

P8-07-09 DevOps中的CI/CD环境搭建与调优

一、知识点回顾

链接: <https://pan.baidu.com/s/1vP0XXpSzSQ6afHesJGm2Uw>

提取码: data

P8架构项目中所需要的Canal 主要用户MySQL主从架构增量或全量备份/导入ES 或Kafka中

二、本节课的主要内容

Java项目 Spring 项目 开发过程开发团队中需要有不同的角色 maven编译

项目/质量管理 禅道(国内)/JIRA(国外)

需要项目管理软件 中小型的公司GitLab 代码托管 CI 持续集成

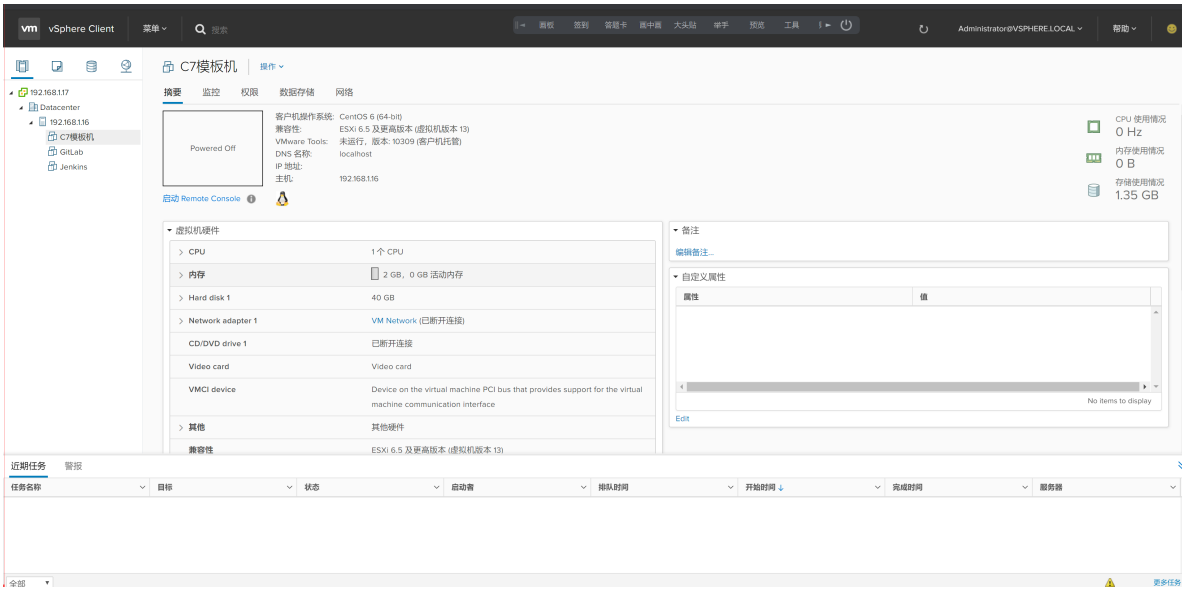
CI 持续集成 开发人员Dev分支开发 Bug修复 提交代码Dev分支 合并分支Master主分支

CD 持续部署 Jenkins/tecmCity

CM 持续监控 Zabbix/等

三、Gitlab 开源的管理软件(注意 阿里云 坑如果不注意/)

克隆一台虚拟机GitLab



GitLab安装

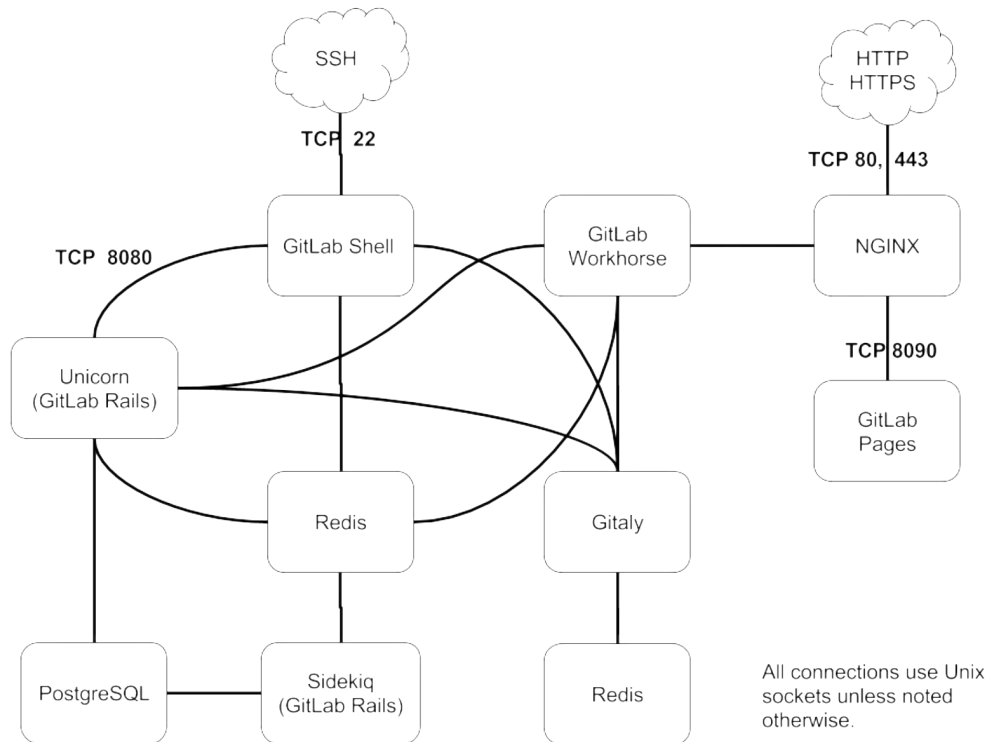
1)在线安装的方式

主机规划

| 主机名称 hostname | 主机规划 (内) | 主机规划 (外) | 安装基础软件 | 系统 | 服务器推 荐配 置 |
|------------------|-------------|-------------|--|------------------------|-----------------|
| GitLab | 10.0.0.10 | 123.57.13.5 | yum install -y vim net-tools lrzsz tree | CentOS7x64 1810 7.6 | 4C 8G |

0x1 GitLab架构

GitLab Application Architecture



国内的源速度比较快/阿里源/清华源

`vim /etc/yum.repos.d/gitlab-ce.repo`

```
[gitlab-ce]
name=Gitlab CE Repository
baseurl=https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el$releasever/
gpgcheck=0
enabled=1
```

`yum makecache && yum install -y gitlab-ce`

```
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

Verifying : audit-libs-2.8.5-4.el7.x86_64 1/14
Verifying : audit-2.8.5-4.el7.x86_64 2/14
Verifying : policycoreutils-2.5-34.el7.x86_64 3/14
Verifying : libsemanage-python-2.5-14.el7.x86_64 4/14
Verifying : python-IPy-0.75-6.el7.noarch 5/14
Verifying : checkpolicy-2.5-8.el7.x86_64 6/14
Verifying : policycoreutils-python-2.5-34.el7.x86_64 7/14
Verifying : audit-libs-python-2.8.5-4.el7.x86_64 8/14
Verifying : libcgrouper-0.41-21.el7.x86_64 9/14
Verifying : gitlab-ce-13.1.3-ce.0.el7.x86_64 10/14
Verifying : setools-libs-3.3.8-4.el7.x86_64 11/14
Verifying : policycoreutils-2.5-29.el7.x86_64 12/14
Verifying : audit-libs-2.8.4-4.el7.x86_64 13/14
Verifying : audit-2.8.4-4.el7.x86_64 14/14

Installed:
gitlab-ce.x86_64 0:13.1.3-ce.0.el7

Dependency Installed:
audit-libs-python.x86_64 0:2.8.5-4.el7 checkpolicy.x86_64 0:2.5-8.el7 libcgrouper.x86_64 0:0.41-21.el7 libsemanage-python.x86_64 0:2.5-14.el7
policycoreutils-python.x86_64 0:2.5-34.el7 python-IPy.noarch 0:0.75-6.el7 setools-libs.x86_64 0:3.3.8-4.el7

Dependency Updated:
audit.x86_64 0:2.8.5-4.el7 audit-libs.x86_64 0:2.8.5-4.el7 policycoreutils.x86_64 0:2.5-34.el7

Complete!
[root@localhost ~]#
```

PS: 还可以自己搭建本地yum源的服务器, 集群内部为了速度, 可以搭建本地yum源

【面试题】SVN Git Gitlab GitHub Gitee/Gitlab坑很多 Redis搭建这个服务的时候有哪些要点?

Redis 默认密码的问题/服务器不安全 有可能被黑。

修改配置文件进行配置

vim /etc/gitlab/gitlab.rb

```
external_url 'IP地址+端口号'
```

gitlab-ctl reconfigure

```
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.promote_secondaries = 1
net.ipv4.conf.all.promote_secondaries = 1
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
* Applying /etc/sysctl.d/90-omnibus-gitlab-kernel.sem.conf ...
kernel.sem = 250 32000 32 262
* Applying /etc/sysctl.d/90-omnibus-gitlab-kernel.shmall.conf ...
kernel.shmall = 4194304
* Applying /etc/sysctl.d/90-omnibus-gitlab-kernel.shmmax.conf ...
kernel.shmmax = 17179869184
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /etc/sysctl.conf ...
-- execute sysctl -e --system
Recipe: gitlab::gitlab-workhorse
* runit_service[gitlab-workhorse] action restart (up to date)
Recipe: monitoring::node-exporter
* runit_service[node-exporter] action restart (up to date)
Recipe: monitoring::gitlab-exporter
* runit_service[gitlab-exporter] action restart (up to date)
Recipe: monitoring::redis-exporter
* runit_service[redis-exporter] action restart (up to date)
Recipe: monitoring::prometheus
* runit_service[prometheus] action restart (up to date)
* execute[reload prometheus] action run
* execute /opt/gitlab/bin/gitlab-ctl hup prometheus
Recipe: monitoring::alertmanager
* runit_service[alertmanager] action restart (up to date)
Recipe: monitoring::postgres-exporter
* runit_service[postgres-exporter] action restart (up to date)
Recipe: monitoring::grafana
* runit_service[grafana] action restart (up to date)

Running handlers:
Running handlers complete
Chef Client finished, 564/1519 resources updated in 13 minutes 20 seconds
gitlab Reconfigured!
[root@localhost ~]#
```

角色管理(思想很重要 开发(宏观) ---->机构师 (微观) ---->送外卖)
P7企业不光是技术

团队管理+项目管理

注意

C7 的防火墙 firewalld

firewall-cmd --zone=public --add-port=8081/tcp --permanent

```
firewall-cmd --reload
```

C6 的防火墙 iptables

或者关闭防火墙 (处于安全性的考虑 建议大家开着)

```
[root@localhost ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-07-09 08:25:47 EDT; 43min ago
     Docs: man:firewalld(1)
   Main PID: 5482 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─5482 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

Jul 09 08:17:19 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Jul 09 08:25:47 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@localhost ~]#
```

如果你用的是云主机 阿里云/金山云/华为云/AWS等

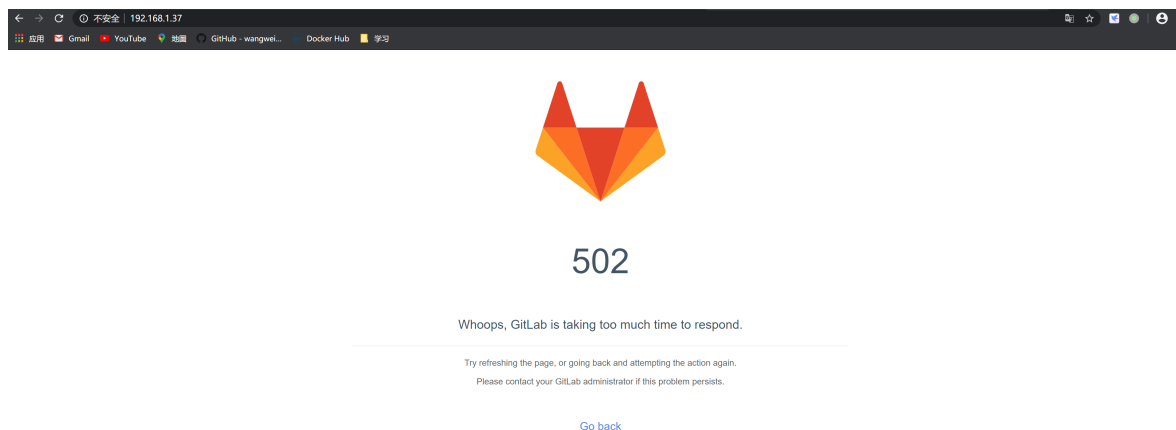
注意：需要把对应的端口添加到 安全组里 否则外网访问的时候出现问题

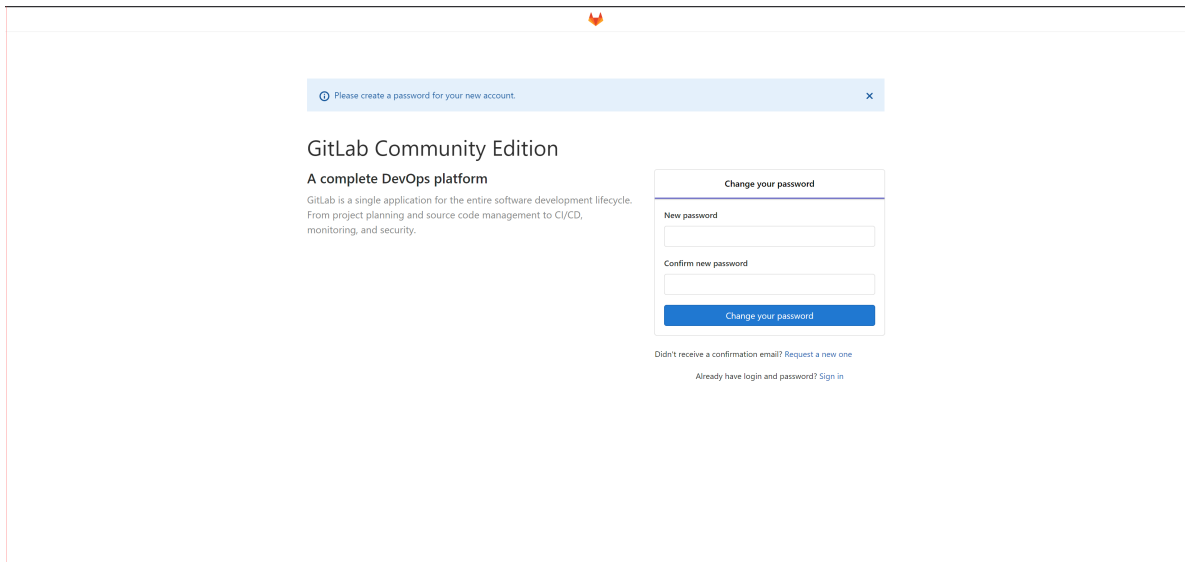
注意：云主机的配置内存不要太小 推荐配置4C 8G 内存太小的话容易出现一些问题。

gitlab-ctl restart

```
[root@localhost ~]# gitlab-ctl restart
ok: run: alertmanager: (pid 9160) 0s
ok: run: gitlab: (pid 9174) 0s
ok: run: gitlab-exporter: (pid 9182) 1s
ok: run: gitlab-workhorse: (pid 9190) 0s
ok: run: grafana: (pid 9198) 1s
ok: run: logrotate: (pid 9208) 0s
ok: run: nginx: (pid 9214) 1s
ok: run: node-exporter: (pid 9220) 0s
ok: run: postgres-exporter: (pid 9232) 1s
ok: run: postgresql: (pid 9244) 0s
ok: run: prometheus: (pid 9246) 0s
ok: run: puma: (pid 9245) 1s
ok: run: redis: (pid 9354) 0s
ok: run: redis-exporter: (pid 9362) 1s
ok: run: sidekiq: (pid 9375) 0s
[root@localhost ~]#
```

到浏览器中输入ip+端口访问





2)离线安装的方式

下载对应的安装包/没有网络/

四、Jenkins 持续部署工具(源码拉取 预处理 编译 构建 打包 镜像 发布)

传统的服务器可以部署

微服务Docker K8S中也可以使用

很多很成熟的插件 提升你的工作效率

hostnamectl set-hostname xxxx

| 主机 hostname | 主机规划 (内) | 主机规划 (外) | 基础软件 | 操作系统 |
|----------------|-------------|-------------|-----------------------|---------------|
| Jenkins | | | vim wget jdk1.8以 上 | C7x64 1810 |

步骤一 安装openjdk

yum install -y java-1.8.0-openjdk*

步骤二 使用清华源进行安装的

<https://mirrors.tuna.tsinghua.edu.cn/jenkins/redhat/jenkins-2.244-1.1.noarch.rpm>

执行以下命令

yum install -y <https://mirrors.tuna.tsinghua.edu.cn/jenkins/redhat/jenkins-2.244-1.1.noarch.rpm>

```
OpenJDK Runtime Environment (build 1.8.0_252-b09)
OpenJDK 64-Bit Server VM (build 25.252-b09, mixed mode)
[root@jenkins ~]# yum install -y https://mirrors.tuna.tsinghua.edu.cn/jenkins/redhat/jenkins-2.244-1.1.noarch.rpm
Loaded plugins: fastestmirror
jenkins-2.244-1.1.noarch.rpm | 64 MB 00:00:06
Examining /var/tmp/yum-root-gc3GhE/jenkins-2.244-1.1.noarch.rpm: jenkins-2.244-1.1.noarch
Marking /var/tmp/yum-root-gc3GhE/jenkins-2.244-1.1.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package jenkins.noarch 0:2.244-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
jenkins noarch 2.244-1.1 /jenkins-2.244-1.1.noarch 64 M
=====
Transaction Summary
=====
Install 1 Package

Total size: 64 M
Installed size: 64 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.244-1.1.noarch 1/1
  Verifying : jenkins-2.244-1.1.noarch 1/1

Installed:
jenkins.noarch 0:2.244-1.1

Complete!
[root@jenkins ~]#
```

步骤三 进行配置

vim /etc/sysconfig/jenkins

踩坑

JENKINS_USER = root

```

## Path:      Development/Jenkins
## Description: Jenkins Automation Server
## Type:      string
## Default:   "/var/lib/jenkins"
## ServiceRestart: jenkins
#
# Directory where Jenkins store its configuration and working
# files (checkouts, build reports, artifacts, ...).
#
JENKINS_HOME="/var/lib/jenkins"

## Type:      string
## Default:   ""
## ServiceRestart: jenkins
#
# Java executable to run Jenkins
# When left empty, we'll try to find the suitable Java.
#
JENKINS_JAVA_CMD=""

## Type:      string
## Default:   "jenkins"
## ServiceRestart: jenkins
#
# Unix user account that runs the Jenkins daemon
# Be careful when you change this, as you need to update
# permissions of $JENKINS_HOME and /var/log/jenkins.
#
JENKINS_USER="jenkins"

## Type:      string
## Default:   "false"
## ServiceRestart: jenkins
#
# Whether to skip potentially long-running chown at the
# $JENKINS_HOME location. Do not enable this, "true", unless
# you know what you're doing. See JENKINS-23273.
#
"/etc/sysconfig/jenkins" 177L, 3937C

```

端口号 8080 tomcat/端口冲突/端口保活

```

JENKINS_JAVA_OPTIONS="-Djava.awt.headless=true"

## Type:      integer(0:65535)
## Default:   8080
## ServiceRestart: jenkins
#
# Port Jenkins is listening on.
# Set to -1 to disable
#
JENKINS_PORT="8080"

## Type:      string
## Default:   ""
## ServiceRestart: jenkins
#
# IP address Jenkins listens on for HTTP requests.
# Default is all interfaces (0.0.0.0).
#
JENKINS_LISTEN_ADDRESS=""

## Type:      integer(0:65535)
## Default:   ""
## ServiceRestart: jenkins

```


然后插件中心使用清华源的插件镜像



<https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json>

获取管理员密码

```
cat /var/lib/jenkins/secrets/initialAdminPassword
```

解锁 Jenkins

为了确保管理员安全地安装 Jenkins，密码已写入到日志中（[不知道在哪里？](#)）该文件在服务器上：

```
/var/lib/jenkins/secrets/initialAdminPassword
```

请从本地复制密码并粘贴到下面。

管理员密码



继续


安装插件

新手入门

| | | | | |
|----------------|--------------------------|-------------------------------------|--------------------------------------|--|
| ✔ Folders | ✔ OWASP Markup Formatter | ✔ Build Timeout | ✔ Credentials Binding | ** Trilead API Folders OWASP Markup Formatter ** Oracle Java SE Development Kit Installer ** Struts ** Pipeline: Step API ** Token Macro Build Timeout ** Credentials ** Plain Credentials ** SSH Credentials Credentials Binding ** SCM API ** Pipeline: API Timestampers ** Script Security ** Pipeline: Supporting APIs ** Durable Task ** Pipeline: Nodes and Processes |
| ✔ Timestampers | 🔄 Workspace Cleanup | 🔄 Ant | 🔄 Gradle | |
| 🔄 Pipeline | 🔄 GitHub Branch Source | 🔄 Pipeline: GitHub Groovy Libraries | 🔄 Pipeline: Stage View | |
| 🔄 Git | 🔄 SSH Build Agents | 🔄 Matrix Authorization Strategy | 🔄 PAM Authentication | |
| 🔄 LDAP | 🔄 Email Extension | 🔄 Mailer | 🔄 Localization: Chinese (Simplified) | |

Jenkins 2.244

设置好用户名以后进入系统

 Jenkins

用户列表

构建历史

系统管理

我的视图

Lockable Resources

新建视图

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

欢迎来到 Jenkins!

添加一个节点或者设置一个集群并配置分布式构建。 了解更多。

开始: 创建一个新任务。

添加说明

查找

通知 1 Azkaban 注销

生成页面
2020-7-9 上午10:01:01秒

REST API

Jenkins 2.244

Jenkins 中文社区