

## 714团队：基础设施与资料整理

### 基础设施

1. 水下机器人
2. 巡检机器人
3. 监控摄像头
4. 工业相机
5. 深度学习服务器
6. 无人飞行器：

### 附录 1：推荐学习资料（更新于2019年11月）

#### 编程开发与软件工程

- C++（编程语言）
- Python（编程语言）
- Git与GitHub（工具）
- Markdown（工具）
- 软件工程（思想）
- WinForm / QT（用户界面实现）
  - WinForm/.NET
  - QT
- Linux（工具）
- MATLAB（工具）
- SketchUP（工具）

#### 图像处理/计算机视觉

- 理论
- 工具
  - OpenCV
  - Halcon
- 学习资源

#### 深度学习&机器学习

- 理论
- 工具

#### 学校及网络资源获取方式

### 附录 2：深度学习服务器搭建与使用

#### 技术方案

- 硬件配置
- 技术方案
- 参考资源/链接

#### 搭建介绍

- 系统
- GPU驱动
- LXD/ZFS安装
  - 安装
  - 配置
    - 网络
- 初始化LXD
- 测试

#### 使用介绍

- 创建容器模板
  - LXC清华源
  - 下载创建
- 配置共享目录和GPU
  - 共享目录设置
- GPU配置
  - 添加GPU
  - 安装显卡驱动
- 配置SSH免密码登录

|            |                 |
|------------|-----------------|
|            | 安装SSH服务         |
|            | 生成RSA文件         |
|            | 测试SSH           |
|            | 配置frp（内网穿透工具）   |
|            | 关于frp           |
|            | 使用方式            |
|            | 服务端配置           |
|            | 客户端配置           |
|            | 在校园网内进行测试       |
| 给用户分配容器    |                 |
|            | 克隆容器            |
|            | 修改hostname      |
|            | 修改SSH密钥文件       |
|            | 修改frp映射端口       |
|            | 交付容器            |
| 拓展         |                 |
|            | LXDUI 可视化界面     |
|            | 资源限制问题          |
|            | 图形界面            |
|            | 驱动的更新           |
|            | 关于镜像自动备份        |
| 备注         |                 |
|            | 容器中无法使用GPU的问题   |
|            | 方案一：重新安装GPU显卡驱动 |
|            | 方案二：重新挂载        |
|            | 注意              |
| 机器视觉项目基本流程 |                 |
|            | 1. 基本流程         |
|            | 2. 确定客户需求       |
|            | 3. 方案设计         |
|            | 4. 软件开发         |
|            | 5. 现场调试         |
|            | 6. 文档交接         |

# 714团队：基础设施与资料整理

## 基础设施

### 1. 水下机器人

水下机器人细节介绍。参考文件：《通用型开架式水下机器人开源手册.pdf》

朱晓坡、侯一兴、黄渊博等已上手。

相关的配套软件：

1. QGC: QGroundControl，开源的地面站控制软件，可用于水下机器人的视频查看和实时控制。
2. 水下巡检与定位上位机系统（团队在研软件）：用于声纳数据的接收，定位解算，视频传输以及缺陷检测的综合系统。
  - o version1.0: 沈晓海
    - 声纳数据接收保存展示，并提供实时解算的数据接口。
    - 2D+3D的定位展示接口，并提供位置接口用于实时展示。
    - 视频流接收、解码、显示和保存的功能实现。

- version2.0: 黄渊博
  - 遥控操作的实现
  - 缺陷检测系统的整合

## 2. 巡检机器人

黄渊博、宋柯正在制作。

上位机平台黄渊博已接手。

相关的配套软件：

1. 巡检机器人系统平台：上海国家电网项目图像处理库+英特尔杯竞赛作品。

## 3. 监控摄像头

海康的多路摄像头，

信息备注：

用户名：admin

密码：qianxin714

录像机采集五路视频（录像机本地可以播放，录像机作为网页服务端也可以播放）

VLC打开RSTP串流：rtsp://admin:qianxin714@169.254.61.108:554/Streaming/Channels/101

chanel: 101 201 401 701 801教程: <https://blog.csdn.net/xiejiaoshu/article/details/71786187>

## 4. 工业相机

两个Baumer(堡盟) 工业相机，多个海康工业相机。另外配备多个镜头。

工业相机的使用需要多了解，多尝试。光源的选择搭配也要通过多尝试来确认。

**光源选择相关资料：**

1. 《灯源种类整理.pdf》
2. 多看淘宝上的光源介绍资料。

**工业相机开发方式：**

1. Halcon的开发方式案例：[采用HALCON机器视觉软件及C#语言检测工件位置的方法](#)
2. Baumer相机的开发方式：Baumer官网的一些有用信息：[工业相机的相关技术信息](#)

文档笔记（Baumer驱动安装后值得注意的文档，介绍了如何利用SDK进行开发）

- Docs
  - Installation\_GuideInstallationGuide\_Win.pdf
    - Installation Guide for Microsoft® Windows®
    - Baumer GAPI SDK v2.10.0
    - 介绍了SDK的开发框架和安装配置，是第一个步骤。
  - Programmers\_Guide
    - ProgrammersGuide\_BaumerFeatures.pdf
      - Programmer's Guide Baumer Features
      - Baumer GAPI SDK v2.10.0
      - 介绍Baumer相机和SDK中，区别于GenTL SFNC标准的功能。有需要可以来这里查询。

- **ProgrammersGuide.pdf**
  - Programmer's GuideBaumer
  - GAPI SDK v2.10.0
  - 编程指导。比较全面，找API就需要参考这一份文档。
- Baumer\_SDK\_ReferenceSDK
  - 参考文档，用于搜索查找信息。
- 指导：
  - GAPI 安装，参考Installation guide。
  - C++/C#，用的是C#方式。
  - 需要了解GAPI的核心概念，参考第三章：Central Idea Behind Baumer GAPI，有助于理解API 接口。了解这几个概念（System, Interface, Device, DataStream, Buffer）
  - 第四章Programming Basics in Baumer GAPI2介绍编程配置。详见C#小节。
    - 需要设置环境变量（安装包形式），如果用的现成的压缩包解压的话，应该可以将Components文件夹下的文件拷贝到Bin即可。
    - 编程最好是通过不同的线程来处理。即采图一个线程，处理一个线程。
  - Baumer Camera Explorer中有ForceIP选项，可以调整IP。
  - 影响datastream效率的是buffer size，这部分需要关注。参考programmingguide 第58页。TLParamsLocked mechanism. 机制
  - 获取缓冲输入后需要转换Image,详见guide 5.5节。具体包括
    - Load image processor object
    - Create Image
    - Transform Image
    - Release Image Objects
    - Release Image Processor

## 5. 深度学习服务器

详见 附录2。

服务器宿主端用户和密码是：ubuntu，hhuuc714。

## 6. 无人飞行器：

型号：DJI M210

热成像相机：DJI XT

相机：Z30

操控方式，万至达已经掌握。

维护问题可以联系：常州赛思网络科技 吴国强 13806123949。

## 附录 1：推荐学习资料（更新于2019年11月）

### 需要注意的几点

1. 各位同学有推荐的相关资源随时进行补充。
2. 清单中资源多，但贪多嚼不烂，注意辨别哪些需要致力研究，哪些仅是随手使用。
3. 除推荐中的工具、书籍等，还应当通过追踪期刊和会议来了解自己领域的最新动态。
4. 文档写于2019年11月，需要大家随时更新维护。

# 编程开发与软件工程

## C++ (编程语言)

C++除了C语言部分，至少要掌握面向对象概念和STL库的使用，而不应仅局限于拿C++写C程序。

- 书籍
  - 基础书籍推荐看《C++ Primer》（不推荐《C++ Primer Plus》）
  - 实用书籍推荐看《Effective C++》
  - 进阶清单链接: <http://bestcbooks.com/recommended-cpp-books/>
- 视频
  - 侯捷系列视频（哔哩哔哩已有搬运）
    - 基础：《C++面向对象开发》
    - 进阶：《STL标准库与泛型编程》
    - 拓展：《C++新标准C++11/14》、《C++内存管理机制》、《C++Startup揭秘》
- 开发工具
  - Visual Studio: Community版本即可。
  - JetBrains CLion: JetBrains系列开发工具可以通过学生邮箱认证。
- 网站推荐
  - C语言中文网 <http://c.biancheng.net/>

## Python (编程语言)

Python作为一门解释型语言，方便快捷。一方面可以当成日常工具使用，另一方面图像处理、深度学习领域有大量开源代码基于Python进行研究。Python是一门实践语言，可以边学边用。 **备注：使用Python3.x版本，不要学Python2.x版本**

- 书籍
  - 入门：《简明Python教程》
  - 进阶：《流畅的Python》
- 视频
  - Coursera课程《Python For Everybody》，覆盖数据结构、网络和数据库和数据可视化等领域
- 开发工具
  - Anaconda (**务必掌握虚拟环境的管理**)
  - Visual Studio Code
  - JetBrains PyCharm: JetBrains系列开发工具可以通过学生邮箱认证。
- 网站
  - Python教程-廖雪峰的官方网站 <https://www.liaoxuefeng.com/wiki/1016959663602400>

## Git与GitHub (工具)

版本管理工具，Git常用于开源项目，SVN常用与团队内部代码管理。SVN可以不学，Git最好掌握，对于项目代码、论文实验代码以及文档版本的管理，能够提升效率。

- Git
  - 《Git 简明指南》 <https://rogerdudler.github.io/git-guide/index.zh.html>
  - Git教程 <https://www.liaoxuefeng.com/wiki/896043488029600>
- GitHub上的[代码管理和团队协作](#) 教程

重点关注以下两个Learning Path：

- [First Day on GitHub](#)

- [First Week on GitHub](#)

## Markdown (工具)

轻量级标记语言，主流的代码托管平台、开源项目的说明文档、技术社区和论坛的问答，均使用Markdown写作。学习成本很低，掌握基本语法即可。

- 《Markdown: 教程》<https://www.markdown.cn/>
- 本地工具推荐：Typora，这份文档就是通过该软件编写的。

## 软件工程 (思想)

软件工程主要指项目开发中涉及编码规范、设计、文档等与技术无直接关系，但又影响到开发质量的合理规范的开发流程。

需要平时多积累。

- 推荐的书籍
  - 《构建之法》
- 推荐的博客
  - 现代软件工程讲义 [源代码管理](#)
  - 现代软件工程讲义 [目录](#)

## WinForm / QT (用户界面实现)

目前为止实验室的PC软件界面开发使用的是C#/.NET配合WinForm，WinForm方便搭建简单的UI界面，但其局限性在于UI可拓展性差，无法搭建美观的界面；

对于仅仅需要界面进行算法调试的，可以使用WinForm搭建。

但C#与C++难以进行高级数据结构的交互；要求.NET环境部署难。目前很少有使用WinForm搭建UI的大型项目。

推荐使用C++/QT进行PC软件界面的开发，UI扩展性强、跨平台、与C++高度结合，不需要再费心思学C#语言。

### WinForm/.NET

- 入门：《C#教程》<http://c.biancheng.net/csharp/>
- 进阶：《Learning Hard C# 学习笔记》

### QT

- 《C++ GUI QT 4编程（第二版）》
- 《QmlBook》<https://github.com/cwc1987/QmlBook-In-Chinese>

## Linux (工具)

掌握Linux的理由如下：1. 大量开源软件和深度学习框架优先支持 2. 与嵌入式系统打交道的最主要方式。深度学习、图像处理等方向的掌握基本使用即可，硬件开发方向的需要深入理解。

Linux可以边学边用，遇到不会的进行查询即可。

- 入门：一般推荐《鸟哥的Linux私房菜》，但其体量大，应当成工具书选择自己需要的进行查阅。
- 推荐使用Ubuntu系统进行Linux初步使用，学会基本的操作命令。

## MATLAB (工具)

MATLAB是经典的仿真工具，提供了大量的函数、绘图工具，也有大量的开源代码和实验都是基于MATLAB，至少要掌握MATLAB的基本使用，看到别人的代码做到心中不慌。比如MATLAB编程风格、如何写脚本和运行、如何查询函数使用等。

- 商业软件都有其详细的使用文档和教程，入门和进阶无需考虑其他市面教材，
- 入门：官网文档《MATLAB 快速入门》
- 进阶：官网文档 <https://ww2.mathworks.cn/help/matlab/index.html> 包括语言基础知识、数学、图形、数据导入和分析、脚本和函数编程、APP构建、高级软件开发等
- 具体方向（图像、视频、音频等）研究则应自行查找相关书籍

## SketchUP (工具)

SketchUP是易于上手的三维绘图工具，在绘制专利、演示等示意图上有帮助。非设计专业，掌握基本使用即可，学会绘制基本元素，利用好模型库中的已有模型。

- 《SketchUp建模新手必读教程合集》 <https://www.asketchup.com/thread-344151-1-1.html>

## 图像处理/计算机视觉

### 理论

- 《数字图像处理》冈萨雷斯著
- 《机器视觉算法与应用》HALCON研发者编著
- 《图像处理，分析与机器视觉》
- 课程推荐：《CS131 Computer Vision: Foundations and Applications》，主要讲解常见算法的原理和实现。
- 书籍推荐（张学武推荐）
  - 清华大学 章毓竟
  - 天津大学 张广军

### 工具

#### OpenCV

开源视觉库，封装了许多算法。

建议看书和查文档API。

- 《OpenCV3编程入门》
- 《Python计算机视觉编程》
- <https://www.pyimagesearch.com/> 该博主提供了一些小项目的教程和代码实现。

#### Halcon

商业工业视觉软件。提供了丰富的工业检测案例，可以拿来参考。安装目录下有详细的使用文档和说明，无需再找其他教程。

- 《manuals》
- 《reference》
- 《solution\_guide》
- 结合《机器视觉算法与应用》一起看。

### 学习资源

<https://github.com/jbhuang0604/awesome-computer-vision> 罗列了与视觉相关的资源

<https://handong1587.github.io/index.html>

## 深度学习&机器学习

### 理论

理论的书看不下去可以先找一些有对应代码的课程或者书籍，先尝试运行。

- 书籍
  - 《机器学习》
  - 《深度学习》
- 课程
  - 机器学习入门：吴恩达《machine learning》（哔哩哔哩）
  - 深度学习入门：吴恩达《深度学习工程师》（网易云课堂）
  - 深度学习进阶：《CS231n: Convolutional Neural Networks for Visual Recognition》
  - 深度学习进阶：<https://www.fast.ai/> 有两门课（哔哩哔哩已搬运），一门注重实践，一门注重前沿

### 工具

- 深度学习框架 目前有许多深度学习框架，TensorFlow、PyTorch、MXnet等。推荐使用PyTorch，方便模型的搭建和调试，更易于学术上进行研究。
  - PyTorch入门：官网文档和《DEEP LEARNING WITH PYTORCH: A 60 MINUTE BLITZ》
  - PyTorch教程：<https://github.com/yunjey/pytorch-tutorial>
- 网站
  - <https://github.com/> 丰富的论文复现代码
  - <https://arxiv.org/> 丰富的计算机领域论文下载（出版和未出版）
  - <https://www.paperweekly.site> 论文追踪
  - Kaggle和天池大数据：深度学习领域竞赛平台，经常会有比赛，通过参加比赛或者学习别人的解决方案来提高自己的实践能力。
- 公众号和微博
  - 机器之心 <https://www.jiqizhixin.com/>
  - 我爱计算机视觉 <https://www.52cv.net/>
  - 爱可可-爱生活 [https://weibo.com/fly51fly?is\\_all=1](https://weibo.com/fly51fly?is_all=1)

### 学校及网络资源获取方式

- 学校数据库 资源大平台，许多非常棒的资源可以从上面获取
  - 知网 国内的学术期刊几乎都可以搜到，下载
  - Springer 出版社，包括期刊文章、会议集和该社的电子书
  - EI 检索网站
  - IEEE 检索各类期刊文章和会议集
- 网络平台
  - <http://sci-hub.cc/>
  - <http://sci-hub.io/>
  - Google Scholar
  - 百度学术 也可以下载一些文档
- 提供代码复现的平台
  - <http://www.cvpapers.com/> 提供计算机视觉顶会的论文，还提供了代码
  - <http://www.gitxiv.com> 基于arXiv和Github的计算机科学协作平台。文章源码同时放出，还可以和原作互动评论



- 电子书籍
  - 外文书籍
    - 查询网络是否有公版书籍
    - 查询Springer 等学校购买版权的网站是否提供下载
    - 查询图灵和异步等电子书社区是否提供电子书
- 实体书籍
  - 查询图书馆是否有相关书籍
  - 买买买，可以报销

## 附录 2：深度学习服务器搭建与使用

### 技术方案

#### 硬件配置

| 指标     | 参数                                |
|--------|-----------------------------------|
| CPU    | Intel至强 E5-2680 v4 *2             |
| GPU    | NVIDIA TITAN V *2                 |
| Memory | 128G                              |
| Disk   | 480G SSD + 8T HDD + (4T RAID HDD) |

#### 技术方案

- 主机系统：Ubuntu 18.04
- 容器管理工具：LXD
- 文件管理系统：ZFS

#### 参考资源/链接

- [搭建多人共用的GPU服务器](#)
- [为实验室建立公用GPU服务器](#)
- [LXD, ZFS and bridged networking on Ubuntu 16.04 LTS+](#)
- [ZFS 与 LXC 与 GPU Passthrough, 以及贵校超算队集群管理](#)
- [搭建公用GPU服务器过程记录](#)
- [搭建实验室公共GPU服务器](#)

### 搭建介绍

#### 方案流程

1. 宿主机安装Ubuntu18.04系统，安装GPU驱动。
2. 安装LXD/ZFS软件并进行配置。
3. 创建容器模板（Ubuntu16.04），包括：网络、GPU驱动、共享目录、SSH登录。
4. 按需分配，克隆容器模板，并做个性化修改。
5. 使用lxdm进行可视化容器管理。

#### 系统

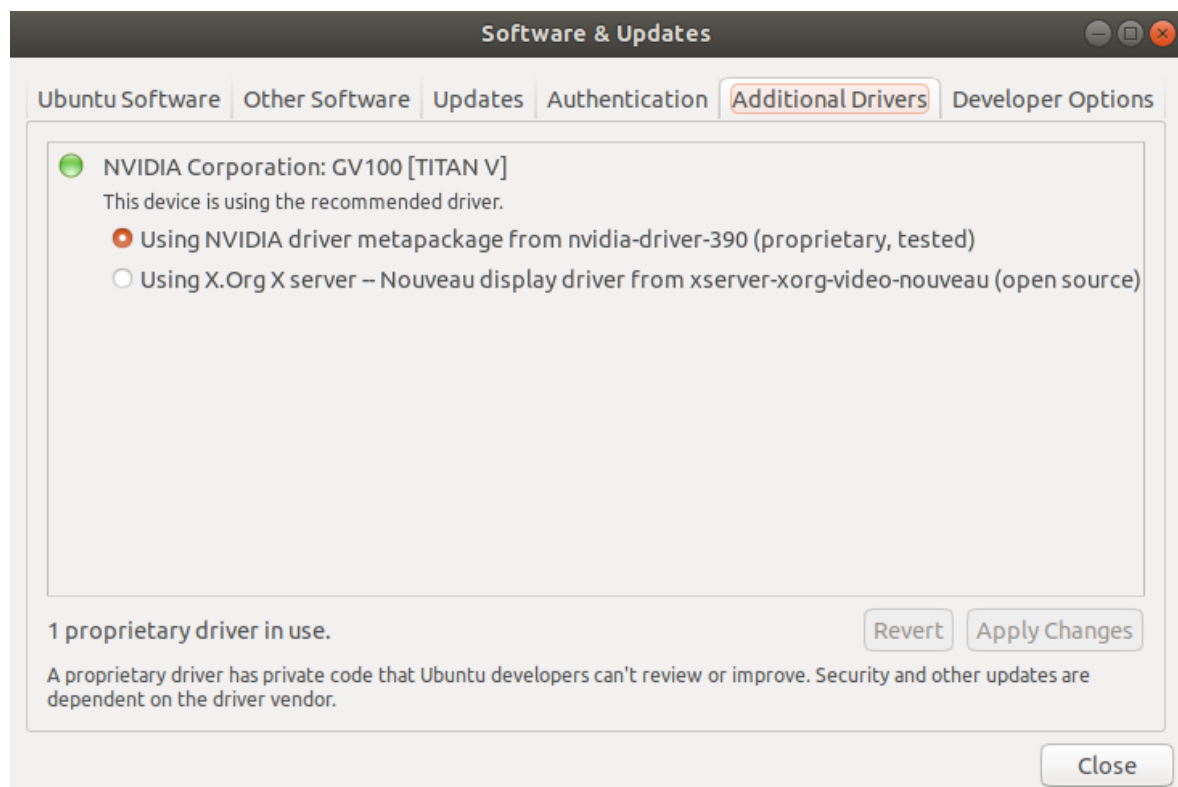
系统选择Ubuntu 18.04。

原因：Ubuntu 16.04下默认下载安装的是LXD2.0，LXD2.0不支持配置GPU到容器中，如果Ubuntu16.04需要配置GPU到容器，则需自行编译LXD3.0。而Ubuntu 18.04默认下载安装的则是LXD3.0的，安装方便。

安装完成后，注意将软件源配置为清华源，提高下载更新速度。具体步骤参考：[Ubuntu 镜像使用帮助](#)

## GPU驱动

往常Linux下的GPU驱动安装是很复杂的，网上NVIDIA Driver教程百花绽放。经过实践发现，装完系统后，使用系统 `软件和更新` 中提供的NVIDIA驱动是可行的，非常方便。



安装完成后，重启，输入 `nvidia-smi` 进行确认，注意显卡驱动版本。

## LXD/ZFS安装

### 安装

- LXD：用于创建和管理容器
- ZFS：用于管理物理磁盘，支持LXD高级功能，负责容器存储。
- Bridge-Utils：用于搭建网桥，负责容器上网。

运行命令进行安装：

```
sudo apt-get install lxd zfsutils-linux bridge-utils
```

### 配置

#### 网络

需要注意的是，不同网络环境下，大家对于上网方式的诉求是不一样的。有些情况是希望用户在自己容器中登录校园网进行上网，有些情况则是上网免费，无所谓个人不个人。不同情况下，对于网络的配置是不同的。

目前我们是这样打算的，宿主机使用我的个人账户登录，每个容器就直接通过宿主机NAT上网，也不需要每个同学登录了，反正费用最后都能报销。对于较大的几十个G的数据集，则建议用户自己下载下来后，通过物理拷贝或者局域网内传输的方式上传到容器。

这种情况的下网络配置是最简单的，因为LXD默认初始化的网络方式就是：宿主机通过DHCP的方式给每个容器分配IP。**如果上网方式跟我的相同的话，可以参考下节的配置。如果不是的话，请留意网络/网桥的配置。**

## 初始化LXD

运行 `sudo lxd init` 进行LXD初始化配置，选项如下图：

```
root@ubuntu:~$ sudo lxd init
Would you like to use LXD clustering? (yes/no) [default=no]: no
Do you want to configure a new storage pool? (yes/no) [default=yes]: yes
Name of the new storage pool [default=default]: lxd_pool
Name of the storage backend to use (dir, zfs) [default=zfs]: zfs
Create a new ZFS pool? (yes/no) [default=yes]: yes
Would you like to use an existing block device? (yes/no) [default=no]: yes
Path to the existing block device: /dev/sda
Would you like to connect to a MAAS server? (yes/no) [default=no]: no
Would you like to create a new local network bridge? (yes/no) [default=yes]: yes
What should the new bridge be called? [default=lxdbr0]: lxdbr0
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]: auto
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]: auto
Would you like LXD to be available over the network? (yes/no) [default=no]: no
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]: yes
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: no
```

- LXD Clustering: 不需要
- new storage pool: 需要创建一个存储池
- Name of storage pool: 给存储池命名
- storage backend: 存储后端，使用ZFS
- Create a new ZFS pool: 需要创建一个ZFS池
- use an existing block device: Yes Path to block device: 使用已有的磁盘分区用于ZFS的存储后端。细节可以参考[Be aware](#)
- MAAS server?: 不知道是啥，不需要
- new local network bridge?: 需要，我只需要使用LXD默认的网桥即可。
- new bridge be called: 给网桥命名
- IPv4: 默认auto
- IPv6: 默认auto
- LXD available over the network? : 默认no
- stale cached? : 默认yes
- YAML printed? : 打印信息，yes/no都行，原谅最后手抖的yno

## 测试

`sudo zpool list lxd` 查看ZFS的后端存储池。

`sudo lxc info` 查看LXD的配置信息。

`sudo lxc profile show default` 查看默认容器配置。

`sudo lxc list` 查看容器列表。

更多命令可以参考[LXD Documentation](#)

到这里就已经安装完这套方案的基础设施了，接下来的步骤可以算是使用指南

## 使用介绍

### 创建容器模板

LXC清华源

使用lxc清华源可以加速镜像的下载。 [LXC Images 镜像帮助](#)

运行如下命令配置：

```
# 创建一个remote链接，指向TUNA镜像站。
sudo lxc remote add tuna-images https://mirrors.tuna.tsinghua.edu.cn/lxc-images/
--protocol=simplestreams --public
```

```
# 查看镜像列表，寻找合适的镜像的FINGERPRINT，用于下载
sudo lxc image list tuna-images:
```

## 下载创建

```
# FINGERPRINT是镜像的指纹，在上条命令下查找，ContainerTemplateName为容器模板名称，自己定义。
sudo lxc launch tuna-images:<FINGERPRINT> <ContainerTemplateName>

# 举例，创建一个名为xenialTemplate的容器。在lxc清华源中24be957c5e9f是Ubuntu16.04的fingerprint
sudo lxc launch tuna-images:24be957c5e9f xenialTemplate
```

运行 `sudo lxc list` 进行容器列表查看。

运行 `sudo lxc exec <ContainerTemplateName> bash` 可进入容器的root用户下 bash。

可以使用 `su ubuntu` 或 `sudo su` 进行用户切换。

如果容器下出现 `sudo: no tty present and no askpass program specified` 的问题。

解决方式：

创建 `/etc/sudoer.d/ubuntu` 文件，编辑内容为

```
ubuntu ALL=(ALL) NOPASSWD:ALL
```

## 配置共享目录和GPU

### 共享目录设置

设置共享目录来实现宿主机与容器之间的文件传输，以及容器与容器之间的文件传输。

```
# 设置键值
sudo lxc config set <ContainerTemplateName> security.privileged true
# 设置共享目录，其中shareName为虚拟的设备名称，lxd会虚拟出该设备并导通接通两者共享目录。
# path1为宿主机下共享目录路径，path2为容器下共享目录路径
sudo lxc config device add <ContainerTemplateName> <shareName> disk source=
<path1> path=<path2>
```

## GPU配置

### 添加GPU

```
# 为容器添加所有GPU：
sudo lxc config device add <ContainerTemplateName> gpu gpu
# 添加指定GPU：
sudo lxc config device add <ContainerTemplateName> gpu0 gpu id=0
```

## 安装显卡驱动

还记得之前记录的宿主机显卡驱动的版本号码，根据版本号去官网下载驱动文件，通过共享目录传至容器中。

例如，宿主机中的NVIDIA Driver Version为390.77，则下载 `NVIDIA-Linux-x86_64-390.77.run`。

安装显卡驱动。

```
# 进入容器
sudo lxc exec <ContainerTemplateName> bash
# 安装驱动
sudo sh NVIDIA-Linux-x86_64-xxx.xx.run --no-kernel-module
```

运行 `nvidia-smi` 进行确认。

## 配置SSH免密码登录

SSH免密登录网上资料一大把，有不明白的地方可以网上找。

### 安装SSH服务

```
# 装OpenSSH服务
sudo apt install openssh-server

# 启动SSH服务
sudo service ssh start

# 查看SSH服务状态
sudo service ssh status
```

### 生成RSA文件

```
# 进入SSH目录
cd ~/.ssh

# 生成RSA
ssh-keygen -t rsa

# 复制公钥内容到authorized_keys
cat id_rsa.pub >> authorized_keys

# 重启SSH服务
sudo service ssh restart
```

### 测试SSH

容器终端运行 `ifconfig`，确认容器IP和网段，这些是LXD自动分配的。如模板容器的IP为10.135.139.83。

宿主机终端运行 `ifconfig`，查看宿主机在网桥lxdbr0下的IP。如宿主机在该网段下的IP为10.135.139.1。

通过共享目录将SSH秘钥 `id_rsa` 文件拷贝到宿主机，宿主机运行如下命令登录容器：

```
# 给容器SSH密钥文件合适权限
sudo chmod 400 id_rsa

# 命令行登录SSH
ssh -i id_rsa ubuntu@10.135.139.83
```

成功后，进行下一步frp的设置（端口转发）。

## 配置frp（内网穿透工具）

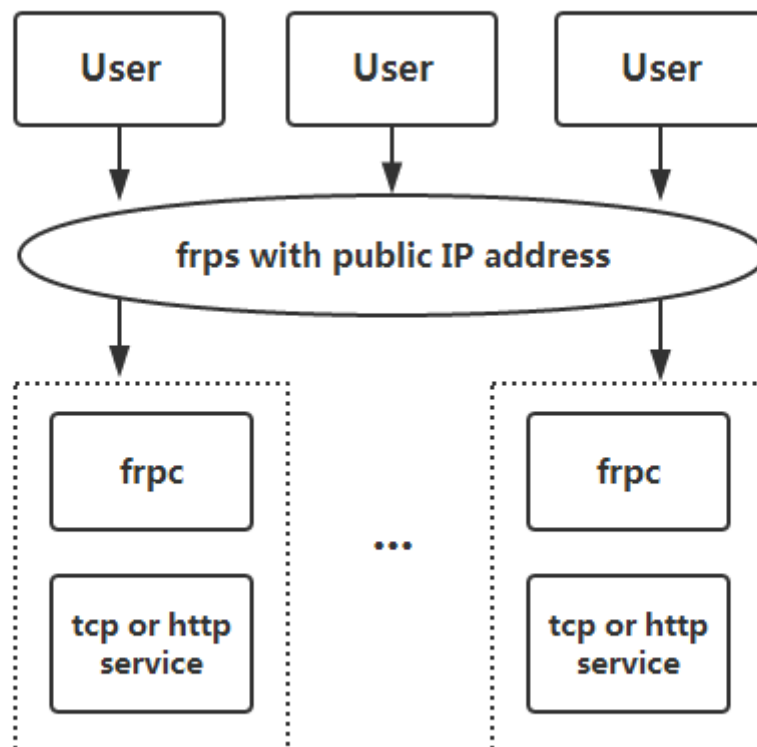
### 关于frp

根据上面的网络配置，可以看到，每个容器其实是处于宿主机构建出来的小型局域网内的，并不暴露在校园网内，也就是说用户从校园网是无法直接访问容器的。

一般来讲，要做的就是宿主机上做端口转发了。我使用的是内网穿透工具frp，配置和使用都很简单。

架在宿主机上，我们可以通过校园网在校园内访问容器；架在公网服务器上，我们就可以在家访问容器。

- [frp gitRepo](#)
- [frp 中文文档](#)



### 使用方式

在 [frp版本发布页面](#) 下载linux的文件，如 `frp_0.21.0_linux_amd64.tar.gz`。解压后文件如下：

- frps: 服务端执行文件
- frps\_full.ini: 服务端参数参考
- frps.ini: 服务端参数文件
- frpc: 客户端执行文件

- frpc\_full.ini: 客户端参数参考
- frpc.ini: 客户端参数文件

## 服务端配置

服务端文件放置在宿主机中，宿主机的frps.ini可供参考，更多参数请看官方文档。

```
[common]
bind_port = 7000

# 限制端口，要对端口做好管理
allow_ports = 2000-3000
```

宿主机执行 `./frps -c frps.ini` 开启服务。

## 客户端配置

客户端文件放置在用户容器中，容器的frpc.ini可供参考，更多参数请看官方文档。

```
[common]
server_addr = 10.135.139.1 # 网段下宿主机的IP
server_port = 7000 # 服务端的bind_port

[ssh-template] # 一个转发实例，注意每个实例名称都不能相同，括号中的名称自定义
type = tcp
local_ip = 127.0.0.1
local_port = 22 # 容器本地SSH访问端口
remote_port = 2011 # 宿主机映射端口
```

容器执行 `./frpc -c frpc.ini` 连接服务。

## 在校园网内进行测试

如上步骤，在已知宿主机校园网IP和容器SSH的映射端口之后，就可以在校园网内访问容器了。

## 给用户分配容器

创建了容器模板之后，就可以按需给用户分配容器了。

## 克隆容器

```
# 克隆容器 参数一为模板容器名称，参数二为目标容器名称
sudo lxc copy <ContainerTemplateName> <newContainerName>

# 运行新容器
sudo lxc start <newContainerName>

# 进入新容器bash
sudo lxc exec <newContainerName> bash
```

## 修改hostname

克隆的容器还保留着模板的hostname，看起来令人不悦。快速修改hostname的步骤如下：

```
# 进入新容器bash
sudo lxc exec <newContainerName> bash

# 将旧的hostname 改成新的
sudo vim /etc/hostname

# 将旧的hostname改成新的，在127.0.0.1后面
sudo vim /etc/hosts

# 重启生效
sudo reboot
```

## 修改SSH秘钥文件

克隆的容器还保留着模板的SSH `id_rsa` 文件，要是大家都用一份`id_rsa`文件访问不同的容器，就太扯了。所以要生成新的 `id_rsa` 文件。快速修改`id_rsa`文件步骤如下：

```
# 进入新容器bash
sudo lxc exec <newContainerName> bash

# 注意切换成ubuntu用户
su ubuntu

# 进入SSH目录
cd ~/.ssh/

# 生成新的id_rsa文件
ssh-keygen -t rsa

# 复制公钥内容到authorized_keys
cat id_rsa.pub >> authorized_keys

# 重启SSH服务
sudo service ssh restart

# 记得将新生成的id_rsa文件拷贝到外部，共SSH访问使用
```

## 修改frp映射端口

道理很简单，要能访问不同的容器，就要将不同容器的SSH 22端口映射到宿主机不同的端口上。

直接修改容器中的`frpc.ini`，将转发实例中的括号名和`remote_port`。

然后容器执行 `./frpc -c frpc.ini` 连接服务即可。

## 交付容器

把容器的SSH秘钥文件、SSH访问IP和SSH访问端口交付给同学即可。

**Tips** 记得交付之前，调用LXD的快照（snapshot）进行初始版本的备份，免得后面弄砸了，又要重新配置。

执行 `sudo lxc snapshot <ContainerName>` 进行快照。

## 拓展

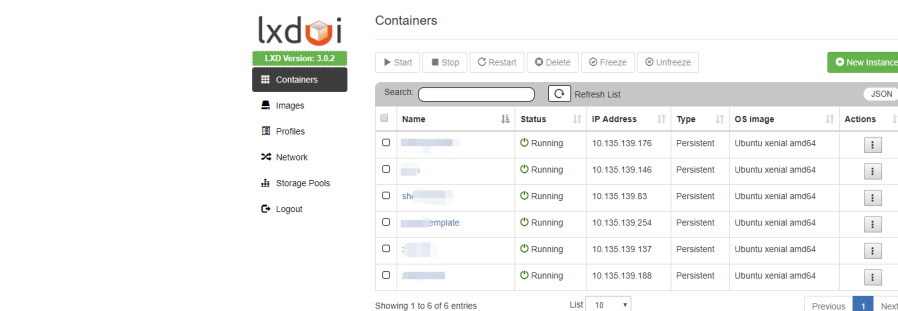
### LXDUI 可视化管理界面



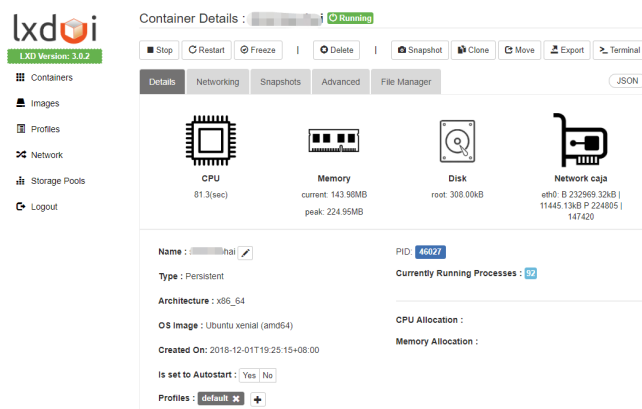
LXDUI是一个LXD/LXC的Web UI工具，支持LXD/LXC的一些基本操作。

具体使用请参考：[lxdui GitRepo](#)

效果如下：



LXDUI v2.1.2  
Copyright © 2018 AdaptiveScale, Inc.



LXDUI v2.1.2  
Copyright © 2018 AdaptiveScale, Inc.

## 资源限制问题

目前是公平的给大家所有的硬件访问权限，后面如果涉及到资源拥挤的话，不可避免需要对每个人的资源进行限制，资源限制请参考：[LXD 2.0 系列（四）：资源控制](#)

## 图形界面

目前我还没有实现图形界面，比较担心的是大家都用图形界面的话，带宽会不会被占用太多。

## 驱动更新

宿主机的软件一般不用更新，毕竟能够维持容器运行即可，所有的操作反正都在容器中进行。

但偶尔难免会出现打开终端双手不受控制的输入update和upgrade的情况，一般情况都还行，但是万一更新显卡驱动，就有糟心事了。

但是别慌，如果由于宿主机更新显卡驱动，导致容器显卡没法用了，就按照之前的安装容器显卡驱动一样，下载和宿主机显卡驱动版本相同的驱动文件，拷到容器内，无内核安装即可。

## 关于镜像自动备份

使用固然方便，如果能替大家实现定时备份容器的镜像就更好了。可以弄崩溃了恢复原来的备份即可。

解决方案参考：

[LXD Automatic snapshotting](#)

[lxdsnap](#)

需要注意的地方就是，关注一下代码和定时设置：定时间隔？哪些要保存？哪些过期删除？千万不要误删了重要的备份节点。

## 备注

### 容器中无法使用GPU的问题

如果遇到断电或者重启，有时候容器虽然挂载了GPU，但仍会遇到无法使用GPU训练的问题，这个时候可以尝试重新挂载GPU。

#### 方案一：重新安装GPU显卡驱动

当没有别的办法来解决的时候，重新安装显卡驱动也不是很麻烦，记得提前备份好显卡驱动的版本号或者备份好显卡驱动文件。遇到重装的时候，直接装就OK。注意主机GPU版本要与容器GPU版本保持一致。

#### 方案二：重新挂载

兴冲冲的装好环境，发现TensorFlow无法使用显卡，原因是宿主机没有/dev/nvidia-uvm设备，需要通过以下命令挂载设备：

```
# 宿主机挂载
/sbin/modprobe nvidia-uvm
D=`grep nvidia-uvm /proc/devices | awk '{print $1}'`
mknod -m 666 /dev/nvidia-uvm c $D 0
```

```
# 先卸载
lxc config device remove yourContainerName nvidia-uvm
# 再挂载
lxc config device add yourContainerName nvidia-uvm unix-char path=/dev/nvidia-uvm
```

## 注意

目前，该套方案运行较为稳定，宿主机端需要操作的是用户容器的创建、删除等操作，即上述教程中的使用部分，搭建部分中的内容切勿随意尝试，避免造成不可逆的影响，同时切勿随意进行重装服务器系统等操作。

## 机器视觉项目基本流程

### 1. 基本流程

确定客户需求、方案设计、软件开发、现场调试、文档交接。

在实际项目中，各个流程可能互相耦合，不过整体流程是基本明确的，整理后如下图。



## 2. 确定客户需求

项目开始，需要准确、详细地了解客户需求，这个过程需要多次现场考察、反复与客户沟通，才能落实客户需求。主要确定项目的应用类型、节拍要求、精度要求、安装空间、光照环境、通讯接口等内容。

**应用类型：**确定视觉应用类型（测量、识别、检测、引导定位），了解产品表面状态、外形尺寸等影响视觉检测因素的变化情况，初步评估能否满足需求

**节拍要求：**客户对生产效率方面的要求，量化视觉检测步骤时间

**精度要求：**针对各检测功能点及客户生产要求量化视觉检测精度

**安装空间：**确认现场环境对视觉系统的安装是否有限制

**光照环境：**确认现场环境是否有强光、日光干扰等特殊影响

**通讯接口：**确认现场与视觉系统配合的数据传输接口类型、I/O接口类型等

## 3. 方案设计

视觉系统是一个各部分互相配合的有机整体，并不是简单的组合，所以一个项目的方案设计关乎着整个项目的成败，从初步方案，到最终方案，以及中间经历的各个版本，需要整个团队共同评审，才能敲定最终方案。整体方案内容主要包括需求分析、视觉硬件设计、视觉软件设计、可行性验证、开发计划。

**需求分析：**整理客户关键需求，并分析需求可行性

**视觉硬件设计：**包括视觉系统平台、相机、镜头、光源的选择

**视觉软件设计：**采用第三方视觉软件，抑或自行开发视觉处理软件

**可行性验证：**搭建软硬件环境，初步测试能否满足客户需求

**开发计划：**罗列项目开发计划，模块化项目节点，跟进项目进度

另外，一个完整的项目应包括机械、电气、视觉等其他部分，以上只是简单陈述下视觉方案的设计内容，而最终呈现给客户的完整项目方案还应包括机械设计、电气设计。

## 4. 软件开发

软件开发主要包括人机交互界面、底层算法，测试运行。

**人机交互界面开发：**简单易用、处理结果直观显示；落实软件框架，功能化软件模块；软件框架多采用生产者/消费者模式，功能模块一般包括图像采集模块、算法处理模块，数据保存模块，通讯模块等。

**底层算法开发：**落实算法处理工具（Halcon、OpenCV、NI Vision等）；开发算法处理流程；生成动态库.DLL

**测试运行：**模拟现场出现的各种情况，测试软件算法的稳定性、鲁棒性。

## 5. 现场调试

现场调试是一个比较繁琐的过程，主要体现在调试过程中的不确定性因素较多，例如环境光的影响、机械振动的影响、硬件工作的稳定性等。主要流程包括设备安装、模块调试、系统联调、自动运行。

**设备安装：**运动部件安装；相机、镜头、光源安装；视觉系统内部线缆附件走线；视觉控制器、光源控制器安装；外部通信、I/O线缆走线等；

**模块调试：**相机功能调试（触发拍照等）；工件检测特征视觉参数调试（相机参数、镜头参数、光源位置和亮度等）；外部通讯调试等；

**系统联调：**调试完整视觉程序；正常生产检测调试等；

**自动运行：**开机自动运行；

## 6. 文档交接

需要与客户进行文档交接时，说明已进入项目尾部，此时应编写操作文档并进行现场培训。

**操作手册：**软件基本操作；常见问题及解决方法；

**现场培训：**项目工作流程；软件操作；问题解决步骤；