ALIBABA CLOUD

阿里云

阿里云Elasticsearch
Beats采集中心

文档版本：20200819

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.安装采集器

安装采集器

本文以Filebeat为例，介绍安装和管理采集器（Beats）的方法。通过采集器，您可以收集云服务器ECS（Elastic Compute Service）中的日志文件、网络数据、服务器指标等数据，发送到阿里云Elasticsearch或Logstash中，进行监控、分析等操作。

安装采集器

前提条件

您已完成以下操作：

- 创建阿里云Elasticsearch实例或Logstash实例。

  具体操作步骤请参见创建阿里云Elasticsearch实例和创建阿里云Logstash实例。

- 开启Elasticsearch实例的自动创建索引功能。

  出于安全考虑，阿里云Elasticsearch默认不允许**自动创建索引**。但是Beats目前依赖该功能，因此如果采集器Output选择为**Elasticsearch**，需要开启**自动创建索引**功能，具体操作步骤请参见开启自动创建索引。

- 创建ECS实例，且该ECS实例与Elasticsearch实例或Logstash实例处于同一专有网络VPC（Virtual Private Cloud）下。

  具体操作步骤请参见使用向导创建实例。

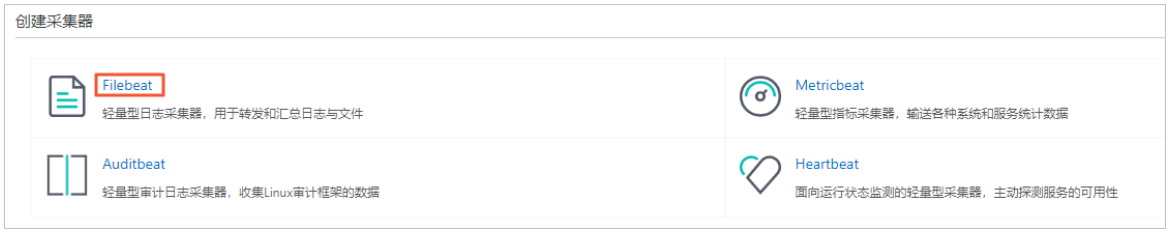
  > **注意**
  > - Beats默认安装目录为*/opt/aliyunbeats/*。安装后，ECS上会生成*conf*、*logs*和*data*这3个目录，分别映射了配置文件、Beats日志文件和Beats数据文件。建议不要删除或修改这3个文件中的内容，否则可能出现异常或者导致数据不正确。当出现问题时，您可以在*logs*目录下查看Beats日志来定位问题。
  > - Beats目前仅支持Aliyun Linux，RedHat和CentOS这三种操作系统。

- 在目标ECS实例上安装云助手和Docker服务。

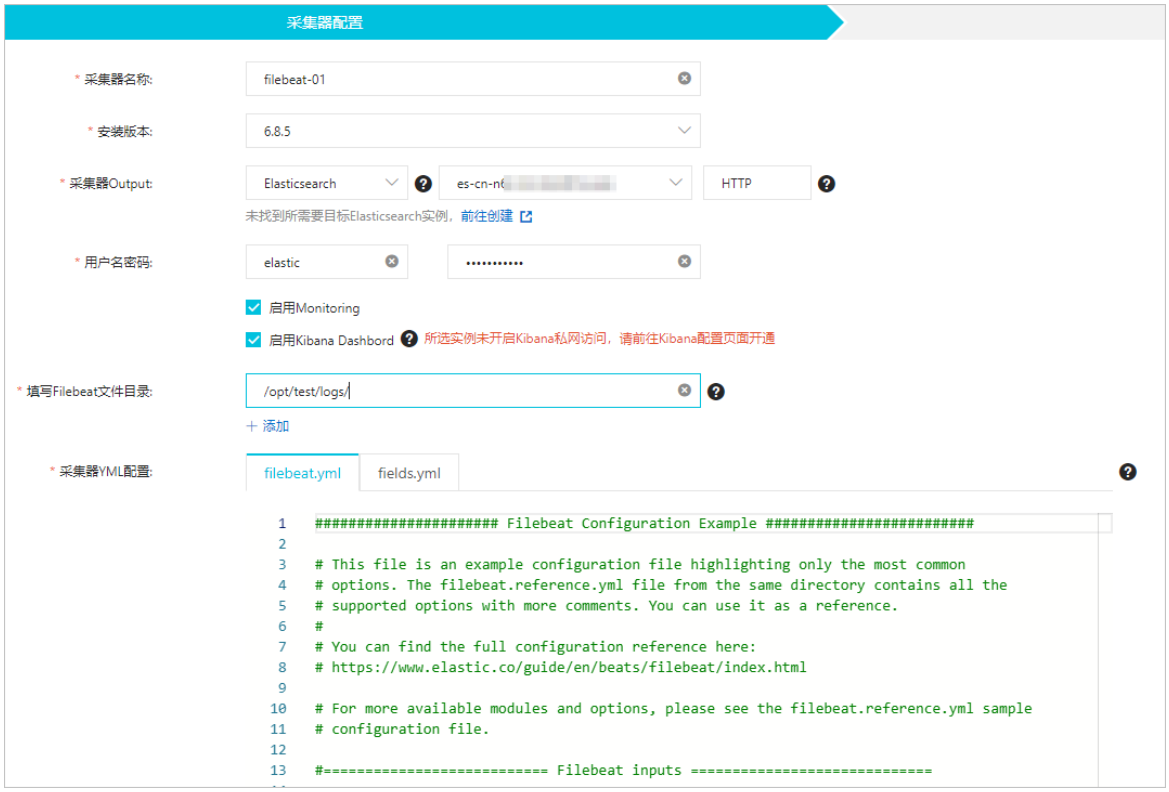
  具体操作步骤请参见安装云助手客户端、部署并使用Docker。

操作步骤

1. 登录阿里云Elasticsearch控制台，在左侧导航栏，单击**Beats数据采集中心**。
2. 在**创建采集器**区域，单击**Filebeat**。



3. 在**采集器配置**中，输入或选择采集器信息。



参数	说明
采集器名称	自定义输入采集器的名称。长度为1~30个字符，以大小写字母开头，可以包含字母、数字、下划线（_）或连字符（-）。
安装版本	目前Filebeat只支持6.8.5版本。
采集器Output	Filebeat采集内容的输出地址。直接关联已经创建的阿里云Elasticsearch或Logstash实例。访问协议需要与所选Elasticsearch实例保持一致。
用户名密码	如果采集器Output选择Elasticsearch，需要提供对应的用户名和密码，使Filebeat有权限向Elasticsearch实例中写入数据。
启用Monitoring	用来监控Filebeat的相关指标。如果采集器Output选择Elasticsearch，Monitor默认使用和采集器Output相同的阿里云Elasticsearch实例。如果采集器Output选择Logstash，则需要在配置文件中进行额外配置。
启用Kibana Dashboard	用来配置默认的Kibana Dashboard。由于阿里云Kibana配置在VPC内，因此需要在Kibana配置页面开通Kibana私网访问功能，详情请参见配置Kibana公网或私网访问白名单。
填写Filebeat文件目录	Filebeat专有的配置项。由于阿里云采用Docker部署Beats，因此需要将您希望采集的目录映射到Docker内，建议与*filebeat.yml*中的 `input.path` 保持一致。
采集器YML配置	采集器配置文件。请根据业务需求，修改采集器YML文件的配置，详情请参见采集器YML配置。

> **注意** 指定采集器Output后，不需要再在采集器YML配置中单独设置Output，否则会提示ECS采集器安装错误。

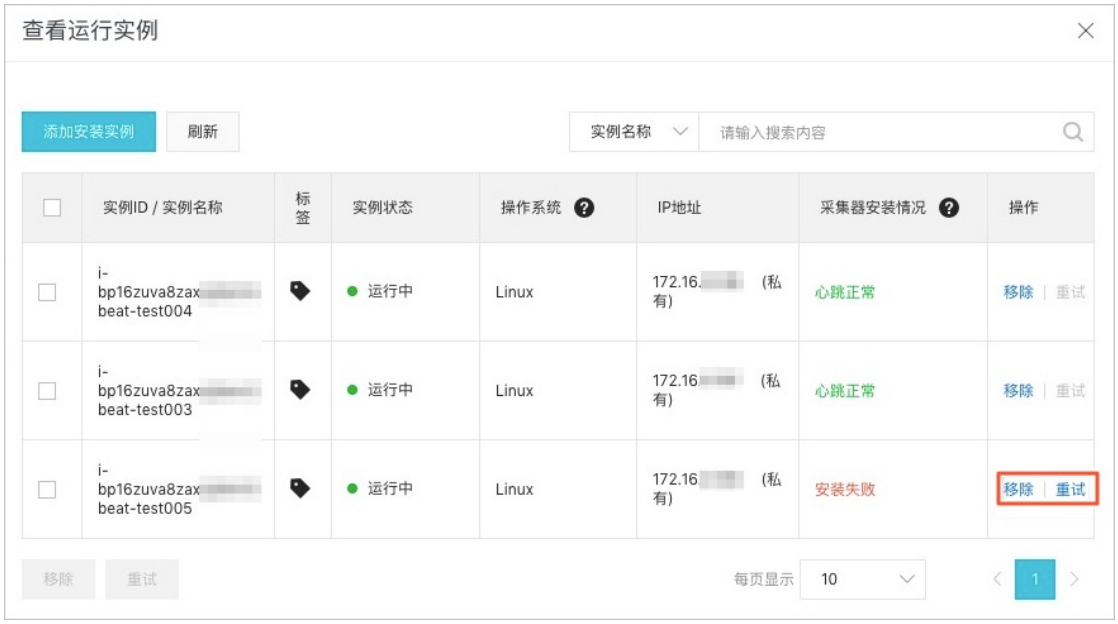
4. 单击**下一步**。
5. 在**采集器安装**向导中，选择需要操作的ECS实例。

> **注意** 采集器安装实例列表中会显示当前账号下，所有和**采集器Output**所选的Elasticsearch实例或Logstash实例处于同一个VPC下的ECS，并且只有已经安装云助手及Docker服务的ECS才能安装采集器。

6. 单击**启动**。
7. 在**启动成功**对话框中，单击**前往采集中心查看**，返回**Beats数据采集中心**页面查看创建成功的采集器。等待**采集器状态**变为**已生效**，说明采集器创建成功。**已生效**中的两个数字分别表示安装成功的ECS数和目标ECS数，如果ECS实例生效成功，则两边数字相等。

8. 查看运行实例。采集器创建成功后，您可以查看运行实例，判断采集器是否安装成功，并根据提示处理异常情况。
   i. 在**采集器管理**区域，单击对应采集器右侧**操作**列下的**查看运行实例**。

ii. 在查看运行实例页面，查看采集器安装情况。采集器安装情况分为3种：心跳正常、心跳异常、安装失败。



> **注意** 在心跳异常或者安装失败的情况下，您可以选择移除或者重试问题节点。如果重试失败，请参见Beats安装失败的排查与解决方法进行排查。

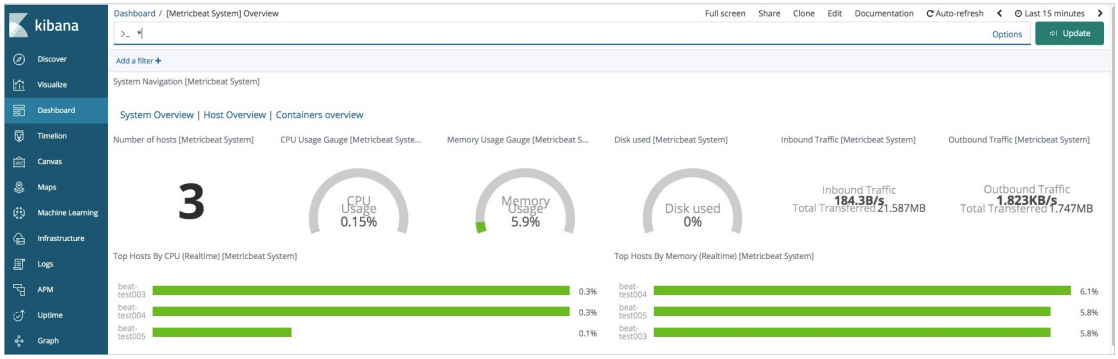
iii. 单击添加安装实例，可继续添加需要安装同样配置和类型的采集器的ECS。

9. （可选）查看Monitoring或Dashboard。如果您在创建采集器时勾选了启用Monitoring或启用Kibana Dashboard，那么Beats启动后，可以在对应Elasticsearch实例的Kibana中查看Monitor信息或Dashboard图表。

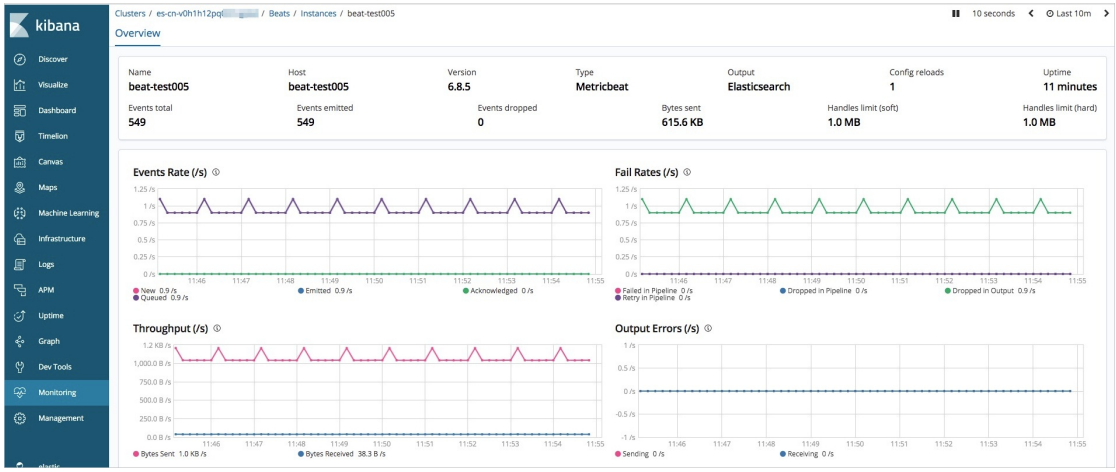
   i. 在采集器管理区域，单击对应采集器右侧操作列下的更多 > 查看Dashboard。

   ii. 在Kibana控制台登录页面，输入用户名和密码，单击登录。

   iii. 在左侧导航栏，单击Dashboard，再单击对应指标，查看该指标的Dashboard图表。



iv. 在左侧导航栏，单击Monitoring，再单击对应监控项，查看该监控项的Monitoring信息。



2.修改采集器配置

修改采集器配置

采集器安装完成后，您可以通过采集器配置功能，修改采集器的配置信息。

配置采集器

前提条件

已完成采集器的安装，详情请参见安装采集器。

操作步骤

1. 登录阿里云Elasticsearch控制台，在左侧导航栏，单击Beats数据采集中心。
2. 在采集器管理区域，单击目标采集器右侧操作列下的采集器配置。
3. 根据需求修改采集器配置信息，单击保存。

参数	说明
采集器名称	自定义输入采集器的名称。长度为1~30个字符，以大小写字母开头，可以包含字母、数字、下划线（_）或连字符（-）。
安装版本	目前Filebeat只支持6.8.5版本。
采集器Output	Filebeat采集内容的输出地址。直接关联已经创建的阿里云Elasticsearch或Logstash实例。访问协议需要与所选Elasticsearch实例保持一致。
用户名密码	如果采集器Output选择Elasticsearch，需要提供对应的用户名和密码，使Filebeat有权限向Elasticsearch实例中写入数据。
启用Monitoring	用来监控Filebeat的相关指标。如果采集器Output选择Elasticsearch，Monitor默认使用和采集器Output相同的阿里云Elasticsearch实例。如果采集器Output选择Logstash，则需要在配置文件中进行额外配置。
启用Kibana Dashboard	用来配置默认的Kibana Dashboard。由于阿里云Kibana配置在VPC内，因此需要在Kibana配置页面开通Kibana私网访问功能，详情请参见配置Kibana公网或私网访问白名单。
填写Filebeat文件目录	Filebeat专有的配置项。由于阿里云采用Docker部署Beats，因此需要将您希望采集的目录映射到Docker内，建议与*filebeat.yml*中的 `input.path` 保持一致。
采集器YML配置	采集器配置文件。请根据业务需求，修改采集器YML文件的配置，详情请参见采集器YML配置。

> 注意 指定采集器Output后，不需要再在采集器YML配置中单独设置Output，否则会提示ECS采集器安装错误。

保存后，采集器状态变为生效中。等待状态变为已生效，即可完成采集器配置信息的修改。

3.采集器YML配置

采集器YML配置

通过采集器YML配置，您可以根据需求修改对应的配置，并启用该配置，完成数据采集任务。本文介绍采集器YML文件的配置方法和配置参数详情。

filebeat.yml配置 metricbeat.yml配置 auditbeat.yml配置 heartbeat.yml配置

前提条件

创建阿里云Elasticsearch（简称ES）实例，并开启实例的**自动创建索引**功能。创建实例的具体步骤请参见创建阿里云Elasticsearch实例。

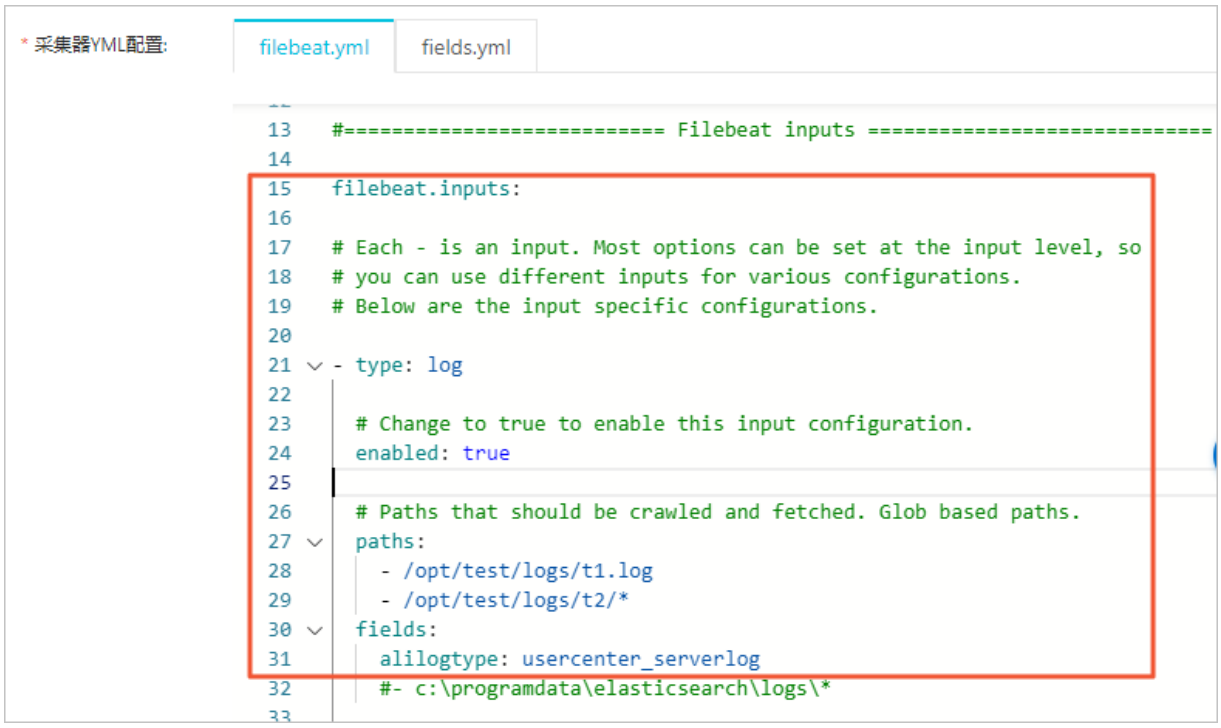
出于安全考虑，阿里云ES默认不允许**自动创建索引**。但是Beats目前依赖该功能，因此如果**采集器Output**选择为**Elasticsearch**，需要开启**自动创建索引**功能，详情请参见开启自动创建索引。

> 说明 Beats中包含很多module供您使用，阿里云Beats未直接提供这些module的单独配置，如果需要使用这些module，请直接在相应的主配置文件中配置。例如，在Metricbeat中启用system module，可以在metricbeat.yml配置中添加如下脚本。

```
metricbeat.modules:
- module: system
metricsets: ["diskio","network"]
diskio.include_devices: []
period: 1s
```

Filebeat配置

通过在filebeat.yml中指定 filebeat.inputs ，来决定如何查找或处理输入数据源。简单的input配置如下。




```
filebeat.inputs:
- type: log
enabled: true
paths:
- /opt/test/logs/t1.log
- /opt/test/logs/t2/*
fields:
alilogtype: usercenter_serverlog
```

注意

- 如果在安装采集器时，指定了采集器Output，那么不需要再在采集器YML配置中单独设置Output，否则会提示ECS采集器安装错误。
- 每个输入源都以 - 开头，可以指定多个 - ，代表多个输入源。

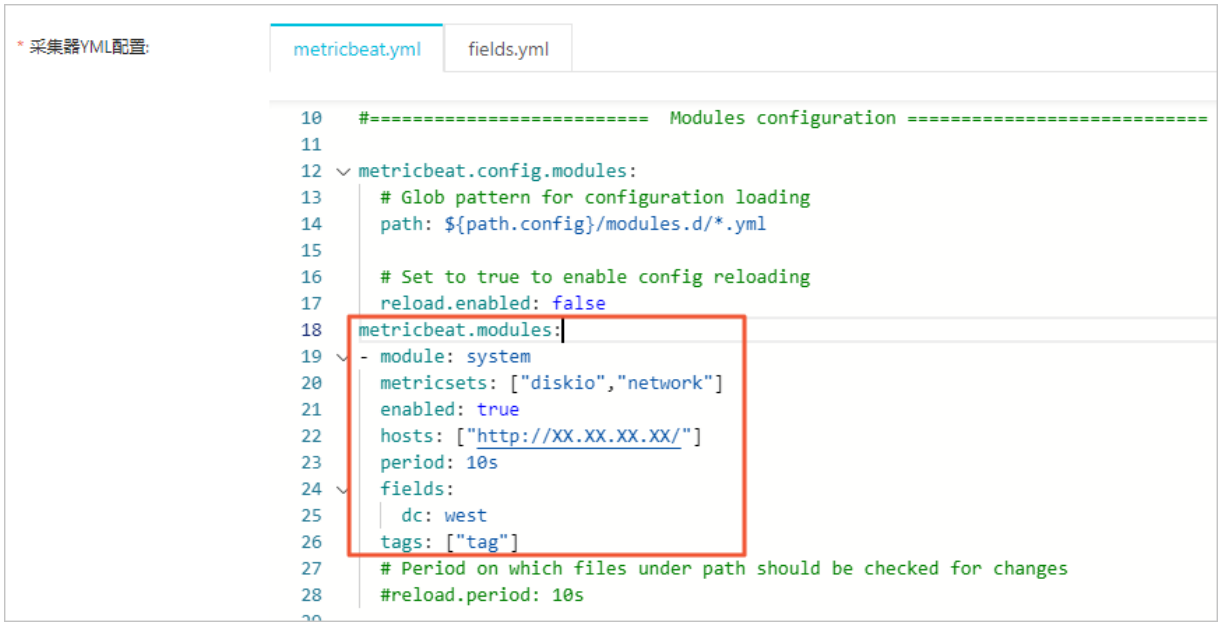
配置	说明
type	输入类型。默认是 log 类型，同时还支持 stdin 、 redis 、 tcp 、 syslog 等类型。
paths	指定要监控的日志。可以指定具体的文件或者目录，指定的文件或目录会被映射进Docker目录中。

配置	说明
enabled	设置配置是否生效。true表示生效，false表示不生效。
fields	指定可选字段。在该字段下，空格缩进两格填写需要添加的字段。例如，设置为 alilogtype: usercenter_serverlog，表示在输出的每条日志中加入该字段，用于标识该日志源的类别，在传输到下一层Logstash时，可以根据该字段类别，对日志进行分类处理。

更多配置详情请参见官方Log input文档。

Metricbeat配置

Metricbeat能够以一种轻量型的方式，输送各种系统和服务统计数据。通过在metricbeat.yml中指定 metricbeat.modules，来指定 module 配置。




```
metricbeat.modules:
- module: system
metricsets: ["diskio","network"]
enabled: true
hosts: ["http://XX.XX.XX.XX/"]
period: 10s
fields:
dc: west
tags: ["tag"]
```

注意 如果在安装采集器时，指定了采集器Output，那么不需要再在采集器YML配置中单独设置Output，否则会提示ECS采集器安装错误。

配置	说明
module	要运行的模块的名称。支持的模块及相关说明请参见Modules。
metricsets	指定要执行的metricsets列表。更多metricsets列表请参见Modules。
enabled	设置配置是否生效。true表示生效，false表示不生效。
period	metricsets执行的频率。如果无法访问系统，metricbeat将为每个时间段返回一个错误信息。
hosts	可选，获取主机列表信息。
fields	指定可选字段。与metricset事件一起发送的字段。
tags	可选，与metricset一起发送的tag列表。

更多配置详情请参见官方Metricbeat配置文档。

Heartbeat配置

Heartbeat能够以一种轻量型的方式，安装在远程服务器，定期检查服务的状态并确定是否可用。与Metricbeat不同，Metricbeat检测服务是启动还是关闭状态，Heartbeat检测服务是否可访问。

通过在heartbeat.yml中指定 heartbeat.monitors ，来指定监控的服务。

> 说明 Heartbeat只需要配置监控的服务即可，为保证可用性，建议至少部署2台云服务器ECS。

采集器YML配置:

heartbeat.yml | fields.yml

```
12  # Define a directory to load monitor definitions from. Definitions take
    the form
13  # of individual yaml files.
14  heartbeat.config.monitors:
15    # Directory + glob pattern to search for configuration files
16    path: ${path.config}/monitors.d/*.yml
17    # If enabled, heartbeat will periodically check the config.monitors
    path for changes
18    reload.enabled: false
19    # How often to check for changes
20    reload.period: 5s
21
22  # Configure monitors inline
23  heartbeat.monitors:
24  - type: http
25    name: ecs_monitor
26    enabled: true
27    urls: ["http://localhost:9200"]
28    schedule: '@every 5s'
29    fields:
30      dc: west
31  #- type: http
```

```
heartbeat.monitors:
- type: http
name: ecs_monitor
enabled: true
urls: ["http://localhost:9200"]
schedule: '@every 5s'
fields:
dc: west
```

注意 如果在安装采集器时，指定了采集器Output，那么不需要再在采集器YML配置中单独设置Output，否则会提示ECS采集器安装错误。

配置	说明
type	monitor类型。支持 icmp 、 tcp 、 http 。
name	monitor名称。该值显示在monitor字段下的Exported fields中，作为job name， type 字段作为job type。
enabled	设置配置是否生效。 true 表示生效， false 表示不生效。
urls	可选，用来连通的服务器列表。
schedule	指定任务计划。设置为 @every 5s ，表示从启动Heartbeat开始，每5秒运行一次任务；设置为 */5 * * * * * ，表示每5秒运行一次任务。
fields	指定可选字段。将该字段添加到output中，作为附加信息。

更多配置详情请参见官方Heartbeat配置文档。

Auditbeat配置

Auditbeat是轻量型的审计日志采集器，可以收集Linux审计框架的数据，监控文件的完整性。Auditbeat能够组合相关消息到一个事件里，生成标准的结构化数据，方便分析，而且能够与Logstash、Elasticsearch和Kibana无缝集成。

注意 Auditbeat依赖于操作系统的Linux Audit Framework，要求OS Kernel版本至少为3.14。且保证Auditd服务为stop状态（可使用 service auditd status 命令查看）。

通过在auditbeat.yml中指定 auditbeat.modules ，来配置Auditbeat采集器。auditbeat.yml主要分为两部分，一部分为模块，另一部分为输出。 启用特定模块，需要在auditbeat.yml中添加特定参数，如下示例展示 auditd 和 file_integrity 配置。

```
auditbeat.modules:
- module: auditd
audit_rules: |
-w /etc/passwd -p wa -k identity
-a always,exit -F arch=b32 -S open,creat,truncate,ftruncate,openat,open_by_handle_at -F exit=-EPERM
-k access
- module: file_integrity
paths:
- /bin
- /usr/bin
- /sbin
- /usr/sbin
- /etc
```

> **注意** 如果在安装采集器时，指定了采集器Output，那么不需要再在采集器YML配置中单独设置Output，否则会提示ECS采集器安装错误。

更多auditbeat.yml配置说明请参见官方Auditbeat配置文档；配置模块（ `module` ）详情请参见Modules。

4.Beats安装失败的排查与解决方法

当您安装采集器（Beats）时，遇到安装失败、心跳异常的情况时，可参考本文的方法进行排查解决。

操作步骤

1. 排查安装Beats服务的ECS的操作系统是否是Aliyun Linux、RedHat或CentOS。
2. 排查安装Beats服务的ECS是否与Elasticsearch或Logstash实例处于同一专有网络下。
3. 排查安装Beats服务的ECS是否安装了云助手和Docker。您可以连接ECS实例，通过以下命令进行验证。
   - 查看云助手服务状态

     ```
     systemctl status aliyun.service
     ```

     正常情况下，返回结果如下。

     ```
     [root@VM01 ~]# systemctl status aliyun.service
     ● aliyun.service - aliyun-assist
        Loaded: loaded (/etc/systemd/system/aliyun.service; enabled; vendor preset: disabled)
        Active: active (running) since Sat 2020-08-15 15:38:25 CST; 4 days ago
      Main PID: 20311 (aliyun-service)
        CGroup: /system.slice/aliyun.service
                └─20311 /usr/sbin/aliyun-service

     Aug 15 15:38:25 VM01 systemd[1]: Stopped aliyun-assist.
     Aug 15 15:38:25 VM01 systemd[1]: Started aliyun-assist.
     ```

     如果未安装云助手，可参见安装云助手客户端进行安装。
   - 查看Docker状态

     ```
     systemctl status docker
     ```

     正常情况下，返回如下结果。

     ```
     [root@VM01 ~]# systemctl status docker
     ● docker.service - Docker Application Container Engine
        Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
        Active: active (running) since Wed 2020-08-19 16:45:21 CST; 1min 39s ago
          Docs: http://docs.docker.com
      Main PID: 14625 (dockerd-current)
        CGroup: /system.slice/docker.service
                ├─14625 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/libexec/docker/docker-runc-current --default-runti.
                └─14632 /usr/bin/docker-containerd-current -l unix:///var/run/docker/libcontainerd/docker-containerd.sock --metric.

     Aug 19 16:45:20 VM01 dockerd-current[14625]: time="2020-08-19T16:45:20.973350772+08:00" level=warning msg="Docker could...yste
     Aug 19 16:45:21 VM01 dockerd-current[14625]: time="2020-08-19T16:45:21.005411644+08:00" level=info msg="Graph migration...cond
     Aug 19 16:45:21 VM01 dockerd-current[14625]: time="2020-08-19T16:45:21.006055965+08:00" level=info msg="Loading contain...tart
     Aug 19 16:45:21 VM01 dockerd-current[14625]: time="2020-08-19T16:45:21.136526271+08:00" level=info msg="Firewalld runni...fals
     Aug 19 16:45:21 VM01 dockerd-current[14625]: time="2020-08-19T16:45:21.239151591+08:00" level=info msg="Default bridge ...dres
     Aug 19 16:45:21 VM01 dockerd-current[14625]: time="2020-08-19T16:45:21.280253025+08:00" level=info msg="Loading contain...done
     Aug 19 16:45:21 VM01 dockerd-current[14625]: time="2020-08-19T16:45:21.442183479+08:00" level=info msg="Daemon has comp...atio
     Aug 19 16:45:21 VM01 dockerd-current[14625]: time="2020-08-19T16:45:21.442217221+08:00" level=info msg="Docker daemon" ...1.13
     Aug 19 16:45:21 VM01 systemd[1]: Started Docker Application Container Engine.
     Aug 19 16:45:21 VM01 dockerd-current[14625]: time="2020-08-19T16:45:21.450179119+08:00" level=info msg="API listen on /....soc
     Hint: Some lines were ellipsized, use -l to show in full.
     ```

     如果未安装Docker，可参见部署并使用Docker进行安装。
4. 检查采集器的YML配置，确认是否已设置如下信息。

```
- type: log
# Change to true to enable this input configuration.
enabled: true
# Paths that should be crawled and fetched. Glob based paths.
paths:
- /var/log/*.log
```

参数	说明
enabled	默认为false，使用时一定要设置为true。
paths	指定日志文件路径，可以采用模糊匹配，例如*.log。

注意

- paths与配置页面填写的Filebeat文件目录存在区别。Filebeat文件目录是Docker映射的目录，只有映射到采集目录下才能采集到paths指定的文件。建议二者保持一致。
- 在配置页面指定采集器Output后，YML配置下不能重新指定Output，否则会提示安装错误。
- 对于采集器YML中默认已经注释掉（#）的参数，需谨慎修改，比如与X-Pack相关的参数设置，否则会导致安装失败。

5. 连接ECS实例，查看*/opt/aliyunbeats/*目录下，是否生成了对应的Beats实例，并确认是否存在conf、data、logs数据。

```
[root@VM01 ~]# cd /opt/aliyunbeats
[root@VM01 aliyunbeats]# ls
ct-cn-77uqof2s7rg
[root@VM01 aliyunbeats]# cd ct-cn-77uqof2s7rg
[root@VM01 ct-cn-77uqof2s7rg ]# ls
filebeat
[root@VM01 ct-cn-77uqof2s7rg ]# cd filebeat
[root@VM01 filebeat]# ls
conf  data  logs
```

您也可以在logs目录下查看Beats日志，方便定位问题。

```
[root@VM01 logs]# tail -n 2 filebeat
2020-08-19T09:25:52.871Z        INFO    [monitoring]    log/log.go:144  Non-zero metrics in the last 30s        {"monitoring"
"metrics": {"beat":{"cpu":{"system":{"ticks":130,"time":{"ms":1}},"total":{"ticks":390,"time":{"ms":5},"value":390},"user":{"
ks":260,"time":{"ms":4}}},"handles":{"limit":{"hard":1048576,"soft":1048576},"open":6},"info":{"ephemeral_id":"f960431f-2849-
e-9dc9-f17        ","uptime":{"ms":1590027}},"memstats":{"gc_next":4285312,"memory_alloc":2157688,"memory_total":36754744}},
lebeat":{"harvester":{"open_files":0,"running":0}},"libbeat":{"config":{"module":{"running":0}},"pipeline":{"clients":1,"even
:{"active":0}}},"registrar":{"states":{"current":1}},"system":{"load":{"1":0.17,"15":0.11,"5":0.09,"norm":{"1":0.17,"15":0.11
":0.09}}}}}}
2020-08-19T09:26:22.872Z        INFO    [monitoring]    log/log.go:144  Non-zero metrics in the last 30s        {"monitoring"
"metrics": {"beat":{"cpu":{"system":{"ticks":130,"time":{"ms":3}},"total":{"ticks":390,"time":{"ms":4},"value":390},"user":{"
ks":260,"time":{"ms":1}}},"handles":{"limit":{"hard":1048576,"soft":1048576},"open":6},"info":{"ephemeral_id":"f960431f-2849-
e-9dc9-f17        ","uptime":{"ms":1620027}},"memstats":{"gc_next":4285312,"memory_alloc":2449992,"memory_total":37047048}},
lebeat":{"harvester":{"open_files":0,"running":0}},"libbeat":{"config":{"module":{"running":0}},"pipeline":{"clients":1,"even
:{"active":0}}},"registrar":{"states":{"current":1}},"system":{"load":{"1":0.11,"15":0.11,"5":0.08,"norm":{"1":0.11,"15":0.11
":0.08}}}}}}
```

6. 查看Beats所在容器的运行状态，并分析日志定位问题。

i. 查看Docker容器中的运行状态。

```
docker ps -a | grep filebeat
```

ii. 当容器处于exited状态时，查看容器输出日志。

```
docker logs -f 容器id
```