ALIBABA CLOUD

阿里云

日志服务

开发指南

文档版本：20220518

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**。
粗体	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**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.可视化开发

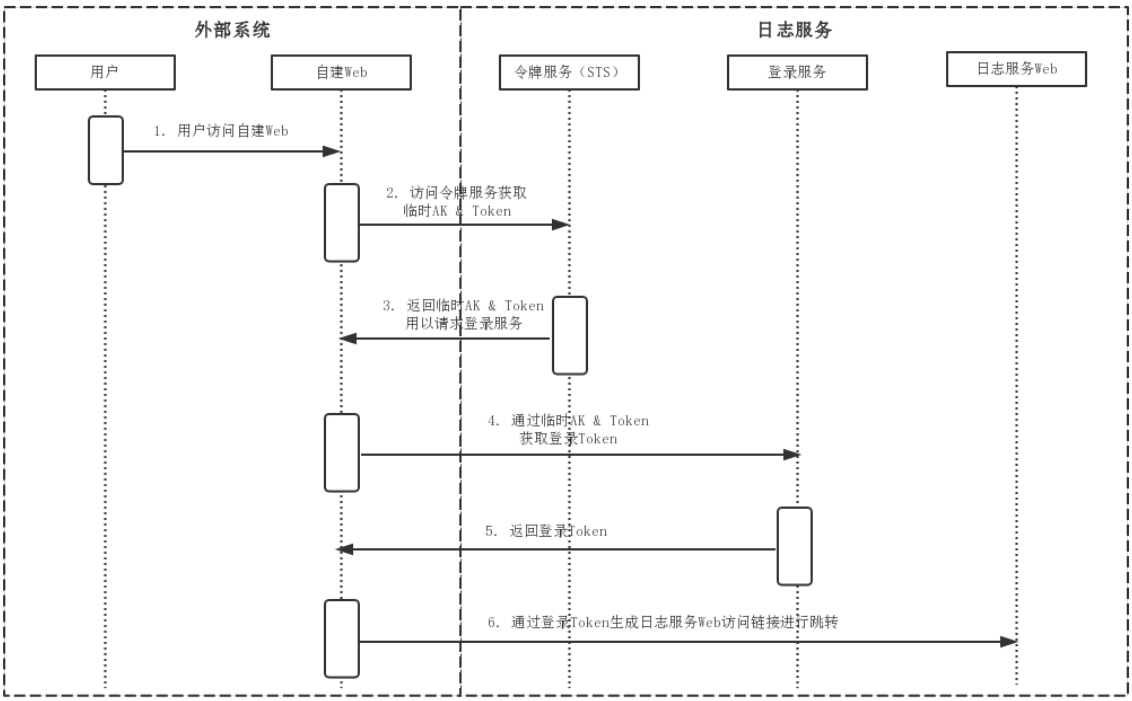
1.1. 控制台内嵌及分享

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，访问后可以通过步骤二：为RAM用户授权限制操作权限，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信息和安全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用户及授权和创建可信实体为阿里云账号的RAM角色及授权。
2. 调用STS SDK服务接口获取临时访问凭证STS AK与SecurityToken。

   > **说明** STS返回的SecurityToken中可能包含特殊字符，请将特殊字符进行URL编码后再输入。

   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
```

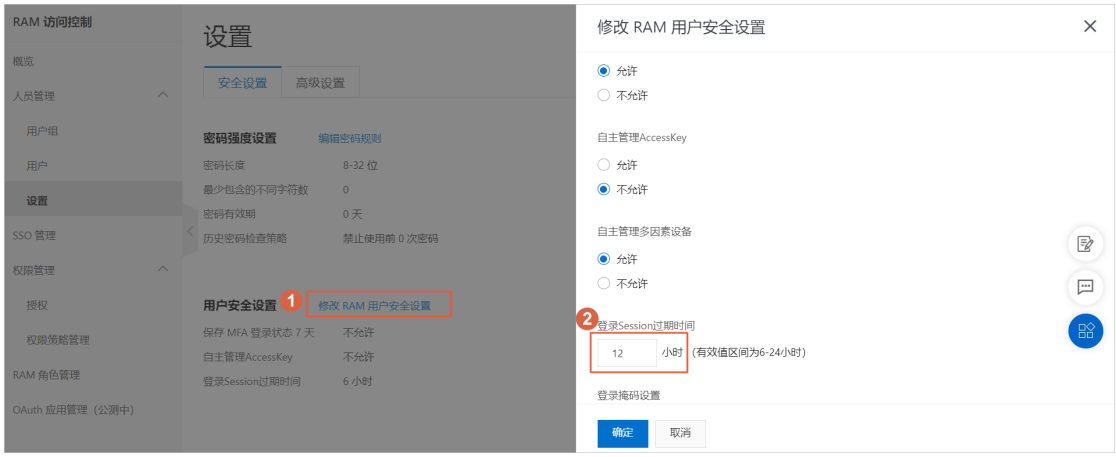
3. 调整SecurityToken有效期。

   SecurityToken默认有效期为15分钟~60分钟，您可以参见本步骤调整SecurityToken的最大有效期为12小时。

   i. 在RAM控制台，修改目标RAM角色的最大会话时间为12小时。

   ii. 在RAM控制台，修改目标RAM用户的登录Session过期时间为12小时。

   iii. 修改免登录代码中的setDurationSeconds字段的值为43200L。

   此处以Java代码为例。

```
AssumeRoleRequest assumeRoleReq = new AssumeRoleRequest();
assumeRoleReq.setRoleArn(roleArn);
assumeRoleReq.setRoleSessionName(roleSession);
assumeRoleReq.setMethod(MethodType.POST);
assumeRoleReq.setDurationSeconds(43200L);
```

4. 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/lognext/project/<Project名称>/logsearch/<日志库名称>?hideTopbar=true&hideSidebar=true
```

- 查询页面：

```
https://sls4service.console.aliyun.com/lognext/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/lognext/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encodeURL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
- Go

1.2. 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的Web页面，支持您快速方便地对日志进行查询和分析。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。通过免登方式嵌入自建Web页面的具体操作，请参见控制台内嵌及分享。

URL格式

所有UI参数均通过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*${ProjectName}*、*${LogstoreName}*、*${savedsearchID}*和*${dashboardID}*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，可将界面设置为黑色主题。

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
hiddenBack	boolean	否	隐藏控制台首页返回按钮。	hiddenBack=true
hiddenChangeProject	boolean	否	隐藏切换Project功能。	hiddenChangeProject=true
hiddenOverview	boolean	否	隐藏Project概览入口。	hiddenOverview=true
ignoreTabLocalStorage	boolean	否	关闭Tab访问的历史记录。	ignoreTabLocalStorage=true

参数名	类型	是否必选	说明	示例
queryTimeType	long	否	指定查询和分析的时间范围。更多信息，请参见queryTimeType指定查询的时间范围说明。取值范围如下： • 1~26：指定查询和分析的时间范围为数字对应的区间。 • -2：自定义（相对）。此时必须配置start和end。例如：start:-10m,end:now。 • -3：自定义（整点）。此时必须配置start和end。例如：start:-2h,end:absolute。 • 99：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。	queryTimeType=1
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

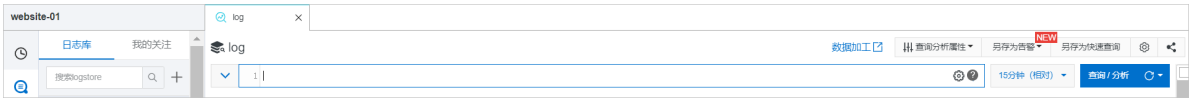
queryTimeType	代表含义
1	1分钟（相对）
2	15分钟（相对）
3	1小时（相对）
4	4小时（相对）
5	1天（相对）
6	1周（相对）
7	30天（相对）
8	1分钟（整点）
9	15分钟（整点）
10	1小时（整点）
11	4小时（整点）
12	1天（整点）
13	1周（整点）
14	30天（整点）

queryTimeType	代表含义
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）
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：

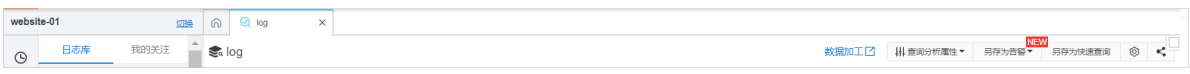
- 使用如下URL隐藏控制台首页返回按钮、Project切换按钮、Project概览入口。

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



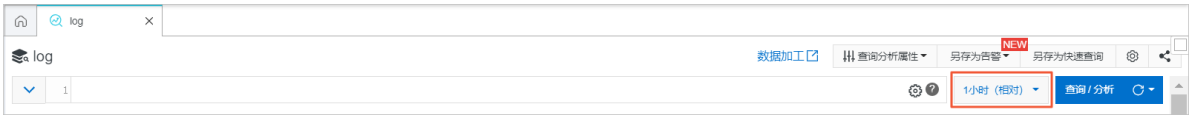
- 使用如下URL隐藏控制台首页返回按钮。

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



- 使用如下URL设置查询和分析时间。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}?queryTimeType=3
```



Logstore查询和分析页面相关参数

使用Logstore查询和分析页面相关参数设置Logstore查询和分析页面。

参数名	类型	是否必选	说明	示例
ProjectName	string	是	Project名称。	website-01
LogstoreName	string	是	Logstore名称。	logstore01
queryString	string	否	使用BASE 64编码后的查询和分析语句。 例如 `*|select count(*)` 语句经过BASE 64编码后为 `KnxzZWxlY3QgY291bnQoKik=`。	KnxzZWxlY3QgY291bnQoKik=
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	隐藏数据加工按钮。	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
hiddenPSql	boolean	否	隐藏SQL增强按钮。	hiddenPSql=true

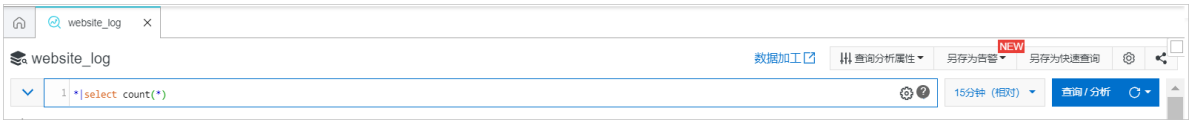
参数名	类型	是否必选	说明	示例
keyDispalyMode	string	否	内容列显示。 • single：整行。 • multi：换行。	keyDispalyMode=single

URL参数及效果示例如下所示：

- 使用如下URL设置查询和分析语句。

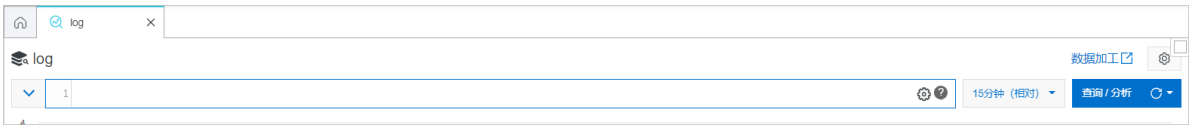
  例如 `*|select count(*)` 语句经过BASE 64编码后为 `KnxzZWxlY3QgY291bnQoKik=` 。

  ```
  https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}?encode=base64&queryString=KnxzZWxlY3QgY291bnQoKik=
  ```



- 使用如下URL隐藏编辑、修改相关的按钮，例如**查询分析属性**、**另存为告警**等。

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



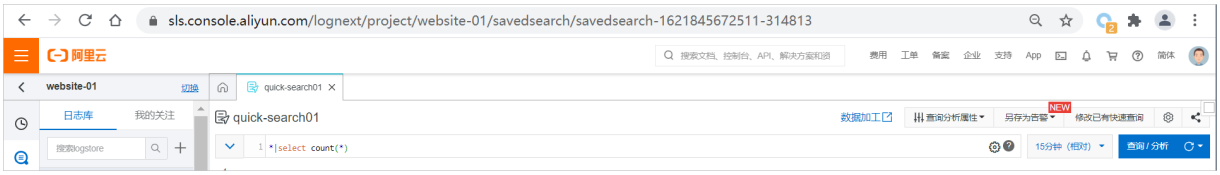
Logstore快速查询页面相关参数

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。	website-01
savedSearchName	string	否	快速查询名称。	quick-search01
savedsearchID	string	是	快速查询ID。 **说明** 您在创建快速查询后，可以在URL中获取快速查询ID。更多信息，请参见获取快速查询ID。	savedsearch-1621845672511-314813

URL参数及效果示例如下所示：

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



仪表盘相关参数

使用仪表盘相关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。	website-01
dashboardName	string	否	仪表盘名称。	网站分析日志
dashboardID	string	是	仪表盘ID。 说明 您在创建仪表盘后，可以在URL中获取仪表盘ID。更多信息，请参见接口说明。	dashboard-1609817292009-742588
filters	string	否	过滤条件，需要使用encodeURIComponent()函数转码。 例如 `filters=key1:value1&filters=key2:value2` 经过转码后为 `filters=key1%3Avalue1%26filters%3Dkey2%3Avalue2` 。	filters=key1%3Avalue1%26filters%3Dkey2%3Avalue2
token	JsonString	否	变量替换，需要使用encodeURIComponent()函数转码。 例如 `token=[{"key": "projectname","value":"1"},{"key": "region", "value": "hangzhou"}]` 经过转码后为 `token=%5B%7B%22key%22%3A%20%22projectname%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22key%22%3A%20%22region%22%2C%20%22value%22%3A%20%22hangzhou%22%7D%5D`	token=%5B%7B%22key%22%3A%20%22projectname%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22key%22%3A%20%22region%22%2C%20%22value%22%3A%20%22hangzhou%22%7D%5D

参数名	类型	是否必选	说明	示例
readOnly	boolean	否	隐藏仪表盘页面中的编辑、设置相关的按钮，例如**编辑**、**告警**等。	readOnly=true
hiddenFilter	boolean	否	隐藏过滤条件。	hiddenFilter=true
hiddenToken	boolean	否	隐藏变量替换。	hiddenToken=true
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
hiddenPSql	boolean	否	隐藏SQL增强按钮。	hiddenPSql=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

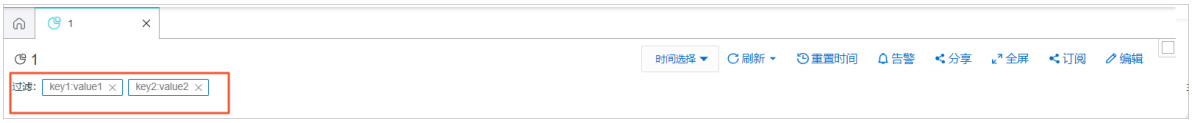
URL参数及效果示例如下所示：

- 使用如下URL使仪表盘页面为只读模式。

```
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。其中，过滤条件`filters=key1:value1&filters=key2:value2`经过转码后为`filters%3Dkey1%3Avalue1%26filters%3Dkey2%3Avalue2`。

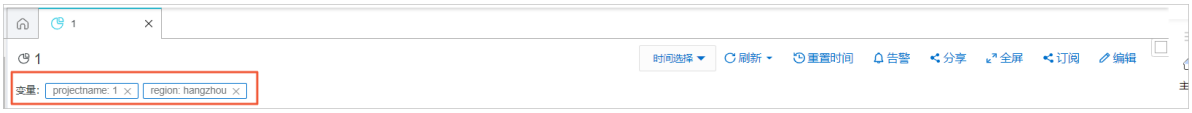
```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboardID}?filters%3Dkey1%3Avalue1%26filters%3Dkey2%3Avalue2
```



- 使用如下URL增加多个变量替换条件projectname=1和region=hangzhou。其中`token=[{"key": "proje`

ctname","value":"1"}, {"key": "region", "value": "hangzhou"}] 经过转码后为 token=%5B%7B%22key%22%3A%20%22projectname%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22key%22%3A%20%22region%22%2C%20%22value%22%3A%20%22hangzhou%22%7D%5D

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboardID}?token=%5B%7B%22key%22%3A%20%22projectname%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22key%22%3A%20%22region%22%2C%20%22value%22%3A%20%22hangzhou%22%7D%5D
```



- 使用如下URL使仪表盘每5分钟刷新一次。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboardId}?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。如果设置treeEncode为base64，则需先使用BASE64完成转码。 例如 {"logstore":{"expand":true,"resourceList":["delete-log"],"template":["savedsearch","alert"]}} 经过BASE64转码后为 eyJsb2dzdG9yZSI6eyJleHBhbmQiOnRydWUsInJlc291cmNlTGlzdCI6WyJkZWxldGUtbG9nIl0sInRlbXBsYXRlIjpbInNhdmVkc2VhcmNoIiwiYWxlcnQiXX19 。	eyJsb2dzdG9yZSI6eyJleHBhbmQiOnRydWUsInJlc291cmNlTGlzdCI6WyJkZWxldGUtbG9nIl0sInRlbXBsYXRlIjpbInNhdmVkc2VhcmNoIiwiYWxlcnQiXX19
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": {
        "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	用于控制Logstore列表中的资源入口。
template	string[]	否	用于控制Logstore列表中的一些功能入口。更多信息，请参见template参数说明。
machineGroup	object	否	用于控制机器组列表再的资源入口。
savedSearch	object	否	用于控制快速查询列表中的资源入口。
alert	object	否	用于控制告警列表中的资源入口。
dashboard	object	否	用于控制仪表盘列表中的资源入口。
etl	object	否	用于控制数据加工列表中的资源入口。

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。

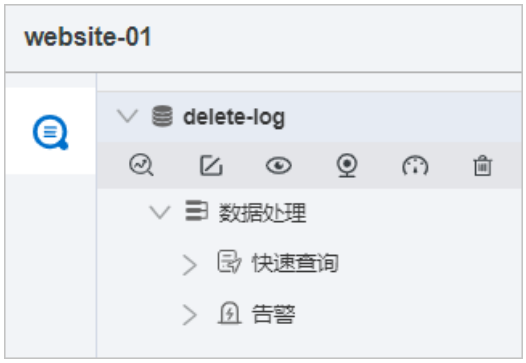
template参数说明

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

URL参数及效果示例如下所示：

使用如下URL设置左侧导航栏。

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}?treeconfig=eyJsb2dzdG9yZSI6eyJleHBhbmQiOnRydWUsInJlc291cmNlTGlzdCI6WyJkZWxldGUtbG9nIl0sInRlbXBsYXRlIjpbInNhdmVkc2VhcmNoIiwiYWxlcnQiXX19&hiddenBack=true&hiddenChangeProject=true&hiddenOverview=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。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/${projectNam
e}/dashboard/${dashboardName}?product=${productCode}">
</body>
</html>
```

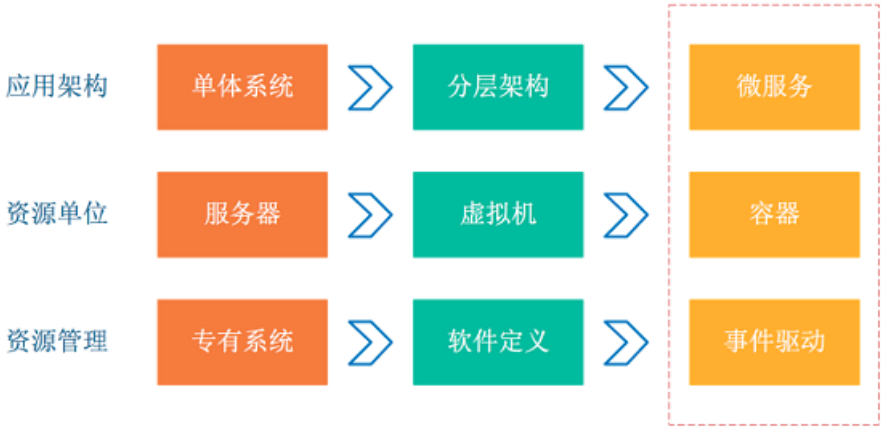
1.3. 对接Jaeger

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。更多信息，请参见XTrace。

Logging，Metrics和Tracing的特点

- Logging用于记录离散的事件

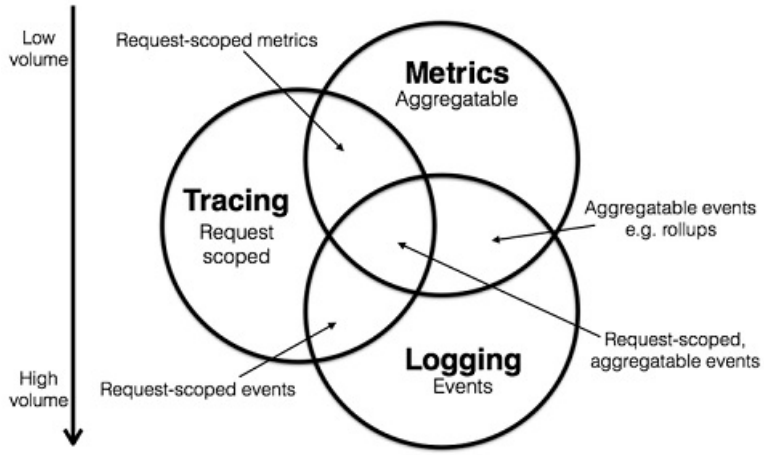
  例如，应用程序的调试信息或错误信息，Logging是我们诊断问题的依据。

- 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

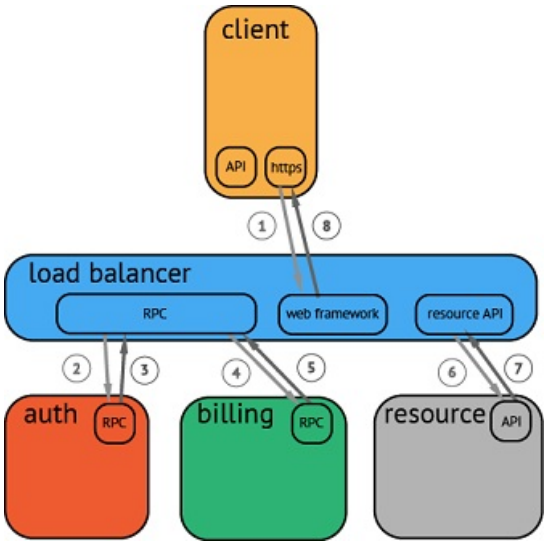
流行的Tracing软件有：

- 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）

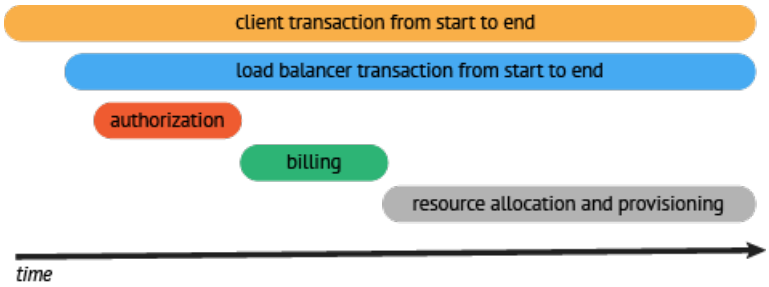
分布式追踪系统发展快，种类多，核心步骤一般有三个：

- 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。

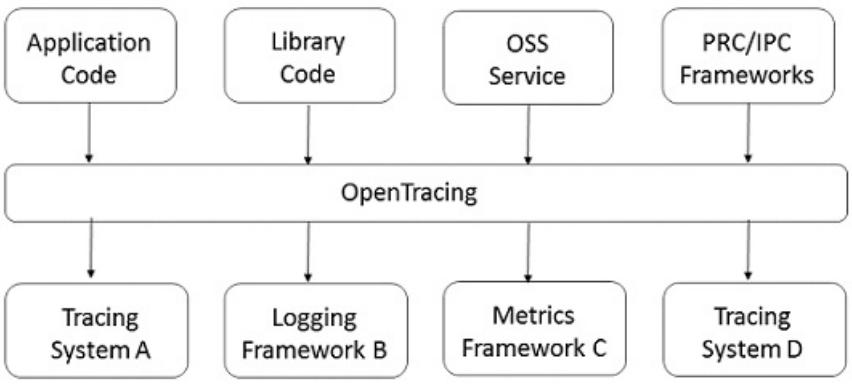


数据被采集存储后，分布式追踪系统一般会选择使用包含时间轴的时序图来呈现这个调用链。但在数据采集过程中，由于需要侵入用户代码，并且不同系统的API并不兼容，导致如果切换追踪系统，会有较大改动。



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。更多信息，请参见OpenTracing。



- 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。

```
单个 Trace 中，span 间的因果关系
        [Span A]  ←←←(the root span)
            |
     +------+------+
     |             |
 [Span B]      [Span C] ←←←(Span C是Span A的孩子节点，ChildOf)
     |             |
 [Span D]      +---+-------+
               |           |
           [Span E]    [Span F] >>> [Span G] >>> [Span H]
                                          ↑
                                          ↑
                                          ↑
                         (Span G在Span F后被调用，FollowsFrom)
```

  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：

```
单个Trace中，span间的时间关系
--|-------|-------|-------|-------|-------|-------|-------|-> time
 [Span A···················································]
   [Span B·················································]
      [Span D··············································]
    [Span C················································]
         [Span E·······]        [Span F··] [Span G··] [Span H··]
```

  每个Span包含以下对象：

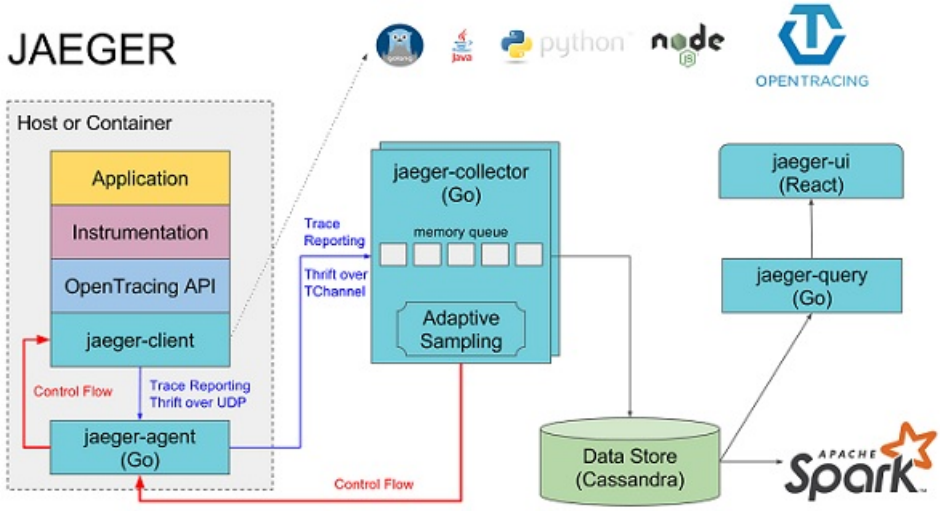
  - An operation name：操作名称。
  - A start timestamp：起始时间。
  - A finish timestamp：结束时间。
  - Span Tag：一组键值对构成的Span标签集合。键值对中，键必须为string，值可以是字符串、布尔、或者数字类型。
  - Span Log：一组span的日志集合。每次log操作包含一个键值对，以及一个时间戳。键值对中，键类型必须为string，值可以是任意类型。不是所有的支持OpenTracing的Tracer，都需要支持所有的值类型。
  - SpanContext：Span上下文对象。每个SpanContext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。更多信息，请参见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。

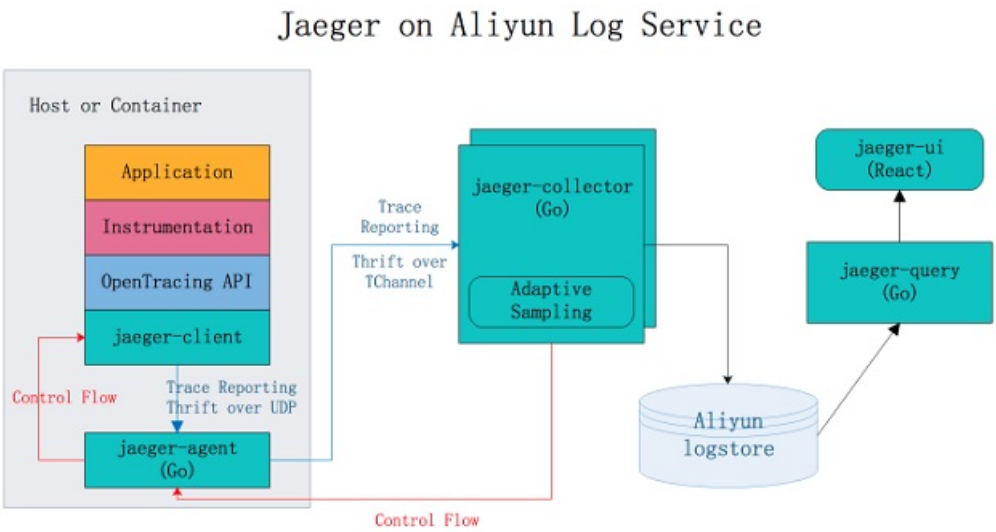


如上图所示，Jaeger主要由以下几部分组成。

- 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Jeager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。 更多信息，请参见Jaeger on Alibaba Cloud Log Service。



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%。更多信息，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。

Jaeger配置示例

HotROD是由多个微服务组成的应用程序，它使用了OpenTracing API记录Trace信息。

下面通过一段视频向您展示如何使用Jaeger on Alibaba Cloud Log Service诊断HotROD出现的问题。视频包含以下内容：

1. 配置日志服务
2. 通过docker-compose运行Jaeger。
3. 运行HotROD。
4. 通过Jaeger UI检索特定的Trace。
5. 通过Jaeger UI查看Trace的详细信息。
6. 通过Jaeger UI定位应用的性能瓶颈。
7. 通过日志服务管理控制台，定位应用的性能瓶颈。
8. 应用程序调用OpenTracing API。

示例中使用的query语句及其含义如下：

- 以分钟为单位统计frontend服务的HTTP GET /dispatch操作的平均延迟以及请求个数。

```
process.serviceName: "frontend" and operationName: "HTTP GET /dispatch" |
select from_unixtime( __time__ - __time__ % 60) as time,
truncate(avg(duration)/1000/1000) as avg_duration_ms,
count(1) as count
group by __time__ - __time__ % 60 order by time desc limit 60
```

- 比较两条Trace各个操作的耗时。

```
TraceID: "Trace1" or TraceID: "Trace2" |
select operationName,
(max(duration)-min(duration))/1000/1000 as duration_diff_ms
group by operationName
order by duration_diff_ms desc
```

- 统计延迟大于1.5 s的Trace的IP情况。

```
process.serviceName: "frontend" and operationName: "HTTP GET /dispatch" and duration > 15
00000000 |
select "process.tags.ip" as IP,
truncate(avg(duration)/1000/1000) as avg_duration_ms,
count(1) as count
group by "process.tags.ip"
```

1.4. 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。更多信息，请参见配置索引。

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Grafana通过MySQL协议连接日志服务。具体操作，请参见通过JDBC协议分析日志。
- 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。

日志示例

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__: 192.168.0.0
__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。

   在where条件中必须包含__date__字段或__time__字段来限制查询的时间范围。__date__字段是timestamp类型，__time__字段是bigint类型。例如：

   - __date__ > '2017-08-07 00:00:00' and __date__ < '2017-08-08 00:00:00'
   - __time__ > 1502691923 and __time__ < 1502692923

   > 注意 使用Java的JDBC、Python的MySQLdb等库在程序中访问日志服务时，服务入口必须使用日志服务的经典网络或VPC网络访问域名，否则会出现连接超时报错 `com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure, Caused by: java.net.ConnectException: Connection timed out: connect` 。

```
/**
* 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[]){
     final String endpoint = "trip-demo.cn-hangzhou-intranet.log.aliyuncs.com"; //包括Project名称和日志服务经典网络或VPC网络访问域名，请根据实际情况替换。更多信息，请参见服务入口。
```

```
        final String port = "10005"; //通过JDBC访问时，默认使用10005端口。
        final String project = "trip-demo"; //Project名称。
        final String logstore = "ebike"; //Logstore名称。
        final String accessKeyId = "";  //阿里云访问密钥AccessKey ID。更多信息，请参见访问密钥
。
        final String accessKey = "";  //阿里云访问密钥AccessKey Secret。
        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+"/"+proje
ct,accessKeyId,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 "+logstor
e+"  " +
                    "where     __date__  >=  '2017-10-11 00:00:00'   " +
                    "     and  __date__  <   '2017-10-12 00:00:00'" +
                    " 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();
```

```
                    e.printStackTrace();
                }
            }
        }
    }
}
```

1.5. 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

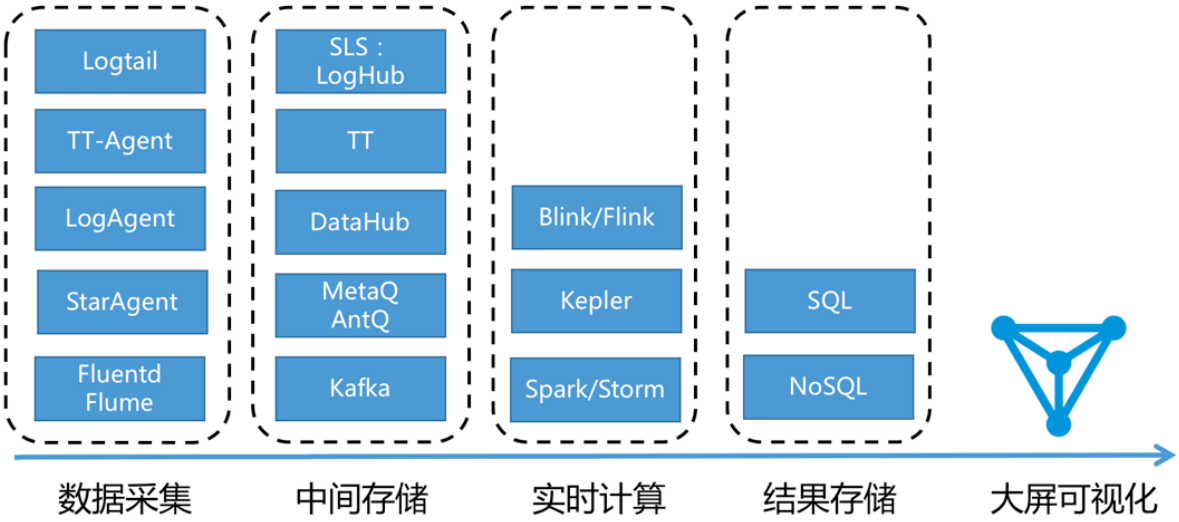
- 已采集日志数据，详情请参见数据采集。
- 已开启并配置索引，详情请参见配置索引。

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/Anlaytics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，请参见服务入口。

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	选择步骤1创建的数据源*log_service_api*。
查询	查询示例如下： `{ "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 }`

   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	from和to分别是日志的起始和结束时间。 **说明** 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&to=1510798877` 后，会按照指定的时间进行计算。
query	查询条件。query的语法请参见分析概述。 **说明** 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。 `date_format(from_unixtime(date_trunc('hour',__time__)) ,'%Y/%m/%d%H:%i:%s')`
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*log_service_api*。
查询	查询示例如下： `{ "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 }` 示例参数说明请参见折线图参数说明。

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
}
```

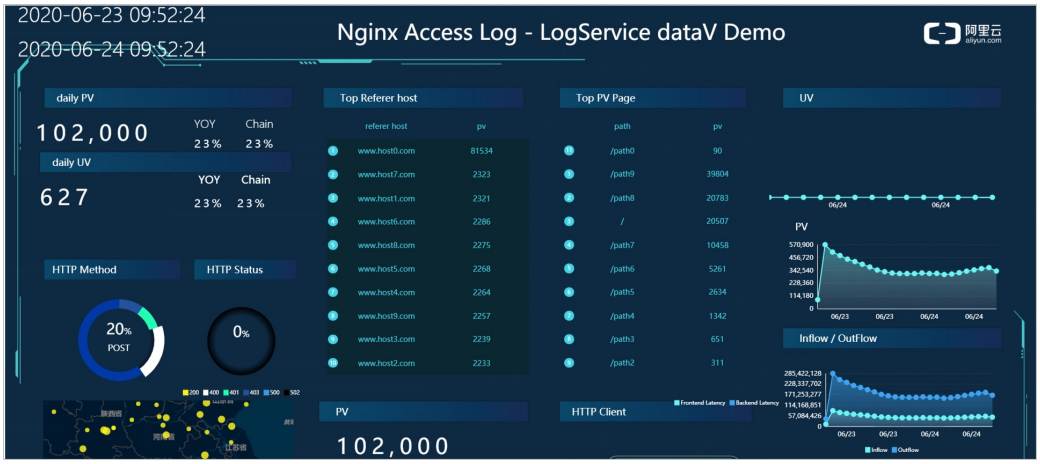
6. 预览和发布。

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。

试用：Demo地址。URL中的参数from和to，您可以随意切换为任意时间。

实时大屏



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。

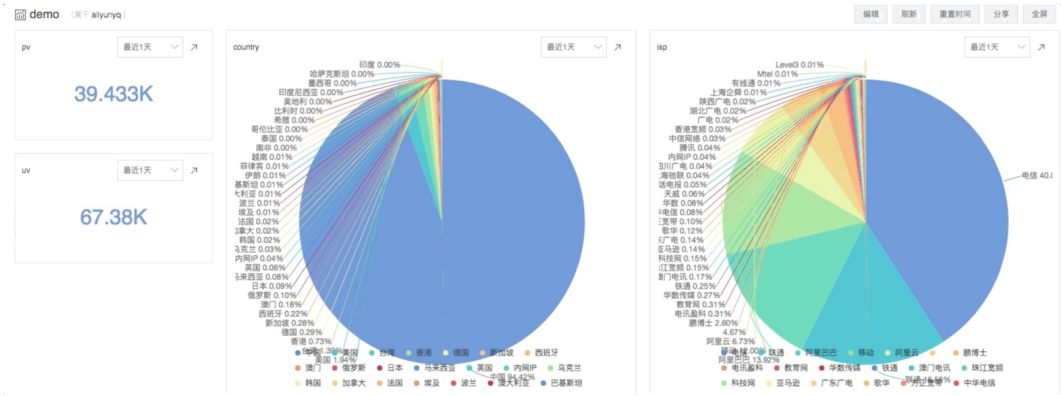
  ```
  host:yunqi.aliyun.com | select approx_distinct(split_part(forward,',',1)) as uv
  ```

- 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
  ```

  示例



大屏效果如下：

大屏效果

1.6. 对接Grafana

本文介绍如何通过Grafana可视化分析日志服务所采集到的Nginx日志。

前提条件

- 已采集Nginx日志数据。更多信息，请参见使用Nginx模式采集日志。
- 已开启并配置索引。更多信息，请参见分析Nginx访问日志。
- 下载数据源插件项目压缩包。

  下载命令为 `wget https://github.com/aliyun/aliyun-log-grafana-datasource-plugin/archive/refs/heads/master.zip` 。

  > **说明** 本文以2.9版本的日志服务插件为例。

- 已安装Grafana。具体操作，请参见Grafana官方文档。

> 说明
> - 本文以8.0.6版本的Grafana为例。
> - 如果您是在本机上安装Grafana，请提前在浏览器中打开3000端口。
> - 如果您需要使用饼图，需执行如下命令安装Pie Chart插件。
>
> ```
> grafana-cli plugins install grafana-piechart-panel
> ```

插件兼容性

Grafana和日志服务插件的兼容关系如下表所示：

Grafana版本	日志服务插件版本
≥8.0.0	2.x
<8.0.0	1.0

步骤一：安装日志服务插件

为Grafana安装日志服务插件的操作步骤如下：

1. 执行以下命令将日志服务插件项目压缩包解压到Grafana插件目录。
   - 使用YUM或RPM安装的Grafana：
     ```
     unzip aliyun-log-grafana-datasource-plugin-master.zip -d /var/lib/grafana/plugins
     ```
   - 使用*.tar.gz*文件安装的Grafana：

     *{PATH_TO}*为Grafana的安装路径。
     ```
     unzip aliyun-log-grafana-datasource-plugin-master.zip -d {PATH_TO}/grafana-8.0.6/data
     /plugins
     ```
2. 修改Grafana配置文件。
   i. 打开配置文件。
      - 使用YUM或RPM安装的Grafana：*/etc/grafana/grafana.ini*
      - 使用*.tar.gz*文件安装的Grafana：*{PATH_TO}/grafana-8.0.6/conf/defaults.ini*

   ii. 在配置文件的[plugins] 节点中，设置allow_loading_unsigned_plugins参数。
   ```
   allow_loading_unsigned_plugins = aliyun-log-service-datasource
   ```
3. 重启Grafana。
   i. 使用kill命令终止Grafana进程。

ii. 执行以下命令启动Grafana。

- 使用YUM或RPM安装的Grafana：

```
systemctl restart grafana-server
```

- 使用*.tar.gz*文件安装的Grafana：

```
./bin/grafana-server web
```

步骤二：添加数据源

为Grafana添加日志服务的数据源的操作步骤如下：

1. 登录Grafana。
2. 在左侧菜单栏，选择  > Data Sources。
3. 在Data Sources页签，单击Add data source。
4. 在Add data source页面，单击LogService对应的Select。
5. 配置数据源。

   重要参数说明如下表所示。

参数	说明
Name	数据源的名称。
Default	打开Default开关。
Endpoint	Project的服务入口，例如 `http://cn-qingdao.log.aliyuncs.com` 。请根据实际情况替换服务入口。更多信息，请参见服务入口。
Project	Project的名称。
Logstore	Logstore的名称。
AccessKeyId	阿里云AccessKey ID，用于标识用户。为保证账号安全，建议您使用RAM用户的AccessKey。如何获取AccessKey，请参见访问密钥。
AccessKeySecret	阿里云AccessKey Secret，用于验证用户的密钥。为保证账号安全，建议您使用RAM用户的AccessKey。如何获取AccessKey，请参见访问密钥。

6. 单击Save & Test。

步骤三：添加仪表盘

为Grafana添加仪表盘的操作步骤如下：

1. 在左侧导航栏，选择  > Dashboards。
2. 在New Panel面板中，单击Choose Visualization。
3. 配置模板变量。

   您可以在Grafana中配置模板变量，实现在同一个图表中通过选择不同的变量值，展示不同的结果。

i. 配置时间区间大小的模板变量。

a. 在New dashboard页面右上角，配置时间区间，然后单击  图标。

b. 单击Variables。

c. 单击Add variable。

d. 按照如下参数配置模板变量，然后单击Add。

重要参数说明如下表所示。

参数	说明
Name	变量名称，例如myinterval。该名称是您配置中使用的变量，此处为myinterval，则查询条件中需写成 `$$myinterval` 。
Type	选择Interval。
Label	配置为time interval。
Values	配置为1m,10m,30m,1h,6h,12h,1d,7d,14d,30d。
Auto Option	打开Auto Option开关，其他参数保持默认配置。

ii. 配置域名的模板变量。

a. 在Variables页面，单击New。

b. 按照如下参数配置模板变量，然后单击Add。

参数	说明
Name	变量名称，例如hostname。该名称是您配置中使用的变量，此处为hostname，则查询条件中需写成 `$hostname` 。
Type	选择Custom。
Label	输入域名。
Custom Options	配置为 `*,example.com,example.org,example.net` ，表示可以查看所有域名的访问情况，也可以分别查看 `example.com` 、 `example.org` 或 `example.net` 的访问情况。
Selection Options	保持默认配置。

iii. 在左侧菜单栏，单击Save。

4. 添加可视化图表。

- 用于展示PV&UV的图表（Graph）

a. 单击右上角  图标。

b. 在New Panel页面，单击Add Query。

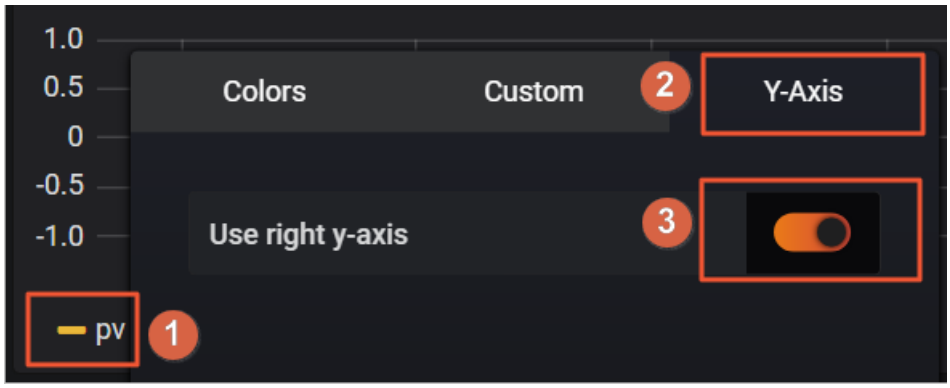
c. 单击  图标，在Visualization下拉列表中选择Graph。

d. 单击  图标，在Title文本框输入UV&PV。

e. 单击  图标，在Query下拉列表中选择Logservice，并完成如下配置。

参数	说明
Query	查询和分析语句示例如下： `$hostname| select approx_distinct(remote_addr) as uv ,count(1) as pv , __time__ - __time__ % $$myinterval as time group by time order by time limit 1000` 在展示结果中，`$hostname`会被替换成您选择的域名。`$$myinterval`会被替换成您选择的时间区间。 注意 myinterval前有2个美元符号（$$），hostname前只有1个美元符号（$）。
X-Column	配置为time，格式为秒级的Unix时间戳。
Y-Column	配置为uv,pv。

f. 如果UV和PV的值相差较大，可配置双Y轴图表。



g. 单击右上角图标，根据页面提示完成保存。

- 用于展示流入流出流量的图表（Graph）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下表所示。

参数	说明
Query	查询和分析语句示例如下： `$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` 在展示结果中，`$hostname`会被替换成您选择的域名。`$$myinterval`会被替换成您选择的时间区间。 注意 myinterval前有2个美元符号（$$），hostname前只有1个美元符号（$）。
X-Column	配置为**time**，格式为秒级的Unix时间戳。
Y-Column	配置为**net_in,net_out**。

- 用于展示HTTP请求方法占比的图表（Pie Chart）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下表所示。

参数	说明
Query	查询和分析语句示例如下： `$hostname \| select count(1) as pv ,method group by method` 在展示结果中，`$hostname`会被替换成您选择的域名。
X-Column	配置为**pie**。
Y-Column	配置为**method,pv**。

- 用于展示HTTP请求状态码占比的图表（Pie Chart）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下表所示。

参数	说明
Query	查询和分析语句示例如下： `$hostname \| select count(1) as pv ,status group by status` 在展示结果中，`$hostname`会被替换成您选择的域名。

参数	说明
X-Column	配置为**pie**。
Y-Column	配置为**status,pv**。

- 用于展示热门访问来源的图表（Pie Chart）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下表所示。

参数	说明
Query	查询和分析语句示例如下： `$hostname \| select count(1) as pv , referer group by referer order by pv desc` 在展示结果中，`$hostname`会被替换成您选择的域名。
X-Column	配置为**pie**。
Y-Column	配置为**referer,pv**。

- 用于展示延时最高页面的图表（Table）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下表所示。

参数	说明
Query	查询和分析语句示例如下： `$hostname \| select URL as top_latency_URL ,request_time order by request_time desc limit 10` 在展示结果中，`$hostname`会被替换成您选择的域名。
X-Column	无需配置。
Y-Column	配置为**top_latency_url,request_time**。

- 用于展示热门页面的图表（Table）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下表所示。

配置项	说明

配置项	说明
Query	查询和分析语句示例如下： `$hostname \| select count(1) as pv, split_part(URL,'?',1) as path group by split_part(URL,'?',1) order by pv desc limit 20` 在展示结果中，`$hostname`会被替换成您选择的域名。
X-Column	无需配置。
Y-Column	配置为**path,pv**。

- 用于展示非200请求的Top页面图表（Table）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下表所示。

配置项	说明
Query	查询和分析语句示例如下： `$hostname not status:200\| select count(1) as pv , url group by url order by pv desc` 在展示结果中，`$hostname`会被替换成您选择的域名。
X-Column	无需配置。
Y-Column	配置为**url,pv**。

- 用于展示平均延时的图表（Singlestat）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下：

配置项	说明
Query	查询和分析语句示例如下： `$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` 在展示结果中，`$hostname`会被替换成您选择的域名。`$$myinterval`会被替换成您选择的时间区间。 注意 myinterval前有2个美元符号（$$），hostname前只有1个美元符号（$）。
X-Column	配置为**time**。

配置项	说明
Y-Column	配置为**upstream_response_time,response_time**。

- 用于展示详细日志的图表（Logs）

  添加步骤请参见添加PV&UV信息图表。

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

- 用于展示客户端统计的图表（Pie Chart）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下：

配置项	说明
Query	查询和分析语句示例如下： `$hostname \| select count(1) as pv, case when regexp_like(http_user_agent , 'okhttp') then 'okhttp' when regexp_like(http_user_agent , 'iPhone') then 'iPhone' when regexp_like(http_user_agent , 'Android') then 'Android' else 'unKnown' end as http_user_agent group by http_user_agent order by pv desc limit 10` 在展示结果中，`$hostname`会被替换成您选择的域名。
X-Column	配置为**pie**。
Y-Column	配置为**http_user_agent,pv**。

- 用于展示1分钟内各个HTTP请求状态数量的图表（Graph）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下所示。

配置项	说明
Query	查询和分析语句示例如下： `$hostname \| select to_unixtime(time) as time,status,count from (select time_series(__time__, '1m', '%Y-%m-%d %H:%i', '0') as time,status,count(*) as count from log group by status,time order by time limit 10000)` 在展示结果中，`$hostname`会被替换成您选择的域名。
X-Column	配置为时间列。
Y-Column	配置为*col1*#:#*col2*。其中*col1*为聚合列，*col2*为其他列。

- 用于展示来源IP分布的图表（Worldmap Panel）

  添加步骤请参见添加PV&UV信息图表，相关参数说明如下。

参数	说明
Query	查询和分析语句示例如下： `$hostname \| select count(1) as pv ,geohash(ip_to_geo(arbitrary(remote_addr))) as geo,ip_to_country(remote_addr) as country from log group by country having geo <>'' limit 1000` 在展示结果中，`$hostname`会被替换成您选择的域名。
X-Column	配置为**map**。
Y-Column	配置为**country,geo,pv**。

参数	说明
Location Data	配置为**geohash**。
Location Name Field	配置为**country**。
geo_point/geohash Field	配置为**geo**。
Metric Field	配置为**pv**。

5. 查看结果。

   您可以在Dashboard页面上方选择统计的时间范围，还可以筛选time interval和hostname。配置示例请参见Demo。

常见问题

- Grafana日志保存在哪里？

  Grafana日志保存在如下文件中：

  - macOS系统：*/usr/local/var/log/grafana*
  - Linux系统：*/var/log/grafana*

- 如果日志中提示**aliyun-log-plugin_linux_amd64: permission denied**，怎么处理？

  请授予插件目录下的*dist/aliyun-log-plugin_linux_amd64*目录执行权限。

1.7. 使用Cloud Toolkit

Alibaba Cloud Toolkit是集开发、测试、运维、诊断、部署为一体的免费本地IDE插件。Alibaba Cloud Toolkit支持在IDE内查询与分析日志服务的数据。

前提条件

- 已开通日志服务。更多信息，请参见开通日志服务。
- 已创建并获取AccessKey。更多信息，请参见访问密钥。

阿里云账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用RAM用户进行API访问或日常运维。RAM用户需具备操作日志服务资源的权限。具体操作，请参见为RAM用户授权。

- 已在IntelliJ IDEA中安装并配置Alibaba Cloud Toolkit。更多信息，请参见在IntelliJ IDEA中安装和配置Cloud Toolkit。

操作步骤

以IntelliJ IDEA为例，介绍Alibaba Cloud Toolkit中日志服务的使用方法。更多信息，请参见Alibaba Cloud Toolkit使用指引。

1. 打开IntelliJ IDEA。
2. 在IntelliJ IDEA顶部菜单栏，选择**Tools > Alibaba Cloud > Preferences...**。
3. 在**Setting**对话框左侧，选择**Alibaba Cloud Toolkit > Accounts**，输入**Access Key ID**和**Access Key Secret**，单击**OK**。
4. 在IntelliJ IDEA底部页签栏，单击**Alibaba Cloud View**。
5. 在弹出面板中，单击**Alibaba Cloud SLS**。
6. 配置日志服务对应的Region和Project。
   工具自动加载Project中的Logstore列表。
7. 单击目标Logstore的**Action**列的查看。

   即可进入目标Logstore的查询和分析页面。查询和分析操作与日志服务控制台的操作相似。更多信息，请参见查询和分析日志。

更多操作

在**Alibaba Cloud SLS**页签，支持以下操作。

操作位置	支持操作	描述
左侧工具栏	在编辑器中打开	在编辑器中打开已加载的全部日志。
	实时刷新	定时查询日志，查询间隔默认为5秒。
	过滤内部字段	日志输出中过滤掉`_`开头的字段。
	回到顶部	回到表格顶部。
	滚至底部	滚动表格至底部并加载更多。
表格项右键	查看上下文	查看当前日志前后若干时间内的日志。支持前后10秒钟、前后1分钟、前后5分钟和前后10分钟。
	在编辑器中打开	在编辑器中打开当前日志。
表格右侧工具列	在编辑器中打开	在编辑器中打开当前日志。

2.使用Cloud Shell

2.1. Cloud Shell概述

阿里云云命令行Cloud Shell已经集成日志服务CLI，您无需部署配置，可以直接通过Cloud Shell管理日志服务资源和下载日志服务数据。

什么是Cloud Shell?

Cloud Shell允许您通过命令行管理阿里云资源。通过浏览器启动Cloud Shell时，Cloud Shell自动为您分配一台Linux管理机，并预装CLI、Terraform等多种云管理工具和ssh、vim、jq等系统工具，供您免费使用。更多信息，请参见什么是云命令行。

目前Cloud Shell已集成日志服务CLI，您无需下载并安装日志服务CLI工具，就可以使用Cloud Shell管理日志服务资源、下载日志数据。

为了避免资源浪费，当Cloud Shell窗口处于30分钟无输入后，将释放为您分配的虚拟机。如果您需要长时间使用Cloud Shell功能，可以下载并部署日志服务CLI。更多信息，请参见使用日志服务CLI。

使用场景

场景	描述
使用Cloud Shell管理日志服务资源	使用Cloud Shell创建Project、Logstore，查询日志等各种操作。
使用Cloud Shell下载日志数据	使用Cloud Shell下载日志数据。

2.2. 使用Cloud Shell管理日志服务资源

Cloud Shell集成了以命令行方式管理日志服务资源和数据的CLI工具。您可以通过Cloud Shell统一的命令格式，管理您在日志服务中的资源。

背景信息

您可以通过使用本教程来管理您日志服务中的如下资源：

- 管理项目（Project）。例如：创建、查询或删除Project等。
- 管理日志库（Logstore）。例如：创建、查询、修改或删除Logstore等。
- 管理消费组、Shard、机器组、投递任务等。例如：创建投递任务等。

步骤一：启动Cloud Shell

1. 登录日志服务控制台。
2. 在页面右上角，单击  图标。
3. 按照界面向导，完成验证。
4. 如果您需要持久化保存常用脚本和文件，在**挂载存储空间**对话框，单击**创建并绑定**。

   关联并挂载一个NAS文件系统可以持久化存储您的常用脚本和文件，否则您的文件会随NAS文件系统释放而销毁。Cloud Shell会为您创建性能型按量付费的NAS文件系统，这可能会产生少量的NAS使用费用。费用详情，请参见通用型NAS计费说明。

> **说明** 第一次连接云命令行时，Cloud Shell会为您自动创建虚拟机，会消耗最长不超过30秒时间。打开多个云命令行窗口时，所有窗口都会连接到同一台虚拟机。虚拟机数量不会因您打开新的命令行窗口而增加。

启动成功后，出现云命令行运行窗口。

```
Welcome to Alibaba Cloud Shell!
Type "help" to learn about Cloud Shell
Type "aliyun" to use Alibaba Cloud CLI
You may be interested in these tutorials below.
--------------------------  |  --------------------------
SLS 日志下载                 |  cloudshell://tutorial/sls-download-log
使用 Aliyun CLI 来管理云资源 |  cloudshell://tutorial/aliyun-cli
For more tutorials, visit https://api.aliyun.com/#/lab
shell@Alicloud:~$
```

步骤二：使用CLI命令管理日志服务资源

Cloud Shell已内置日志服务CLI，您无需配置日志服务CLI，可以直接使用Cloud Shell来管理您的日志数据。

支持的日志服务CLI子命令，包括创建Project、Logstore等。更多信息，请参见日志服务CLI。

- 语法格式

```
aliyunlog log <subcommand> [parameters | global options]
```

  - subcommand：支持的子命令。更多信息，请参见子命令列表。
  - parameters：支持的参数，每个子命令支持的参数列表请参见具体子命令描述。
  - global options：支持的全局参数。更多信息，请参见全局参数。

> **说明**
>
> - 操作非默认地域下日志服务资源时，需要切换地域。您可以通过 `--region-endpoint` 指定地域。
> - 目前Cloud Shell位于上海地域，如果当前Logstore不在上海地域，下载日志会产生一定的公网流量费用。价格详情请参见产品定价。

- 示例
  - 创建一个Project。

```
aliyunlog log create_project --project_name="project-a" --project_des="project created from cli"
```

  - 创建一个Logstore。

```
aliyunlog log create_logstore --project_name="aliyun-test-project" --logstore_name="logstore-a"
```

  - 查询日志。

```
aliyunlog log get_log_all --project="aliyun-test-project" --logstore="logstore-a" --from_time="2021-05-28 15:33:00+8:00" --to_time="2021-05-28 15:40:00+8:00" --query="level:Information|select event_id as Key1,COUNT(*) as Key2 group by Key1" --reverse=true
```

（可选）步骤三：查看帮助

您可以通过如下命令获取某一项操作的详细信息。

```
aliyunlog log [subcommand] help
```

例如，执行如下命令，可以查看list_project的使用语法。

```
aliyunlog log list_project help
```

更多参考

- 使用云命令行
- 使用日志服务CLI

2.3. 使用Cloud Shell下载日志数据

通过Cloud Shell可以快速下载较大数量的日志文件，并快速上传至OSS。本文介绍使用Cloud Shell下载日志数据。

操作步骤

1. 登录日志服务控制台。
2. 在Project列表区域，单击目标Project。
3. 在**日志存储 > 日志库**页签中，单击目标Logstore。
4. 在查询和分析框中，输入查询语句，选择时间范围，单击**查找/分析**。
5. 在**原始日志**页签中，单击  图标。
6. 在**日志下载**对话框中，选择**通过Cloud Shell下载**，然后单击**确定**。

   > **说明** 目前Cloud Shell位于上海地域，如果当前Logstore不在上海地域，下载日志会产生一定的公网流量费用。价格详情请参见产品定价。

7. 在**下载文件**对话框， 单击**确认**。
8. （可选）在Cloud Shell窗口，将下载的日志数据上传到OSS。
   i. 查看当前可以操作的OSS Bucket列表，以便确认OSS保存路径。

   ```
   aliyun oss ls
   ```

   执行完成后，返回可以操作的OSS Bucket列表。

   ```
   CreationTime                              Region    StorageClass    BucketName
   2021-09-24 02:03:08 +0000 UTC        oss-cn-beijing        Standard    oss://fyytse
   t
   2021-09-26 03:27:10 +0000 UTC       oss-cn-hangzhou        Standard    oss://demo
   Bucket Number is: 2
   ```

ii. 进入*aliyunlog*目录，执行 `ll` 找到下载的日志文件。

```
total 8640
drwxr-xr-x 2 shell shell    4096 Sep 28 10:21 ./
drwxr-xr-x 6 shell shell    4096 Sep 28 10:24 ../
-rw-r--r-- 1 shell shell 3198090 Sep 28 10:03 data_20210928_100331.txt
```

iii. 将日志文件上传至OSS。

```
aliyun oss cp data_20210928_100331.txt oss://demo --region cn-hangzhou
```

返回如下信息表示上传成功。

```
Succeed: Total num: 1, size: 3,198,090. OK num: 1(upload 1 files).
average speed 12792000(byte/s)
0.250823(s) elapsed
```