



APUSIC  
固若长城  
睿比世界

# 安装手册

金蝶Apusic研发运营平台v1.0

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

## 版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

## 免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

## 商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。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。

# 目录

## 1 产品安装手册

- 1.1 产品介绍
- 1.2 范围和读者
- 1.3 约定与术语
- 1.4 安装环境要求
- 1.5 配置要求
- 1.6 推荐配置
- 1.7 产品安装说明
  - 1.7.1 docker-compose模式安装
    - 1.7.1.1 产品解压
    - 1.7.1.2 安装依赖工具
    - 1.7.1.3 生成特征码
    - 1.7.1.4 导入镜像
    - 1.7.1.5 安装K8S组件
      - 1.7.1.5.1 安装kubeverla组件
      - 1.7.1.5.2 安装velaux组件
      - 1.7.1.5.3 安装fluxcd组件
      - 1.7.1.5.4 安装自定义扩展
      - 1.7.1.5.5 卸载K8S组件
    - 1.7.1.6 安装ADOP
      - 1.7.1.6.1 配置初始化
      - 1.7.1.6.2 系统启动
      - 1.7.1.6.3 停止服务
    - 1.7.1.7 登录ADOP
- 1.8 附录
  - 1.8.1 常见问题

# 1 产品安装手册

## 1.1 产品介绍

金蝶Apusic研发运营平台（Apusic DevOps Platform，简称：ADOP）是金蝶天燕云计算股份有限公司经过多年经验积累，维护实践、自主研发和技术创新，包含着软件生命周期中计划-代码编写-构建-测试-发布-部署-运行-监控的全流程管理，借助 DevOps 体系持续改进提升组织流程和工程实践的流动性、反馈效率，并建立学习型组织。逐步从传统管理模式向 IT 敏捷转变，最终达到业务敏捷的目标。

## 1.2 范围和读者

本手册介绍ADOPV1.0产品的使用详细说明，适用于ADO产品的用户，ADO产品技术顾问，ADO产品维护人员，以及希望学习了解ADO产品的相关人员。

## 1.3 约定与术语

一些约定的缩略词诠释

ADOP 金蝶天燕云平台-DevOps (Apusic DevOps)

## 1.4 安装环境要求

## 1.5 配置要求

安装ADOPV1.0产品的最低配置要求见下表

资源环境	要求
操作系统	Linux Red Hat 5.2或以上(及其他Kernel 2.25或以上linux版本)
CPU	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz或以上
内存	8G或以上
硬盘	可用空间128G或以上
浏览器	FireFox 21及以上、Chrome 23及以上、IE 10及以上

docker-CE	20.10.12或以上
docker-compose	1.28.5或以上

## 1.6 推荐配置

安装ADOPV1.0产品的推荐的配置见下表：

资源环境	要求
操作系统	Linux Red Hat 5.2或以上(及其他Kernel 2.25或以上linux版本)
CPU	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz或以上
内存	16G或以上
硬盘	可用空间200G或以上
浏览器	FireFox 21及以上、Chrome 60及以上
docker-CE	20.10.12或以上
docker-compose	1.28.5或以上

## 1.7 产品安装说明

### 1.7.1 docker-compose模式安装

ADOPV1.0支持docker-compose安装，安装前请在安装服务器上安装对应版本的docker

#### 1.7.1.1 产品解压

- 下载对应版本的adop-vX.X.0\_release.tar.gz安装包，并放置到部署服务器的/opt目录下。
- 解压产品安装包,这里以1.0.0为例

```
tar zxvf adop-v1.0.0_release.tar.gz
cd adop
```

为避免安装包过大，也会采用分包的方式进行发送，产品包可能为多个

```
adop-v1.0.0_release.tar.gz.aa
adop-v1.0.0_release.tar.gz.ab
```

此时应将所有包拷贝到/opt目录下，使用如下命令进行解压

```
$ cat adop-v1.0.0_release.tar.gz.* | tar -zxvf -
```

### 1.7.1.2 安装依赖工具

```
# 设置可执行权限
$ chmod +x tools/*
# 加入$PATH
$ cp -r tools/* /usr/sbin
```

### 1.7.1.3 生成特征码

特征码用于申请license，必须输入所在服务器绑定网卡的IP，若输入错误，则生成的特征码为空。根据特征获取license后，将license文件放入/opt/adop/adop/license目录中，即可

```
$ license gc --ip=xxx.xxx.xxx.xxx

2023/12/13 11:02:54 输入的IP地址为 xxx.xxx.xxx.xxx
2023/12/13 11:02:54 特征码为 SZTY1996305462
```

### 1.7.1.4 导入镜像

需要在安装服务器和k8s的每个node上都需执行此步骤，以便将镜像分发到对应的服务器上

#### 1. 导入kubeverla相关镜像

```
$ cd /opt/adop
$ docker load -i images/kubevela/vela-1.5.0.gz

Loaded image: harbor.apusic.com/oamdev/vela-cli:v1.6.1
Loaded image: harbor.apusic.com/oamdev/vela-core:v1.5.0
```

```
Loaded image: harbor.apusic.com/oamdev/cluster-gateway:v1.4.0
Loaded image: harbor.apusic.com/oamdev/hello-world:v1
Loaded image: harbor.apusic.com/oamdev/alpine-k8s:1.18.2
Loaded image: harbor.apusic.com/oamdev/kube-webhook-
certgen:v2.4.1
Loaded image: harbor.apusic.com/oamdev/vela-cli:v1.5.0
```

## 2. 导入velaux相关镜像

```
$ cd /opt/adop
$ docker load -i images/velaux/velaux-1.5.0.gz

Loaded image: harbor.apusic.com/oamdev/velaux:v1.5.0
Loaded image: harbor.apusic.com/oamdev/vela-apiserver:v1.5.0
```

## 3. 导入fluxcd相关镜像

```
$ cd /opt/adop
$ docker load -i images/fluxcd/fluxcd.gz

Loaded image: harbor.apusic.com/fluxcd/helm-controller:v0.21.0
Loaded image: harbor.apusic.com/fluxcd/image-automation-
controller:v0.22.1
Loaded image: harbor.apusic.com/fluxcd/image-reflector-
controller:v0.18.0
Loaded image: harbor.apusic.com/fluxcd/kustomize-
controller:v0.25.0
Loaded image: harbor.apusic.com/fluxcd/notification-
controller:v0.23.5
Loaded image: harbor.apusic.com/fluxcd/source-
controller:v0.24.4
```

## 4. 导入adop相关镜像

```

$ cd /opt/adop
$ docker load -i images/adop/adop.gz

Loaded image: harbor.apusic.com/adop/adop-server:develop-amd64
Loaded image: harbor.apusic.com/adop/adop-nexus-proxy:develop-amd64
Loaded image: harbor.apusic.com/apusic/adop-nginx:1.20-amd64
Loaded image: harbor.apusic.com/apusic/adop-nexus3:3.34.1
Loaded image: harbor.apusic.com/kingdee/postgres:13.3-amd64
...

```

### 1.7.1.5 安装K8S组件

K8S部件用于环境管理与应用部署，如仅使用流水线功能，则可跳过该步骤。

该服务器应提前配置好K8S客户端，并能够使用kubectl访问K8S集群。

```

# 使用如下命令进行验证
$ kubectl version
$ kubectl get pods

```

#### 1.7.1.5.1 安装kubeverla组件

- 使用helm进行安装

需要注意的是，因为我们使用了--wait 参数，因此，执行命令后会卡在当前命令行，直到vela安装完成

```

$ helm install --create-namespace -n vela-system kubeverla -f
./kubeverla/vela-core/apusic-values.yaml kubeverla/vela-core --wait
NAME: kubeverla
LAST DEPLOYED: Fri Dec 8 16:54:40 2023
NAMESPACE: vela-system
STATUS: deployed
REVISION: 1
NOTES:
Welcome to use the KubeVela! Enjoy your shipping application
journey!

```

```

      ,
      //,
      ///
      ./ //***
      ,/// //****
      .///// //*****
      //**** //*****
      //**** //*****
      ,//**** //*****
      ,//**** //*****.
      .//**** //*****
      //**** //*****.
      *//**** //*****
#@@@@@@@@@@@@@*   ., , ***/ //*****
/@@@@@@@@@@@@@#
 *@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@&
 .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.

```

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
.&@@@*   *@@@&   ,@@@&.

```

```

_ _ _ _ _
| / / _ _ | | _ _ _ \ \ / / _ _ | | _ _ _
| ' / | | | | | ' _ \ / _ \ \ \ / / _ \ | | / _ \ | | | | |
| . \ | | _ | | | _ ) | | _ / \ v / | _ / | | ( | |
| _ | \ \ \ _ , _ | | _ . _ / \ _ | \ / \ _ | | | \ _ , _ |

```

You can refer to <https://kubvela.io> for more details.

另起一个shell窗口，检查kubvela是否安装成功，看到两个pod都已经ready,则表示kubvela安装结束

```

$ kubectl get pod -n vela-system
NAME                                READY   STATUS    RESTARTS
AGE

```

```
kubevela-cluster-gateway-89f9fb4f-tnd2s    1/1    Running    0
3m46s
kubevela-vela-core-6c487c79d6-dfdv6      1/1    Running    0
3m46s
```

### 1.7.1.5.2 安装velaux组件

```
$ kubectl apply -f kubevela/velaux/velaux-manifest.yaml

application.core.oam.dev/addon-velaux created
configmap/trait-uischema-storage created
configmap/workflowstep-uischema-deploy created
configmap/policy-uischema-override created
configmap/workflowstep-uischema-notification created
configmap/trait-uischema-scaler created
configmap/workflowstep-uischema-webhook created
configmap/addon-uischema-velaux created
configmap/component-uischema-webservice created
configmap/component-uischema-task created
configmap/component-uischema-k8s-objects created
configmap/workflowstep-uischema-deploy2env created
configmap/resource-view created
configmap/service-view created
configmap/component-service-view created
configmap/application-resource-detail-view created
configmap/cloud-resource-view created
configmap/cloud-resource-secret-view created
configmap/collect-logs created
configmap/component-pod-view created
configmap/pod-view created
```

查看velaux 是否安装成功, 看到apiserver-f55c9cf66-mwqm7 和velaux-75cffd6c5b-ls4p6 已经ready,则表示velaux已经安装完成

```
$ kubectl get pod -n vela-system
```

NAME	READY	STATUS	RESTARTS
apiserver-f55c9cf66-mwqm7 72s	1/1	Running	0
kubevela-cluster-gateway-89f9fb4f-tnd2s 36m	1/1	Running	0
kubevela-vela-core-6c487c79d6-dfdv6 36m	1/1	Running	0
velaux-75cffd6c5b-ls4p6 70s	1/1	Running	0

查看velaux 访问地址，看到velaux暴露的nodeport端口为：32601,因此velaux的访问地址为：<http://:32601>

```
$ kubectl get svc velaux -n vela-system
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
velaux 8m41s	NodePort	10.68.152.86	<none>	80:32601/TCP	

velaux的默认用户名/密码为: admin/VelaUX12345

### 1.7.1.5.3 安装fluxcd组件

```
$ helm install --create-namespace -n flux-system fluxcd  
kubevela/fluxcd
```

```
NAME: fluxcd  
LAST DEPLOYED: Fri Dec 8 17:51:04 2023  
NAMESPACE: flux-system  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None
```

查看fluxcd是否安装完成,看到所有的pod都已经ready, 表示fluxcd安装完成

```
$ kubectl get pod -n flux-system
```

NAME	READY	STATUS	RESTARTS	AGE
helm-controller-79684958c-2ztmj	1/1	Running	0	68s
image-automation-controller-5fdbb84d88-6q79d	1/1	Running	0	68s
image-reflector-controller-66548d4589-xp79g	1/1	Running	0	68s
kustomize-controller-98758d45f-nz6f8	1/1	Running	0	68s
notification-controller-869969dbd6-s49z8	1/1	Running	0	68s
source-controller-7d77bfdc9b-mkm7k	1/1	Running	0	68s

#### 1.7.1.5.4 安装自定义扩展

加载kubeverla自定义扩展traits(安装完成后, 请务必加载下面的几个traits,应用部署时需要用到)

```
## apply init-container traits
```

```
$ docker run --rm -it -v $(pwd)/kubeverla/traits:/root/traits/ -v
 ~/.kube:/root/.kube harbor.apusic.com/oamdev/vela-cli:v1.5.0 def
 apply /root/traits/init-container.cue
 TraitDefinition init-container in namespace vela-system updated.
```

```
## apply gateway-adop traits
```

```
$ docker run --rm -it -v $(pwd)/kubeverla/traits:/root/traits/ -v
 ~/.kube:/root/.kube harbor.apusic.com/oamdev/vela-cli:v1.5.0 def
 apply /root/traits/gateway-adop.cue
 TraitDefinition gateway-adop created in namespace vela-system.
```

```
## apply javaservice-native traits
```

```
$ docker run --rm -it -v $(pwd)/kubevela/traits/:/root/traits/ -v
 ~/.kube:/root/.kube harbor.apusic.com/oamdev/vela-cli:v1.5.0 def
 apply /root/traits/javaservice-native-adop.cue
 ComponentDefinition javaservice-native created in namespace vela-
 system.

## apply storage-adop traits
$ docker run --rm -it -v $(pwd)/kubevela/traits/:/root/traits/ -v
 ~/.kube:/root/.kube harbor.apusic.com/oamdev/vela-cli:v1.5.0 def
 apply /root/traits/storage-adop.cue
 TraitDefinition storage-adop created in namespace vela-system.
```

### 1.7.1.5.5 卸载K8S组件

#### 1. 卸载fluxcd

```
## helm uninstall <INSTANCE_NAME> -n <NAMESPACE>
$ helm uninstall fluxcd -n flux-system

release "fluxcd" uninstalled
```

#### 2. 卸载velaux

```
$ docker run --rm -it -v ~/.kube:/root/.kube
 harbor.apusic.com/oamdev/vela-cli:v1.5.0 addon disable velaux
 {"level":"info","ts":1702024918.553614,"caller":"addon/versioned_regist
 the latest version v1.8.1 by default, you can use --version to
 speicify"}
 Successfully disable addon:velaux
```

#### 3. 卸载kubevela

##### 1. 先卸载vela的所有应用

#### 查看所有已安装的应用

```
$ kubectl get applications -A
```

使用 `kubectl delete applications <NAME> -n <NAMESPACE>` 删除应用，直到所有应用被成功卸载

例如: `kubectl delete applications appl -n default`

## 2. 卸载vela-core

```
$ helm uninstall kubevela -n vela-system
```

```
release "kubevela" uninstalled
```

## 3. 卸载CRD

```
$ kubectl get crd |grep oam | awk '{print $1}' | xargs kubectl delete crd
```

```
customresourcedefinition.apiextensions.k8s.io
```

```
"applicationrevisions.core.oam.dev" deleted
```

```
customresourcedefinition.apiextensions.k8s.io
```

```
"applications.core.oam.dev" deleted
```

```
customresourcedefinition.apiextensions.k8s.io
```

```
"componentdefinitions.core.oam.dev" deleted
```

```
customresourcedefinition.apiextensions.k8s.io
```

```
"definitionrevisions.core.oam.dev" deleted
```

```
customresourcedefinition.apiextensions.k8s.io
```

```
"envbindings.core.oam.dev" deleted
```

```
customresourcedefinition.apiextensions.k8s.io
```

```
"healthscopes.core.oam.dev" deleted
```

```
customresourcedefinition.apiextensions.k8s.io
```

```
"manualscalertraits.core.oam.dev" deleted
```

```
customresourcedefinition.apiextensions.k8s.io
```

```
"policies.core.oam.dev" deleted
```

```

customresourcedefinition.apiextensions.k8s.io
"policydefinitions.core.oam.dev" deleted
customresourcedefinition.apiextensions.k8s.io
"resourcetrackers.core.oam.dev" deleted
customresourcedefinition.apiextensions.k8s.io
"rollouts.standard.oam.dev" deleted
customresourcedefinition.apiextensions.k8s.io
"scopeddefinitions.core.oam.dev" deleted
customresourcedefinition.apiextensions.k8s.io
"traitdefinitions.core.oam.dev" deleted
customresourcedefinition.apiextensions.k8s.io
"workflows.core.oam.dev" deleted
customresourcedefinition.apiextensions.k8s.io
"workflowstepdefinitions.core.oam.dev" deleted
customresourcedefinition.apiextensions.k8s.io
"workloaddefinitions.core.oam.dev" deleted

```

## 1.7.1.6 安装ADOP

### 1.7.1.6.1 配置初始化

- 修改kubeconfig 分别修改 acmp/config/kubeconfig, adop-server/config/kubeconfig,修改其内容为当前k8s集群的config, k8s集群的config默认存放路径为: /root/.kube/config
- 修改各个服务的配置文件

#### # 进入对应目录

```
cd /opt/adop/adop/docker-compose
```

1. 修改 acmp/config/config-docker-compose.yaml,将其中的ip形式的: host的值改为当前安装服务器的IP
2. 修改 adop-server/config/config-docker-compose.yaml,将其中的ip形式的: host的值改为当前安装服务器的IP
  1. 修改: kubevela.host 的值为安装的kubevela的地址,kubevela.userName和kubevela.password的值为对应的用户密码

3. 修改 nexus-proxy/config/config-docker-compose.yaml,将其中的ip形式的: host的值改为当前安装服务器的IP
- 创建挂载目录并授权

```
$ mkdir -p /opt/adop/data
$ chmod 777 /opt/adop/data
$ mkdir -p /opt/adop/data/jenkins_home
$ chmod 777 /opt/adop/data/jenkins_home
$ mkdir -p /opt/adop/data/jenkins_slave_home/workspace
$ chmod 777 /opt/adop/data/jenkins_slave_home/workspace
$ mkdir -p /opt/adop/data/jenkins_slave_home1/workspace
$ chmod 777 /opt/adop/data/jenkins_slave_home1/workspace
$ mkdir -p /opt/adop/data/jenkins_slave_home2/workspace
$ chmod 777 /opt/adop/data/jenkins_slave_home2/workspace
```

#### 1.7.1.6.2 系统启动

- 启动服务 首次启动服务后, 需要进行数据初始化

```
$ docker-compose up -d
```

- 初始化数据 系统需要一定时间进行准备, 请确认系统启动完成后, 再进行初始化。

```
cd /opt/adop/adop/docker-compose/nexus/script
chmod +x initNexus.sh
./initNexus.sh
```

- 重启服务

```
cd /opt/adop/adop/docker-compose
$ docker-compose down
$ docker-compose up -d
# 查看系统运行状态
$ docker-compose ps
```

### 1.7.1.6.3 停止服务

服务停止后，对应的数据目录应保留。如确认不再使用后，可以删除/opt/adop目录，并清理相关镜像即可。

```
cd /opt/adop/adop/docker-compose
$ docker-compose up -d
```

### 1.7.1.7 登录ADOP

- 登录系统并测试 平台侧系统管理员：用户名：admin 密码：123456

租户侧测试用户：租户：test 用户名：bob 密码：123456

提示：系统管理用户登录后，需要及时修改密码，租户测试用户需要在系统启用后删除或关闭该租户。

#### 登录web控制台

登录访问web控制台平台侧<http://serverIP/#/login>

输入初始化账号：admin 密码：123456

登录访问web控制台租户侧<http://serverIP/#/tenant/login>

输入初始化租户名：test 用户名：bob 密码：123456



## 1.8 附录

## 1.8.1 常见问题

- 问：构建时访问不到外部的代码仓库怎么办？

答：ADOP平台使用主机网络，可以检查主机是否可以访问，若主机不能访问时，可以在/etc/hosts文件中增加对应的域名映射

- 问：验证码无法显示怎么办？

答：造成验证码无法显示的与后端系统无法正常访问有关，可以通过重启网关docker-compose restart nginx进行重试，若依旧不能访问，则需要确认后端服务adop-server是否运行正常

全国统一服务热线  
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。

**Apusic**  
金蝶天燕

云计算国家标准制定企业  
金蝶集团旗下基础软件企业  
信息技术应用创新核心企业  
官网: [www.apusic.com](http://www.apusic.com)

