

Performance Monitoring

Thanh-Phuong Pham

Motivation

Applications & Systems

“Where does our data come from?”

Prometheus

- ◆ Monitoring system
- ◆ More than a time series database
- ◆ Supporting the collection of application and system data

Open Source

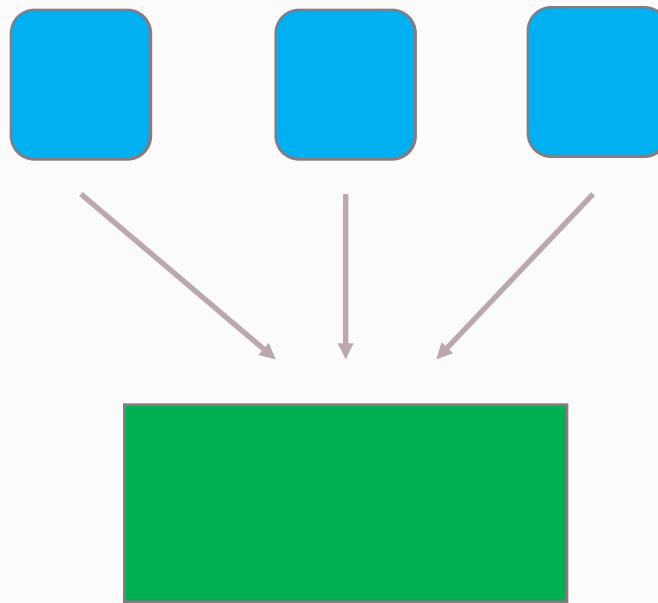
Major Components

- ◆ Exporter
- ◆ Pushgateway
- ◆ Client libraries
 - Go, Java, Python, and Ruby.
 - Third-party for C#/.Net, Node.js, Haskell, Erlang and Rust

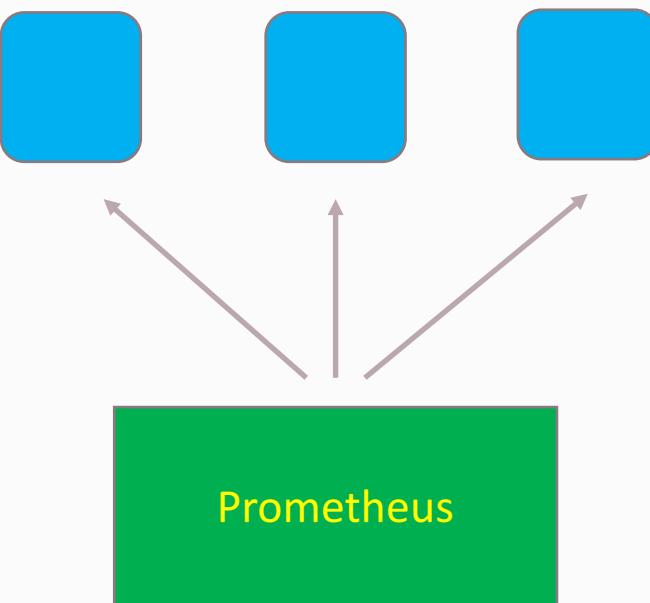
More ...

Operation

Push



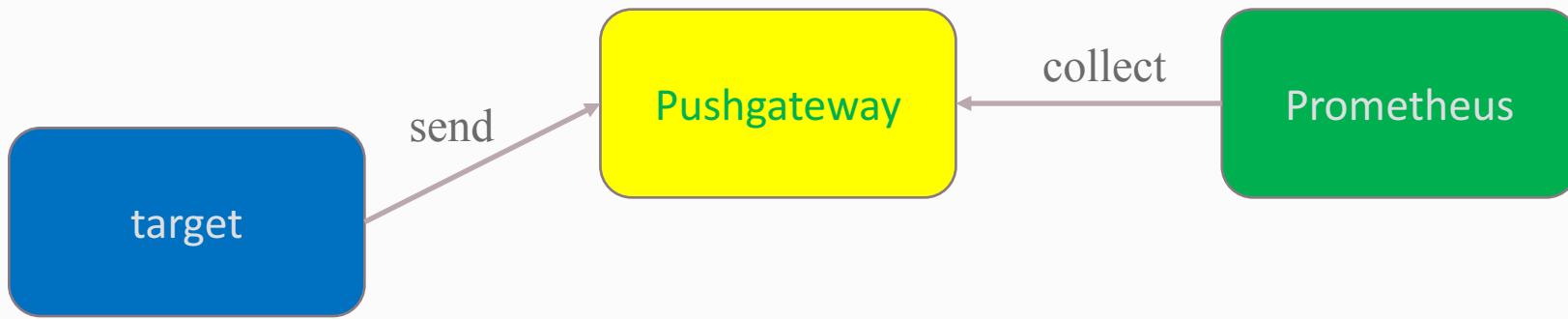
Pull



Exporter



Pushgateway



Instrumentation

Client libraries: Java, Go, Python, Ruby.

Third-party libraries: C#/.Net, Node.js, Erlange, etc

Python

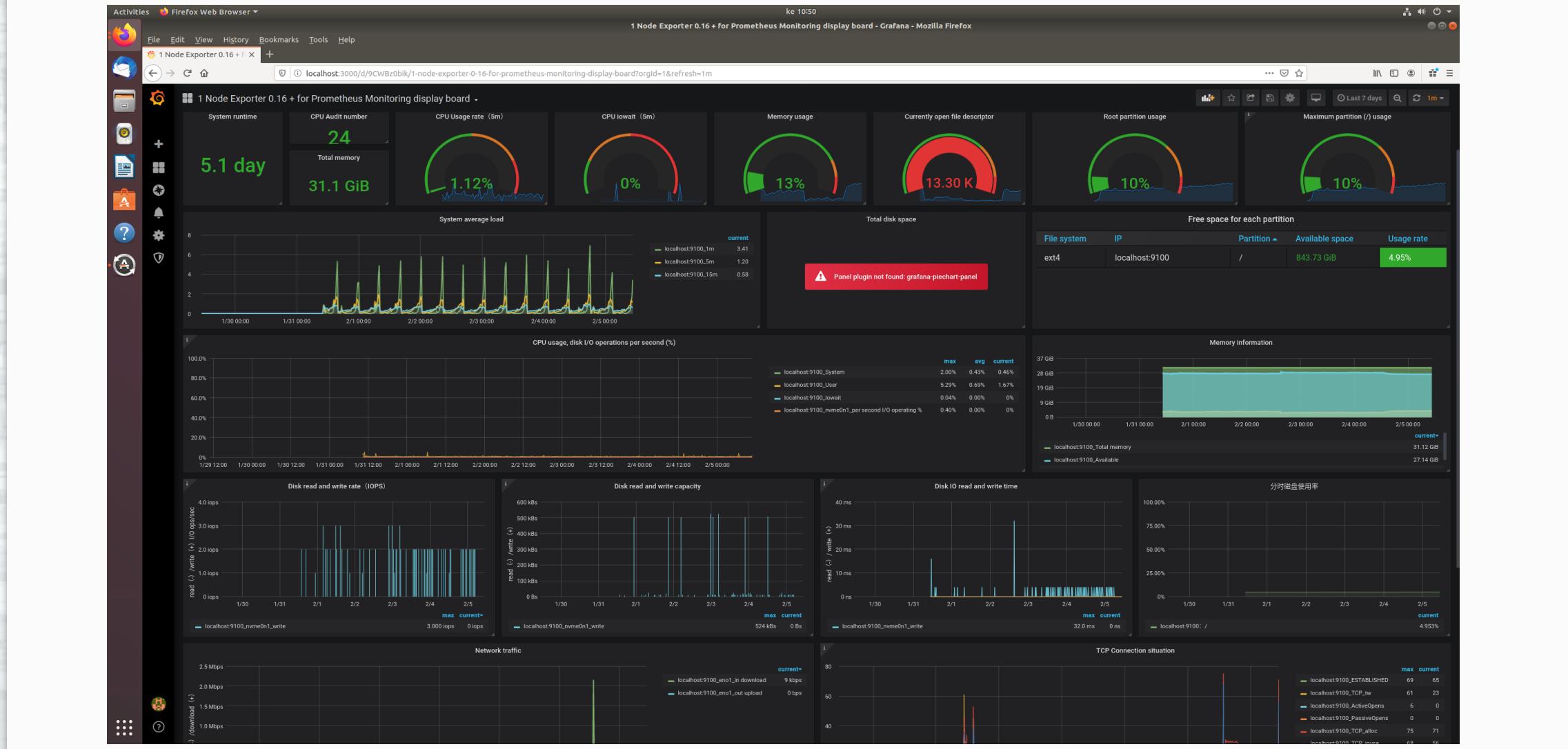
```
import http.server
from prometheus_client import start_http_server
from prometheus_client import Counter

REQUESTS = Counter('hello_worlds_total', 'Hello Worlds requested.')

class MyHandler(http.server.BaseHTTPRequestHandler):
    def do_GET(self):
        REQUESTS.inc()
        self.send_response(200)
        self.end_headers()
        self.wfile.write(b"Hello World")

if __name__ == "__main__":
    start_http_server(8000)
    server = http.server.HTTPServer(('localhost', 8001), MyHandler)
    server.serve_forever()
```

Visualization



```
port http.server
om prometheus_client import start_http_server
om prometheus_client import Counter

QUESTS = Counter('hello_worlds_total', 'Hello Worlds requested.')

class MyHandler(http.server.BaseHTTPRequestHandler):
    def do_GET(self):
        REQUESTS.inc()
        self.send_response(200)
        self.end_headers()
        self.wfile.write(b"Hello World")

__name__ == "__main__":
start_http_server(8000)
server = http.server.HTTPServer(('localhost', 8001), MyHandler)
server.serve_forever()
```

Your Project

Enjoy monitoring your applications