

# 安装部署 gitlab

## 1. gitlab 配置

## 1.下载gitlab安装包

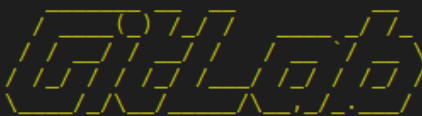
```
wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/gitlab-ce-16.9.1-
ce.0.el7.x86_64.rpm
```

## 2.安装gitlab

```
yum install -y git tar policycoreutils-python openssh-server
rpm -ivh gitlab-ce-16.9.1-ce.0.el7.x86_64.rpm
```

安装成功会有如下信息:

```
[root@localhost home]# rpm -ivh gitlab-ce-16.9.1-ce.0.el7.x86_64.rpm
warning: gitlab-ce-16.9.1-ce.0.el7.x86_64.rpm: Header V4 RSA/SHA1 Signature, key ID f27eab47: NOKEY
Preparing...                               ##### [100%]
Updating / installing...
 1:gitlab-ce-16.9.1-ce.0.el7                ##### [100%]
It looks like GitLab has not been configured yet; skipping the upgrade script.
```



```
Thank you for installing GitLab!
GitLab was unable to detect a valid hostname for your instance.
Please configure a URL for your GitLab instance by setting `external_url`
configuration in /etc/gitlab/gitlab.rb file.
Then, you can start your GitLab instance by running the following command:
  sudo gitlab-ctl reconfigure
```

For a comprehensive list of configuration options please see the Omnibus GitLab readme <https://gitlab.com/gitlab-org/omnibus-gitlab/blob/master/README.md>

Help us improve the installation experience, let us know how we did with a 1 minute survey:  
[https://gitlab.fra1.qualtrics.com/jfe/form/SV\\_6kVgZANTHu01bZb?installation=omnibus&release=16-9](https://gitlab.fra1.qualtrics.com/jfe/form/SV_6kVgZANTHu01bZb?installation=omnibus&release=16-9)

### 3.修改IP端口

#### 编辑gitlab配置文件

```
vi /etc/gitlab/gitlab.rb
```

#### 修改对外访问的Ip端口：

```
external_url 'http://localhost:8081' # 修改成 http://实际ip:需要访问的端口
```

```
##!  
##! Note: During installation/upgrades, the value of the environment variable  
##! EXTERNAL_URL will be used to populate/replace this value.  
##! On AWS EC2 instances, we also attempt to fetch the public hostname/IP  
##! address from AWS. For more details, see:  
##! https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html  
external_url 'http://90.███.███.███:8080/'  
## Roles for multi-instance Gitlab
```

设置的端口不能被占用，如设置的端口已经使用，请自定义其它端口，并在防火墙设置开放范围内的端口

- 执行systemctl status firewalld命令查看服务器OS防火墙的状态。如果防火墙已开启（active），执行如下操作开通防火墙端口；如果防火墙没有开启（inactive），请跳过以下步骤。
- 执行firewall-cmd --query-port=8081/tcp命令查看端口是否开通，提示“no”表示端口未开通。
- 执行firewall-cmd --add-port=8081/tcp --permanent命令永久开通端口，提示“success”表示开通成功。
- 执行firewall-cmd --reload命令重新载入配置。
- 再次执行firewall-cmd --query-port=8081/tcp命令查看端口是否开通，提示“yes”表示端口已开通。

### 4.重新加载配置

```
gitlab-ctl reconfigure #重新生成相关配置文件，执行此命令时间比较长
```

```
Running handlers:  
[2024-02-27T14:42:04+08:00] INFO: Running report handlers  
Running handlers complete  
[2024-02-27T14:42:04+08:00] INFO: Report handlers complete  
Infra Phase complete, 580/1601 resources updated in 04 minutes 50 seconds  
  
Notes:  
Default admin account has been configured with following details:  
Username: root  
Password: You didn't opt-in to print initial root password to STDOUT.  
Password stored to /etc/gitlab/initial_root_password. This file will be cleaned up in first reconfigure run after 24 hours.  
  
NOTE: Because these credentials might be present in your log files in plain text, it is highly recommended to reset the password following https://docs.gitlab.com/ee/security/reset\_user\_password.html#reset-your-root-password.  
gitlab_Reconfigured!
```

### 5.配置gitlab开机自动启动

```
systemctl enable gitlab-runsvdir.service
systemctl start gitlab-runsvdir.service
# 关闭gitlab的自动启动命令: systemctl disable gitlab-runsvdir.service
```

## 6.启动gitLab

```
gitlab-ctl restart
```

启动成功会有如下信息

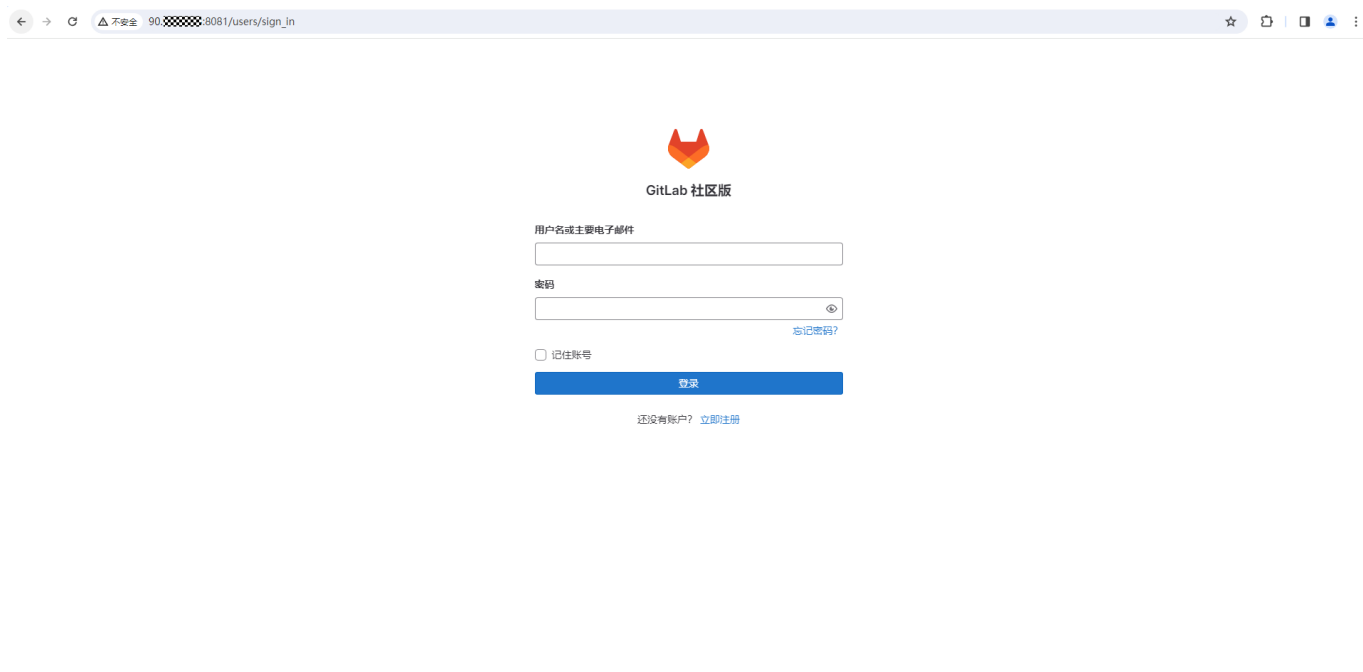
```
[root@localhost home]# gitlab-ctl restart
ok: run: alertmanager: (pid 75137) 0s
ok: run: gitaly: (pid 75155) 0s
ok: run: gitlab-exporter: (pid 75242) 0s
ok: run: gitlab-kas: (pid 75553) 0s
ok: run: gitlab-workhorse: (pid 75580) 1s
ok: run: logrotate: (pid 75610) 0s
ok: run: nginx: (pid 75636) 0s
ok: run: node-exporter: (pid 75730) 1s
ok: run: postgres-exporter: (pid 75754) 0s
ok: run: postgresql: (pid 75794) 1s
ok: run: prometheus: (pid 75796) 0s
ok: run: puma: (pid 75920) 0s
ok: run: redis: (pid 75925) 0s
ok: run: redis-exporter: (pid 75950) 1s
ok: run: sidekiq: (pid 76017) 0s
```

## 7.查看 gitlab 版本

```
cat /opt/gitlab/embedded/service/gitlab-rails/VERSION # 回显应为16.9.1
```

## 8.页面访问 gitlab

```
http://ip:8081/ # 端口根据个人配置进行更改
```



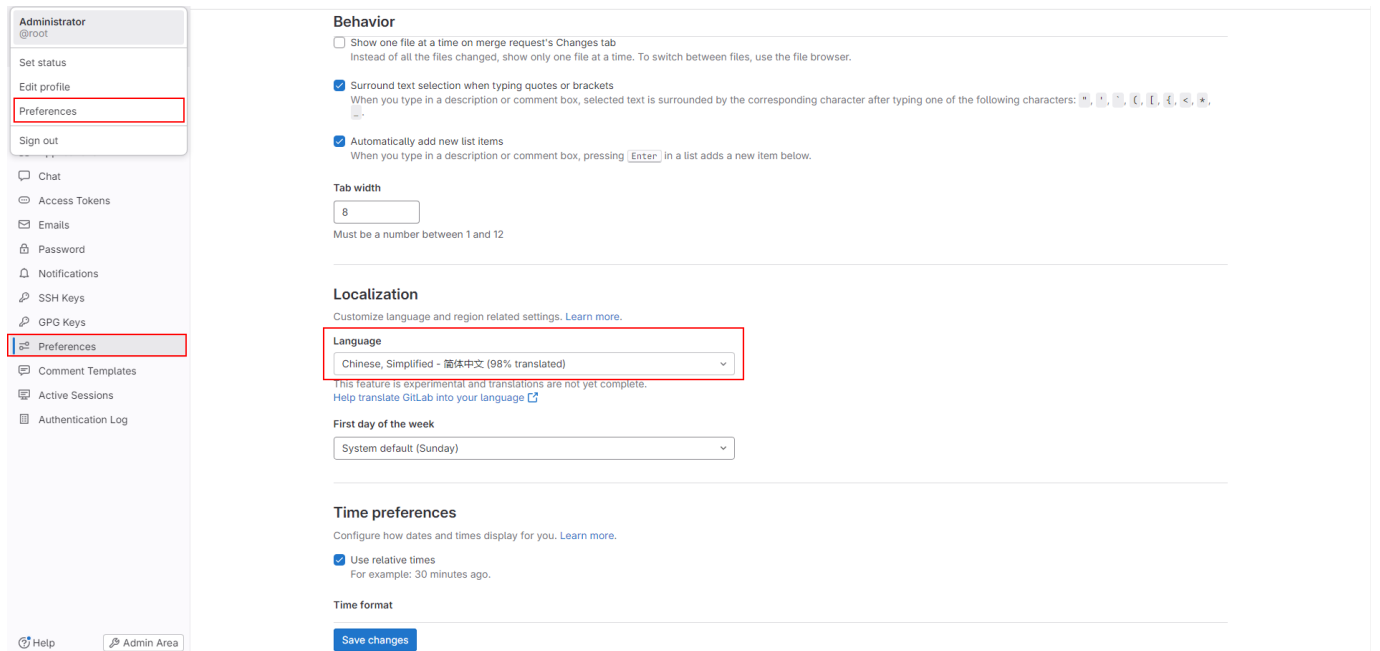
## 9. 登录 gitlab

默认账户名是root,密码存放在配置文件 /etc/gitlab/initial\_root\_password

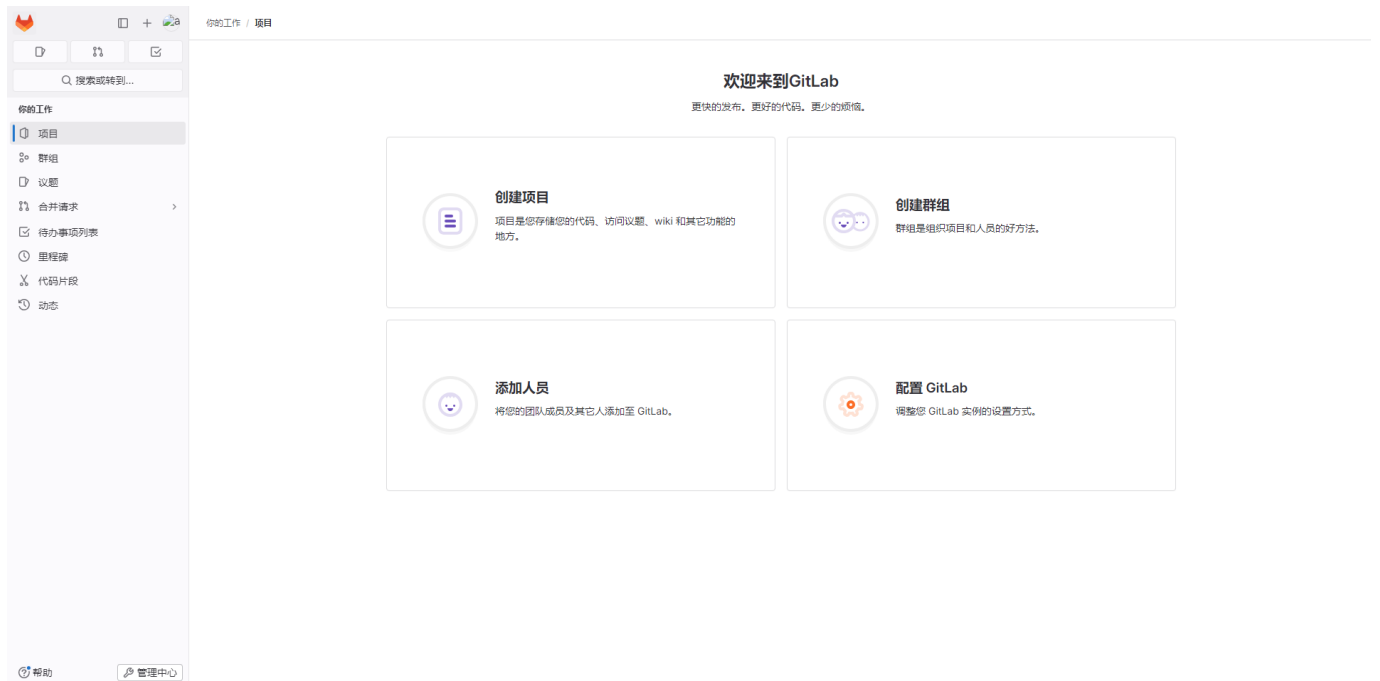
```
[root@localhost home]# cat /etc/gitlab/initial_root_password
# WARNING: This value is valid only in the following conditions
# 1. If provided manually (either via 'GITLAB_ROOT_PASSWORD' environment variable or via 'gitlab_rails['initial_root_password']' setting in 'gitlab.rb', it was provided before database was se
# 2. Password hasn't been changed manually, either via UI or via command line.
#
# If the password shown here doesn't work, you must reset the admin password following https://docs.gitlab.com/ee/security/reset_user_password.html#reset-your-root-password.
Password: 1+zpvMCPp0EF/CPxJtj5V+qILNOX4AFpsmRhagXlhps=
# NOTE: This file will be automatically deleted in the first reconfigure run after 24 hours.
```

## 10. 设置为简体中文

先登录gitLab，登录成功后，在gitlab后台的系统设置里配置简体中文。



完成后刷新页面即可



## 2. 安装部署 gitlab Runner

### 1. 下载gitlab Runner安装包

```
wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-runner/yum/el7-aarch64/gitlab-runner-16.9.0-1.aarch64.rpm
```

# 也可在 <https://mirrors.tuna.tsinghua.edu.cn/> 或 <https://gitlab-runner-downloads.s3.amazonaws.com/latest/index.html> 中下载符合自己环境的gitlab Runner安装包，这里以gitlab-runner-16.9.0-1.aarch64.rpm为例，进行手动下载后传至服务器

## 2.安装gitlab Runner安装包

```
yum install -y git tar  
rpm -ivh gitlab-runner-16.9.0-1.aarch64.rpm
```

## 3.指定gitlab-runner

```
useradd Kunpeng_staff  
gitlab-runner uninstall  
gitlab-runner install --working-directory /home/Kunpeng_staff --user Kunpeng_staff
```

## 4.启动gitlab-runner

systemctl daemon-reload	#重新加载配置
systemctl start gitlab-runner	#启动服务
systemctl enable gitlab-runner	#设置开机启动
systemctl restart gitlab-runner	#重启服务

## 4.查看gitlab-runner

```
systemctl status gitlab-runner
```

## 成功启动如下图所示

```
[root@localhost home]# systemctl status gitlab-runner
● gitlab-runner.service - GitLab Runner
   Loaded: loaded (/etc/systemd/system/gitlab-runner.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2016-02-27 16:08:09 CST; 36min ago
     Main PID: 123788 (gitlab-runner)
       Tasks: 40 (limit: 814436)
      Memory: 27.0M
     CGroup: /system.slice/gitlab-runner.service
             └─ 123788 /usr/bin/gitlab-runner run --working-directory /home/gitlab-runner --config /etc/gitlab-runner/config.toml --service gitlab-runner --user gitlab-runner

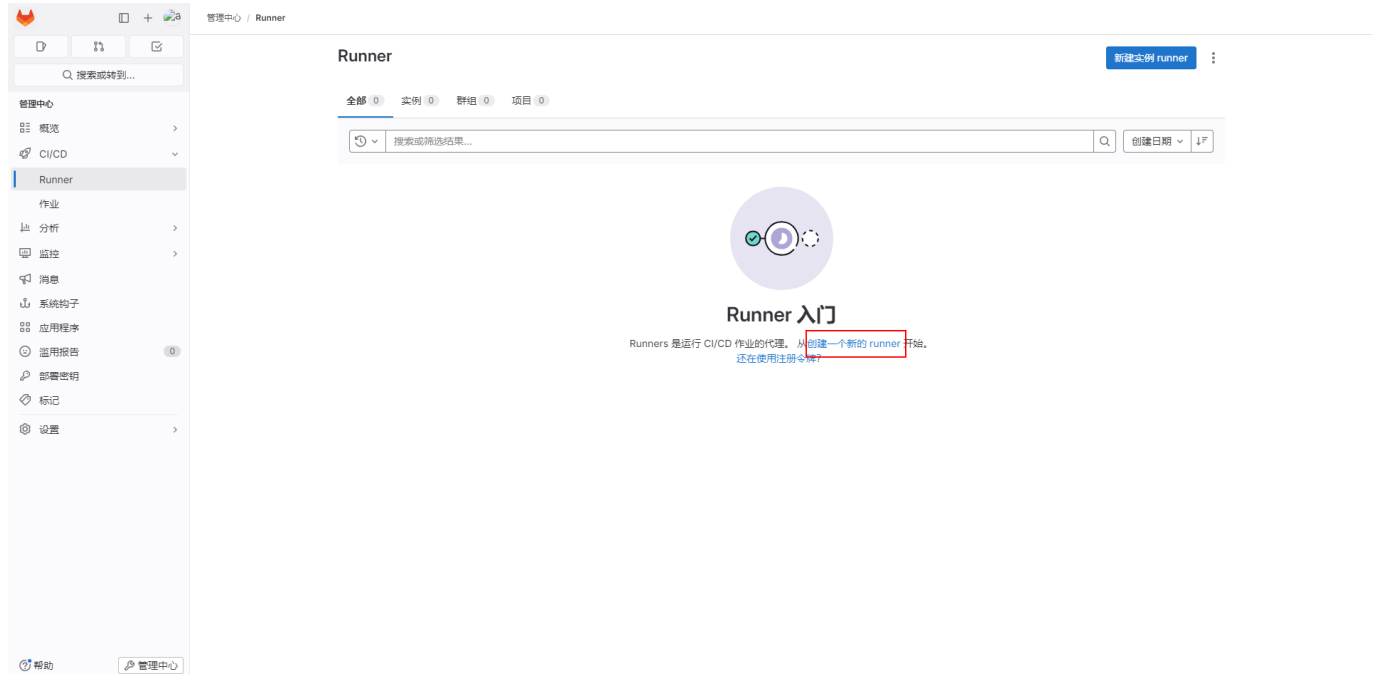
2月 27 16:08:09 localhost.localdomain gitlab-runner[123788]: Runtime platform arch=arm64 os=linux pid=123788 revision=656c1943 version=16.9.0
2月 27 16:08:09 localhost.localdomain gitlab-runner[123788]: Starting multi-runner from /etc/gitlab-runner/config.toml... builds=0 max_builds=0
2月 27 16:08:09 localhost.localdomain gitlab-runner[123788]: Running in system-mode.
2月 27 16:08:09 localhost.localdomain gitlab-runner[123788]: Created missing unique system ID system_id=s_c2c042740ad2
2月 27 16:08:09 localhost.localdomain gitlab-runner[123788]: Configuration loaded builds=0 max_builds=1
2月 27 16:08:09 localhost.localdomain gitlab-runner[123788]: Listen address not defined, metrics & debug endpoints disabled builds=0 max_builds=1
2月 27 16:08:09 localhost.localdomain gitlab-runner[123788]: [session server] listen address not defined, session endpoints disabled builds=0 max_builds=1
2月 27 16:08:09 localhost.localdomain gitlab-runner[123788]: Initializing executor providers builds=0 max_builds=1
2月 27 16:10:36 localhost.localdomain systemd[1]: gitlab-runner.service: Current command vanished from the unit file, execution of the command list won't be resumed.
[root@localhost home]#
```

## 5.设置权限

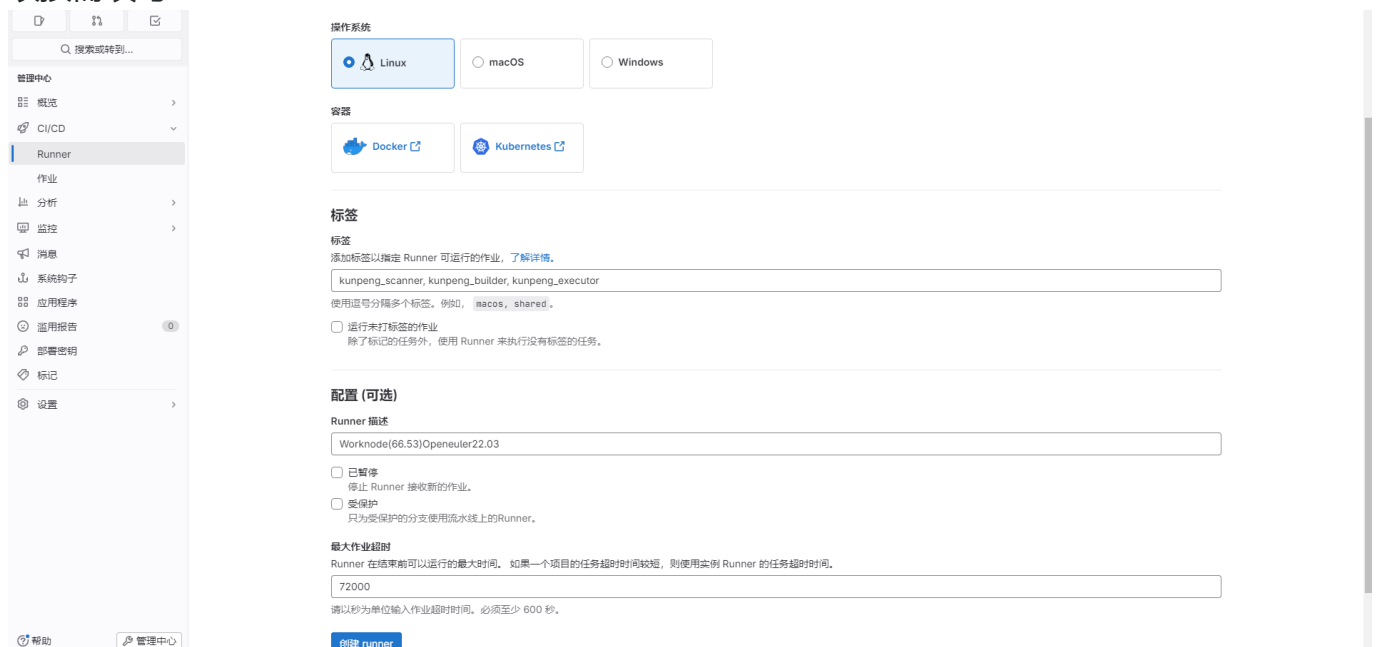
```
chown -R Kunpeng_staff.Kunpeng_staff /home/gitlab-runner
```

## 6.gitlab runner注册服务

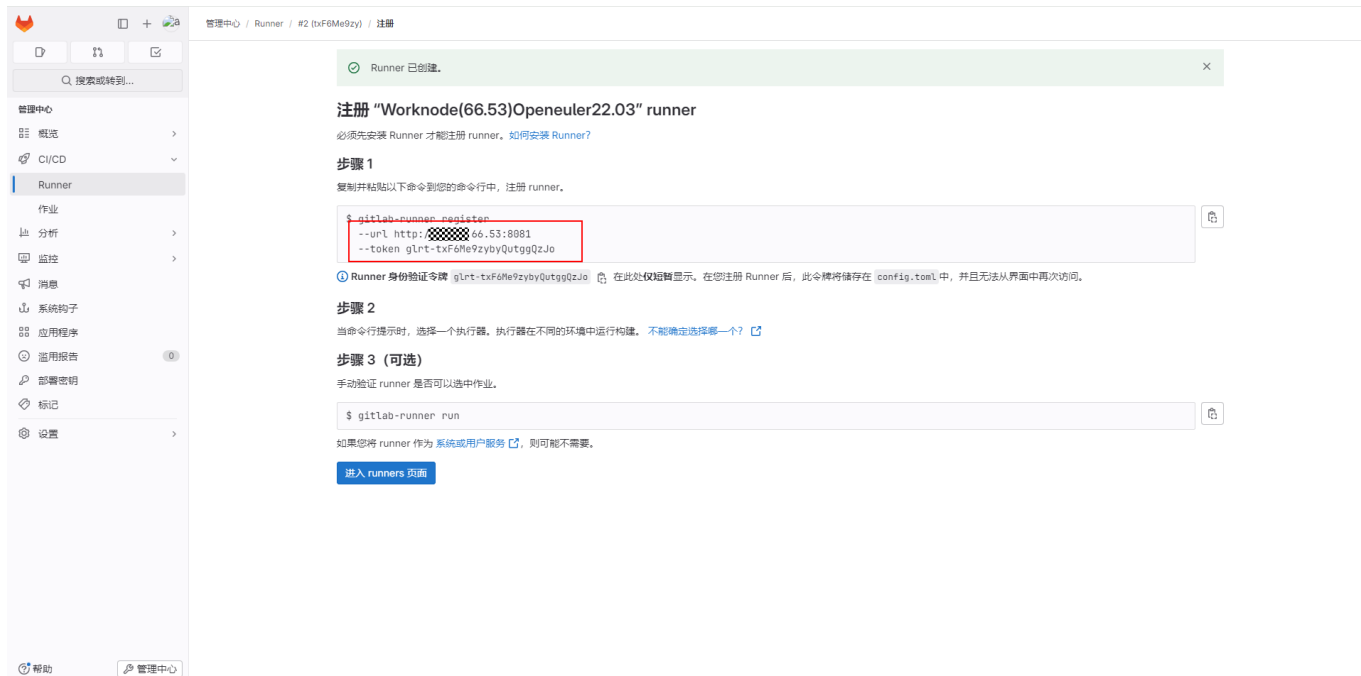
登录gitlab，在管理中心Runner中新建一个runner



标签多个时用逗号隔开，只可为kunpeng\_scanner, kunpeng\_builder, kunpeng\_executor，其他选项按需填写



根据所示url，token去gitlab-runner环境注册



运行 gitlab-runner register，参数如图根据实际情况填写，注册成功后会显示Runner registered successfully

```
[root@localhost home]# gitlab-runner register
Runtime platform                          arch=arm64 os=linux pid=233027 revision=656c1943 version=16.9.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://66.53:8081
Enter the registration token:
glrt-txF6Me9zybyQutggQzJo
Verifying runner... is valid                      runner=txF6Me9zy
Enter a name for the runner. This is stored only in the local config.toml file:
[localhost.localdomain]: Kunpeng-gitlab-runner
Enter an executor: custom, virtualbox, docker-windows, docker+machine, kubernetes, docker-autoscaler, instance, shell, ssh, parallels, docker:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!

Configuration (with the authentication token) was saved in "/etc/gitlab-runner/config.toml"
```

点击进入runners页面



搜索或转到...

管理中心

作业

分析

监控

消息

系统构子

应用程序

应用报告

部署密钥

标记

设置

帮助

管理中心

管理中心 / Runner / #2 (txF6Me9zy) / 注册

Runner 已创建。

### 注册 "Worknode(66.53)Openeuler22.03" runner

必须先安装 Runner 才能注册 runner。 [如何安装 Runner?](#)

#### 步骤 1

复制并粘贴以下命令到您的命令行中，注册 runner。

```
$ gitlab-runner register
--url http://66.53:8081
--token glrt-txF6Me9zybyQutggQzJo
```

Runner 身份验证令牌 `glrt-txF6Me9zybyQutggQzJo` 在此处仅短暂显示。在您注册 Runner 后，此令牌将储存在 `config.toml` 中，并且无法从界面中再次访问。

#### 步骤 2

当命令行提示时，选择一个执行器。执行器在不同的环境中运行构建。 [不能确定选择哪一个?](#)

#### 步骤 3 (可选)

手动验证 runner 是否可以选中作业。

```
$ gitlab-runner run
```

如果您将 runner 作为 系统或用户服务，则可能不需要。

进入 runners 页面

搜索或转到...

管理中心

作业

分析

监控

消息

系统构子

应用程序

应用报告

部署密钥

标记

设置

帮助

管理中心

管理中心 / Runner

Runner

新建实例 runner

全部 1 实例 1 群组 0 项目 0

搜索或筛选结果...

创建日期

在线 离线 Stale

1 0 0

状态	Runner	所有者
<div>在线</div>	<div>#2 (txF6Me9zy)</div> <div>版本 16.9.0 · Worknode(66.53)Openeuler22.03</div> <div>最后联系: 34秒前</div> <div><a href="#">kubernetes_scanner</a> <a href="#">kubernetes_builder</a> <a href="#">kubernetes_executor</a></div>	管理员