CI

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.

CI，Continuous Integration，持续集成，是软件开发过程中一个非常重要的环节，在互联网敏捷开发的过程中，持续集成通常用来进行日常编译和自动化测试，来保证及时发现提交的问题，避免影响项目进度。

通常持续集成的过程包括：

•提交（合并）代码

•编译

•测试

•发布

不同的项目步骤有所不同。

部署CI需要的工具

1. 、Docker ：Docker是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中,然后发布到任何流行的Linux机器或Windows 机器上,也可以实现虚拟化,容器是完全使用沙箱机制,相互之间不会有任何接口。
2. 、Gitlab-runner: GitLab Runner是一个开源项目，用于运行您的作业并将结果发送回GitLab。

部署过程

## Docker Toolbox

在windows10家庭版中不支持Docker，需要安装Docker Toolbox。

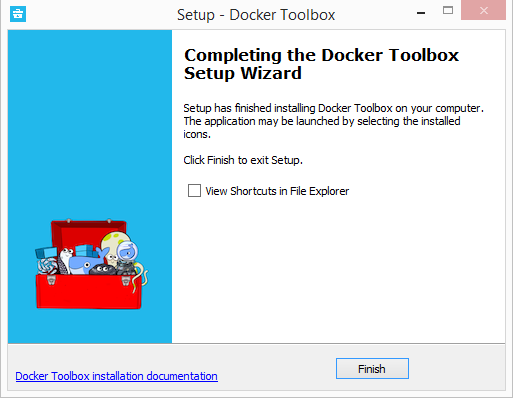
Docker Toolbox包含以下Docker工具：

* Docker CLI客户端，用于运行Docker Engine创建映像和容器
* Docker Machine，因此您可以从Windows终端运行Docker Engine命令
* Docker Compose用于运行docker-compose命令
* Kitematic，Docker GUI
* 为Docker命令行环境预先配置的Docker QuickStart shell
* Oracle VM VirtualBox

## 步骤1：检查版本

要运行Docker，计算机必须具有运行Windows 7或更高版本的64位操作系统。此外，必须确保在计算机上启用了虚拟化。

## 步骤2：安装Docker Toolbox



## 步骤3：验证安装

将Docker QuickStart的目标设为Git安装目录下bin文件夹里的bash.sh。

（如果windows有添加hyper-v，需要在windows功能和cmd里设置关闭，否则会出错）

键入docker run hello-world命令，以验证安装。

如果看见这条信息，表示安装成功：

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker Engine CLI client contacted the Docker Engine daemon.

2. The Docker Engine daemon pulled the "hello-world" image from the Docker Hub. (Assuming it was not already locally available.)

3. The Docker Engine daemon created a new container from that image which runs the executable that produces the output you are currently reading.

4. The Docker Engine daemon streamed that output to the Docker Engine CLI client, which sent it to your terminal.

## Gitlab-runner

在命令行中输入：

docker pull gitlab/gitlab-runner

拉取gitlab-runner镜像，等待完成。

然后再输入：

docker run -d --name gitlab-runner --restart always -v /srv/gitlab-runner/config:/etc/gitlab-runner -v /var/run/docker.sock:/var/run/docker.sock gitlab/gitlab-runner:latest

创建并运行gitlab-runner容器。

在命令行中输入以下命令注册一个新的runner实例:

docker exec -it gitlab-runner gitlab-runner register

之后依次输入gitlab实例URL、token、description、tags、执行者、镜像。

创建好后，就可以在gitlab上看见创建的runner。

再添加.gitlab-ci.yml脚本，就可以使用CI了。