

ALIBABA CLOUD

# 阿里云

## 弹性容器实例

### 弹性容器实例 ECI公共云合集

文档版本：20210303

 阿里云

## 法律声明

阿里云提醒您在使用或阅读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。
Courier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。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[ ] 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.CLI 参考	05
1.1. 快速开始	05
1.2. 使用场景	06
1.3. 简介	08
2.常见问题	10

# 1.CLI 参考

## 1.1. 快速开始

通过命令行管理ECI实例。

ECI-Client是一个基于阿里云SDK开发的命令行客户端，通过它可以快速创建、查看和销毁您的ECI实例。

### 安装

ECI-Client目前支持Linux系统和Mac系统，暂不支持Windows系统。打开本地终端窗口，执行以下命令来安装ECI-Client。

```
sudo bash -c "$(curl -s https://eci-release.oss-cn-beijing.aliyuncs.com/install.sh)"
```

### 配置

使用ECI-Client，需要预先配置阿里云密钥，以便让ECI-Client有权限连接到阿里云，语法如下。

```
eci config set-context \  
--access-key-id XXXxxx \  
--access-secret XXXxxx \  
--region-id cn-beijing \  
--security-group-id sg-xxx \  
--v-switch-id vsw-xxx
```

#### 说明

- 如果需要VPC网络访问外网，请绑定NAT网关。具体操作，请参见[外网访问](#)。
- 如果需要外网访问到ECI实例，需要添加相应的入方向安全组规则。具体操作，请参见[添加安全组规则](#)。

如果你除了阿里云密钥之外，没有任何VPC、虚拟交换机、安全组等必要的网络资源，可以使用以下命令自动初始化网络资源。

```
eci config set-context \  
--access-key-id XXXxxx \  
--access-secret XXXxxx \  
--init
```

### 使用

大多数情况下，你可以使用与Docker相同的语法来管理ECI实例，只需要把命令行开头的 `docker` 改为 `eci`。

使用 `centos` 镜像创建一个ECI实例，命令执行成功后将输出实例ID。

```
eci -tid centos bash -c "while true; do sleep 5; date; done"
```

列出所有ECI实例，以下命令将打印每个实例的ID、名字、创建时间、状态等信息。

```
eci ps
```

## 相关文档

更多使用方式请参见手册[ECI-CLI](#)。

## 1.2. 使用场景

介绍在什么场景下应该使用ECI命令行客户端。

对于需要构建集群服务的用户而言，我们推荐通过ACK/ASK来使用ECI，这能充分发挥ECI的优势。对于较小体量用户，或者您只需要部署几个单机应用，则ECI-Client将是更快速和便捷的途径，尤其是，如果您熟悉Docker命令行，则可以零成本开始使用ECI，因为ECI-Client提供了与Docker相同的语法来管理ECI实例。本文介绍几个典型的使用场景。

### 测试ECI应用

如果你正在基于ECI开发自己的应用，那你可能需要频繁的部署ECI，以便验证自己的应用是否符合预期，这时使用ECI-Client来部署实例将是不二之选。

假设我们现在构建了一个镜像 `centos:8.1.1911`，并已经上传到了Docker镜像仓库，现在我们要测试它是否按预期工作，只需要一条命令就可以部署到ECI。

```
eci run -w 5 centos:8.1.1911 sleep 3600
```

实例启动完成后可以登录到实例容器中，假设实例ID为 `eci-uf60grb03kz2nlm23j19`。

```
eci exec -ti eci-uf60grb03kz2nlm23j19 bash
```

### 简易模型训练

我们准备了一个镜像，它内部包含了一个python程序，容器启动后，该程序会从网上下载一些数据，然后使用 `tensorflow` 库来训练这些数据，训练后的数据保存在 `/tmp/imagenet`。

```
eci run --name test-tensor \  
--type ecs.gn6i-c4g1.xlarge \  
--gpu 1 \  
-w 5 \  
--volume /tmp/imagenet \  
registry.cn-hangzhou.aliyuncs.com/eci/tensorflow:1.0 \  
python models/tutorials/image/imagenet/classify_image.py
```

如上，因为 `tensorflow` 程序会依赖GPU，所以我们通过 `--type` 选项为实例指定一个GPU类型的规格，`--gpu` 用于为实例中的容器分配GPU，`--volume` 选项用于挂载一个云盘到实例中，用于保存程序的结果数据。

该镜像体积较大，ECI需要一些时间来下载镜像，通过执行以下命令查看实例的ID，名字、运行状态等。

```
eci ps
```

等待实例启动成功后，通过以下命令可以查看程序的日志输出，假设实例ID为 `eci-uf60grb03kz2nlm23j19`。

```
eci logs eci-uf60grb03kz2nlm23j19
```

## 部署个人网站

现在我们使用ECI-Client部署一个稍微复杂的应用，一个Wordpress网站，且这个网站依赖一个MySQL数据库，我们在Docker官方镜像仓库中找到这两个相应的镜像`wordpress:4.9.8-php5.6-apache`和`mysql:5.7`。

1. 部署MySQL数据库，并指定它的根密码为 `root`，命令执行成功后将输出实例ID。

```
$ eci run -w 5 --name mysql -e MYSQL_ROOT_PASSWORD=root -tid mysql:5.7
eci-uf62qbyl7x82kbfp11fu
```

部署以后检查实例的状态，直到它的状态变为 `Running`。

```
$ eci ps
CG_ID          FIRST_IMAGE COMMAND   CREATED          STATUS   PUBLIC_IP NAME
eci-uf62qbyl7x82kbfp11fu mysql:5.7 2020-10-21T08:50:08Z Running      mysql
```

2. 获取数据库的内网IP地址，以下用到了 `jq` 命令，如果本地没有 `jq` 命令，请先安装它。

```
$ eci inspect eci-uf62qbyl7x82kbfp11fu | jq -r .ContainerGroups[0].IntranetIp
192.168.0.28
```

3. 部署Wordpress，我们需要通过环境变量 `-e` 告诉它数据库的地址、用户名和密码，最后我们通过 `-w` 选项为网站分配一个公网IP并指定它的带宽为5MB。命令执行成功后将输出实例的ID。

```
$ eci run --name wordpress \
-e WORDPRESS_DB_HOST=192.168.0.28 \
-e WORDPRESS_DB_USER=root \
-e WORDPRESS_DB_PASSWORD=root \
-w 5 \
-tid wordpress:4.9.8-php5.6-apache
eci-uf672fxow2lxc4n8pj1
```

等待它的状态变为`Running`，此时我们可以看到实例的公网IP，本示例中为 `100.100.100.2`。

```
$ eci ps
CG_ID          FIRST_IMAGE          COMMAND CREATED          STATUS PUBLIC_IP  NAME
eci-uf672fxow2lxc4n8pj1  wordpress:4.9.8-php5.6-apache  2020-10-21T09:27:19Z  Running 100.100.100.2  wordpress
eci-uf62qbyl7x82kbf11fu  mysql:5.7            2020-10-21T08:50:08Z  Running          mysql
```

4. 现在浏览器中输入 `http://100.100.100.2`，即可访问网站。

## 1.3. 简介

如何通过阿里云命令行工具 CLI管理弹性容器实例ECI。

阿里云命令行工具（Alibaba Cloud Command Line Interface）是在Alibaba Cloud SDK for Go之上构建的开源工具。您可以在命令行Shell中，使用aliyun命令与阿里云服务进行交互，管理您的阿里云资源。

阿里云CLI提供了对阿里云云产品OpenAPI的直接访问。您可以基于阿里云CLI开发Shell脚本，来管理和维护自己的阿里云资源。使用之前，确保您已经开通了要使用的云产品并已了解该产品API的使用。

目前CLI支持管理ECI实例的API：

命令	说明
CreateContainerGroup	创建容器组。
CreateImageCache	创建镜像缓存。
DeleteContainerGroup	删除容器组。
DeleteImageCache	删除镜像缓存。
DescribeContainerGroupMetric	获取容器组监控信息。
DescribeContainerGroupPrice	获取容器组价格信息。
DescribeContainerGroups	获取容器组信息。
DescribeContainerLog	获取容器组日志信息。
DescribeImageCaches	获取镜像缓存信息。
DescribeMultiContainerGroupMetric	获取多个容器组监控信息。



---

命令	说明
DescribeRegions	获取开放的地区。
ExecContainerCommand	在容器内部执行命令。
RestartContainerGroup	重启容器组。
UpdateContainerGroup	更新容器组。

## 2. 常见问题

本文汇总了使用ECI的常见问题，包括计费、实例、容器、镜像、网络、存储等问题。

- 产品计费
  - [Job任务类ECI实例如何收费？](#)
- 配额限制
  - [如何查看 vCPU和virtual node的使用配额？](#)
  - [如何处理ValueExceeded问题？](#)
- 实例与容器
  - [如何加速创建ECI实例？](#)
  - [为什么ECI控制台和监控数据显示的实例规格不一致？](#)
  - [如何避免OperationDenied.NoStock问题？](#)
  - [如何解决Back-off restarting failed container事件？](#)
  - [Pod显示已调度至virtual-kubelet，但一直未成功，怎么办？](#)
  - [创建Pod后一直Pending状态怎么办？](#)
  - [添加了虚拟节点的Kubernetes集群的kube-proxy、coredns被调度到了虚拟节点，启动失败怎么办？](#)
  - [在ASK Ingress中设置了鉴权，但不生效是什么原因？](#)
- 镜像相关
  - [是否支持私有镜像？](#)
  - [镜像缓存是否支持更新？](#)
- 网络相关
  - [ECI实例是否支持修改安全组？](#)
  - [如何从外网访问容器组？](#)
  - [是否支持端口映射？](#)
  - [集群升级后，服务IP地址ping不通怎么办？](#)
- 存储相关
  - [ECI是否可以和ECS共享NAS文件存储？](#)
  - [是否支持数据持久化？](#)
  - [如何外挂存储？](#)
  - [是否支持指定ephemeral-storage来修改临时存储？](#)
- 日志与监控
  - [为什么采集不到ECI实例日志？](#)

### Job任务类ECI实例如何收费？

Job、CronJob等任务类容器在运行完成后，ECI实例会进入运行成功（Succeeded）或者运行失败（Failed）状态。此时，ECI实例不论是否删除，都不再进行计费。

ECI实例的计费时长从下载容器镜像（docker pull）开始，至ECI实例运行结束（进入Succeeded或Failed状态）。更多信息，请参见[计费概述](#)。

## 如何查看vCPU和virtual node的使用配额？

ECI的实例数和virtual node支持的Pod数量，依赖于对应地域您的vCPU配额（quota）和已使用量。ECI的配额与ECS的配额共用，如果当前的配额无法满足您的业务需求，请[提交工单](#)申请提升配额。

在控制台查看配额的方法如下：

1. 登录[ECI控制台](#)。
2. 在顶部菜单栏左上角处，选择地域。
3. 在左侧导航栏，单击[权益配额](#)。

权益配额包含已使用量和使用上限，如下图所示。



名称	资源类型	描述	已使用/使用上限	操作
按量实例vCpu总数量上限	实例	用户可拥有的按量实例vCpu的最大数量	1/1000	<a href="#">工单申请</a>
镜像缓存总数量上限	镜像缓存	用户可拥有的镜像缓存最大数量	0/500	

## 如何处理ValueExceeded问题？

当创建ECI实例时遇到ValueExceeded问题，则表示配额不足，您可以[提交工单](#)申请提升配额。

## 如何加速创建ECI实例？

问题描述：

在标准节点上创建一个Pod只需要两到三秒，而创建ECI实例时使用了镜像缓存，Pod却用了十几秒才启动。

问题解释：

这种情况是正常的。因为在标准节点上创建Pod时，不需要单独申请资源，直接在节点上创建容器，因此时间比较快。而在创建ECI实例时，系统需要申请资源，如果指定了多可用区，还需要在库存不足时逐个尝试各个可用区是否有库存，因此时间相对会慢一些。

如果您指定了多可用区创建ECI实例，建议您把库存比较多的可用区ID放在第一个，这样可以避免由重试导致的启动延迟。

## 为什么ECI控制台和监控数据显示的实例规格不一致？

问题描述：

创建了一个0.5C1G规格的ECI实例，但是在监控数据查到的ECI实例的规格是2C2G。

问题解释：

上述现象是正常的，2C2G是虚拟机的规格，而非ECI实例的规格。如果您购买的是0.5C1G规格的ECI实例，最终也只能使用对应规格的资源。

## 如何避免OperationDenied.NoStock问题？

在创建ECI实例的时候，如果遇到OperationDenied.NoStock问题，表示当前地域和可用区的ECI实例已售罄。建议您采用多可用区、多规格的方式创建ECI实例。具体操作，请参见[多可用区创建](#)和[多规格创建](#)。

## 如何解决Back-off restarting failed container事件？

如果ECI实例事件中一直出现Back-off restarting failed container的Warning事件，一般是由于通过指定的镜像启动容器后，容器内部没有常驻进程，导致容器启动成功后立即退出，从而进行了持续的重启。

对于CentOS、busybox等基础镜像，您需要在创建容器时设置启动命令。具体操作，请参见[使用CentOS 镜像创建ECI](#)。

## Pod显示已调度至virtual-kubelet，但一直未成功，怎么办？

问题描述：

在使用虚拟节点virtual-kubelet或者Serverless Kubernetes过程中，可能会出现Pod显示已经调度到virtual-kubelet节点，但是一直没有新的事件更新的现象。这种情况下，您需要获取对应的virtual-kubelet日志，获取报错信息并进行问题排查。

The screenshot displays the console interface for an Elastic Container Instance. At the top, there's a table listing pods. One pod is highlighted with a yellow status icon and the text 'PodScheduled: True'. Below this, the 'PodScheduled' event is expanded, showing its status as 'True' and the time as '2019-08-23 15:11:27'. At the bottom, another event is shown with the message 'Successfully assigned vk/nginx-gpu-demo-1-6c769bdf6-5kjpj to virtual-kubelet'.

### 说明

如果有新的事件产生，您可以先根据产生的事件进行报错分析和处理。

解决方法：

1. 在容器服务控制台的集群页面，单击目录集群操作列下的更多>通过Cloud Shell登录集群。
2. 运行以下命令获取virtual-kubelet Pod名称。

```
kubectl -n kube-system get pods
```

```

shell@Alicloud:~$ kubectl -n kube-system get pods
NAME                                     READY   STATUS    RESTARTS   AGE
ack-virtual-node-affinity-admission-controller-b877f9dcb-tsz78  1/1     Running   0           3d21h
ack-virtual-node-controller-5664954b96-zhrcc  1/1     Running   0           3d21h
alibaba-log-controller-6/d6rd6558-g69t6      1/1     Running   0           35d
alicloud-application-controller-5c9d6f9ff8-w2nkr  1/1     Running   0           35d
alicloud-disk-controller-65c588684d-b2rfs      1/1     Running   0           35d
alicloud-monitor-controller-99d88f745-mc6dk  1/1     Running   0           35d
aliyun-acr-credential-helper-7946599dd7-hmzv2  1/1     Running   0           35d
coredns-7d577ff8d7-7sn4z                     1/1     Running   0           35d
coredns-7d577ff8d7-c4rr9                     1/1     Running   0           35d
flexvolume-c7c6c                             1/1     Running   0           35d
flexvolume-lnpvr                             1/1     Running   0           35d
flexvolume-z24nc                             1/1     Running   0           35d
kube-proxy-worker-4jjl8                      1/1     Running   0           35d
kube-proxy-worker-7vzp9                      1/1     Running   0           35d
kube-proxy-worker-kls4w                      1/1     Running   0           35d
logtail-ds-gkplj                             1/1     Running   0           35d
logtail-ds-jvxx4                             1/1     Running   0           35d
logtail-ds-snczh                             1/1     Running   0           35d
metrics-server-6b46f99d99-mhn7n             1/1     Running   0           35d
nginx-ingress-controller-68c56cd9b6-gjvts     1/1     Running   0           35d
nginx-ingress-controller-68c56cd9b6-wf2ht     1/1     Running   0           35d
terway-9lstv                                 2/2     Running   0           35d
terway-q5lqc                                 2/2     Running   0           35d
terway-wszqg                                 2/2     Running   0           35d
tiller-deploy-5c9c59b4dc-96jt6              1/1     Running   0           35d
virtual-kubelet-autoscaler-76479fdcf-lvmwz    1/1     Running   0           3d21h
shell@Alicloud:~$

```

- 运行以下命令获取对应Pod的日志信息。其中ack-virtual-node-controller-xxxxxxx请替换为步骤2获取的Pod名称。

```

kubectl -n kube-system logs ack-virtual-node-controller-xxxxxxx

virtual-kubelet-autoscaler-76479fdcf-lvmwz    1/1     Running   0           3
shell@Alicloud:~$ kubectl -n kube-system logs ack-virtual-node-controller-5664954b96-zhrcc

```

- 根据最新的日志报错进行处理，或者提供RequestID和报错信息，发起工单处理流程。

```

time="2019-08-23T07:11:30Z" level=warning msg="requesting \"ok/nginx-gpu-demo-1-fc769bdf6-5klpj\" due to failed sync: failed to sync pod \"ok/nginx-gpu-demo-1-fc769bdf6-5klpj\" in the provider: SDK.ServerError\n\nerrorCode: InternalError\n\nrecommendations: [\"requestId: 960B5E6-D421-4EB1-BE0E-C03D005332F\", \"message: An unknown error occurred while processing your request.\"] namespace= node=virtual-kubelet operatingSystem=Linux provider=alibabacloud
time="2019-08-23T07:11:30Z" level=warning msg="skipping update of pod nginx-gpu-demo-1-fc769bdf6-5klpj as pod updates are not supported" namespace= node=virtual-kubelet operatingSystem=Linux provider=alibabacloud
shell@Alicloud:~$

```

## 创建Pod后一直Pending状态怎么办？

问题描述：

创建Pod后几个小时内，一直处于为Pending状态。查看Pod事件列表或实例事件列表，显示在挂载Volume时，由于连接API Server超时导致失败。

解决方法：

该问题通常是由于实例与API Server网络不通导致，您可以按以下思路排查：

- 检查Pod与集群的API Server是否在同一个专有网络VPC中。
- 如果为集群SLB添加了访问控制，请确保已将Pod所处网段加入到访问控制列表中。

## 添加了虚拟节点的Kubernetes集群的kube-proxy、coredns被调度到了虚拟节点，启动失败怎么办？

kube-proxy、coredns调度的时候是忽略taints的，所以有可能被调度到虚拟节点上。此时，您可以编辑kube-proxy、coredns的yaml解决问题，需要增加的yaml如下：

```
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: type
          operator: NotIn
          values:
            - virtual-kubelet
```

## 在ASK Ingress中设置了鉴权，但不生效是什么原因？

问题描述：

在 `nginx-ingress` 中设置了解析 `nginx.ingress.kubernetes.io/auth-url` ，但不生效。

问题解释：

ASK中基于SLB的Ingress Controller，不支持设置鉴权URL。

ACK是支持的。

## 是否支持私有镜像？

支持私有镜像。

支持使用阿里云镜像仓库搭建私有镜像，也支持您自己搭建的镜像仓库。

## 镜像缓存是否支持更新？

目前镜像缓存不支持修改更新，建议您重新创建镜像缓存后，再删除原有镜像缓存。

## ECI实例是否支持修改安全组？

ECI实例不支持修改安全组，如果您想要变更安全组，需要重新创建ECI实例。

## 如何从外网访问容器组？

如果您需要ECI实例能够访问外网，或者被外网访问，您需要为其配置EIP或者NAT网关。更多信息，请参见[外网访问](#)。

## 是否支持端口映射？

ECI实例不支持端口映射。

在同一个VPC网络中，您可以通过ECI实例的IP+Port进行访问（默认会将容器的端口开放）。

如果你需要支持外网访问，您可以配置EIP或者NAT网关。更多信息，请参见[外网访问](#)。

## 集群升级后，服务IP地址ping不通怎么办？

在2020年10月之前，每个Service IP会对应一张虚拟网卡，所以能ping通。但在2020年10月份之后，为了优化高并发，Service IP只存在ipvs的规则中，所以无法ping通（ipvs是基于IP+Port做数据转发的，所以ping包无法做转发）。

## ECI是否可以和ECS共享NAS文件存储？

支持ECI与ECS共享NAS文件存储，您可以通过配置挂载点进行NAS文件系统的挂载。具体操作，请参见[挂载NAS](#)。

## 是否支持数据持久化？

支持数据持久化。容器组支持创建有状态应用，您可以在创建容器组时挂载数据卷，将数据写到数据卷上，实现数据持久化。具体操作，请参见[数据卷](#)。

## 如何外挂存储？

针对一些在ECI实例运行期间对磁盘有较高的IO要求、同时有大量的临时数据需要存放的业务，如运行时产生的log文件，建议采用外置的数据卷来外挂存储。

您可以根据业务需求的场景来选择创建ECI实例外挂的存储（数据卷类型为FlexVolume）：

- [NAS](#)
- [云盘](#)

## 是否支持指定ephemeral-storage来修改临时存储？

ECI不支持指定ephemeral-storage参数。默认提供40 GiB的临时存储，如果您需要更大的临时存储，可以采用以下方式：

1. 创建镜像缓存，并指定需要的存储大小。
2. 使用该镜像缓存创建ECI。

创建成功即可获得与该镜像缓存大小相同的临时存储。

## 为什么采集不到ECI实例日志？

如果您在Pod中设置了日志服务的环境变量 `aliyun_logs_{Logstore名称}`，但在日志服务中却没有找到ECI实例的日志，可能是以下原因导致：

- ECI实例运行时间太短

ECI实例启动后，如果业务容器在20秒内执行完毕，这种情况下，日志服务可能还没有来得及收集日志，容器就已经退出且日志相关的Volume已经被卸载，因此日志服务无法收集到日志。

- 收集路径（Path）错误

首次在Pod中指定环境变量收集日志时，ECI会自动在日志服务中创建Logstore和Path，后续再次创建Pod时也只能使用该Path。如果使用了其它Path，则会导致日志服务收集不到日志。如果您想要更换Path，可以将Logstore一起更换，此时ECI会自动在日志服务中创建新的Logstore。