



# JMeter性能监控平台 与 自动化压测

主讲：老钱

霍格沃兹测试学院



## 涉及到的Docker知识点

- ✓ 事先安装 Docker
  - ❖ Docker安装参考文档：  
[https://yeasy.gitbooks.io/docker\\_practice/content/install/](https://yeasy.gitbooks.io/docker_practice/content/install/)
  
- ✓ Docker的基础使用方法
  - ❖ 获取镜像：  
[https://yeasy.gitbooks.io/docker\\_practice/content/image/pull.html](https://yeasy.gitbooks.io/docker_practice/content/image/pull.html)
  
  - ❖ 启动容器：  
[https://yeasy.gitbooks.io/docker\\_practice/content/container/run.html](https://yeasy.gitbooks.io/docker_practice/content/container/run.html)
  
  - ❖ 进入容器：  
[https://yeasy.gitbooks.io/docker\\_practice/content/container/attach\\_exec.html](https://yeasy.gitbooks.io/docker_practice/content/container/attach_exec.html)



# Agenda

- 1 部署性能监控平台
- 2 运行与结果展示
- 3 自动化压力测试



分享课题：搭建JMeter性能监控平台

# 部署性能监控平台

# 01



# 测试报告 .vs. 性能监控

- ✓ JMeter原生测试测试报告带来的“痛苦”
  - ❖ 不具备实时性
  - ❖ 报告中的数据是测试时间段内的平均值
  - ❖ 长相问题
  
- ✓ 性能监控平台的优势
  - ❖ 实时展示JMeter压测数据
  - ❖ 数据范围可选
  - ❖ 界面更友好





## 性能监控平台组成

- ✓ JMeter：压测工具，产生压测数据
- ✓ InfluxDB: 开源时序数据库，特别适合用于处理和分析资源监控数据，用于存储压测数据
- ✓ Grafana：度量分析与可视化图标展示工具，可以支持不用种类的数据源，用于将存储于InfluxDB中的数据以图表的形式展示出来



## 性能监控平台部署 - InfluxDB

- ✓ 部署方法： Docker 部署
- ✓ 事先准备：
  - ✓ 安装Docker
  - ✓ 下载influxDB镜像：`docker pull influxdb`



## 性能监控平台部署 - InfluxDB

- ✓ 启动 influxDB 容器
- ✓ 将新 influxDB 容器命名为 jmeterdb, 执行docker创建容易命令:

```
docker run -d -p 8086:8086 -p 8083:8083 --name=jmeterdb influxdb
```

- ✓ 执行如下命令进入容器内部

```
docker exec -it jmeterdb bash
```



## 性能监控平台部署 - InfluxDB

- ✓ 在容器内部创建jmeter数据库，用于收集JMeter发送的压测数据
- ✓ 执行 influx命令进入命令台
- ✓ 执行类sql语句 创建jmeter数据库：create database jmeter;
- ✓ 创建完jmeter数据库之后执行如下命令验证数据库已经创建成功：  
show databases;

```
[> show databases;
name: databases
name
----
_internal
jmeter
>
```



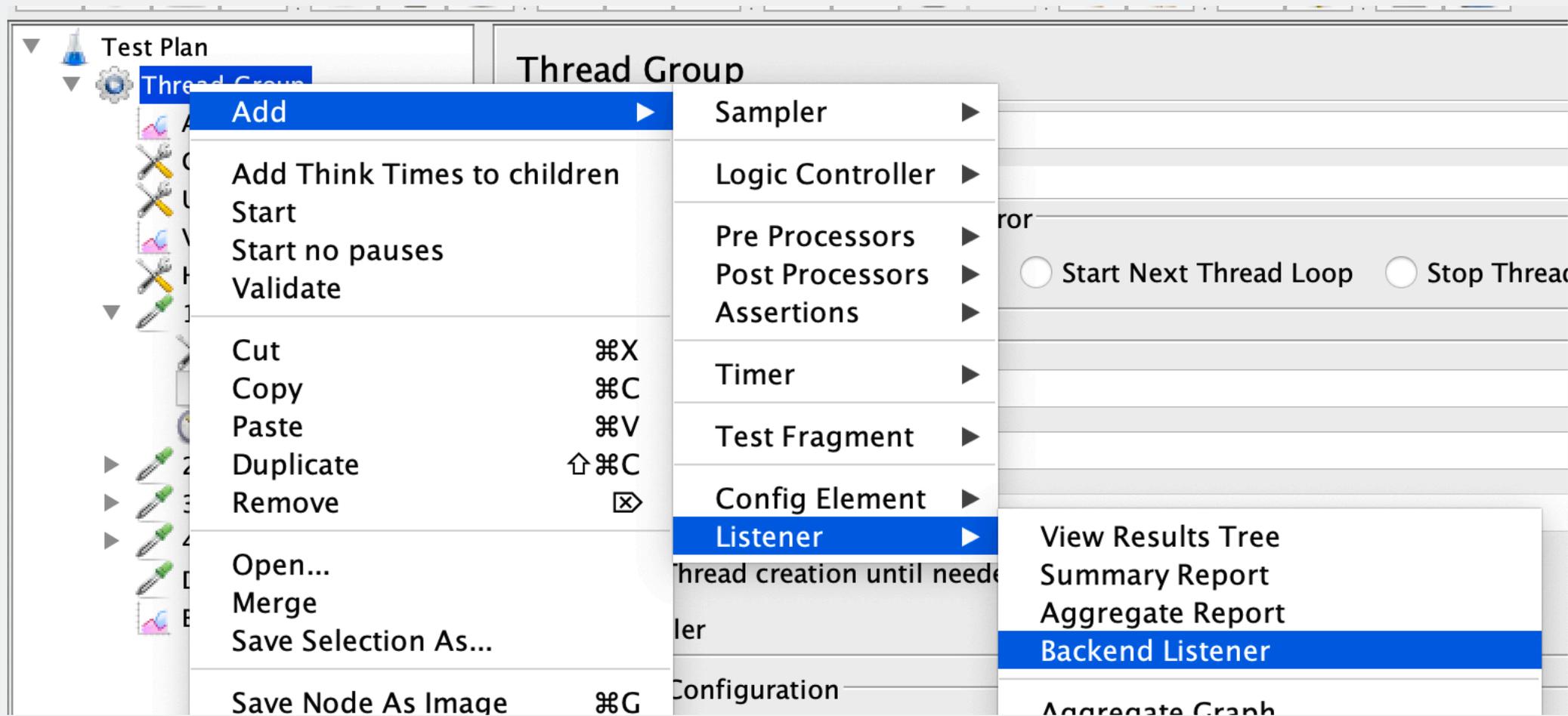
# 性能监控平台部署 - InfluxDB

- ✓ **Demo – 实操创建 InfluxDB**
- ✓ **Docker 常用基础命令小汇总**
  - ✓ 列出已下载docker镜像： `docker images`
  - ✓ 列出运行中的docker 容器： `docker ps`
  - ✓ 列出运行中+已停止docker 容器： `docker ps -a`
  - ✓ 停止容器： `docker stop [docker container id]`
  - ✓ 启动容器： `docker start [docker container id]`
  - ✓ 删除容器： `docker rm [docker container id]`



# 性能监控平台部署 - JMeter

- ✓ 在JMeter中添加 Backend Listener 组件，用于收集数据并发送给 influxdb





## 性能监控平台部署 - JMeter

- ✓ 在Backend Listener implementation 中选择 InfluxdbBackendListenerClient （注意：必须使用jmeter 5.0以上版本，否则需要单独装插件才可以看到这个组件）
- ✓ 在influxdbURL中将实际的influxdb hostname填写进去
- ✓ 在application中填写 order
- ✓ 在testTitle中填写 Order Testing
- ✓ 其余配置保持不变



# 性能监控平台部署 - JMeter

## ✓ 配置参考图

### Backend Listener

Name: Backend Listener

Comments:

Backend Listener implementation `org.apache.jmeter.visualizers.backend.influxdb.InfluxdbBackendListenerClient`

Async Queue size `5000`

#### Parameters

Name:	Value
<code>influxdbMetricsSender</code>	<code>org.apache.jmeter.visualizers.backend.influxdb.HttpMetricsSender</code>
<code>influxdbUrl</code>	<code>http://localhost:8086/write?db=jmeter</code>
<code>application</code>	<code>order</code>
<code>measurement</code>	<code>jmeter</code>
<code>summaryOnly</code>	<code>true</code>
<code>samplersRegex</code>	<code>.*</code>
<code>percentiles</code>	<code>90;95;99</code>
<code>testTitle</code>	<code>Order Testing</code>
<code>eventTags</code>	



# 性能监控平台部署 - JMeter

- ✓ 运行JMeter, 然后在 influxdb中检验是否已经能够接受到数据
- ✓ 在influxdb 命令台中使用查询语句, 检查是否已经能够收到数据

```
> use jmeter
Using database jmeter
> select * from jmeter
name: jmeter
time                application avg          count countError endedT hit max  maxAT meanAT min minAT pct90.0
  rb      sb      startedT statut transaction  -----
-----
--      --      -----
1565506448941000000 order          internal          10          0          0          0
      10
1565506453946000000 order          244.96          25          0          25 1216          1          1214
      26225 8630      all      all
1565506453946000000 order          internal          0          0          10          10          10
      10
```



## 性能监控平台部署 - JMeter

- ✓ **Demo – 实操创建 JMeter BackEnd Listener**



# 性能监控平台部署 - Grafana

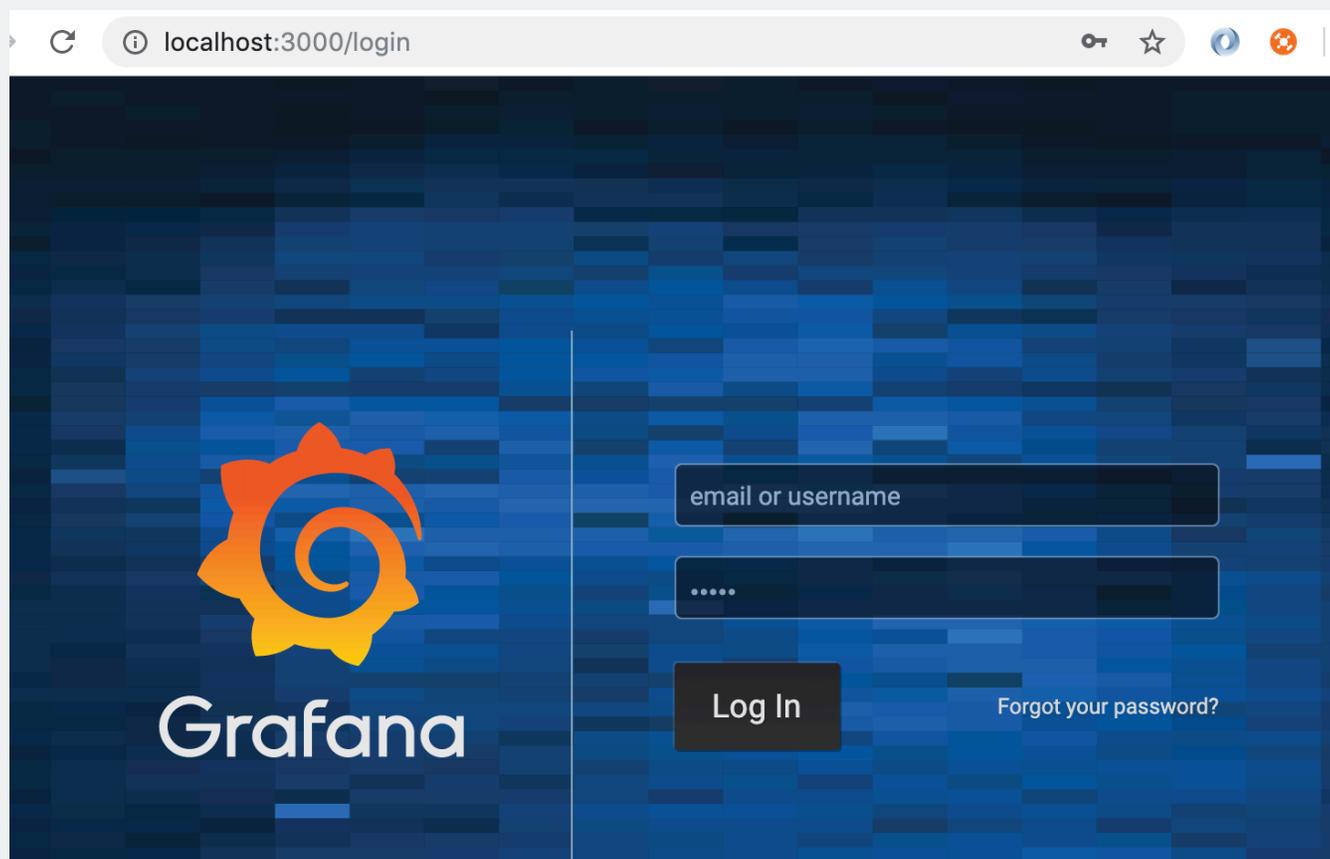
- ✓ 部署方法： Docker 部署
- ✓ 下载镜像： docker pull grafana/grafana
- ✓ 启动镜像： docker run -d -p 3000:3000 --name=jmeterGraf grafana/grafana

```
bogon:shell jizhi.qian$ docker run -d -p 3000:3000 --name=jmeterGraf grafana/grafana
c1908cce5d1124fff445b065630dfdd4d8f4811657486327c8999160c140569a
bogon:shell jizhi.qian$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
NAMES
c1908cce5d11        grafana/grafana    "/run.sh"          11 seconds ago     Up 10 seconds
    jmeterGraf
dd23e4f77c33        influxdb           "/entrypoint.sh infl..." About an hour ago   Up About an hour
p    jmeterdb
```



## 性能监控平台部署 - Grafana

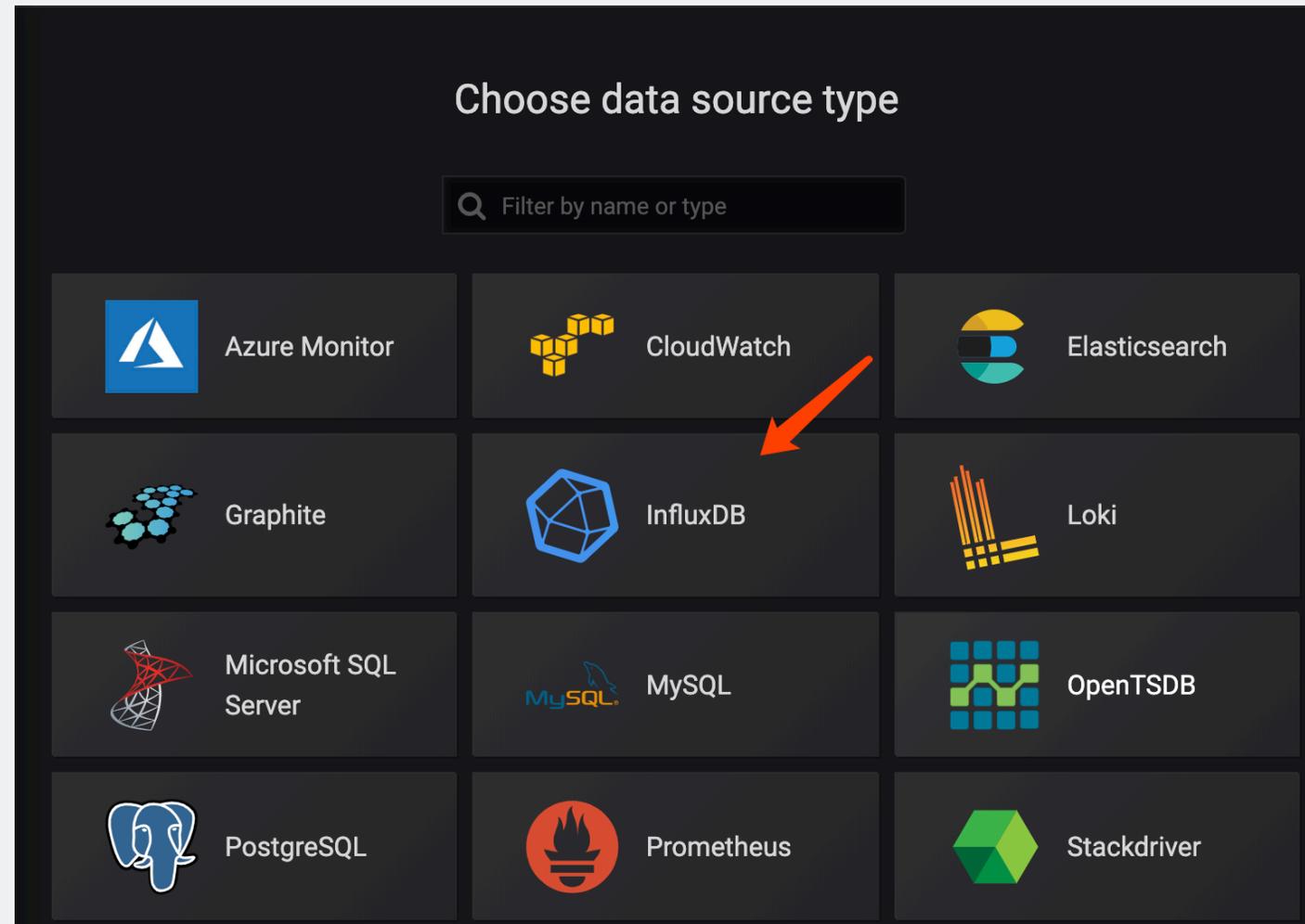
- ✓ 访问 Grafana 的控制台链接，验证它已经运行正常
- ✓ 默认用户名/密码：admin / admin，登录之后请跳过修改密码





# 性能监控平台部署 – Grafana

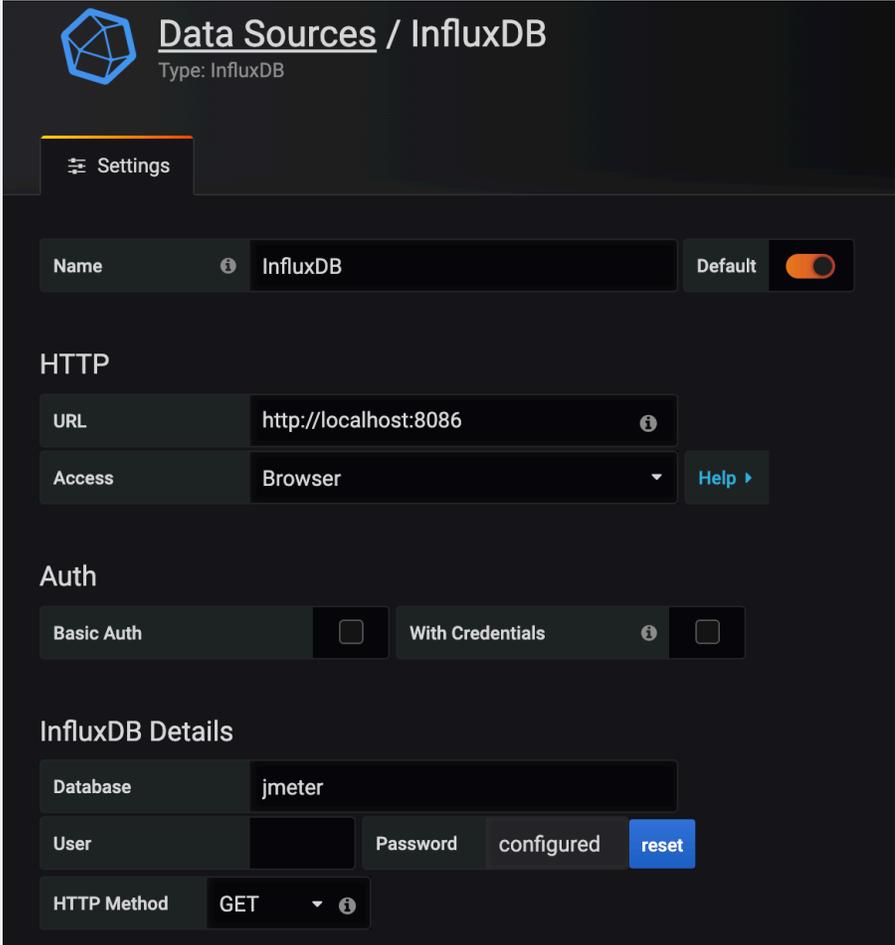
- ✓ 在Grafana中添加数据源
- ✓ 选择 Add data source
- ✓ 找到 InfluxDB
- ✓ 单击选择该DB





# 性能监控平台部署 – Grafana

- ✓ 配置InfluxDB 数据源
  - ✓ 输入 URL : http://[hostname or IP]:8086
  - ✓ Access : browser
  - ✓ Database : jmeter
- ✓ 其余参数保持默认值



The screenshot shows the Grafana configuration page for an InfluxDB data source. The page is titled "Data Sources / InfluxDB" and has a "Type: InfluxDB" label. A "Settings" tab is active. The configuration includes:

- Name:** InfluxDB (Default is checked)
- HTTP:**
  - URL:** http://localhost:8086
  - Access:** Browser (Help link available)
- Auth:**
  - Basic Auth:**  (With Credentials: )
- InfluxDB Details:**
  - Database:** jmeter
  - User:** (empty) Password: configured (reset button)
  - HTTP Method:** GET



# 性能监控平台部署 – Grafana

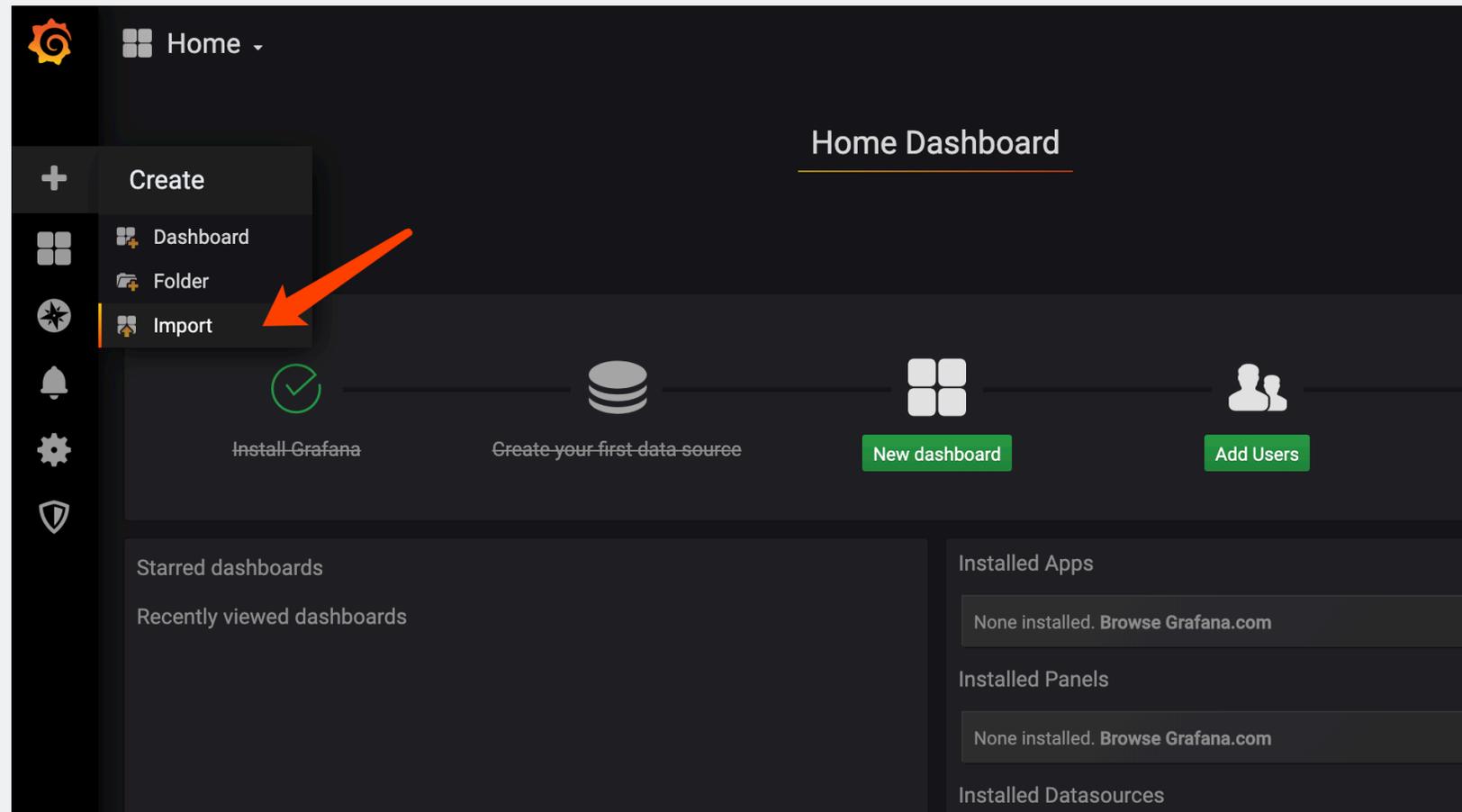
- ✓ 单击 Save & Test 按钮，确认返回消息提示正确确认信息

The screenshot displays the 'InfluxDB Details' configuration page in Grafana. At the top, the 'Database' is set to 'jmeter'. The 'User' field is empty, and the 'Password' is 'configured', with a 'reset' button next to it. The 'HTTP Method' is set to 'GET'. A green notification bar at the top right indicates 'Datasource updated'. Below this, a 'Database Access' section provides information about database permissions and query syntax, including examples like 'SHOW MEASUREMENTS ON \_internal' and 'SELECT \* FROM "\_internal"."database" LIMIT 10'. The 'Min time interval' is set to '10s'. At the bottom, another green notification bar states 'Data source is working'. The 'Save & Test' button is highlighted in green, indicating it is the active state.



# 性能监控平台部署 – Grafana

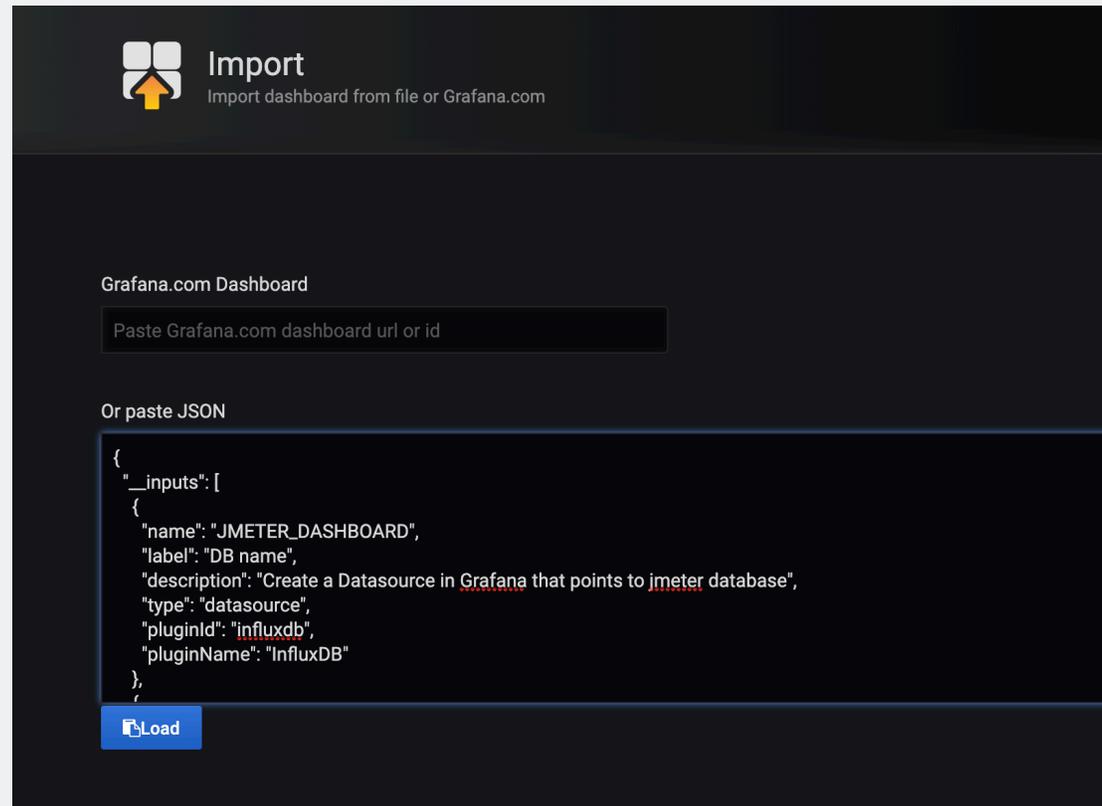
- ✓ 在Grafana 内导入JMeter Dashboard
- ✓ 进入Home Dashboard 页面，选择Import





## 性能监控平台部署 – Grafana

- ✓ 打开 jmeter 项目中的 shell/ jmeter\_dashboard.json 文件
- ✓ 将json文本复制 / 粘贴到 paste JSON 文本框中
- ✓ 单击Load按钮导入





# 性能监控平台部署 – Grafana

- ✓ 在 DB name 中选择 InfluxDB
- ✓ 单击 Import 按钮完成 Dashboard 导入

Import  
Import dashboard from file or Grafana.com

Importing Dashboard from [Grafana.com](#)

Published by

Updated on

Options

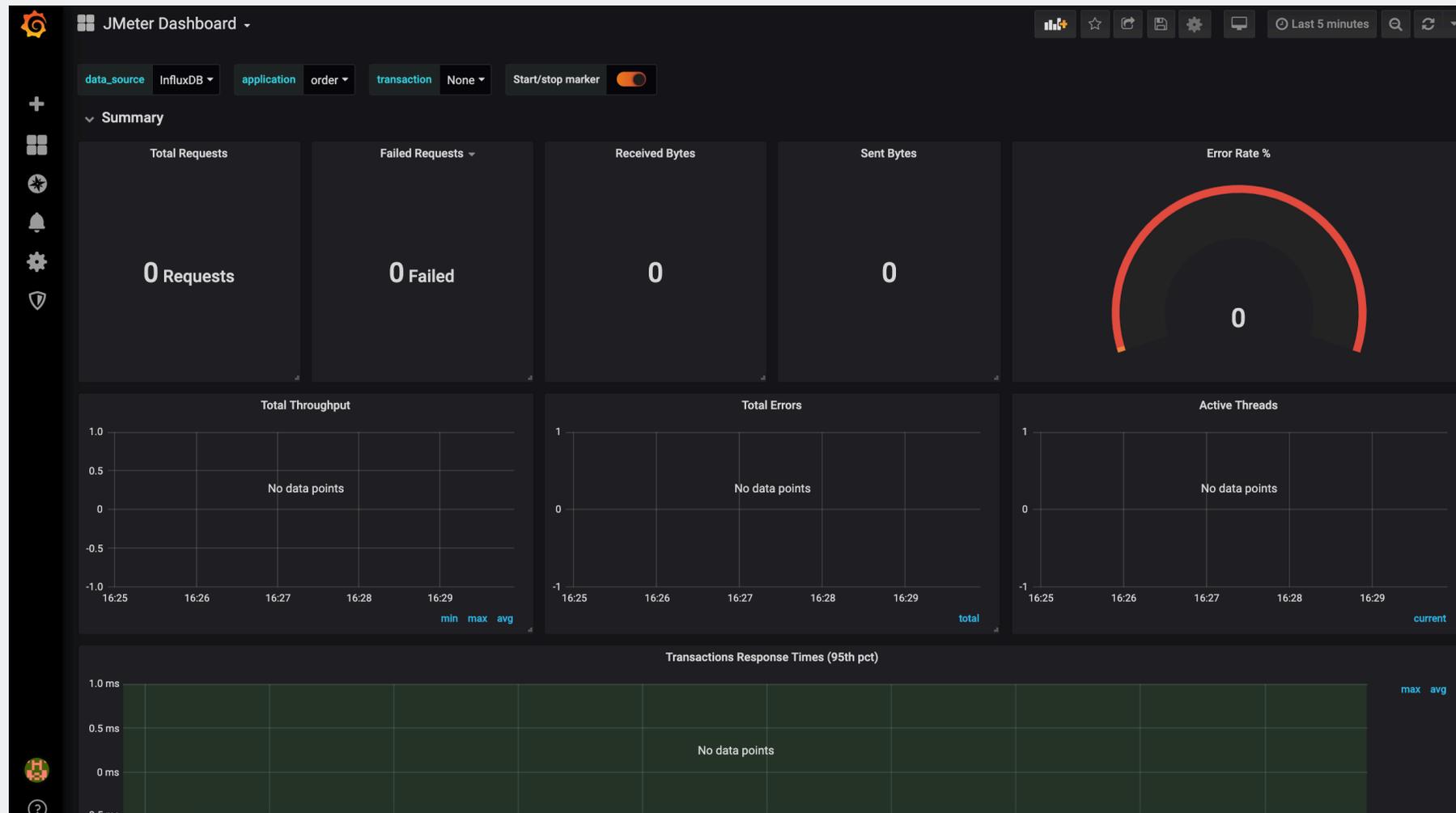
Name	JMeter Dashboard	✓
Folder	General	▼
Unique identifier (uid)	auto-generated	change
DB name	InfluxDB	✓
Measurement name	jmeter	✓
Backend send interval	5	✓

Import Cancel



# 性能监控平台部署 – Grafana

- ✓ 打开刚刚导入的JMeter Dashboard 查看结果





## 性能监控平台部署 - Grafana

- ✓ **Demo – 实操部署 Grafana**
- ✓ **更多Grafana Dashboard 可以参考官网**

<https://grafana.com/grafana/dashboards?dataSource=influxdb>



## 本章小结

- ✓ 部署性能监控平台
  - ❖ InfluxDB
  - ❖ JMeter BackEnd Listener
  - ❖ Grafana
- ✓ Docker 部署



# Agenda

- 1 部署性能监控平台**
- 2 运行与结果展示
- 3 自动化压力测试



分享课题：搭建JMeter性能监控平台

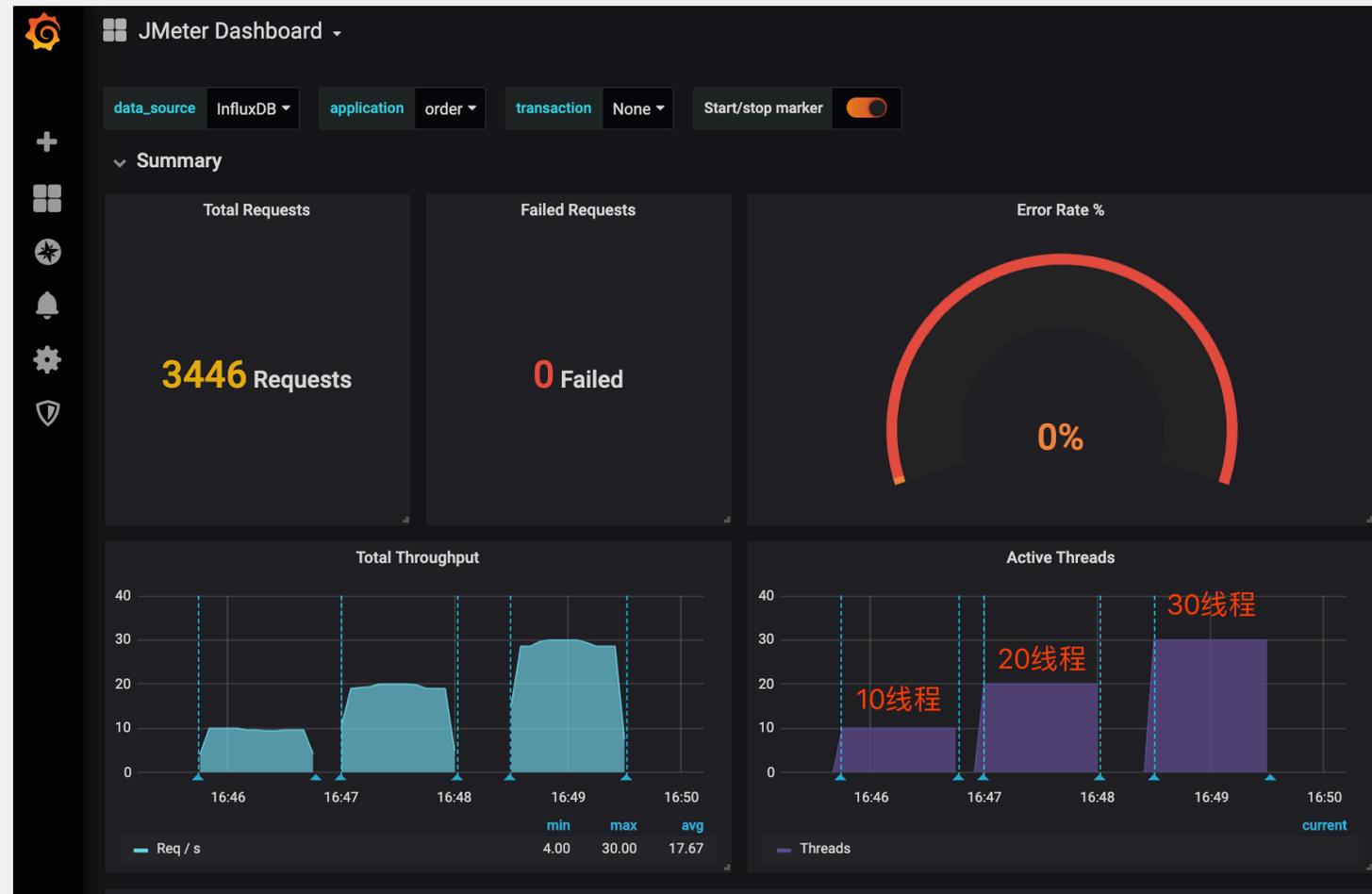
# 运行与结果展示

# 02



## 运行与结果展示

- ✓ 打开JMeter，将线程数分别调整为 10，20，30；持续压测时间为1分钟
- ✓ 分别对系统施加压力测试，然后在 Grafana 面板中可以看到运行结果





## 运行与结果展示

- ✓ **Demo – 实操运行演练**



# Agenda

- 1 部署性能监控平台**
- 2 运行与结果展示**
- 3 自动化压力测试



分享课题：搭建JMeter性能监控平台

# 自动化压力测试

# 03



# 自动化压力测试

- ✓ 为啥要自动化压测呢？
  - ❖ 手动逐步加压
  - ❖ 需要人肉改并发数，然后等待完成
  - ❖ 烦！！！！
- ✓ 所以，制定好策略，让程序自动加压，自动等待；完成后看压测监控记录，或者坐收报告
- ✓ 计算机努力的干活，我去做更重要的事情
- ✓ 希望测试生涯由此变得美好一些 😊



# 自动化压力测试

✓ 实现思路：

- ❖ JMeter 脚本（.jmx 文件）- 压测逻辑
- ❖ Shell – 控制逻辑
- ❖ 不能绕开麻烦，并发数的更改 – 想想Linux的三剑客之一哦
- ❖ JMeter 静默运行 – 脱离UI限制，使得自动化运行更稳定

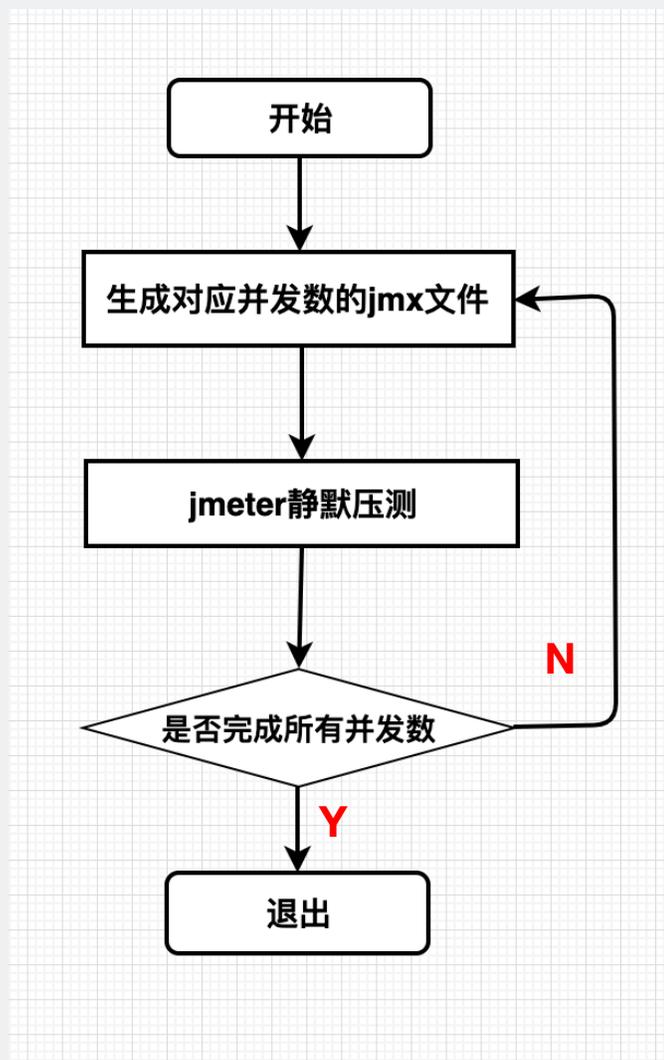
```
jmeter -n -t $jmx_file -l $jtl_file
```

注：jtl 文件是jmeter压测请求响应数据的原始文件



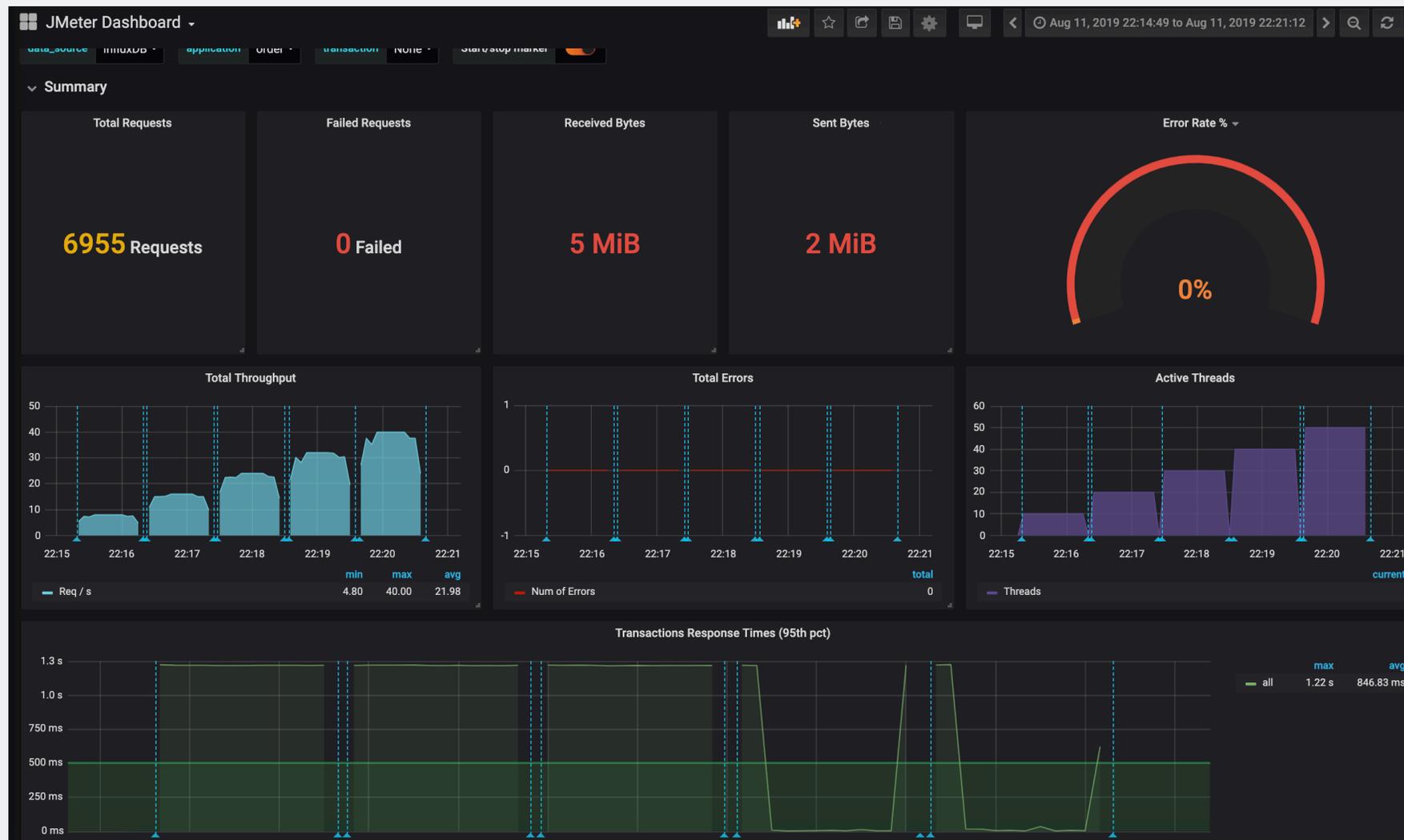
# 自动化压力测试

## ✓ 程序流程图





## ✓ 运行效果





## 自动化压力测试

- ✓ 源码解读与实操应用
- ✓ 源码位置：`{jmeter}/automation/auto_stress_test.sh`



# Agenda

- 1 部署性能监控平台
- 2 运行与结果展示
- 3 自动化压力测试



## 课程总结

- ✓ JMeter性能监控平台
- ✓ 搭建方法与JMeter中的配置步骤
- ✓ 自动化压测的程序流程思路与实操演练



谢谢

THANKS