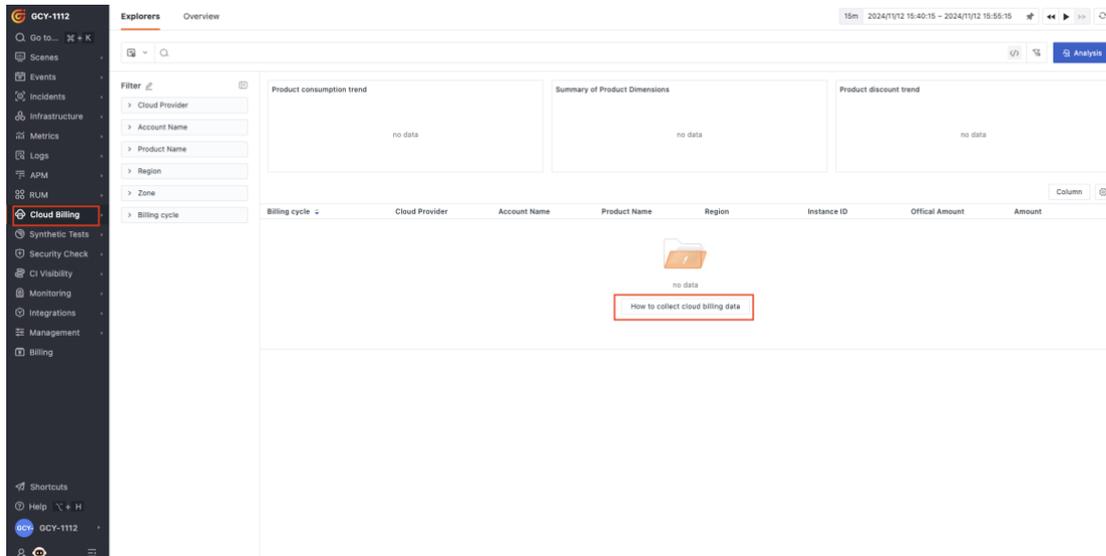


云账单管理功能使用指南 |

GUANCE
GUANCE
GUANCE

Step 1: 前往云账单管理页

- 1、点击菜单栏“Cloud Billing”，首次使用可查看提示“How to collect cloud billing data”进行操作。



- 2、根据提示云账单分析包括三个步骤：

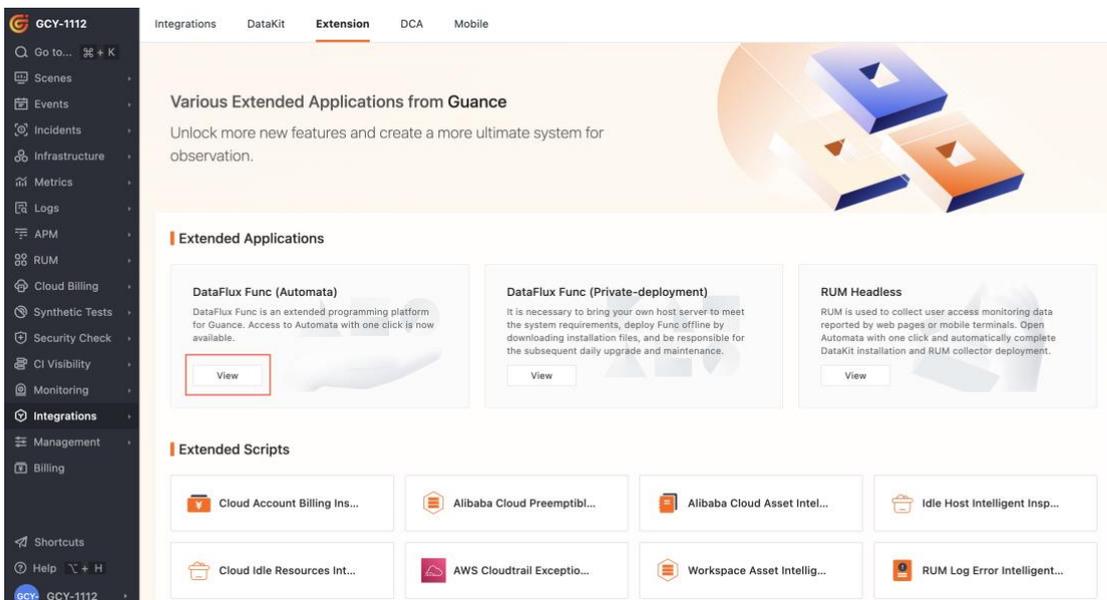
- Install Func: 安装 Func，有两种方式，一种是直接在“Integrations” - “Extension”，直接开通 DataFlux Func，另外一种是通过手动部署 Func，这两种方式都会在下面进行介绍。
- Install the Script: 安装集成脚本
- View In Dashboard: 效果展示



Step 2: 安装 DataFlux Func

方法一：开通观测云内置的 DataFlux Func

- 1、在观测云控制台 “Integrations” - “Extension” ，在 “DataFlux Func (Automata)” 选择 “View” 。



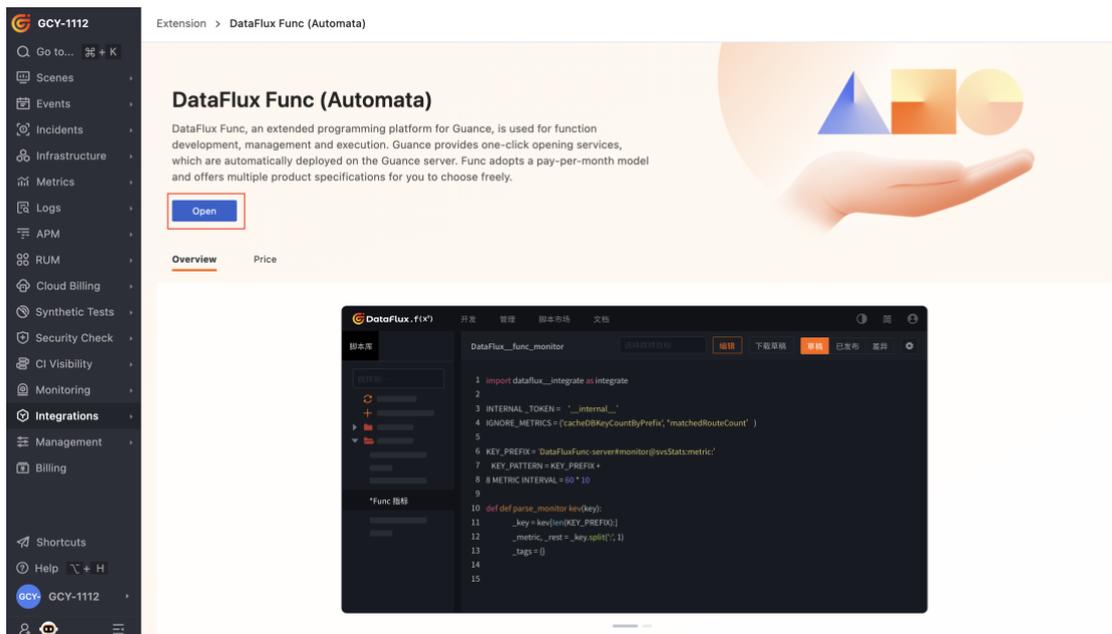
- 2、点击 “Open” 。

注意：此处默认为灰色不可点，若您需要开通，可以“联系我们”

400-882-3320

sales@guance.com

加入白名单。



Extension > DataFlux Func (Automata)

DataFlux Func (Automata)

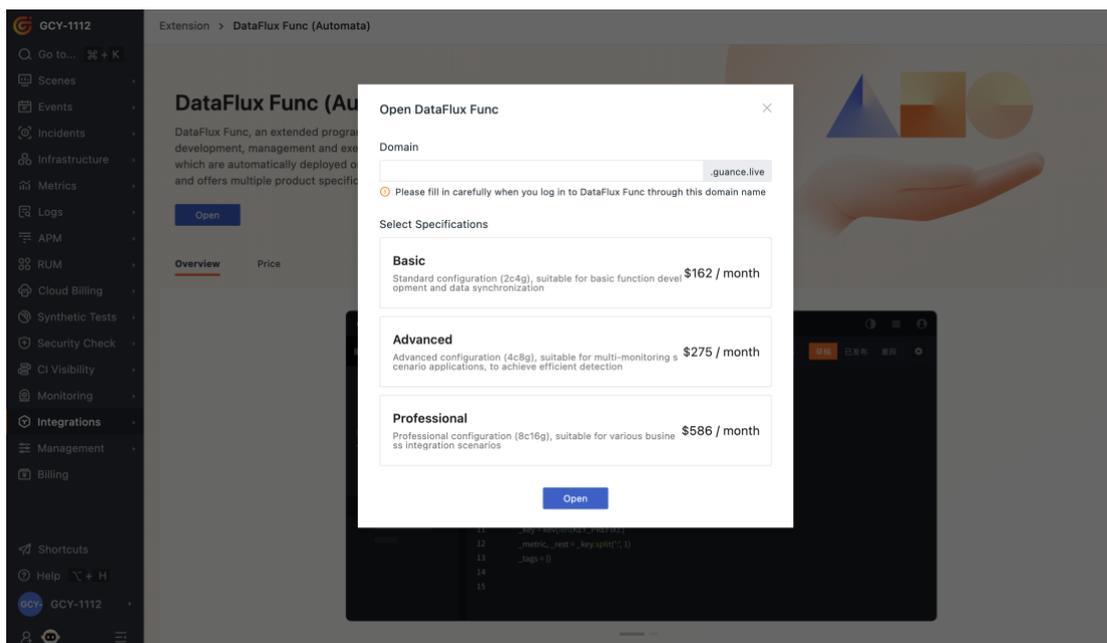
DataFlux Func, an extended programming platform for Guance, is used for function development, management and execution. Guance provides one-click opening services, which are automatically deployed on the Guance server. Func adopts a pay-per-month model and offers multiple product specifications for you to choose freely.

[Open](#)

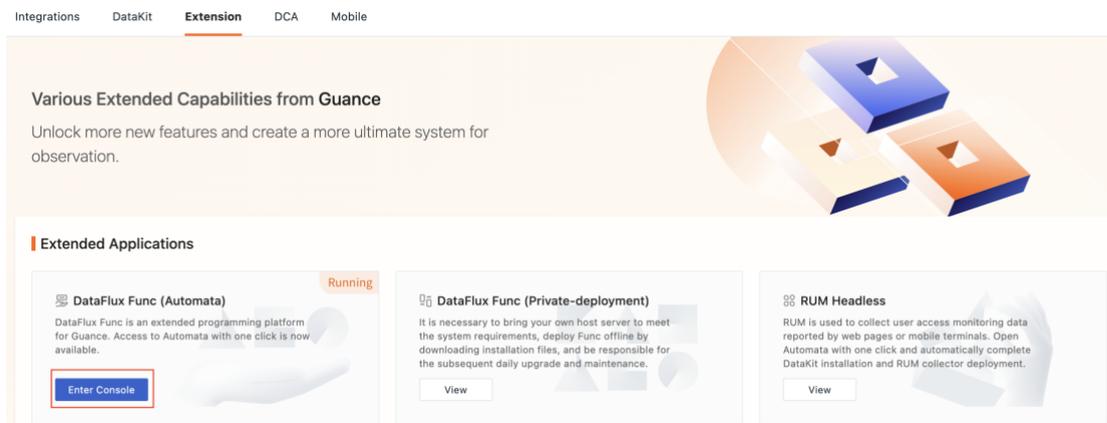
Overview Price

```
1 import dataflux__integrate as integrate
2
3 INTERNAL_TOKEN = "...internal..."
4 IGNORE_METRICS = ["cacheDBKeyCountByPrefix", "matchedRouteCount"]
5
6 KEY_PREFIX = "DataFluxFunc-server#monitor@svsStats.metric"
7 KEY_PATTERN = KEY_PREFIX +
8 METRIC_INTERVAL = 60 * 10
9
10 def def parse_monitor key(key):
11     _key = key[1:] if KEY_PREFIX in key else key
12     _metric, _rest = _key.split("#", 1)
13     _tags = []
14
15
```

3、输入“Domain”，“Select Specifications”，点击“Open”。



4、开通成功后，可在“Extension” - “DataFlux Func (Automata)” ，点击“Enter Console” ，即可登录到 Func 控制台。



方法二：手动部署 DataFlux Func

1、登录到用于部署 DataFlux Func 的主机，执行以下命令下载“DataFlux Func GSE”。

注意：部署 DataFlux Func 至少需要 2 核 4G 的主机，并需要开通 8088 端口。

Bash

```
/bin/bash -c "$(curl -fsSL func.guance.com/download)" -- --for=GSE
```

```
Last login: Tue Nov 12 09:58:42 2024 from .....
ubuntu@ip-172-31-3-118:~$ /bin/bash -c "$(curl -fsSL func.guance.com/download)" -- --for=GSE
Downloading file https://static.guance.com/dataflux-func/portable-common/x86_64/docker-24.0.9.tgz
--2024-11-12 09:52:09-- https://static.guance.com/dataflux-func/portable-common/x86_64/docker-24.0.9.tgz
Resolving static.guance.com (static.guance.com)... 116.207.98.38, 116.207.98.39, 116.207.98.40, ...
Connecting to static.guance.com (static.guance.com)|116.207.98.38|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 78199284 (67M) [application/octet-stream]
Saving to: 'docker-24.0.9.tgz'

docker-24.0.9.tgz      100%[=====] 66.95M  15.3MB/s   in 4.5s
2024-11-12 09:52:09 (14.9 MB/s) - 'docker-24.0.9.tgz' saved [78199284/78199284]

Downloading file https://static.guance.com/dataflux-func/portable-common/x86_64/mysql.tar.gz
--2024-11-12 09:52:09-- https://static.guance.com/dataflux-func/portable-common/x86_64/mysql.tar.gz
Resolving static.guance.com (static.guance.com)... 116.207.98.38, 116.207.98.40, 116.207.98.36, ...
Connecting to static.guance.com (static.guance.com)|116.207.98.38|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12126674 (11.6M) [application/gzip]
Saving to: 'mysql.tar.gz'

mysql.tar.gz          100%[=====] 11.65M  6.27MB/s   in 1.6s
```

2、下载完成后，可以看到目录“dataflux-func-portable-x86_64-5.1.7-GSE”，进入此目录，执行以下安装命令，即可完成安装 DataFlux Func GSE。

Bash

```
sudo /bin/bash run-portable.sh
```

```
ubuntu@ip-172-31-3-118:~$ ls
dataflux-func-portable-x86_64-5.1.7-GSE
ubuntu@ip-172-31-3-118:~$ cd dataflux-func-portable-x86_64-5.1.7-GSE/
ubuntu@ip-172-31-3-118:~/dataflux-func-portable-x86_64-5.1.7-GSE$ ls
dataflux-func.tar.gz  docker-24.0.9.tgz  docker-stack.example.yaml  docker.service  image-list  mysql.tar.gz  redis.tar.gz  run-portable.sh  version
ubuntu@ip-172-31-3-118:~/dataflux-func-portable-x86_64-5.1.7-GSE$ sudo /bin/bash run-portable.sh
```

安装完成后，在最后一行会提示 DataFlux Func 登录方式“http://<IP or Domain>:8088”。

```
Port:
 8088
Installed dir:
 /usr/local/dataflux-func
To shut down:
 sudo docker stack remove dataflux-func
To start:
 sudo docker stack deploy dataflux-func -c /usr/local/dataflux-func/docker-stack.yaml
To uninstall:
 sudo docker stack remove dataflux-func
 sudo rm -rf /usr/local/dataflux-func
 sudo rm -f /etc/logrotate.d/dataflux-func

Now open http://<IP or Domain>:8088/ and have fun!
ubuntu@ip-172-31-3-118:~/dataflux-func-portable-x86_64-5.1.7-GSE$
```

3、稍等片刻，根据上一步提示的方式，在浏览器输入“http://服务器 IP 地址:8088”，点击“Save and init DB”初始化数据库。

注意：选择“Show more configs”，可以手动配置日志等级、MySQL、Redis 和

管理员账号等信息，若非必要，默认即可。

DataFlux f(x³)
Version: 5.1.7 (GSE)

Language: English

Some basic setup should be done before DataFlux Func starts up.

If you are deploying manually, please ensure that the following conditions are met:

- MySQL 5.7 or above
- Empty database exists in MySQL and the database encoding is `utf8mb4`
- There is a user in MySQL who can log in remotely and has permission to the empty database (including CREATE permission)
- Keep the option `innodb_large_prefix` ON in MySQL 5.7 (Official MySQL 5.7 doc)
- Redis 5.0 or above, at least 1 GB available memory and NOT running in Cluster Mode

Administrator username ADMIN_USERNAME
admin

Administrator password (default is admin) ADMIN_PASSWORD

Repeat administrator password ADMIN_PASSWORD_REPEAT

Guance Node GUANCE_NODE
Do not create Guance connector

Cannot find the Guance Node you need?
 Show more configs

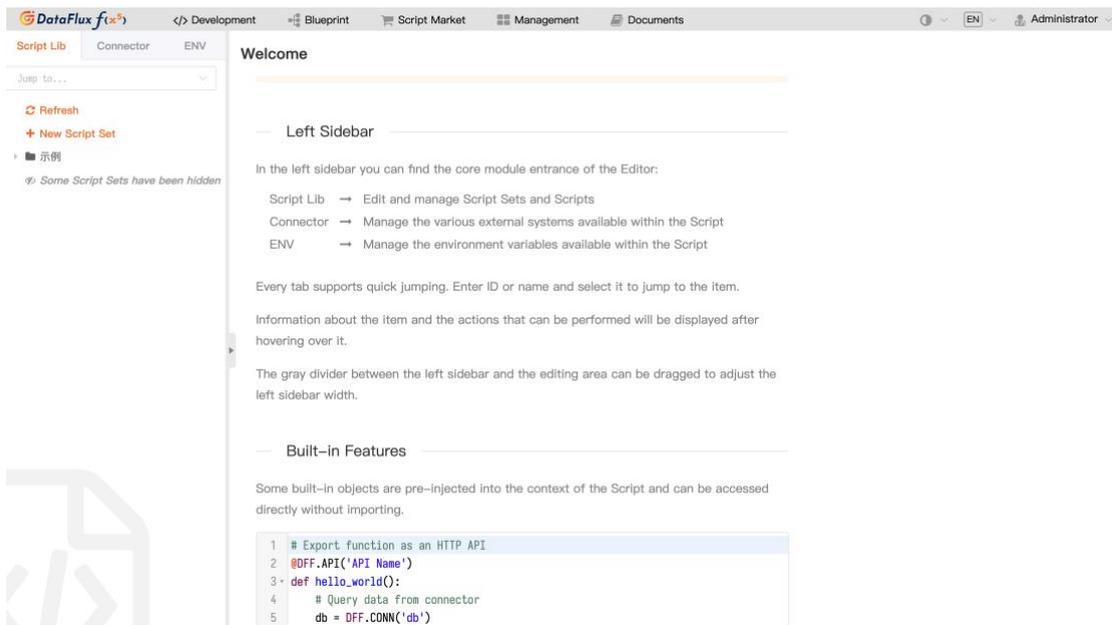
Save and init DB

4、稍等片刻，初始化完成后即自动跳转到登录界面，在“Username”和“Password”输入“admin”，按照页面提示输入“Captcha”，点击“Sign In”。

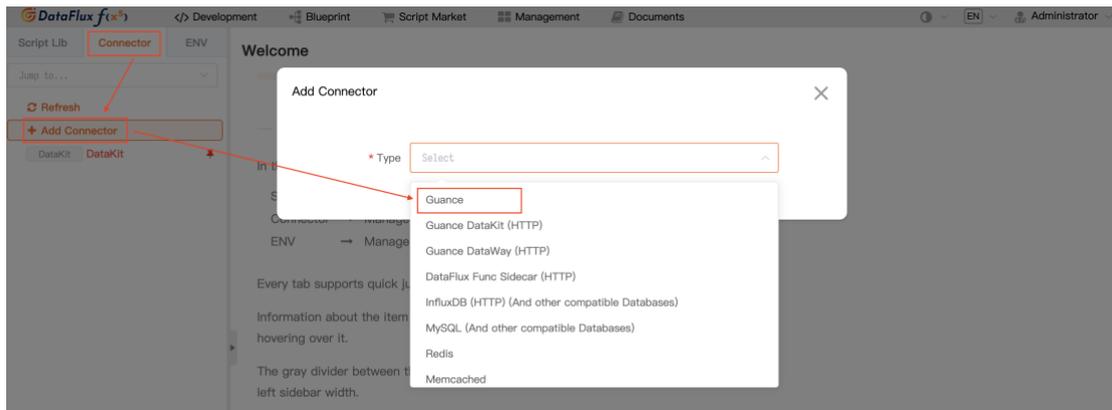
若初始化时未修改密码，初始用户名和密码都是 admin。



5、登录后，进入 DataFlux Func。

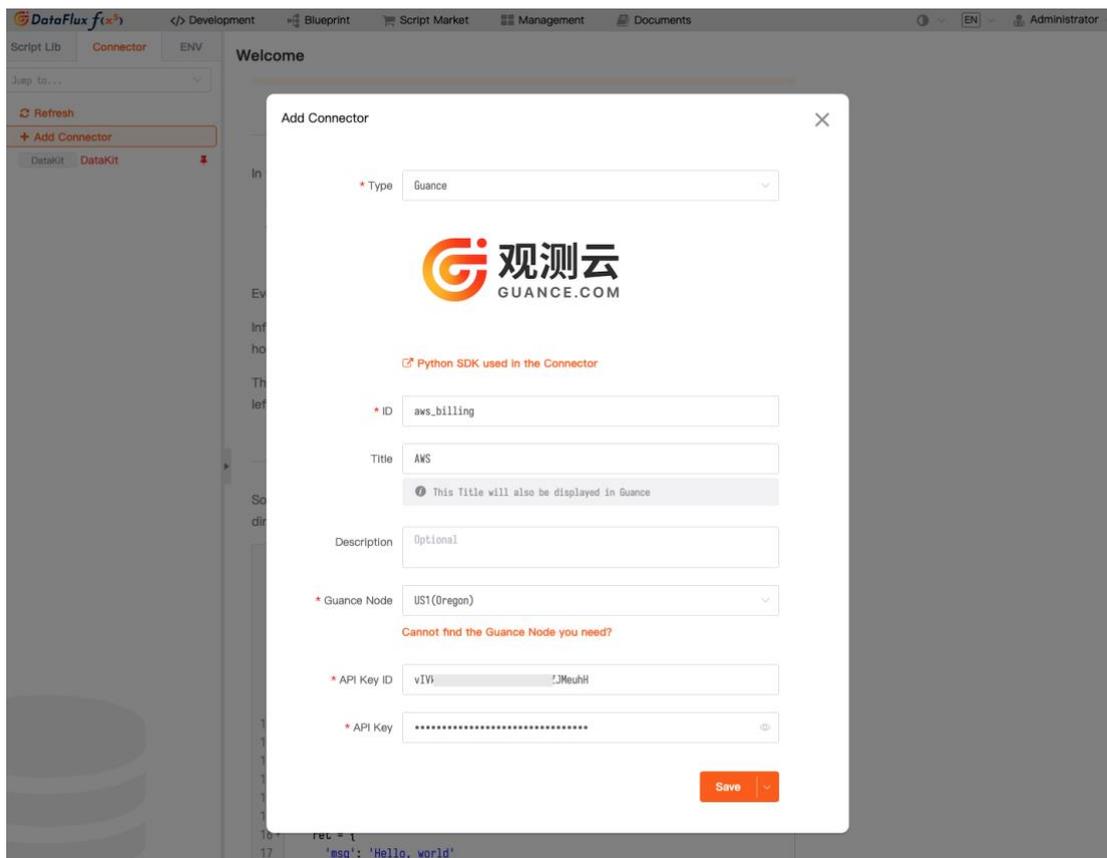


6、点击左侧 “Connector” ，点击 “+ Add Connector” ，在弹出的对话框中的 “Type” ，选择 “Guance” 。

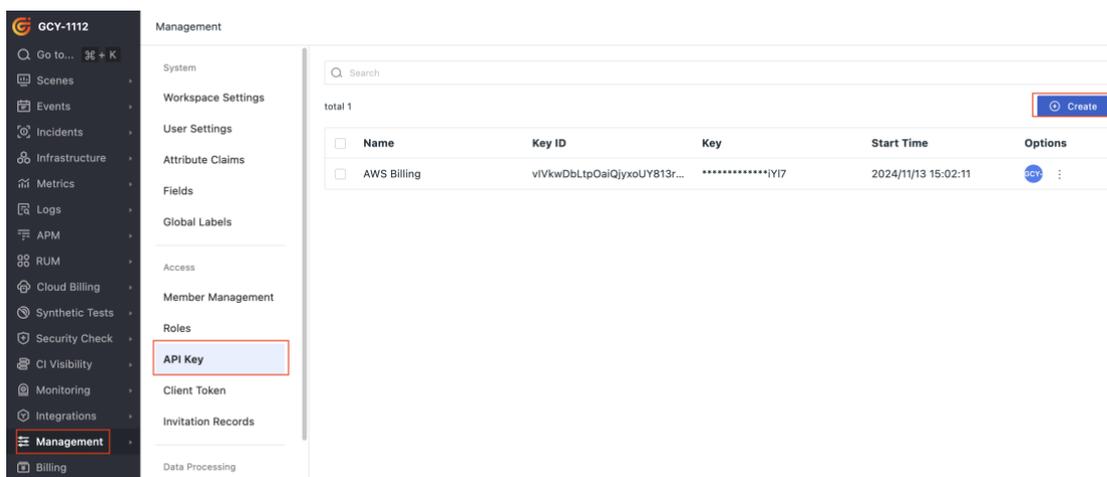


7、输入相关信息，点击“Save”。

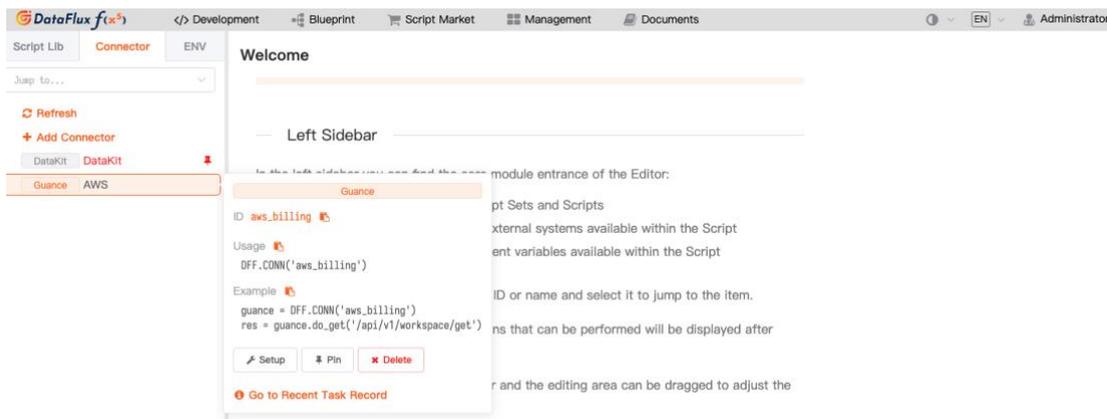
- ID: 用于在脚本中调用，后面会用到，如“aws_billing”；
- Title: 标题可选，如“AWS”；
- Description: 描述可选；
- Guance Node: 观测云节点，需要上报数据的工作空间所属的站点，如“US1(Oregon)”
- API Key ID / API Key: 需要上报数据的工作空间的 API 信息，见下面的截图说明。



API Key ID / API Key, 可以在需要上报的工作空间 “Management” - “API Key”, 点击 “Create” 进行创建。



8、创建完连接器后, 可在连接器列表查看。

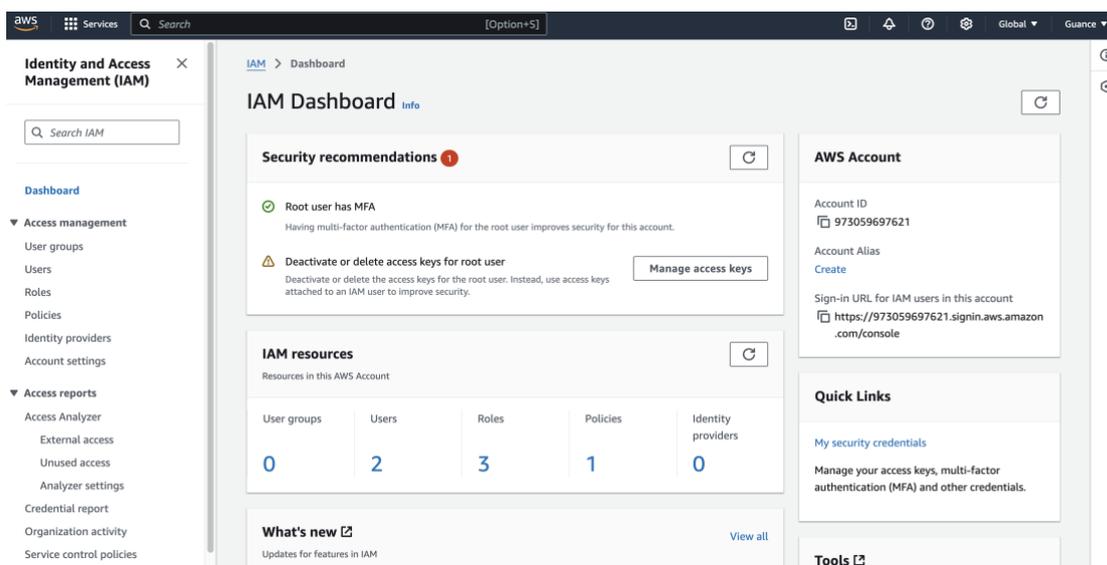


Step 3: 安装 AWS 云账单集成脚本

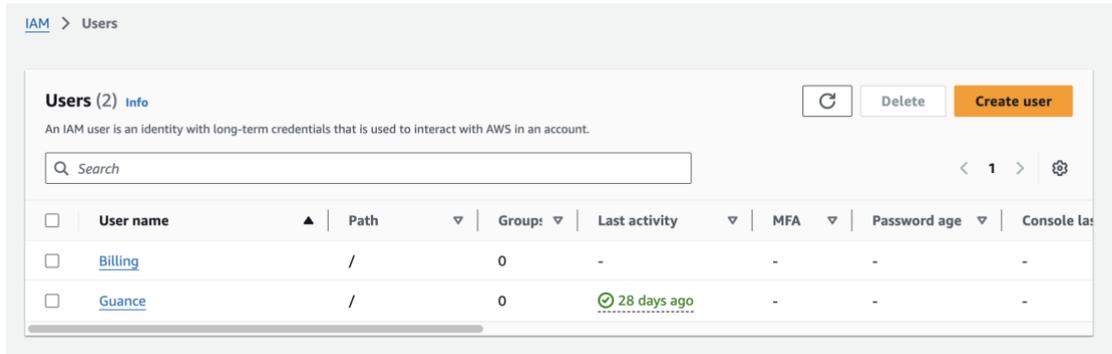
采集 AWS 云账单数据首先需要获取已经授权 AWS 账号的 AK 和 AKS 信息，然后在 DataFlux Func 的脚本库对 AWS 账单集成进行配置发布，发布后，AWS 的账单数据就可以上报到观测云控制台进行查看分析。

获取 AK、AKS 信息

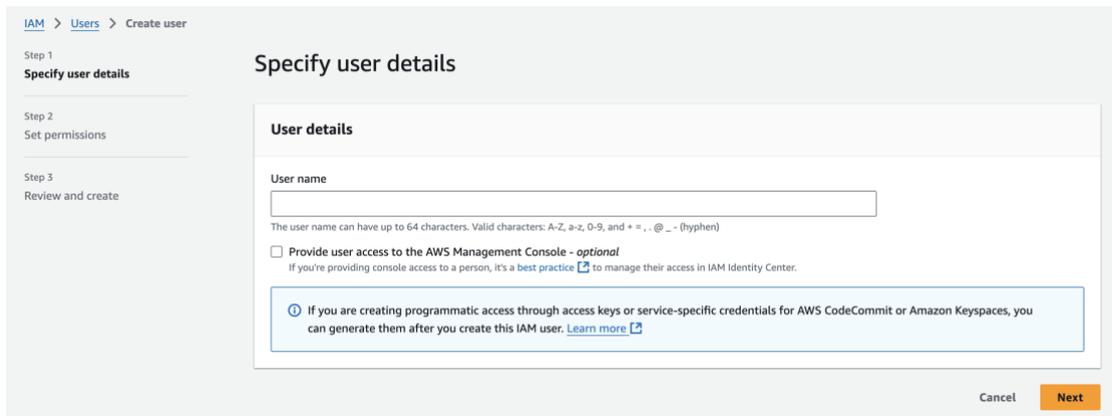
1、登录到 AWS IAM 控制台，在“IAM Dashboard”，点击“Users”。



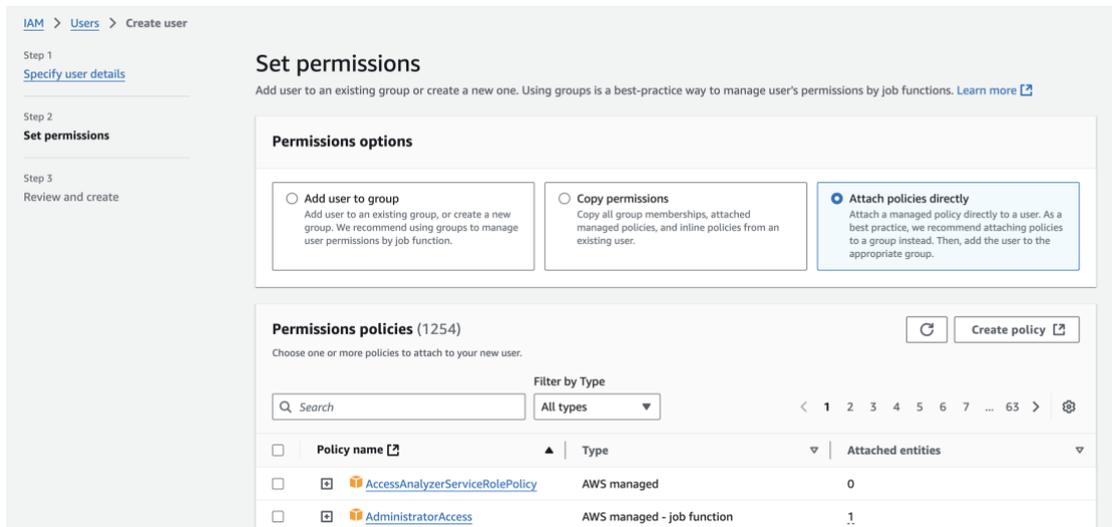
2、在“Users”页面，点击右上角“Create user”。



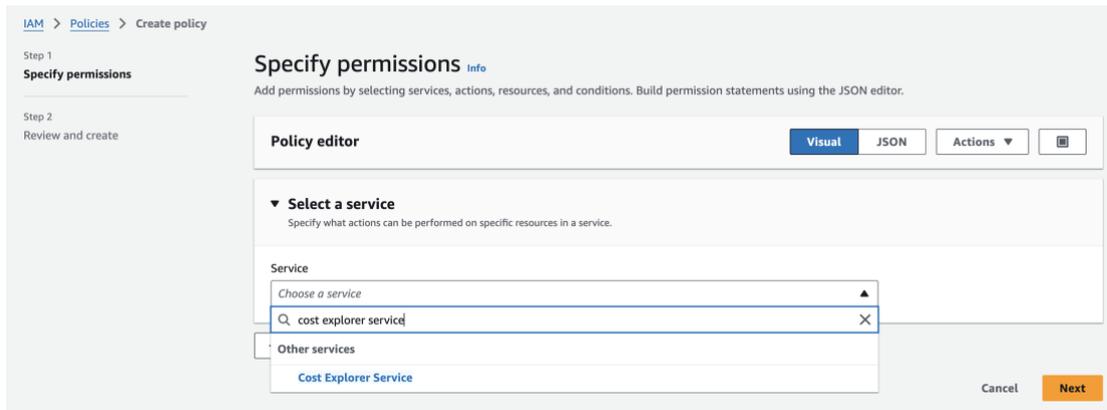
3、输入 “User name” ， 点击 “Next” 。



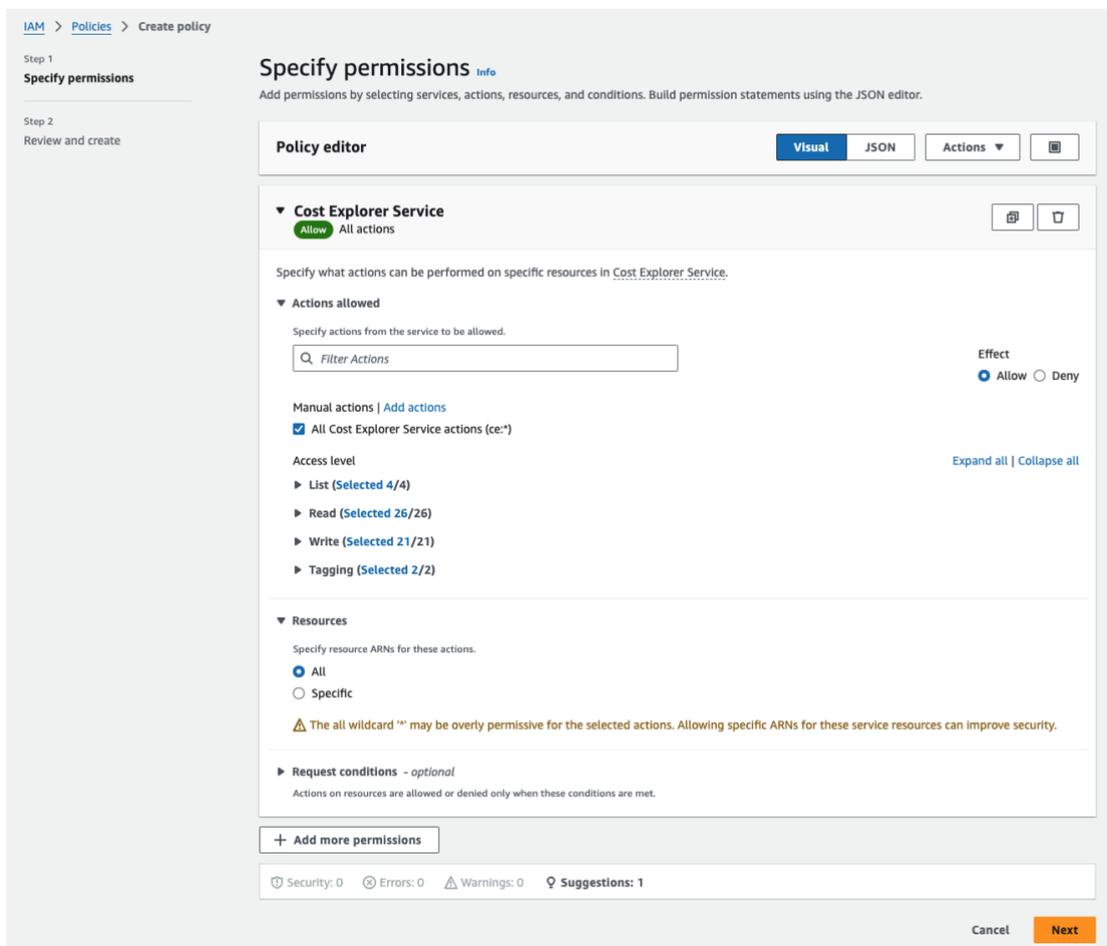
4、选择 “Attach policies directly” ， 点击 “Create policy” 。



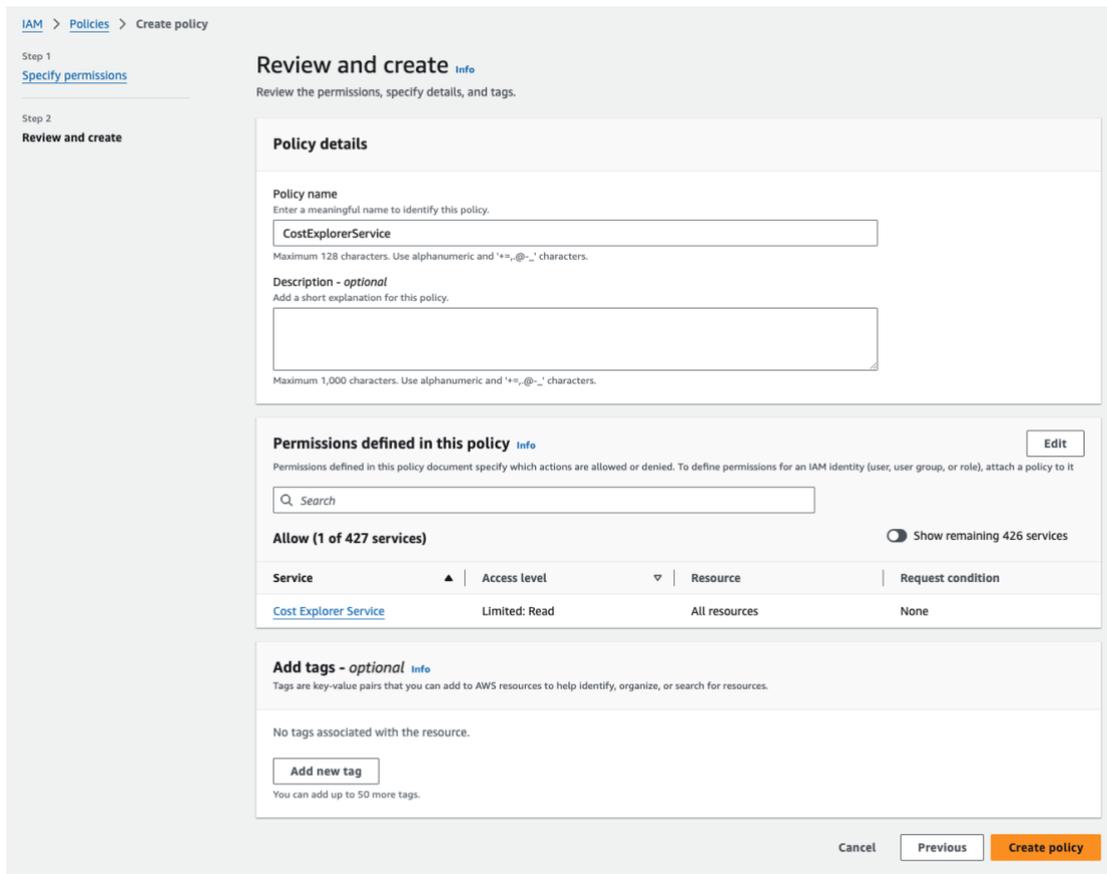
5、选择服务 “Cost Explorer Service” 。



6、在 “Cost Explorer Service” 选择操作，建议选择 “All Cost Explorer Service actions (ce: *)” ，然后点击 “Next” 。

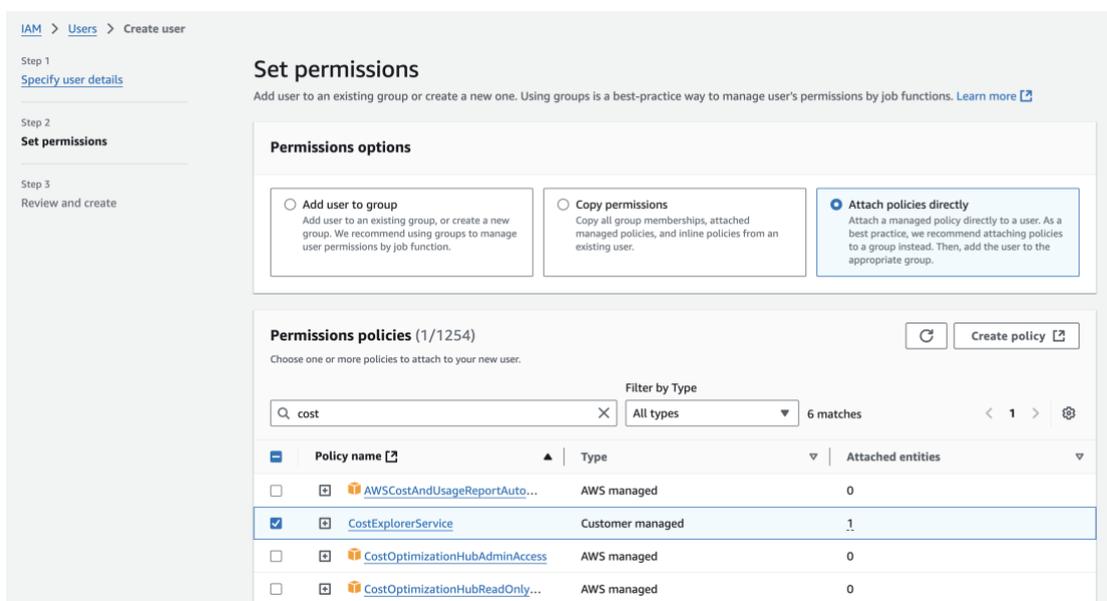


7、配置好服务权限以后，输入 “Policy name” ，点击 “Create policy” 。

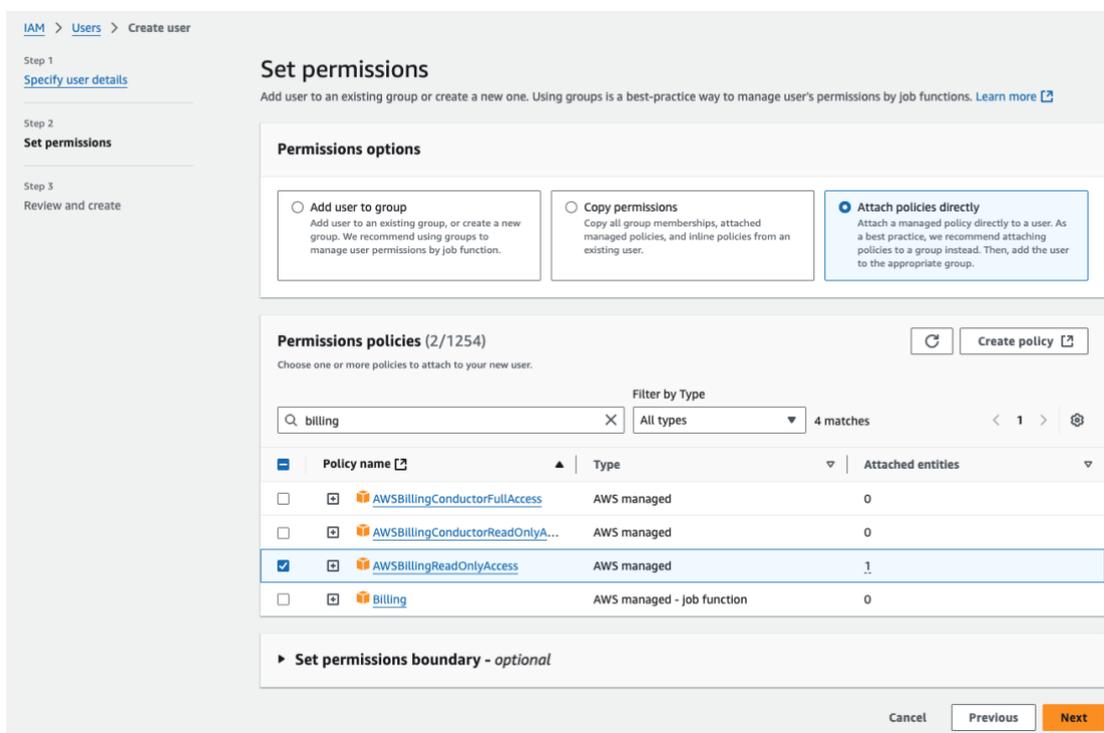


8、策略创建完成后，返回设置权限。

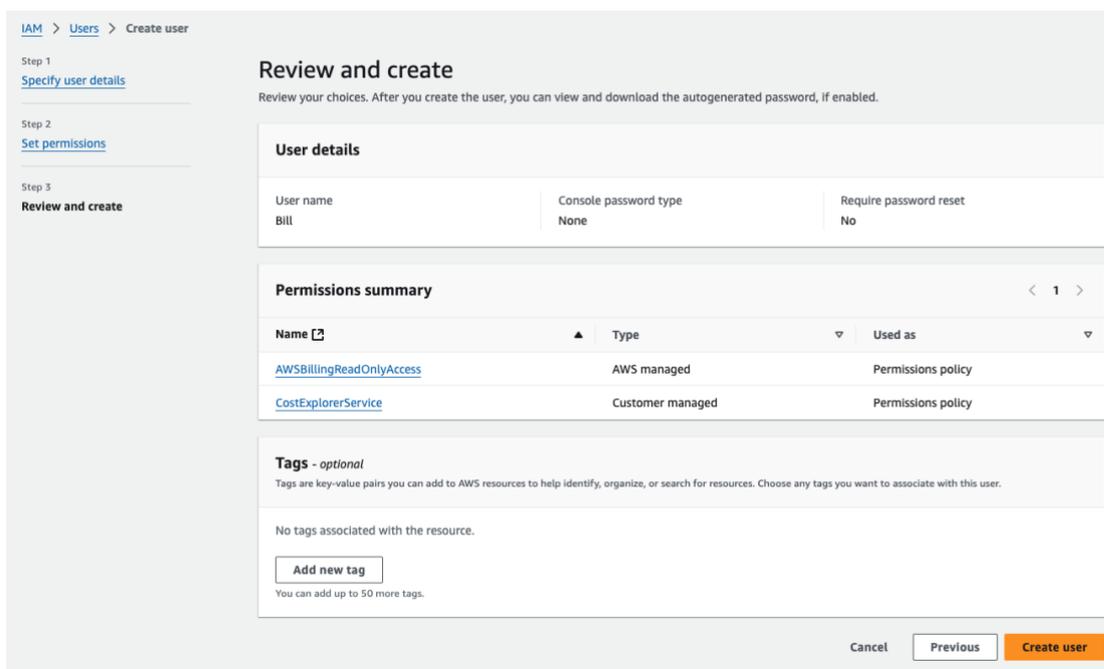
在“Attach policies directly” - “Permissions policies”，输入关键字“cost”，搜索并勾选创建的策略名称，如“CostExplorerService”。



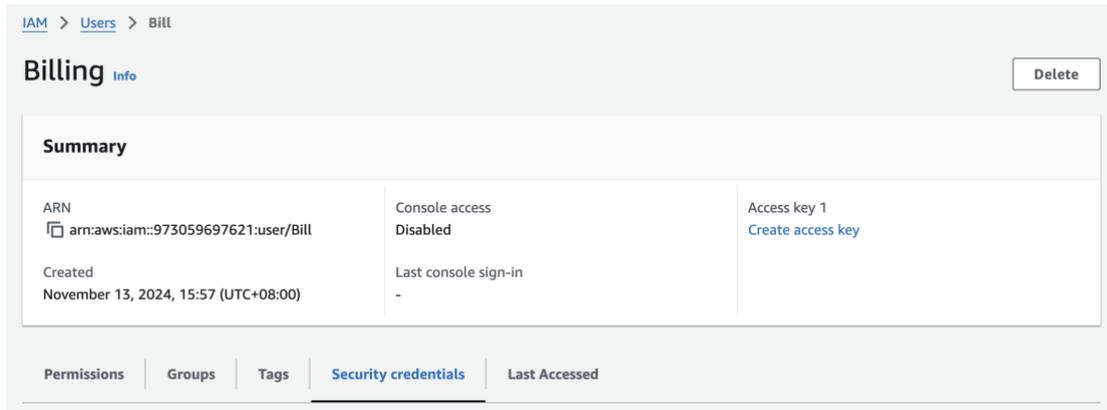
搜索关键字 “billing” ，勾选另外一个策略 “AWSBillingReadOnlyAccess” ，点击 “Next” 。



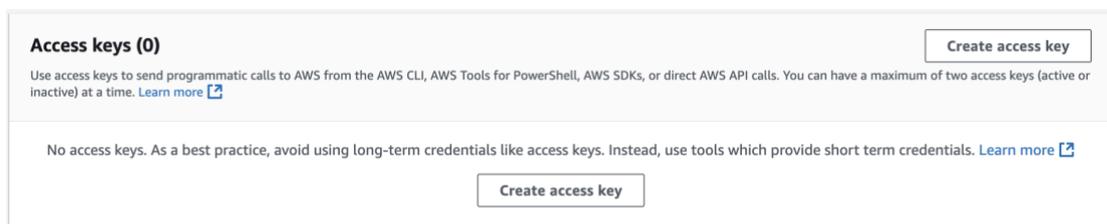
9、权限设置完成后，确认没有问题后，点击 “Create user” 。



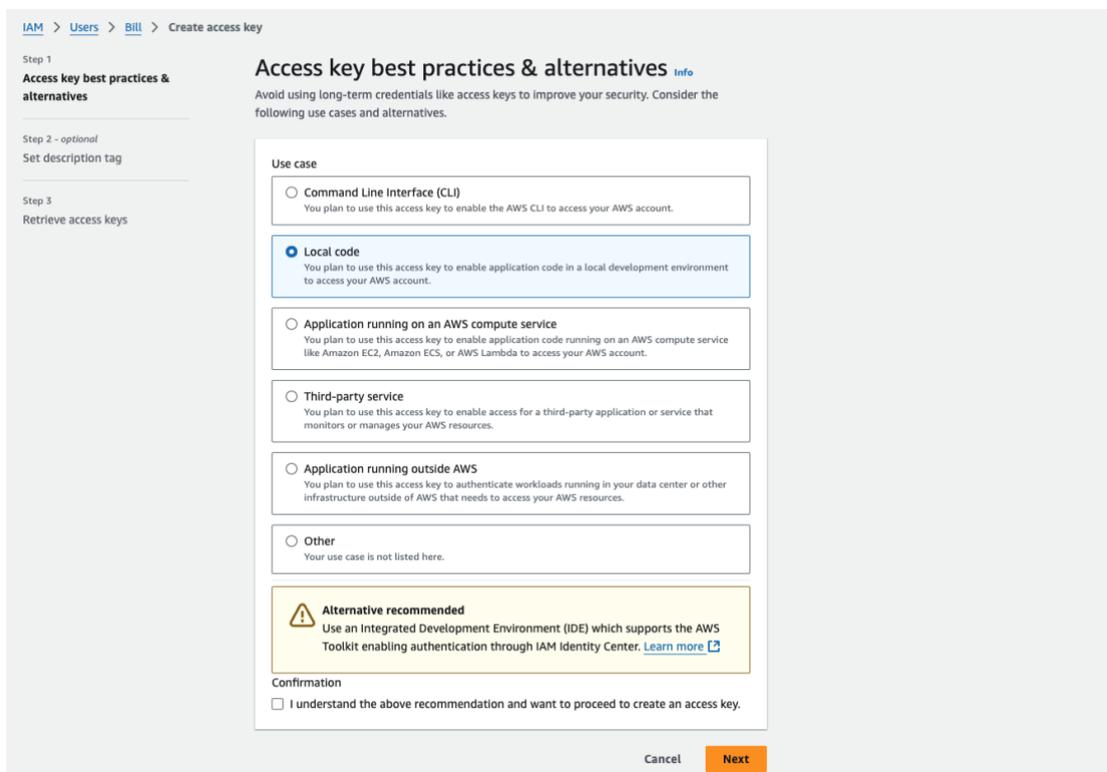
10、用户创建完成后，进入 “Users” 页面，选择 “Security credentials” 。



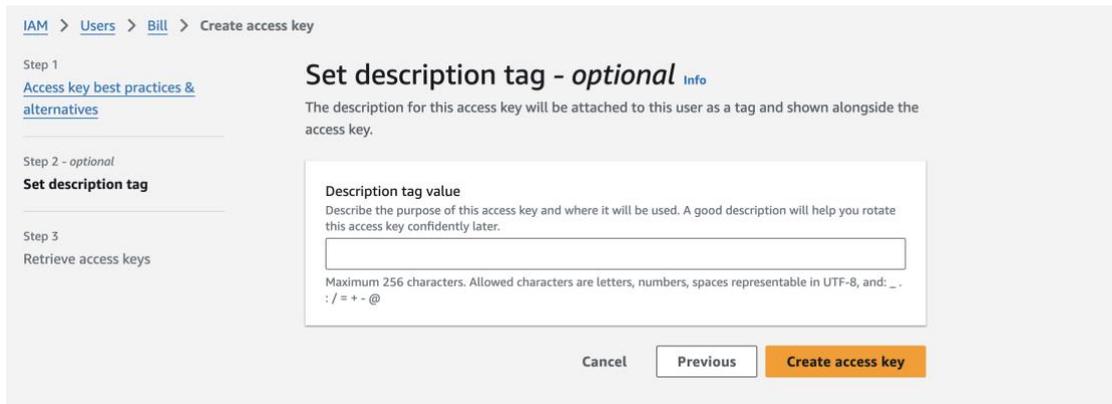
11、在 “Access keys” ， 点击 “Create access key” 。



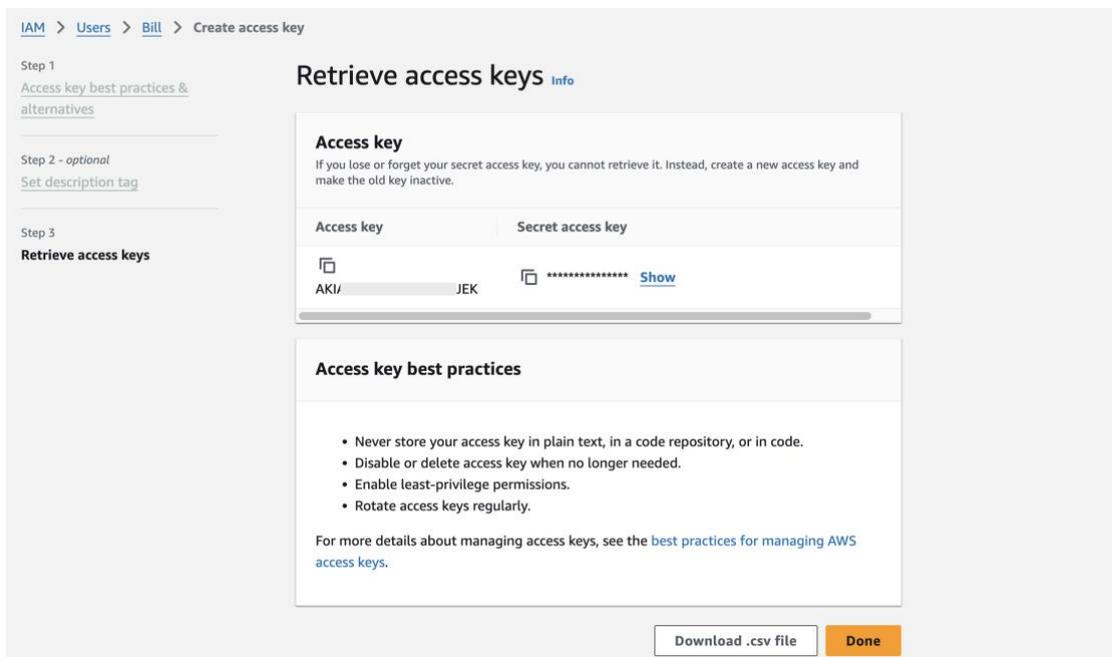
12、选择 “Local code” ， 点击 “Next” 。



13、可选择设置描述标签， 然后点击 “Create access key” 。

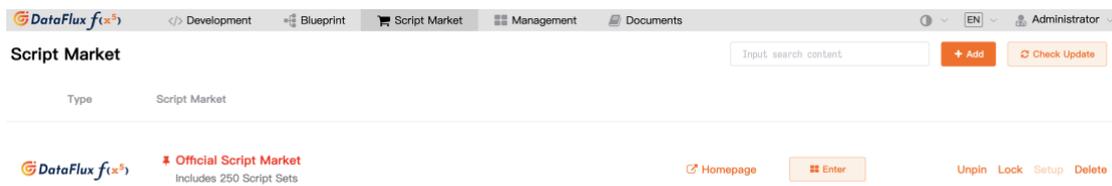


14、创建完成后，可点击“Download .csv file”下载密钥，用于稍后的 Func 配置。



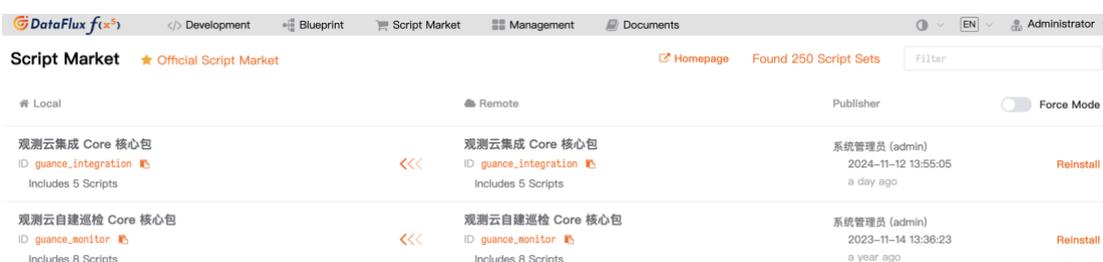
安装 AWS 云账单集成脚本

1、在已安装的 DataFlux Func 控制台，点击“Script Market” - “Official Script Market” - “Enter”，进行观测云官方脚本市场。

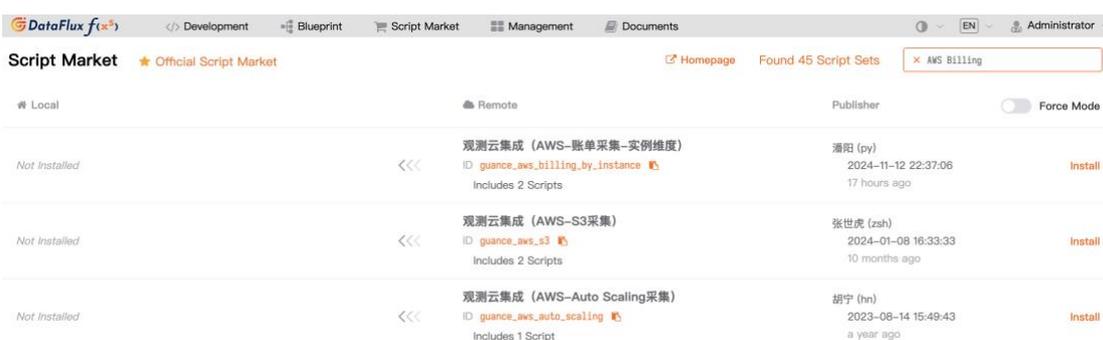


2、确认“观测云集成 Core 核心包”是否已经安装。

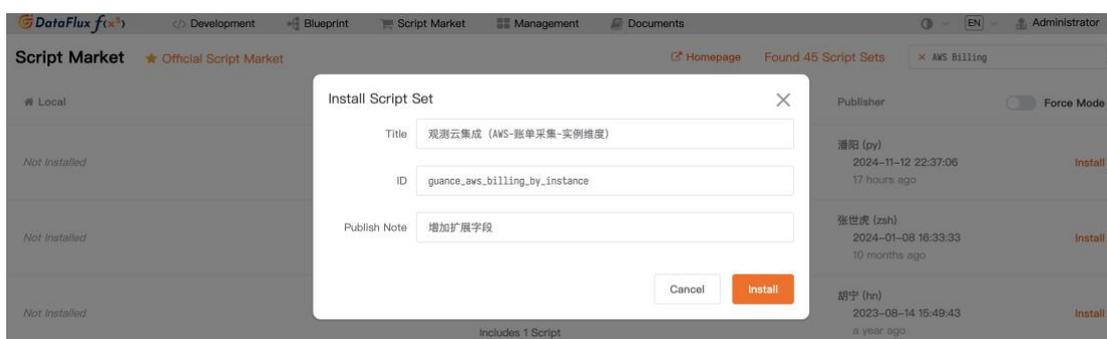
若已经安装且脚本有更新，则提示「升级」，点击「升级」即升级到最新的脚本。



3、搜索关键字“AWS Billing”，点击“Install”，安装搜索到的“观测云集成 (AWS-账单采集-实例维度)”。



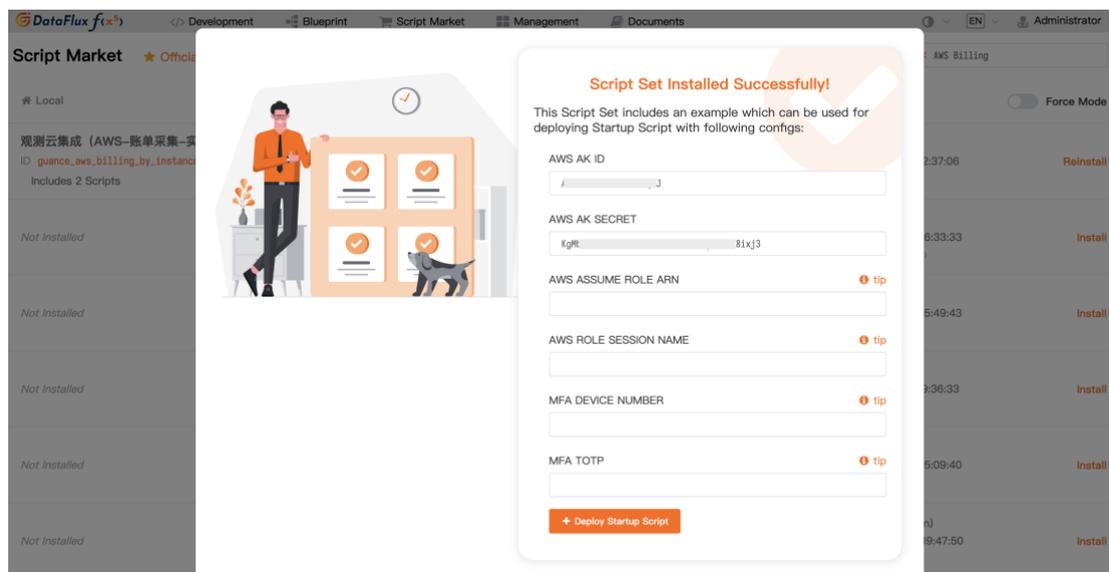
4、在弹出对话框，点击“Install”。



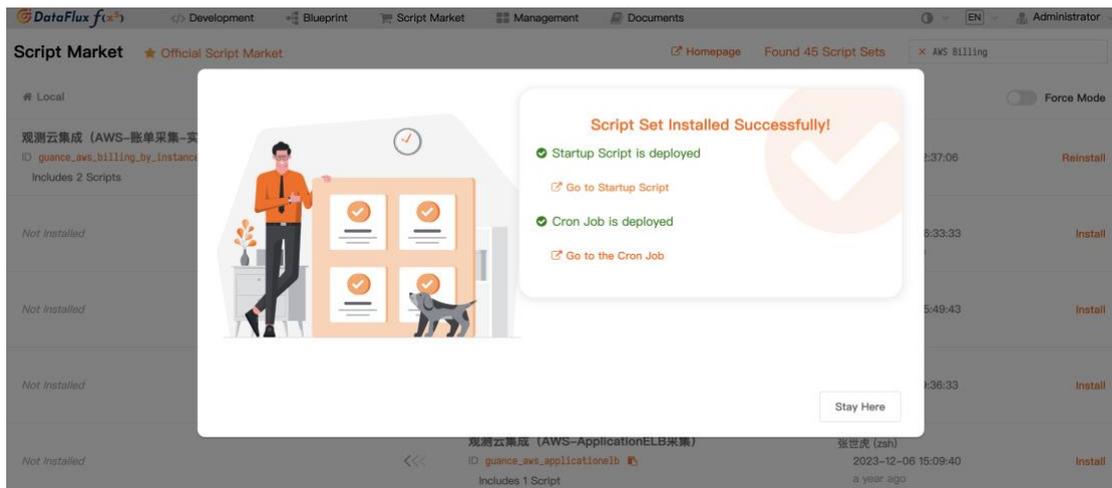
5、在弹出的对话框，输入在 AWS 创建的“AK/AKS”，点击“Deploy Startup Script”。

以下四个参数可置空：

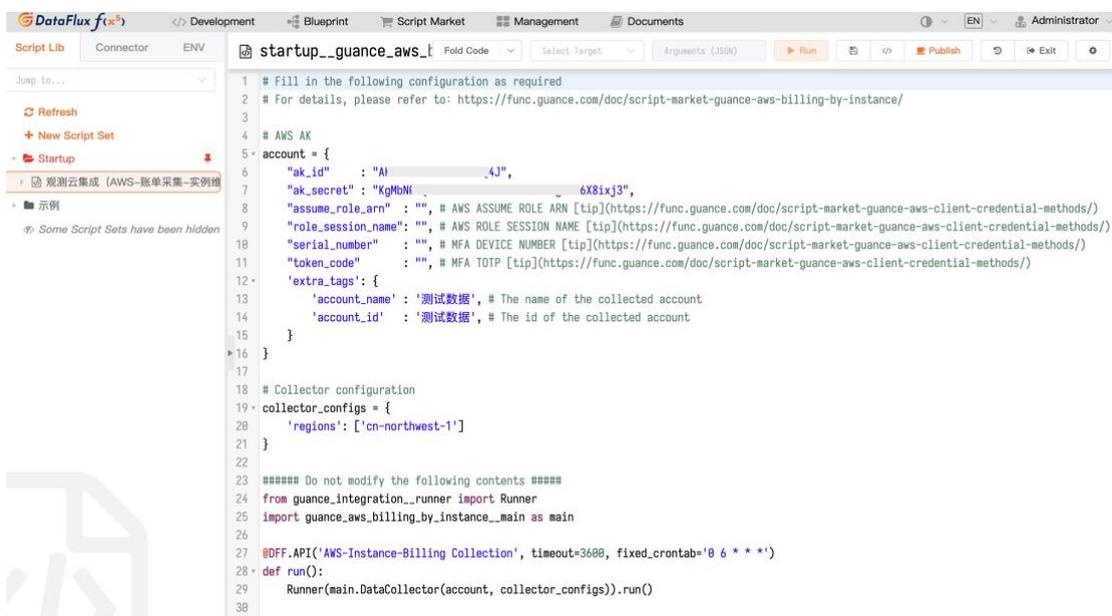
- AWS ASSUME ROLE ARN：要代入的 IAM 角色的 Amazon 资源名称 (ARN)。
- AWS ROLE SESSION NAME：为此次代入角色会话指定的名称，用于唯一标识会话。
- MFA DEVICE NUMBER：MFA 设备的序列号，通常是虚拟 MFA 设备的 ARN。
- MFA TOTP：MFA 设备生成的一次性密码 (TOTP)。



6、安装成功后，点击 “Go to Startup Script” 。



7、在“Script Lib”可以查看到安装的 AWS 账单集成脚本。



8、按照下面的提示完成脚本配置，选择“run”，点击执行按钮“Run”，确认脚本能正常运行后，点击“Publish”即完成 Func 的配置。

注意：为了保证数据的完整性，所采集的账单数据为采集日期前一天的数据。

- 1) ak_id / ak_secret: 在 AWS IAM 控制台获取的 AK 和 AKS;
- 2) 可置空 (在通过 AWS API 查询账单时，如果有设置的话需要使用以下参数来代入角色并进行多因素身份验证 (MFA)，如果不需要设置直接填写 AK 就可以);

- `assume_role_arn`: 要代入的 IAM 角色的 Amazon 资源名称 (ARN)。
- `role_session_name`: 为此次代入角色会话指定的名称, 用于唯一标识会话。
- `serial_number`: MFA 设备的序列号, 通常是虚拟 MFA 设备的 ARN。
- `token_code`: MFA 设备生成的一次性密码 (TOTP)。

3) `extra_tags`: 采集账号的名字和 id ;

4) `regions`: 账号所在地域, 可以填写多个。注意: 国内的账号所在的地域和海外的账号所在的地域不能同时在一个脚本中运行, 需要分开配置两个脚本进行数据采集;

5) `fixed_crontab`: 定时任务, 此处配置的 [6], 说明在每天 6 点自动执行脚本任务;

注意: AWS Cost Explorer API 调用收费为每个分页收费 \$0.01。

6) `guance_id`: 调用的连接器, 如果不添加, 即自动调用连接器列表第一个;

7) 输出: 点击 “Run” 返回的结果, 若报错说明需要修改脚本配置。

```

1 # Fill in the following configuration as required
2 # For details, please refer to: https://func.guance.com/doc/script-market-guance-aws-billing-by-instance/
3
4 # AWS AK
5 account = {
6     "ak_id" : "AK[REDACTED]Q4J", 1
7     "ak_secret" : "KqMb[REDACTED]X8ixi3", 1
8     "assume_role_arn" : "", # AWS ASSUME ROLE ARN [tip](https://func.guance.com/doc/script-market-guance-aws-client-credential-methods/)
9     "role_session_name" : "", # AWS ROLE SESSION NAME [tip](https://func.guance.com/doc/script-market-guance-aws-client-credential-methods/)
10 }
11 "serial_number" : "", # MFA DEVICE NUMBER [tip](https://func.guance.com/doc/script-market-guance-aws-client-credential-methods/)
12 "token_code" : "", # MFA TOTP [tip](https://func.guance.com/doc/script-market-guance-aws-client-credential-methods/)
13 'extra_tags': {
14     'account_name': 'aws-bill', # The name of the collected account 3
15     'account_id': 'aws-bill', # The id of the collected account 3
16 }
17
18 # Collector configuration
19 collector_configs = {
20     'regions': ['us-east-1','us-west-2'] 4
21 }
22
23 ##### Do not modify the following contents #####
24 from guance_integration_runner import Runner
25 import guance_aws_billing_by_instance_main as main
26
27 @DFF.API('AWS-Instance-Billing Collection', timeout=3600, fixed_crontab='0 6 * * *') 5
28 def run():
29     Runner(main.DataCollector(account, collector_configs), guance_id='aws_billing').run() 6

```

Output #2

```

#2
Executed Func: startup_guance_aws_billing_by_instance.run()
[11-13 17:00:34] [+1185ms] [1185ms] 数据将写入ID为'aws_billing'的观测云
[11-13 17:00:34] [+0ms] [1186ms] 执行第 [1] 个采集器: 'aws_billing_by_instance'
[11-13 17:00:34] [+0ms] [1186ms] 采集第 [1] 个账号...
[11-13 17:00:34] [+2ms] [1188ms] 第 [1] 个账号采集完毕, 共执行 [2毫秒], 期间调用API [0次] 7

```

效果展示

查看器

在“Cloud Billing” - “Explorers”，支持查看所有的账单明细，支持通过灵活筛选，定位关键费用。

Explorers Overview

3d 2024/11/10 17:13:13 ~ 2024/11/13 17:13:13

cloud_provider:aws

Filter

- Cloud Provider
 - aws
 - huaweicloud
 - aliyun
- Account Name
- Product Name
- Region
- Zone
- Billing cycle

Product consumption trend

Summary of Product Dimensions

Product discount trend

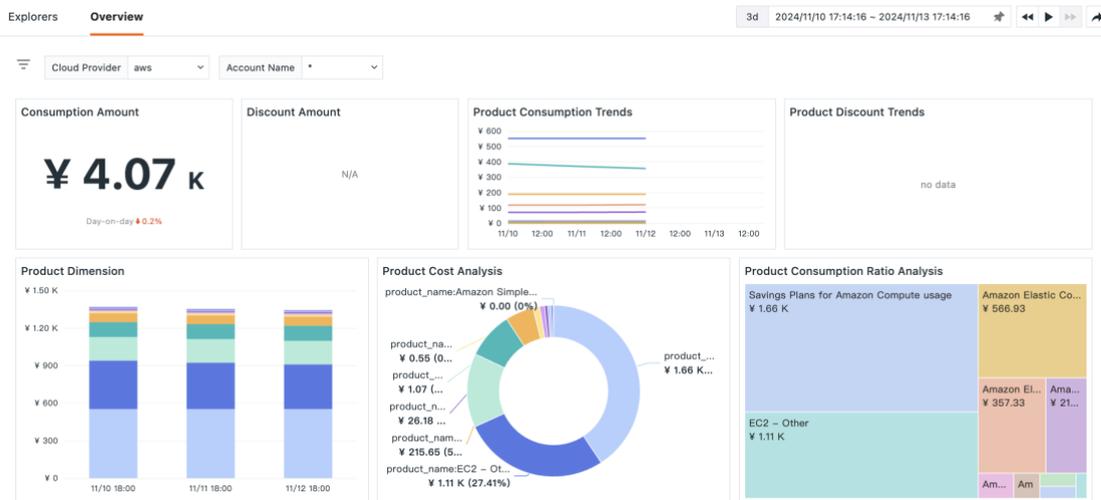
total 132 results

Billing cycle	Cloud Provider	Account Name	Product Name	Region	Instance ID	Official Amount	Amount
2024-11-11	aws	aws-china	Amazon Elast...	-	i-...	-	178.35832
2024-11-11	aws	aws-china	Amazon Cost	-	Nc	-	0.14
2024-11-11	aws	aws-china	Amazon Elast...	-	i-...	-	51.6576
2024-11-11	aws	aws-china	Amazon Elast...	-	i-...	-	14.196
2024-11-11	aws	aws-china	Amazon Secur...	-	Nc	-	0.35535454
2024-11-11	aws	aws-china	Amazon Elast...	-	i-...	-	51.6576
2024-11-11	aws	aws-china	Amazon Elast...	-	i-...	-	6.47944566
2024-11-11	aws	aws-china	Amazon Elast...	-	i-...	-	178.35832
2024-11-11	aws	aws-china	Amazon Simpl...	-	Nc	-	72.9477286

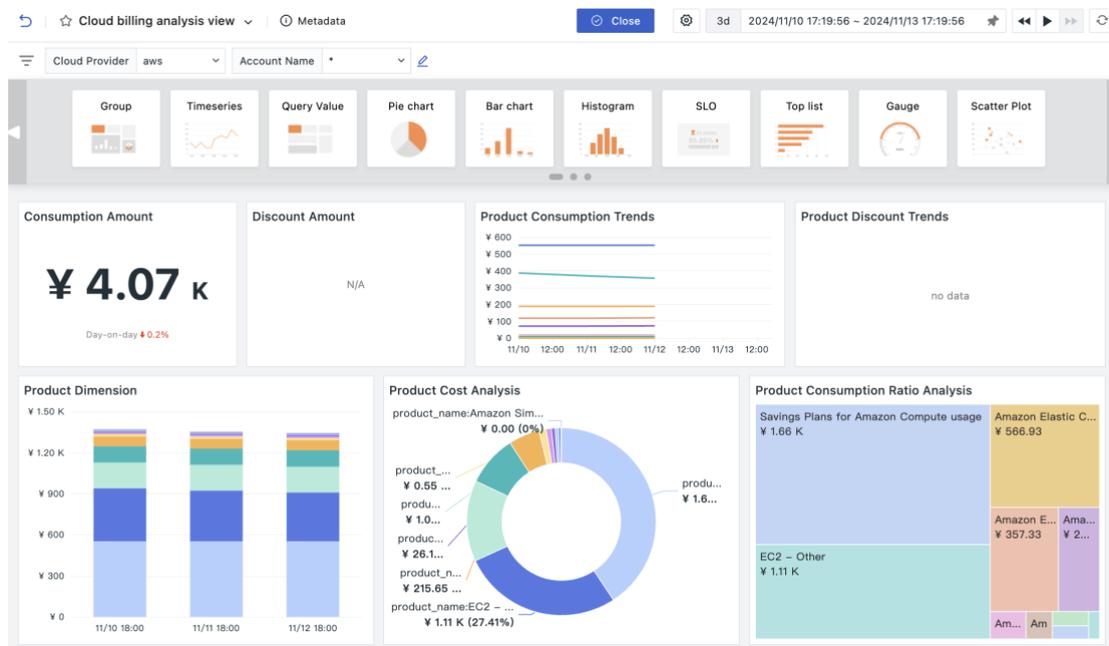
云账单分析

支持通过可视化的方式，全局了解云资源的当前使用状况及未来趋势。

- 在“Cloud Billing” - “Overview”，查看绑定账号的费用分析。



- 在场景仪表板，自定义适合的云账单视图。

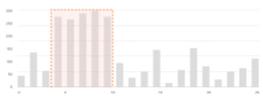
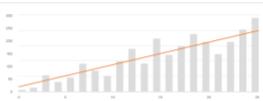
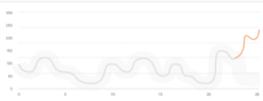


监控告警

通过设置监控告警，实时追踪各项服务的异常消费，避免意外的费用超支。

- 在 “Intelligent Monitoring” - “Cloud Billing Smart Monitoring” 开启云账单智能监控。

Intelligent Monitoring > Intelligent Monitoring Create

 Host Detection Data Range: Metric(M) Description: Automatically detects the host and finds abnormal CPU and memory conditions of the host. For example, soaring memory usage caused by memory leak	
 Logs Detection Data Range: Log(L) Description: Automatically detect exceptions in the log, including the number of logs, the number of error logs	
 APM Detection Data Range: Trace(T) Description: Automatically detects exceptions in applications, including the number of application requests, the number of erroneous requests, and request delays	
 RUM detection Data Range: RUM(R) Description: Automatically detects the performance problems of APP/Website, and find out which view pages make users have poor interactive experience. Relevant detection indicators include LCP, FID, CLS, Loading Time and so on	
 Kubernetes Smart Monitoring Data Range: Metric(M) Description: Automatically detect abnormalities in Kubernetes through intelligent algorithms. Detection indicators include total number of Pods, Pod restarts, API QPS, etc	
 Cloud Billing Smart Monitoring Data Range: Billing(B) Description: Automatically detect abnormal account billing fees in different cloud vendors through intelligent algorithms, and the detection indicators include billing fees	

- 根据提示配置云账单监控，配置完成后，点击 “Save” 。

Add Tags

1 Detection Configuration

Monitor Name

AWS Billing 11/256

Select detection range

Cloud provider aws Account Name All Product Name All

2 Event Notice

Event Content (optional)

B I H + Link + Variables

+ Advanced

Please input the event content, for example:
The current host `{{host}}` memory usage is `{{ Result | to_fixed(2) }}%`, which is out of the set range. Please keep an eye on it.
* Detect Object: `{{df_dimension}}`
* Monitor: `{{df_monitor_checker_name}}`
* Alert Strategy: `{{df_monitor_name}}`

Related issues

3 Alert configuration

Alert Strategy

Note: The event level triggered by intelligent monitoring is error

Please select v

4 Authorization

Action Authorization

Custom operation permissions (if not enabled, follow the "monitor" default permissions)

Save Cancel

- 若账单异常，即可在“Events” - “Intelligent Monitoring”查看对应的异常事件。若配置告警策略，可通过设置通知对象，把异常信息直接发送给到通知对象。