

DaoCloud Enterprise

DaoCloud Enterprise Product Documentation

Table of contents

1. 安装依赖项	3
1.1 在线安装依赖项	3
1.2 离线安装依赖项	4
2. 脚本与离线包都位于同一目录层级	4
3. 安装简介	5
3.1 社区版安装流程	5
3.2 商业版安装流程	6
3.3 联系我们	6
4. 安装器 Release Notes	7
4.1 2022-2-28	7
4.2 2022-12-30	7
4.3 2022-11-30	8
5. 卸载	10
5.1 卸载社区版	10
6. Commercial	11
6.1 集群配置文件 clusterConfig.yaml	11
6.2 部署架构	15
6.3 部署规划	17
6.4 文件下载	18
6.5 假定 VERSION 为 v0.4.0	18
6.6 假定 VERSION 为 v0.4.0	18
6.7 准备工作	19
6.8 集群部署模式使用场景	21
6.9 离线安装 DCE 5.0 商业版	24
7. Community	27
7.1 集群资源规划	27
7.2 K8s	28
7.3 Kind	35

1. 安装依赖项

安装 DCE 5.0 之前，需要先安装一些依赖项。

- 对于社区版，请在 K8s Master 节点上安装依赖项。
- 对于商业版，请在[火种节点](#)上安装依赖项。

!!! note

安装的依赖项大致包括：

```
- podman
- helm
- skopeo
- kind
- kubect1
- yq
- minio client
```

安装过程中如果您的环境存在一些工具且版本低于我们定义的版本，会将对应工具进行强制更新替换。

1.1 在线安装依赖项

1. 下载脚本。

```
shell
curl -LO https://proxy-giniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/install_prerequisite.sh
```

为 `install_prerequisite.sh` 添加可执行权限：

```
bash
chmod +x install_prerequisite.sh
```

2. 开始在线安装前置依赖。

- 对于社区版

```
bash
bash install_prerequisite.sh online community
```

- 对于商业版

```
bash
bash install_prerequisite.sh online full
```

1.2 离线安装依赖项

离线安装意味着目标主机的网络处于离线状态，无法下载所需依赖项，所以需先在一个在线环境中制作好离线包。

1. 制作离线包。

- 社区版制作命令

```
bash
bash install_prerequisite.sh export community
```

- 商业版制作命令

```
bash
bash install_prerequisite.sh export full
```

!!! note

当上述命令执行完成后，会在当前目录生成名为 pre_pkgs.tar.gz 的压缩包，该压缩包中会包含安装所需的所有文件。

2. 上传目录中的所有文件到离线环境。

```
```bash
```

## 2. 脚本与离线包都位于同一目录层级

```
$ tree . . |—— install_prerequisite.sh |—— pre_pkgs.tar.gz ```
```

### 3. 执行离线安装。

- 对于社区版

```
bash
bash install_prerequisite.sh offline community
```

- 对于商业版

```
bash
bash install_prerequisite.sh offline full
```

接下来就可以安装 DCE 5.0 了。

## 3. 安装简介

DCE 5.0 有两个版本：社区版和商业版。

版本	包含的模块	描述
社区版	全局管理 容器管理 可观测性	永久免费授权，3 个模块会保持持续更新，可随时 <a href="#">下载子模块的离线包</a>
商业版	全局管理 容器管理 可观测性 应用工作台 多云编排 微服务引擎 服务网格 精选中间件 镜像仓库	联系我们授权：电邮 <a href="mailto:info@daocloud.io">info@daocloud.io</a> 或致电 400 002 6898，各个模块可按需自由组合，可随时 <a href="#">下载子模块的离线包</a>

### 3.1 社区版安装流程

DCE 5.0 社区版的安装流程如下图：

```
flowchart TB
 subgraph second[使用说明]
 direction TB
 U[] --> free[申请免费体验]
 free --> kpanda[kpanda[容器管理]]
 kpanda --> ghippo[ghippo[全局管理]]
 ghippo --> insight[insight[可观测性]]
 insight --> ask[提问!!!]
 end

 subgraph first[安装社区版]
 direction TB
 S[] --> plan[资源规划]
 plan --> k8s[准备 K8s 集群]
 k8s --> tools[安装依赖项]
 tools --> kind[通过 kind 集群
在线安装]
 tools --> s1[通过标准 k8s 集群
在线安装]
 tools --> s2[通过标准 k8s 集群
离线安装]
 end

 start([fa-user DCE 5.0 社区版
安装和使用流程]) --> first
 start --> second

 classDef grey fill:#ddddd,stroke:#fffff,stroke-width:px,color:#000000, font-size:15px;
 classDef white fill:#ffffff,stroke:#000,stroke-width:px,color:#000,font-weight:bold;
 classDef spacewhite fill:#ffffff,stroke:#fff,stroke-width:0px,color:#000;
 classDef plain fill:#ddd,stroke:#fff,stroke-width:1px,color:#000;
 classDef k8s fill:#326ce5,stroke:#fff,stroke-width:1px,color:#fff;
 classDef cluster fill:#fff,stroke:#bbb,stroke-width:1px,color:#326ce5;

 class plan,k8s,tools,kind,s1,s2,kpanda,ghippo,insight,free,ask cluster;
 class start plain
 class S,U spacewhite

 click plan "https://docs.daocloud.io/install/community/resources/"
 click k8s "https://docs.daocloud.io/install/community/kind/online/#kind"
 click tools "https://docs.daocloud.io/install/install-tools/"
 click kind "https://docs.daocloud.io/install/community/kind/online/"
 click s1 "https://docs.daocloud.io/install/community/k8s/online/"
 click s2 "https://docs.daocloud.io/install/community/k8s/offline/"

 click kpanda "https://docs.daocloud.io/kpanda/intro/what/"
 click ghippo "https://docs.daocloud.io/ghippo/intro/what/"
 click insight "https://docs.daocloud.io/insight/intro/what/"
 click free "https://docs.daocloud.io/dce/license0/"
 click ask "https://docs.daocloud.io/install/intro/#_4"
```

!!! tip

- 上图中的蓝色文字可点击跳转
- 可参考[保姆式安装 DCE 5.0](../blogs/dce5-0328.md)

## 3.2 商业版安装流程

DCE 5.0 商业版的安装流程如下图：

```
flowchart TB
 start([fa:fa-user DCE 5.0 商业版
安装流程]) --> deploy[部署规划]
 deploy --> tools[在火种节点上安装依赖项]
 tools --> download[下载离线包]
 download --> config[编辑并配置
clusterConfig.yaml]

 config --> typical[经典模式
多数据中心]
 typical --> mng2[自动创建 K8s 管理集群
安装 Kubean Operator]
 typical --> gsc2[自动创建
K8s 全局服务集群
安装 DCE 常规组件]

 config --> simple[简约模式
单数据中心]
 simple --> k8s1[自动创建
一个 K8s 管理集群]
 simple --> gsc1[自动在此集群
安装所有 DCE 组件]
```

```
classDef grey fill:#ddddd,stroke:#fffff,stroke-width:1px,color:#000000, font-size:15px;
classDef white fill:#fffff,stroke:#000,stroke-width:1px,color:#000,font-weight:bold
classDef spacewhite fill:#fffff,stroke:#fff,stroke-width:0px,color:#000
classDef plain fill:#ddd,stroke:#fff,stroke-width:1px,color:#000;
classDef k8s fill:#326ce5,stroke:#fff,stroke-width:1px,color:#fff;
classDef cluster fill:#fff,stroke:#bbb,stroke-width:1px,color:#326ce5;

class deploy,tools,download,config cluster
class typical,k8s,mng2,gsc2,k8s2 k8s
class simple,start plain
class gsc1,k8s1 grey

click deploy "https://docs.daocloud.io/install/commercial/deploy-plan/"
click tools "https://docs.daocloud.io/install/install-tools/"
click download "https://docs.daocloud.io/install/commercial/download-file/"
click config "https://docs.daocloud.io/install/commercial/clusterconfig/"
click typical,simple "https://docs.daocloud.io/install/commercial/start-install/"
```

## 3.3 联系我们

DaoCloud Enterprise 5.0 还处于发布初期，安装流程可能会有变更。请收藏此页，关注更新动态，更多操作文档也在制作之中。

- 若有任何安装或使用问题，请[提出反馈](#)。
- 欢迎扫描二维码，与开发者畅快交流：

社区版交流群

[下载 DCE 5.0](#){ .md-button .md-button--primary } [申请社区免费体验](#){ .md-button .md-button--primary }

## 4. 安装器 Release Notes

本页列出安装器的 Release Notes，便于您了解各版本的演进路径和特性变化。

### 4.1 2022-2-28

#### v0.5.0

##### 新功能

- 新增 离线包分离 osPackage，需要在集群配置文件中定义 `osPackagePath`
- 新增 支持 addon 离线化，需要在集群配置文件中定义 `addonOfflinePackagePath`
- 新增 离线安装支持操作系统 REHL 8.4、REHL 7.9

##### 优化

- 优化 升级了前置依赖工具的版本

##### 修复

- 修复 安装器命令行 `-j` 参数效验检测问题
- 修复 前置依赖工具安装路径问题
- 修复 主机清单密码纯数字无效问题
- 修复 运行时为 Docker 时，无法拉取内建仓库镜像问题

##### 已知问题

- 非最小化安装的 REHL8 上由于预装的 runc 导致安装器安装失败，临时解决方案：安装前在上述每台节点上执行 `rpm -qa | grep runc && yum remove -y runc`
- 非最小化安装的 REHL8 上有非法的内核参数设置，临时解决方案：安装前在上述每台节点上执行 `eval $(grep -i 'vm.maxmapcount' /etc/sysctl.conf -r /etc/sysctl.d | xargs -L1 | awk -F ':' '{printf("sed -i -r \"s/(%s)/#\1/\" %s; ", $2, $1)}') && sysctl --system`
- helm 并发安装存在潜在风险，失败后无法继续执行安装

### 4.2 2022-12-30

#### v0.4.0

##### 新功能

- 新增 clusterConfig 的语法从 v1alpha1 升级为 v1alpha2，语法有不兼容变更，可以查看文档
- 新增 不再在全局服务集群上安装永久 Harbor 和永久 MinIO
- 新增 火种节点需要永久存在，用户安装 minio、chart museum、registry
- 新增 商业版新增安装 contour 作为 默认的 ingress-controller
- 新增 商业版新增安装 cert-manager
- 新增 支持私钥模式的集群部署
- 新增 支持外置镜像仓库进行部署

## 优化

- 优化 离线包不再包括操作系统的 ISO，需要单独下载，在纯离线的环境下，需要在 clusterConfig 文件中定义 ISO 的绝对路径
- 优化 商业版使用 Contour 作为默认的 ingress-controller
- 优化 MinIO 支持使用 VIP
- 优化 coredns 自动注入仓库 VIP 解析
- 优化 优化离线包制作流程，加速打包 Docker 镜像
- 优化 优化了离线包的大小
- 优化 基础设施支持 1.25：升级 redis-operator、eck-operator、hwameiStor
- 优化 升级到 keycloakx
- 优化 Istio 版本升级至 v1.16.1

## 修复

- 修复 HwameiStor 的 operator 会随机错误清理集群中的 Job
- 优化 mysql 增强半同步，解决频繁主从切换的数据不一致
- 优化 解决 mysql 磁盘被 clusterPedia 写爆
- 优化 coredns 自动注入仓库 VIP 解析
- 优化 公有云上 svc 的修复
- 优化 修复各种子模块的镜像和 helm 问题

## 已知问题

- 默认安装模式下暂不支持未分区的 SSD 盘，如果要支持，需要手工干涉。
- 纯离线环境，默认没有应用商店。请手动将火种节点的 chart-museum 接入到 global 集群，仓库地址：`http://{火种 IP}:8081`，用户名 rootuser，密码 rootpass123
- metallb 社区有已知问题，在主网卡有 dadfailed 的 IPV6 回环地址，metallb 无法工作，安装之前需要确保主网卡没有 dadfailed
- insight-api-server 启动中如果机器太卡，在 Liveness 健康检查周期内，无法完成数据库的初始化（migrate）动作，导致需要手动介入
- clusterConfig 配置文件中里的 iso 路径必须是绝对路径，不支持相对路径
- kubean 默认 k8s 版本和离线包仍然限制在 k8s 1.24 版本，还未能更新到 1.25（PG 暂不支持）
- external-Registry 如果是 Harbor，暂时不会自动创建 Project，需要提前手动创建
- Docker 运行时，无法拉取 built-in 仓库，将在下个版本修复
- 禁用 IPV6 之后，podman 无法启动火种节点的 kind 集群

## 4.3 2022-11-30

---

### v0.3.29

---

## 新功能

- 新增 ARM64 支持：构建 ARM64 离线包
- 新增 支持麒麟 kylin v10 sp2 离线包
- 新增 基础设施支持 1.25：升级 redis-operator、eck-operator、hwameiStor 等组件
- 新增 支持私钥模式的集群部署
- 新增 工作负载基于自定义指标弹性伸缩，更加贴近用户实际业务弹性扩缩容需求



## 优化

- 优化 永久 harbor 使用 operator 创建，开启 HTTPS，并使用 Postgresql operator
- 优化 商业版使用 Contour 作为默认的 ingress-controller
- 优化 MinIO 支持使用 VIP
- 优化 coredns 自动注入仓库 VIP 解析
- 优化 优化离线包制作流程，加速打包 Docker 镜像

## 修复

- 修复 修复了公有云 Service 的问题
- 修复 修复了各个子模块的镜像和 Helm 的问题
- 修复 修复了离线包加载的缺陷修复

## 已知问题

- 因为部分 Operator 需升级到支持 K8s 1.25，导致 DCE 5.0 向下不支持 K8s 1.20
- Kubean 默认 K8s 版本和离线包仍然限制在 1.24 版本，还未能更新到 1.25（由于 postgres-operator 暂不支持）
- Image Load 情况下，istio-ingressgateway imagePullPolicy 为 always
- ARM 版本，不能执行安装脚本的第 16 步（harbor），因为 harbor 暂时不支持 ARM。需要修改 manifest.yaml 文件，postgresql operator 为 fasle，执行安装命令时要添加 `-j 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15`
- 在容器管理界面上创建新集群，无法对接 HTTPS 的仓库，需要手动修改 kubean job 的 ConfigMap 和 CR
- 永久 MinIO 的 PVC 的大小默认是 30G，会不够用（承载 kubean 二进制、ISO 以及 Harbor 镜像仓库），需要进行扩容操作

## 5. 卸载

---

### 5.1 卸载社区版

---

依次执行以下命令从集群卸载 DCE 5.0，卸载过程中不会进行任何备份，请谨慎操作。

```
shell linenums="1"
kubectl -n mcamel-system delete mysql mcamel-common-mysql-cluster
kubectl -n mcamel-system delete elasticsearches mcamel-common-es-cluster-masters
kubectl -n mcamel-system delete redisfailover mcamel-common-redis-cluster
kubectl -n ghippo-system delete gateway ghippo-gateway
kubectl -n istio-system delete requestauthentications ghippo
helm -n mcamel-system uninstall eck-operator mysql-operator redis-operator
helm -n ghippo-system uninstall ghippo
helm -n insight-system uninstall insight-agent insight
helm -n ipavo-system uninstall ipavo
helm -n kpanda-system uninstall kpanda
helm -n istio-system uninstall istio-base istio-ingressgateway istiod
```

## 6. Commercial

### 6.1 集群配置文件 clusterConfig.yaml

此 YAML 文件包含了集群的各项配置字段，安装之前必须先配置此文件。该文件将定义部署的负载均衡类型、部署模式、集群节点信息等关键参数。默认位于 `offline/sample/` 目录。

#### ClusterConfig 示例

以下是一个 ClusterConfig 文件示例。

```
```yaml title="clusterConfig.yaml" apiVersion: provision.daocloud.io/v1alpha2 kind: ClusterConfig metadata: spec: clusterName: my-cluster # (1) loadBalancer: type: metallb # (2) istioGatewayVip: xx.xx.xx.xx/32 # (3) insightVip: xx.xx.xx.xx/32 # (4) # privateKeyPath: /root/.ssh/id_rsa_sample # (5) masterNodes: - nodeName: "g-master1" ip: xx.xx.xx.xx ansibleUser: "" ansiblePass: "" - nodeName: "g-master2" ip: xx.xx.xx.xx ansibleUser: "" ansiblePass: "" - nodeName: "g-master3" ip: xx.xx.xx.xx ansibleUser: "" ansiblePass: "" workerNodes: - nodeName: "g-worker1" ip: xx.xx.xx.xx ansibleUser: "" ansiblePass: "*" nodeTaints: # (6) - "node.daocloud.io/es-only=true:NoSchedule" - nodeName: "g-worker2" ip: xx.xx.xx.xx ansibleUser: "" ansiblePass: "*" nodeTaints: # (7) - "node.daocloud.io/es-only=true:NoSchedule" - nodeName: "g-worker3" ip: xx.xx.xx.xx ansibleUser: "" ansiblePass: "****" nodeTaints: # (8) - "node.daocloud.io/es-only=true:NoSchedule" ntpServer: # (9) - 0.pool.ntp.org - ntpl.aliyun.com - ntp.ntsc.ac.cn registry: type: built-in # (10) # builtinRegistryDomainName: ${跟上述配置仓库地址一致。如果是 built-in，则填写火种节点 IP}。 # (11) # type: external # (12) # externalRegistry: external-registry.daocloud.io # (13) # externalRegistryUsername: admin # (14) # externalRegistryPassword: Harbor12345 # (15) # externalScheme: https # (16) addonOfflinePackagePath: "Please-replace-with-Your-Real-Addon-Offline-Package-PATH-on-bootstrap-Node" # (17) # kubean 所需要的仓库配置 imageConfig: imageRepository: http://${IP_ADDRESS_OF_BOOTSTRAP_NODE} # (18) binaryRepository: http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean # (19) # RPM 或者 DEB 安装的源头 repoConfig: repoType: centos # (20) isoPath: "Please-replace-with-Your-Real-ISO-PATH-on-bootstrap-Node" # (21) osPackagePath: "Please-replace-with-Your-Real-OS-Package-PATH-on-bootstrap-Node" # (22) dockerRepo: "http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/centos/$releasever/os/$basearch" # (23) # dockerRepo: "http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/redhat/$releasever/os/$basearch" # (24) extraRepos: - http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/centos-iso/$releasever/os/$basearch - http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/centos/$releasever/os/$basearch k8sVersion: v1.24.7 auditConfig: logPath: /var/log/audit/kube-apiserver-audit.log logHostPath: /var/log/kubernetes/audit # policyFile: /etc/kubernetes/audit-policy/apiserver-audit-policy.yaml # logMaxAge: 30 # logMaxBackups: 10 # logMaxSize: 100 # policyCustomRules: > # - level: None # users: [] # verbs: [] # resources: [] network: cni: calico clusterCIDR: 100.96.0.0/11 serviceCIDR: 100.64.0.0/13 cri: criProvider: containerd # criVersion: 1.6.8 # (25)
```

1. 集群名称
2. 有 3 个选项: NodePort (default)、metallb、cloudLB (Cloud Controller)，建议生产环境使用 metallb
3. 当 loadBalancer.type 是 metallb 时必须填，为 DCE 提供 UI 和 OpenAPI 访问权限
4. 别丢弃/32，当 loadBalancer.type 是 metallb 时必须填，用作 Global 集群的 Insight 数据采集入口，子集群的 insight-agent 可以向这个 VIP 报告数据
5. 指定 ssh 私钥，定义后无需再定义节点的 ansibleUser、ansiblePass
6. 对于 7 节点模式：至少 3 个工作节点应该被打上该污点，仅作为 ES 的工作节点
7. 对于 7 节点模式：至少 3 个工作节点应该被打上该污点，仅作为 ES 的工作节点
8. 对于 7 节点模式：至少 3 个工作节点应该被打上该污点，仅作为 ES 的工作节点
9. 可以使用自己搭建的 ntpServer
10. 支持 3 个选项: built-in, external, online
11. 可选。内置镜像仓库的域名，并在每个节点的 /etc/hosts 区域进行域名解析的配置
12. 使用已有的仓库，需要保证网络联通

13. 已有镜像仓库的 IP 地址或者域名
14. 只有 type: external 且推镜像时需要用户名和密码的情况下才需要定义此项
15. 只有 type: external 且推镜像时需要用户名和密码的情况下才需要定义此项
16. 这是一个占位符
17. addon 离线包文件的绝对路径，如果不需要 addon 离线化可以注释掉
18. 如果选择了已有仓库，需要填写外部镜像仓库地址
19. 如果选择了已有仓库，需要填写外部 MinIO 地址
20. `centos` 表示使用 CentOS、RedHat、kylin AlmaLinux 或 Fedora；`debian` 表示使用 Debian；`ubuntu` 表示使用 Ubuntu
21. 操作系统 ISO 文件的绝对路径，不得为空
22. 操作系统 osPackage 文件的绝对路径，不得为空
23. 如果是 kylin，安装器将会选择 containerd，所以需要将 dockerRepo 设置为空
24. 如果是 redhat
25. criVersion 仅在 online 模式下生效，请勿将其设置为 offline 模式

关键字段

该 YAML 文件中的关键字段说明，请参阅下表。

字段	说明	默认值
clusterName	在 KuBean Cluster 里的 Global 集群命名	-
loadBalancer.type	所使用的 LoadBalancer 的模式，物理环境用 metallb，POC 用 NodePort，公有云和 SDN CNI 环境用 cloudLB	NodePort (default)、metallb、cloudLB (Cloud Controller)
loadBalancer.istioGatewayVip	如果负载均衡模式是 metallb，则需要指定一个 VIP，供给 DCE 的 UI 界面和 OpenAPI 访问入口	-
loadBalancer.insightVip	如果负载均衡模式是 metallb，则需要指定一个 VIP，供给 Global 集群的 insight 数据收集入口使用，子集群的 insight-agent 可上报数据到这个 VIP	-
privateKeyPath	KuBean 部署集群的 SSH 私钥文件路径，如果填写则不需要定义 ansibleUser、ansiblePass	-
masterNodes	Global 集群: Master 节点列表，包括 nodeName/ip/ansibleUser/ansiblePass 几个关键字段	-
workerNodes	Global 集群: Worker 节点列表，包括 nodeName/ip/ansibleUser/ansiblePass 几个关键字段	-
registry.type	k8s 组件和 DCE 组件的镜像拉取仓库的类型，有 online (在线环境)、built-in (使用火种节点内置的仓库)、external (使用已有的外置仓库)	online
registry.builtinRegistryDomainName	如果使用 built-in (使用火种节点内置的仓库)，如果需要自动化植入仓库域名到各个节点的 /etc/hosts 和 CoreDNS 的 hosts，可指定域名	-
registry.externalRegistry	如果 registry.type = external 则需要定义该字段需要指定 external 仓库的 IP 或者域名(使用已有的外置仓库)，如果使用 Harbor，需要提前创建相应 Project	-
registry.externalRegistryUsername	如果 registry.type = external 则需要定义该字段外置仓库的用户名，用于推送镜像	-
registry.externalRegistryPassword	如果 registry.type = external 则需要定义该字段外置仓库的密码，用于推送镜像	-
imageConfig.imageRepository	如果是离线安装，KuBean 安装集群时的本地镜像仓库来源	-
imageConfig.binaryRepository	如果是离线安装，KuBean 安装集群时的本地二进制仓库来源	[https://files.m.daocloud.io] (https://files.m.daocloud.io/)
repoConfig	RPM 或者 DEB 安装的源头，如果离线模式下，是安装器启动的 MinIO	-
repoConfig.`repoType`	支持 `centos` (包含CentOS, RedHat,kylin AlmaLinux or Fedora)、`debian`、`ubuntu`	-
repoConfig.`dockerRepo`	如果是 kylin，安装器将会选择 containerd，所以需要将 dockerRepo 设置为空	-
repoConfig.isoPath	操作系统 ISO 文件的路径，离线模式下不能为空	-
k8sVersion	KuBean 安装集群的 K8s 版本必须跟 KuBean 和离线包相匹配	-
ntpServer	可用的 NTP 服务器，供给新节点同步时间	-
network.cni	CNI 选择，比如 Calico、Cilium	calico
network.clusterCIDR	Cluster CIDR	-
network.serviceCIDR	Service CIDR	-
auditConfig	k8s api-server 的审计日志配置	默认关闭

场景配置说明

负载均衡选择

支持 3 个选项: NodePort、metallb、cloudLB

```

```yaml
loadBalancer:
 type: metallb # (1)
 istioGatewayVip: xx.xx.xx.xx/32 # (2)
 insightVip: xx.xx.xx.xx/32 # (3)

 # 如果 loadBalancer 是 NodePort
 type: NodePort

 # cloudLB 特性目前处于 todo 状态

```

1. 支持 3 个选项: NodePort(default), metallb, cloudLB (Cloud Controller)
2. 如果 loadBalancer != metallb，请移除这一行
3. 记住要保留 /32

## 1/4/7 节点模式

```
all in one 模式
masterNodes:
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"

3 节点模式
masterNodes:
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"

7 节点模式
masterNodes:
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 - nodeName: "g-master1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
workerNodes:
 - nodeName: "g-worker1"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 nodeTaints: # (1)
 - "node.daocloud.io/es-only=true:NoSchedule"
 - nodeName: "g-worker2"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 nodeTaints: # (2)
 - "node.daocloud.io/es-only=true:NoSchedule"
 - nodeName: "g-worker2"
 ip: xx.xx.xx.xx
 ansibleUser: "root"
 ansiblePass: "dangerous"
 nodeTaints: # (3)
 - "node.daocloud.io/es-only=true:NoSchedule"
```

1. 对于 7 节点模式：至少 3 个工作节点应该被打上该污点，仅作为 ES 的工作节点
2. 对于 7 节点模式：至少 3 个工作节点应该被打上该污点，仅作为 ES 的工作节点
3. 对于 7 节点模式：至少 3 个工作节点应该被打上该污点，仅作为 ES 的工作节点

## 镜像仓库模式

支持三种模式：online, built-in, external

```
registry:

 type: online # (1)

 type: built-in # (2)
 builtinRegistryDomainName: # (3)

 type: external # (4)
 externalRegistry: external-registry.daocloud.io # (5)
 externalRegistryUsername: admin # (6)
```

```
externalRegistryPassword: Harbor12345 # (7)
externalScheme: https # (8)
```

1. 在线模式，设置为在线模式后，无需定义 spec.imageConfig、spec.repoConfig
2. 使用内置的仓库，由安装器进行部署安装
3. 可选。内置镜像仓库的域名，并在每个节点的 /etc/hosts 和 coredns 的 hosts 区域进行域名解析的配置
4. 使用已有的仓库，需要保证网络联通
5. 已有镜像仓库的 IP 地址或者域名
6. 只有 type: external 且推镜像时需要用户名和密码的情况下才需要定义此项
7. 只有 type: external 且推镜像时需要用户名和密码的情况下才需要定义此项
8. 这是一个占位符

## Kubean 组件安装集群配置

```
kubean 所需要的仓库配置
imageConfig:
 imageRepository: http://${IP_ADDRESS_OF_BOOTSTRAP_NODE} 或者 上述“自定义的内置域名” # (1)
 binaryRepository: http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean # (2)

RPM 或者 DEB 安装的源头
repoConfig:
 repoType: centos # (3)
 isoPath: "Please-replace-with-Your-Real-ISO-PATH-on-bootstrap-node" # (4)
 osPackagePath: "Please-replace-with-Your-Real-OS-Package-PATH-on-bootstrap-node" # (5)
 dockerRepo: "http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/centos/$releasever/os/$basearch"

 # dockerRepo: "" # (6)
 # dockerRepo: "http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/redhat/$releasever/os/$basearch" # (7)

extraRepos:
 - http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/centos-iso/$releasever/os/$basearch
 - http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/centos/$releasever/os/$basearch

 # 如果系统是 RedHat 8, 需要使用以下参数
 #- http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/redhat-iso/$releasever/os/$basearch/AppStream
 #- http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/redhat-iso/$releasever/os/$basearch/BaseOS
 #- http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/redhat/$releasever/os/$basearch

 # 如果系统是 kylin, 需要使用以下参数
 #- http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/kylin-iso/$releasever/os/$basearch
 #- http://${IP_ADDRESS_OF_BOOTSTRAP_NODE}:9000/kubean/kylin/$releasever/os/$basearch
```

1. 如果选择了已有的仓库，需要填写外部镜像仓库地址
2. 如果选择了已有的仓库，需要填写外部 MinIO 地址
3. centos 表示使用 CentOS、RedHat、kylin AlmaLinux 或 Fedora；debian 表示使用 Debian；ubuntu 表示使用 Ubuntu
4. 操作系统 ISO 文件路径，不得为空
5. 操作系统 osPackage 文件的绝对路径，不得为空
6. 如果是 kylin，安装器将会选择 containerd，所以需要将 dockerRepo 设置为空
7. 如果是 redhat

## 6.2 部署架构

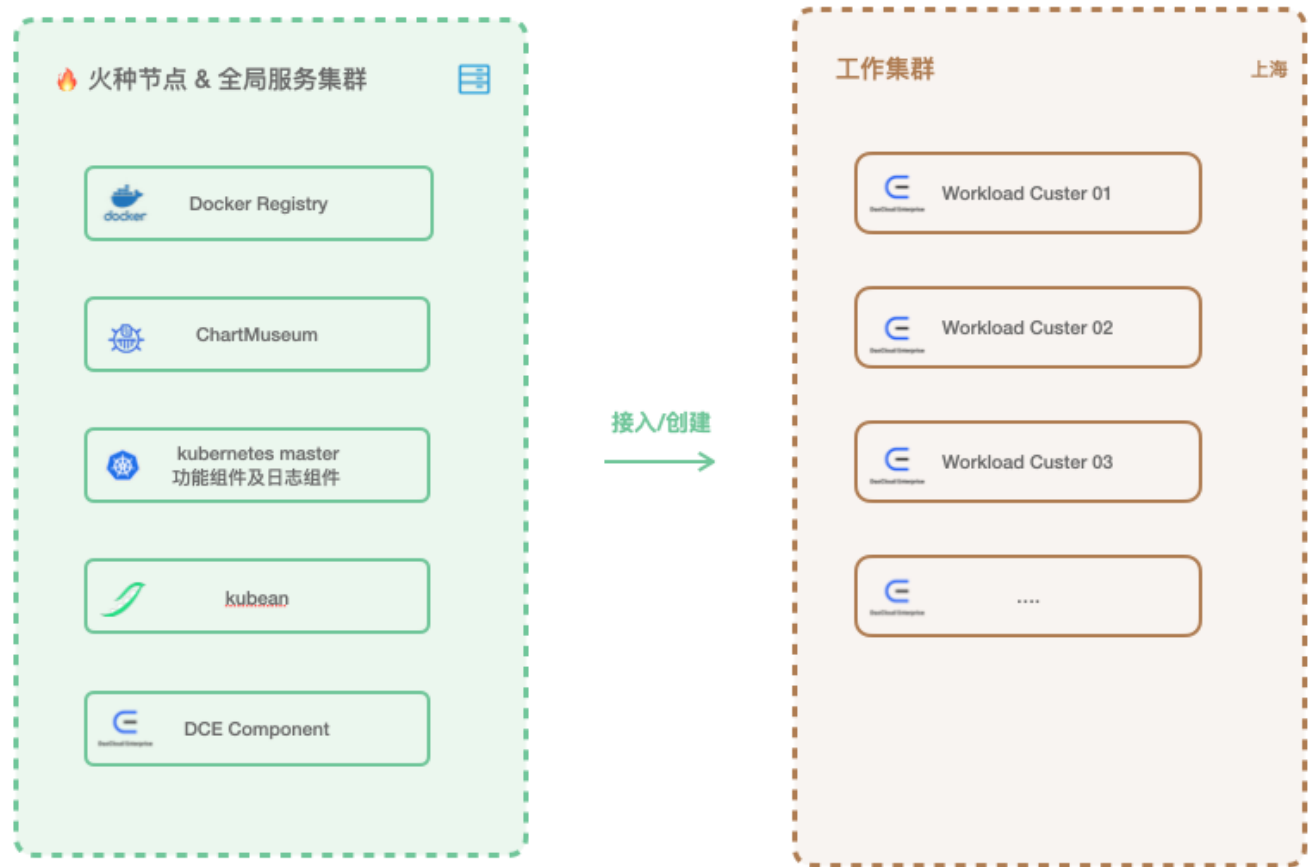
DCE 5.0 提供了三种部署架构：all-in-one、4 节点、7 节点。

以下描述了架构中涉及到名词介绍：

名词	介绍
火种节点	火种节点也称为 bootstrap 节点，安装部署程序的执行，并运行平台所需的镜像仓库和 chart museum。
全局服务集群	用来部署 DCE 5.0 的所有组件，以及Kubean，其中 Kubean 用来管理集群的生命周期。
工作集群	支撑业务应用和服务（成功安装 DCE 之后再部署工作集群）
DCE Componet	DCE 5.0 的所有组件，包含全局管理、容器管理、可观测性、应用工作台、多云编排、镜像仓库、微服务引擎、服务网格、中间件等产品模块。

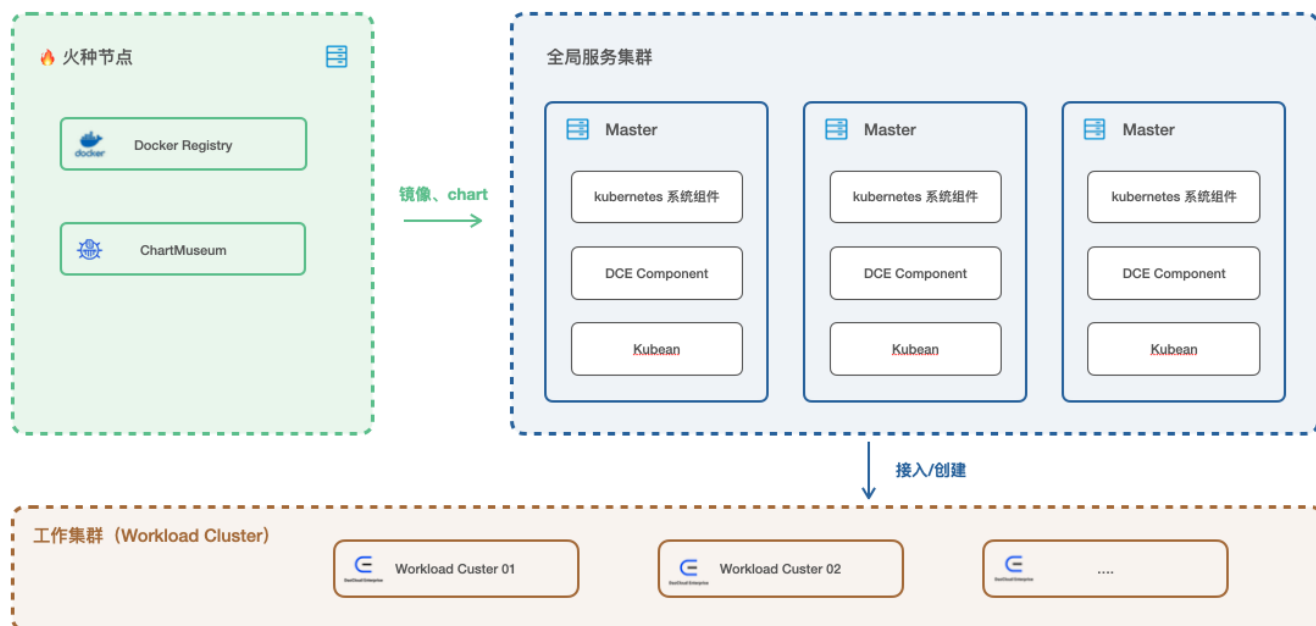
### all-in-one

all-in-one 模式仅需要一台主机，仅建议个人客户体验时采用此模式进行安装 DCE 5.0。使用 all-in-one 模式时，建议使用最小化安装 DCE 5.0，即安装命令后添加 `-z` 参数。



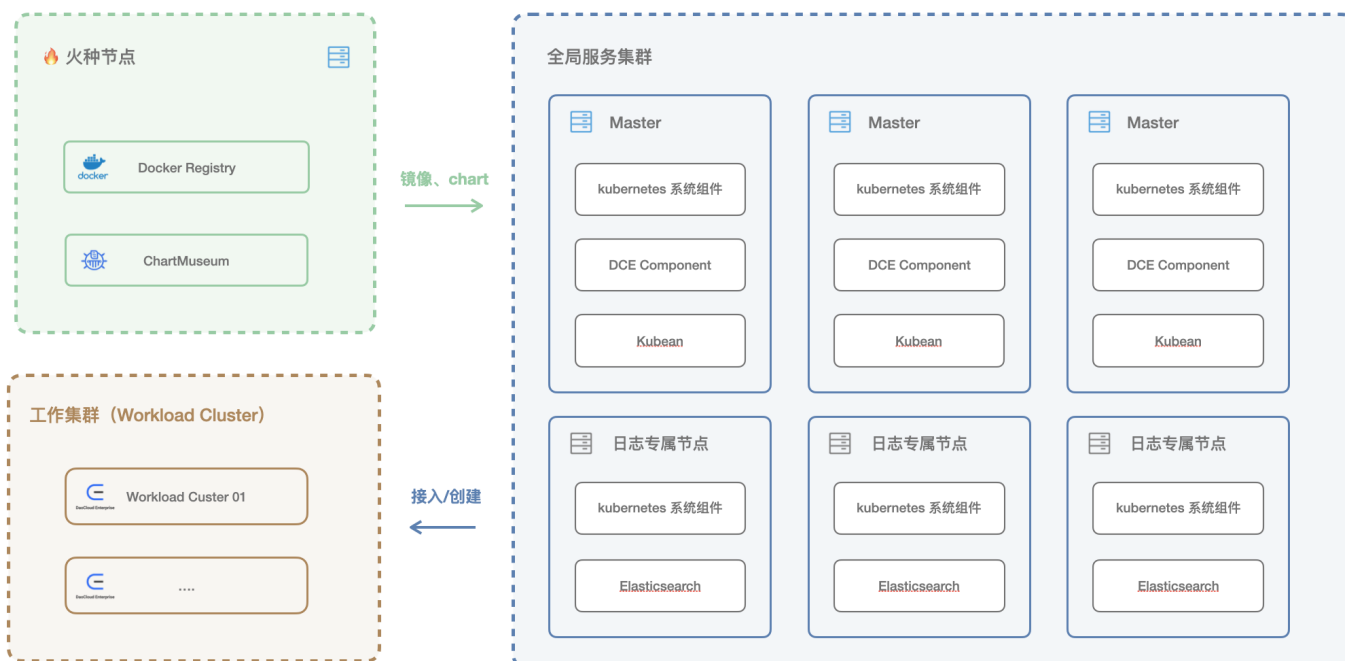
### 4 节点模式

4 节点模式由 1 个火种节点、3 个集群的主节点组成，仅建议 POC 或测试环境时采用此模式进行安装 DCE 5.0。



## 7 节点模式 (1 + 6)

7 节点模式由 1 个火种节点、3 个集群的主节点、3 个集群的工作节点组成，其中工作节点作为 Elasticsearch 组件的专属节点。建议客户生产环境中采用此模式进行安装 DCE 5.0。





## 6.3 部署规划

部署 DCE 5.0 时需要先做好软件规划、硬件规划、网络规划。

### 软件规划

#### 服务器系统要求

架构	操作系统	内核版本	备注
AMD 64	centos 7.X	Kernel 3.10.0-1127.el7.x86_64 on an x86_64	操作系统推荐 CentOS 7.9
	Redhat 8.X	4.18.0-305.el8.x86_64	操作系统推荐 Redhat 8.4
	Redhat 7.X	3.10.0-1160.el7.x86	操作系统推荐 Redhat 7.9
ARM 64	银河麒麟 OS V10 SP2	4.19.90-24.4.v2101.ky10.aarch64	-

#### 客户端浏览器的要求

- Firefox  $\geq$  49
- Chrome  $\geq$  54

### 硬件规划

类型	具体要求
cpu	不得超售
内存	不得超售
硬盘	iops > 500 吞吐量 >200 M/s

### 网络规划

#### 使用 metallb 作为全局集群的负载均衡方式

资源	要求	说明
istioGatewayVip	1 个	如果负载均衡模式是 metallb, 则需要指定一个 VIP, 供给 DCE 的 UI 界面和 OpenAPI 访问入口
insightVip	1 个	如果负载均衡模式是 metallb, 则需要指定一个 VIP, 供给 Global 集群的 insight 数据收集入口使用, 子集群的 insight-agent 可上报数据到这个 VIP
网络速率	1000 M/s	不低于千兆, 建议万兆。
协议	-	支持 IPv6。
ip 地址范围	-	服务器不得使用 172.16-32 网段的 ip, 如果已经使用, 无法更改, 就需要修改每一台服务器上的 docker 配置, 加上 bip 参数, 才能绕过这个 ip 段。Kubernetes 集群使用 10.96.0.0/12 作为 cluster IP 范围段, 客户不能使用。
路由	-	服务器有 default 或指向 0.0.0.0 这个地址的路由。

## 6.4 文件下载

本页列出了安装器和镜像包的下载地址。

### 下载离线镜像包

#### AMD 架构离线镜像包

1. 在火种节点上下载离线镜像包，可从[下载中心](#)下载其他版本的离线包。

```
```shell
```

6.5 假定 VERSION 为 v0.4.0

```
export VERSION=v0.4.0 wget https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/offline-centos7-$VERSION-amd64.tar
```
```

2. 解压离线包。

```
bash
tar -xvf offline-centos7-$VERSION-amd64.tar
```

#### ARM 架构离线镜像包

1. 在火种节点上下载离线镜像包，可从[下载中心](#)下载其他版本的离线包。

```
```shell
```

6.6 假定 VERSION 为 v0.4.0

```
export VERSION=v0.4.0 wget https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/offline-kylin-v10sp2-$VERSION-
arm64.tar ```
```

2. 解压离线包。

```
bash
tar -xvf offline-kylin-v10sp2-$VERSION-arm64.tar
```

下载 ISO 离线包

CPU 架构	操作系统版本	下载地址
AMD64	Centos 7	https://mirrors.tuna.tsinghua.edu.cn/centos/7.9.2009/isos/x86_64/CentOS-7-x86_64-DVD-2009.iso
ARM64	Kylin Linux Advanced Server release V10 (Sword) SP2	申请地址: https://www.kylinos.cn/scheme/server/1.html 注意: 麒麟操作系统需要提供个人信息才能下载使用, 下载时请选择 V10 (Sword) SP2

6.7 准备工作

一下描述了部署 DCE 5.0 需要进行的准备工作。

机器准备

all-in-One 模式

数量	服务器角色	服务器用途	cpu 数量	内存容量	系统硬盘	未分区的硬盘
1	all in one	镜像仓库、chart museum 、global 集群本身。	16 core	32G	200G	400G

4 节点模式

数量	服务器角色	服务器用途	cpu 数量	内存容量	系统硬盘	未分区的硬盘
1	火种节点	1. 执行安装部署程序 2. 运行平台所需的镜像仓库和chart museum。	2	4G	200G	-
3	master	运行 DCE 5.0 组件 2. 运行kubernetes 系统组件	8	16G	100G	200G

7 节点模式

数量	服务器角色	服务器用途	cpu 数量	内存容量	系统硬盘	未分区的硬盘
1	火种节点	1. 执行安装部署程序 2. 运行平台所需的镜像仓库和chart museum。	2	4G	200G	-
3	master	运行 DCE 5.0 组件 2. 运行kubernetes 系统组件。	8	16G	100G	200G
3	worker	单独运行日志相关组件。	8	16G	100G	-

前置检查

机器检查

检查项	具体要求	说明
用户权限	root	必须使用 root 用户部署，各个服务器也必须允许 root 用户 ssh 登录。
swap	关闭	如果不满足，系统会有一定几率出现 io 飙升，造成 容器运行时 卡死。
防火墙	关闭（不强制）	-
selinux	关闭（不强制）	-
时间同步	所有集群节点要求时间必须同步	docker 和 Kubernetes 官方要求。否则 kube.conf 会报错 Unable to connect to the server: x509: certificate has expired or is not yet
时区	所有服务器时区必须统一	建议设置为 Asia/Shanghai。 参考命令：timedatectl set-timezone Asia/Shanghai
nameserver	/etc/resolv.conf 至少有一个 nameserver	coredns 要求，否则会有报错。该 nameserver 在纯离线环境下可以是一个不存在的 IP 地址。Centos8minial默认没有/etc/resolv文件，需要手动创建

火种机器依赖组件检查

检查项	版本要求	说明
podman	v4.4.1	-
helm	≥ 3.11.1	-
skopeo	≥ 1.11.1	-
kind	v0.17.0	-
kubect1	≥ 1.25.6	-
yq	≥ 4.31.1	-
minio client	-	mc.RELEASE.2023-02-16T19-20-11Z

如果不存在依赖组件，通过脚本进行安装依赖组件。

```
# 下载脚本
curl -LO https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/install_prerequisite.sh

# 添加可执行权限
chmod +x install_prerequisite.sh

# 开始安装
bash install_prerequisite.sh online full
```

6.8 集群部署模式使用场景

DCE 5.0 提供了**四种集群角色**，用以满足不同的使用场景。用户可以基于自身业务特性和基础设施场景，自由组合不同的集群部署组合。

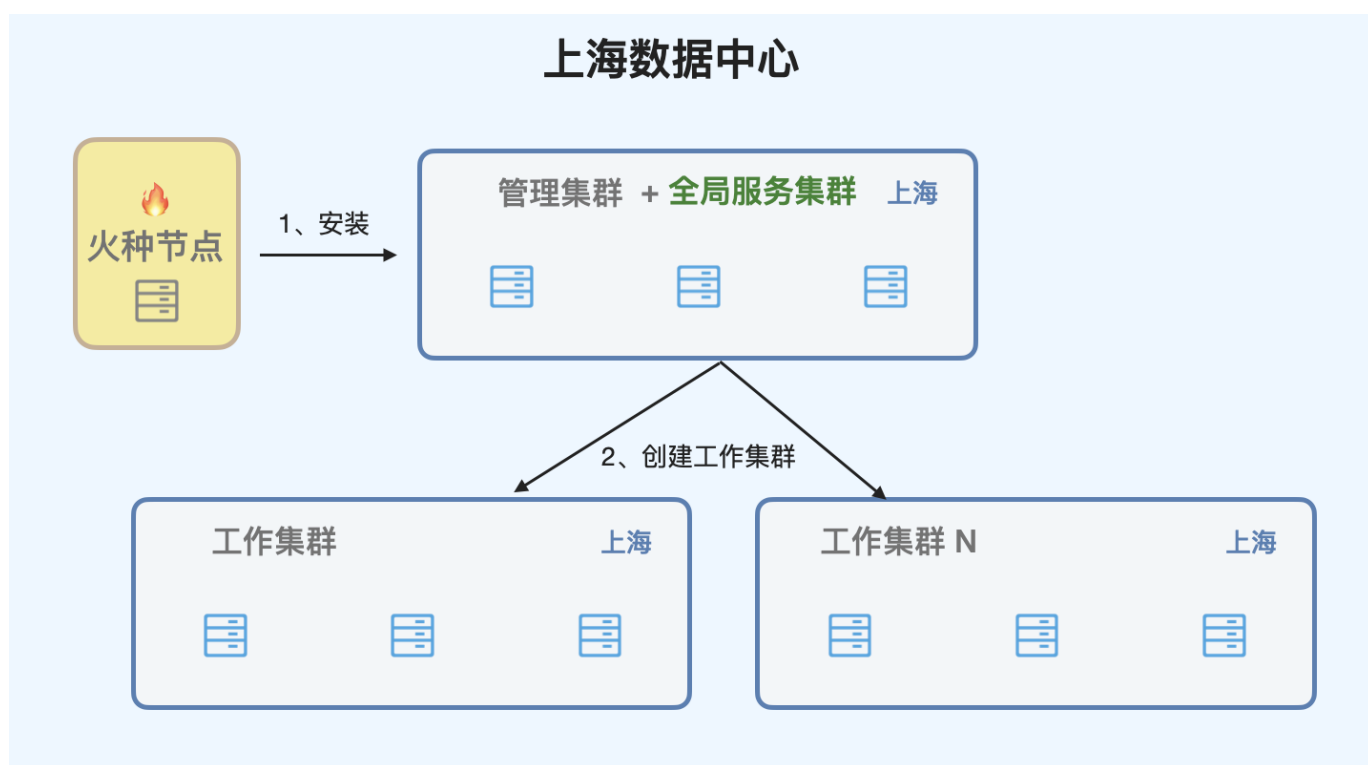
本文将对几种常用使用场景进行说明。

单数据中心场景

当用户大部分业务集中在某一个区域的单数据中心时，无跨网需求，且对数据容灾要求较低的场景下。推荐使用简约模式。即只需一个集群用于运行平台相关组件（全局服务集群+管理集群二合一），另外部署多个集群用于运行业务负载（工作集群）。单个集群节点数量由具体业务场景决定，推荐每个集群使用 3 个 master 节点，以实现使用高可用。

具体部署方案依据业务需要，参考如下部署流程：

单数据中心单管理集群

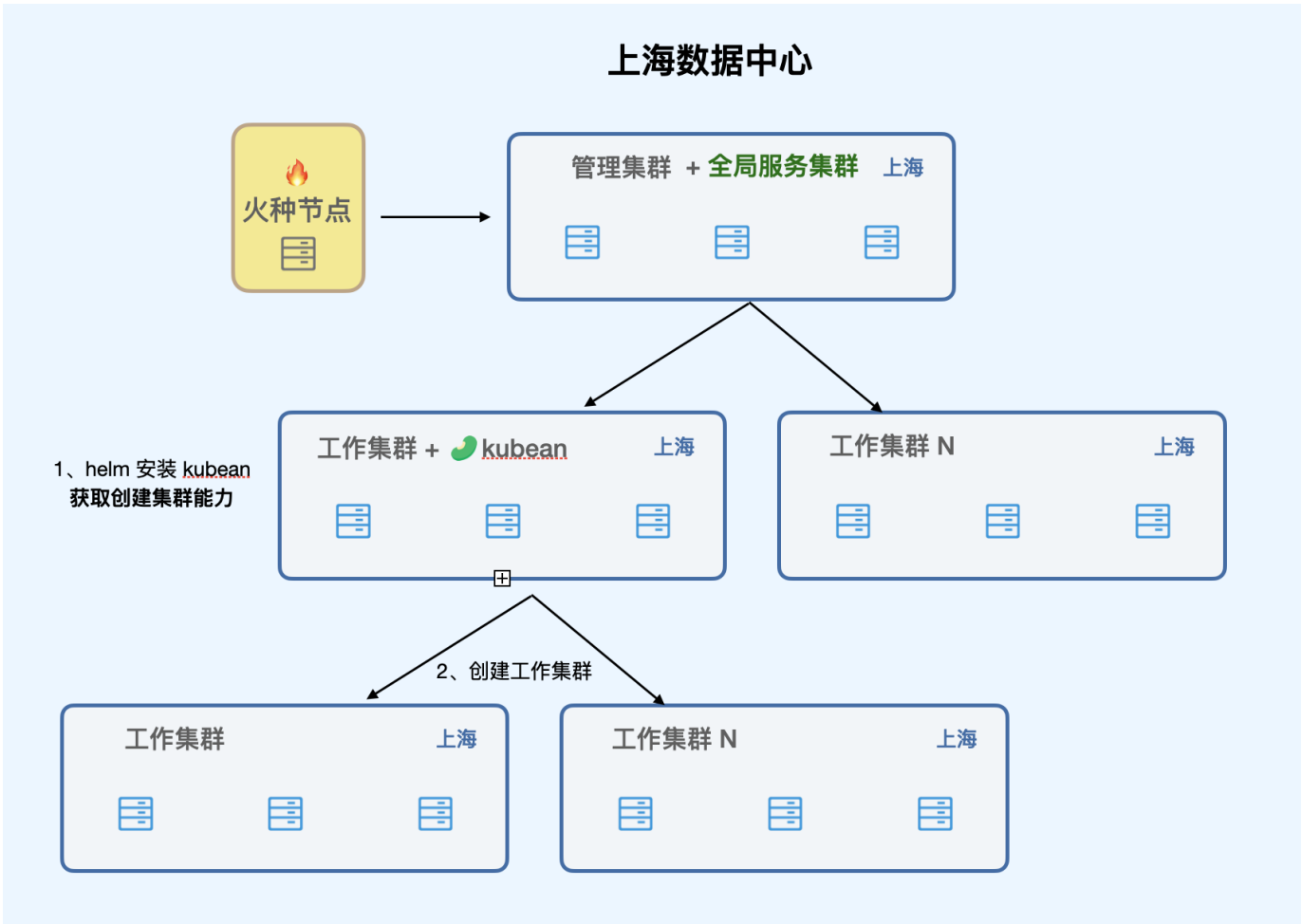


前提：准备一台节点，节点的操作系统和架构需与待创建集群的节点保持一致。

1. 在火种节点部署安装器，使用安装器安装一个包含全局服务集群和管理集群两个集群角色的集群。
2. 按需在平台下的容器管理模块上基于管理集群创建一个或多个工作集群。

单数据中心多管理集群

当用户需要在当前单数据中心内新增一个管理集群用于管理新业务集群的生命周期，此时无需再次使用火种节点的方式进行安装，只需要在已经创建的某个工作集群上安装 kubean Operator 组件即可赋予这个集群管理集群的能力和角色。如下图：



前提：已经完成上一步单数据中心单管理集群的部署。

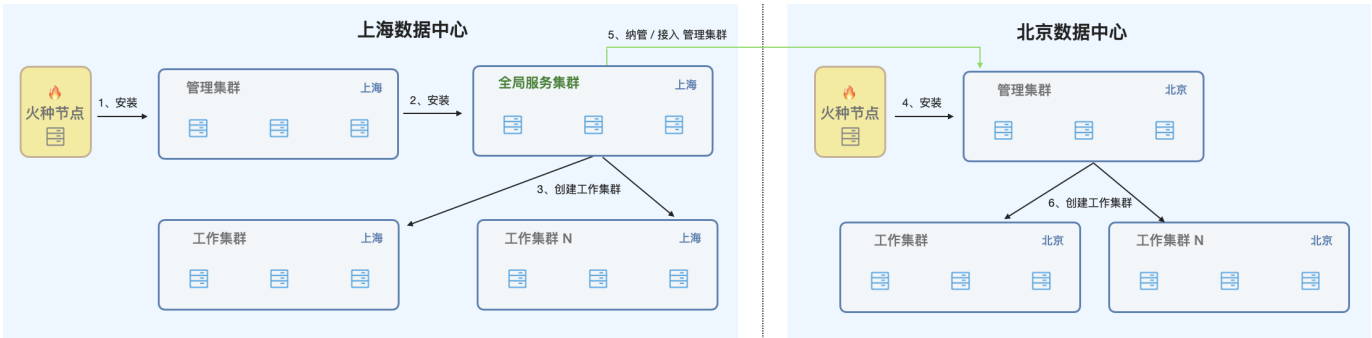
1. 在某个工作集群详情界面使用 `helm 模板` 安装 `kubean`，等待 `kubean` 状态变为运行中。
2. 当前工作集群安装完成 `kubean` 后，集群角色将自动变为管理集群、按需在平台下的容器管理模块上基于管理集群创建一个或多个工作集群。

多数据中心场景

当用户存在多个数据中心，或不同数据中心网络存在隔离，例如在两地三中心等容灾场景下，用户在不同区域、不同数据中心有集群生命周期管理需求。推荐使用经典模式。此时可以在不同的数据中心或区域各部署一个管理集群，并将所有的管理集群接入进全局服务集群进行纳管，从而实现统一管理不同区域内集群的生命周期。

多数据中心多管理集群

前提：准备一台节点，节点的操作系统和架构需与待创建集群的节点保持一致。



上海数据中心

1. 在火种节点部署安装器，使用安装器安装一个管理集群。
2. 管理集群安装完成后将基于集群配置问价自动创建一个全局服务集群。
3. 按需在平台下的容器管理模块上基于管理集群创建一个或多个工作集群。

在跨区域和网络隔离场景下，需要统一纳管其它据中心的集群生命生命周期，请参考如下配置流程。

北京数据中心

前提：在北京数据中心准备一台节点，节点的操作系统和架构需与待创建集群的节点保持一致。

1. 在火种节点部署安装器，使用安装器安装一个管理集群。
2. 在平台下的容器管理模块（运行在上海数据中心）接入北京数据中心新安装的管理集群。
3. 按需在北京（运行在上海数据中心）下的容器管理模块上基于北京数据中心于管理集群创建一个或多个工作集群。

6.9 离线安装 DCE 5.0 商业版

请在安装之前确保您以阅读并理解[部署规划](#)、[部署架构](#)、[准备工作](#)

请查看[产品发布记录](#)，避免所安装版本的已知问题，以及查看新增功能

离线安装步骤

第 1 步：下载离线包

请根据业务环境下载对应的离线包。

离线镜像包

可以在[下载中心](#)下载最新版本并使用

CPU 架构	版本	下载地址
AMD64	v0.5.0	https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/offline-v0.5.0-amd64.tar
ARM64	v0.5.0	https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/offline-v0.5.0-arm64.tar

下载完毕后解压离线包：

```
# 以 amd64 架构离线包为例
tar -xvf offline-v0.5.0-amd64.tar
```

ADDON 离线包

安装器从 v0.5.0 版本，支持了 addon 的离线包导入能力，如果支持 addon 中所有的 helm chart 离线化。

首先需要事先下载好离线包，并在集群配置文件（clusterConfig.yaml）中定义 `addonOfflinePackagePath`。

CPU 架构	版本	下载地址
AMD64	v0.5.2	https://qiniu-download-public.daocloud.io/DaoCloud_DigitalX_Addon/addon-offline-full-package-v0.5.2-amd64.tar.gz
ARM64	v0.5.2	https://qiniu-download-public.daocloud.io/DaoCloud_DigitalX_Addon/addon-offline-full-package-v0.5.2-arm64.tar.gz

!!! note

目前 addon 的离线包最新版本为 v0.5.2，安装器 v0.5.0 支持 addon 的 v0.5.X 版本。

ISO 离线包

ISO 离线包需要在集群配置文件中配置，请根据操作系统进行下载。

CPU 架构	操作系统版本	下载地址
AMD64	Centos 7	https://mirrors.tuna.tsinghua.edu.cn/centos/7.9.2009/isos/x86_64/CentOS-7-x86_64-DVD-2009.iso
	Redhat 7/8	https://developers.redhat.com/products/rhel/download#assembly-field-downloads-page-content-61451 注意：Redhat 操作系统需要 Redhat 的账号才可以下载
ARM64	Kylin Linux Advanced Server release V10 (Sword) SP2	申请地址： https://www.kylinos.cn/scheme/server/1.html 注意：麒麟操作系统需要提供个人信息才能下载使用，下载时请选择 V10 (Sword) SP2

OSPACKAGE 离线包

安装器从 v0.5.0 版本，需要提供操作系统的 osPackage 离线包，并在集群配置文件（clusterConfig.yaml）中定义 `osPackagePath`。

其中 Kubean 提供了不同操作系统的osPackage 离线包，可以前往 <https://github.com/kubean-io/kubean/releases/tag/v0.4.4> 查看。

操作系统版本	下载地址
Centos 7	https://github.com/kubean-io/kubean/releases/download/v0.4.4/os-pkgs-centos7-v0.4.4.tar.gz
Redhat 8	https://github.com/kubean-io/kubean/releases/download/v0.4.4/os-pkgs-redhat8-v0.4.4.tar.gz
Redhat 7	https://github.com/kubean-io/kubean/releases/download/v0.4.4/os-pkgs-redhat7-v0.4.4.tar.gz
Kylin Linux Advanced Server release V10 (Sword) SP2	https://github.com/kubean-io/kubean/releases/download/v0.4.4/os-pkgs-kylinv10-v0.4.4.tar.gz

第 2 步：配置集群配置文件

集群配置文件会在离线镜像包 offline/sample 目录下，集群配置文件具体的参数介绍请参考 [clusterConfig.yaml](#)。

!!! note

目前离线镜像包中提供了标准的 7 节点模式模板。

第 3 步：开始安装

1. 执行以下命令开始安装 DCE 5.0，安装器二进制文件位置为 `offline/dce5-installer`

```
shell
./offline/dce5-installer cluster-create -c ./offline/sample/clusterConfig.yaml -m ./offline/sample/manifest.yaml -p ./offline/
```

!!! note

安装器脚本命令说明：

- c 来指定集群配置文件，必选
- m 参数指定 manifest 文件，
- p 指定离线镜像文件
- z 最小化安装
- d 开启 debug 模式
- 更多命令请使用 --help 查询

2. 安装完成后，命令行会提示安装成功。恭喜您！现在可以通过屏幕提示的 URL 使用默认的用户和密码（admin/changeme）探索全新的 DCE 5.0 啦！

```

5:18PM: TEST SUITE: None
5:18PM: NOTES:
5:18PM:
5:18PM: _ _ _ _ _
5:18PM: | _ |   C) | _ | _ | / _ | _ | _ |
5:18PM: | _ | _ _ _ _ _ | _ | _ | _ | _ | _ |
5:18PM: | _ | _ \ _ / _ \ _ \ _ | _ | _ / _ \ _ \ _ |
5:18PM: | _ | _ | _ \ _ | C) | _ | _ | C) | C) |
5:18PM: | _ | _ | _ | _ | _ | _ | _ | _ \ _ \ _ \ _ |
5:18PM: | _ |
5:18PM: | _ |
5:18PM:
5:18PM: 1. Get the Insight UI URL by running these commands:
5:18PM: export POD_NAME=$(kubectl get pods --namespace insight-system -l "app.kubernetes.io/name=ui,app.kubernetes.io/instance=insight" -o jsonpath="{.items[0].metadata.name}")
5:18PM: export CONTAINER_PORT=$(kubectl get pod --namespace insight-system $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
5:18PM: kubectl --namespace insight-system port-forward $POD_NAME 8080:$CONTAINER_PORT
5:18PM: echo "Visit http://127.0.0.1:8080 to use your application"
5:18PM: Install IPavo Dashboard v0.4.1 🐙
5:18PM: "ipavo" already exists with the same configuration, skipping
5:18PM: Hang tight while we grab the latest from your chart repositories...
5:18PM: ...Successfully got an update from the "ipavo" chart repository
5:18PM: Update Complete. *Happy Helming!*
5:18PM: Release "ipavo" does not exist. Installing it now.
5:18PM: NAME: ipavo
5:18PM: LAST DEPLOYED: Mon Sep 19 17:18:37 2022
5:18PM: NAMESPACE: ipavo-system
5:18PM: STATUS: deployed
5:18PM: REVISION: 1
5:18PM: NOTES:
5:18PM: 1. Get the application URL by running these commands:
5:18PM: Install Insight Agent v0.9.4
5:18PM: "insight" already exists with the same configuration, skipping
5:18PM: Hang tight while we grab the latest from your chart repositories...
5:18PM: ...Successfully got an update from the "insight" chart repository
5:18PM: Update Complete. *Happy Helming!*
5:18PM: Release "insight-agent" does not exist. Installing it now.
5:19PM: NAME: insight-agent
5:19PM: LAST DEPLOYED: Mon Sep 19 17:19:01 2022
5:19PM: NAMESPACE: insight-system
5:19PM: STATUS: deployed
5:19PM: REVISION: 1
5:19PM: -----
5:19PM: [!] Use your Web Browser to access DaoCloud Enterprise 5th Portal at : http://172.30.47.104:31227
5:19PM: -----
5:19PM: -----
5:19PM: All set. Enjoy your journey to cloud native with DaoCloud Enterprise 5th !
5:19PM: -----
[root@localhost ~]#

```

```
!!! success
```

请记录好提示的 URL，方便下次访问。

3. 成功安装 DCE 5.0 商业版之后, 请联系我们授权: 电邮 info@daocloud.io 或致电 400 002 6898。

7. Community

7.1 集群资源规划

您可以在标准 Kubernetes 集群上安装 DCE 5.0，也可以在 kind 集群上安装 DCE 5.0。

标准 Kubernetes 集群资源的要求如下：

资源项	要求
节点硬件	3 Master + 2 Worker Master 节点: 4 Core, 8 G; 系统盘 100G Worker 节点: 4 Core, 8 G; 系统盘 100G
K8s 版本	推荐 K8s 官方最高稳定版本，目前推荐 v1.24，最低支持 v1.20
支持的 CRI	Docker 和 containerd
CPU 架构	x86_64，暂时不支持 ARM

kind 集群资源的要求如下，仅推荐用于测试和开发。

资源项	要求
节点硬件	CPU > 10 核、内存 > 12 GB、磁盘空间 > 100 GB
K8s 版本	推荐 K8s 官方最高稳定版本，目前推荐 v1.24，最低支持 v1.20
支持的 CRI	Docker 和 containerd
CPU 架构	x86_64，暂时不支持 ARM

7.2 K8s

使用 Kubernetes 集群离线安装社区版

本页简要说明 DCE 5.0 社区版的离线安装步骤。

点击[社区版部署 Demo](#)可观看视频演示。

准备工作

- 准备一个 Kubernetes 集群，集群配置请参考文档[资源规划](#)。

!!! note

- 存储：需要提前准备好 StorageClass，并设置为默认 SC
- 确保集群已安装 CoreDNS
- 如果是单节点集群，请确保您已移除该节点的污点

- 安装依赖项。

!!! note

如果集群中已安装所有依赖项，请确保依赖项版本符合要求：

- helm ≥ 3.11.1
- skopeo ≥ 1.11.1
- kubect1 ≥ 1.25.6
- yq ≥ 4.31.1

下载和安装

1. 在 k8s 集群控制平面节点（Master 节点）下载社区版的对应离线包并解压，或者从[下载中心](#)下载离线包并解压。

```
```bash
```

#### 假定版本 VERSION=0.5.0

```
export VERSION=v0.5.0 wget https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/offline-community-$VERSION-
amd64.tar tar -xvf offline-community-$VERSION-amd64.tar ```
```

2. 准备镜像，需要将离线包的镜像推送到镜像仓库。

- 下载镜像导入脚本。

```
bash
wget https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/offline_image_handler.sh
```

为 `offline_image_handler.sh` 添加可执行权限：

```
bash
chmod +x offline_image_handler.sh
```

- 执行脚本推送镜像到镜像仓库中。请注意目前存在已知问题，需要执行脚本之前需要在 Harbor 中创建 `docker.m.daocloud.io`、`elastic.m.daocloud.io`、`gcr.m.daocloud.io`、`ghcr.io`、`ghcr.m.daocloud.io`、`k8s-gcr.m.daocloud.io`、`quay.io`、`quay.m.daocloud.io`、`release.daocloud.io` 仓库。

```
```bash
```

指定镜像仓库地址

```
export REGISTRY_ADDR=registry.daocloud.io:30080
```

指定镜像仓库用户名和密码

```
export REGISTRY_USER=admin export REGISTRY_PASS=Harbor12345
```

指定离线包解压目录

```
export OFFLINE_DIR=$(pwd)/offline
```

执行脚本导入镜像

```
./offline_image_handler.sh import ```
```

```
!!! note
```

- 若导入镜像的过程出现失败，则失败会被跳过且脚本将继续执行。
- 失败镜像信息将被记录在脚本同级目录 `import_image_failed.list` 文件中，便于定位。
- 如果 `docker pull` 镜像时报错：`http: server gave HTTP response to HTTPS client`，请启用 `Insecure Registry`。

- 在集群的每个节点上运行 `vim /etc/docker/daemon.json` 命令以编辑 `daemon.json` 文件，输入以下内容并保存更改。（如果需要启用 `Insecure Registry` 则执行此步骤）

```
json title="daemon.json"
{
  "insecure-registries" : ["172.30.120.180:80"]
}
```

```
!!! note
```

请确保将 ``172.30.120.180:80`` 替换为您自己的 Harbor 仓库地址。
对于 Linux，`daemon.json` 文件的路径为 ``/etc/docker/daemon.json``。

- 运行以下命令重启 Docker。（如果需要启用 `Insecure Registry` 则执行此步骤）

```
bash
sudo systemctl daemon-reload
sudo systemctl restart docker
```

3. 在 k8s 集群控制平面节点（Master 节点）下载 `dce5-installer` 二进制文件。

```
```shell
```

## 假定 VERSION 为 v0.5.0

```
export VERSION=v0.5.0 curl -Lo ./dce5-installer https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5-installer-$VERSION ``
```

### 4. 设置集群配置文件 clusterConfig.yaml

- 如果是非公有云环境（虚拟机、物理机），请启用负载均衡（metallb），以规避 NodePort 因节点 IP 变动造成的不稳定。请仔细规划您的网络，设置 2 个必要的 VIP，配置文件范例如下：

```
yaml title="clusterConfig.yaml"
apiVersion: provision.daocloud.io/v1alpha2
kind: ClusterConfig
spec:
 loadBalancer:
 type: metallb
 istioGatewayVip: 10.6.229.10/32 # (1)
 insightVip: 10.6.229.11/32 # (2)
 registry: # (3)
 type: external
 externalRegistry: registry.daocloud.io:30080 # (4)
```

- 这是 Istio gateway 的 VIP，也会是 DCE 5.0 的控制台的浏览器访问 IP
- 这是 Global 集群的 Insight-Server 采集所有子集群的监控指标的网络路径所用的 VIP
- 如果不指定外部仓库则不需要定义
- 这是提前准备的、已有的镜像仓库的地址

- 如果是公有云环境，并通过预先准备好的 Cloud Controller Manager 的机制提供了公有云的 k8s 负载均衡能力，配置文件范例如下：

```
yaml title="clusterConfig.yaml"
apiVersion: provision.daocloud.io/v1alpha2
kind: ClusterConfig
spec:
 loadBalancer:
 type: cloudLB
 registry: # (1)
 type: external
 externalRegistry: registry.daocloud.io:30080 # (2)
```

- 如果不指定外部仓库则不需要定义
  - 这是提前准备的、已有的镜像仓库的地址
- 如果使用 NodePort 暴露控制台（仅推荐 PoC 使用），配置文件范例如下：

```
yaml title="clusterConfig.yaml"
apiVersion: provision.daocloud.io/v1alpha2
kind: ClusterConfig
spec:
 loadBalancer:
 type: NodePort
 registry: # (1)
 type: external
 externalRegistry: registry.daocloud.io:30080 # (2)
```

- 如果不指定外部仓库则不需要定义
- 这是提前准备的、已有的镜像仓库的地址

### 5. 安装 DCE 5.0。

```
shell
./dce5-installer install-app -c clusterConfig.yaml -p offline
```

!!! note

- 参数 `-p` 指定解压离线包的 `offline` 目录。
- 有关 `clusterConfig.yaml` 文件设置, 请参考[在线安装第 2 步](online.md#\_2)。
- `-z` 最小化安装
- `-c` 指定集群配置文件。使用 `NodePort` 暴露控制台时不需要指定 `-c`。
- `-d` 开启 `debug` 模式

6. 安装完成后, 命令行会提示安装成功。恭喜您! 现在可以通过屏幕提示的 URL 使用默认的账户和密码 (admin/changeme) 探索全新的 DCE 5.0 啦!

```
5:18PM: TEST SUITE: None
5:18PM: NOTES:
5:18PM: _____
5:18PM: | _ _ | () | _ | _ | / _ _ | _ | _ | _ |
5:18PM: | | _ _ _ _ _ | | | | | | | | | | | | | |
5:18PM: | | | ' _ \ | / _ ' | _ \ | | | | / _ \ | _ \ |
5:18PM: | | | | \ \ \ | | | | | | | | | | | | | | |
5:18PM: | _ _ | | _ \ | _ \ | | | | \ _ \ | _ \ | _ \ |
5:18PM: | _ _ |
5:18PM: | _ \
5:18PM:
5:18PM: 1. Get the Insight UI URL by running these commands:
5:18PM: export POD_NAME=$(kubectl get pods --namespace insight-system -l "app.kubernetes.io/name=ui,app.kubernetes.io/instance=insight" -o jsonpath="{.items[0].metadata.name}")
5:18PM: export CONTAINER_PORT=$(kubectl get pod --namespace insight-system $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
5:18PM: kubectl --namespace insight-system port-forward $POD_NAME 8080:$CONTAINER_PORT
5:18PM: echo "Visit http://127.0.0.1:8080 to use your application"
5:18PM: Install IPavo Dashboard v0.4.1 🍀
5:18PM: "ipavo" already exists with the same configuration, skipping
5:18PM: Hang tight while we grab the latest from your chart repositories...
5:18PM: ...Successfully got an update from the "ipavo" chart repository
5:18PM: Update Complete. #Happy Helming!*
5:18PM: Release "ipavo" does not exist. Installing it now.
5:18PM: NAME: ipavo
5:18PM: LAST DEPLOYED: Mon Sep 19 17:18:37 2022
5:18PM: NAMESPACE: ipavo-system
5:18PM: STATUS: deployed
5:18PM: REVISION: 1
5:18PM: NOTES:
5:18PM: 1. Get the application URL by running these commands:
5:18PM: Install Insight Agent v0.9.4
5:18PM: "insight" already exists with the same configuration, skipping
5:18PM: Hang tight while we grab the latest from your chart repositories...
5:18PM: ...Successfully got an update from the "insight" chart repository
5:18PM: Update Complete. #Happy Helming!*
5:18PM: Release "insight-agent" does not exist. Installing it now.
5:19PM: NAME: insight-agent
5:19PM: LAST DEPLOYED: Mon Sep 19 17:19:01 2022
5:19PM: NAMESPACE: insight-system
5:19PM: STATUS: deployed
5:19PM: REVISION: 1
5:19PM: -----
5:19PM: [!] Use your Web Browser to access DaoCloud Enterprise 5th Portal at : http://172.30.47.104:31227
5:19PM: -----
5:19PM: -----
5:19PM: All set. Enjoy your journey to cloud native with DaoCloud Enterprise 5th !
5:19PM: -----
[root@localhost ~]# _
```

!!! success

请记录好提示的 URL, 方便下次访问。

7. 另外, 安装 DCE 5.0 成功之后, 您需要正版授权后使用, 请参考[申请社区免费体验](#)。

## 使用 Kubernetes 集群在线安装社区版

本页简要说明 DCE 5.0 社区版的在线安装步骤。

!!! note

- 点击[在线安装社区版](../../videos/install.md)可观看视频演示。
- 如果需要离线安装，请查阅[离线安装步骤](offline.md)。

### 准备工作

- 准备一个 K8s 集群，集群配置请参考文档 [集群资源规划](#)。

!!! note

- 存储：需要提前准备好 StorageClass，并设置为默认 SC
- 确保集群已安装 CoreDNS
- 如果是单节点集群，请确保您已移除该节点的污点

- [安装依赖项](#)。

如果集群中已安装所有依赖项，请确保依赖项版本符合要求：

- helm  $\geq$  3.11.1
- skopeo  $\geq$  1.11.1
- kubect1  $\geq$  1.25.6
- yq  $\geq$  4.31.1

### 下载和安装

1. 在 K8s 集群控制平面节点（Master 节点）下载 dce5-installer 二进制文件（也可以[通过浏览器下载](#)）。

```
```shell
```

假定 VERSION 为 v0.5.0

```
export VERSION=v0.5.0 curl -Lo ./dce5-installer https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/dce5-installer-$VERSION chmod +x ./dce5-installer ```
```


2. 设置配置文件 clusterConfig.yaml

- 如果使用 NodePort 暴露控制台（仅推荐 PoC 使用），直接执行下一步。
- 如果是非公有云环境（虚拟机、物理机），请启用负载均衡（metallb），以避免 NodePort 因节点 IP 变动造成的不稳定。请仔细规划您的网络，设置 2 个必要的 VIP，配置文件范例如下：

```
yaml title="clusterConfig.yaml"
apiVersion: provision.daocloud.io/v1alpha2
kind: ClusterConfig
spec:
  loadBalancer:
    type: metallb
    istioGatewayVip: 10.6.229.10/32 # (1)
    insightVip: 10.6.229.11/32 # (2)
```

- 这是 Istio gateway 的 VIP，也是 DCE 5.0 控制台的浏览器访问 IP
- 全局服务集群的 Insight-Server 采集所有子集群监控指标的网络路径所用的 VIP

- 如果是公有云环境，并通过预先准备好的 Cloud Controller Manager 的机制提供了公有云的 K8s 负载均衡能力，配置文件范例如下：

```
yaml title="clusterConfig.yaml"
apiVersion: provision.daocloud.io/v1alpha2
kind: ClusterConfig
spec:
  loadBalancer:
    type: cloudLB
```

3. 安装 DCE 5.0。

```
shell
./dce5-installer install-app -c clusterConfig.yaml -z
```

!!! note

- `-z`` 最小化安装
- `-c`` 指定集群配置文件。使用 NodePort 暴露控制台时不需要指定 `-c``。
- `-d`` 开启 debug 模式

- 安装完成后，命令行会提示安装成功。恭喜您！现在可以通过屏幕提示的 URL 使用 默认的账户和密码（admin/changeme）探索全新的 DCE 5.0 啦！

[illegible]

!!! success

请记录好提示的 URL，方便下次访问。

5. 另外，安装 DCE 5.0 成功之后，您需要正版授权后使用，请参考[申请社区免费体验](#)。

7.3 Kind

使用 kind 集群在线安装 DCE 5.0 社区版

本页说明如何使用 kind 集群在线安装 DCE 5.0 社区版。

!!! note

点击[在线安装社区版](../../videos/install.md#3)可观看视频演示。

准备工作

- 准备一台机器，机器资源建议：CPU > 10 核、内存 > 12 GB、磁盘空间 > 100 GB。
- 确保节点上已经安装 Docker、Kubect1。

安装 kind 集群

1. 下载 kind 的二进制文件包。

```
shell
curl -Lo ./kind https://qiniu-download-public.daocloud.io/kind/v0.17.0/kind-linux-amd64
```

为 kind 添加可执行权限：

```
bash
chmod +x ./kind
```

2. 在环境变量中加入 kind。

```
bash
mv ./kind /usr/bin/kind
```

3. 查看 kind 版本。

```
shell
kind version
```

预期输出如下：

```
console
kind v0.17.0 go1.19.2 linux/amd64
```

4. 设置 kind_cluster.yaml 配置文件。

注意，暴露集群内的 32000 端口到 kind 对外的 8888 端口（可自行修改），配置文件示例如下：

```
yaml title="kind_cluster.yaml"
apiVersion: kind.x-k8s.io/v1alpha4
kind: Cluster
nodes:
- role: control-plane
  extraPortMappings:
  - containerPort: 32088
    hostPort: 8888
```

5. 创建一个名为 fire-kind-cluster 的 v1.21.1 示例集群。

```
shell
kind create cluster --image release.daocloud.io/kpanda/kindest-node:v1.21.1 --name=fire-kind-cluster --config=kind_cluster.yaml
```

预期输出如下：

```
console
Creating cluster "fire-kind-cluster" ...
```

```

✓ Ensuring node image (release.daocloud.io/kpanda/kindest-node:v1.21.1) [?]
✓ Preparing nodes 📦
✓ Writing configuration 📄
✓ Starting control-plane 🚦
✓ Installing CNI 🐳
✓ Installing StorageClass 🗄️

Set kubectl context to "kind-fire-kind-cluster"

```

- 查看新创建的集群。

```

shell
kind get clusters

```

预期输出如下：

```

console
fire-kind-cluster

```

安装 DCE 5.0 社区版

- 安装依赖项。

!!! note

如果集群中已安装所有依赖项，请确保依赖项版本符合要求：

```

- helm ≥ 3.11.1
- skopeo ≥ 1.11.1
- kubectl ≥ 1.25.6
- yq ≥ 4.31.1

```

- 在 kind 主机下载 dce5-installer 二进制文件（也可以通过[浏览器下载](#)）。

```

```shell

```

## 假定 VERSION 为 v0.5.0

```

export VERSION=v0.5.0 curl -Lo ./dce5-installer https://proxy-qiniu-download-public.daocloud.io/DaoCloud_Enterprise/dce5/dce5-installer-$VERSION ```

```

- 获取 kind 所在主机的 IP，例如 10.6.3.1，执行以下命令开始安装。

```

shell
chmod +x ./dce5-installer
./dce5-installer install-app -z -k 10.6.3.1:8888

```

!!! note

kind 集群仅支持 NodePort 模式。

- 安装完成后，命令行会提示安装成功。恭喜您！现在可以通过屏幕提示的 URL 使用 默认的账户和密码（admin/changeme）探索全新的 DCE 5.0 啦！

[illegible]

!!! success

- 请记录好提示的 URL，方便下次访问。
- 成功安装 DCE 5.0 社区版后，请[申请社区免费体验](../../dce/license0.md)。
- 如果安装过程中遇到什么问题，欢迎扫描二维码，与开发者畅快交流：

! [社区版交流群] (.../.../.../images/assist.png)