

# 在 KubeSphere 中部署 ClickHouse 集群

[ClickHouse](#) 是一款用于联机分析 (OLAP) 的列式数据库管理系统 (DBMS)。[RadonDB ClickHouse](#) 是一款深度定制的 ClickHouse 集群应用，完美保持了 ClickHouse 集群功能特性，并具备集群自动管理、集群数据重分布、高性能低成本等优势功能特性。

本教程演示了如何在 KubeSphere 上部署 ClickHouse Operator 和 ClickHouse 集群。

## 准备工作

- 请确保[已启用 OpenPitrix 系统](#)。
- 您需要创建一个企业空间、一个项目和一个用户帐户 (`project-regular`) 供本教程操作使用。该帐户需要是平台普通用户，并邀请至项目中赋予 `operator` 角色作为项目操作员。本教程中，请以 `project-regular` 身份登录控制台，在企业空间 `demo-workspace` 中的 `demo-project` 项目中进行操作。有关更多信息，请参见[创建企业空间、项目、帐户和角色](#)。
- 请确保 KubeSphere 集群[已开启外网访问](#)。

## 动手实验

### 步骤 1：部署 ClickHouse Operator

- 以 `admin` 身份登录 KubeSphere 的 Web 控制台，并使用[工具箱](#)中的 **Kubectl** 执行以下命令来安装 ClickHouse Operator。建议至少准备 2 个可用集群节点。

```
kubectl apply -f https://raw.githubusercontent.com/radondb/radondb-clickhouse-kubernetes/main/clickhouse-operator-install.yml
```

ClickHouse Operator 将会被安装在 `kube-system` 命名空间下，因此一个 Kubernetes 集群只需要安装一次 ClickHouse Operator。

#### 预期结果

```
customresourcedefinition.apiextensions.k8s.io/clickhouseinstallations.clickhouse.altinity.com configured
customresourcedefinition.apiextensions.k8s.io/clickhouseinstallationtemplates.clickhouse.altinity.com created
customresourcedefinition.apiextensions.k8s.io/clickhouseoperatorconfigurations.clickhouse.altinity.com created
serviceaccount/clickhouse-operator created
clusterrolebinding.rbac.authorization.k8s.io/clickhouse-operator-kube-system created
configmap/etc-clickhouse-operator-files created
configmap/etc-clickhouse-operator-confd-files created
configmap/etc-clickhouse-operator-configd-files created
configmap/etc-clickhouse-operator-templatesd-files created
configmap/etc-clickhouse-operator-usersd-files created
deployment.apps/clickhouse-operator created
service/clickhouse-operator-metrics created
```

2. 执行如下命令可查看 ClickHouse Operator 资源状态。

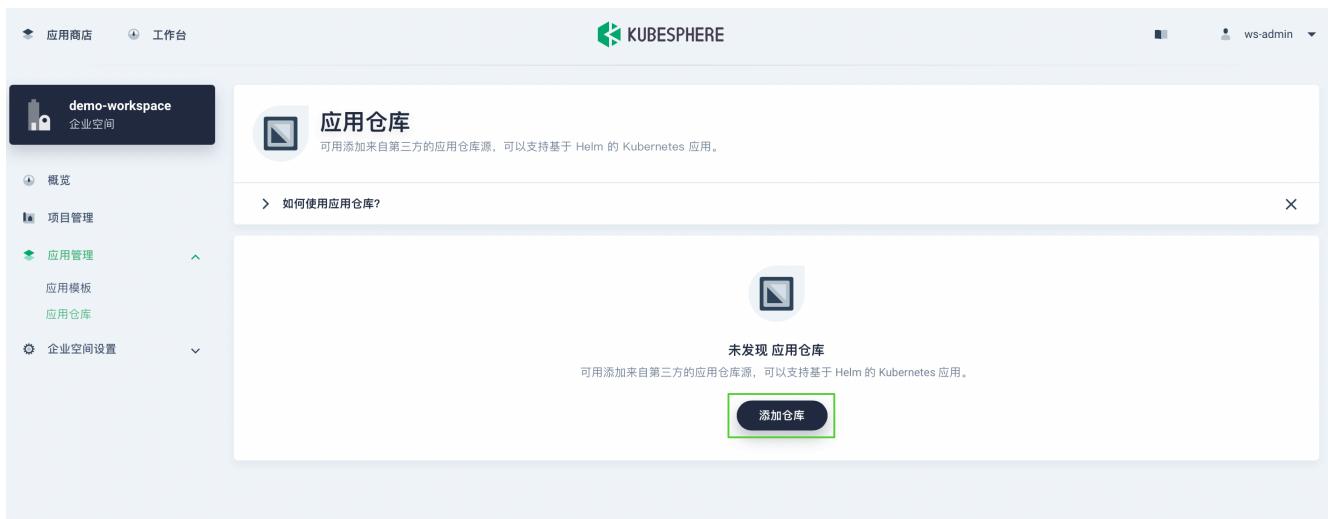
```
kubectl get all --selector=app=clickhouse-operator -n kube-system
```

#### 预期结果

NAME	READY	STATUS	RESTARTS	AGE
pod/clickhouse-operator-644fcb8759-9tfcx	2/2	Running	0	4m32s
<hr/>				
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	
PORT(S)	AGE			
service/clickhouse-operator-metrics	ClusterIP	10.96.72.49	<none>	
8888/TCP	4m32s			
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/clickhouse-operator	1/1	1	1	4m32s
NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/clickhouse-operator-644fcb8759	1	1	1	4m32s

## 步骤 2：添加应用仓库

1. 以 `ws-admin` 身份登录 KubeSphere 的 Web 控制台。在企业空间中，进入应用管理下的应用仓库页面，点击添加仓库。



2. 在出现的对话框中，输入 `clickhouse` 作为应用仓库名称，输入 `https://radondb.github.io/radondb-clickhouse-kubernetes/` 作为仓库的 URL。点击验证以验证 URL。在 URL 旁边呈现一个绿色的对号，验证通过后，点击确定继续。

应用仓库名称 \*

URL

√

所输入的 URL 需要先验证才可进行添加或编辑操作

描述信息

描述信息不超过 256 个字符

取消 确定

3. 将仓库成功导入到 KubeSphere 之后，在列表中可查看 ClickHouse 仓库。

The screenshot shows the KubeSphere web interface. In the top left, there are links for '应用商店' (Application Store) and '工作台' (Workstation). The top center features the KubeSphere logo. On the right, it says 'ws-admin'. The main area has a dark sidebar on the left with sections like '概览' (Overview), '项目管理' (Project Management), '应用管理' (Application Management), '应用模板' (Application Templates), '应用仓库' (Application Repository), and '企业空间设置' (Enterprise Space Settings). The main content area is titled '应用仓库' (Application Repository) and contains a note: '可用添加来自第三方的应用仓库源, 可以支持基于 Helm 的 Kubernetes 应用。' Below this is a search bar with placeholder text '请输入名称进行查找' (Enter name to search). A table lists one item: 'clickhouse' with status '成功' (Success) and URL 'https://radondb.github.io/radondb-clickhouse-kubernetes/'. At the bottom, it says '共 1 个条目' (1 item found).

## 步骤 3：部署 ClickHouse 集群

1. 以 `project-regular` 身份登录 KubeSphere 的 Web 控制台。在 `demo-project` 项目中，进入应用负载下的应用页面，点击部署新应用。

This screenshot shows the '应用' (Application) page within the 'demo-project' project. The sidebar includes '概览', '应用负载', '应用' (selected), '服务', '工作负载', '任务', '应用路由', '容器组', '存储管理', '配置中心', '监控告警', and '项目设置'. The main content area has tabs for '应用模板' (Application Templates) and '自制应用' (Custom Application). It displays a note: '应用为用户提供完整的业务功能, 由一个或多个特定功能的组件组成。' and a '了解更多' (Learn More) link. A large green button labeled '部署新应用' (Deploy New Application) is highlighted with a green border. Below it is a table listing one application: 'mysql-inner' with status '运行中' (Running), application 'mysql', version '1.6.8 [5.7.31]', and last updated '2021-04-30 14:05:15'.

2. 在对话框中，选择来自应用模板。

The screenshot shows the '部署新应用' (Deploy New Application) dialog box. It features a decorative background with a chair and books. The title is '部署新应用'. Below it is a sub-section titled '来自应用商店' (From Application Store) with the text '来自KubeSphere官方应用商店, 提供高质量应用和简易的部署方式'. Another section, '来自应用模板' (From Application Template), is highlighted with a green border and the text '来自于企业空间的自制应用模板以及应用仓库中添加的第三方Helm 应用模板'.

3. 从下拉菜单中选择 `clickhouse` 应用仓库，然后点击 **clickhouse-cluster**。

The screenshot shows the KubeSphere application catalog interface. At the top, there is a search bar with the placeholder "您可以根据相关条件进行过滤" (You can filter by related conditions) and a dropdown menu set to "clickhouse". Below the search bar, there are two application card entries:

- clickhouse-operator**: A Helm chart for Kubernetes. It has a "C" icon, developer information "开发者: ws-admin" and "最新: -", and a brief description "clickhouse-operator Helm chart for Kubernetes".
- clickhouse-cluster**: A Helm Chart for deploying a sharded and replicated ClickHouse. It has a yellow bar icon, developer information "开发者: ws-admin" and "最新: -", and a brief description "Helm Chart for deploying a sharded and replicated ClickHouse".

4. 在配置文件选项卡，可以直接通过控制台查看配置信息，也可以通过下载默认 `values.yaml` 文件查看。在版本列框下，选择一个版本号，点击部署以继续。

The screenshot shows the configuration page for the 'clickhouse-cluster' Helm Chart. On the left, there's a code editor displaying the 'values.yaml' file with various configuration parameters for the ClickHouse cluster. On the right, there's a summary section with the following details:

- 版本:** 0.1.0 [19.17] (最新版本)
- 基本信息:**
  - 类别: -
  - 首页: -
  - 上架时间: 2021年04月23日
  - 应用编号: app-3kk3nqj9lv1r3v

A green box highlights the '部署' (Deploy) button at the top right.

```
# Configuration for the ClickHouse cluster to be started
clickhouse:
  # default cluster name
  clusterName: all-nodes
  # shards count can not scale in this value.
  shardcount: 1
  # replicas count can not modify this value when the cluster has already created.
  replicascount: 2

  # ClickHouse server image configuration
  image: alpinelinux/clickhouse-server:v19.17.6.36-stable
  imagePullPolicy: Always

  resources:
    memory: "1Gi"
    cpu: "0.5"

  # User Configuration
  user:
    - username: clickhouse
      password: clickhouseoperator
    networks:
      - "127.0.0.1"
      - "::/0"
```

5. 在基本信息页面，确认应用名称、应用版本以及部署位置。点击下一步以继续。

The screenshot shows the 'Basic Information' configuration page for deploying the ClickHouse cluster. It includes the following fields:

- 基本信息** tab selected.
- 应用名称 \***: clickhouse-app
- 应用版本 \***: 0.1.0 [19.17] (最新版本)
- 描述信息**: A large text input field with placeholder text '描述信息不超过 256 个字符'.
- 部署位置** section:
  - demo-workspace (企业空间)
  - default (集群)
  - demo-project (项目)
- 下一步** button at the bottom right.

6. 在应用配置页面，可以编辑 `values.yaml` 文件，也可以直接点击部署使用默认配置。

The screenshot shows the 'Application Configuration' tab of the ClickHouse cluster Helm Chart configuration. It displays the `values.yaml` file content:

```
1 # Configuration for the ClickHouse cluster to be started
2 clickhouse:
3   # default cluster name
4   clusterName: all-nodes
5   # shards count can not scale in this value.
6   shardsCount: 1
7   # replicas count can not modify this value when the cluster has already created.
8   replicasCount: 2
9
10  # ClickHouse server image configuration
11  image: = /clickhouse-server:v19.17.6.36-stable
12  imagePullPolicy: Always
13
14  resources:
15    memory: "1Gi"
16    cpu: "0.5"
17
18  # User Configuration
19  user:
20    - username: clickhouse
21      password: clickh0use0perator
22      networks:
```

At the bottom right, there are two buttons: 'Previous Step' and 'Deploy'.

7. 等待 ClickHouse 集群正常运行。可在工作负载下的应用页面，查看部署的应用。

The screenshot shows the 'Applications' section of the KubeSphere web console under the 'demo-project'. It lists two applications: 'clickhouse-app' and 'mysql-inner'. The 'clickhouse-app' entry is highlighted with a green border.

名称	状态	应用	版本	上次更新时间
C clickhouse-app	运行中	clickhouse-cluster	0.1.0 [19.17]	2021-05-07 15:46:01
M mysql-inner	运行中	mysql	1.6.8 [5.7.31]	2021-04-30 14:05:15

## 步骤 4：查看 ClickHouse 集群状态

1. 以 `project-regular` 身份登录 KubeSphere 的 Web 控制台。
2. 进入应用负载下的工作负载页面，点击有状态副本集，查看集群状态。

The screenshot shows the 'Workload' section of a cloud management platform. On the left, a sidebar for 'demo-project' lists various application components like '应用', '服务', '工作负载', etc. The main area is titled '工作负载' (Workload) and displays three stateful sets:

- chi-clickhouse-app-all-nodes-0-1: Status '运行中 (1/1)', created at 2021-05-07 15:51:13
- chi-clickhouse-app-all-nodes-0-0: Status '运行中 (1/1)', created at 2021-05-07 15:46:02
- zk-clickhouse-app: Status '运行中 (3/3)', created at 2021-05-07 15:46:01

At the bottom, it says '共 3 个条目' (3 items) and '1 / 1'.

进入一个有状态副本集群详情页面，点击监控标签页，可查看一定时间范围内的集群指标。



3. 进入应用负载下的容器组页面，可查看所有状态的容器。

The screenshot shows the 'Pods' (容器组) section of the application load balancer. The sidebar lists components like '应用', '服务', '工作负载', etc. The main area is titled '容器组' (Pods) and displays a list of running containers:

名称	节点	容器组 IP	应用	更新时间
chi-clickhouse-app-all-nodes-0-1-0	worker-p002(192.168.0.5)	192.168.0.208		2021-05-07 15:51:27
chi-clickhouse-app-all-nodes-0-0-0	worker-p001(192.168.0.4)	192.168.0.200		2021-05-07 15:46:15
zk-clickhouse-app-2	worker-p002(192.168.0.5)	192.168.0.207		2021-05-07 15:46:21
zk-clickhouse-app-1	worker-p001(192.168.0.4)	192.168.0.204		2021-05-07 15:46:21
zk-clickhouse-app-0	worker-p002(192.168.0.5)	192.168.0.176		2021-05-07 15:46:21
mysql-inner-79d68d6b76-cjv7w	worker-p002(192.168.0.5)	192.168.0.141		2021-04-30 14:05:29

At the bottom, it says '共 6 个条目' (6 items) and '1 / 1'.

#### 4. 进入存储管理下的存储卷页面，可查看存储卷，所有组件均使用了持久化存储。

名称	状态	访问模式	挂载	创建时间
data-chi-clickhouse-app-all-nodes-0-1-0 csi-standard	准备就绪	ReadWriteOnce	已挂载	2021-05-07 15:51
data-chi-clickhouse-app-all-nodes-0-0-0 csi-standard	准备就绪	ReadWriteOnce	已挂载	2021-05-07 15:46
datadir-volume-clickhouse-app-demo-project-zk-clickhouse-app-2 csi-standard	准备就绪	ReadWriteOnce	已挂载	2021-05-07 15:46
datadir-volume-clickhouse-app-demo-project-zk-clickhouse-app-1 csi-standard	准备就绪	ReadWriteOnce	已挂载	2021-05-07 15:46
datadir-volume-clickhouse-app-demo-project-zk-clickhouse-app-0 csi-standard	准备就绪	ReadWriteOnce	已挂载	2021-05-07 15:46
mysql-inner csi-standard	准备就绪	ReadWriteOnce	已挂载	2021-04-30 14:05

查看某个存储卷用量信息，以其中一个数据节点为例，可以看到当前存储的存储容量和剩余容量等监控数据。

资源状态

元数据 事件 快照信息

存储卷

Kubernetes 采集的是存储卷的设备用量数据，未挂载的存储卷暂时采集不到，并且对于如 OpenEBS/Local PV、NFS 等路径型存储卷通常与实际用量有一定出入。详见 [存储卷监控数据分析](#)。

0.36%

9.73 Gi 剩余存储 9.78 Gi 存储卷容量

Inode 使用率 (%)

15:08:07 15:15:07 15:22:07 15:29:07 15:36:07 15:43:07 15:50:07 15:57:07 16:08:07

已挂载容器组

请输入关键字过滤

chi-clickhouse-app-all-nodes-0-1-0 创建于 18 分钟前 worker-p002(192.16... 节点 208 容器组 IP CPU 1 m 内存 43.42 Mi

#### 5. 在项目概览页面，可查看当前项目资源使用情况。

## 步骤 5：访问 ClickHouse 集群

1. 以 `admin` 身份登录 KubeSphere 的 Web 控制台，将鼠标悬停在右下角的锤子图标上，选择 **Kubectl**。
2. 打开终端窗口，执行如下命令，并输入 ClickHouse 集群用户名和密码。

```
kubectl edit chi <app name> -n <project name>
```

以下命令示例中 **app name** 为 `clickhouse-app`，**project name** 为 `demo-project`。

```

configuration:
clusters:
- address: {}
layout:
replicasCount: 2
shardsCount: 1
name: all-nodes
templates:
podTemplate: pod-template-with-volume
zookeeper: {}
users:
clickhouse/networks/ip:
- 127.0.0.1
- ::/0
clickhouse/password: clickhouseoperator
clickhouse/profile: default
clickhouse/quotas: default
readonly/profile: readonly
zookeeper:
nodes:
- host: zk-clickhouse-app-0.zk-server-clickhouse-app.demo-project
port: 2181
- host: zk-clickhouse-app-1.zk-server-clickhouse-app.demo-project
port: 2181
- host: zk-clickhouse-app-2.zk-server-clickhouse-app.demo-project
port: 2181
defaults:
distributedDDL: {}
templates:
dataVolumeClaimTemplate: data
logVolumeClaimTemplate: data
podTemplate: pod-template-with-volume
- /tmp/kubectl-edit-pzf13.yaml 56/436 12k

```

3. 执行如下命令，访问 ClickHouse 集群，并可通过 `show databases` 命令查看数据库。

```
kubectl exec -it <pod name> -n <project name> -- clickhouse-client --user=<user name> --password=<user password>
```

- 以下命令示例中 **pod name** 为 `chi-clickhouse-app-all-nodes-0-1-0`，**project name** 为 `demo-project`，**user name** 为 `clickhouse`，**password** 为 `clickh0use0perator`。
- 可在应用负载的容器组下获取 **pod name**。

The screenshot shows a terminal window titled "Kubectl". The command entered is:

```
/ # kubectl exec -it chi-clickhouse-app-all-nodes-0-1-0 -n demo-project -- clickhouse-client --user=clickhouse --password=clickh0use0perator
```

Defaulting container name to clickhouse.  
Use 'kubectl describe pod/chi-clickhouse-app-all-nodes-0-1-0 -n demo-project' to see all of the containers in this pod.  
ClickHouse client version 19.17.6.36.  
Connecting to localhost:9000 as user clickhouse.  
Connected to ClickHouse server version 19.17.6 revision 54428.

```
chi-clickhouse-app-all-nodes-0-1-0.chi-clickhouse-app-all-nodes-0-1.demo-project.svc.cluster.local :> show databases
```

SHOW DATABASES

```
+-----+  
| name |  
+-----+  
| default |  
| system |  
+-----+
```

2 rows in set. Elapsed: 0.050 sec.

```
chi-clickhouse-app-all-nodes-0-1-0.chi-clickhouse-app-all-nodes-0-1.demo-project.svc.cluster.local :> _
```

On the right side of the terminal window, there is a sidebar with "kubectl" sections:

- kubectl 常用命令**  
查阅更多命令请参照 [官方文档](#)
- kubectl 输出格式**
  - 显示 Pod 的更多信息  
`kubectl get pod <pod-name> -o wide`
  - 以 yaml 格式显示 Pod 的详细信息  
`kubectl get pod <pod-name> -o yaml`
- kubectl 操作**
  - 1. 创建资源对象**
    - 根据 yaml 配置文件一次性创建 service 和 rc  
`kubectl create -f my-service.yaml -f my-rc.yaml`
    - 对目录下所有 .yaml、.yml、.json 文件进行创建操作  
`kubectl create -f <directory>`
  - 2. 查看资源对象**
    - 查看所有 Pod 列表  
`kubectl get pods`
    - 查看 rc 和 service 列表  
`kubectl get rc,service`
  - 3. 查看资源详情**
    - 显示 pod 的详细信息  
`kubectl get pod --export-xml`
    - \* 不再显示帮助信息