

Monitoring your services





Somkiat Puisungnoen

Somkiat Puisungnoen

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Intro

Software Craftsmanship

Software Practitioner at สยามชัมนาณกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Somkiat Puisungnoen 15 mins · Bangkok · ⚙️

Java and Bigdata



Facebook somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help ▾

somkiat.cc
@somkiat.cc

Home Posts Videos Photos

Liked Following Share ... + Add a Button



Agenda



Monitoring your services

- Observability of service
- Enabled your springboot project
- Workshop



UI testing with Robotframework

- Pyramid testing
- UI testing with Robotframework
- Installation and configuration
- Workshop



Monitoring and Metric



Observability your services

- Audit logging
- Application metrics
- Distributed tracing
- Health check API
- Exception tracking
- Log aggregation
- Log deployments and changes



Observability your services

Audit logging
Application metrics
Distributed tracing
Health check API
Exception tracking
Log aggregation
Log deployments and changes



Metric in Spring Boot

Spring Boot Actuator for Spring Boot 1.x
MicroMeter for Spring Boot 2.0

<https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready.html>



**[https://github.com/up1/
demo_springboot_moniotring](https://github.com/up1/demo_springboot_moniotring)**



Spring Boot Actuator (1)

Add library to pom.xml

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```



Spring Boot Actuator (2)

Enabled endpoint in application.properties

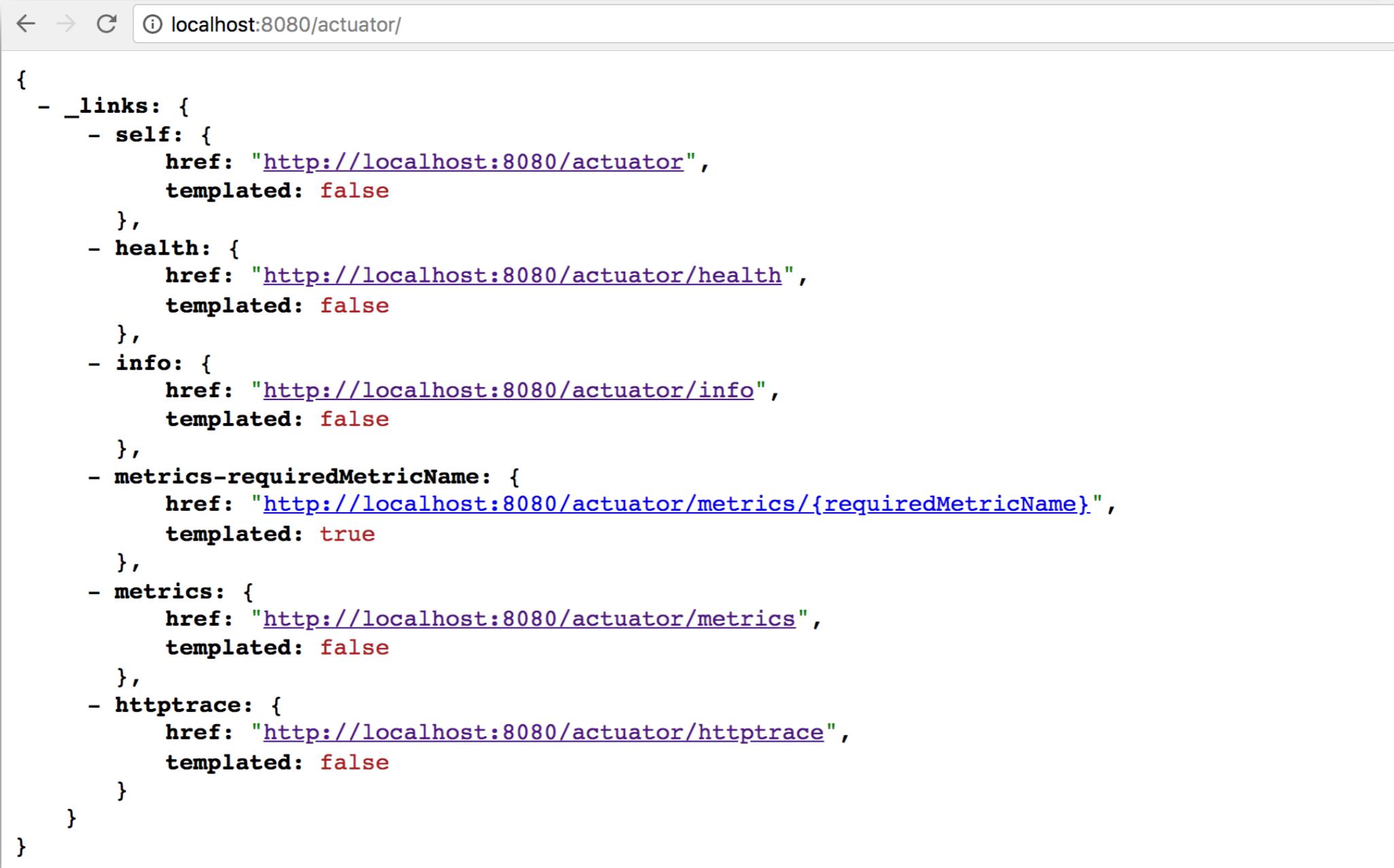
```
info.app.name=Toy Store
info.app.description=This is my first spring boot application
info.app.version=1.0.0

management.endpoints.web.exposure.include=health,info,metrics,httptrace
```



Spring Boot Actuator (3)

List of endpoints = /actuator/



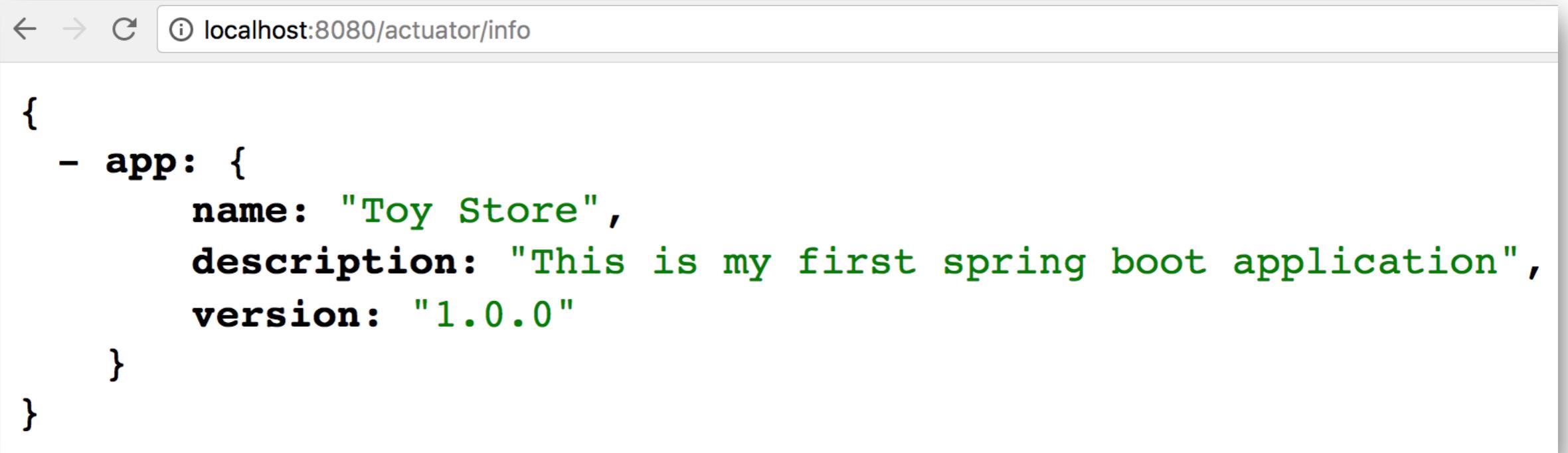
The screenshot shows a browser window with the URL `localhost:8080/actuator/` in the address bar. The page displays a JSON object representing the available endpoints:

```
{  
  - _links: {  
    - self: {  
        href: "http://localhost:8080/actuator",  
        templated: false  
      },  
    - health: {  
        href: "http://localhost:8080/actuator/health",  
        templated: false  
      },  
    - info: {  
        href: "http://localhost:8080/actuator/info",  
        templated: false  
      },  
    - metrics-requiredMetricName: {  
        href: "http://localhost:8080/actuator/metrics/{requiredMetricName}",  
        templated: true  
      },  
    - metrics: {  
        href: "http://localhost:8080/actuator/metrics",  
        templated: false  
      },  
    - httptrace: {  
        href: "http://localhost:8080/actuator/httptrace",  
        templated: false  
      }  
  }  
}
```



Spring Boot Actuator (4)

Info endpoint = /actuator/info



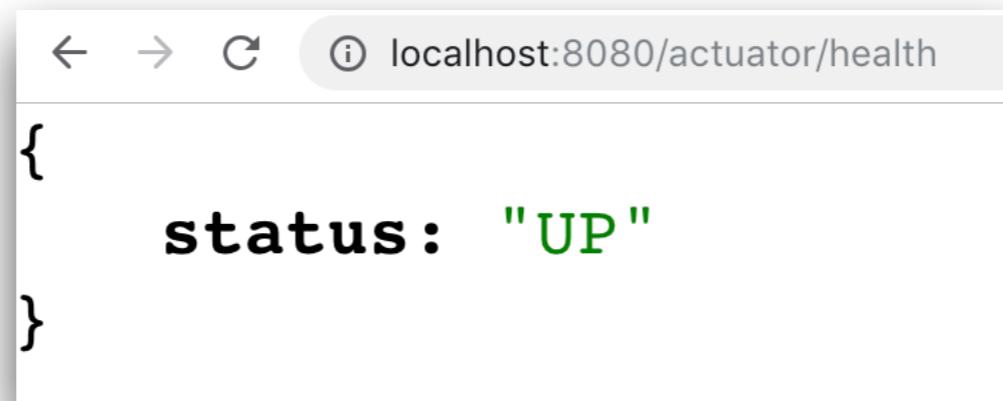
A screenshot of a web browser window displaying the JSON output of the /actuator/info endpoint. The URL in the address bar is 'localhost:8080/actuator/info'. The JSON response is as follows:

```
{  
  - app: {  
      name: "Toy Store",  
      description: "This is my first spring boot application",  
      version: "1.0.0"  
    }  
}
```



Spring Boot Actuator (5)

Info endpoint = /actuator/health



A screenshot of a web browser window displaying the JSON response from the `/actuator/health` endpoint. The URL in the address bar is `localhost:8080/actuator/health`. The response body contains the following JSON object:

```
{  
  status: "UP"  
}
```



Spring Boot Actuator (6)

Info endpoint = /actuator/httptrace

```
← → ⌂ ⓘ localhost:8080/actuator/httptrace

{
  - traces: [
    - {
      timestamp: "2018-03-06T13:33:02.800Z",
      principal: null,
      session: null,
      - request: {
        method: "GET",
        uri: "http://localhost:8080/prometheus",
        - headers: {
          - host: [
            "localhost:8080"
          ],
          - user-agent: [
            "Prometheus/2.0.0"
          ],
          - accept: [
            "text/plain;version=0.0.4;q=1,*/*;q=0.1"
          ],
          - accept-encoding: [
            "gzip"
          ],
          - x-prometheus-scrape-timeout-seconds: [
            "5.000000"
          ]
        },
        remoteAddress: null
      },
    }
  ]
}
```



Spring Boot Actuator (7)

List of metrics endpoint = /actuator/metrics

```
← → ⌂ ⓘ localhost:8080/actuator/metrics

{
  - names: [
    "jvm.buffer.memory.used",
    "jvm.memory.used",
    "jvm.gc.memory.allocated",
    "jvm.memory.committed",
    "http.server.requests",
    "jdbc.connections.min",
    "tomcat.sessions.created",
    "tomcat.sessions.expired",
    "hikaricp.connections.usage",
    "tomcat.global.request.max",
    "tomcat.global.error",
    "jvm.gc.max.data.size",
    "logback.events",
    "system.cpu.count",
    "jvm.memory.max",
    "jdbc.connections.active",
    "jvm.buffer.total.capacity",
    "jvm.buffer.count",
    "process.files.max",
    "jvm.threads.daemon",
```



Spring Boot Actuator (8)

/actuator/metrics/http.server.requests

```
← → ⌂ ⓘ localhost:8080/actuator/metrics/http.server.requests

{
  name: "http.server.requests",
  - measurements: [
    - {
      statistic: "COUNT",
      value: 269
    },
    - {
      statistic: "TOTAL_TIME",
      value: 1.1072010200000002
    },
    - {
      statistic: "MAX",
      value: 0.04373569
    }
  ],
  - availableTags: [
    - {
      tag: "exception",
      - values: [
        "None"
      ]
    },
    - {
      tag: "method",
      - values: [
        "GET"
      ]
    },
  ],
}
```



Keep and Visualize Metric of Spring Boot Services



Sample Architecture

Spring Boot



Prometheus



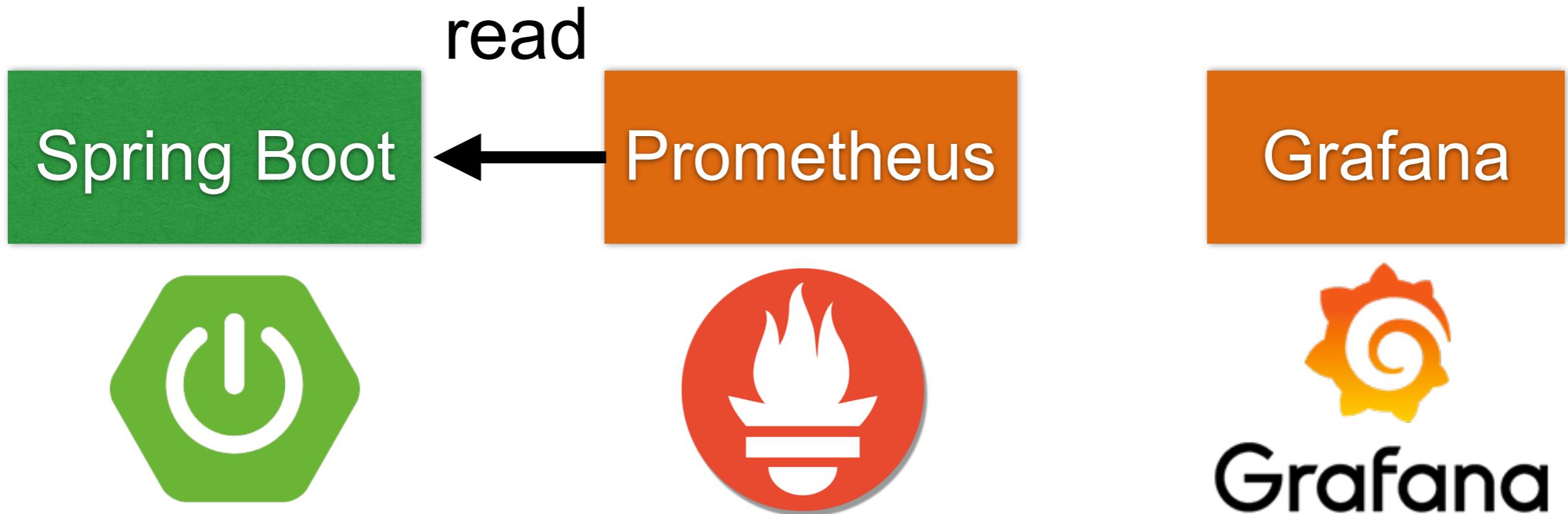
Grafana



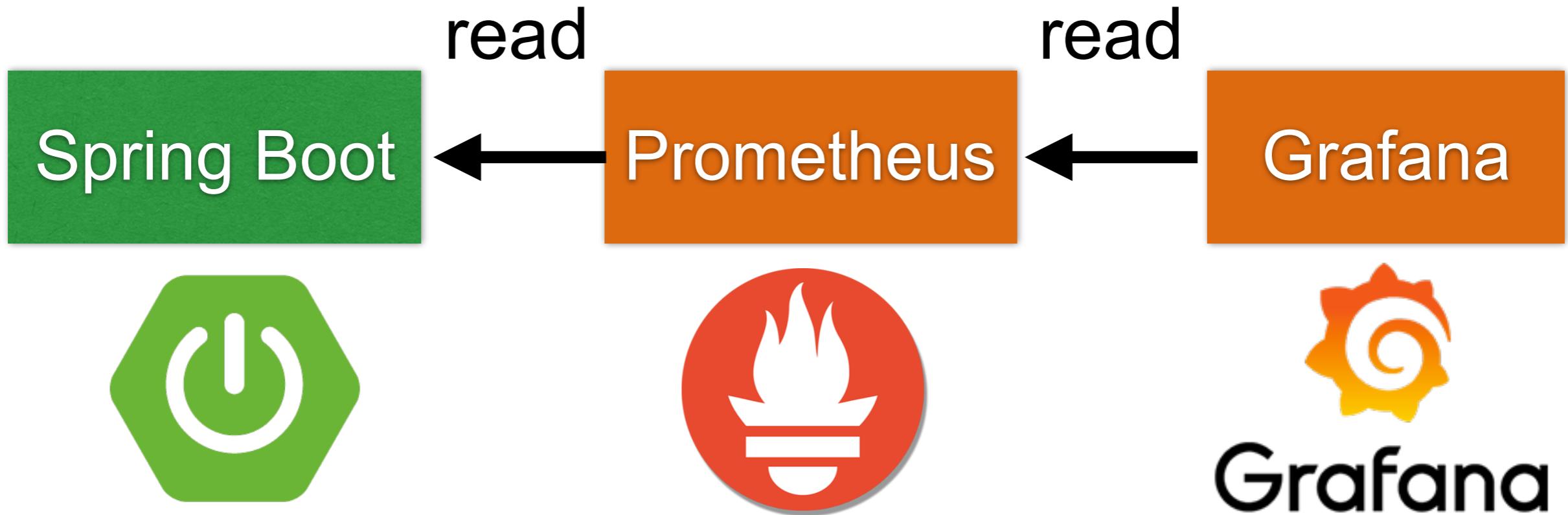
Grafana



Sample Architecture



Sample Architecture



Service metric for Prometheus

<https://prometheus.io/>



Enable Prometheus (1)

Add library to pom.xml

```
<dependency>
    <groupId>io.micrometer</groupId>
    <artifactId>micrometer-registry-prometheus</artifactId>
    <version>1.0.1</version>
</dependency>
```



Enable Prometheus (2)

Enabled endpoint in application.properties

```
management.endpoints.web.exposure.include  
=....,prometheus
```



Enable Prometheus (3)

New endpoint = actuator/prometheus

```
← → ⌂ ⓘ localhost:8080/actuator/prometheus

# HELP jvm_memory_used_bytes The amount of used memory
# TYPE jvm_memory_used_bytes gauge
jvm_memory_used_bytes{area="nonheap",id="Code Cache",} 1.49056E7
jvm_memory_used_bytes{area="nonheap",id="Metaspace",} 5.6766712E7
jvm_memory_used_bytes{area="nonheap",id="Compressed Class Space",} 7617096.0
jvm_memory_used_bytes{area="heap",id="PS Eden Space",} 1.7135864E7
jvm_memory_used_bytes{area="heap",id="PS Survivor Space",} 1.6235192E7
jvm_memory_used_bytes{area="heap",id="PS Old Gen",} 2.1936456E7
# HELP hikaricp_connections_idle Idle connections
# TYPE hikaricp_connections_idle gauge
hikaricp_connections_idle{pool="HikariPool-1",} NaN
# HELP tomcat_threads_config_max
# TYPE tomcat_threads_config_max gauge
tomcat_threads_config_max{name="http-nio-8080",} 200.0
# HELP tomcat_servlet_error_total
# TYPE tomcat_servlet_error_total counter
tomcat_servlet_error_total{name="default",} 0.0
# HELP jvm_threads_peak The peak live thread count since the Java virtual machine start
# TYPE jvm_threads_peak gauge
jvm_threads_peak 28.0
# HELP hikaricp_connections_pending Pending threads
# TYPE hikaricp_connections_pending gauge
hikaricp_connections_pending{pool="HikariPool-1",} NaN
# HELP system_cpu_count The number of processors available to the Java virtual machine
```



Keep data in Prometheus

<https://prometheus.io/>



Prometheus



Prometheus

DOCS

DOWNLOAD

COMMUNITY

BLOG



From metrics to insight

Power your metrics and alerting with a leading
open-source monitoring solution.

GET STARTED

DOWNLOAD

Prometheus v2.0 is available now — [Read the announcement blog post!](#)

<https://prometheus.io/>



Spring Framework
© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

Download

<https://prometheus.io/download/>

prometheus

The Prometheus monitoring system and time series database. [GitHub](#) [prometheus/prometheus](#)

File name	OS	Arch	Size
prometheus-2.9.1.darwin-amd64.tar.gz	darwin	amd64	45.54 MiB
prometheus-2.9.1.linux-amd64.tar.gz	linux	amd64	45.48 MiB
prometheus-2.9.1.windows-amd64.tar.gz	windows	amd64	45.05 MiB



Start service

\$prometheus



Check Data in Prometheus

http://localhost:9090/

The screenshot shows the Prometheus web interface at the URL `http://localhost:9090/graph`. The interface has a dark header bar with navigation icons and the URL. Below the header is a menu bar with links for **Prometheus**, **Alerts**, **Graph**, **Status**, and **Help**. A checkbox labeled **Enable query history** is checked. A text input field for the expression query is present, containing the placeholder `- insert metric at cursor -`. Below the input field are two tabs: **Graph** (which is selected) and **Console**. A table section displays the results of the query, with columns for **Element** and **Value**. The table shows the message `no data`. At the bottom left is a blue button labeled **Add Graph**.



Check Target in Prometheus

Status -> Targets

The screenshot shows the Prometheus web interface with the following details:

- Top Navigation:** Prometheus, Alerts, Graph, Status ▾, Help.
- Main Title:** Targets
- Filter Buttons:** All (selected), Unhealthy.
- Target List:** prometheus (1/1 up) [show less]
- Context Menu (Targets):** Runtime & Build Information, Command-Line Flags, Configuration, Rules, Targets (selected), Service Discovery.
- Table Headers:** Endpoint, State, Labels, Last Scrape, Scrape Duration, Error.
- Table Data:** One row for "http://localhost:9090/metrics":
 - Endpoint: http://localhost:9090/metrics
 - State: UP
 - Labels: instance="localhost:9090" job="prometheus"
 - Last Scrape: 10.477s ago
 - Scrape Duration: 5.835ms
 - Error: (empty)



Add service to Prometheus

Edit file prometheus.yml



Stop and Start service

\$prometheus



Check Target in Prometheus

Status -> Targets

The screenshot shows the Prometheus web interface with the following details:

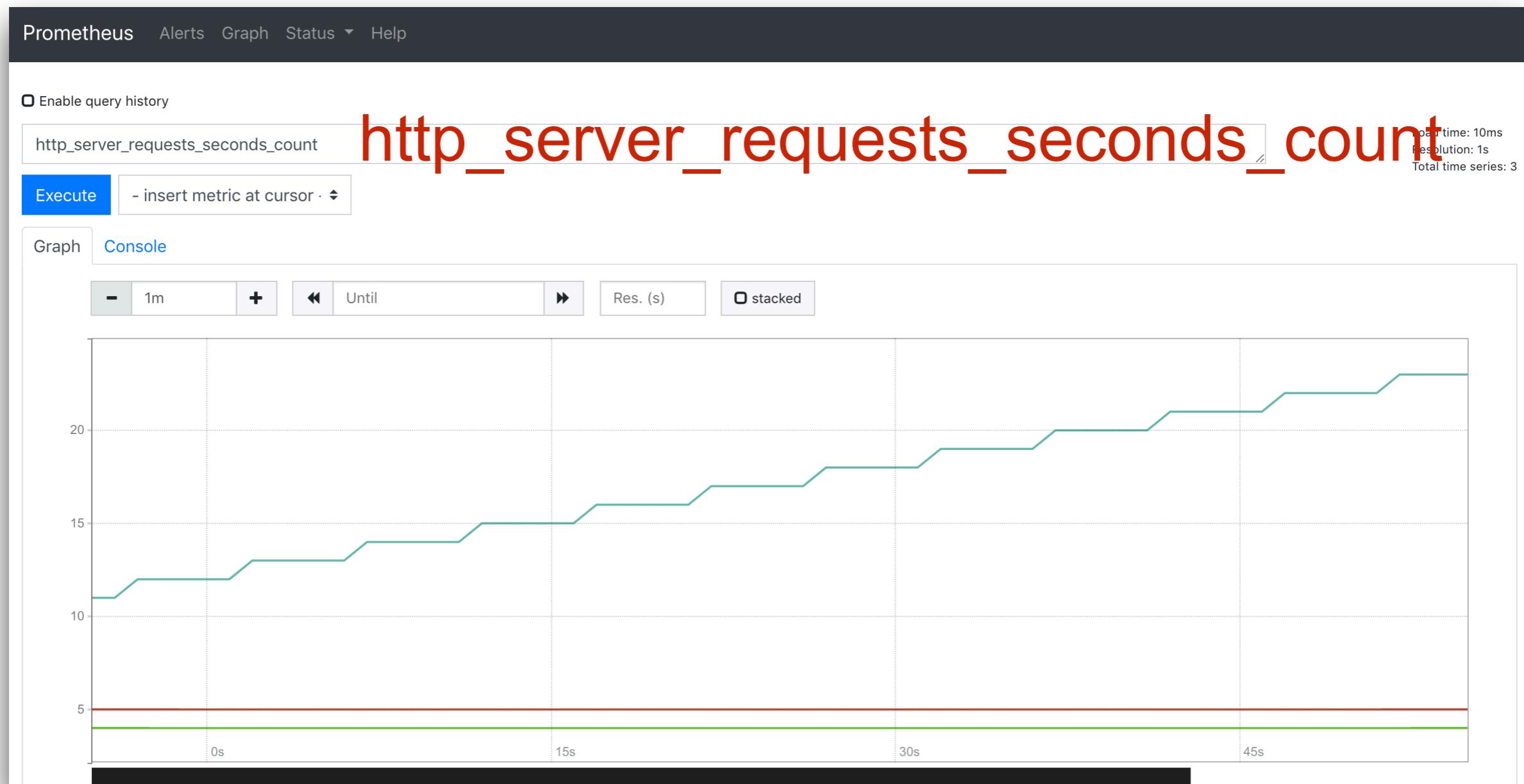
- Top navigation bar: Prometheus, Alerts, Graph, Status ▾, Help.
- Section title: Targets.
- Filter buttons: All (selected), Unhealthy.
- Target list header: demo_service (1/1 up) [show less].
- Table of targets:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8080/actuator/prometheus	UP	instance="localhost:8080" job="demo_service"	1.035s ago	10.87ms	



Check Data in Prometheus

http://localhost:9090/



Show data in Grafana

<https://grafana.com/>



Grafana

The screenshot shows the Grafana Labs website. At the top, there's a navigation bar with links for Plugins (currently installed), Grafana, GrafanaCloud, Services, Dashboards, Plugins, and Get Grafana. Below the navigation, a large banner features the text "The open platform for beautiful analytics and monitoring". The banner also includes icons for Grafana TestData, Kentik Connect Pro, NS1 for Grafana, and Kubernetes. A prominent orange button labeled "Get Grafana" is visible. The bottom of the page features the Grafana logo.

<https://grafana.com/>



Download

<https://grafana.com/get>

Download Grafana or let us, **the creators** host it



You Run It

[Download](#)

- 100% Open Source.
- Runs on all major operating systems.
- Can be paired with any [support plan](#).



We Host It

[Get your **free** instance now](#)

- Free for **1** User, up to **5** dashboards.
- No installation required, instant deploy.
- Can scale up to thousands of users.
[Learn more about Hosted Grafana](#).



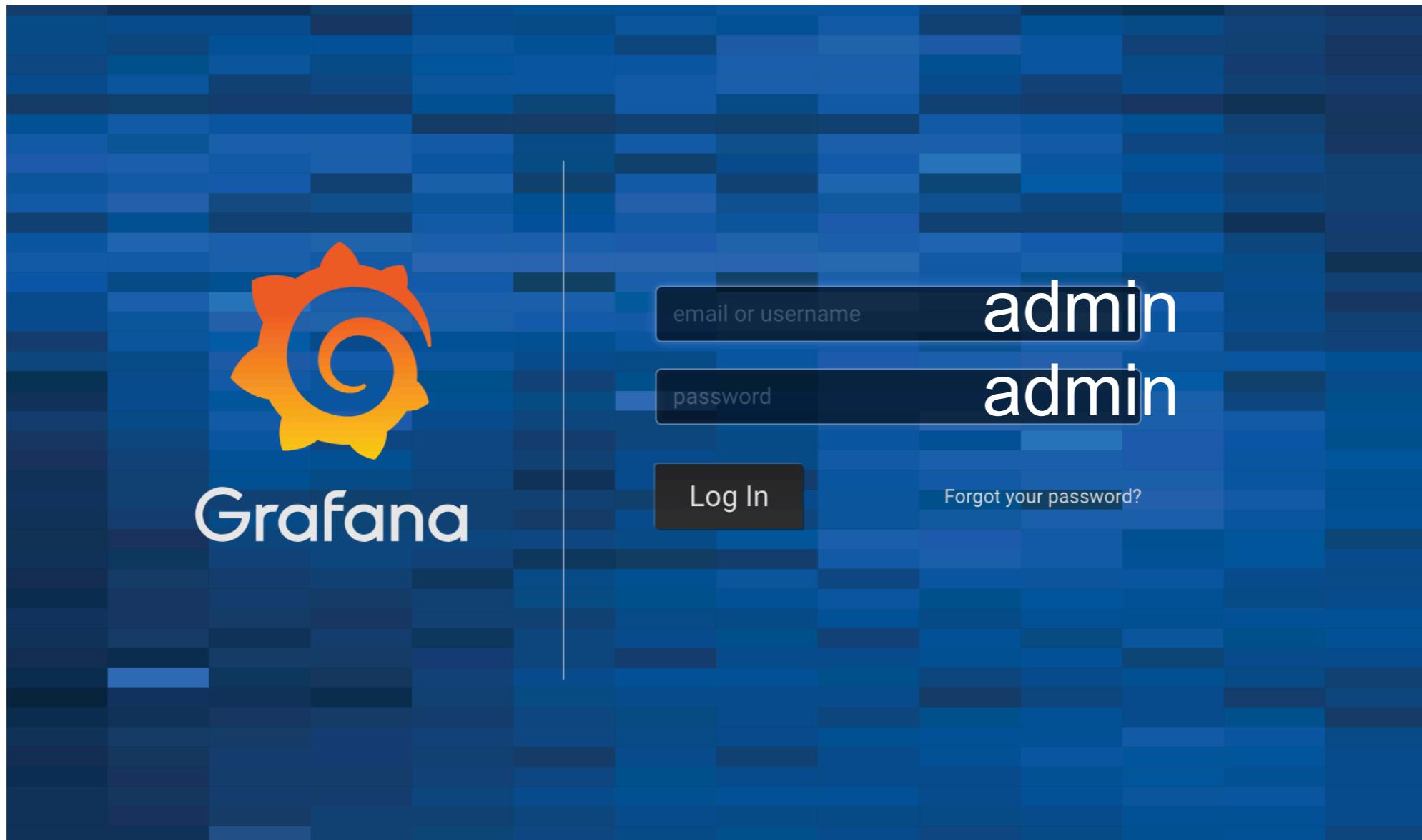
Start service

`$./bin/grafana-server`



Check Grafana

<http://localhost:3000/>



Configuration

Home ▾ ⚙️ 💻

Home Dashboard

The dashboard features a top navigation bar with icons for Home, Settings, and Help. Below the title, there's a horizontal bar with five items: "Install Grafana" (green checkmark icon), "Add data source" (orange database icon, highlighted in green), "Create your first dashboard" (grey grid icon), "Invite your team" (grey user icon), and "Install apps & plugins" (grey sunburst icon). The main area contains sections for "Starred dashboards" and "Recently viewed dashboards" (both empty), and a sidebar with sections for "Installed Apps" (empty), "Installed Panels" (empty), and "Installed Datasources" (empty).

Install Grafana Add data source Create your first dashboard Invite your team Install apps & plugins

Starred dashboards

Recently viewed dashboards

Installed Apps

None installed. [Browse Grafana.com](#)

Installed Panels

None installed. [Browse Grafana.com](#)

Installed Datasources

None installed. [Browse Grafana.com](#)



Add data source

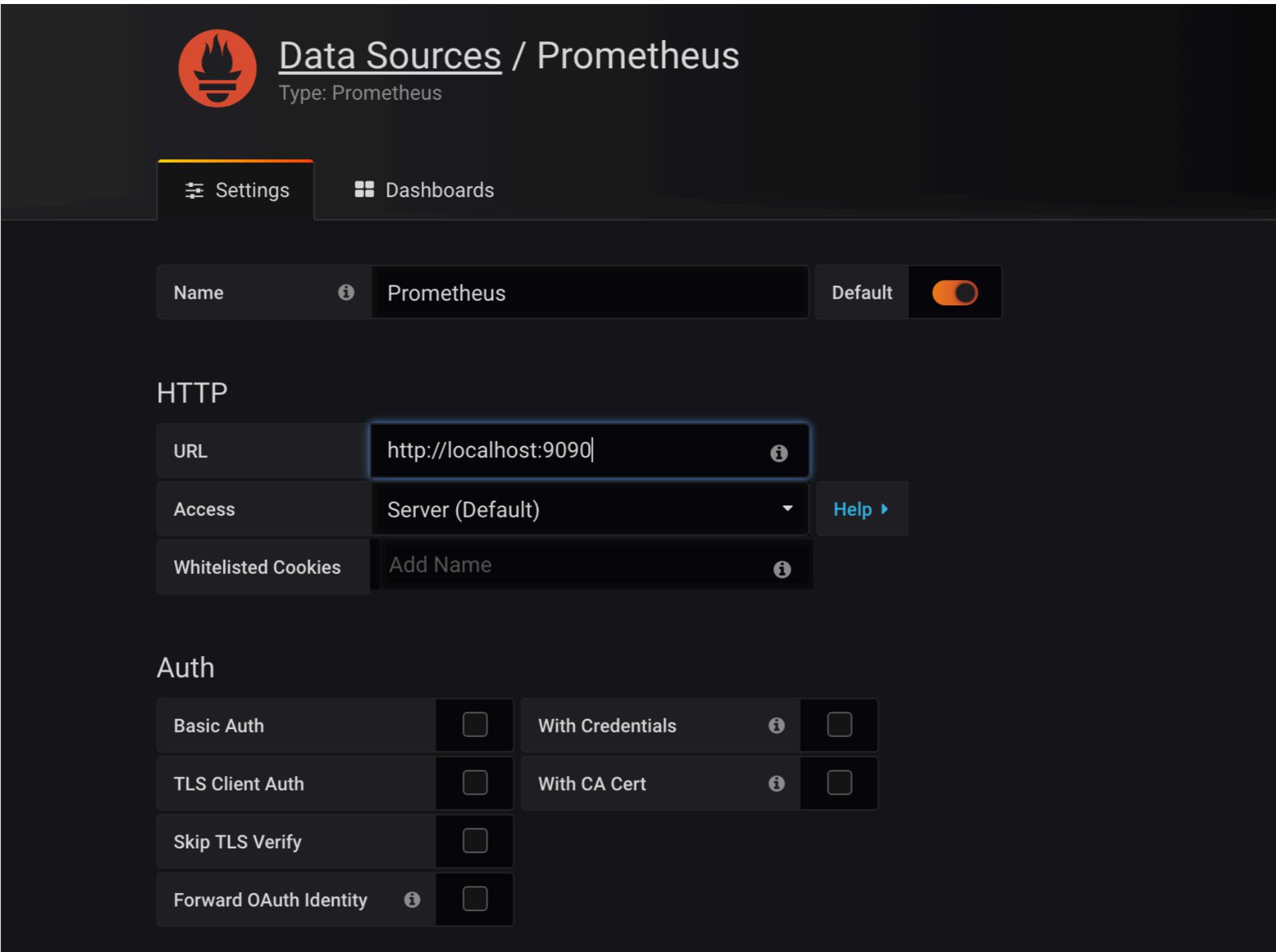
The screenshot shows a dark-themed user interface for managing data sources. At the top left is a gear icon labeled "Configuration" and "Organization: Main Org.". Below the header is a navigation bar with tabs: "Data Sources" (highlighted in orange), "Users", "Teams", "Plugins", "Preferences", and "API Keys". The main content area is titled "Choose data source type" and features a search bar with the placeholder "Filter by name or type". A 3x3 grid of data source icons and names is displayed:

Azure Monitor	CloudWatch	Elasticsearch
Graphite	InfluxDB	Loki
Microsoft SQL Server	MySQL	OpenTSDB
PostgreSQL	Prometheus	Stackdriver

The "Prometheus" card is highlighted with a white border.



Configuration prometheus



The screenshot shows the 'Data Sources / Prometheus' configuration page in Grafana. The top navigation bar includes a red Prometheus icon, the title 'Data Sources / Prometheus', and the subtitle 'Type: Prometheus'. Below the navigation are two tabs: 'Settings' (selected) and 'Dashboards'. A search bar at the top right contains the text 'Prometheus'. To the right of the search bar is a 'Default' toggle switch which is turned on.

HTTP

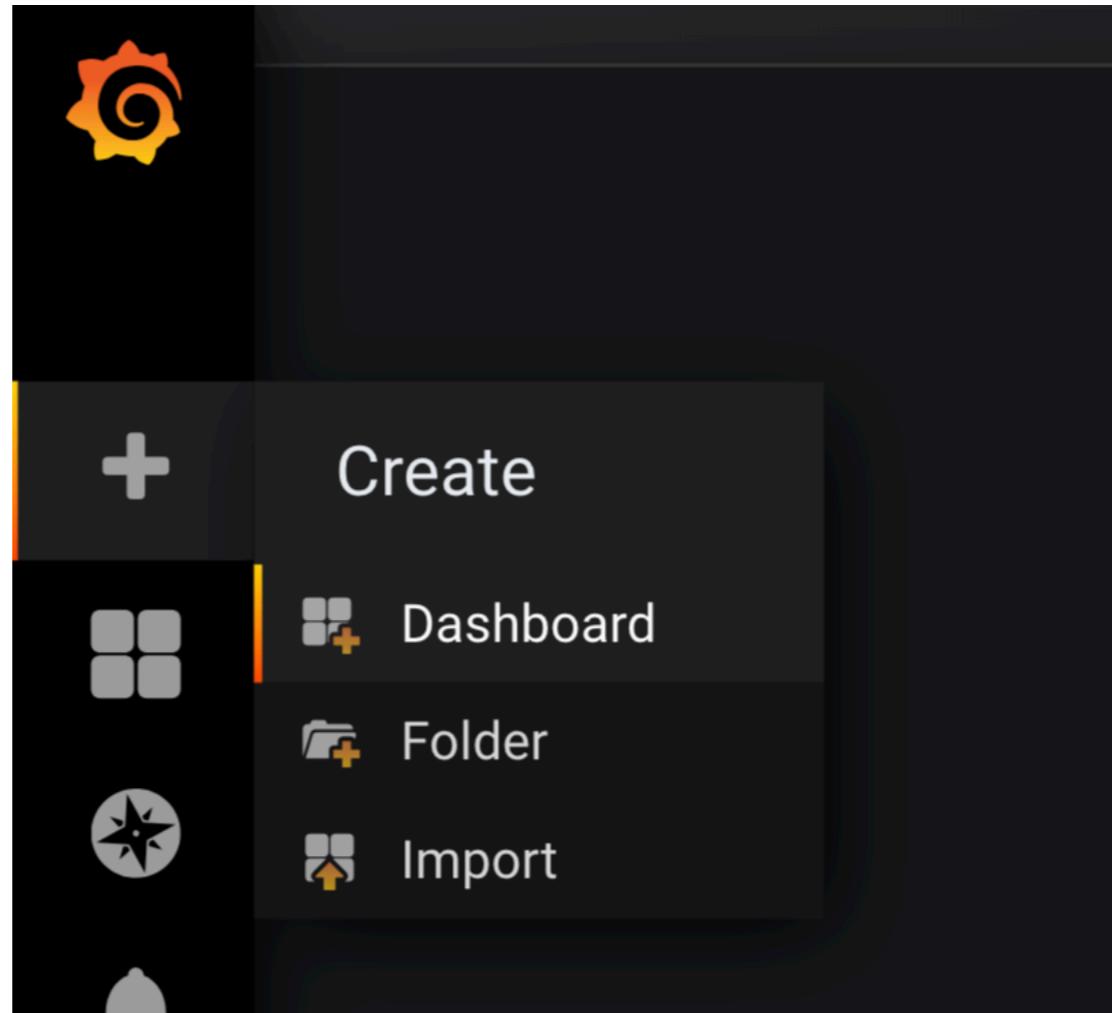
URL	http://localhost:9090	Help
Access	Server (Default)	Help
Whitelisted Cookies	Add Name	Help

Auth

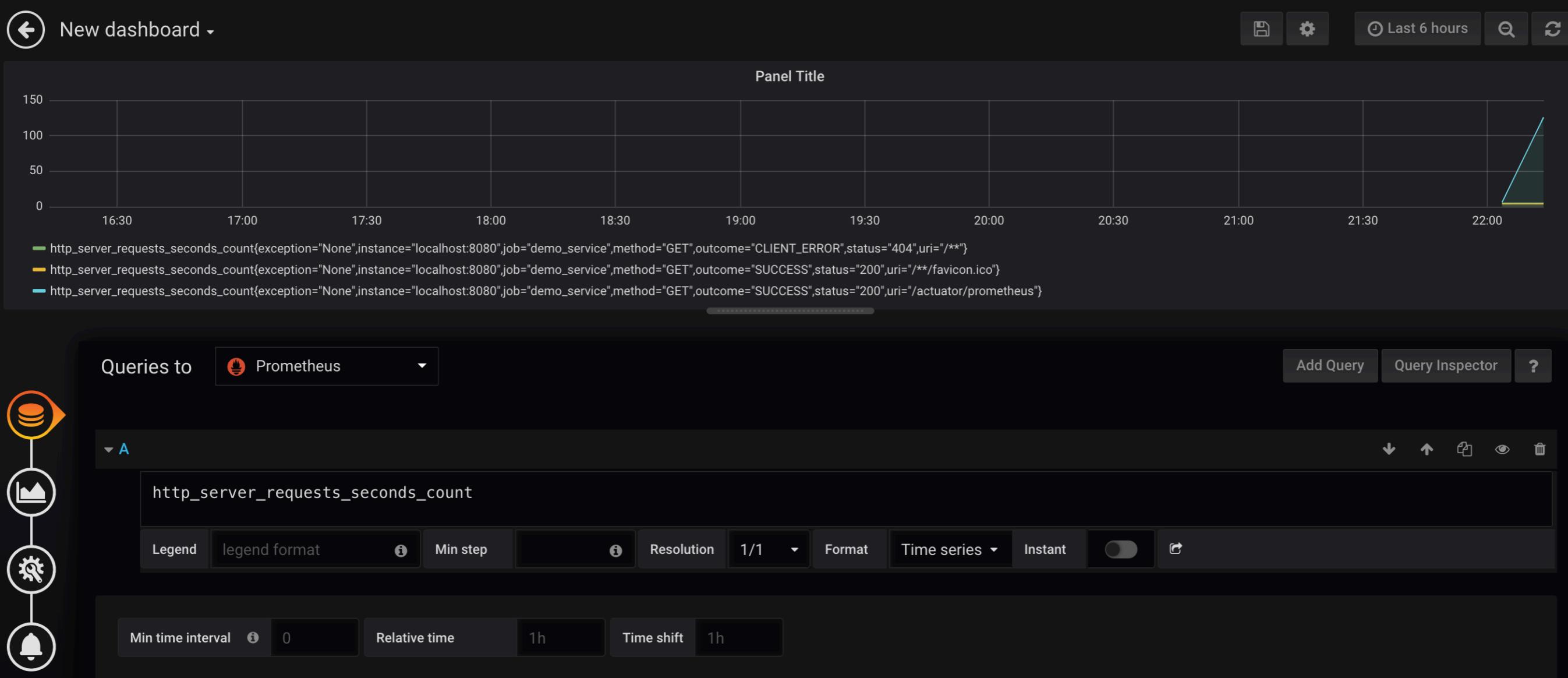
Basic Auth	<input type="checkbox"/>	With Credentials	<input type="checkbox"/>
TLS Client Auth	<input type="checkbox"/>	With CA Cert	<input type="checkbox"/>
Skip TLS Verify	<input type="checkbox"/>		
Forward OAuth Identity	Help	<input type="checkbox"/>	



Create dashboard of service



Create dashboard of service



Take to your home

Always improve, always practice



UI testing with Robotframework



<https://robotframework.org/>



Robotframework

The screenshot shows the official website for Robot Framework. The header features a large teal background with the text "ROBOT FRAMEWORK" in white and black. To the left is a black sidebar with a white icon of a robot head and a list of links: INTRODUCTION, EXAMPLES, LIBRARIES, TOOLS, DOCUMENTATION, SUPPORT, RPA, FOUNDATION, SHOP, ROBOCON, and USERS.

INTRODUCTION

Robot Framework is a generic open source automation framework for acceptance testing, acceptance test driven development (ATDD), and robotic process automation (RPA). It has easy-to-use tabular test data syntax and it utilizes the keyword-driven testing approach. Its testing capabilities can be extended by test libraries implemented either with Python or Java, and users can create new higher-level keywords from existing ones using the same syntax that is used for creating test cases.

Robot Framework project is hosted on [GitHub](#)

Tweets by @robotframework

Robot Framework Retweeted
Tatu Aalto @AaltoTatu
@robotframework SeleniumLibrary 4.0.0 alpha 1 is out: [github.com/robotframework.../Selenium](https://github.com/robotframework/SeleniumLibrary)

robotframework/Seleni...
Web testing library for Ro...
[github.com](https://github.com/robotframework/SeleniumLibrary)

Apr 19, 2019

<https://robotframework.org/>



Robotframework

Python
Robotframework
Browser driver



Install python

<https://www.python.org/>

The screenshot shows the Python.org homepage with a dark blue header featuring the Python logo and the word "python™". A navigation bar below has links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area is titled "Download the latest version for Mac OS X" and includes a yellow "Download Python 3.7.3" button. It also provides links for other operating systems and developer resources. To the right is a graphic of two boxes descending from the sky on parachutes. Below the main content is a section for specific releases and a footer with a logo and copyright information.

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for Mac OS X

[Download Python 3.7.3](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases

Looking for a specific release?
Python releases by version number:



Install robotframework

```
$pip install robotframework
```

```
$robot
```

```
[ ERROR ] Expected at least 1 argument, got 0.
```

```
Try --help for usage information.
```



Browser testing

Using selenium project

The screenshot shows the SeleniumHQ website homepage. The header features the SeleniumHQ logo (a stylized 'Se' with a checkmark) and the text 'SeleniumHQ Browser Automation'. It includes a search bar with 'search selenium:' and a 'Go' button, along with navigation links for 'edit this page', 'Projects', 'Download', 'Documentation', 'Support', and 'About'.

What is Selenium?

Selenium automates browsers. That's it! What you do with that power is entirely up to you. Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should!) be automated as well.

Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks.

Which part of Selenium is appropriate for me?

Selenium WebDriver 	Selenium IDE
-------------------------------	-------------------------

If you want to

- create robust, browser-based regression automation suites and tests
- scale and distribute scripts across many environments

If you want to

- create quick bug reproduction scripts
- create scripts to aid in automation-aided

Selenium is a suite of tools to automate web browsers across many platforms.

Selenium...

- runs in [many browsers](#) and [operating systems](#)
- can be controlled by many [programming languages](#) and [testing frameworks](#).

[Download Selenium](#)

<https://www.seleniumhq.org/>



Install selenium library

```
$pip install robotframework-seleniumlibrary
```

<https://github.com/robotframework/SeleniumLibrary/>



Workshop

Create test case

Run test case

Result



Create file demo.robot

```
demo.robot ×  
1 *** Settings ***  
2 Library SeleniumLibrary  
3  
4 *** Variables ***  
5  
6 *** Test Cases ***  
7  
8 *** Keywords ***  
9
```



Run test

\$robot demo.robot

```
[ ERROR ] Suite 'Demo' contains no tests.
```

```
Try --help for usage information.
```



Workshop

<https://www.it.kmitl.ac.th/th/>

