

解决方案实践

第三方云集群迁移到华为云 CCE

文档版本 1.0
发布日期 2023-03-28



版权所有 © 华为技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

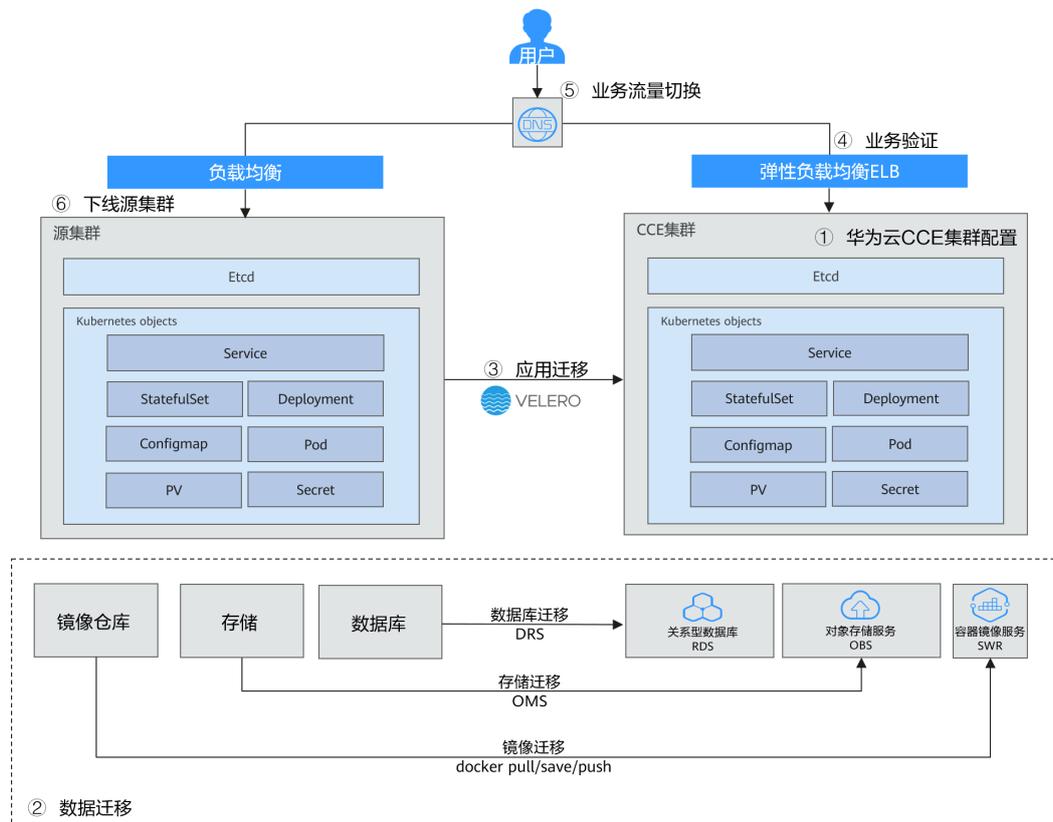
1 方案概述	1
2 资源与成本规划	3
3 实施步骤	4
3.1 数据迁移.....	4
3.2 迁移工具安装.....	4
3.3 集群内资源迁移（Velero）.....	9
3.4 集群内资源迁移（E-Backup）.....	11
3.5 准备对象存储及 Velero.....	16
3.6 备份原 ACK 集群的 Kubernetes 对象.....	17
3.7 在 CCE 集群恢复 Kubernetes 对象.....	17
3.8 资源更新适配.....	18
3.9 调试启动应用.....	21
3.10 其它.....	22
4 修订记录	23

1 方案概述

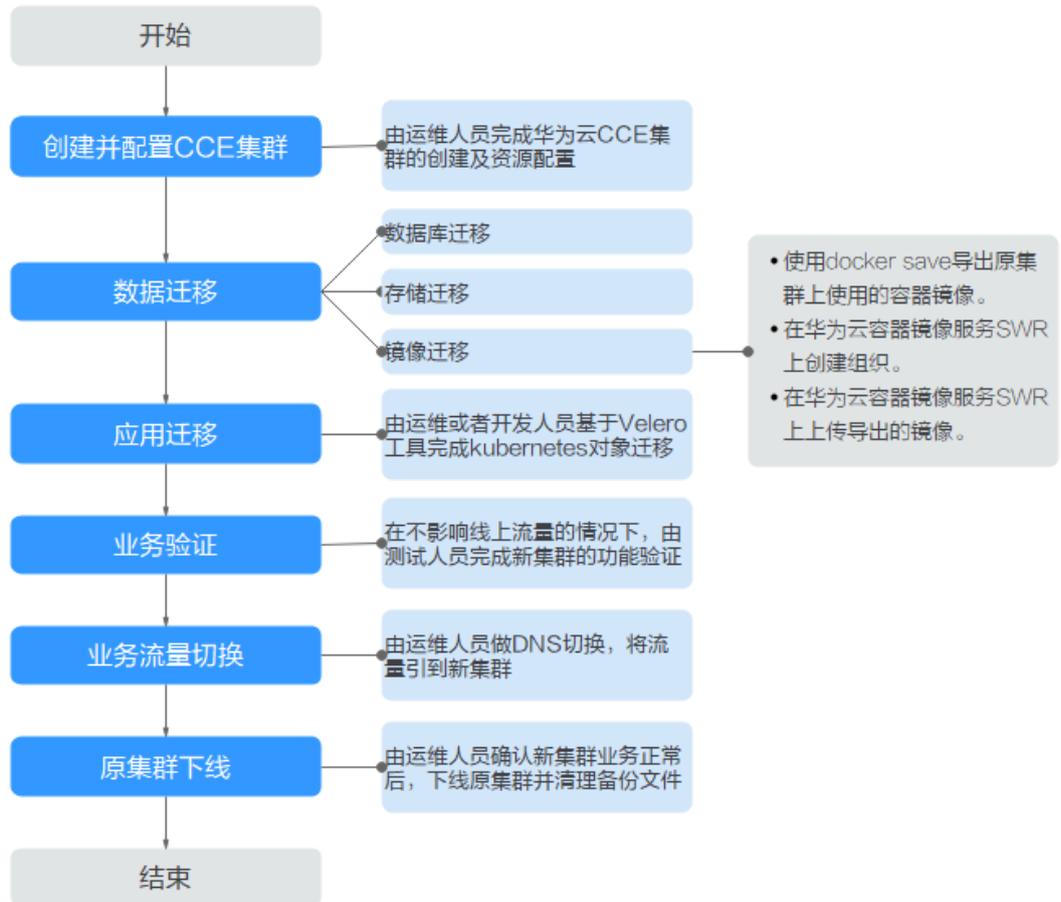
本文将介绍WordPress应用从阿里云ACK迁移到华为云CCE为例来讲解，并假设您已在阿里云上部署了WordPress应用，并创建了自己的博客。

本文简单介绍如何通过如下6个步骤，将应用从阿里云ACK集群平滑迁移到华为云CCE集群，并尽量确保迁移期间对业务无影响。

迁移方案



迁移步骤



2 资源与成本规划

须知

本文提供的成本预估费用仅供参考，资源的实际费用以华为云管理控制台显示为准。

完成本实践所需的资源如下：

表 2-1 资源和成本规划

资源	资源说明	数量	每月费用（元）
云容器引擎 CCE	<ul style="list-style-type: none">● CCE集群版本：v1.21。● 虚拟机节点规格：4核8G或以上规格、EulerOS2.9操作系统。● 建议选择按需计费。	1	3.72元/小时

3 实施步骤

3.1 数据迁移

数据库与存储迁移

数据库迁移

由运维或者开发人员基于华为云[数据复制服务DRS](#)完成数据库迁移，详情请参见[跨云数据库在线迁移](#)。

存储迁移

由运维或者开发人员基于华为云[对象存储迁移服务OMS](#)完成对象存储中的数据迁移，详情请参见[对象存储迁移服务 OMS](#)。

📖 说明

目前对象存储迁移服务OMS支持将亚马逊云、阿里云、微软云、百度云、金山云、青云、七牛云、腾讯云平台的对象存储数据迁移到华为云[对象存储服务OBS](#)。

- 在华为云对象存储服务OBS上创建桶，详情请参见[创建桶](#)。
- 在华为云对象存储迁移服务OMS上创建迁移任务，详情请参见[创建单个迁移任务](#)。

容器镜像迁移

步骤1 导出ACK集群上使用的容器镜像。

按照阿里云容器镜像服务上的操作指南拉取镜像到客户端机器。

步骤2 将导出的容器镜像文件上传到华为云SWR。

使用docker pull命令将镜像上传到华为云，具体操作方法请参见[客户端上传镜像](#)。

----结束

3.2 迁移工具安装

Velero是开源的 Kubernetes 集群备份、迁移工具，集成了Restic工具对PV数据的备份能力，可以通过Velero工具将原集群中的K8s资源对象（如Deployment、Job、

Service、ConfigMap等)和Pod挂载的持久卷数据保存备份上传至对象存储。在发生灾难或需要迁移时,目标集群可使用Velero从对象存储中拉取对应的备份,按需进行集群资源的还原。

根据**迁移方案**所述,在迁移开始前需准备临时的对象存储用于存放资源的备份文件, Velero支持使用OBS或者**MinIO**对象存储。对象存储需要准备足够的存储空间用于存放备份文件,请根据您的集群规模和数据量自行估算存储空间。建议您使用OBS进行备份存储,可直接参考**安装Velero**进行Velero的部署。

CCE支持使用E-Backup插件做备份恢复, E-Backup兼容Velero,且使用OBS作为后端存储。您可以在自建集群使用Velero,在CCE中使用E-Backup。

- 不选择E-Backup:源自建集群和CCE中目标集群都需要安装Velero,请按照本章节内容安装。迁移请按照**3.3 集群内资源迁移 (Velero)**操作。
- 选择E-Backup:源自建集群安装Velero,且使用OBS作为后端存储,按照**安装Velero**操作;CCE中目标集群安装E-Backup,安装E-Backup和迁移按照**3.4 集群内资源迁移 (E-Backup)**操作。

前提条件

- 原始自建集群Kubernetes版本需1.10及以上,且集群可正常使用DNS与互联网服务。
- 若您使用OBS存放备份文件,需已有OBS操作权限用户的AK/SK,请参考**获取访问密钥 (AK/SK)**。
- 若您使用MinIO存放备份文件,则安装MinIO的服务器需要绑定EIP并在安全组中开放MinIO的API端口和Console端口。
- 已创建迁移的目标CCE集群。
- 原集群和目标集群中需要至少各拥有一个空闲节点,节点规格建议为4U8G及以上。

安装 MinIO

MinIO 是一个兼容S3接口协议的高性能对象存储开源工具。若使用MinIO进行存放集群迁移的备份文件,您需要一台临时服务器用于部署MinIO并对外提供服务。若您使用OBS存放备份文件,请忽略此步骤,前往**安装Velero**。

MinIO的安装位置选择有如下几种:

- 集群外临时ECS
将MinIO服务端安装在集群外,能够保障集群发生灾难性故障时,备份文件不会受到影响。
- 集群内的空闲节点
您可以远程登录节点安装MinIO服务端,也可以选择容器化安装MinIO,请参考Velero官方文档<https://velero.io/docs/v1.7/contributions/minio/#set-up-server>。

须知

如使用容器化安装MinIO:

- Velero官方文档提供的YAML文件中存储类型为empty dir，建议将其修改为HostPath或Local类型，否则容器重启后将永久丢失备份文件。
- 您需将MinIO服务对外提供访问，否则将无法在集群外下载备份文件，可选择将Service修改为NodePort类型或使用其他类型的公网访问服务。

无论使用何种方法进行部署，安装MinIO的服务器需要有**足够的存储空间**，且均需要**绑定EIP并在安全组中开放MinIO的服务端口**，否则将无法上传（下载）备份文件。

本示例选择在一台集群外的临时ECS上安装MinIO，步骤如下。

步骤1 下载MinIO对象存储。

```
mkdir /opt/minio
mkdir /opt/miniodata
cd /opt/minio
wget https://dl.minio.io/server/minio/release/linux-amd64/minio
chmod +x minio
```

步骤2 设置MinIO的用户名及密码。

此方法设置的用户名及密码为临时环境变量，在服务重启后需要重新设定，否则会使用默认root凭据minioadmin:minioadmin来创建服务。

```
export MINIO_ROOT_USER=minio
export MINIO_ROOT_PASSWORD=minio123
```

步骤3 创建服务，其中/opt/miniodata/为MinIO 存储数据的本地磁盘路径。

MinIO的API端口默认为9000，console端口默认为随机生成，您可使用--console-address参数指定console访问端口。

```
./minio server /opt/miniodata/ --console-address ":30840" &
```

📖 说明

安装MinIO工具的服务器需开放防火墙、安全组中对应的API和console端口，否则将无法访问对象桶。

步骤4 浏览器访问http://{minio所在节点的eip}:30840，可进入MinIO console界面。

----结束

安装 Velero

首先前往OBS控制台或MinIO console界面，创建存放备份文件的桶并命名为**velero**。此处桶名称可自定义，但安装Velero时必须指定此桶名称，否则将无法访问导致备份失败，参见[步骤4](#)。

须知

- **原集群和目标集群**中均需要安装部署Velero实例，安装步骤一致，分别用于备份和恢复。
- CCE集群的Master节点不对外提供远程登录端口，您可通过kubect操作集群完成Velero安装。
- 如果备份资源量较大，请调整Velero及Restic工具的cpu和内存资源（建议调整至1U1G及以上）。
- 用于存放备份文件的对象存储桶**需要是空桶**。

从Velero官方发布路径<https://github.com/vmware-tanzu/velero/releases>下载最新的稳定版二进制文件，本文以Velero 1.7.0版本为例。原集群和目标集群中的安装过程一致，请参考如下步骤。

步骤1 下载Velero 1.7.0版本的二进制文件。

```
wget https://github.com/vmware-tanzu/velero/releases/download/v1.7.0/velero-v1.7.0-linux-amd64.tar.gz
```

步骤2 安装Velero客户端。

```
tar -xvf velero-v1.7.0-linux-amd64.tar.gz  
cp ./velero-v1.7.0-linux-amd64/velero /usr/local/bin
```

步骤3 创建备份对象存储访问密钥文件credentials-velero。

```
vim credentials-velero
```

文件内容如下，其中的AK/SK请根据实际情况进行替换。使用OBS时，可参考[获取访问密钥（AK/SK）](#)获取AK/SK。如使用MinIO，此处AK/SK则为[步骤2](#)中所创建的用户名及密码。

```
[default]  
aws_access_key_id = {AK}  
aws_secret_access_key = {SK}
```

步骤4 部署Velero服务端。注意其中--bucket参数需要修改为已创建的对象存储桶名称，本例中为**velero**。关于更多自定义安装参数，请参考[自定义安装Velero](#)。

```
velero install \  
--provider aws \  
--plugins velero/velero-plugin-for-aws:v1.2.1 \  
--bucket velero \  
--secret-file ./credentials-velero \  
--use-restic \  
--use-volume-snapshots=false \  
--backup-location-config region=cn-north-4,s3ForcePathStyle="true",s3Url=http://obs.cn-north-4.myhuaweicloud.com
```

表 3-1 Velero 安装参数说明

安装参数	参数说明
--provider	声明使用“aws”提供的插件类型。
--plugins	使用AWS S3兼容的API组件，本文使用的OBS和MinIO对象存储均支持该S3协议。
--bucket	用于存放备份文件的对象存储桶名称，需提前创建。
--secret-file	访问对象存储的密钥文件，即 步骤3 中创建的“credentials-velero”文件。

安装参数	参数说明
--use-restic	使用Restic工具支持PV数据备份，建议开启，否则将无法备份存储卷资源。
--use-volume-snapshots	是否创建 VolumeSnapshotLocation 对象进行PV快照，需要提供快照程序支持。该值设为false。
--backup-location-config	对象存储桶相关配置，包括region、s3ForcePathStyle、s3Url等。
region	对象存储桶所在区域。 <ul style="list-style-type: none"> • OBS：请根据实际区域填写，如“cn-north-4”。 • MinIO：参数值为minio。
s3ForcePathStyle	参数值为“true”，表示使用S3文件路径格式。
s3Url	对象存储桶的API访问地址。 <ul style="list-style-type: none"> • OBS：该参数值需根据对象存储桶地域决定，参数值为“http://obs.{region}.myhuaweicloud.com”。例如区域为北京四（cn-north-4），则参数值为“http://obs.cn-north-4.myhuaweicloud.com”。 • MinIO：该参数值需根据MinIO安装节点的IP及暴露端口确定，参数值为“http://{minio所在节点的eip}:9000”。 <p>说明</p> <ul style="list-style-type: none"> - s3Url中的访问端口需填写MinIO的API端口，而非console端口。MinIO API端口默认为9000。 - 访问集群外安装的MinIO时，需填写其公网IP地址。

步骤5 Velero实例将默认创建一个名为velero的namespace，执行以下命令可查看pod状态。

```
$ kubectl get pod -n velero
NAME                READY STATUS RESTARTS AGE
restic-rn29c        1/1   Running 0      16s
velero-c9ddd56-tkzpk 1/1   Running 0      16s
```

说明

为防止在实际生产环境中备份时出现内存不足的情况，建议您修改Restic和Velero分配的CPU和内存大小。

步骤6 查看Velero工具与对象存储的对接情况，状态需要为available。

```
$ velero backup-location get
NAME PROVIDER BUCKET/PREFIX PHASE LAST VALIDATED ACCESS MODE DEFAULT
default aws velero Available 2021-10-22 15:21:12 +0800 CST ReadWrite true
```

----**结束**

3.3 集群内资源迁移（Velero）

操作场景

本文使用Wordpress应用为例，将自建Kubernetes集群中应用整体迁移到CCE集群。Wordpress应用包含Wordpress和MySQL两个组件，均为容器化实例，分别绑定了两个Local类型的本地存储卷，并通过NodePort服务对外提供访问。

迁移前通过浏览器访问Wordpress站点，创建站点名称为“Migrate to CCE”，并发布一篇文章用于验证迁移后PV数据的完整性。Wordpress中发布文章会被存储在MySQL数据库的“wp_posts”表中，若迁移成功，数据库中的内容也将会被全量搬迁至新集群，可依此进行PV数据迁移校验。

前提条件

- 请在迁移前提前清理原集群中异常的Pod资源。当Pod状态异常但是又挂载了PVC的资源时，在集群迁移后，PVC状态会处于pending状态。
- 请确保CCE侧集群中没有与被迁移集群侧相同的资源，因为Velero工具在检测到相同资源时，默认不进行恢复。
- 为确保集群迁移后容器镜像资源可以正常拉取，请将镜像资源迁移至容器镜像服务（SWR）。
- CCE不支持ReadWriteMany的云硬盘存储，在原集群中存在该类型资源时，需要先修改为ReadWriteOnce。
- Velero集成Restic工具对存储卷进行备份还原，当前不支持HostPath类型的存储卷，详情请参见[Restic限制](#)。若您需备份该类型的存储卷，请将HostPath类型替换为Local类型。当备份任务中存在HostPath类型的存储，该类型存储卷将会被自动跳过并产生Warning信息，并不会导致备份失败。

原集群应用备份

步骤1 （可选）如果需要对Pod中指定的存储卷数据进行备份，需对Pod添加annotation，标记模板如下：

```
kubectl -n <namespace> annotate <pod/pod_name> backup.velero.io/backup-volumes=<volume_name_1>,<volume_name_2>,...
```

- <namespace>：Pod所在的namespace。
- <pod_name>：Pod名称。
- <volume_name>：Pod挂载的持久卷名称。可通过describe语句查询Pod信息，Volume字段下即为该Pod挂载的所有持久卷名称。

对Wordpress和MySQL的Pod添加annotation，pod名称分别为wordpress-758fbf6fc7-s7fsr和mysql-5ffdfbc498-c45lh。由于Pod在默认命名空间default下，-n <NAMESPACE>参数可省略：

```
kubectl annotate pod/wordpress-758fbf6fc7-s7fsr backup.velero.io/backup-volumes=wp-storage  
kubectl annotate pod/mysql-5ffdfbc498-c45lh backup.velero.io/backup-volumes=mysql-storage
```

步骤2 对应用进行备份。备份时可以根据参数指定资源，若不添加任何参数，则默认对整个集群资源进行备份，详细参数请参考[Resource filtering](#)。

- **--default-volumes-to-restic**：表示使用Restic工具对Pod挂载的所有存储卷进行备份，不支持HostPath类型的存储卷。如不指定该参数，将默认对[步骤1](#)中

annotation指定的存储卷进行备份。此参数仅在[安装Velero](#)时指定“--use-restic”后可用。

```
velero backup create <backup-name> --default-volumes-to-restic
```

- **--include-namespaces:** 用于指定namespace下的资源进行备份。

```
velero backup create <backup-name> --include-namespaces <namespace>
```

- **--include-resources:** 用于指定资源进行备份。

```
velero backup create <backup-name> --include-resources deployments
```

- **--selector:** 用于指定与selector相匹配的资源备份。

```
velero backup create <backup-name> --selector <key>=<value>
```

本文指定default命名空间下的资源进行备份，wordpress-backup为备份名称，进行应用恢复时还需指定相同的备份名称。示例如下：

```
velero backup create wordpress-backup --include-namespaces default --default-volumes-to-restic
```

回显如下，表示成功创建备份任务：

```
Backup request "wordpress-backup" submitted successfully. Run `velero backup describe wordpress-backup` or `velero backup logs wordpress-backup` for more details.
```

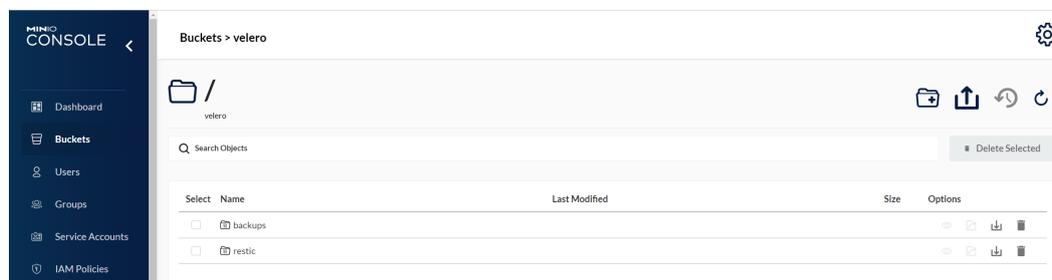
步骤3 查看备份情况。

```
velero backup get
```

回显如下：

NAME	STATUS	ERRORS	WARNINGS	CREATED	EXPIRES	STORAGE
LOCATION	SELECTOR					
wordpress-backup	Completed	0	0	2021-10-14 15:32:07 +0800 CST	29d	default
<none>						

同时可前往对象桶中查看备份的文件，其中backups路径为应用资源备份，restic路径为PV数据备份。



----结束

目标集群应用恢复

由于自建集群与后端的存储基础设施不同，集群迁移后会遇到Pod无法挂载PV的问题。因此在进行迁移时需要对新集群中的StorageClass进行适配，从而在创建工作负载时可以屏蔽两个集群之间底层存储接口的差异，申请相应类型的存储资源，相关操作请参考[StorageClass更新适配](#)。若您使用对象存储迁移服务OMS完成存储迁移，可参考[对接已有对象存储](#)将对象存储桶挂载到应用实例。

- 步骤1** 通过kubectl连接CCE集群，这里选择创建一个原集群中相同名称StorageClass来完成适配。

本例中原集群的StorageClass名为local，存储类型为本地磁盘。本地磁盘存储完全依赖节点可用性，数据容灾性能差，节点不可用时将直接影响已有存储数据。因此CCE集群中的存储资源选用云硬盘存储卷，后端存储介质使用SAS存储。

📖 说明

- 包含PV数据的应用在CCE集群中进行恢复时，定义的StorageClass将会根据PVC动态创建相应的存储资源（如云硬盘）并挂载，请您知悉。
- 此处集群的存储资源可以根据需求进行更改，并非仅限于云硬盘存储卷。若需要挂载其他类型的存储，如文件存储、对象存储，请参考[StorageClass更新适配](#)进行适配。

被迁移侧集群YAML文件：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer
```

迁移侧集群YAML文件示例如下：

```
allowVolumeExpansion: true
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local
  selfLink: /apis/storage.k8s.io/v1/storageclasses/csi-disk
parameters:
  csi.storage.k8s.io/csi-driver-name: disk.csi.everest.io
  csi.storage.k8s.io/fstype: ext4
  everest.io/disk-volume-type: SAS
  everest.io/passthrough: "true"
provisioner: everest-csi-provisioner
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

步骤2 使用Velero工具创建restore，指定名称为wordpress-backup的备份，将Wordpress应用恢复至CCE集群。

```
velero restore create --from-backup wordpress-backup
```

可通过**velero restore get**语句查看应用恢复情况。

步骤3 恢复完成后查看应用实例是否正常运行，可能存在其他的更新适配问题，请参考[3.8 资源更新适配](#)中的步骤排查解决。

----结束

3.4 集群内资源迁移（E-Backup）

操作场景

本节介绍在自建集群中使用开源Velero备份，在CCE集群使用E-Backup插件做恢复的方法。

本文使用Wordpress应用为例，将自建Kubernetes集群中的应用整体迁移到CCE集群。Wordpress应用包含Wordpress和MySQL两个组件，均为容器化实例，分别绑定了两个Local类型的本地存储卷，并通过NodePort服务对外提供访问。

迁移前通过浏览器访问Wordpress站点，创建站点名称为“Migrate to CCE”，并发布一篇文章用于验证迁移后PV数据的完整性。Wordpress中发布的文章会被存储在MySQL数据库的“wp_posts”表中，若迁移成功，数据库中的内容也将会被全量搬迁至新集群，可依此进行PV数据迁移校验。

前提条件

- 请在迁移前提前清理原集群中异常的Pod资源。当Pod状态异常但是又挂载了PVC的资源时，在集群迁移后，PVC状态会处于pending状态。
- 请确保CCE侧集群中没有与被迁移集群侧相同的资源，因为Velero工具在检测到相同资源时，默认不进行恢复。
- 为确保集群迁移后容器镜像资源可以正常拉取，请将镜像资源迁移至容器镜像服务（SWR），具体操作方法请参见[客户端上传镜像](#)。
- CCE不支持ReadWriteMany的云硬盘存储，在原集群中存在该类型资源时，需要先修改为ReadWriteOnce。
- Velero集成Restic工具对存储卷进行备份还原，当前不支持HostPath类型的存储卷，详情请参见[Restic限制](#)。若您需备份该类型的存储卷，请将HostPath类型替换为Local类型。当备份任务中存在HostPath类型的存储，该类型存储卷将会被自动跳过并产生Warning信息，并不会导致备份失败。

原集群应用备份

步骤1 （可选）如果需要对Pod中指定的存储卷数据进行备份，需对Pod添加annotation，标记模板如下：

```
kubectl -n <namespace> annotate <pod/pod_name> backup.velero.io/backup-volumes=<volume_name_1>,<volume_name_2>,...
```

- <namespace>：Pod所在的namespace。
- <pod_name>：Pod名称。
- <volume_name>：Pod挂载的持久卷名称。可通过describe语句查询Pod信息，Volume字段下即为该Pod挂载的所有持久卷名称。

对Wordpress和MySQL的Pod添加annotation，pod名称分别为wordpress-758fbf6fc7-s7fsr和mysql-5ffdfbc498-c45lh。由于Pod在默认命名空间default下，-n <NAMESPACE>参数可省略：

```
kubectl annotate pod/wordpress-758fbf6fc7-s7fsr backup.velero.io/backup-volumes=wp-storage  
kubectl annotate pod/mysql-5ffdfbc498-c45lh backup.velero.io/backup-volumes=mysql-storage
```

步骤2 对应用进行备份。备份时可以根据参数指定资源，若不添加任何参数，则默认对整个集群资源进行备份，详细参数请参考[Resource filtering](#)。

- **--default-volumes-to-restic**：表示使用Restic工具对Pod挂载的所有存储卷进行备份，不支持HostPath类型的存储卷。如不指定该参数，将默认对步骤1中annotation指定的存储卷进行备份。此参数仅在[安装Velero时](#)指定“--use-**restic**”后可用。

```
velero backup create <backup-name> --default-volumes-to-restic
```

- **--include-namespaces**：用于指定namespace下的资源进行备份。

```
velero backup create <backup-name> --include-namespaces <namespace>
```

- **--include-resources**：用于指定资源进行备份。

```
velero backup create <backup-name> --include-resources deployments
```

- **--selector**：用于指定与selector相匹配的资源备份。

```
velero backup create <backup-name> --selector <key>=<value>
```

本文指定default命名空间下的资源进行备份，wordpress-backup为备份名称，进行应用恢复时还需指定相同的备份名称。示例如下：

```
velero backup create wordpress-backup --include-namespaces default --default-volumes-to-restic
```

回显如下，表示成功创建备份任务：



目标集群应用恢复（E-Backup）

如果在CCE中使用E-Backup恢复集群，可以使用如下步骤。

将某个立即备份作为数据源，恢复应用到另一个集群中，全场景适用。

编辑 Restore 模板，如下所示，随后通过 `kubectl create` 命令创建。

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: restore-01
  namespace: velero
spec:
  backupName: wordpress-backup
  includedNamespaces:
  - default
  storageClassMapping:
    local: csi-disk
  imageRepositoryMapping:
    quay.io/coreos: swr.cn-north-4.myhuaweicloud.com/everest
```

- **backupName**: 指定某个立即备份作为数据源，对该备份中的内容进行恢复，必填项。
- **storageClassMapping**: 改变备份资源PV、PVC等使用的storageClassName，要求StorageClass类型相同。本示例中将本地local改为CCE支持的csi-disk。
- **imageRepositoryMapping**: 改变备份资源的images字段，用于仓库的映射关系，不包含镜像名字和标签的改变（防止迁移和升级耦合在一起），比如：`quay.io/coreos/etcd:2.5` 搬迁到SWR后，使用本地镜像仓库下 `swr.cn-north-4.myhuaweicloud.com/everest/etcd:2.5`，配置格式为：`quay.io/coreos: swr.cn-north-4.myhuaweicloud.com/everest`

这里如果配置了**storageClassMapping**和**imageRepositoryMapping**，那**镜像更新适配**和**StorageClass更新适配**就无需再次适配。

其他参数请参见**E-Backup**。

恢复执行后，可通过如下命令查看恢复状态。

```
$ kubectl -n velero get restores restore-01 -o yaml | grep " phase"
phase: Completed
```

状态Completed表示恢复完成，此时可以去CCE控制台查看具体应用恢复情况。

3.5 准备对象存储及 Velero

由运维或者开发人员基于Velero工具完成Kubernetes对象迁移。

准备对象存储 MinIO

MinIO官网地址: <https://docs.min.io/>

准备对象存储，保存其AK/SK。

步骤1 安装MinIO。

MinIO is a high performance,distributed,Kubernetes Native Object Storage.

```
# 二进制安装
mkdir /opt/minio
mkdir /opt/miniodata
cd /opt/minio
wget https://dl.minio.io/server/minio/release/linux-amd64/minio
chmod +x minio
export MINIO_ACCESS_KEY=minio
export MINIO_SECRET_KEY=minio123
./minio server /opt/miniodata/ &
浏览器输入: http://{minio所在节点的eip}:9000 (注意防火墙、安全组需要放开对应端口)

# kubectl容器化安装
# 如果需要将minio发布为集群外可访问的服务, 请修改00-minio-deployment.yaml中的服务类型为NodePort或LoadBalancer
kubectl apply -f ./velero-v1.4.0-linux-amd64/examples/minio/00-minio-deployment.yaml
```

步骤2 创建后面迁移需要使用的桶。

打开minio的web页面
使用MINIO_ACCESS_KEY/MINIO_SECRET_KEY登录minio, 本文中为minio/minio123
单击 '+' 上方的 "Create bucket", 创建桶, 本文中桶名为velero

----结束

准备 Velero

Velero官网地址: <https://velero.io/docs/v1.4/contributions/minio/>

Velero is an open source tool to safely backup and restore, perform disaster recovery, and migrate Kubernetes cluster resources and persistent volumes.

在ACK和CCE的可执行kubectl命令的节点上执行如下操作:

步骤1 下载迁移工具Velero

从<https://github.com/heptio/velero/releases>下载最新的稳定版
本文下载的是velero-v1.4.0-linux-amd64.tar.gz

步骤2 安装Velero客户端

```
mkdir /opt/ack2cce
cd /opt/ack2cce
tar -xvf velero-v1.4.0-linux-amd64.tar.gz -C /opt/ack2cce
cp /opt/ack2cce/velero-v1.4.0-linux-amd64/velero /usr/local/bin
```

步骤3 安装Velero服务端

```
cd /opt/ack2cce
# 准备minio认证文件, ak/sk要正确
vi credentials-velero
```

```
[default]
aws_access_key_id = minio
aws_secret_access_key = minio123

# 安装velero服务端，注意s3Url要修改为正确的minio地址
velero install \
--provider aws \
--plugins velero/velero-plugin-for-aws:v1.0.0 \
--bucket velero \
--secret-file ./credentials-velero \
--use-restic \
--use-volume-snapshots=false \
--backup-location-config region=minio,s3ForcePathStyle="true",s3Url=http://{minio所在节点的eip}:9000

----结束
```

3.6 备份原 ACK 集群的 Kubernetes 对象

步骤1 如果需要备份带PV数据的wordpress应用，请先给对应pod加annotation，不备份PV可跳过此步。

```
# kubectl -n YOUR_POD_NAMESPACE annotate pod/YOUR_POD_NAME backup.velero.io/backup-
volumes=YOUR_VOLUME_NAME_1,YOUR_VOLUME_NAME_2,...

[root@iZbp1cqobeh1iyyf7qgvvzZ ack2cce]# kubectl get pod -n wordpress
NAME READY STATUSRESTARTS AGE
wordpress-67796d86b5-f9bfm 1/1 Running 1 39m
wordpress-mysql-645b796d8d-6k8wh 1/1 Running 0 38m

[root@iZbp1cqobeh1iyyf7qgvvzZ ack2cce]# kubectl -n wordpress annotate pod/wordpress-67796d86b5-
f9bfm backup.velero.io/backup-volumes=wordpress-pvc
pod/wordpress-67796d86b5-f9bfm annotated
[root@iZbp1cqobeh1iyyf7qgvvzZ ack2cce]# kubectl -n wordpress annotate pod/wordpress-
mysql-645b796d8d-6k8wh backup.velero.io/backup-volumes=wordpress-mysql-pvc
pod/wordpress-mysql-645b796d8d-6k8wh annotated
```

步骤2 执行备份任务

```
[root@iZbp1cqobeh1iyyf7qgvvzZ ack2cce]# velero backup create wordpress-ack-backup --include-
namespaces wordpress
Backup request "wordpress-ack-backup" submitted successfully.
Run `velero backup describe wordpress-ack-backup` or `velero backup logs wordpress-ack-backup` for more
details.
```

步骤3 查看备份任务是否成功

```
[root@iZbp1cqobeh1iyyf7qgvvzZ ack2cce]# velero backup get
NAME STATUS CREATED EXPIRES STORAGE LOCATION SELECTOR
wordpress-ack-backup InProgress 2020-07-07 20:31:19 +0800 CST 29d default<none>
[root@iZbp1cqobeh1iyyf7qgvvzZ ack2cce]# velero backup get
NAME STATUS CREATED EXPIRES STORAGE LOCATION SELECTOR
wordpress-ack-backup Completed 2020-07-07 20:31:19 +0800 CST 29d default<none>
```

----结束

3.7 在 CCE 集群恢复 Kubernetes 对象

创建 StorageClass

本示例WordPress应用使用阿里云SSD类型持久化数据卷，相应的在CCE中需要适配成华为云SSD。

本示例中使用的StorageClass是alicloud-disk-ssd，需要创建一个同名的SC，但后端存储介质使用华为云SSD存储。此处请根据自己的应用来适配。

```
[root@cnode-ropr hujun]# cat cce-sc-csidisk-ack.yaml
allowVolumeExpansion: true
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: alicloud-disk-ssd
  selfLink: /apis/storage.k8s.io/v1/storageclasses/csi-disk
parameters:
  csi.storage.k8s.io/csi-driver-name: disk.csi.everest.io
  csi.storage.k8s.io/fstype: ext4
  everest.io/disk-volume-type: SSD
  everest.io/passthrough: "true"
provisioner: everest-csi-provisioner
reclaimPolicy: Delete
volumeBindingMode: Immediate

[root@cnode-ropr hujun]# kubectl create -f cce-sc-csidisk-ack.yaml
```

恢复应用

```
[root@cnode-ropr hujun]# velero restore create --from-backup wordpress-ack-backup
Restore request "wordpress-ack-backup-20200707212519" submitted successfully.
Run `velero restore describe wordpress-ack-backup-20200707212519` or `velero restore logs wordpress-ack-backup-20200707212519` for more details

[root@cnode-ropr hujun]# velero restore get
NAME BACKUP STATUS WARNINGS ERRORS CREATED SELECTOR
wordpress-ack-backup-20200708112940 wordpress-ack-backup Completed 0 02020-07-08 11:29:42 +0800 CST <none>
```

此时查看wordpress应用运行情况，可能会有镜像拉取失败，服务访问不通等问题，需要进行适配处理。

3.8 资源更新适配

镜像更新适配

由于本例使用的Wordpress和MySQL镜像均可从SWR正常拉取，因此不会出现镜像拉取失败（ErrImagePull）问题。如迁移应用为私有镜像，请执行以下步骤完成镜像更新适配。

步骤1 将镜像资源迁移至容器镜像服务（SWR），具体步骤请参考[客户端上传镜像](#)。

步骤2 登录SWR控制台查看获取迁移后的镜像地址。

镜像地址格式如下：

```
'swr:{区域}.myhuaweicloud.com/{所属组织名称}/{镜像名称}:{版本名称}'
```

步骤3 使用如下命令对工作负载进行修改，并将YAML文件中的image字段替换成迁移后的镜像地址。

```
kubectl edit deploy wordpress
```

步骤4 查看应用实例运行情况。

----结束

访问服务更新适配

集群迁移后，原有集群的访问服务可能无法生效，可执行如下步骤更新服务。如原集群中设置了Ingress资源，迁移后需重新对接ELB，您可参考[添加Ingress-对接已有ELB](#)。

步骤1 通过kubectl连接集群。

步骤2 编辑对应Service的YAML文件，修改服务类型及端口。

```
kubectl edit svc wordpress
```

LoadBalancer资源进行更新时，需要重新对接ELB。请参考[通过kubectl命令行创建-使用已有ELB](#)，添加如下Annotation：

annotations:

```
kubernetes.io/elb.class: union #共享型ELB
kubernetes.io/elb.id: 9d06a39d-xxxx-xxxx-xxxx-c204397498a3 #ELB的ID，可前往ELB控制台查询
kubernetes.io/elb.subnet-id: f86ba71c-xxxx-xxxx-xxxx-39c8a7d4bb36 #集群所在子网的ID
kubernetes.io/session-affinity-mode: SOURCE_IP #开启会话保持，基于源IP地址
```

步骤3 浏览器访问查看服务是否可用。

----结束

StorageClass 更新适配

由于集群的存储基础设施不同，迁移后的集群将无法正常使用挂载存储卷，您可执行以下方法的任意一种来完成存储卷的更新适配。

须知

两种StorageClass的适配方法均需在目标集群中于恢复应用前完成，否则可能出现PV数据资源无法恢复的情况，此时在完成StorageClass适配后使用Velero重新恢复应用即可，请参考[目标集群应用恢复](#)。

方式一：创建ConfigMap映射

步骤1 在CCE集群中创建如下所示的ConfigMap，将原集群使用的StorageClass映射到CCE集群默认的StorageClass。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: change-storageclass-plugin-config
  namespace: velero
labels:
  app.kubernetes.io/name: velero
  velero.io/plugin-config: "true"
  velero.io/change-storage-class: RestoreItemAction
data:
  {原集群StorageClass name01}: {目标集群StorageClass name01}
  {原集群StorageClass name02}: {目标集群StorageClass name02}
```

步骤2 执行以下命令，应用上述的 ConfigMap 配置。

```
$ kubectl create -f change-storage-class.yaml
configmap/change-storageclass-plugin-config created
```

----结束

方式二：创建同名StorageClass

步骤1 查询CCE支持的默认StorageClass。

```
kubectl get sc
```

回显如下：

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION	AGE		

csi-disk	everest-csi-provisioner	Delete	Immediate	true	3d23h
csi-disk-topology	everest-csi-provisioner	Delete	WaitForFirstConsumer	true	3d23h
csi-nas	everest-csi-provisioner	Delete	Immediate	true	3d23h
csi-obs	everest-csi-provisioner	Delete	Immediate	false	3d23h
csi-sfsturbo	everest-csi-provisioner	Delete	Immediate	true	3d23h

表 3-2 StorageClass

StorageClass名称	对应的存储资源
csi-disk	云硬盘
csi-disk-topology	延迟绑定的云硬盘
csi-nas	文件存储
csi-obs	对象存储
csi-sfsturbo	极速文件存储

步骤2 通过如下命令将所需的StorageClass详细信息输出为YAML格式。

```
kubectl get sc <storageclass-name> -o=yaml
```

步骤3 复制YAML文件并创建一个新的StorageClass。

编辑StorageClass名称，将其命名为原有集群中使用的名称，用于调用云上基础存储资源。

以csi-obs的YAML文件为例。请删除metadata字段下如斜体部分所示的不必要信息，并修改加粗部分，其余参数不建议修改。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  creationTimestamp: "2021-10-18T06:41:36Z"
  name: <your_storageclass_name> #命名为原有集群中使用的StorageClass名称
  resourceVersion: "747"
  selfLink: /apis/storage.k8s.io/v1/storageclasses/csi-obs
  uid: 4dbbe557-ddd1-4ce8-bb7b-7fa15459aac7
parameters:
  csi.storage.k8s.io/csi-driver-name: obs.csi.everest.io
  csi.storage.k8s.io/fstype: obsfs
  everest.io/obs-volume-type: STANDARD
provisioner: everest-csi-provisioner
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

说明

- 极速文件存储无法通过StorageClass直接创建，您需要前往SFS Turbo控制台创建相同VPC子网下且安全组开放入方向端口(111、445、2049、2051、2052、20048)的极速文件存储。
- CCE不支持ReadWriteMany的云硬盘存储，在原集群中存在该类型资源时，需要先修改为ReadWriteOnce。

步骤4 参考[目标集群应用恢复](#)进行集群应用恢复，检查PVC是否创建成功。

```
kubectl get pvc
```

回显如下，其中VOLUME列为通过StorageClass自动创建的PV名称。

```
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE
pvc Bound pvc-4c8e655a-1dbc-4897-ae6c-446b502f5e77 5Gi RWX local 13s
```

----结束

数据库更新适配

本例中数据库为本地MySQL数据库，迁移后无需重新配置。若您通过**数据复制服务DRS**将本地数据库迁移至云数据库RDS，则在迁移后需重新配置数据库的访问，请您根据实际情况进行配置。

说明

- 若云数据库RDS实例与CCE集群处于同一VPC下，则可通过内网地址访问，否则只能通过绑定EIP的方式进行公网访问。建议使用内网访问方式，安全性高，并且可实现RDS的较好性能。
- 请确认RDS所在安全组入方向规则已对集群放通，否则将连接失败。

步骤1 登录RDS控制台，在该实例的“基本信息”页面获取其“内网地址”及端口。

步骤2 使用如下命令对Wordpress工作负载进行修改。

```
kubectll edit deploy wordpress
```

设置env字段下的环境变量：

- WORDPRESS_DB_HOST：数据库的访问地址和端口，即上一步中获取的内网地址及端口。
- WORDPRESS_DB_USER：访问数据库的用户名。
- WORDPRESS_DB_PASSWORD：访问数据库的密码。
- WORDPRESS_DB_NAME：需要连接的数据库名。

步骤3 检查RDS数据库是否正常连接。

----结束

3.9 调试启动应用

调试并访问应用，检查数据。

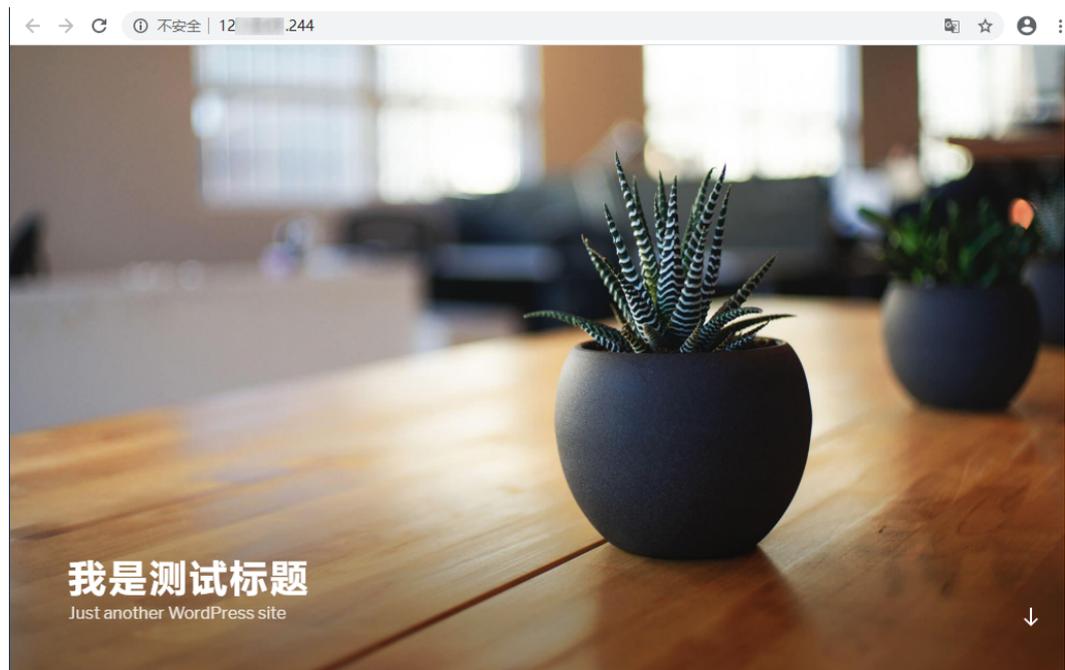
步骤1 登录**CCE控制台**，在左侧导航栏中选择“资源管理 > 网络管理”，单击wordpress服务后方的弹性公网访问地址。

图 3-1 获取访问地址



步骤2 访问正常，迁移成功。

图 3-2 wordpress 欢迎页



----结束

3.10 其它

业务验证

在不影响线上流量的情况下，由测试人员完成新集群的功能验证。

- 配置测试域名
- 测试业务功能
- 测试监控日志告警等运维功能

业务流量切换

由运维人员做DNS切换，将流量引到新集群。

- DNS流量切换：调整DNS配置实现流量切换
- 客户端流量切换：升级客户端代码或更新配置实现流量切换

原 ACK 集群下线

由运维人员确认新集群业务正常后，下线原集群并清理备份文件。

- 确认新集群业务正常
- 下线原集群
- 清理备份文件

4 修订记录

表 4-1 修订记录

发布日期	修订记录
2022-5-30	第一次正式发布
2023-03-28	修订实施步骤