安装部署 gitlab

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-serve
rpm -ivh gitlab-ce-16.9.1-ce.0.el7.x86_64.rpm
```

安装成功会有如下信息：

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
a． 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。
b． 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。
c． 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。
d． 执行firewall-cmd --reload命令重新载入配置。
e． 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
```
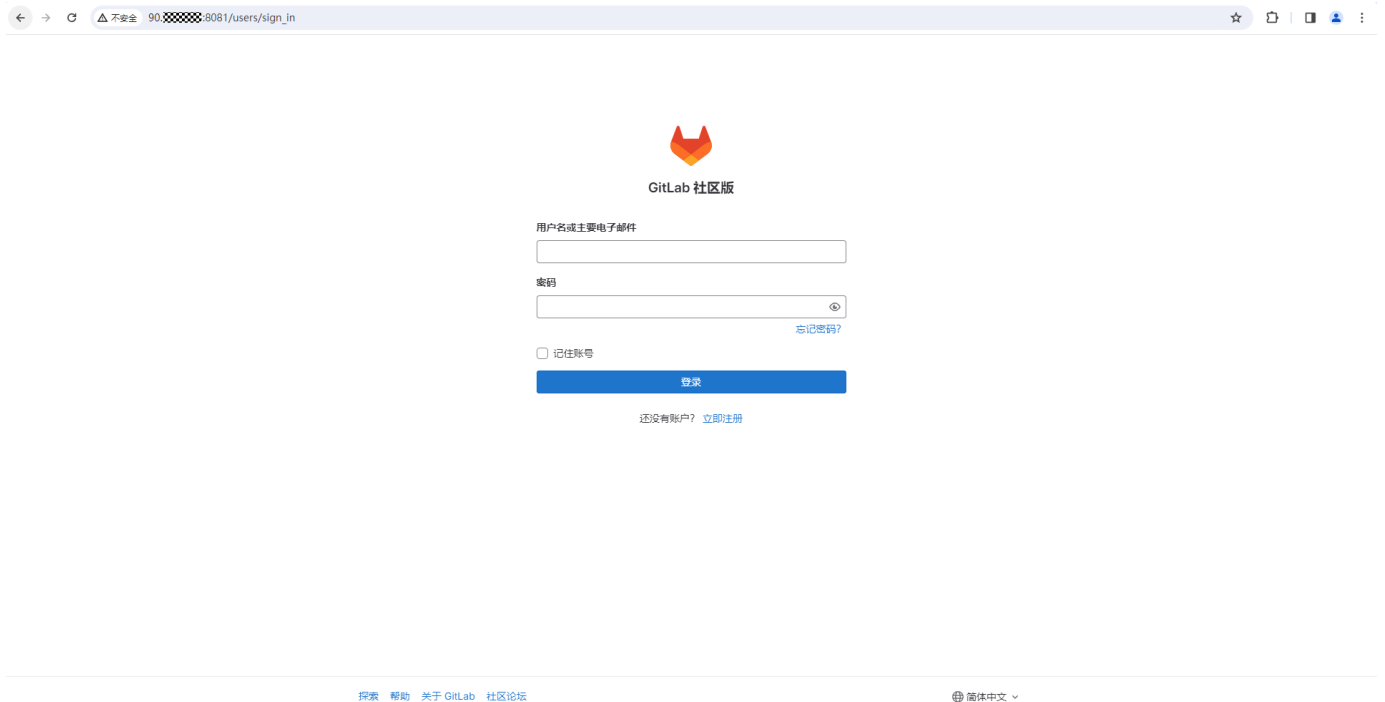
启动成功会有如下信息

```
[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
```

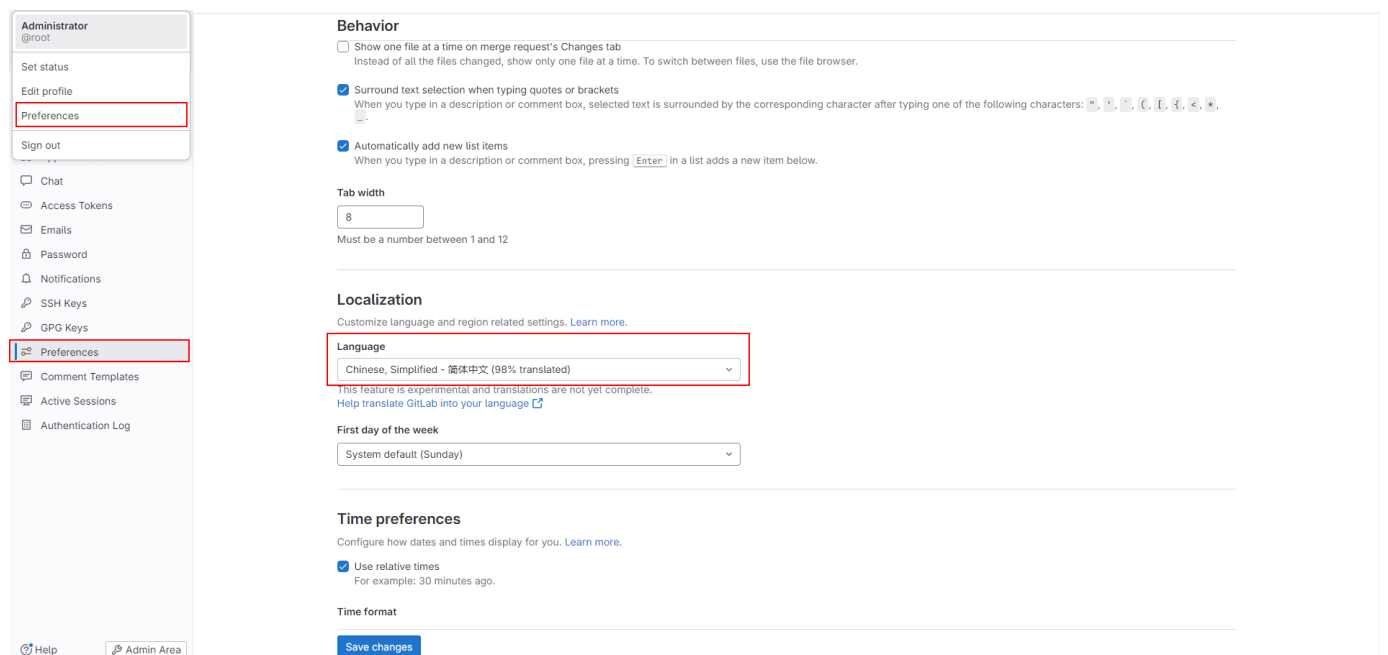## 8.页面访问 gitlab

```
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
eded for the first time (usually, the first reconfigure run).
#          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+zpvMCPpOEF/CPxJtj5V+qILNOX4AFpsmRhagX1hps=

# 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

你的工作

项目

群组

议题

合并请求

待办事项列表

里程碑

代码片段

动态

搜索或转到...

帮助

管理中心

欢迎来到GitLab

更快的发布。更好的代码。更少的烦恼。

创建项目

项目是您存储您的代码、访问议题、wiki 和其它功能的地方。

创建群组

群组是组织项目和人员的好方法。

添加人员

将您的团队成员及其它人添加至 GitLab。

配置 GitLab

调整您 GitLab 实例的设置方式。