

网站建设 Confluence Helm 资产文档

一、资产基本介绍

● 资产简介

Atlassian Confluence（简称Confluence）是一个专业的wiki程序。它是一个知识管理的工具，通过它可以实现团队成员之间的协作和知识共享。

Confluence是一个专业的企业知识管理与协同软件，也可以用于构建企业wiki。使用简单，但它强大的编辑和站点管理特征能够帮助团队成员之间共享信息、文档协作、集体讨论，信息推送。

Confluence为团队提供一个协作环境。在这里，团队成员齐心协力，各擅其能，协同地编写文档和管理项目。从此打破不同团队、不同部门以及个人之间信息孤岛的僵局，Confluence真正实现了组织资源共享。

Confluence Helm 模版可以在 Kubernetes 平台上一键部署一个可扩展的Confluence系统，同时集成时速云公有云 PaaS 平台的运维功能，实现对 Confluence 系统的自动化运维

● 资产依赖

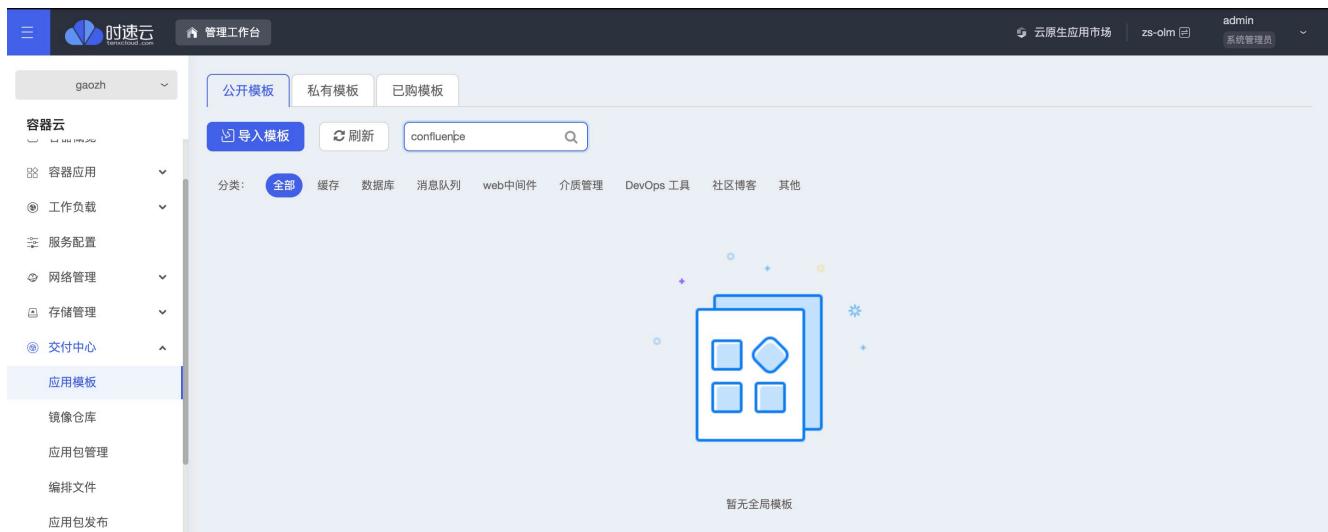
- Confluence Helm包：192.168.1.52:/root/helm/confluence-3.1.0.tgz Md5值:
fbc1e9a7fde3a3cdbaaa9400940ee0dd
- Confluence 镜像：dev-registry.tenxcloud.com/system_containers/confluence-server:7.12.0 镜像
ID: 6d6504b3648b

二、资产购买流程

- 在“云市场”中查询需要购买的资产
- 查看资产详情
- 购买资产：在资产详情中点击“购买”
 - 阅读《云原生应用市场用户协议》，同意后勾选“我已阅读并同意...”确认
 - 点击“支付”
- 查看已购资产：购买资产后会自动跳转到“已购资产”页面显示被购买的资产

三、资产部署指南

- 查询购买(或导入)Confluence模版
 - 在容器云--交付中心--应用模版 里搜索 " confluence "



- 如果没有搜到Confluence模版，也可以在容器云--交付中心--应用模版--私有模版里导入私有Confluence模版

The image consists of two screenshots of the TimeCloud management console illustrating the import process of the Confluence template.

Screenshot 1: Importing the Confluence Template

This screenshot shows the 'Import Template' dialog box. It features a central area with a placeholder icon and the text '点击或将文件拖拽到这里导入 支持扩展名: tar.gz, tgz'. Below this is a note: '① 应用模板配置文件请参考Helm规范, 文件小于 1024kb, 导入模板名称如果相同已有模板将会被覆盖。' At the bottom are '取消' and '开始导入' buttons.

Screenshot 2: Imported Template Confirmation

This screenshot shows the 'Private Template' list after the import. It includes a success message '模板导入成功' with a checkmark icon. The list now shows two items: 'confluence' (Wiki collaboration platform) and 'wordpress' (Web publishing platform for building bl...). Both entries have deployment and detail buttons below them.

● 安装 Confluence 模版

- 点击“部署”，选择版本开始安装

The screenshot shows the Tenx Cloud management console interface. On the left, a sidebar menu includes '容器云' (Container Cloud), '工作负载' (Workload), '服务配置' (Service Configuration), '网络管理' (Network Management), '存储管理' (Storage Management), '交付中心' (Delivery Center), and '应用模板' (Application Template). The '应用模板' item is currently selected. The main area displays a list of available templates under the '私有模板' (Private Template) tab. Two templates are shown: 'confluence' (Wiki collaboration platform) and 'wordpress' (Web publishing platform for building bl...). Both templates have a deployment button labeled '部署'. A modal window for the 'confluence' template is open, showing its version as '3.1.0'. The deployment status for 'confluence' is '1' minute ago.

The screenshot shows the deployment configuration page for the 'confluence' application template. The configuration fields are as follows:

- 模板名称: confluence
- 模板版本: 3.1.0
- * 集群: 默认集群
- 发布名称: confluence
- 描述: confluence demo
- 配置文件: value.yaml

A code editor window titled 'value.yaml (读写)' shows the YAML configuration file content:

```

value.yaml (读写)
1 - image:
2   repository: dev-registry.tenxcloud.com/system_containers/confluence-server
3   tag: 7.12.0
4   pullPolicy: Always
5   pullSecrets: {}
6   #   - myRegistryKeySecretName
7
8 ## String to partially override wiki.fullname template (will maintain the release

```

- 集群: 选择项目所授权的集群
- 发布名称: 指定模版应用的名称, 如: "confluence"
- 描述: 模版应用的描述信息, 如: "confluence demo"
- 配置文件:

image:

```

    repository: dev-registry.tenxcloud.com/system_containers/confluence-
server
    tag: 7.12.0
    pullPolicy: Always
    pullSecrets: {}
    # - myRegistryKeySecretName

## String to partially override wiki.fullname template (will maintain the
release name)
nameOverride: ""

## String to fully override wiki.fullname template
fullnameOverride: ""

## ref: https://kubernetes.io/docs/tasks/configure-pod-
container/configure-service-account/
serviceAccount:
    # Specifies whether a service account should be created
    create: false
    # Annotations to add to the service account
    annotations: {}
    # The name of the service account to use.
    # If not set and create is true, a name is generated using the fullname
    template
    name: ""

## ref: https://kubernetes.io/docs/reference/generated/kubernetes-
api/v1.17/#podsecuritycontext-v1-core
podSecurityContext:
    fsGroup: 2002

## Security context
## ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-
context/
securityContext: {}
    # capabilities:
    # drop:
    # - ALL
    # readOnlyRootFilesystem: true
    # runAsNonRoot: true
    # runAsUser: 1000

## Service/Networking
## ref: https://kubernetes.io/docs/concepts/services-networking/service/
service:
    ## For minikube, set this to NodePort, elsewhere use LoadBalancer
    type: ClusterIP
    ## Use serviceLoadBalancerIP to request a specific static IP, otherwise
    leave blank

```

```

##  

## Avoid removing the http connector, as the Synchrony proxy health  

check, still requires HTTP  

## HTTP Port, must be the same as ATL_TOMCAT_PORT (default: 8090)  

port: 8090  

## HTTPS Port, in case ATL_TOMCAT_SCHEME is set to 'https'  

httpsPort:  

loadBalancerIP:  

## Use nodePorts to requests some specific ports when usin NodePort  

## nodePorts:  

##   http: <to set explicitly, choose port between 30000-32767>  

##   https: <to set explicitly, choose port between 30000-32767>  

##  

nodePorts:  

  http:  

  https:  

## Configure the ingress resource that allows you to access the  

## Confluence installation. Set up the URL  

## ref: http://kubernetes.io/docs/user-guide/ingress/  

ingress:  

## Set to true to enable ingress record generation  

enabled: false  

annotations: {}  

# kubernetes.io/ingress.class: nginx  

# kubernetes.io/tls-acme: "true"  

hosts:  

  - host: confluence-server.local  

    paths: []  

tls: []  

# - secretName: confluence-server.local-tls  

#   hosts:  

#     - confluence-server.local  

## ref: https://kubernetes.io/docs/concepts/configuration/manage-compute-  

resources-container/  

resources:  

  requests:  

    cpu: 500m  

    memory: 4Gi  

  limits:  

    cpu: 4  

    memory: 8Gi  

## Replication (without ReplicaSet)  

## ref:  

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/  

replicaCount: 1

```

```

## Node labels for pod assignment
## ref: https://kubernetes.io/docs/user-guide/node-selection/
nodeSelector: {}

## Tolerations for pod assignment
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-
toleration/
tolerations: []

## Affinity for pod assignment
## ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-
node/#affinity-and-anti-affinity
affinity: {}

## Pod annotations
## ref: https://kubernetes.io/docs/concepts/concepts/overview/working-with-
objects/annotations/
podAnnotations: {}

## Persistent Volume Claim
## Confluence Home directory
## https://kubernetes.io/docs/concepts/storage/persistent-volumes/
persistence:
  enabled: true
  annotations: {}
  ## existingClaim needs the existing PVC name
  existingClaim: ""
  accessMode: ReadWriteOnce
  size: 1Gi

  ## If defined, storageClassName: <storageClass>
  ## If set to "-", storageClassName: "", which disables dynamic
provisioning
  ## If undefined (the default) or set to nil, no storageClassName spec is
  ## set, choosing the default provisioner. (gp2 on AWS, standard on
  ## GKE, AWS & OpenStack)
  ##
  storageClass: nfs-94

# Additional volume mounts
extraVolumeMounts: []
  ## Example: Mount CA file
  # - name: ca-cert
  #   subPath: ca_cert
  #   mountPath: /path/to/ca_cert

# Additional volumes
extraVolumes: []
  ## Example: Add secret volume

```

```

# - name: ca-cert
#   secret:
#     secretName: ca-cert
#     items:
#       - key: ca-cert
#         path: ca_cert

## Use an alternate scheduler, e.g. "stork".
## ref: https://kubernetes.io/docs/tasks/administer-cluster/configure-
multiple-schedulers/
schedulerName: ""

## Container Probes
## ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-
lifecycle/#container-probes
## ref: https://kubernetes.io/docs/tasks/configure-pod-
container/configure-liveness-readiness-probes/#configure-probes
## Depending what values we give, Confluence won't be reachable. In doubt,
leave it as it is.
readinessProbe: {}
# httpGet:
#   path: /status
#   port: http
# initialDelaySeconds: 300
# periodSeconds: 30
# failureThreshold: 6
# timeoutSeconds: 10

livenessProbe: {}
# httpGet:
#   path: /status
#   port: http
# initialDelaySeconds: 480
# periodSeconds: 30
# failureThreshold: 6
# timeoutSeconds: 10

## Environment Variables that will be injected in the ConfigMap
## Default values unless otherwise stated
envVars:
  ## Memory / Heap Size (JVM_MINIMUM_MEMORY) Mandatory, see @Notes above
  ## default: 1024m
  JVM_MINIMUM_MEMORY: 2048m
  ## Memory / Heap Size (JVM_MAXIMUM_MEMORY) Mandatory, see @Notes above
  ## default: 1024m
  JVM_MAXIMUM_MEMORY: 2048m
  #
  ## Tomcat and Reverse Proxy Settings
  ## Confluence running behind a reverse proxy server options

```

```

## Note - When ingress is enabled:
## These values are set automatically. Do not uncomment these proxy
settings.

# ATL_PROXY_NAME: ""
# ATL_PROXY_PORT: ""
# ATL_TOMCAT_PORT: 8090
# ATL_TOMCAT_SCHEME: http
# ATL_TOMCAT_SECURE: false
# ATL_TOMCAT_CONTEXTPATH: ""
#
## Tomcat/Catalina options
## ref: https://tomcat.apache.org/tomcat-7.0-doc/config/index.html
# ATL_TOMCAT_MGMT_PORT: 8000
# ATL_TOMCAT_MAXTHREADS: 100
# ATL_TOMCAT_MINSPARETHREADS: 10
# ATL_TOMCAT_CONNECTIONTIMEOUT: 20000
# ATL_TOMCAT_ENABLELOOKUPS: false
# ATL_TOMCAT_PROTOCOL: "HTTP/1.1"
# ATL_TOMCAT_ACCEPTCOUNT: 10
#
## Cookie age (Remember Me maximum time remain logged-in)
# ATL_AUTOLOGIN_COOKIE_AGE: 1209600
#
## Home directory. This may be on a mounted volume; if so it
## should be writable by the user confluence. See note below about UID
mappings.

# CONFLUENCE_HOME: ""
#
## Optional connection pool database settings
# ATL_DB_POOLMINSIZE: 20
# ATL_DB_POOLMAXSIZE: 100
# ATL_DB_TIMEOUT: 30
# ATL_DB_IDLETESTPERIOD: 100
# ATL_DB_MAXSTATEMENTS: 0
# ATL_DB_VALIDATE: false
# ATL_DB_ACQUIREINCREMENT: 1
# ATL_DB_VALIDATIONQUERY: "select 1"
## End of Environment Variables (envVars)

## JVM_SUPPORT_RECOMMENDED_ARGS
## Additional container environment variables
# extraEnv: "-XX:MaxMetaspaceSize=512m -XX:MaxDirectMemorySize=10m -
Dsynchrony.memory.max=0m"

```

- 基础配置说明:
 - * image.repository: 镜像地址, 指定具体的jira-software地址, 如: "dev-registry.tenxcloud.com/system_containers/confluence-server"
 - * image.tag: 镜像tag, 如: "7.12.0"
 - * resources.requests: 每个Pod 请求的 CPU、内存资源大小, 推荐使用 2C/4G 配置
 - * resources.limits: 每个Pod 请求的 CPU、内存资源大小, 推荐使用 4C/8G 配置, 如果资源充足可以适当设置大一些
 - * persistence.storageClass: 集群使用的存储类名称, 从“容器云--存储管理--存

储卷--创建存储卷--存储类下拉列表”中可以查看到可以使用的存储类

The screenshot shows the 'Create Storage' dialog. On the left, there's a sidebar with 'Container Cloud' navigation items: Container Application, Workload, Service Configuration, Network Management, Storage Management (selected), Storage Snapshot, Delivery Center, Traditional Application, and Stack. In the main area, there are fields for 'Storage Name' (请输入存储名称), 'Match Persistent Volume' (radio buttons for 'Dynamic Create Persistent Volume' and 'Match Existing Persistent Volume'), 'Storage Class' (dropdown menu showing 'my-openebs' and 'nfs-94'), 'Access Mode' (radio buttons for 'my-openebs' and 'nfs-94'), and 'Storage Size' (input field set to 1 Gi). At the bottom are 'Confirm' and 'Cancel' buttons.

* persistence.size: 存储大小，可根据存储的资源情况进行设置

- 点击“确定”: 自动跳转到 模版应用 菜单

The screenshot shows the 'Template Application' list page. The sidebar has 'Container Cloud' navigation items: Application, Service, Template Application (selected), Operator Application, Workload, Service Configuration, Network Management, and Storage Management. The main area displays a table of deployed applications: 'confluence' (status: 已部署, template version: confluence-3.1.0, application version: 1, created 9 minutes ago) and 'wordpress' (status: 已部署, template version: wordpress-11.1.5, application version: 1, created 30 minutes ago). There are buttons for '刷新' (Refresh), '删除' (Delete), and search. A success message '部署成功' (Deployment successful) is visible at the top right.

- 点击 模版应用 名称，进入 模版应用 详情

The screenshot shows two views of the 'Template Application Details' page for 'confluence'.

Top View:

- Resource Table:**

资源名称	类型	创建时间	操作
confluence	ConfigMap	10分钟前	查看Yaml
confluence	Service	10分钟前	查看Yaml
confluence	Deployment	10分钟前	查看Yaml
- History Versions Table:**

历史应用名称	状态	版本号	模板版本	创建时间	操作
confluence	已部署	1	confluence-3.1.0	11分钟前	当前版本

Bottom View:

- Resource Table:**

容器组名称	状态	镜像	访问地址	创建时间	操作
confluence-69869cccdcd-k2x5q	运行中 已重启 0 次	dev-registry.tenxcloud.com/syst...	172.31.54.240	15分钟前	终端

- 查看 实例运行 状态

The screenshot shows the 'Deployment Details' page for 'confluence'.

Deployment Status:

- 状态: 运行中 1/1
- 所属服务: confluence
- 创建时间: 2021-07-22 10:30:54
- 实例选择器: app.kubernetes.io/instance: confluence,app.kubernetes....
- 节点选择器: -

Container Group Table:

容器组	控制器配置	容器配置	事件	监控	实时日志	弹性伸缩	部署版本 (回滚)
confluence							

● 配置 Confluence 集群外访问

- 在“容器云--网络管理--应用路由”页面，点击“添加路由规则”

- 在“添加路由规则”页面，选择一个服务出口代理 confluence 服务

The screenshot shows the 'Add Routing Rule' interface. On the left sidebar, under '应用路由', there are several options like '工作负载', '服务配置', '网络管理', etc. The main form has fields for '规则名称' (rule name) set to 'confluence', '规则描述' (rule description) as 'confluence route', '选择出口' (select exit) as '2.63', '选择服务' (select service) as 'confluence', and '端口协议' (port protocol) as 'TCP'. Below this is a table for '服务端口协议' (service port protocol) and '代理端口协议' (proxy port protocol), showing '8090' and 'TCP' respectively. At the bottom are '取消' (Cancel) and '创建' (Create) buttons.

- 规则名称：这条路由规则的名称，如："confluence"
- 选择出口：选择一个平台的服务访问出口
- 选择服务：confluence 服务，如："confluence"
- 点击“创建”后，可以看到生成的路由规则

The screenshot shows the 'Service Proxy' page. The left sidebar has '应用路由' selected. The main area displays a table of rules. A green success message at the top right says '添加路由规则成功' (Routing rule added successfully). The table lists two rules: 'confluence' (rule name, description 'confluence route', service 'confluence', proxy IP '192.168.2.63', created 9 minutes ago) and 'wordpress' (rule name, description 'wordpress route', service 'wordpress', proxy IP '192.168.2.63', created 1 hour ago). Each row has '编辑' (Edit) and '删除' (Delete) buttons.

● 验证 Confluence 服务状态

- 在“容器云--容器应用--容器服务”列表中，找到被代理的 confluence 服务，点击“查看地址”，点击地址旁边的拷贝图标保存地址信息，用于后面访问验证。

The screenshot shows the 'Service' list page. The left sidebar has '服务' selected. The main table lists services: 'confluence' (status '全部运行'), 'wordpress' (status '全部运行'), and 'gaozh-mysql' (status '全部运行'). The 'confluence' row is expanded to show its details: '容器端口: 8090', '集群内: confluence.gaozh:8090', and '服务代理-TCP: 192.168.2.63:44792'. There are '编辑 Yaml' and '...' buttons for each service entry.

- 集群内：在 Kubernetes 集群内访问 confluence 服务，使用这个地址

- 服务代理-TCP：在 Kuberenetes 集群外访问 confluence 服务，使用这个地址
- 打开浏览器输入集群外访问地址，如"192.168.2.63:44792"

The screenshots show the initial setup steps for Confluence:

- Enter your Confluence license key**: The page displays the Server ID (B5PM-NMM0-Y37C-TD69) and a large license key. A link to get an evaluation license is available.
- Choose your deployment type**: The user selects "Standalone".
- Set up your database**: The user configures MySQL settings, including a connection string (jdbc:mysql://gaozh-mysql.gaozh.svc:3306/confluence?useSSL=false), username (root), and password. A successful database connection test is shown.

四、应用运维指南；（补充界面部署方式）

- 监控信息查看：在“容器云--工作负载--部署”，点击进入 Confluence 副本集

部署详情

confluence

状态: 运行中 1/1

所属服务: confluence

创建时间: 2021-07-22 15:48:11

标签: app.kubernetes.io/instance: confluence ...

节点选择器: --

容器组	控制器配置	容器配置	事件	监控	实时日志	弹性伸缩	部署版本 (回滚)	租赁信息
容器组: confluence... x	容器: 全部 x			导出监控数据				
自定义日期 1小时 6小时 24小时 7天 30天								

CPU

时间间隔: 1分钟

实例	CPU 使用率 (%)
100.000%	100.000%
80.000%	80.000%
60.000%	60.000%
40.000%	40.000%

- 日志信息查看: 在“容器云--工作负载--部署”，点击进入 Confluence 副本集

部署详情

confluence

状态: 运行中 1/1

所属服务: confluence

创建时间: 2021-07-22 15:48:11

标签: app.kubernetes.io/instance: confluence ...

容器组	控制器配置	容器配置	事件	监控	实时日志	弹性伸缩	部署版本 (回滚)	租赁信息
容器组: confluence-69869cccd... x	容器: 全部 x							
历史日志								

```

[confluence] [2021-07-22 15:49:23] WARNING: An illegal reflective access operation has occurred
[confluence] [2021-07-22 15:49:23] WARNING: Illegal reflective access by
org.apache.felix.framework.ext.ClassPathExtenderFactory$DefaultClassLoaderExtender (file:/opt/atlassian/confluence/confluence/WEB-INF/lib/
org.apache.felix.framework-5.6.12.jar) to method java.net.URLClassLoader.addURL(java.net.URL)
[confluence] [2021-07-22 15:49:23] WARNING: Please consider reporting this to the maintainers of
org.apache.felix.framework.ext.ClassPathExtenderFactory$DefaultClassLoaderExtender
[confluence] [2021-07-22 15:49:23] WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
[confluence] [2021-07-22 15:49:23] WARNING: All illegal access operations will be denied in a future release
[confluence] [2021-07-22 15:49:23] 22-Jul-2021 07:49:50.606 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler
["http-nio-8090"]
[confluence] [2021-07-22 15:49:51] 22-Jul-2021 07:49:51.033 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [70039]
milliseconds
  
```

- 事件信息查看: 在“容器云--工作负载--部署”，点击进入 Confluence 副本集

部署详情

confluence

状态: 运行中 1/1

所属服务: confluence

创建时间: 2021-07-22 15:48:11

标签: app.kubernetes.io/instance: confluence ...

容器组	控制器配置	容器配置	事件	监控	实时日志	弹性伸缩	部署版本 (回滚)	租赁信息
容器组: confluence... x								
刷新 滚动发布 ...								

事件	操作	描述	时间
Started	Started container confluence	27 分钟前	
Pulling	Pulling image "dev-registry.tenxcloud.com/system_containers/confluence-server:7.12.0"	27 分钟前	
Pulled	Successfully pulled image "dev-registry.tenxcloud.com/system_containers/confluence-server:7.12.0"	27 分钟前	
Created	Created container confluence	27 分钟前	
Scheduled	Successfully assigned gaozh/confluence-69869cccd-gfpc9 to press2.76	27 分钟前	

- 审计信息查看：在“安全和运维--平台运维--操作审计--审计记录”，选择“容器云/容器应用/模版应用”、相应租户、项目后点击“立即查询”

The screenshot shows the Bitnami Cloud Platform Management Workbench interface. The top navigation bar includes the Bitnami logo, a search bar, and user information (admin, system administrator). The left sidebar has a tree view with nodes like 大屏驾驶舱, 操作审计 (selected), 审计记录 (selected), 审计分析, 平台数据备份, 资源报表, 应用监控, 日志服务, 事件管理, 应用告警, and 监控面板. The main content area displays a table of audit logs. The table columns are: 时间 (Time), 持续时间 (Duration), 操作类型 (Operation Type), 对象及类型 (Object and Type), 项目 (Project), 集群名 (Cluster Name), 状态 (Status), 发起者 (Initiator), and 源 IP (Source IP). There are 27 entries listed, all showing successful deletions of template applications ('类型: 模板应用') from the 'gaozh' project ('对象: confluence...' or 'wordpress...').

时间	持续时间	操作类型	对象及类型	项目	集群名	状态	发起者	源 IP
2021-07-22 13:22:41	262 毫秒	删除	类型: 模板应用 对象: confluence...	gaozh	普通集群: 默认集群	成功	admin	192.168.4.179
2021-07-22 09:30:21	421 毫秒	删除	类型: 模板应用 对象: wordpress...	gaozh	普通集群: 默认集群	成功	admin	192.168.4.179
2021-07-21 19:32:38	195 毫秒	删除	类型: 模板应用 对象: wordpress	gaozh	普通集群: 默认集群	成功	admin	192.168.4.179
2021-07-21 19:11:01	310 毫秒	删除	类型: 模板应用 对象: wordpress	gaozh	普通集群: 默认集群	成功	admin	192.168.4.179
2021-07-21 18:31:31	309 毫秒	删除	类型: 模板应用 对象: wordpress	gaozh	普通集群: 默认集群	成功	admin	192.168.4.179
2021-07-21 18:09:35	291 毫秒	删除	类型: 模板应用 对象: wordpress	gaozh	普通集群: 默认集群	成功	admin	192.168.4.179