

MuxiStudio k3s 搭建

<https://zxc0328.github.io/2019/06/04/k3s-setup/>

老板的 blog，先看看。

主要操作：

已有条件：

1. 几个阿里云子账号，有管理 ECS 的权限，每个子账号一般有一个 ECS。
2. 一个七牛云 bucket 并且搞好了自定义域名。

开始搭建时需要保存原来集群 `/root/` 下的配置文件，例如 k3s-test cluster 下的，建议使用 tar+gzip 打包后 scp 到本地。然后可以上传一份到我们的七牛云上面。

格式化每个 ECS 系统为 **CentOS 7.6**，密码为 **Muxistudio304**。

然后正式开始：

1. 建立云企业网，同时将每个 ECS 的安全组先全部打开。（阿里云有一个安全组的概念，也就是一个防火墙，默认只开启了几个公网端口供我们访问。）
2. 任意选择一个主机作为 master 节点，运行 **修改后的 install.sh**。
3. 从 master 节点找到 master 的 IP 和通信鉴权的 Token 在其他 node 上作为环境变量再运行 install.sh。

注意事项：

1. 云企业网的介绍文档：<https://help.aliyun.com/product/59006.html>

阿里云如今的 VPC 网络是二层隔离的，我们没有办法搞到一堆二层互通的机器，用云企业网可以让网络做一些隔离并且比直接工网 IP 通信会快一些。

有概率会遇到企业网内的主机 IP 段冲突的问题，可以提工单解决。

2. 安装脚本获取镜像过慢 timeout。

我们暂时的解决方案是：

1. 下载所需的 binary 到本地并上传到 七牛云 OSS 。
2. 修改官方的 install.sh 中的镜像地址。

install.sh 我已经修改过，只需要下载镜像，并上传到七牛云上。但是有可能 install.sh 会更新，建议自己修改。

七牛云空间已经创建，为 k3s-binary。绑定了域名 k3soss.muxixyz.com

七牛云需要放几个文件，除下面几个还有 老板文档中的 `k3s-airgap-images-amd64.tar`。

<input type="checkbox"/>	文件名	文件类型	存储类型	文件大小	最后更新	操作
<input type="checkbox"/>	install.sh	application/x-shellscript	标准存储	20.46 KB	2020-03-04 17:13:18	详情 更多
<input type="checkbox"/>	k3	application/x-executable	标准存储	49.86 MB	2020-04-21 23:44:49	详情 更多
<input type="checkbox"/>	k3s-ci-builds	application/xml	标准存储	272.03 KB	2020-03-04 16:58:56	详情 更多
<input type="checkbox"/>	sha256sum-amd64.txt	text/plain	标准存储	164 B	2020-03-04 16:56:03	详情 更多

3. 修改 install.sh

偷懒，我直接在 install.sh 的开头加了下面几行，这是所有 node 都需要的。

检测下有没有 wget，没有 yum install 一下。

```
echo "===== "
echo "move airgap-image"
wget http://k3soss.muxixyz.com/k3s-airgap-images-amd64.tar
sudo mkdir -p /var/lib/rancher/k3s/agent/images/
sudo cp ./k3s-airgap-images-amd64.tar /var/lib/rancher/k3s/agent/images/
export INSTALL_K3S_SKIP_START=true
```

然后修改下面的 URL

```
GITHUB_URL=http://k3soss.muxixyz.com
STORAGE_URL=http://k3soss.muxixyz.com/k3s-ci-builds
DOWNLOADER=
```

官方的下载 URL 有很多版本所以会有更多 path

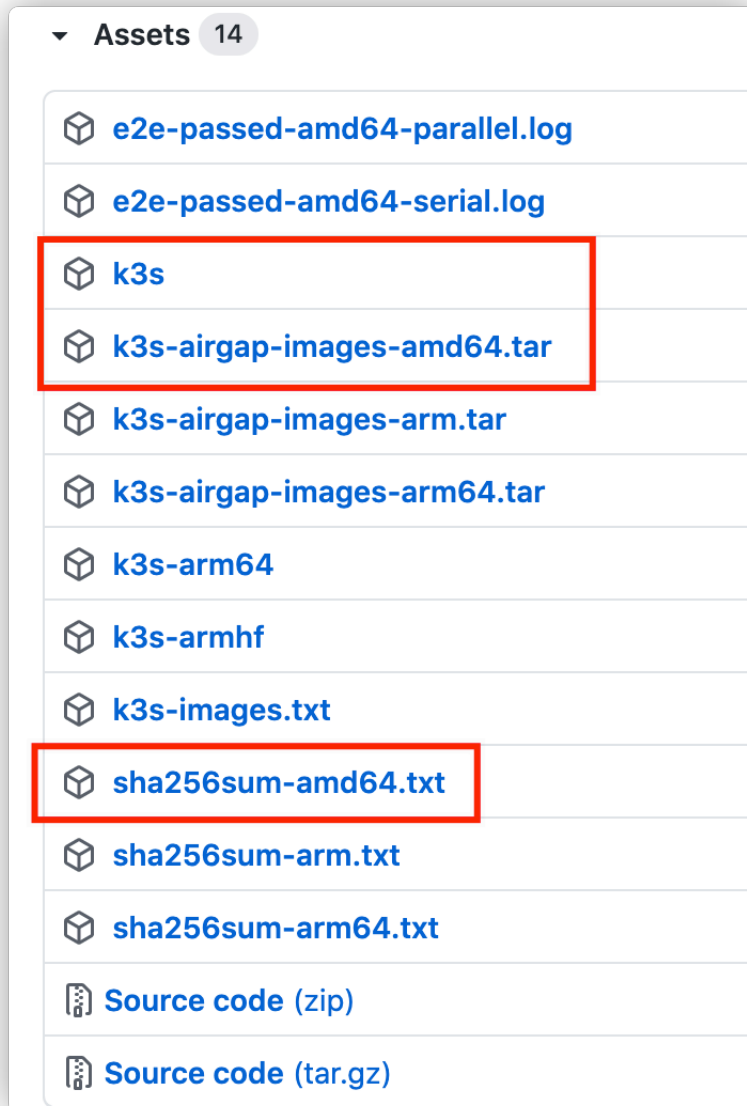
```
# --- download binary from github url ---
download_binary() {
    if [ -n "${INSTALL_K3S_COMMIT}" ]; then
        BIN_URL=${STORAGE_URL}/k3s${SUFFIX}-${INSTALL_K3S_COMMIT}
    else
        BIN_URL=${GITHUB_URL}/download/${VERSION_K3S}/k3s${SUFFIX}
    fi
    info "Downloading binary ${BIN_URL}"
    download ${TMP_BIN} ${BIN_URL}
}

# ----- 修改后 -----
# --- download binary from github url ---
download_binary() {
    if [ -n "${INSTALL_K3S_COMMIT}" ]; then
        BIN_URL=${STORAGE_URL}/k3s${SUFFIX}-${INSTALL_K3S_COMMIT}
    else
        BIN_URL=${GITHUB_URL}/k3s
```

```
fi
info "Downloading binary ${BIN_URL}"
download ${TMP_BIN} ${BIN_URL}
}
```

当然也可以像官方的 GitHub 一样搞一个层次架构来同步所有 image。
可以摸索一下。

建议同步所有 amd64 的 GitHub release 内容。 <https://github.com/rancher/k3s/releases>



以及一个不能忘记的 `k3s-ci-builds`。 <https://storage.googleapis.com/k3s-ci-builds>