

Kibana入门

Kibana 是一款开源的数据分析和可视化平台，它是 Elastic Stack 成员之一，设计用于和 Elasticsearch 协作。您可以使用 Kibana 对 Elasticsearch 索引中的数据进行搜索、查看、交互操作。您可以很方便的利用图表、表格及地图对数据进行多元化的分析和呈现。

官网: <https://www.elastic.co/cn/kibana>



配置和安装

到下载地址，选择对应的版本: <https://www.elastic.co/cn/downloads/kibana>

Download Kibana

Want it hosted? Deploy on Elastic Cloud. [Get Started »](#)

Version: 7.9.1

Release date: September 04, 2020

License: [Elastic License](#)

Downloads: [↑ WINDOWS](#) [sha asc](#)

[↑ LINUX 64-BIT](#) [sha asc](#)

[↑ DEB 64-BIT](#) [sha asc](#)

[↑ MAC](#) [sha asc](#)

[↑ RPM 64-BIT](#) [sha asc](#)

[↑ LINUX AARCH64](#) [sha asc](#)

下载完成后，将文件拷贝到我们的服务器上，然后解压

```
# 解压
tar -zxvf kibana-7.9.1-linux-x86_64.tar.gz

# 重命名
mv kibana-7.9.1-linux-x86_64 kibana
```

然后在进入kibana目录，找到config文件夹下的kibana.yml进行配置的修改

```
vim /soft/kibana/config/kibana.yml
```

然后找到下面的内容

```
#对外暴露服务的地址
server.host: "0.0.0.0"

#配置Elasticsearch
elasticsearch.url: "http://127.0.0.1:9200"
```

启动

修改配置完成后，我们就可以启动kibana了

```
#启动
./bin/kibana
```

点击启动，发现报错了

```
[root@ElasticStack config]# cd ..
[root@ElasticStack kibana]# ./bin/kibana
Kibana should not be run as root. Use --allow-root to continue.
[root@ElasticStack kibana]# su elsearch
```

原因是kibana不能使用root用户进行启动，所以我们切换到elsearch用户

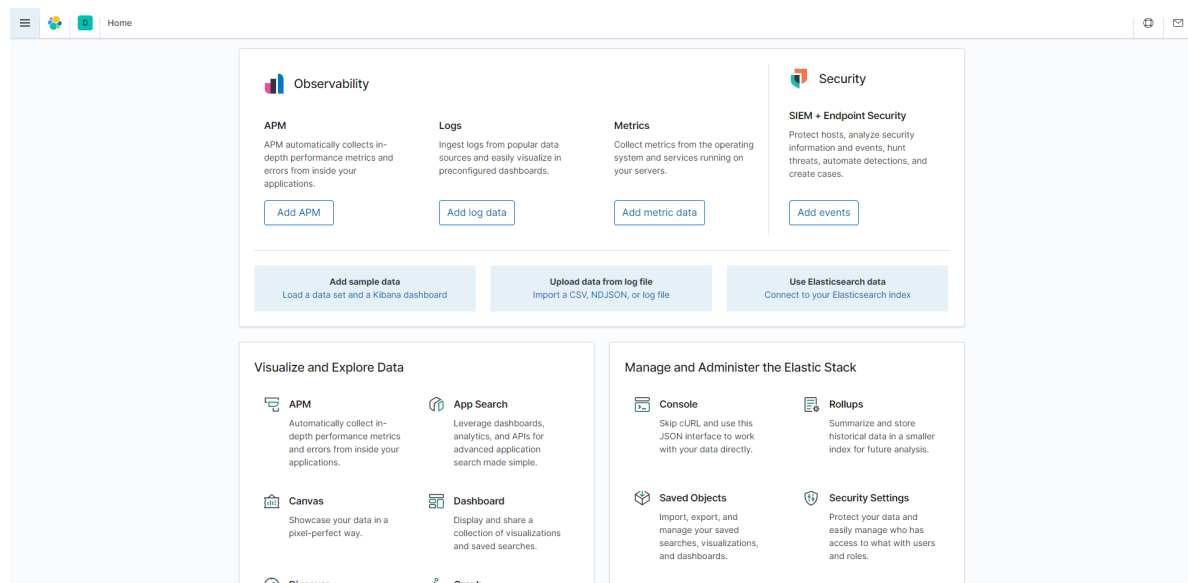
```
# 将soft文件夹的所有者改成elsearch
chown elsearch:elsearch /soft/ -R

# 切换用户
su elsearch

# 启动
./bin/kibana
```

然后打开下面的地址，即可访问我们的kibana了

<http://202.193.56.222:5601/>



功能说明

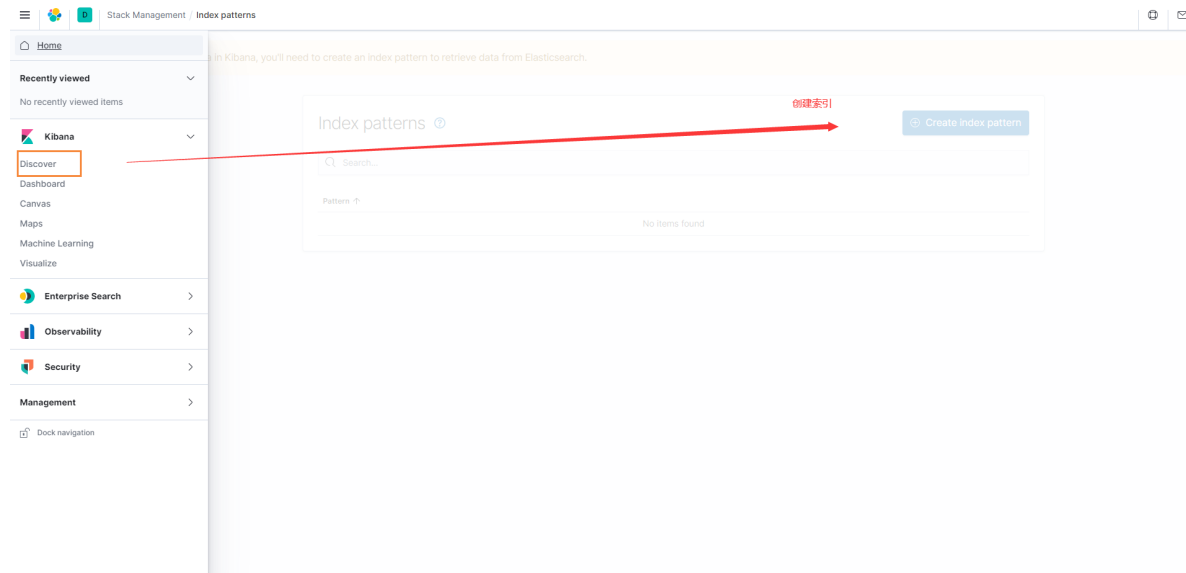


- Discover: 数据探索
- Visualize: 可视化
- Dashboard: 仪表盘
- Timelion: 时序控件
- Canvas: 画布
- Machine Learning: 机器学习
- Infrastructure: 基本信息
- Logs: 数据日志展示
- APM: 性能监控

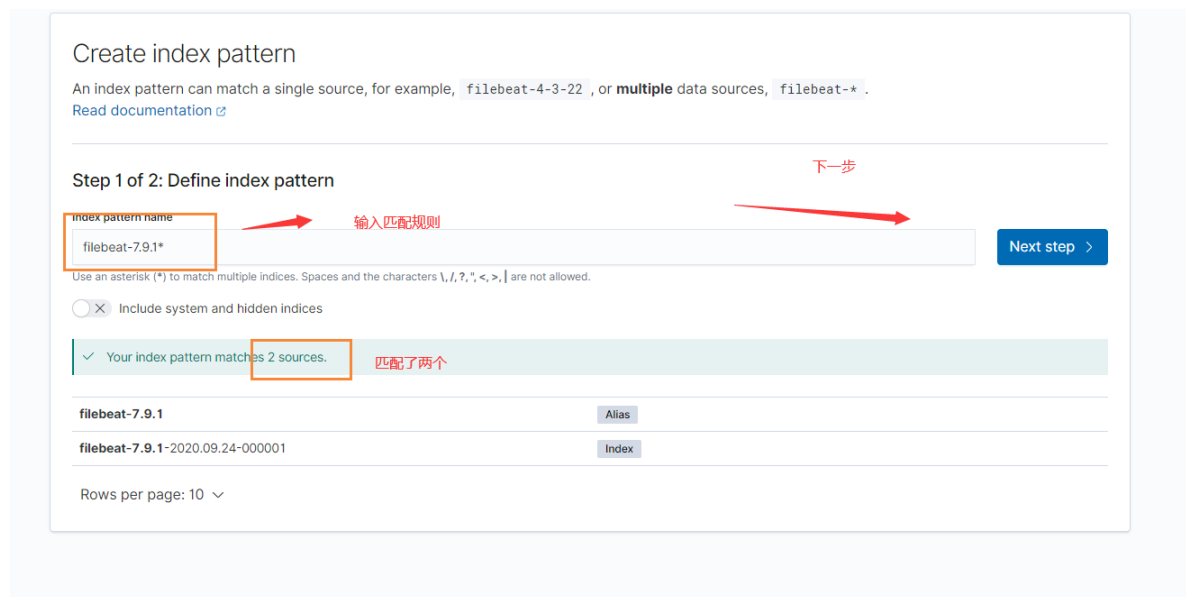
- Dev Tools: 开发者工具
- Monitoring: 监控
- Management: 管理

数据探索

先添加索引信息



然后我们就输入匹配规则进行匹配



然后选择时间字段，一般选择第一个

Create index pattern

An index pattern can match a single source, for example, `filebeat-4-3-22`, or **multiple** data sources, `filebeat-*`.

[Read documentation](#)

Step 2 of 2: Configure settings

filebeat-7.9.1*

Select a primary time field for use with the global time filter.

Time field

@timestamp

Refresh

然后选择时间字段，一般使用第一个

[Show advanced options](#)

[< Back](#)

Create index pattern

索引创建完毕后

★ filebeat-7.9.1*

Time Filter field name: @timestamp

Default

This page lists every field in the **filebeat-7.9.1*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#).

Fields (4827)

Scripted fields (0)

Source filters (0)

Search

All field types

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				
_type	string		●	●	
activemq.caller	string		●	●	
activemq.log.stack_trace	string		●	●	
activemq.thread	string		●	●	
activemq.user	string		●	●	

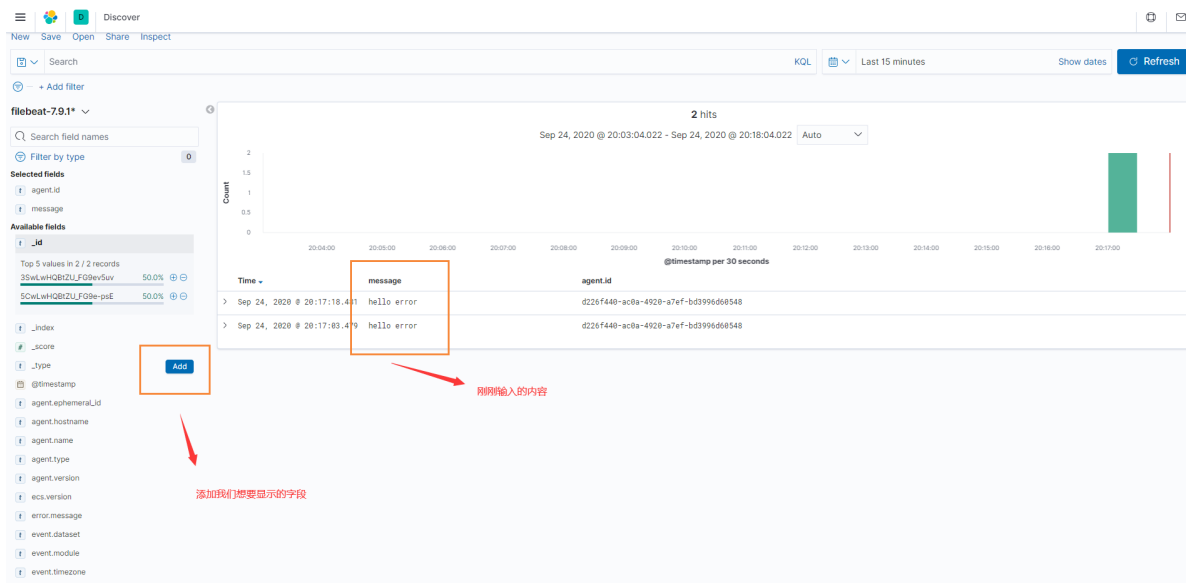
Rows per page: 10

< 1 2 3 4 5 ... 483 >

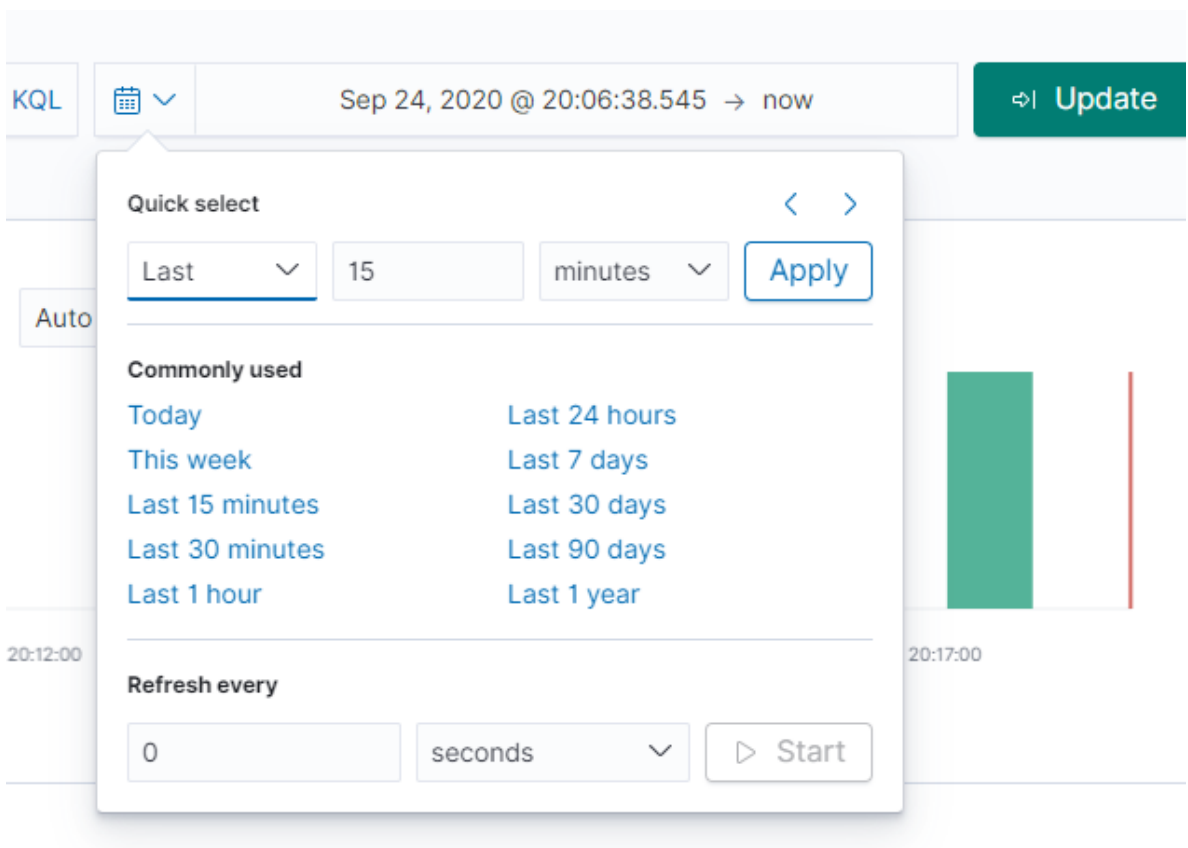
然后我们就可以往nginx error.log日志文件中，添加几天错误记录

```
echo "hello error" >> error.log
```

我们追加了两条数据，然后到kibana的discover中，刷新页面，就能够看到我们刚添加的日志了，同时我们点击右侧还可以选择需要展示的字段，非常的方便



点击右上角，我们还可以针对时间来进行过滤



Metricbeat仪表盘

现在将Metricbeat的数据展示在Kibana中，首先需要修改我们的MetricBeat配置

#修改metricbeat配置

setup.kibana:

host: "192.168.40.133:5601"

#安装仪表盘到kibana【需要确保kibana在正常运行，这个过程可能会有些耗时】

./metricbeat setup --dashboards

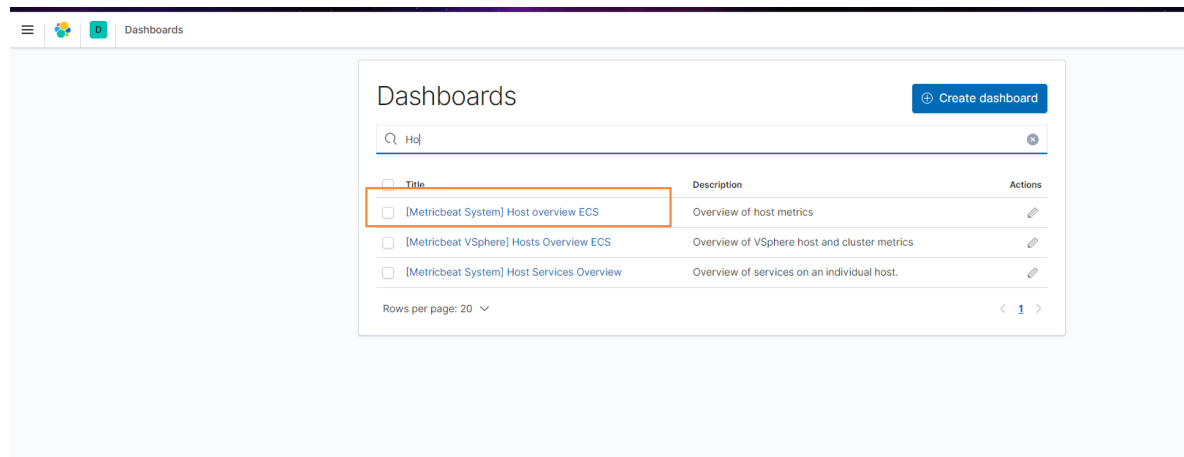
安装完成后，如下所示

```
[root@ElasticStack metricbeat]# vim metricbeat.yml
[root@ElasticStack metricbeat]# ./metricbeat setup --dashboards
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
```

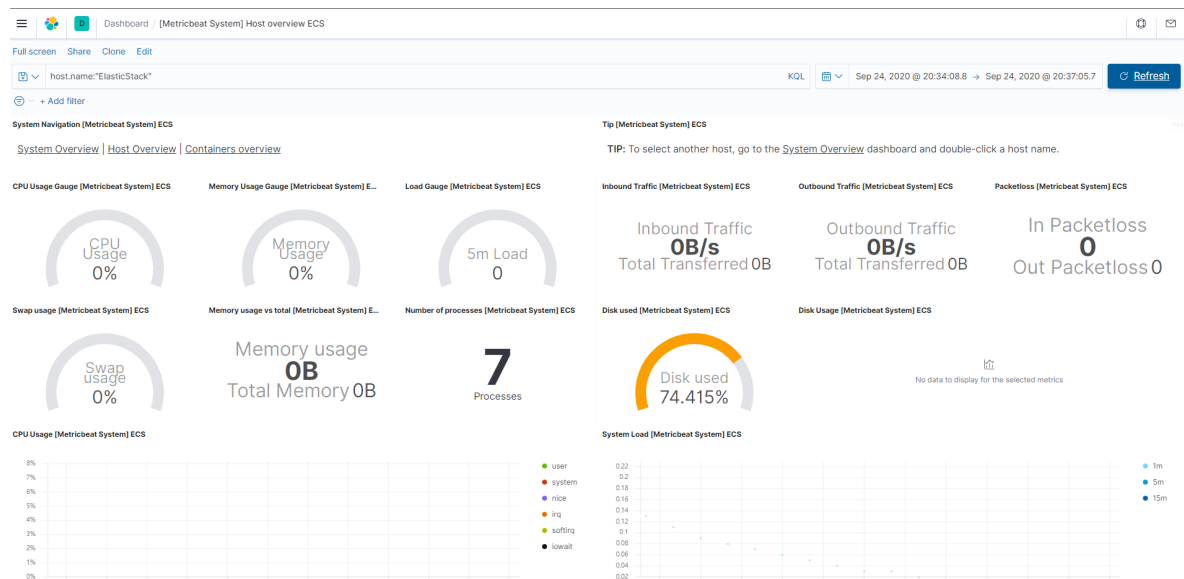
然后我们启动Metricbeat

```
./metricbeat -e
```

然后到kibana页面下，找到我们刚刚安装的仪表盘



然后我们能够看到非常多的指标数据了



Nginx指标仪表盘【Metricbeat】

选择Metricbeat的nginx仪表盘即可

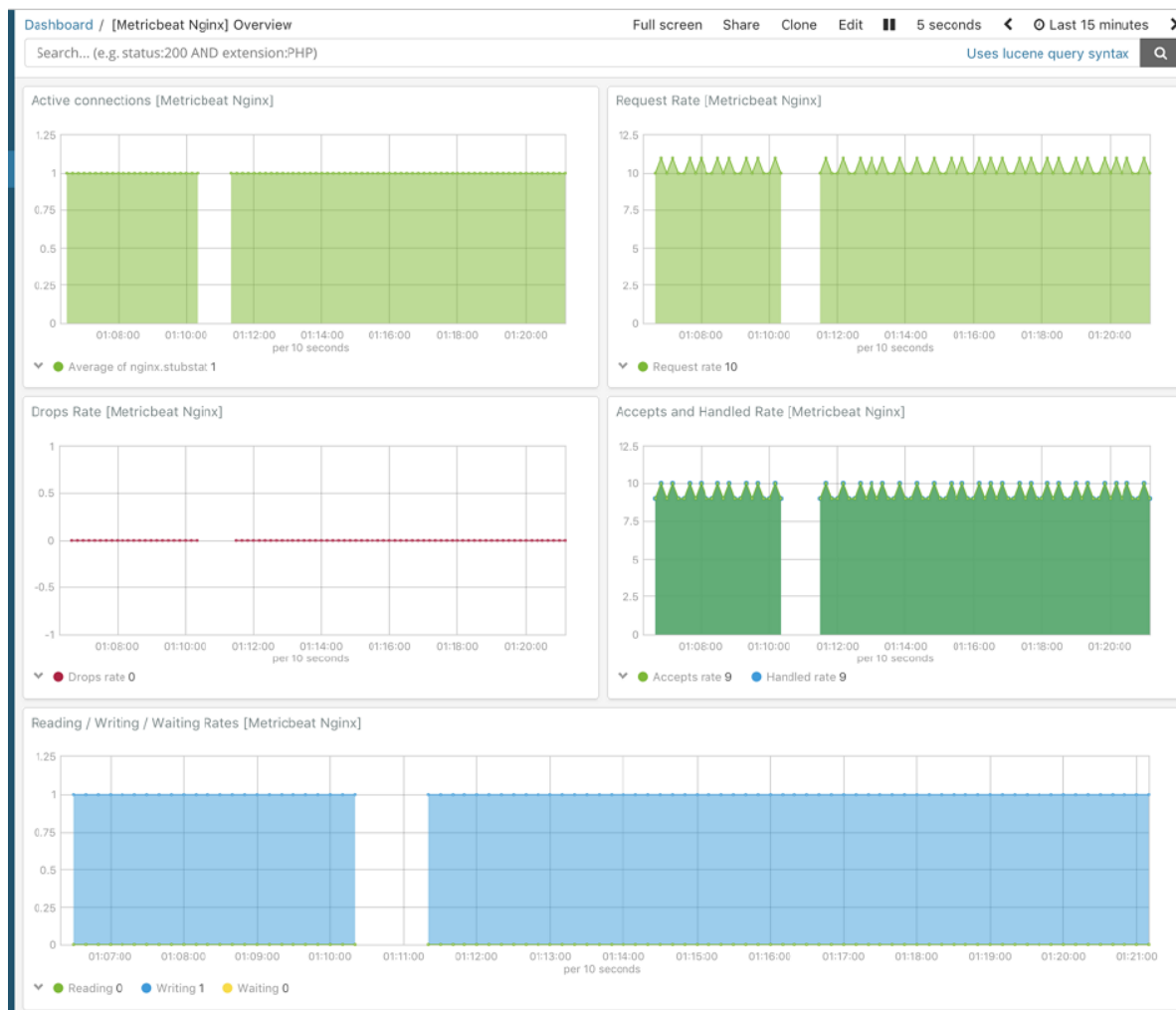
Dashboards

Create new dashboard

Title	Description	Actions
<input type="checkbox"/> [Metricbeat Nginx] Overview	Overview dashboard for the Nginx module in Metricbeat	Edit

Rows per page: 20

然后就能够看到Nginx的指标信息了



Nginx日志仪表盘

我们可以和刚刚Metricbeat的仪表盘一样，也可以将filebeat收集的日志记录，推送到Kibana中
首先我们需要修改filebeat的 mogublog-nginx.yml配置文件


```
filebeat.inputs:
setup.template.settings:
  index.number_of_shards: 1
output.elasticsearch:
  hosts: ["127.0.0.1:9200"]
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yaml
  reload.enabled: false
setup.kibana:
  host: "127.0.0.1:5601"
```

然后按照仪表盘

```
./filebeat -c mogublog-nginx.yml setup
```

等待一会后，仪表盘也安装成功了

```
[elasticsearch@elasticstack filebeat]$ vim mogublog-nginx.yml
[elasticsearch@elasticstack filebeat]$ ./filebeat -c mogublog-nginx.yml setup
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Setting up ML using setup --machine-learning is going to be removed in 8.0.0. Please use the ML app instead.
See more: https://www.elastic.co/guide/en/machine-learning/current/index.html
Loaded machine learning job configurations
Loaded ingest pipelines
```

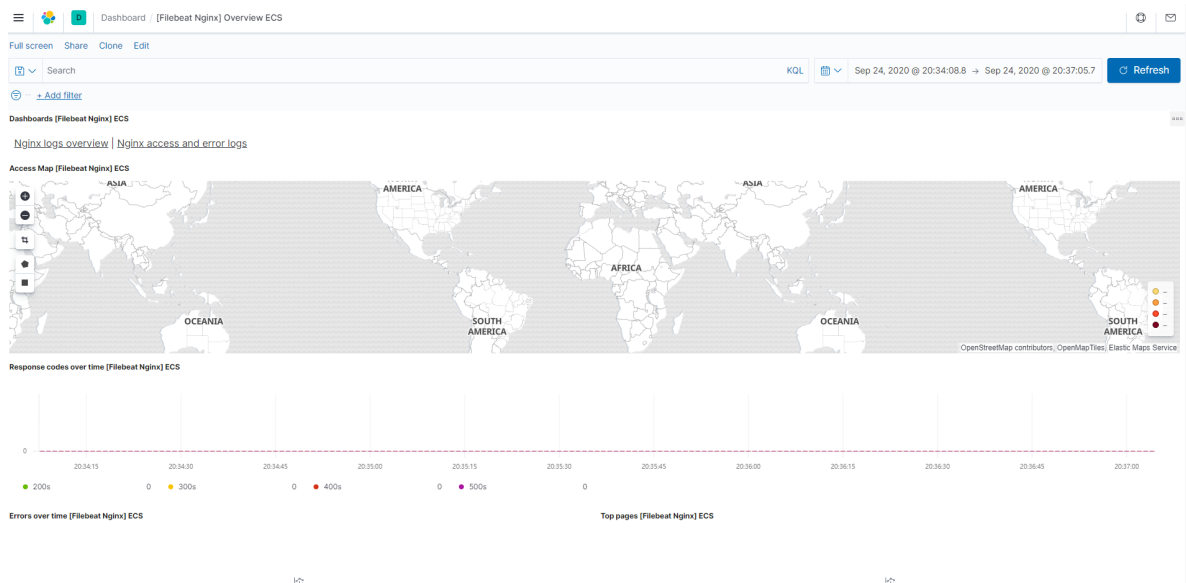
然后我们启动filebeat即可

```
./filebeat -e -c mogublog-nginx.yml
```

启动完成后，我们回到我们的Kibana中，找到Dashboard，添加我们的filebeat - nginx即可



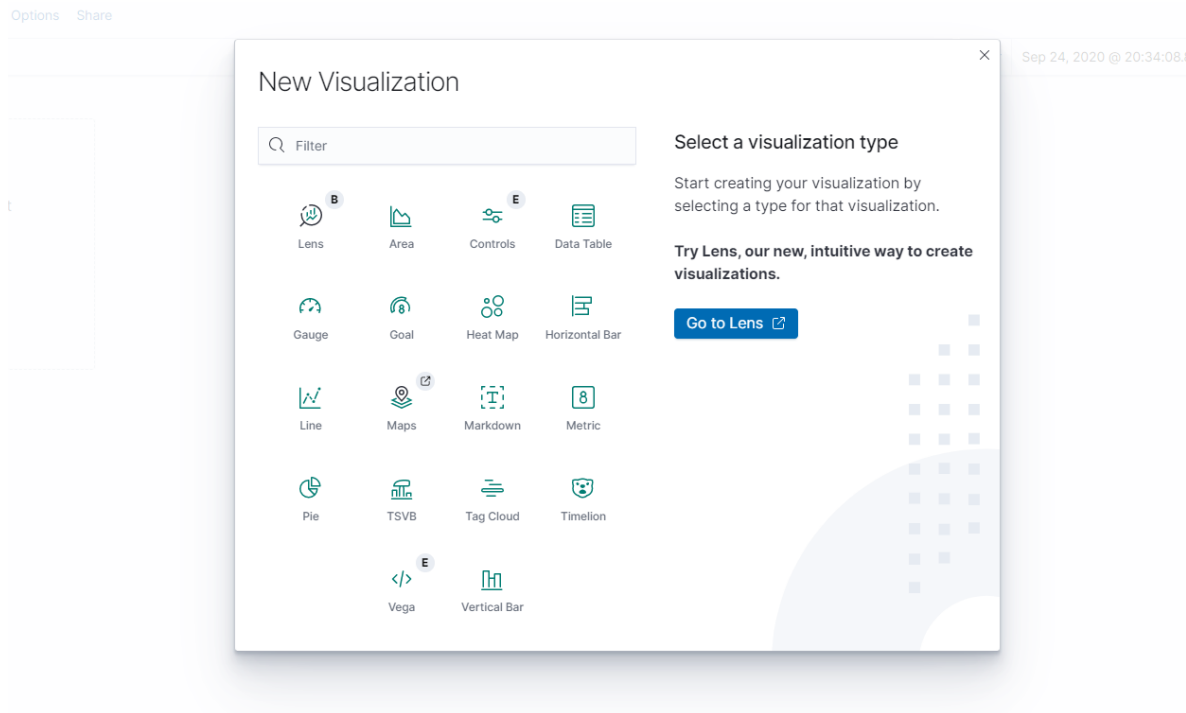
然后就能看到我们的仪表盘了，上图就是请求的来源



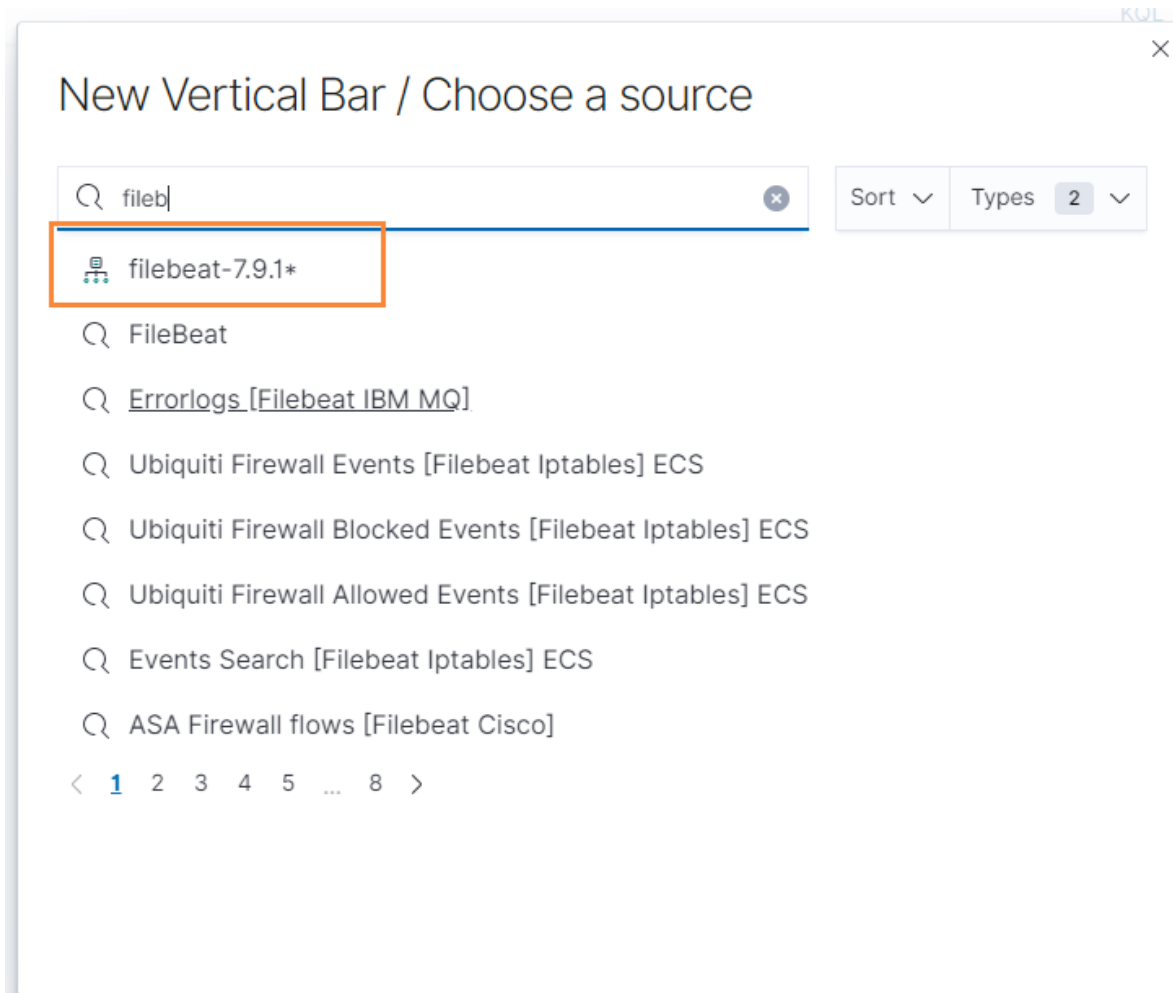
需要注意的是，这些仪表盘本身是没有的，我们需要通过filebeat来进行安装

Kibana自定义仪表盘

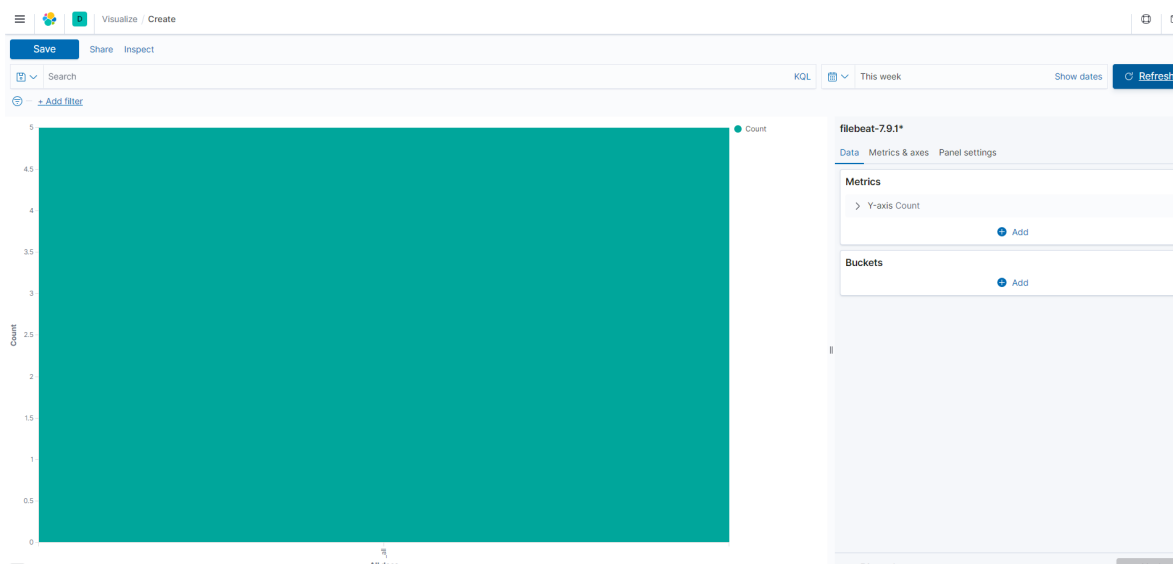
在Kibana中，我们也可以自定义图标，如制作柱形图



我们选择最下面的 Vertical Bar，也就是柱形图，然后在选择我们的索引

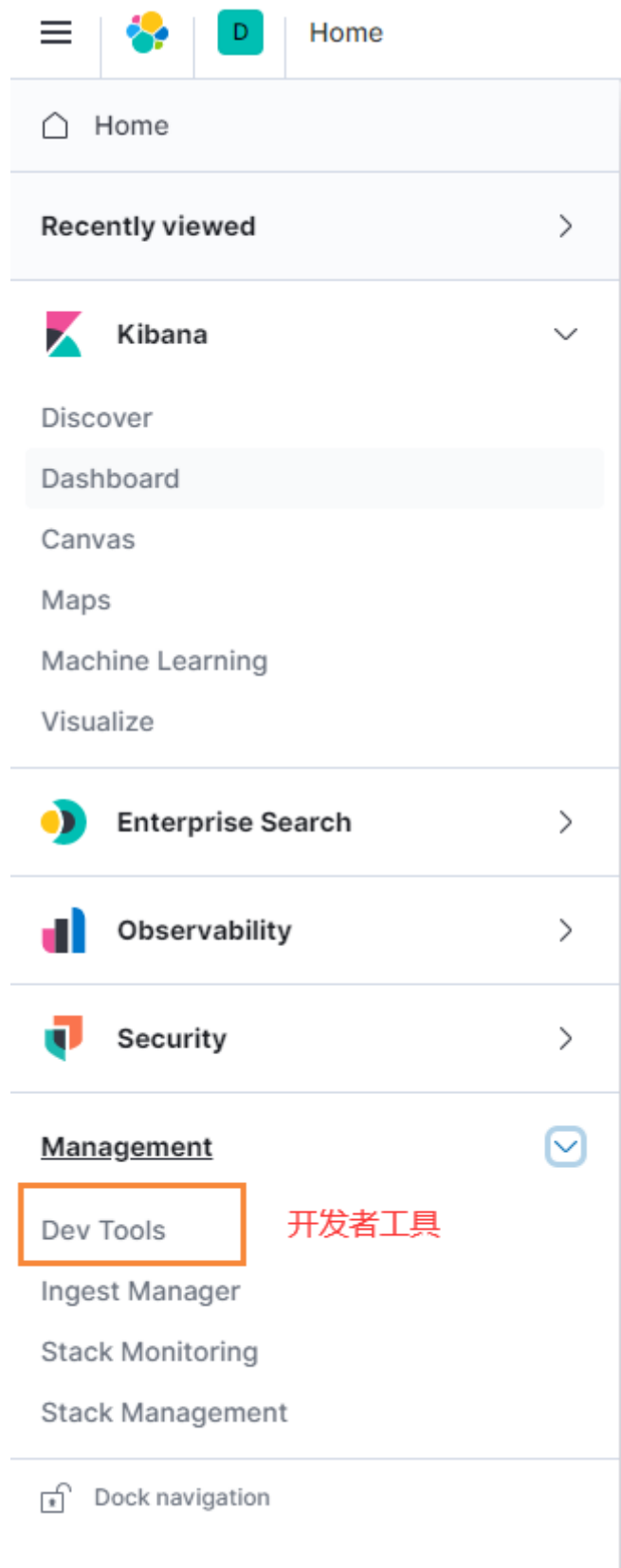


这样就出来了



开发者工具

在Kibana中，为开发者的测试提供了便捷的工具使用，如下：



我们就可以在这里面写一些请求了

```
1
2 POST itcast/user/1001
3 {
4   "name": "孙武",
5   "age": 20,
6   "mail": "111@qq.com"
7 }
8
9 GET itcast/user/1001
```

输入命令，然后执行

