

Gitlab Helm资产

一、资产基本介绍

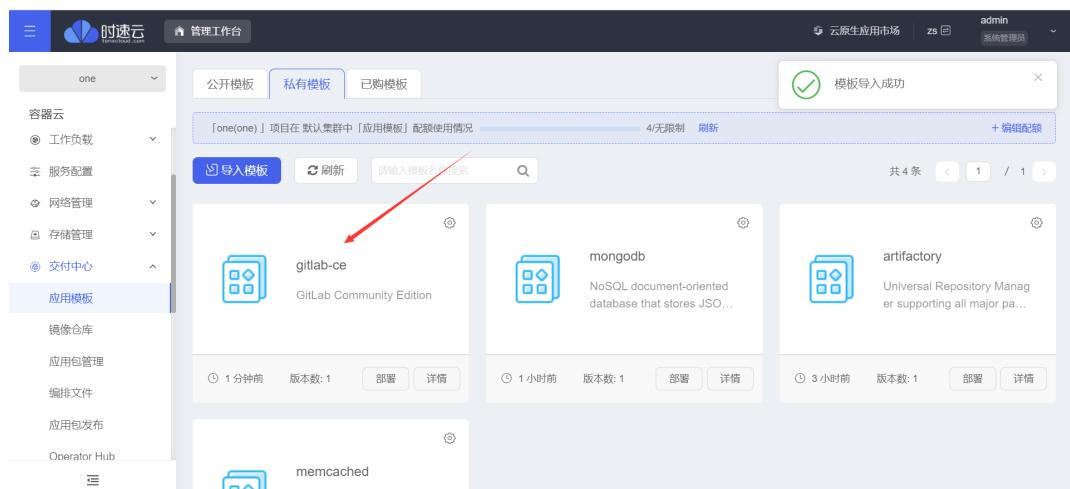
三、资产注册流程

- 导入私有模板资产

- 导入



- 成功



- 登录到平台管理后台
- 在左侧菜单栏找到 **资产管理**

- 点击添加资产

资产列表：支持您接入容器镜像、Helm模板、Operator、API等介质作为资产，可上架至市场中供购买者订购

点击添加资产，进入到创建资产流程

+ 添加资产 刷新 搜索

资产名称 资产编码 资产版本/API 交付类型 资产分类 资产版本状态 审批状态 上架时间 操作

共计 0 条 < 1 / 0 >

暂无数据

○ 添加资产基础信息

■ 交付类型选择有四种，选择Helm Chart应用

* 交付类型

容器镜像 Operator Helm Chart 应用 API

Helm Chart 应用指将 Helm 应用模版导入平台后在资产商店上架对外开放，用户订购后可创建模版应用

■ helm描述与icon信息

* 资产分类 基础应用 网站建设 企业应用 数据库中间件集群 API 服务 物联网 人工智能

* 资产名称 helm-gitlab

* 概述 helm gitlab

* 描述 helm gitlab

* 资产图标

资产图标尺寸为 96×96 px，格式支持 JPG 和 PNG

■ 设置规格，定价

定价支持按n年, n月计算

规格与定价

* 规格与定价	商品规格名称	价格	有效时长	操作
	1	1	30 日	<input type="button"/>

添加规格

■ 其他信息

其他信息

服务商网站	服务商网站 支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)
使用帮助	使用帮助 支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)
使用条款	使用条款 支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)
客户案例	客户案例 支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)

* 资产截图


+

建议图片 264 * 146px, 格式支持 JPG 和 PNG, 最多上传 2 张

平台支持范围

示例:	服务时间: 7*24 小时 服务热线: 400-400-4400 服务邮箱: xxxx@ss.cc 服务内容: 产品售后、技术咨询、商用配置等
-----	--

支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)

● 资产添加Helm

- 在资产管理列表位置，找到新增资产

资产列表：支持您接入容器镜像、Helm模版、Operator、API等介质作为资产，可上架至市场中供购买者订购

+ 添加资产	刷新	请输入资产名称搜索	计 5 条 / 1 >					
资产名称	资产编码	资产版本/API	交付类型	资产分类	资产版本状态	审批状态	上架时间	操作
helm-gitlab	ASID-aainYjBaGqWR	0 个	Helm	企业应用	-	-	-	<input type="button" value="创建版本"/> ...

- 点击进入详情，添加helm定义

helm-gitlab

交付类型: 企业应用	资产概述: helm gitlab	<input type="button" value="资产订单"/> ...
资产分类: Helm	资产创建时间: 2 分钟前	

资产版本 资产信息 规格与定价 用户

「未定义」(未上架)	* 资产版本: v1 * 版本描述: v1 * 模板所在项目: zs one (one) * 选择模板: gitlab-ce 0.2.2
------------	---

- 添加一个版本后，需要上架，可以在市场进行售卖

资产版本 资产信息 规格与定价 用户

+ 创建版本

v1 (未发布)

版本 v1 申请上架 删除版本

1 创建版本 1分钟前
2 上架审核 (1-2个工作日)
3 上架售卖
4 下架审核 (1-2个工作日)
5 已下架

* 资产版本 v1
* 版本描述 v1
* 模板所在项目 zs one (one)
* 选择模板 gitlab-ce 0.2.2

编辑

资产版本 资产信息 规格与定价 用户

+ 创建版本 申请上架 撤销申请

v1 (上架审核中, 未上架)

1 创建版本 2分钟前
2 上架审核 (1-2个工作日)
3 上架售卖
4 下架审核 (1-2个工作日)
5 已下架

* 资产版本 v1
* 版本描述 v1
* 模板所在项目 zs one (one)
* 选择模板 gitlab-ce 0.2.2

编辑

○ 审批

刷新 资产名称 点击升序 共 5 条 < 1 >

状态	申请人	资产名称	资产版本	资产类型	资产用户量	申请原因	申请时间	审批人	操作
● 上架待审核	admin	helm-gitlab	v1	Helm	0	-	2021-07-21 17:31:39	-	<button>通过</button> <button>拒绝</button>

审批通过

刷新 资产名称 点击升序 共 5 条 < 1 >

状态	申请人	资产名称	资产版本	资产类型	资产用户量	申请原因	申请时间	审批人	操作
● 上架通过	admin	helm-gitlab	v1	Helm	0	-	2021-07-21 17:31:39	admin	-

时速云 管理工作台 云原生应用市场 zs-street 普通应用

服务全局 - 不要求项目选择

我是用户 云市场 已购资产 订单管理 我是服务商 资产管理 出售管理

云原生应用市场

搜索框: gitlab

商品分类 全部 基础应用 网站建设 企业应用 数据库中间件集群 API 服务 物联网 人工智能

支付方式 全部 容器镜像 Operator Helm Chart 应用 API

价格 全部 免费 1 - 99 99 - 998 999 - 9998 9999 以上

默认 价格 更新时间 浏览量

为您找到 1 条结果

应用 已认证

GitLab

helm-gitlab 0 helm gitlab

1 T 购买

四、购买资产

• 购买 helm-gitlab 这个资产

The screenshot shows the TenxCloud Marketplace interface. At the top, there is a search bar with placeholder text '请输入名称关键字' and a '搜索' button. To the right, there are links for '控制台' and 'zs-street'. Below the header, there are navigation tabs: '容器镜像', 'Operator', 'Helm Chart 应用', and 'API'. The main content area displays a product card for 'helm-gitlab'. The card includes the GitLab logo, the product name 'helm-gitlab Helm 已认证', a timestamp '9分钟前更新', and a download count '0'. On the right side of the card is a blue '立即购买' button. Below the card, there is a detailed configuration panel with tabs for '产品介绍', '价格' (selected), '版本', and '服务支持'. The '价格' tab shows the following information:

购买方式	套餐
规格	1
有效时长	30 日
价格	1 T 合: 0.03 T / 天

At the bottom of the configuration panel is a blue '购买' button.

• 支付

支付有两种方式，购买的时候直接付款，或者先生成订单，在一定时间范围内，继续支付。

The screenshot shows the '我的订单' (My Orders) page in the TenxCloud Marketplace. At the top, there is a search bar with placeholder text '请输入名称关键字' and a '搜索' button. To the right, there are links for '控制台' and 'zs-street'. Below the header, there are navigation tabs: '容器镜像', 'Operator', 'Helm Chart 应用', and 'API'. The main content area displays a single order entry. The order details are as follows:

购买商品	helm-gitlab
规格	1
有效时长	30 日
应付金额	1 T

Below the order details, there is a section for payment method selection. It shows two dropdown menus: one for '付款项目' (Payment Item) with options 'zs' and 'one', and another for '账户余额' (Account Balance) with the note '(余额: 7 T)'. There is also a checked checkbox for accepting the '云原生应用市场用户协议' (Cloud Native Application Market User Agreement). At the bottom is a blue '支付' (Pay) button.



• 完成支付

订单ID	资产类型	资产名称	订单类型	购买者	出售者	应付/实付	订单状态	下单时间	支付时间	操作
AOID-J9...	Helm C...	helm-git...	新购	zs-street	admin	1T/1T	● 已支付	2021-07...	2021-07...	详情 ...

• 查看已购资产

资产名称	交付类型	初次购买时间	到期时间	使用情况	操作
helm-gitlab	Helm Chart 应用	2021-07-21 17:39:32	2021-08-20 17:40:51	-	资产详情 ...
helm-mongodb	Helm Chart 应用	2021-07-21 16:16:38	2021-08-20 16:20:02	-	资产详情 ...
helm-artifactory	Helm Chart 应用	2021-07-21 14:42:58	2021-08-20 14:44:41	-	资产详情 ...
helm-mem	Helm Chart 应用	2021-07-20 16:22:08	2021-08-19 16:37:14	-	资产详情 ...

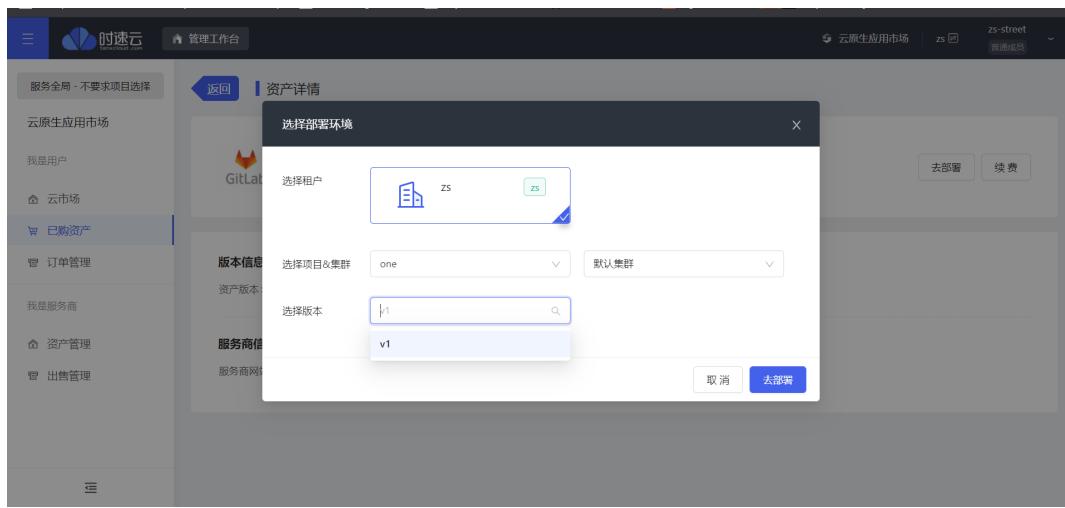
五、部署

• 部署已购资产

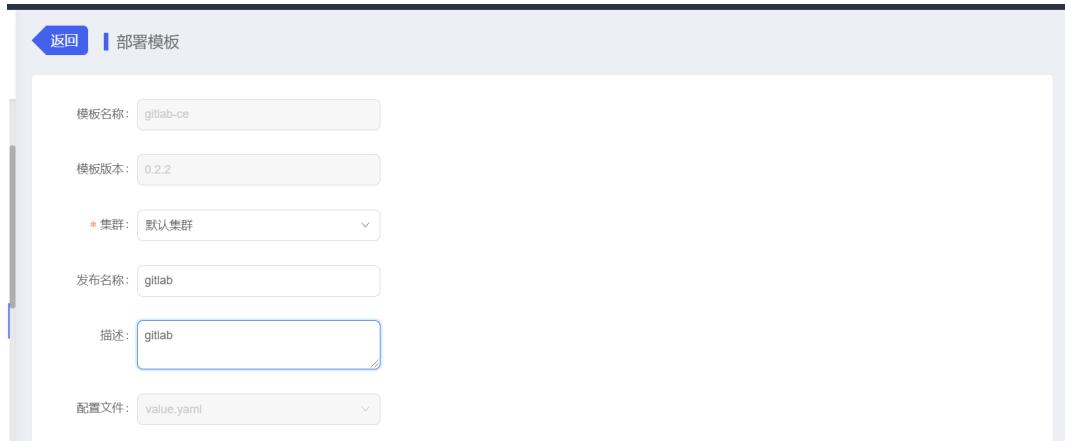
- 找到已购资产，点击部署，或者从资产商店 已购资产详情 点击部署

容器云	公开模板	私有模板	已购模板
工作负载			
服务配置			
网络管理			
存储管理			
交付中心			
应用模板			
镜像仓库			
应用包管理			
编排文件			
应用包发布			

The screenshot shows a list of pre-installed application templates. A red arrow points to the '部署' (Deploy) button for the 'gitlab-ce' template.



- 填写部署名称



修改 `values.yaml`, 给gitlab, redis, pg配置存储.

```

## GitLab CE image
## ref: https://hub.docker.com/r/gitlab/gitlab-ce/tags/
##
image: gitlab/gitlab-ce:9.4.1-ce.0

## Specify a imagePullPolicy
## 'Always' if imageTag is 'latest', else set to 'IfNotPresent'
## ref: http://kubernetes.io/docs/user-guide/images/#pre-pulling-images
##
# imagePullPolicy:

## The URL (with protocol) that your users will use to reach the
install.
## ref:
https://docs.gitlab.com/omnibus/settings/configuration.html#configuring-
the-external-url-for-gitlab
##
externalurl: http://your-domain.com/

## Change the initial default admin password if set. If not set, you'll
be
## able to set it when you first visit your install.
##
# gitlabRootPassword: ""

## For minikube, set this to NodePort, elsewhere use LoadBalancer

```

```

## ref: http://kubernetes.io/docs/user-guide/services/#publishing-
services---service-types
##
serviceType: ClusterIP

## Ingress configuration options
##
ingress:
  annotations:
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
  enabled: false
  tls:
    # - secretName: gitlab.cluster.local
    #   hosts:
    #     - gitlab.cluster.local
  url: gitlab.cluster.local

## Configure external service ports
## ref: http://kubernetes.io/docs/user-guide/services/
sshPort: 22
httpPort: 80
httpsPort: 443
## livenessPort Port of liveness probe endpoint
livenessPort: http
## readinessPort Port of readiness probe endpoint
readinessPort: http

## Configure resource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resources:
  ## GitLab requires a good deal of resources. We have split out
Postgres and
  ## redis, which helps some. Refer to the guidelines for larger
installls.
  ## ref: https://docs.gitlab.com/ce/install/requirements.html#hardware-
requirements
  requests:
    memory: 1Gi
    cpu: 500m
  limits:
    memory: 2Gi
    cpu: 1

## Enable persistence using Persistent Volume Claims
## ref: http://kubernetes.io/docs/user-guide/persistent-volumes/
## ref: https://docs.gitlab.com/ce/install/requirements.html#storage
##
persistence:
  ## This volume persists generated configuration files, keys, and
certs.
  ##
  gitlabEtc:
    enabled: true
    size: 1Gi
    ## If defined, volume.beta.kubernetes.io/storage-class:
<storageClass>

```

```

## Default: volume.alpha.kubernetes.io/storage-class: default
##
storageClass: 'cephrbd' ##### 这里添加gitlabEtc存储
accessMode: ReadWriteOnce
## This volume is used to store git data and other project files.
## ref:
https://docs.gitlab.com/omnibus/settings/configuration.html#storing-git-data-in-an-alternative-directory
##
gitlabData:
  enabled: true
  size: 10Gi
  ## If defined, volume.beta.kubernetes.io/storage-class:
<storageClass>
  ## Default: volume.alpha.kubernetes.io/storage-class: default
  ##
  storageClass: 'cephrbd' ##### 这里添加gitlabData存储
  accessMode: ReadWriteOnce

## Configuration values for the postgresql dependency.
## ref:
https://github.com/kubernetes/charts/blob/master/stable/postgresql/README.md
##
postgresql:
  # 9.6 is the newest supported version for the GitLab container
  imageTag: "9.6"
  cpu: 1000m
  memory: 1Gi

  postgresUser: gitlab
  postgresPassword: gitlab
  postgresDatabase: gitlab

  persistence:
    size: 10Gi
    storageClass: 'cephrbd' # 这里配置pg的存储
## Configuration values for the redis dependency.
## ref:
https://github.com/kubernetes/charts/blob/master/stable/redis/README.md
##
redis:
  redisPassword: "gitlab"

  resources:
    requests:
      memory: 1Gi

  persistence:
    size: 10Gi
    storageClass: 'cephrbd' # 这里配置redis的存储

```

- 部署成功

「one(one)」项目中「模板应用」配额使用情况 5/无限制 刷新

部署成功

C 刷新 D 删除 请输入模板应用名称搜索 搜索 共计5条 < 1 >

□ 模板应用	状态	模板版本	应用版本	创建时间	操作
gitlab	已部署	gitlab-ce-0.2.2	1	1分钟前	查看/编辑 ...

pvc

时速云 管理工作台 one - 默认集群 容器云 存储管理 存储卷

+ 创建存储卷 C 刷新 D 删除 输入名称进行搜索 搜索 共计6条 < 1 >

□ 卷名称	状态	绑定存储卷(PV)	存储类	存储大小	访问模式	操作
gitlab-gitlab-ce-etc	已使用	pvc-e06b69ba-6b3d-4816-842f-00e0fac69cab	undefined	1Gi	单主机读写	查看Yaml ...
gitlab-postgresql	已使用	pvc-639e906b-6e37-4d44-93b0-b9dec2ec9a2f	undefined	4Gi	单主机读写	查看Yaml ...
gitlab-redis	已使用	pvc-70efa063-5c35-4166-be24-4e8b63f827e5	undefined	2Gi	单主机读写	查看Yaml ...
gitlab-gitlab-ce-data	已使用	pvc-6f757fe5-3354-41a9-a6da-485de090498a	undefined	2Gi	单主机读写	查看Yaml ...

deployments

时速云 管理工作台 one - 默认集群 容器云 部署

+ Deployment C 刷新 D 删除 搜索 共计5条 < 1 >

□ 名称	状态	镜像	创建时间	操作
gitlab-gitlab-ce	运行中	查看镜像地址	6分钟前	滚动发布 ...
gitlab-postgresql	运行中	查看镜像地址	6分钟前	滚动发布 ...
gitlab-redis	运行中	查看镜像地址	6分钟前	滚动发布 ...

时速云 管理工作台 one - 默认集群 容器云 模板应用

gitlab 状态: 已部署 资源: 12

当前版本: 1 描述: gitlab

资源 历史版本 C 刷新 共计12条 < 1 >

资源名称	类型	创建时间	操作
gitlab-postgresql	Secret	13分钟前	查看Yaml
gitlab-redis	Secret	13分钟前	查看Yaml
gitlab-gitlab-ce	Secret	13分钟前	查看Yaml

• 访问

◦ 添加路由

The screenshot shows the ZStack Cloud Management Platform interface. On the left, there's a sidebar with various service categories like Container Cloud, Application Routing, DNS Record, etc. The main area is titled 'Route Rule' and contains fields for 'Rule Name' (gitlab), 'Description' (gitlab), 'Export Selection' (选择出口: 缺省服务代理), 'Service Selection' (选择服务: gitlab-gitlab-ce), and 'Port Protocol' (端口协议: TCP). A red arrow points to the '选择出口' field with the text '服务, 这里选择<部署名称>-gitlab-ce'. At the bottom, there are 'Cancel' and 'Create' buttons.

○ 查看路由信息

This screenshot shows the 'Route Rule Details' page for the 'gitlab' rule. It displays basic information: Rule Name (gitlab), Export Type (服务代理), Bound Service (gitlab-gitlab-ce), Service Name (-), and IP (192.168.2.217). Below this, under 'Route Rules', it lists a single entry for TCP port 80 (Container Port) mapped to 39420 (Proxy Port). There are 'Edit' and 'Delete' buttons at the top right of the rule card.

○ 访问

The screenshot shows a web browser window with the URL http://192.168.2.217:39420/users/password/edit?reset_password_token=BsgwsgnxFJt6aG7X63hN. The page has a blue header bar with the message 'Please create a password for your new account.' Below it, the GitLab logo and 'Community Edition' are visible. The main content area says 'Open source software to collaborate on code' and 'Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.' To the right, there's a 'Change your password' form with fields for 'New password' and 'Confirm new password', and a 'Change your password' button. At the bottom, there are links for 'Didnt receive a confirmation email? Request a new one' and 'Already have login and password? Sign in'.

← → C ▲ 不安全 | 192.168.2.217:39420/users/sign_in

IETF | RFCs PaaS B 当前sprint... olm-api OSDev.org + Index... Expanded Main P... 6.828 / Fall 2014 nginx location https://www.json... 无痕模式 更新 阅读清单

Your password has been changed successfully.

GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in Register

Username or email

Password

Remember me [Forgot your password?](#)

Sign in

Didn't receive a confirmation email? [Request a new one.](#)

Explore Help About GitLab

← → C ▲ 不安全 | 192.168.2.217:39420

IETF | RFCs PaaS B 当前sprint... olm-api OSDev.org + Index... Expanded Main P... 6.828 / Fall 2014 nginx location https://www.json... 无痕模式 更新 阅读清单

Projects

Customize your experience

Change syntax themes, default project pages, and more in preferences.

Check it out

Welcome to GitLab

Code, test, and deploy together

Add user

Add your team members and others to GitLab.

New user

正在连接...

← → C ▲ 不安全 | 192.168.2.217:39420/groups/new

IETF | RFCs PaaS B 当前sprint... olm-api OSDev.org + Index... Expanded Main P... 6.828 / Fall 2014 nginx location https://www.json... 无痕模式 更新 阅读清单

Groups

New Group

Group path

Group name

Description

Group avatar The maximum file size allowed is 200KB.

Visibility Level Private The group and its projects can only be viewed by members.

internal The group and any internal projects can be viewed by any logged in user.

Public The group and any public projects can be viewed without any authentication.

New project

Create or Import your project from popular Git services

Project path: http://192.168.2.217:39420/ test1

Project name: api

Want to house several dependent projects under the same namespace? [Create a group](#)

Import project from:

- GitHub
- Bitbucket
- GitLab.com
- Google Code
- Fogbugz
- Gitea
- git Repo by URL
- GitLab export

Project description (optional)

Description format:

Visibility Level:

- Private: Project access must be granted explicitly to each user.
- Internal: The project can be accessed by any logged in user.
- Public: The project can be accessed without any authentication.

正在等待 192.168.2.217 的响应...

test1 / api

You won't be able to pull or push project code via SSH until you [add an SSH key](#) to your profile

Don't show again | Remind later

A
api

Star 0 HTTP <http://your-domain.com/test1/api>

The repository for this project is empty

If you already have files you can push them using command line instructions below.

Otherwise you can start with adding a [README](#), a [LICENSE](#) or a [.gitignore](#) to this project.

You will need to be owner or have the master permission level for the initial push, as the master branch is automatically protected.

Command line instructions

Git global setup

Project Repository Issues 0 Merge Requests 0 Pipelines Wiki Snippets Members Settings

Files Commits Branches Tags Contributors Graph Compare Charts

The file has been successfully created.

master api / README.md

Add readme.md
Administrator committed about a minute ago 63fc4082

README.md 10 Bytes

Find file Blame History Permalink Edit Replace Delete

README