Server.
- Setup a dashboard.
- Run queries and save the dashboard.

* There are scripts available to automate the install:

- Ubuntu/Debian: grafana-install-ubuntu.sh
- CentOS: grafana-install-centos.sh
  - *Set the script to executable and run as sudo*
  - *Make sure that port 3000 is open!*

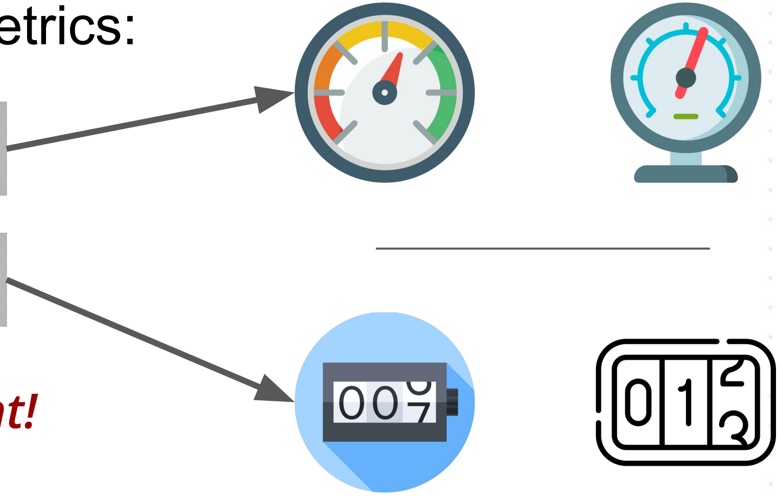
Otherwise, you can install to a variety of platforms:
https://grafana.com/grafana/download/
https://grafana.com/docs/grafana/latest/setup-grafana/installation/

Let's check out Lab 03!

Monitoring Metrics

Monitoring Metrics

- There are two main types of metrics:
  - **Gauges** Absolute Value
  - **Counters** Cumulative



Both are important!

- Others metrics include:
  - Histograms
  - Summaries
  - Aggregations

Lab 04 - node_exporter

In this lab we'll:

- Install the node_exporter to our second virtual machine.
  - Note: It listens on port 9100 by default.
- Modify the prometheus.yml configuration file on the main system.
- Query the remote system from the Prometheus web UI.
- Query the remote system from Grafana.

* There is an automated script for Ubuntu and Debian.

- `labs/lab-04/node-exporter.sh`
- It should work on most other systemd-based Linux systems as well.
- Or, install from package manager (older version)
  - `sudo apt install prometheus-node-exporter`
- Or, download it from:
  - https://prometheus.io/download/
  - https://github.com/prometheus/node_exporter

Let's check out Lab 04!

Instrumenting Code

Instrumenting Code

Basic Definition: **Instrumentation** is the ability to monitor and measure your product’s performance.

Have your own custom application? *Instrument* your code!

- This requires:
  - An application
  - A Prometheus client library

Supported client libraries:

- Go
- Python
- Rust
- Ruby
- Java

*Unofficial list can be found here.

Lab 05 - Instrumenting Code

In this brief lab we'll:

- ➢ Install the Python client library for Prometheus.
- ➢ Examine, (modify), and copy the supplied Python script.
- ➢ Add the web server to the Prometheus configuration.
- ➢ Run the Python script.
- ➢ Scrape metrics!

Let's check out Lab 05!

Monitoring Linux Systems

Lab 06 - Monitoring Linux

In this lab we'll:

- ➢ Install the node_exporter Grafana Dashboard
- ➢ Test against nodes and analyze the results
- ➢ Install the alert_manager
- ➢ Configure and view an alert

Let's check out Lab 06!

Monitoring Kubernetes

Lab 07 - Monitoring Kubernetes

In this lab we'll:

- Install Prometheus and Grafana to a Kubernetes cluster using Helm.
  - I'll also briefly show the minikube install.
- Connect to Prometheus and Start Monitoring.
- Connect to Grafana Dashboards and monitor the cluster.

Note: There will be slight differences for those of you working with a minikube.

Let's check out lab 07!

WRAP UP!

Thank you for attending today!

Questions? Go to:
http://prowse.tech
Or my Discord Server