

1. 介绍（此文档使用的版本 grafana-enterprise-10.0.3-1）

2. 清空之前的实验数据

3. 使用配置文件方式配置 Datasource

3.1. 创建配置文件

3.2. 重启服务并检查是否生效

4. 使用配置文件方式配置 Dashboard

4.1. 创建配置文件

5. 配置 Dashboard JSON 文件

5.1. 下载 JSON 文件到本地

5.2. 添加变量

5.3. 验证

1. 介绍（此文档使用的版本 grafana-enterprise-10.0.3-1）

本篇文章主要介绍如何实现自动化配置 Grafana 的数据源，Dashboard 等功能。主要作用是提高自动化部署的能力和效率，通过提前准备好的文件配置数据源等功能，就不用点点点了。

2. 清空之前的实验数据

接上篇[4-云原生监控体系-Grafana-基本使用] 的配置环境讲解。现在需要情况原来部署的 Grafana 数据，便于接下里的实验和学习。

停止服务

```
systemctl stop grafana-server.service
```

删除数据

```
rm -rf /var/lib/grafana/grafana.db
```

重启启动服务

```
systemctl start grafana-server
```

重新登录，设置密码。就像刚刚部署好之后的流程一样。

3. 使用配置文件方式配置 Datasource

数据源的配置文件应该放置在如下目录下: `/etc/grafana/provisioning/datasources/`

3.1. 创建配置文件

数据源配置文件 `datasource.yml` 内容如下

Grafana 支持配置多个 数据源，如下示例是配置了 2 个数据源。

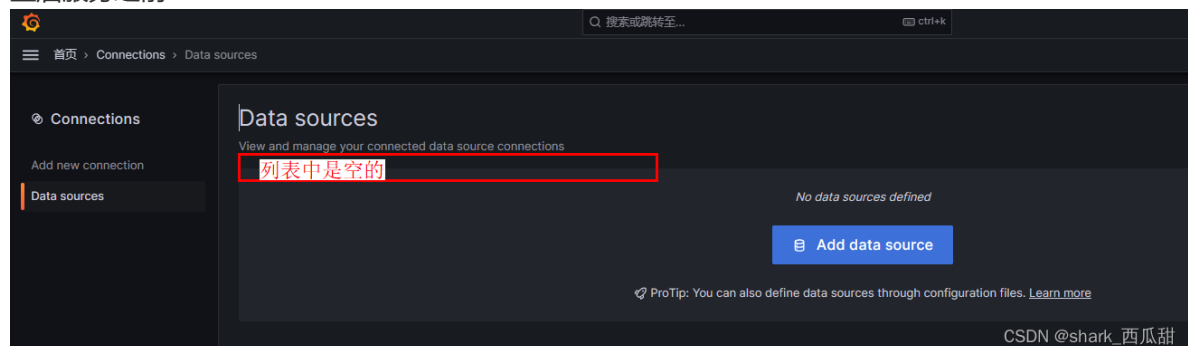
```

apiVersion: 1
datasources:
- name: Prometheus # 第一个数据源的名称，可以自定义
  type: prometheus # 数据源类型
  isDefault: true # 默认的数据源
  access: proxy
  orgId: 1
  uid: PBFA97CFB590B2093 # 自定义，需要在 Grafana 中唯一，并且会在后面的
Dashboard 配置文件中使用的
  url: http://10.10.40.179:9090 # 数据源的IP地址和端口
- name: cs1-prometheus # 如下内容是第二个数据源的信息，当然可以是zabbix等其他的数据
源
  type: prometheus
  access: proxy
  orgId: 1
  uid: cs1cigsever
  url: http://11.1.1.1:30786 # 这是个测试样例，实际环境中没有
  editable: false # 是否允许在网页上编辑此数据源的配置
  # user: '' # 配置 mysql、PostgreSQL等数据源的用户
  basicAuth: true
  basicAuthUser: shark # 基础认证的 用户名
  secureJsonData:
    basicAuthPassword: 123456 # 仅适用于 HTTP 协议认证配置密码
    # password: 123456 # 适用于 HTTP 协议、MySQL、Postgresql 认证配置密码

```

3.2. 重启服务并检查是否生效

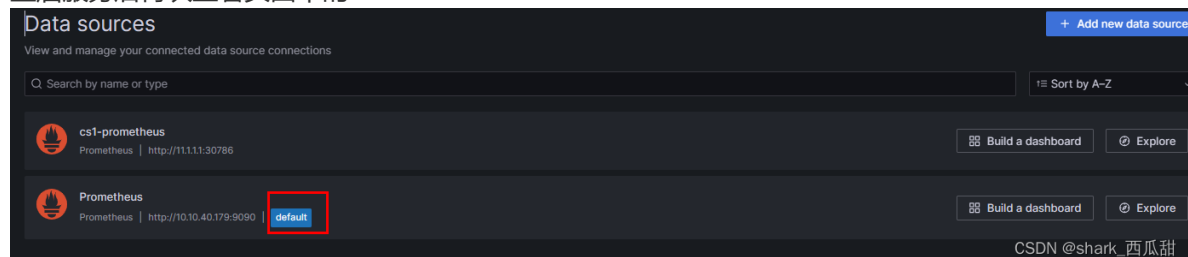
重启服务之前



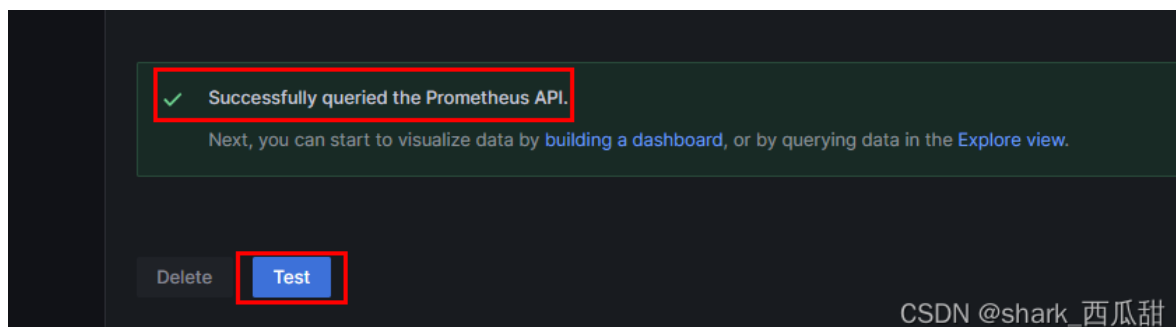
重启服务

```
systemctl restart grafana-server
```

重启服务后再次查看页面中的 Data source



点击任何一个数据源，进入详情页面，进行测试



4. 使用配置文件方式配置 Dashboard

Dashboard 的配置文件应该放置在如下目录下: `/etc/grafana/provisioning/dashboards/`

此目录下可以放置多个配置文件, 每个配置文件可以对应一个服务, 配置文件中可以指定一个目录, 目录专门放置此服务所有的 Dashboard 文件。

也可以把所有服务使用一个配置文件, 并把所有的 Dashboard 文件放到一个目录中。

4.1. 创建配置文件

如下是一个配置文件实例

```
apiVersion: 1
providers:
  # 字符串类型的名称, 需要唯一
  - name: 'Dashboard for default'
    orgId: 1
    # 逻辑上的文件夹, 会体现在页面的配置上.
    folder: ''
    # 文件夹UID。如果未指定, 将自动生成
    folderUid: ''
    type: file
    # 默认情况下, 如果删除文件, Grafana会删除数据库中的仪表板。
    # 这里禁止这种行为
    disableDeletion: false
    # Grafana 重新加载更新仪表板文件到自己数据库中的频率
    updateIntervalSeconds: 10
    # 允许从UI更新已设置的仪表板, false 不允许更新
    allowUiUpdates: false
    options:
      # 磁盘上仪表板文件的路径, type 的是 file 时, 必须配置
      path: /etc/grafana/dashboards
      # 使用文件系统中的文件夹名称在Grafana中创建 folder
      foldersFromFilesStructure: true
```

当Grafana启动时, 它将更新/插入配置路径中可用的所有 dashboard 文件。然后, 稍后每隔 `updateIntervalSeconds` 轮询该路径, 并查找更新的json文件, 并将其更新/插入数据库。

Dashboards are provisioned to the root level if the folder option is missing or empty.

如果 `folder` 选项没有配置或者为空值, 那么 dashboards 将设置为根级别。

更详细的配置说明参考官方文档: <https://grafana.com/docs/grafana/latest/administration/provisioning/>


```

    "name": "DS_PROMETHEUS",
    "options": [],
    "query": "prometheus",
    "queryValue": "",
    "refresh": 1,
    "regex": "/^Prome.*/",
    "skipUrlSync": false,
    "type": "datasource"
  },
  // 添加内容结束
  {
    ...
  }
]
}

```

关于变量名 `"name": "DS_PROMETHEUS"`，需要把变量名配置给需要设置数据源的地方，比如如下内容：

```

{
  "datasource": {
    "type": "prometheus",
    "uid": "${DS_PROMETHEUS}" // 这里就是需要变量名的地方
  },

```

可以使用 sed 批量替换为我们定义的变量名，以我们的使用的这个为例，需要做如下替换。

```
sed -i 's/DS_TEST-DS_PROMETHEUS/DS_PROMETHEUS/g' node-exporter.json
```

关于正则 `"regex": ""`

`/^Prome.*/` // 正则，书写格式: /你的正则/, 例如: `/^Prometheus.*/`, 这个很重要, 用于匹配到 grafana 已经配置生效的数据源名称。

在此文档所用环境中是 `datasource.yml` 文件中字段 `name` 的值。

```

apiVersion: 1
datasources:
- name: Prometheus # 第一个数据源的名称，可以自定义
  ...

```

可以留空，留空后需要配置如下内容，表示选择默认值。

```

"templating": {
  "list": [
    // 如下内容是添加的
    {
      "current": {
        "selected": false,
        "text": "default",
        "value": "default"
      },
      ...
    }
  ]
}

```

5.3. 验证

配置完成后，重启服务

```
systemctl restart grafana-server
```

修改 Dashboard JSON 文件，或者向其所在的目录 /etc/grafana/dashboards 添加新的文件，无需重启 服务。

访问页面中的 **仪表板** 查看效果

