

ALIBABA CLOUD

# Alibaba Cloud

日志服务  
开发指南

文档版本：20201022

 阿里云

## 法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

# 通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
<code>Courier</code> 字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
<i>斜体</i>	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1. 命令行工具CLI	05
2. 可视化开发	06
2.1. 控制台内嵌及分享	06
2.2. 配置控制台内嵌参数	08
2.3. 对接Jaeger	20
2.4. 对接JDBC	23
2.5. 对接DataV	27
2.6. 对接Grafana	32

# 1. 命令行工具CLI

日志服务命令行工具CLI支持以命令行工具完成常见功能，例如日志查询、完整性检查与自动分页，并支持多账户与跨域复制。

## 安装

日志服务CLI支持以下操作系统：

- Windows
- Mac OS
- Linux

安装步骤请参见[日志服务CLI安装](#)。

## 主要功能

日志服务命令行工具CLI的主要功能包括：

- 配置CLI
- 创建Logtail配置
- 跨域复制项目配置
- 读取日志
- 查询日志
- 迁移Elasticsearch数据
- 高速跨域复制日志，历史数据重新索引与数仓投递

更多信息请参见[日志服务CLI](#)。

## 2. 可视化开发

### 2.1. 控制台内嵌及分享

日志服务提供嵌入式集成查询分析和仪表盘等页面功能，方便您查看日志数据。

#### 背景信息

为使您在配置完成日志服务采集和查询分析等功能后，能直接使用日志服务的查询分析和仪表盘页面功能，并将日志数据分享给其他用户进行查看，同时避免使用RAM会带来众多RAM账号的管理成本。日志服务提供指定日志库查询页面和仪表盘页面嵌入自建Web功能，不需要登录日志服务控制台即可直接访问日志服务查询分析和仪表盘等页面，访问后可以通过[授权RAM用户](#)限制操作权限，例如授予只读权限。

操作流程如下图所示。



#### 操作步骤

1. 登录自建Web，通过Web服务端访问令牌服务（STS）获取临时AK对和安全Token。
  - STS使用说明请参见[通过STS实现跨账号访问日志服务资源](#)。
  - 授予指定日志库的访问权限请参见[创建RAM用户及授权](#)。
2. 调用STS SDK服务接口获取日志服务登录Token。

#### 说明

- STS返回的安全Token中可能包含特殊字符，请将特殊字符进行URL编码后再输入。
- 由于Token有效期为3小时，建议您在自建Web设置链接，每次请求时生成新的登录Token，通过302请求返回进行跳转。

请求示例：

```
http://signin.aliyun.comsignin-intl.aliyun.com/federation?Action=GetSigninToken
    &AccessKeyId=<STS返回的临时AK>
    &AccessKeySecret=<STS返回的临时Secret>
    &SecurityToken=<STS返回的安全Token>
    &TicketType=mini
```

3. 生成免登录链接。

## i. 获取日志服务页面链接。

## ■ 完整查询分析页面：

```
https://sls4service.console.aliyun.com/next/project/<Project名称>/logsearch/<日志库名称>?hideTopbar=true&hideSidebar=true
```

## ■ 查询页面：

```
https://sls4service.console.aliyun.com/next/project/<Project名称>/logsearch/<日志库名称>?isShare=true&hideTopbar=true&hideSidebar=true
```

## ■ 仪表盘页面：

```
https://sls4service.console.aliyun.com/next/project/<Project名称>/dashboard/<仪表盘ID>?isShare=true&hideTopbar=true&hideSidebar=true
```

 说明 上述链接中仪表盘ID为网页链接上的ID，并非仪表盘的展示名称。

## ii. 将获取到的登录Token与日志服务页面链接一并生成免登录访问链接。请求示例：

```
http://signin.aliyun.comsignin-intl.aliyun.com/federation?Action=Login
    &LoginUrl=<登录失效跳转的地址，一般配置为自建Web配置302跳转的URL。>
    &Destination=<实际访问日志服务页面，支持查询页面和仪表盘页面。如果有参数，则需要使用encodeURIComponent对参数进行转码。>
    &SigninToken=<获取的登录Token>
```

## 示例

Java、PHP和Python代码示例如下：

- **Java**

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-sts</artifactId>
  <version>3.0.0</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>3.5.0</version>
</dependency>
<dependency>
  <groupId>org.apache.httpcomponents</groupId>
  <artifactId>httpclient</artifactId>
  <version>4.5.5</version>
</dependency>
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.47</version>
</dependency>
```

- PHP
- Python

## 2.2. 配置控制台内嵌参数

通过设置日志服务控制台内嵌参数，可以调整内嵌页面的显示效果。

日志服务提供免登方式嵌入自建Web页面，支持您快速方便地对日志进行查询分析。在此基础上，还提供了一系列UI参数与第三方自建Web页面进行融合展示。免登方式嵌入自建Web页面详情请参见[控制台内嵌及分享](#)。

### URL格式

所有UI参数均以URL格式进行控制，URL格式示例如下：

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}?
参数1&参数2
```

#### 说明

- 除`${ProjectName}`和`${LogstoreName}`之外的参数必须位于URL末尾的问号(?)之后。
- 支持同时设置多个参数，参数之间通过and (&) 连接。
- 增加参数`theme=dark&sls_iframe=true`，可将页面设置为黑色主题。



## 公共参数

使用公共参数设置控制台页面。

参数名	类型	是否必选	说明	示例
hideSidebar	boolean	否	隐藏侧边栏。	hideSidebar=true
hideTopbar	boolean	否	隐藏顶部阿里云标题栏。	hideTopbar=true
hiddenBack	boolean	否	隐藏Logstore查询分析页面返回按钮。	hiddenBack=true
hiddenChangeProject	boolean	否	隐藏Project概览。	hiddenOverview=true
hiddenOverview	boolean	否	关闭Tab访问的历史记录。	ignoreTabLocalStorage=true
keyFilter	JSON	否	<p>过滤显示导航栏资源列表，支持过滤的资源类型如下：</p> <ul style="list-style-type: none"> <li>logstore：日志库</li> <li>savedsearch：快速查询</li> <li>dashboard：报表</li> </ul> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>通过短划线 (-) 表示模糊匹配。</li> <li>格式为JSON类型，必须经 encodeURL 进行两次转码。</li> </ul> </div>	<pre>{   "logstore": [     "logstore-xx"],   "savedsearch": [     "savedsearch-xx"],   "dashboard": [     "dashboard-xx"]} </pre>
queryTimeType	long	否	<p>指定查询的时间范围，详情请参见<a href="#">queryTimeType指定查询的时间范围说明</a>。取值范围为：</p> <ul style="list-style-type: none"> <li>1~26：表示指定查询的时间范围为数字对应的区间。</li> <li>-2：表示自定义（相对）。此时必须配置 start 和 end。例如：start:-10m,end:now。</li> <li>-3：表示自定义（整点）。此时必须配置 start 和 end。例如：start:-2h,end:absolute。</li> <li>99：表示自定义时间范围。此时必须设置 startTime 和 endTime，且只能配置为时间戳。</li> </ul>	queryTimeType=1

参数名	类型	是否必选	说明	示例
startTime	timestamp (date)	否	指定查询时间范围的起始时间。 当queryTimeType设置为99时生效。	startTime=1547776643
endTime	timestamp (date)	否	指定查询时间范围的结束时间。 当queryTimeType设置为99时生效。	endTime=1547776731

### queryTimeType指定查询的时间范围说明

queryTimeType	代表含义
1	1分钟（相对）
2	15分钟（相对）
3	1小时（相对）
4	4小时（相对）
5	1天（相对）
6	1周（相对）
7	30天（相对）
8	1分钟（整点）
9	15分钟（整点）
10	1小时（整点）
11	4小时（整点）
12	1天（整点）
13	1周（整点）
14	30天（整点）
15	今天（整点）
16	昨天（整点）
17	前天（整点）
18	本周（整点）
19	上周（整点）
20	本月（整点）
21	本季度（整点）

queryTimeType	代表含义
22	今天（相对）
23	5分钟（相对）
24	今年（整点）
25	本月（相对）
26	上月（整点）
27	本周（相对）
99	自定义时间，此时要传入startTime和endTime。

URL参数及效果示例如下所示：

- 使用如下URL隐藏Logstore查询分析页面返回按钮、Project选择、Project概览入口和顶部的阿里云标题栏。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}
?hiddenBack=true&hiddenChangeProject=true&hiddenOverview=true&hideTopbar=true
```

- 使用如下URL隐藏侧边栏。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}
/?hideSidebar=true
```

- 使用如下URL隐藏Logstore查询分析页面的返回按钮。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}
/?hiddenBack=true
```

- 使用如下URL过滤导航栏资源列表。

例如在URL中设置参数keyFilter，通过设置的JSON对象对左侧资源列表进行筛选。需要指定显示名称包含aegis-和名称为500osslog的日志库、包含OSS的快速查询以及包含DDOS的报表。

原始JSON为{"logstore":["aegis-","500osslog"],"savedserach":["oss"],"dashboard":["ddos"]},其中aegis-表示模糊匹配名称中包含aegis的所有日志库，500osslog表示精确匹配名称为500osslog的日志库，即使用名称和短横线(-)来表示模糊查询。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}
/?keyFilter=%7B"logstore":%5B"aegis-","500osslog"%5D,"savedsearch":%5B"oss"%5D,"dashboard":%5B"ddos"%5D%7D
```

- 使用如下URL设置时间选择器。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}
/?queryTimeType=2
```

## Logstore查询分析页面相关参数

使用Logstore查询分析页面相关参数设置Logstore查询分析页面。

参数名	类型	是否必选	说明	示例
ProjectName	string	是	Project名称。	无
LogstoreName	string	是	Logstore名称。	无
queryString	string	否	预先搜索的字符串。	queryString=\${使用base64编码后的sql语句}
readOnly	boolean	否	隐藏编辑、修改按钮，例如分享、查询分析属性，另存为快速查询、另存为告警等。	readOnly=true
encode	string	否	为了避免queryString中出现特殊字符，建议使用encode=base64，此时queryString为base64编码后的字符串。	encode=base64
hiddenEtl	boolean	否	隐藏数据加工入口。	hiddenEtl=true
hiddenShare	boolean	否	隐藏分享按钮。	hiddenShare=true
hiddenIndexSetting	boolean	否	隐藏索引设置按钮。	hiddenIndexSetting=true
hiddenSavedSearch	boolean	否	隐藏快速查询按钮。	hiddenSavedSearch=true
hiddenAlert	boolean	否	隐藏告警按钮。	hiddenAlert=true
hiddenQuickAnalysis	boolean	否	默认收起快速分析栏。	hiddenQuickAnalysis=true
hiddenDownload	boolean	否	隐藏下载功能。	hiddenDownload=true

参数名	类型	是否必选	说明	示例
keyDispalyMode	string	否	内容列显示。 • single: 整行。 • multi: 换行。	keyDispalyMode=single

URL参数及效果示例如下所示：

- 使用如下URL设置预先搜索的字符串。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}
?queryString=${使用base64编码后的sql语句}
```

- 使用如下URL隐藏编辑、修改按钮。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}
?readOnly=true
```

- 使用如下URL隐藏数据加工入口。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}
?hiddenEtl=true
```

## Logstore快速查询页面相关参数

使用Logstore快速查询页面相关参数设置快速查询页面。

参数名	类型	是否必选	说明	示例
ProjectName	string	是	Project名称。	无
savedSearchName	string	是	快速查询名称。	{savedSearchName}

URL参数及效果示例如下所示：

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/savedsearch/${savedSearchName}
```

## 仪表盘内嵌参数

使用仪表盘内嵌参数设置仪表盘页面。

参数名	类型	是否必选	说明	示例
ProjectName	string	是	Project名称。	无
dashboardName	string	是	仪表盘名称。	无
filters	string	否	过滤条件，需要手动转码。 例如：转码前格式为 <code>encodeURIComponent('filters=key1:value1&amp;filters=key2:value2')</code> 。转码后为 <code>"filters%3Dkey1%3Avalue1%26filters%3Dkey2%3Avalue2"</code> 。	<code>encodeURIComponent('filters=key1:value1&amp;filters=key2:value2')</code>
token	JsonString	否	变量替换，需要手动转码。 例如：转码前为 <code>encodeURIComponent('token=[{"key": "projectName", "value": "1"}, {"key": "xxx", "value": "yy"}]')</code> 。转码后为 <code>"token%3D%5B%7B%22key%22%3A%20%22projectName%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22key%22%3A%20%22xxx%22%2C%20%22value%22%3A%20%22yy%22%7D%5D"</code> 。	<code>encodeURIComponent('token=[{"key": "projectName", "value": "1"}, {"key": "xxx", "value": "yy"}]')</code>
readOnly	boolean	否	隐藏仪表盘页面的编辑、设置按钮，例如编辑、告警等按钮。	<code>readOnly=true</code>
hiddenFilter	boolean	否	隐藏过滤条件。	<code>hiddenFilter=true</code>
hiddenToken	boolean	否	隐藏变量替换。	<code>hiddenToken=true</code>

参数名	类型	是否必选	说明	示例
hiddenProject	boolean	否	隐藏Project信息。	hiddenProject=true
hiddenEdit	boolean	否	隐藏编辑按钮。	hiddenEdit=true
hiddenReport	boolean	否	隐藏订阅按钮。	hiddenReport=true
hiddenTitleSetting	boolean	否	隐藏标题设置按钮。	hiddenTitleSetting=true
hiddenReset	boolean	否	隐藏重置时间按钮。	hiddenReset=true
autoFresh	string	否	报表定时刷新，例如：30s、5min，最小刷新间隔必须大于15s。	autoFresh=5m

URL参数及效果示例如下所示：

- 使用如下URL使仪表盘页面为只读模式，即不可编辑模式。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboardId}/?readOnly=true
```

- 使用如下URL为仪表盘增加两个过滤条件key1=value1和key2=value2。

仪表盘中所有图表的查询分析语句会针对过滤条件先进行一次过滤，再继续执行查询语句中的条件。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboardId}?encodeURIComponent('filters=key1:value1&filters=key2:value2')
```

- 使用如下URL增加多个变量替换条件。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboardId}?encodeURIComponent('token=[{"key": "projectName", "value": "1"}, {"key": "xxx", "value": "yy"}]')
```

- 使用如下URL使仪表盘每5分钟刷新一次。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboard_name}?autoFresh=5m
```

## 树状结构参数

树状结构参数用于定义控制台的左侧导航栏。

参数名	类型	是否必选	说明	示例
treeConfig	JSON	否	定义左侧导航树状结构，需要手动转码。完整示例  例如转码前为 {"logstore": {"expand":true,"resourceList":["delete-log"],"template":["savedsearch","alert"]}}。转码后为 eyjsb2dzdG9yZSI6eyJleHBhb mQiOnRydWUsInJlc291cmNlTGldCl6WyJkZWxldGUtbG9nIl0sInRlbXBsYXRlljpbInNhdmVkc2VhcmNoliwiYWxlcniX19。	treeConfig=eyjsb2dzdG9yZSI6eyJleHBhb mQiOnRydWUsInJlc291cmNlTGldCl6WyJkZWxldGUtbG9nIl0sInRlbXBsYXRlljpbInNhdmVkc2VhcmNoliwiYWxlcniX19
treeEncode	string	否	treeConfig的编码方式，默认为空，表示不编码。如果要编码，仅支持base64。	treeEncode=base64

treeConfig参数完整示例如下所示：

```

{
  "logstore": {
    "search": true,
    "expand": true,
    "resourceList": [
      "L1",
      "L2"
    ],
    "template": [
      "favor",
      "logtail",
      "import",
      "etl",
      "savedsearch",
      "alert",
      "export",
      "consumergroup",
      "dashboard"
    ]
  },
  "machineGroup": {

```



```
machineGroup : {
  "search": true,
  "resourceList": [
    "m1",
    "m2"
  ]
},
"savedSearch": {
  "search": true,
  "resourceList": [
    "s1",
    "s2"
  ]
},
"alarm": {
  "search": true,
  "resourceList": [
    "a1",
    "a2"
  ]
},
"dashboard": {
  "search": true,
  "resourceList": [
    "d1",
    "d2"
  ]
},
"etl": {
  "search": true,
  "resourceList": [
    "e1",
    "e2"
  ]
}
}
```

## treeConfig参数说明

参数名	类型	是否必选	说明
logstore	Object	否	Logstore列表的资源控制。
template	string[]	否	Logstore列表内的一些功能控制，详情请参见 <a href="#">template参数说明</a> 。
machineGroup	Object	否	机器组列表的资源控制。
savedSearch	Object	否	快速查询列表的资源控制。
alert	Object	否	告警列表的资源控制。
dashboard	Object	否	仪表盘列表的资源控制。
etl	Object	否	数据加工列表的资源控制。

### 资源相关参数说明

参数名	类型	是否必选	说明
search	boolean	否	是否显示搜索框，默认为true，表示不显示。
resourceList	String[]	否	显示当前资源的列表。如果为空数组，则显示空列表，如果不设置该参数则全部显示，精准匹配。默认显示全部列表。
expand	boolean	否	是否展开列表，默认为false，表示不展开，仅针对Logstore列表有效。

### template参数说明

参数名	类型	是否必选	说明
favor	string	否	我的关注
logtail	string	否	Logtail配置
import	string	否	数据导入
etl	string	否	数据加工
savedsearch	string	否	快速查询
alert	string	否	告警
export	string	否	数据导出
consumergroup	string	否	数据消费
dashboard	string	否	可视化仪表盘

使用如下URL设置左侧导航栏。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}?treeconfig=eyJsb2dzdG9yZSI6eyJleHBhbmQiOnRydWUsInJlc291cmNlTGZldCI6WyJkZWxldGUtbG9nIl0sInRlbXBsYXRlIjpblnNhdmVkc2VhcmNoliwiYWxlcniXX19&hiddenBack=true&hiddenChangeProject=true&hiddenOverview=true&hideTopbar=true&treeEncode=base64&ignoreTabLocalStorage=true
```



## 仪表盘高阶参数

将iframe容器嵌入仪表盘页面时，无法确定iframe容器的高度，有可能导致两层滚动条，包括嵌入页面外层滚动条和iframe内部报表的滚动条。此时，您可以使用仪表盘高阶参数用于自适应仪表盘高度。



对于外层iframe的高度可以通过日志服务postmessage的dashboardHeight来获取，并设置为iframe高度，从而避免该问题。示例代码如下所示：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>POST message测试</title>
</head>
<style>
  * {
    padding: 0;
    margin: 0;
  }

  iframe {
    display: block;
    width: 100%;
  }
</style>
<body>
  <script>
    window.addEventListener('message',function(e){
      console.log(e.data.dashboardHeight)
      document.getElementById('test').style.height = e.data.dashboardHeight + 'px'
    });
  </script>
  <div style="height: 700px;">somethings</div>
  <iframe id="test" src="http://sls4service.console.aliyun.com/lognext/project/${projectName}/dashbo
  ard/${dashboardName}?hideTopbar=true&product=${productCode}">
</body>
</html>
```

## 2.3. 对接Jaeger

本文介绍如何使用Jaeger客户端调用数据对接日志服务。

### 背景信息

容器、Serverless编程方式提升了软件交付与部署的效率。在架构的演化过程中，可以看到以下变化。

- 应用架构从单体系统逐步转变为微服务，其中业务逻辑变为微服务之间的调用与请求。
- 资源角度来看，传统服务器这个物理单位逐渐淡化，变为了虚拟资源模式。

□

从以上两个变化可以看到这种弹性、标准化的架构背后，原本运维与诊断的需求也变得复杂。为应对这种变化趋势，诞生了一系列面向DevOps的诊断与分析系统，包括集中式日志系统（Logging）、集中式度量系统（Metrics）和分布式追踪系统（Tracing）。

除Jaeger外，阿里云还提供支持OpenTracing链路追踪产品XTrace。

## Logging, Metrics和Tracing的特点

- Logging用于记录离散的事件

例如，应用程序的调试信息或错误信息，Logging是我们诊断问题的依据。

- Metrics用于记录可聚合的数据

例如，队列的当前深度可被定义为一个度量值，在元素入队或出队时被更新；HTTP请求个数可被定义为一个计数器，新请求到来时进行累加。

- Tracing用于记录请求范围内的信息

例如，一次远程方法调用的执行过程和耗时。Tracing是我们排查系统性能问题的利器。

通过以上信息，可以对已有系统进行分类。例如，Zipkin专注于tracing领域；Prometheus开始专注于metrics，随着时间推移可能会集成更多的tracing功能，但不太可能深入logging领域；ELK、阿里云日志服务这样的系统开始专注于logging领域，但同时也不断地集成其他领域的特性到系统中来，正向上图中的圆心靠近。

更多三者关系详细信息请参见[Metrics, tracing, and logging](#)。下面为您重点介绍Tracing。

## Tracing

流行的Tracing软件有：

- Dapper (Google)：各Tracer的基础
- StackDriver Trace (Google)
- Zipkin (Twitter)
- Appdash (golang)
- 鹰眼 (Taobao)
- 谛听 (盘古, 阿里云云产品使用的Trace系统)
- 云图 (蚂蚁Trace系统)
- sTrace (神马)
- X-ray (AWS)

分布式追踪系统发展快，种类多，核心步骤一般有三个：

- 代码埋点
- 数据存储
- 查询展示

下图是一个分布式调用的例子，客户端发起请求，请求首先到达负载均衡器，接着经过认证服务，计费服务，然后请求资源，最后返回结果。

数据被采集存储后，分布式追踪系统一般会选择使用包含时间轴的时序图来呈现这个调用链。但在数据采集过程中，由于需要侵入用户代码，并且不同系统的API并不兼容，导致如果切换追踪系统，会有较大改动。

## OpenTracing

为了解决不同的分布式追踪系统API不兼容的问题，诞生了OpenTracing规范。OpenTracing是一个轻量级的标准化层，它位于应用程序/类库和追踪或日志分析程序之间。

- OpenTracing优势：
  - OpenTracing已进入CNCF，为全球的分布式追踪，提供统一的概念和数据标准。
  - OpenTracing通过提供平台无关、厂商无关的API，使开发人员能够方便的添加或更换追踪系统。
- OpenTracing数据模型：

OpenTracing中的Trace通过归属于此调用链的Span来隐性的定义。特别说明，一条Trace可以被认为是一个由多个Span组成的有向无环图（DAG图），Span与Span的关系被命名为References。如下示例，调用链就是由8个Span组成的。



另外，基于时间轴的时序图可以更好的展现Trace，可以使用以下示例如下：



每个Span包含以下对象：

- An operation name：操作名称。
- A start timestamp：起始时间。
- A finish timestamp：结束时间。
- Span Tag：一组键值对构成的Span标签集合。键值对中，键必须为string，值可以是字符串、布尔、或者数字类型。

- **Span Log**: 一组span的日志集合。每次log操作包含一个键值对，以及一个时间戳。键值对中，键类型必须为string，值可以是任意类型。不是所有的支持OpenTracing的Tracer，都需要支持所有的值类型。
- **SpanContext**: Span上下文对象。每个SpanContext包含以下状态：
  - 任何一个OpenTracing的实现，都需要将当前调用链的状态（例如：Trace和Span的ID），依赖一个独特的Span去跨进程边界传输。
  - **Baggage Items**是Trace的随行数据，是一个键值对集合，存在于Trace中，也需要跨进程边界传输。
- **References (Span间关系)**: 相关的零个或者多个Span (Span间通过SpanContext建立这种关系)。

更多关于OpenTracing数据模型的知识，请参见[OpenTracing 数据模型](#)。

关于所有的OpenTracing实现，请参见[OpenTracing](#)。在这些实现中，比较流行的为[Jaeger](#)和[Zipkin](#)。

## Jaeger

Jaeger是Uber推出的一款开源分布式追踪系统，兼容OpenTracing API。

□

如上图所示，Jaeger主要由以下几部分组成。

- **Jaeger Client**: 为不同语言实现了符合OpenTracing标准的SDK。应用程序通过API写入数据，client library把Trace信息按照应用程序指定的采样策略传递给jaeger-agent。
- **Agent**: 它是一个监听在UDP端口上接收span数据的网络守护进程，它会将数据批量发送给collector。它被设计成一个基础组件，部署到所有的宿主机上。Agent将client library和collector解耦，为client library屏蔽了路由和发现collector的细节。
- **Collector**: 接收jaeger-agent发送来的数据，然后将数据写入后端存储。Collector被设计成无状态的组件，因此您可以同时运行任意数量的jaeger-collector。
- **Data Store**: 后端存储被设计成一个可插拔的组件，支持将数据写入cassandra、elastic search。
- **Query**: 接收查询请求，然后从后端存储系统中检索Trace并通过UI进行展示。Query是无状态的，您可以启动多个实例，把它们部署在Nginx负载均衡器后面。

## Jaeger on Alibaba Cloud Log Service

[Jaeger on Alibaba Cloud Log Service](#)是基于Jaeger开发的分布式追踪系统，支持将采集到的追踪数据持久化到日志服务中，并通过Jaeger的原生接口进行查询和展示。

□

Jaeger的优势：

- 原生Jaeger仅支持将数据持久化到cassandra和elasticsearch中，您需要自行维护后端存储系统的稳定性，调节存储容量。Jaeger on Alibaba Cloud Log Service借助阿里云日志服务的海量数据处理能力，让您享受Jaeger在分布式追踪领域给您带来便捷的同时无需过多关注后端存储系统的问题。
- Jaeger UI部分仅提供查询、展示Trace的功能，对分析问题、排查问题支持不足。使用Jaeger on Alibaba Cloud Log Service，您可以借助日志服务强大的查询分析能力，助您更快分析出系统中存在的问题。
- 相对于Jaeger使用elasticsearch作为后端存储，使用日志服务的好处是支持按量付费，成本仅为elasticsearch的13%。详情请参见[查询分析全方位对比 \(ELK\)](#)。

Jaeger on Alibaba Cloud Log Service配置步骤请参见[GitHub](#)。

## 2.4. 对接JDBC

本文以实际案例演示如何使用JDBC连接日志服务、读取日志数据，及使用MySQL协议和SQL语法来计算日志。

## 前提条件

已为目标字段设置字段索引并开启统计功能，详情请参见[开启并配置索引](#)。

## 背景信息

MySQL是当前流行的关系型数据库，很多软件支持通过MySQL传输协议和SQL语法获取MySQL数据。您只需要对SQL语法熟悉，即可完成对接。日志服务提供了MySQL协议查询和分析日志数据。您可以使用标准MySQL客户端连接到日志服务，使用标准的SQL语法计算和分析日志。支持MySQL传输协议的客户端包括MySQL client，JDBC和Python MySQLdb。

JDBC的使用场景：

- 使用可视化类工具，例如DataV、Tableau或Kibana来通过MySQL协议连接日志服务。
- 使用Java的JDBC、Python的MySQLdb等库在程序中访问日志服务，在程序中处理查询结果。

## 数据示例

以共享单车日志为例，共享单车日志内容包括用户年龄、性别、电量使用量、车辆ID、操作延时、纬度、锁类型、经度、操作类型、操作结果和开锁方式。数据保存在名为project:trip\_demo的Project下，名为Logstore:ebike的Logstore中。Project所在地域是cn-hangzhou。

日志示例如下：

```
Time :10-12 14:26:44
__source__: 11.164.232.105
__topic__: v1
age: 55
battery: 118497.673842
bikeid: 36
gender: male
latency: 17
latitude: 30.2931185245
lock_type: smart_lock
longitude: 120.052840484
op: unlock
op_result: ok
open_lock: bluetooth
userid: 292
```

## JDBC统计

1. 创建一个Maven项目，在POM依赖中添加JDBC依赖，示例代码如下所示。



```
<dependency>
  <groupId>MySQL</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.6</version>
</dependency>
```

## 2. 使用Java程序通过JDBC查询日志数据，示例代码如下所示。

```
/**
 * Created by mayunlei on 2017/6/19.
 */
import com.mysql.jdbc.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.Statement;
/**
 * Created by mayunlei on 2017/6/15.
 */
public class jdbc {
    public static void main(String args[]){
        //配置您创建的Project名称，Logstore名称等相关信息。
        final String endpoint = "cn-hangzhou-intranet.sls.aliyuncs.com";//日志服务内网或VPC域名。
        final String port = "10005";//通过JDBC访问日志默认使用10005作为端口号。
        final String project = "trip-demo";
        final String logstore = "ebike";
        final String accessKeyId = "";
        final String accessKey = "";
        Connection conn = null;
        Statement stmt = null;
        try {
            //步骤1：加载JDBC驱动。
            Class.forName("com.mysql.jdbc.Driver");
            //步骤2：创建一个链接。
            conn = DriverManager.getConnection("jdbc:mysql://" + endpoint + ":" + port + "/" + project, access
            KeyId, accessKey);
            //步骤3：创建statement。
            stmt = conn.createStatement();
            //步骤4：定义查询语句，查询2017年10月11日全天日志中满足条件op = "unlock"的日志条数。
            String sql = "select count(1) as pv,avg(latency) as avg_latency from "+logstore+" " +
                "where __date__ >= '2017-10-11 00:00:00' " +
                " and __date__ < '2017-10-12 00:00:00' " +
                " order by avg_latency";
```

```
        and op = unlock";

//步骤5：执行查询条件。
ResultSet rs = stmt.executeQuery(sql);
//步骤6：提取查询结果。
while(rs.next()){
    //Retrieve by column name
    System.out.print("pv:");
    //获取结果中的pv
    System.out.print(rs.getLong("pv"));
    System.out.print(" ; avg_latency:");
    //获取结果中的avg_latency
    System.out.println(rs.getDouble("avg_latency"));
    System.out.println();
}
rs.close();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (SQLException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (stmt != null) {
        try {
            stmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if (conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}
```

## 2.5. 对接DataV

DataV是阿里云的可视化产品，能帮助您通过图形化的界面轻松搭建专业水准的可视化应用，丰富表现日志分析数据。本文档介绍如何通过日志服务对接DataV进行大屏数据展示。

### 前提条件

- 已采集日志数据，详情请参见[数据采集](#)。
- 已开启并配置索引，详情请参见[开启并配置索引](#)。

### 背景信息

实时大屏广泛应用于大型在线促销活动。实时大屏基于流式计算架构，该架构包含以下模块：

- 数据采集：将来自各源头数据实时采集。
- 中间存储：利用类Kafka Queue进行生产系统和消费系统解耦。
- 实时计算：数据处理关键环节，订阅实时数据，通过计算规则对窗口中数据进行运算。
- 结果存储：计算结果数据存入SQL和NoSQL。
- 可视化：通过API调用结果数据进行展示。

在阿里集团内，有大量成熟的产品可以完成此类工作，一般可供选型的产品如下：



日志服务支持通过日志服务查询分析API直接对接DataV进行大屏数据展示。



### 功能特点

计算方式根据数据量、实时性和业务需求会分为以下两种。

- 实时计算（流计算）：固定的计算+变化的数据
- 离线计算（数据仓库+离线计算）：变化的计算+固定的数据

在对实时性有要求的日志分析场景中，日志服务为您提供实时索引LogHub中数据机制，可通过LogSearch/Analytics直接进行查询分析。这种方式具有以下优势：

- 快速：一秒内查询（5个条件），可处理10亿级数据。一秒内分析（5个维度聚合+GroupBy），可聚合亿级别数据，无需等待和预计算结果。
- 实时：99.9%情况下可做到日志产生1秒内反馈到大屏。
- 动态：无论修改统计方法还是补数据，支持实时刷新显示结果，无需等待重新计算。

这种方式具有以下限制：

- 数据量：单次计算数据量限制为百亿行，当超过百亿行，需要限定时间段。
- 计算灵活度：计算限于SQL92语法，不支持自定义UDF。

### DataV配置步骤

1. 创建DataV数据源。
  - i. 登录[DataV数据可视化控制台](#)。

ii. 在我的数据页签，单击添加数据，具体参数配置如下：

配置项	说明
类型	在下拉菜单中选择简单日志服务 SLS。
自定义数据源名称	为数据源配置一个自定义名称。例如log_service_api。
AppKey	主账号的AccessKey ID，或者有权限读取日志服务的RAM用户 AccessKey ID。
AppSecret	主账号的AccessKey Secret，或者有权限读取日志服务的RAM用户AccessKey Secret。
Endpoint	填写日志服务Project所在地域的Endpoint，请参见 <a href="#">服务入口</a> 。

2. 打开可视化画布。

- 在我的可视化页签，选择已存在目标可视化项目，单击编辑。
- 在我的可视化页签，单击新建可视化，创建步骤请参见[使用模板创建PC端可视化应用](#)。

3. 创建折线图并添加过滤器。

- i. 创建一个折线图。在左侧组件列表中，单击折线图 > 基本折线图，在右侧数据页签数据源项单击配置数据源修改组件样式配置，具体配置如下：

配置项	说明
数据源类型	选择简单日志服务 SLS。
选择已有数据源	选择 <a href="#">步骤1</a> 创建的数据源log_service_api。
查询	<p>查询示例如下：</p> <pre>                     {                       "projectName": "dashboard-demo",                       "logStoreName": "access-log",                       "topic": "",                       "from": ":from",                       "to": ":to",                       "query": ".*  select approx_distinct(remote_addr) as uv ,count(1) as pv , date_format(from_unixtime(date_trunc('hour',_time_)) , '%Y/%m/%d %H:%i:%s') as time group by time order by time limit 1000" ,                       "line": 100,                       "offset": 0                     }                 </pre>

查询示例参数说明：

配置项	说明
projectName	Project名称。
logstoreName	Logstore名称。
topic	日志主题，如果您没有设置日志主题，此处请留空。
from、to	<p>from和to分别是日志的起始和结束时间。</p> <p><b>说明</b> 示例中填写的是 :from 和 :to 。在测试时，您可以先填写 unix time，例如1509897600。发布之后换成 :from 和 :to ，然后您可以在URL参数里控制这两个数值的具体时间范围。例如，预览时的URL是 http://datav.aliyun.com/screen/86312 ，打开 http://datav.aliyun.com/screen/86312?from=1510796077&amp;to=1510798877 后，会按照指定的时间进行计算。</p>
query	<p>查询条件。query的语法请参见<a href="#">实时分析简介</a>。</p> <p><b>说明</b> query中的时间格式，一定要是2017/07/11 12:00:00这种，所以采用以下方式把时间对齐到整点，再转化成目标格式。</p> <pre>date_format(from_unixtime(date_trunc('hour',__time__)), '%Y/%m/%d%H:%i:%s')</pre>
line	请填写默认值100。
offset	请填写默认值0。

- ii. 配置完成后，单击查看数据响应结果。
- iii. 新建过滤器。选中数据过滤器，然后单击添加过滤器后的加号 (+) 新建一个过滤器。

过滤器内容请按照以下格式填写。

```
return Object.keys(data).map((key) => {
  let d= data[key];
  d["pv"] = parseInt(d["pv"]);
  return d;
});
```

- 在过滤器中，要把y轴用到的结果显示为int类型，上述样例中，y轴为pv，所以需要转换pv列。
- 可以看到在结果中有t和pv两列，您可以将x轴配置为t，y轴配置成pv。

4. 创建饼状图并添加过滤器。

- i. 新建轮播饼图。在左侧组件列表中，单击选择饼图 > 轮播饼图，在右侧数据页签数据源项单击配置数据源修改组件样式配置，具体配置如下：

配置项	说明
数据源类型	选择简单日志服务 SLS。
选择已有数据源	选择步骤1创建的数据源 <i>log_service_api</i> 。
查询	<p>查询示例如下：</p> <pre>                     {                       "projectName": "dashboard-demo",                       "logStoreName": "access-log",                       "topic": "",                       "from": 1509897600,                       "to": 1509984000,                       "query": "*  select count(1) as pv ,method group by method" ,                       "line": 100,                       "offset": 0                     }                 </pre> <p>示例参数说明请参见折线图参数说明。</p>

- ii. 配置完成后，单击查看数据响应结果。
- iii. 添加一个过滤器。选中数据过滤器，然后单击添加过滤器后的加号 (+) 新建一个过滤器。  
过滤器内容请按照以下格式填写。

```

return Object.keys(data).map((key) => {
  let d= data[key];
  d["pv"] = parseInt(d["pv"]);
  return d;
})
    
```

饼图的type配置为method， value配置为pv。

- 5. 回调ID动态获取时间范围，以下步骤演示如何动态的显示15分钟的日志。

- i. 创建一个静态数据源，使用值默认即可，添加一个过滤器。代码示例如下：

```
return [{
  value:Math.round(Date.now()/1000)
}];
```

```
return [{
  value:Math.round((Date.now() - 24 * 60 * 60 * 1000)/1000)
}];
```

- ii. 在交互页签，选中启用对应的响应事件，并将value绑定到变量中。



- iii. 在数据视图，通过 `:from` 和 `:to` 引用回调ID，示例如下：

```
{
  "projectName": "dashboard-demo",
  "logStoreName": "access-log",
  "topic": "",
  "from": ":from",
  "to": ":to",
  "query": "*| select count(1) as pv ,referer group by pv desc limit 30",
  "line": 100,
  "offset": 0
}
```

## 6. 预览和发布。

- i. 单击预览图标对已编辑的视图进行预览。
- ii. 单击发布图标即完成视图发布。

## 案例：调整不同统计口径下的云栖大会网站访问实时大屏

云栖大会期间有一个临时需求，需要您统计大会网站的全国各地访问量以在实时大屏上显示。此前您已配置采集全量日志数据，并且在日志服务中打开了查询分析，所以只要输入查询分析Query即可。在此过程中，需求是不断调整的，如下所列：

- 原始需求：在云栖大会的第一天，您需要统计UV（当日点击用户数量）。

您要查询所有访问日志中nginx下forward字段的数据（该字段记录访问用户的一个或多个IP，每条日志一个forward字段），通过 `approx_distinct(forward)` 计算去重后的IP地址数量，获取从云栖大会首日零时到当前时刻的点击UV数，可以使用如下语句。

```
*| select approx_distinct(forward) as uv
```

- 需求第一次调整：云栖大会的第二天，需求调整为您需要统计yunqi.aliyun.com这个域名下的用户访问量数据。

您可以增加一个过滤条件host进行实时查询，使用如下语句。

```
host:yunqi.aliyun.com | select approx_distinct(forward) as uv
```

- 需求第二次调整：在统计过程中，您发现Nginx访问日志forward字段存在多个IP，您默认只要第一个IP。

使用如下语句。

```
host:yunqi.aliyun.com | select approx_distinct(split_part(forward,',',1)) as uv
```

- 需求第三次调整：云栖大会的第三天，需求被加上限制条件，您需要剔除通过UC浏览器访问并点击该浏览器广告而来的用户访问量，统计非UC浏览器广告导流、不重复IP的全国各地用户访问量。

此时您可以加上一个过滤条件not，使用如下语句。

```
host:yunqi.aliyun.com not URL:uc-iflow | select approx_distinct(split_part(forward,',',1)) as uv
```

## 2.6. 对接Grafana

本文主要通过Grafana示例演示如何将日志服务采集的Nginx日志进行实时可视化分析。

### 前提条件

- 已采集Nginx日志数据。详情请参见[Nginx模式](#)。
- 已开启并配置索引，详情请参见[分析Nginx访问日志](#)。

### 流程架构

日志从收集到分析的流程架构如下。



### 操作步骤

1. 安装Grafana。详情请参见[Grafana官方文档](#)。

以安装Ubuntu为例，需执行以下安装命令。

```
wget https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_4.5.2_amd64.deb
sudo apt-get install -y adduser libfontconfig
sudo dpkg -i grafana_4.5.2_amd64.deb
```

若您需要使用饼状图，则需执行以下命令安装Pie chart插件。

```
grafana-cli plugins install grafana-piechart-panel
```

2. 安装日志服务插件。


请确认Grafana的插件目录位置。在Ubuntu的插件目录/var/lib/grafana/plugins/安装插件，重启grafana-server。

以Ubuntu系统为例，执行以下命令安装插件，并重启grafana-server。




```
cd /var/lib/grafana/plugins/
git clone https://github.com/aliyun/aliyun-log-grafana-datasource-plugin
service grafana-server restart
```

### 3. 配置日志服务数据源。

 **说明** 若您是在本机部署，默认是安装在3000端口。请提前在浏览器打开3000端口。

i. 登录Grafana。

ii. 在左侧菜单栏，单击  图标 > Data Sources。

iii. 在Data Sources页签，单击Add data source，选中LogService，单击Select，请参考如下说明配置数据源。

配置项	说明
Name	Name表示名称，请自定义一个数据源的名称。
HTTP	<ul style="list-style-type: none"> <li>▪ URL示例：<code>http://dashboard-demo.cn-hangzhou.log.aliyuncs.com</code>。<code>dashboard-demo</code>是project名称，<code>cn-hangzhou.log.aliyuncs.com</code>是project所在地域的Endpoint，在配置自己的数据源时，需要替换成自己的project和region地址，请参见<a href="#">服务入口</a>。</li> <li>▪ Access可以选择Server，也可以选择Browser。</li> <li>▪ Whitelisted：添加访问白名单。</li> </ul>
Auth	采用默认配置。
log service details	日志服务详细配置，分别填写Project和Logstore，以及具备读取权限的AccessKey。AccessKey可以是主账号的AccessKey，也可以是子帐号的AccessKey。


iv. 单击Save & Test。

### 4. 添加Dashboard。

i. 在左侧导航栏，选中  图标 > Dashboards，添加一个Dashboard。

ii. 配置模板变量。

在Grafana中可以配置模板变量，在同一个视图中，通过选择不同的变量值，展示不同的视图。本文档主要配置每个时间区间的大小，以及不同域名的访问情况。

- a. 在New dashboard页面右上角，通过下拉框选择配置一个时间区间，单击页面右上角的  图标 > **Variables** > **Add variable**，进入配置页面。
- b. 变量的名称是您在配置中使用的变量，在这里起名为myinterval，在查询条件中，要写成 `$myinterval`，参考以下样例进行配置，并单击Add。

配置项	说明
Name	变量名称，您可以命名为myinterval。
Type	选择Interval。
Lable	输入time interval。
Internal Options	<ul style="list-style-type: none"> <li>▪ value输入 <code>1m,10m,30m,1h,6h,12h,1d,7d,14d,30d</code>。</li> <li>▪ 选中Auto Option。</li> </ul>

c. 配置一个域名模板。

通常一个VPS上可以挂载多个域名，则需要查看不同域名的访问情况。





在Variable页面，单击New，参考以下说明配置域名模板。

配置项	说明
Name	变量名称，您可以命名为hostname。
Type	选中Custom。
Lable	输入域名。
Custom Options	<p>Values separated by输入 <code>*,www.host.com,www.host0.com,www.host1.com</code>。表示可以查看所有域名的访问情况，也可以分别查看 <code>www.host.com</code>、<code>www.host0.com</code> 或 <code>www.host1.com</code> 的访问情况。</p>
Selection Options	配置为默认值。

- d. 在左侧菜单栏，单击Save保存配置完成的模板。

iii. 视图配置示例。

■ 配置PV & UV

- a. 单击右上角  图标。
- b. 在New Panel页面，单击Add Query。
- c. 单击  图标，在Visualization下拉列表选中Graph，创建一个Graph视图。
- d. 单击  图标，在Title文本框输入UV & PV。
- e. 单击  图标，在Query下拉列表选中Logservice，配置项请参见如下说明。

配置项	说明
Query	<pre>\$hostname  select approx_distinct(remote_addr) as uv ,count(1) as pv , __time__ - __time__ % \$\$myinterval as time group by time order by time limit 1000</pre> <p>上述Query中的\$hostname，在实际展示时，会替换成用户选择的域名。\$\$myinterval，则会替换成时间区间，注意myinterval前有两个\$符号，而hostname有一个。</p>
X-Column	time
Y-Column	uv,pv

若UV和PV的值相差较大，可通过调整为双Y轴图表展示。单击图表下方的图示线条（如下图①），单击Y-Axis（如下图②），打开Use right y-axis开关（如下图③）。

■ 配置出入网带宽


添加出入网带宽的视图，创建步骤请参见配置PV & UV。

主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select sum(body_byte_sent) as net_out, sum(request_length) as net_in, __time__ - __time__ % \$\$myinterval as time group by __time__ - __time__ % \$\$myinterval limit 10000</pre>
X-Column	Time
Y-Column	net_in,net_out

■ 不同HTTP方法的占比

添加不同HTTP方法的占比的视图，创建步骤请参见配置PV & UV。

单击  图标，在Visualization下拉列表选中Pie Chart，创建一个饼图。主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select count(1) as pv ,method group by method</pre>
X-Column	pie
Y-Column	method,pv

■ 不同HTTP状态码占比


添加不同HTTP状态码占比的视图，创建步骤请参见配置PV & UV。

单击  图标，Visualization选中Pie Chart，创建一个饼图。主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select count(1) as pv ,status group by status</pre>
X-Column	pie
Y-Column	status,pv

■ 热门来源页面


添加热门来源页面的视图，创建步骤请参见配置PV & UV。

单击  图标，在Visualization下拉列表选中Pie Chart，创建一个饼图。主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select count(1) as pv , referer group by referer order by pv desc</pre>
X-Column	pie
Y-Column	referer,pv

### ■ 延时最高页面


添加延时最高页面的视图，创建步骤请参见[配置PV & UV](#)。

单击  图标，在Visualization下拉列表选中Table，创建一个表格，展示URL和对应的延时。主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select URL as top_latency_URL ,request_time order by request_time desc limit 10</pre>
X-Column	X-Column不填写内容
Y-Column	top_latency_url,request_time

### ■ 热门页面


添加热门页面的视图，创建步骤请参见[配置PV & UV](#)。

单击  图标，在Visualization下拉列表选中Table，创建一个表格。主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select count(1) as pv, split_part(URL,'?',1) as path group by split_part(URL,'?',1) order by pv desc limit 20</pre>
X-Column	X-Column不填写内容
Y-Column	path,pv

### ■ 非200请求top页面

添加非200请求top页面的视图，创建步骤请参见[配置PV & UV](#)。

单击  图标，在Visualization下拉列表选中Table，创建一个表格。主要配置项如下：

配置项	说明
Query	<pre>\$hostname not status:200  select count(1) as pv , url group by url order by pv desc</pre>
X-Column	X-Column不填写内容
Y-Column	url,pv

■ 前后端平均延时


添加前后端平均延时的视图，创建步骤请参见[配置PV & UV](#)。


主要配置如下：

配置项	说明
Query	<pre>\$hostname   select avg(request_time) as response_time, avg( upstream_response_time) as upstream_response_time, __time__ - __time__ % \$\$myinterval as time group by __time__ - __time__ % \$\$myinterval limit 10000</pre>
X-Column	time
Y-Column	upstream_response_time,response_time

■ 设置Logs


添加Logs视图，创建步骤请参见[配置PV & UV](#)。

单击  图标，在Visualization下拉列表选中Logs，创建一个Logs。配置页面输入Logs Per Page即完成。

 **说明** 每页最多展示100条，即Logs Per Page最大值为100。

■ 客户端统计

添加客户端统计的视图，创建步骤请参见[配置PV & UV](#)。

单击  图标，在Visualization下拉列表选中Pie Chart，创建一个饼图。主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select count(1) as pv, case when regexp_like(http _user_agent , 'okhttp') then 'okhttp' when regexp_like(http_use r_agent , 'iPhone') then 'iPhone' when regexp_like(http_user_a gent , 'Android') then 'Android' else 'unKnown' end as http_use r_agent group by http_user_agent order by pv desc limit 10</pre>
X-Column	pie
Y-Column	http_user_agent,pv

■ 设置流图


添加流图，创建步骤请参见配置PV & UV。

主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select to_unixtime(time) as time,status,count from (select time_series(__time__, '1m', '%Y-%m-%d %H:%i', '0') as time,status,count(*) as count from log group by status,time order by time limit 10000)</pre>
X-Column	配置为时间列。
Y-Column	col1#:#col2，其中col1为聚合列，col2为其他列。

■ 设置地图

添加地图，创建步骤请参见配置PV & UV。

单击  图标，在Visualization下拉列表选中Worldmap Panel，创建一个地图。

Queries主要配置项如下：

配置项	说明
Query	<pre>\$hostname   select count(1) as pv ,geohash(ip_to_geo(arbitrary(remote_addr))) as geo,ip_to_country(remote_addr) as country from log group by country having geo &lt;&gt;" limit 1000</pre>
X-Column	map
Y-Column	country,geo,pv

Visualization主要配置项如下，其他配置为默认值。

配置项	说明
Location Data	geohash
Location Name Field	country
geo_point/geohash Field	geo
Metric Field	pv

iv. 单击右上角  图标，保存Dashboard。

#### 5. 查看结果。

打开Dashboard首页查看效果。示例请参见[Demo](#)。

您可以在Dashboard页面上方选择统计的时间范围，也可以选择统计的时间粒度或不同的域名。整个Nginx访问统计的Dashboard完成，您可以从视图挖掘有价值的信息。