

阿里云

弹性容器实例 存储

文档版本：20220527

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 <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.数据卷	05
2.使用云盘作为数据卷	10
3.自定义临时存储空间大小	14
4.设置自动驱逐临时存储空间不足的Pod	18
5.存储	20
5.1. ACK使用云存储	20
5.2. ASK使用云存储	21
5.3. 非PVC方式	23
5.3.1. 挂载云盘	23
5.3.2. 挂载NAS	26
5.3.3. 挂载OSS	28
5.4. 挂载stdlog (FlexVolume)	31

1.数据卷

容器中的文件在磁盘上是临时存放的，这给容器中运行的特殊应用程序带来一些问题。首先，当容器崩溃时，容器会以干净的状态重建，导致容器中的文件将会丢失。其次，当在一个ECI中同时运行多个容器时，常常需要在这些容器之间共享文件。ECI支持通过数据卷的方式来解决这两个问题。

目前ECI支持4种类型的数据卷：

- ConfigFile
- EmptyDir
- NFS
- FlexVolume

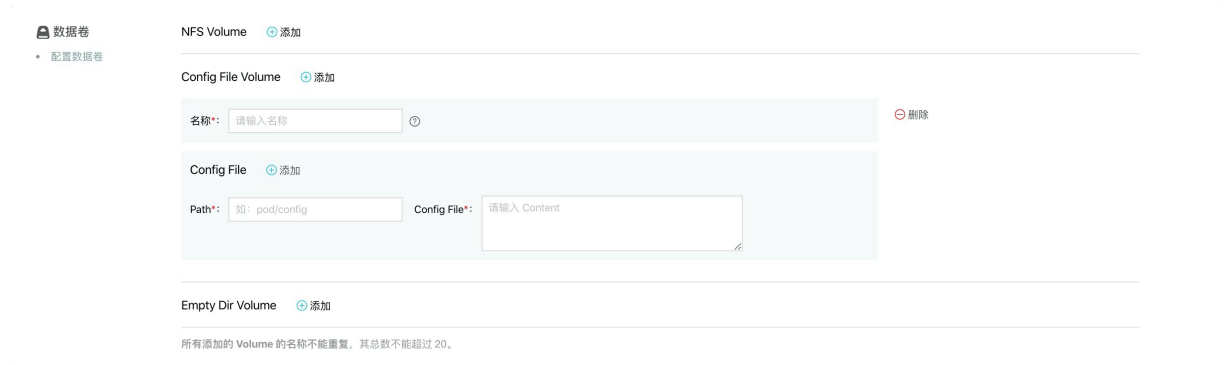
使用数据卷时，需要先声明具体的数据卷，然后挂载到ECI中。

控制台

- ConfigFile

ConfigFile提供了向ECI注入配置数据的方法。

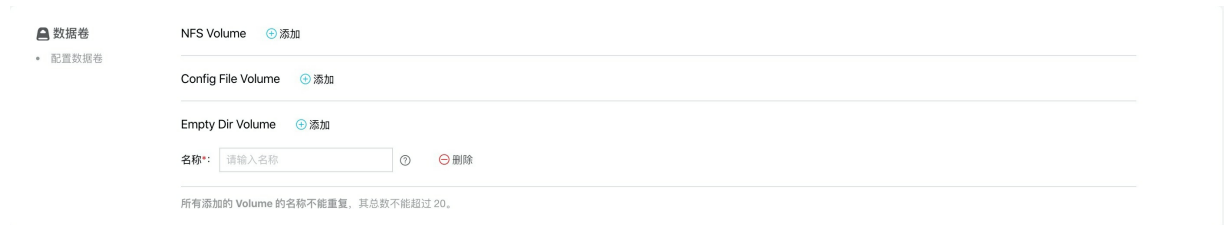
通过控制台声明ConfigFile（只需以明文形式填入配置数据，控制台会自动进行Base64编码）：



- EmptyDir

EmptyDir可以被同一个ECI中的所有容器访问，因此您可以使用EmptyDir在同一个ECI的不同容器之间共享数据。当ECI删除后，EmptyDir上保存的数据也会一并删除。

通过控制台声明EmptyDir：



- NFS

NFS数据卷能将NFS（网络文件系统）挂载到您的ECI中。不像EmptyDir那样会在删除ECI的同时也会被删除，NFS数据卷的内容在删除ECI时会被保存，数据卷只是被卸载掉了。这意味着NFS数据卷可以被预先填充数据，并且这些数据可以在ECI之间“传递”。

挂载点							内核缺陷说明	如何挂载文件系统	挂载问题排查	开机自动挂载	添加挂载点	
挂载点类型	VPC	交换机	挂载地址	挂载命令	权限组	状态	操作					
专有网络	vpc-m5e22mviwgt	vsw-m5ekw0g1n	121.11. cn-beijing.nas.aliyuncs.com	V3 Mount: sudo mount -t nfs -o vers=3,nolock,proto=tcp,noresvport 13 wqd11.cn-beijing.nas.aliyuncs.com:/mnt V4 Mount: sudo mount -t nfs -o vers=4,minorversion=0,noresvport 14 wqd11.cn-beijing.nas.aliyuncs.com:/mnt	VPC默认权限组 (全部允许)	可用	修改权限组	已挂载客户端	激活	禁用	删除	

说明

需要先准备好NFS，推荐使用阿里云NAS，可以前往NAS控制台进行创建，并完成挂载点添加。ECI和NAS需要在同一个VPC网络中。若需要使用其他NFS服务，需要ECI具备外网访问能力。

通过控制台声明 NFS：

数据卷

配置数据卷

NFS Volume 添加

名称: 请输入名称

Server: 如: yournfsserver.cn-hangzhou.nas.aliyuncs.com 选择

Path: 如: /data/container

Read Only

删除

Config File Volume 添加

Empty Dir Volume 添加

所有添加的 Volume 的名称不能重复，其总数不能超过 20。

FlexVolume

FlexVolume是自Kubernetes 1.2版本以来一直存在的out-of-tree插件接口，通过FlexVolume可以自定义扩展存储驱动的方式。

目前ECI通过alicloud/disk存储驱动支持了云盘的使用，后续将会支持更多的驱动类型。

通过FlexVolume使用云盘

ECI支持新建和使用已有云盘两种声明方式来使用云盘，并且只能声明使用一个云盘。

- 新建：根据设置的云盘大小，跟随ECI实例的创建和销毁，动态的创建和销毁云盘，适用于运行期间对磁盘有较高IO要求的场景。
- 使用已有云盘：选择已有的云盘，并且写入云盘的数据可以在ECI销毁时依然保留，适用于MySQL、Redis等场景。

通过控制台选择已有云盘：

数据卷

配置数据卷

NFS Volume 添加

Config File Volume 添加

Empty Dir Volume 添加

FlexVolume

名称: disk

驱动类型: 云盘

云盘实例 ID: d-14n8qfmf8ozdpst6kp 重新选择云盘实例 ID

删除

云盘实例ID 和 云盘大小 二选一即可，仅支持添加 一个 云盘。

所有添加的 Volume 的名称不能重复，其总数不能超过 20。

通过控制台新建云盘：

数据卷

配置数据卷

NFS Volume

添加

Config File Volume

添加

Empty Dir Volume

添加

FlexVolume

名称: disk

驱动类型: 云盘

云盘大小: 20 GIB

删除

云盘实例ID 和 云盘大小 二选一即可，仅支持添加一个云盘。

所有添加的 Volume 的名称不能重复，其总数不能超过 20。

OpenAPI方式

- 通过OpenAPI声明ConfigFile。

```
Volume.1.Name=configfiledemo
Volume.1.Type=ConfigFileVolume #参数为固定值。
Volume.1.ConfigFileVolume.ConfigFilePath.1.Content=bGllbWk=
Volume.1.ConfigFileVolume.ConfigFilePath.1.Path=configpath
```

注意

需要先将配置数据进行Base64编码。

- 通过OpenAPI声明EmptyDir。

```
Volume.1.Name=emptydirdemo
Volume.1.Type=EmptyDirVolume #参数为固定值。
```

注意

- 参数中增加Volume.1.EmptyDirVolume.Medium=Memory。
- Memory的EmptyDir使用的是内存，注意内存大小，用超过了容易oom。
- Memory的EmptyDir的大小为实例规格的50%，暂时无法更改。
- 不使用Memory的EmptyDir不会占用系统内存。

- 通过OpenAPI声明NFS 。

```
Volume.1.Name=nfsdemo
Volume.1.Type=NFSVolume #参数为固定值。
Volume.1.NFSVolume.Path=/share
Volume.1.NFSVolume.Server=3f9cd4a596-naw76.cn-shanghai.nas.aliyuncs.com
Volume.1.NFSVolume.ReadOnly=False
```

- 通过OpenAPI使用已有云盘（静态云盘）：

```
Volume.1.Name=flexvolumedemo
Volume.1.Type=FlexVolume #参数为固定值。
Volume.1.FlexVolume.Driver=alicloud/disk #参数为云盘的驱动。
Volume.1.FlexVolume.FsType=ext4 #支持ext3、ext4、xfs
Volume.1.FlexVolume.Options={"volumeId":"d-bp1j17ifxfasvts3tf40"} #d-bp1j17ifxfasvts3tf40
为已经存在的云盘实例id。
```

通过OpenAPI新建云盘（动态云盘）：

```
Volume.1.Name=flexvolumedemo
Volume.1.Type=FlexVolume #参数为固定值。
Volume.1.FlexVolume.Driver=alicloud/disk #参数为云盘的驱动。
Volume.1.FlexVolume.FsType=ext4
Volume.1.FlexVolume.Options={"volumeSize":"20", "performanceLevel":"PL1"} #volumeSize指定云
盘大小，取值范围20-500，单位为GiB；performanceLevel为云盘性能等级，可选值为：PL1、PL2、PL3。
```

④ 说明

- 动态创建的云盘默认随ECI实例一起创建，一起销毁。
- 当volumeId为空，volumeSize字段为必填，当2个字段都有值，则volumeSize将被忽略。
- 动态创建的云盘默认随ECI实例一起创建，一起销毁。
- 动态创建的云盘默认采用：ESSD云盘，暂不支持指定云盘种类。
- 单ECI实例最大支持挂载6块云盘作为数据卷。
- 同一云盘同一时刻只能挂载在一个ECI/ECS实例中。
- 如果给定的云盘未格式化，则系统会自动格式化为FsType指定的类型，目前仅支持ext3、ext4、xfs。
- volumeSize指定云盘大小，取值范围20-500，单位为GiB。

数据卷的挂载

创建了数据卷后，需要挂载到ECI实例中进行使用。

所有的数据卷挂载到容器目录后，都会覆盖原有的内容，尤其是EmptyDir会直接清空原内容，所以VolumeMount参数的挂载目录一定要慎重。

- 通过控制台挂载数据卷：

- 通过OpenAPI挂载数据卷：


```
Container.1.VolumeMount.1.Name=volumename #参数为声明的Volume名称。  
Container.1.VolumeMount.1.MountPath=/volume #参数为挂载的路径。  
Container.1.VolumeMount.1.ReadOnly=False
```

2.使用云盘作为数据卷

ECI支持云盘作为数据卷，其采用FlexVolume的方式将阿里云云盘挂载到容器内：FlexVolume.Driver的值设定为：alicloud/disk，标识采用阿里云云盘。

其中数据卷类型为：

- FlexVolume

OpenAPI直接调用方式

对于openapi方式 `CreateContainerGroup` 接口提供了更加灵活的使用云盘作为后端存储的方法：静态&动态。

静态云盘使用场景：

已经购买了云盘实例的情况，希望写入云盘的数据可以在ECI销毁的时候依然保留，如MySQL、Redis等，此时可以通过 `Options.volumeId` 参数传入云盘实例ID。

```
Volume.N.Name=my-disk-volume
Volume.N.Type=FlexVolume
Volume.N.FlexVolume.Driver=alicloud/disk
Volume.N.FlexVolume.FsType=ext4
Volume.N.FlexVolume.Options={"volumeId":"d-2zebuamrpar7xnj****"}
```

动态云盘使用场景：

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，实例销毁后不需要保存数据，如运行时产生的log文件；可以通过 `Options.volumeSize` 传入希望随实例一起创建的云盘容量。随实例一起创建云盘同时支持性能等级、加密等参数，更多信息，请参见[Options参数说明](#)。

```
Volume.N.Name=my-disk-volume
Volume.N.FlexVolume.Driver=alicloud/disk
Volume.N.FlexVolume.FsType=ext4
Volume.N.FlexVolume.Options={"volumeSize":"200"}
```

说明

如果给定的云盘未格式化，则系统会自动格式化为 `FsType` 指定的类型，目前仅支持ext3、ext4、xfs

注意

- 动态创建的云盘默认随ECI实例一起创建，一起销毁。
- 当`volumeId`为空，`volumeSize`字段为必填，当2个字段都有值，则`volumeSize`将被忽略
- 动态创建的云盘默认随ECI实例一起创建，一起销毁。
- 动态创建的云盘默认采用：ESSD云盘，暂不支持指定云盘种类。
- 单ECI实例最大支持挂载6块云盘作为数据卷。
- 同一云盘同一时刻只能挂载在一个ECI/ECS实例中。

示例

- 通过PythonSDK（版本>=1.0.8）创建应用Mysql，其中数据文件保存在 `d-2zefzudpm3649e****` 的云盘中

```
from aliyunsdkeci.request.v20180808.CreateContainerGroupRequest import CreateContainerGroupRequest
request = CreateContainerGroupRequest()
request.set_SecurityGroupId(secureGroup-xxx)
request.set_VSwitchId(vSwitch-xxx)
request.set_ContainerGroupName(name-xxx)
request.set_EipInstanceId(eip-xxx)
request.set_RestartPolicy('Always')
#volumes,第一个为云盘卷,用于保存Mysql数据文件;第二个为ConfigFileVolume,保存配置文件
volume1 = {
    'Name': 'mysql-data-volume',
    'Type': 'FlexVolume',
    'FlexVolume.Driver': 'alicloud/disk',
    'FlexVolume.FsType': 'ext4',
    'FlexVolume.Options': '{\"volumeId\": \"d-2zefzudpm3649e****\"}',
}
#Content需要BASE64编码
configFileToPath = {
    'Path': 'my.cnf',
    'Content': 'aGVsbG8gd29ybGQ=',
}
volume2 = {
    'Name': 'mysql-config-volume',
    'Type': 'ConfigFileVolume',
    'ConfigFileVolume.ConfigFileToPaths': [configFileToPath],
}
#安装volume到containerGroup
request.set_Volumes([volume1, volume2])
#volumeMounts,挂载containerGroup安装的数据卷到容器的目录
volume_mount1 = {
    'Name': 'mysql-data-volume',
    'MountPath': '/var/lib/mysql',
    'ReadOnly': False,
}
volume_mount2 = {
    'Name': 'mysql-config-volume',
    'MountPath': '/etc/mysql/conf.d/',
    'ReadOnly': False,
}
#environment variable
env = {
    'Key': 'MYSQL_ROOT_PASSWORD',
    'Value': '123456',
}
#container ports
port = {
    'Protocol': 'TCP',
    'Port': 3306,
}
```

```

# containers
container = {
    'Image': 'mysql:5.7',
    'Name': 'mysql',
    'Cpu': 0.5,
    'Memory': 1.0,
    'ImagePullPolicy': 'Always',
    'VolumeMounts': [volume_mount1, volume_mount2],
    'EnvironmentVars': [env],
    'Ports': [port],
}
request.set_Containers([container])
response = eciClient.do_action_with_exception(request)
print response

```

virtual kubelet方式

具体操作，请参见云盘（FlexVolume）。

Options参数说明

名称	类型	必选	示例值	描述
volumeId	String	否	d-2zebuamrpar7xnjf****	已有云盘实例ID。
volumeSize	Integer	否	100	单盘容量（GiB），取值范围为：20~32768。动态创建云盘必填参数。
performanceLevel	String	否	PL1	性能等级。更多信息，请参见云盘性能，动态创建云盘可选参数，默认值：PL1。
deleteWithInstance	Boolean	否	false	是否随实例释放，动态创建云盘可选参数，默认值为：true。

名称	类型	必选	示例值	描述
encrypted	Boolean	否	false	是否加密云盘，动态创建云盘可选参数，默认值：false。 更多信息，请参见 加密云盘 。
kmsKeyId	否	否	2ea91e17-6de6-4b4f-a3ea-0e382cc8****	云盘使用的KMS密钥ID。动态创建云盘可选参数。

3. 自定义临时存储空间大小

ECI实例默认提供20 GiB的免费存储空间，如果该存储空间大小无法满足您的需求，您可以自定义增加临时存储空间大小。本文介绍如何自定义设置临时存储空间大小。

背景信息

每个ECI实例默认提供20 GiB的免费存储空间，如果您的存储需求大于20 GiB时，可以自定义增加临时存储空间大小。新增加的临时存储空间按照ESSD PL1类型的块存储按量价格进行计费。不同地域的价格可能不同，更多信息，请参见[块存储定价页](#)。

Kubernetes方式

Kubernetes场景下，支持通过以下方式自定义设置临时存储空间大小：

- 方式一：添加annotation

添加annotations可以指定临时存储空间大小，对应的字段为k8s.aliyun.com/eci-extra-ephemeral-storage，设置的值即为增加的临时存储空间大小。

- 方式二：设置emptyDir

自定义设置emptyDir Volume可以增加临时存储空间。如果使用的emptyDir Volume大小不超出20 GiB，则不会增加临时存储空间；如果超出了20 GiB，则超出的部分即为临时存储空间大小。

- 方式三：设置容器的resource

在容器的resource中可以声明ephemeral-storage相关request，如果声明的ephemeral-storage大小不超出20 GiB，则不会增加临时存储空间；如果超出了20 GiB，则超出的部分即为临时存储空间大小。

② 说明

三种方式均可以设置临时存储空间大小，但是语义有所差异：

- 方式一为limit语义，创建的临时存储空间大小是准确的。
- 方式二和方式三为request语义，创建临时存储空间时，可能还需要考虑镜像缓存因素，即实际创建的临时存储空间大小可能和request声明的不一致。

配置示例如下：

1. 准备YAML文件。

假设YAML文件命名为extra-storage.yaml，三种方式对应的YAML文件内容示例如下：

- 示例一：添加annotation

```
apiVersion: v1
kind: Pod
metadata:
  name: test
  annotations:
    k8s.aliyun.com/eci-extra-ephemeral-storage: "30Gi" #自定义设置临时存储空间大小
spec:
  containers:
  - name: nginx
    image: nginx:latest
    imagePullPolicy: IfNotPresent
    restartPolicy: Always
```

◦ 示例二：设置emptyDir

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - image: nginx:latest
    name: nginx
    volumeMounts:
    - mountPath: /extra-storage
      name: cache-volume
  volumes:
  - name: cache-volume
    emptyDir:
      sizeLimit: 50Gi #自定义设置emptyDir Volume的大小，超出20 GiB的部分为临时存储空间
```

◦ 示例三：设置容器的resource

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - name: nginx
    image: nginx:latest
    imagePullPolicy: IfNotPresent
    resources:
      requests:
        ephemeral-storage: 50Gi #在容器中的resource中，声明ephemeral-storage相关request，超出20 GiB的部分为临时存储空间
```

2. 创建Pod。

```
kubectl apply -f extra-storage.yaml
```

3. 查看临时存储空间是否生效。

- 查看临时存储空间大小。

```
kubectl describe pod test | grep k8s.aliyun.com/eci-extra-ephemeral-storage
```

返回结果如下，已增加30 GiB的临时存储空间。

```
k8s.aliyun.com/eci-extra-ephemeral-storage: 30Gi
```

- 通过exec命令查看容器的文件系统分区是否扩容成功。

```
kubectl exec -it test -- sh
```

```
df -h
```

返回结果如下，存储空间总共约70 GiB，其中20 GiB为系统需要占用的空间，实际可用空间为临时存储空间30 GiB加上默认可用的20 GiB，约50 GiB。

Filesystem	Size	Used	Avail	Use%	Mounted on
overlay	69G	4.7G	61G	8%	/
tmpfs	64M	0	64M	0%	/dev
tmpfs	1.9G	0	1.9G	0%	/sys/fs/cgroup
/dev/root	69G	4.7G	61G	8%	/etc/hosts
shm	64M	0	64M	0%	/dev/shm
tmpfs	1.9G	12K	1.9G	1%	/run/secrets/kubernetes.io/serviceaccount
tmpfs	1.9G	0	1.9G	0%	/proc/acpi
tmpfs	1.9G	0	1.9G	0%	/sys/firmware

OpenAPI方式

调用CreateContainerGroup接口创建ECI实例时，您可以通过EphemeralStorage参数来设置临时存储空间大小。EphemeralStorage的参数说明如下表所示。更多信息，请参见[CreateContainerGroup](#)。

名称	类型	是否必选	示例值	描述
EphemeralStorage	Integer	否	50	临时存储空间大小。单位：GiB。

如果您使用SDK，Python SDK的示例如下：


```
#!/usr/bin/env python
#coding=utf-8

from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkkeci.request.v20180808.CreateContainerGroupRequest import CreateContainerGroupRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')

request = CreateContainerGroupRequest()
request.set_accept_format('json')

request.set_SecurityGroupId("sg-uf6biempwqvodk7*****")
request.set_VSwitchId("vsw-uf6mhqg2wiq9iifhn*****")
request.set_ContainerGroupName("test")
request.set_Containers([
    {
        "Image": "nginx",
        "Name": "nginx"
    }
])
request.set_EphemeralStorage(50) # 自定义设置临时存储空间大小

response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

4. 设置自动驱逐临时存储空间不足的Pod

本文介绍Kubernetes场景下如何设置自动驱逐临时存储空间不足的ECI Pod，避免因Pod临时存储空间不足而影响业务运行。

适用场景

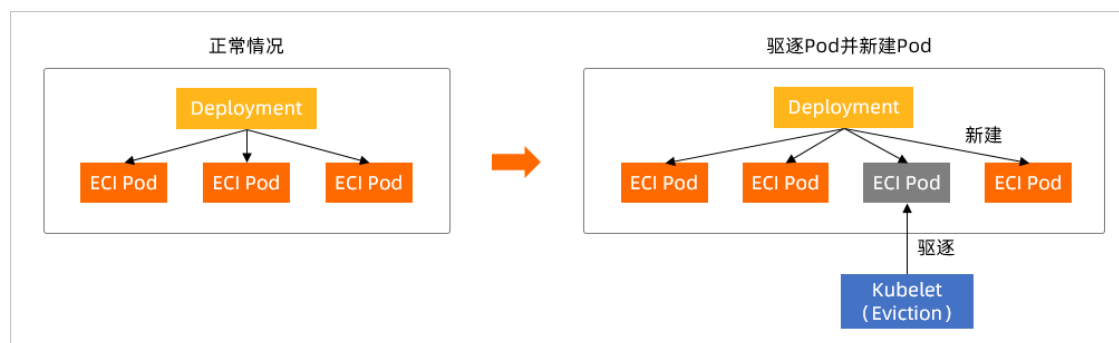
本文适用于Kubernetes场景，请确保您已完成Kubernetes与ECI的对接工作。相关参考如下：

- 对接阿里云容器服务
- 对接自建Kubernetes集群

功能说明

默认情况下，当ECI Pod的临时存储空间不足时，系统不会做任何处理，可能会影响业务运行。基于Kubernetes的驱逐和调度机制，ECI支持为Pod添加 `k8s.aliyun.com/eci-eviction-enable: "true"` 的Annotation，实现在某一ECI Pod的临时存储空间不足（低于100 MiB）时，自动驱逐该Pod（Pod的状态将变更为Failed）。

一般情况下，业务应用部署在Deployment上。如果被驱逐的Pod由Deployment管理，Deployment会自动创建新的Pod，如下图所示。



配置示例

配置时，`k8s.aliyun.com/eci-eviction-enable: "true"` 的Annotation需添加在Pod的metadata下。例如配置Deployment时，Annotation需添加在spec>template>metadata下。YAML示例如下：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 4
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      name: nginx-test
      annotations:
        k8s.aliyun.com/eci-eviction-enable: "true"      #设置自动驱逐临时存储空间不足的Pod
      labels:
        app: nginx
        alibabacloud.com/eci: "true"
    spec:
      containers:
        - name: nginx
          image: registry-vpc.cn-beijing.aliyuncs.com/eci_open/nginx:1.14.2
```

5. 存储

5.1. ACK使用云存储

基于CSI和Flexvolume插件，ACK支持使用云盘、文件存储NAS和对象存储OSS等阿里云存储服务。本文介绍ACK支持的存储插件，以及如何使用云存储。

存储插件介绍

ACK支持CSI和FlexVolume两种存储插件，根据集群的Kubernetes版本，在创建集群时，您可以选择使用CSI或者FlexVolume插件。两者的说明如下表所示。

插件名称	说明
CSI	CSI是当前Kubernetes社区推荐的插件实现方案。阿里云提供的CSI插件主要包括以下几部分： - CSI-Plugin：实现数据卷的挂载、卸载功能。 - CSI-Provisioner：实现数据卷的自动创建能力，目前支持云盘、NAS两种数据卷创建能力。
FlexVolume	Flexvolume是Kubernetes社区较早实现的数据卷扩展机制。阿里云提供的FlexVolume插件主要包括以下几部分： - Flexvolume：支持数据卷的挂载和卸载功能。 - Disk-Controller：实现云盘卷的自动创建功能。 - Nas-Controller：实现NAS卷的自动创建功能。

说明

- 对于新建的集群，推荐您使用CSI插件。
- 不支持FlexVolume插件和CSI插件在同一个集群中使用。
- 不支持FlexVolume插件转换到CSI插件。

使用云存储

阿里云提供针对各种存储资源（块、文件和对象）的低成本、高可靠、高可用的存储服务，您可以根据业务负载的存储需求来选择合适的云存储服务。相关说明如下表所示。

云存储	说明	参考文档 (CSI)	参考文档 (FlexVolume)

云存储	说明	参考文档 (CSI)	参考文档 (FlexVolume)
云盘	云盘是一种数据块级别的块存储产品，采用分布式多副本机制，具有低时延、高性能、持久性、高可靠等性能，可以随时创建、扩容以及释放。更多信息，请参见云盘概述。 云盘为非共享存储，一个云盘只能挂载到一个Pod上。挂载时，支持静态数据卷和动态数据卷。	云盘存储卷概述	云盘存储卷概述
NAS	NAS是一种可共享访问、弹性扩展、高可靠以及高性能的分布式文件系统，支持NFS和SMB协议。更多信息，请参见文件存储NAS概述。 NAS为共享存储，一个NAS可以挂载到多个Pod上。挂载时，支持静态数据卷和动态数据卷。	NAS存储卷概述	NAS存储卷概述
OSS	OSS是一个海量、安全、低成本、高可靠的存储空间，适合存储非结构化数据（如图片、音视频等）。更多信息，请参见对象存储OSS概述。 OSS为共享存储，一个OSS可以挂载到多个Pod上。挂载时，仅支持静态数据卷。	OSS存储卷概述	OSS存储卷概述

5.2. ASK使用云存储

基于CSI和Flexvolume插件，ASK支持使用云盘、文件存储NAS和对象存储OSS等阿里云存储服务。本文介绍ASK支持的存储插件，以及如何使用云存储。

存储插件说明

ASK支持CSI和FlexVolume两种存储插件，根据集群的Kubernetes版本，在创建集群时，您可以选择使用CSI或者FlexVolume插件。两者的说明如下表所示。

插件名称	说明
CSI	CSI是当前Kubernetes社区推荐的插件实现方案。阿里云提供的CSI插件主要包括以下几部分： - CSI-Plugin：实现数据卷的挂载、卸载功能。 - CSI-Provisioner：实现数据卷的自动创建能力，目前支持云盘、NAS两种数据卷创建能力。

插件名称	说明
FlexVolume	Flexvolume是Kubernetes社区较早实现的数据卷扩展机制。阿里云提供的FlexVolume插件主要包括以下几部分： • Flexvolume：支持数据卷的挂载和卸载功能。 • Disk-Controller：实现云盘卷的自动创建功能。 • Nas-Controller：实现NAS卷的自动创建功能。

 说明

- 对于新建的集群，推荐您使用CSI插件。
- 不支持FlexVolume插件和CSI插件在同一个集群中使用。
- 不支持FlexVolume插件转换到CSI插件。

使用云存储

阿里云提供针对各种存储资源（块、文件和对象）的低成本、高可靠、高可用的存储服务，您可以根据业务负载的存储需求来选择合适的云存储服务。相关说明如下表所示。

云存储	说明	参考文档 (CSI)	参考文档 (FlexVolume)
云盘	云盘是一种数据块级别的块存储产品，采用分布式多副本机制，具有低时延、高性能、持久性、高可靠等性能，可以随时创建、扩容以及释放。更多信息，请参见云盘概述。 云盘为非共享存储，一个云盘只能挂载到一个Pod上。挂载时，支持静态数据卷和动态数据卷。	云盘存储卷概述	云盘存储卷概述
NAS	NAS是一种可共享访问、弹性扩展、高可靠以及高性能的分布式文件系统，支持NFS和SMB协议。更多信息，请参见文件存储NAS概述。 NAS为共享存储，一个NAS可以挂载到多个Pod上。挂载时，仅支持静态数据卷。  说明 ASK中的Pod均为ECI Pod，由于ECI不支持Privileged，因此ASK不支持动态挂载NAS。	NAS存储卷概述	NAS存储卷概述

5.3. 非PVC方式

5.3.1. 挂载云盘

云盘是阿里云提供的数据块级别的存储产品，具有低时延、高性能、持久性、高可靠等特点。在阿里云ACK或者ASK集群中，挂载云盘到ECI Pod时，您可以通过PVC的方式挂载，也可以通过非PVC的方式挂载。使用非PVC方式挂载云盘时，支持随Pod创建并挂载云盘，或者直接挂载已有云盘。本文介绍如何使用非PVC方式，挂载云盘到ECI Pod上。

前提条件

集群中已正确部署Virtual kubelet（VK）。

说明

ASK集群已默认集成VK，其它集群需要您自行部署VK。更多信息，请参见[对接概述](#)。

注意事项

- 云盘为非共享存储，一个云盘只能挂载到一个Pod。
- 云盘不支持跨可用区挂载。

随Pod创建并挂载云盘

1. 连接ACK或ASK集群。
2. 准备YAML。

将以下内容保存为test-ack-disk.yaml。

```
apiVersion: v1
kind: Pod
metadata:
  name: test-ack-disk
  labels:
    alibabacloud.com/eci: "true"
spec:
  containers:
    - image: registry-vpc.cn-beijing.aliyuncs.com/eci_open/nginx:1.14.2
      name: test-container
      volumeMounts:
        - mountPath: /cache-test
          name: cache-volume
  volumes:
    - name: cache-volume
      csi:
        driver: diskplugin.csi.alibabacloud.com
        fsType: ext4
        volumeAttributes:
          volumeSize: "20"
```

volumeAttributes中支持设置的云盘属性参数如下表所示：

参数	描述
volumeSize	必填参数。云盘大小，单位为GiB。取值范围为20~32768。
performanceLevel	云盘性能等级，更多信息，请参见 块存储性能 。
deleteWithInstance	是否随实例释放，默认为false。
encrypted	是否加密云盘，默认为false。更多信息，请参见 加密概述 。
kmsKeyId	加密云盘时，云盘使用的KMS密钥ID。

3. 创建ECI Pod，同时创建并挂载云盘。

```
kubectl create -f test-ack-disk.yaml
```

4. 查看结果。

i. 查看Pod信息。

```
kubectl get pods -o wide
```

预期返回：

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED NODE	READINESS GATES					
test-ack-disk	1/1	Running	0	84s	172.16.XX.XXX	virtual-kubelet-cn-beijing-k
	<none>		<none>			

查看Pod中的文件目录，可以看到已生成云盘对应的挂载目录 `/cache-test`。

```
shell@Alicloud:~$ kubectl exec -it test-ack-disk -- bash
root@test-ack-disk:/# ls
bin boot cache-test dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@test-ack-disk:/# ls /cache-test
lost+found
```

ii. 查看云盘。

登录[ECS管理控制台](#)，选择VK所在地域，在云盘页面，您可以看到系统已自动创建云盘，该云盘名称为cache-volume，大小为20 GiB。

挂载已有云盘

挂载已有云盘前，请确保已有云盘未进行分区格式化，并确保该云盘与VK所属的地域可用区一致。

② 说明

VK对应节点的名称表明了其所属可用区。例如virtual-kubelet-cn-beijing-k表示VK部署在华北2（北京）地域可用区K。

1. 获取云盘信息。

- i. 登录[ECS管理控制台](#)。
- ii. 在云盘页面，找到已有云盘，记录云盘ID。

同一云盘不支持挂载到多个Pod，请确保已有云盘的状态为待挂载。

2. 准备YAML。

将以下内容保存为disk-test-ack.yaml。

```
apiVersion: v1
kind: Pod
metadata:
  name: disk-test-ack
  labels:
    alibabacloud.com/eci: "true"
spec:
  containers:
    - image: registry-vpc.cn-beijing.aliyuncs.com/eci_open/nginx:1.14.2
      name: test-container
      volumeMounts:
        - mountPath: /cache-test
          name: cache-volume
  volumes:
    - name: cache-volume
      csi:
        driver: diskplugin.csi.alibabacloud.com
        fsType: ext4
        volumeAttributes:
          volumeId: "d-2ze2qt6x88mv2ih8****"
```

在volumeAttributes中必须通过volumeId指定要挂载的云盘，即云盘ID。

3. 创建ECI Pod并挂载云盘。

```
kubectl create -f disk-test-ack.yaml
```

4. 查看结果。

- i. 查看Pod信息。

```
kubectl get pods -o wide
```

预期返回：

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
	NOMINATED	READYNESS	GATES			
disk-test-ack	1/1	Running	0	32s	172.16.XX.XXX	virtual-kubelet-cn-beijing-k
	<none>		<none>			

查看Pod中的文件目录，可以看到已生成云盘对应的挂载目录 `/cache-test`。

```
shell@Alicloud:~$ kubectl exec -it disk-test-ack -- bash
root@disk-test-ack:/# ls
bin boot cache-test dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@disk-test-ack:/# ls /cache-test
lost+found
```

- ii. 查看云盘。

登录[ECS管理控制台](#)，选择VNode所在地域，在云盘页面，您可以看到云盘的状态已变更为使用中。

5.3.2. 挂载NAS

NAS是阿里云提供的一种可共享访问、弹性扩展、高可靠以及高性能的分布式文件系统。在阿里云ACK或者ASK集群中，挂载NAS到ECI Pod时，您可以通过PVC的方式挂载，也可以通过非PVC的方式挂载。本文介绍如何使用非PVC方式，挂载NAS到ECI Pod上。

前提条件

集群中已正确部署Virtual kubelet（VK）。

② 说明

ASK集群已默认集成VK，其它集群需要您自行部署VK。更多信息，请参见[对接概述](#)。

注意事项

- NAS为共享存储，一个NAS可以挂载到多个Pod上。此时，如果多个Pod同时修改相同数据，需要应用自行执行数据同步。
- 在卸载NAS前，请勿删除NAS挂载点，否则可能会造成操作系统无响应。

配置示例

1. 获取NAS文件系统的挂载点信息。

如果您没有创建NAS文件系统和挂载点，请先在NAS控制台创建。创建时，请选择集群所在地域，集群所属的VPC。具体操作，请参见[创建文件系统](#)和[管理挂载点](#)。

- i. 登录[NAS控制台](#)。
- ii. 选择集群所属地域。
- iii. 在文件系统列表页面，找到目标NAS文件系统，单击文件系统ID。
- iv. 在文件系统详情页，选择[挂载使用](#)，然后在右侧[挂载点地址](#)处，复制挂载点地址。

2. 准备YAML。

将以下内容保存为test-ack-nas.yaml。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-ack-nas
  labels:
    alibabacloud.com/eci: "true"
spec:
  replicas: 2
  selector:
    matchLabels:
      alibabacloud.com/eci: "true"
  template:
    metadata:
      labels:
        alibabacloud.com/eci: "true"
    spec:
      containers:
        - name: nginx
          image: registry-vpc.cn-beijing.aliyuncs.com/eci_open/nginx:1.14.2
          ports:
            - containerPort: 80
          volumeMounts:
            - name: cache-volume
              mountPath: /cache-test
      volumes:
        - name: cache-volume
          csi:
            driver: nasplugin.csi.alibabacloud.com
            fsType: nas
            volumeAttributes:
              server: "0389a***-nh7m.cn-beijing.extreme.nas.aliyuncs.com"
              path: "/"
              vers: "3"
              options: "nolock,tcp,noresvport"
```

volumeAttributes中需配置NAS相关参数，说明如下：

参数	描述
server	NAS的挂载点地址。
path	挂载子目录。极速型NAS需要以 <code>/share</code> 为父目录，例如 <code>/share/path1</code> 。
vers	挂载NAS的NFS协议版本号。推荐使用v3，极速型NAS仅支持v3。
options	挂载选项。建议使用NAS服务推荐选项，例如 <code>nolock,tcp,noresvport</code> 。

3. 创建ECI Pod并挂载NAS。

```
kubectl create -f test-ack-nas.yaml
```

4. 查看结果。

```
kubectl get pods -o wide
```

预期返回：

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED NODE	READINESS GATES					
test-ack-nas-6bb987f4b8-mc7px	1/1	Running	0	64s	172.16.XX.XXX	virt
ual-kubelet-cn-beijing-k	<none>	<none>				
test-ack-nas-6bb987f4b8-vn17c	1/1	Running	0	64s	172.16.XX.XXX	virt
ual-kubelet-cn-beijing-k	<none>	<none>				

查看Pod中的文件目录，可以看到已生成NAS对应的挂载目录 `/cache-test`，且第一个Pod中写入的文件可以在第二个Pod中查看，即两个Pod共享使用NAS存储。

```
shell@Alicloud:~$ kubectl exec -it test-ack-nas-6bb987f4b8-mc7px -- bash
root@test-ack-nas-6bb987f4b8-mc7px:/# ls
bin boot cache-test dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@test-ack-nas-6bb987f4b8-mc7px:/# ls /cache-test
root@test-ack-nas-6bb987f4b8-mc7px:/# echo "hello nas">/cache-test/ack-nas
root@test-ack-nas-6bb987f4b8-mc7px:/# ls /cache-test
ack-nas
root@test-ack-nas-6bb987f4b8-mc7px:/# cat /cache-test/ack-nas
hello nas
root@test-ack-nas-6bb987f4b8-mc7px:/# exit
exit
shell@Alicloud:~$ kubectl exec -it test-ack-nas-6bb987f4b8-vn17c -- bash
root@test-ack-nas-6bb987f4b8-vn17c:/# ls /cache-test
ack-nas
root@test-ack-nas-6bb987f4b8-vn17c:/# cat /cache-test/ack-nas
hello nas
```

5.3.3. 挂载OSS

OSS是阿里云提供的一个海量、安全、低成本、高可靠的存储空间，适合存储非结构化数据（如图片、音视频等）。在阿里云ACK或者ASK集群中，挂载OSS到ECI Pod时，您可以通过PVC的方式挂载，也可以通过非PVC的方式挂载。本文介绍如何使用非PVC方式，挂载OSS到ECI Pod上。

前提条件

集群中已正确部署Virtual kubelet（VK）。

② 说明

ASK集群已默认集成VK，其它集群需要您自行部署VK。更多信息，请参见[对接概述](#)。

注意事项

- OSS为共享存储，一个OSS可以挂载到多个Pod上。
- 挂载目录中建议存放的文件数不要超过1000个。文件数量过多时，OSSFS会占用大量内存，可能会导致Pod发生OOM（Out Of Memory）事件。

配置示例

1. 获取OSS Bucket信息。

如果您没有创建OSS Bucket，请先在OSS管理控制台创建OSS Bucket。具体操作，请参见[创建存储空间](#)。

- i. 登录[OSS管理控制台](#)。
- ii. 在Bucket列表页面，找到目标Bucket，单击Bucket名称。
- iii. 在Bucket详情页面，选择概览，然后在访问域名处，复制Endpoint。

Bucket和集群属于相同地域时，请复制VPC 网络访问（内网）地址；Bucket和集群属于不同地域时，请复制外网地址。

2. 选择授权方式。

- o 方式一：（推荐）使用RAM角色授权。

请创建RAM角色并为该角色授权。创建时，可信实体类型为[阿里云服务](#)，角色类型为[普通服务角色](#)，受信服务为[云服务器](#)；为角色授权时，请选择[AliyunOSSFullAccess](#)权限策略。

具体操作，请参见[创建RAM角色](#)和[为RAM角色授权](#)。

- o 方式二：使用AccessKey直接授权。

请获取AccessKey ID和AccessKey Secret。具体操作，请参见[获取AccessKey](#)。

3. 准备YAML。

将以下内容保存到test-ack-oss.yaml。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-ack-oss
  labels:
    alibabacloud.com/eci: "true"
spec:
  replicas: 2
  selector:
    matchLabels:
      alibabacloud.com/eci: "true"
  template:
    metadata:
      labels:
        alibabacloud.com/eci: "true"
    spec:
      containers:
        - name: nginx
          image: registry-vpc.cn-beijing.aliyuncs.com/eci_open/nginx:1.14.2
          ports:
            - containerPort: 80
          volumeMounts:
            - name: cache-volume
              mountPath: /cache-test
      volumes:
        - name: cache-volume
          csi:
            driver: ossplugin.csi.alibabacloud.com
            fsType: fuse
            volumeAttributes:
              bucket: "oss-test"
              url: "oss-cn-beijing-internal.aliyuncs.com"
              otherOpts: "-o max_stat_cache_size=0 -o allow_other -o connect_timeout=5
-o readwrite_timeout=5"
            ramRole: "<your RAM Role Name>"
```

上述示例采用RAM角色来授权。如果您想要使用AccessKey直接授权，请将

`ramRole: "<your RAM Role Name>"` 替换为以下YAML：

```
akId: "<your AccessKey ID>"
akSecret: "<your AccessKey Secret>"
```

`volumeAttributes`中需配置OSS相关参数，说明如下：

参数	描述
bucket	OSS Bucket的名称。 目前只支持挂载Bucket，不支持挂载Bucket下的子目录或者文件。

参数	描述
url	OSS的接入域名（Endpoint）。 ◦ Bucket和集群属于相同地域时，请使用内网地址。 ◦ Bucket和集群属于不同地域时，请使用外网地址。
otherOpts	挂载OSS时支持输入定制化参数，格式为 <code>-o *** -o ***</code> ，例如： <code>-o max_stat_cache_size=0 -o allow_other</code> 。
ramRole	使用RAM角色授权时，对应的RAM角色。
akId和akSecret	使用AccessKey直接授权时，对应的AccessKey ID和AccessKey Secret。

4. 创建ECI Pod并挂载OSS。

```
kubectl create -f test-ack-oss.yaml
```

5. 查看结果。

```
kubectl get pods -o wide
```

预期返回：

NAME	READY	STATUS	RESTARTS	AGE	IP
NODE	NOMINATED	NODE	READINESS	GATES	
test-ack-oss-655db9d64d-5q7d9	1/1	Running	0	46s	172.16.XX.XXX
virtual-kubelet-cn-beijing-k	<none>		<none>		
test-ack-oss-655db9d64d-m5vct	1/1	Running	0	46s	172.16.XX.XXX
virtual-kubelet-cn-beijing-k	<none>		<none>		

查看Pod中的文件目录，可以看到已生成OSS对应的挂载目录 `/cache-test`，且第一个Pod中写入的文件可以在第二个Pod中查看，即两个Pod共享使用OSS存储。

```
shell@Alicloud:~$ kubectl exec -it test-ack-oss-655db9d64d-5q7d9 -- bash
root@test-ack-oss-655db9d64d-5q7d9:/# ls
bin boot cache-test dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@test-ack-oss-655db9d64d-5q7d9:/# ls /cache-test
root@test-ack-oss-655db9d64d-5q7d9:/# echo "hell oss">/cache-test/ack-oss
root@test-ack-oss-655db9d64d-5q7d9:/# ls /cache-test
ack-oss
root@test-ack-oss-655db9d64d-5q7d9:/# cat /cache-test/ack-oss
hell oss
root@test-ack-oss-655db9d64d-5q7d9:/# exit
exit
shell@Alicloud:~$ kubectl exec -it test-ack-oss-655db9d64d-m5vct -- bash
root@test-ack-oss-655db9d64d-m5vct:/# ls /cache-test
ack-oss
root@test-ack-oss-655db9d64d-m5vct:/# cat /cache-test/ack-oss
hell oss
```

5.4. 挂载stdlog（FlexVolume）

本文介绍如何挂载stdlog，实现将容器的标准输出日志以root权限挂载到ECI实例内。

前提条件

K8s集群中已正确部署Virtual kubelet（VK）。

? 说明

阿里云ASK集群已默认集成VK，其它集群需要您自行部署VK。更多信息，请参见[对接概述](#)。

配置示例

ECI支持通过FlexVolume的方式将容器的标准输出日志以root权限挂载到ECI实例内。配置示例如下：

1. 准备YAML配置文件。

```
vim flexvolume_stdlog.yaml
```

flexvolume_stdlog.yaml的内容如下：

```
apiVersion: v1
kind: Pod
metadata:
  name: test-flexvolume-stdlog
spec:
  containers:
  - image: nginx:latest
    name: test-container
    volumeMounts:
    - mountPath: /cache-test
      name: cache-volume
  volumes:
  - name: cache-volume
    flexVolume:
      driver: alicloud/pod-stdlog
```

2. 部署Pod。

```
kubectl create -f flexvolume_stdlog.yaml
```

3. 查看挂载结果。

使用ls命令查看Pod内的文件目录，可以看到容器的标准日志已经以root权限挂载到 `/cache-test` 目录下，您可以在容器内使用相关日志。

```
shell@Alicloud:~$ kubectl get pod test-flexvolume-stdlog
NAME                READY   STATUS    RESTARTS   AGE
test-flexvolume-stdlog 1/1     Running   0           10m
shell@Alicloud:~$ kubectl exec -it test-flexvolume-stdlog bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
root@test-flexvolume-stdlog:/# ls -l /cache-test/
total 4
drwxr-xr-x 2 root root 4096 Aug 24 09:06 test-container
root@test-flexvolume-stdlog:/# ls -l /cache-test/test-container/
total 4
-rw-r----- 1 root root 1894 Aug 24 09:06 0.log
```