onlyoffice部署集成

——张帅杰

# 前言

## Onlyoffice是什么？

是一个document server文档服务器；为用户提供word，excel电子表格和ppt演示在浏览器中预览和编辑功能。

## Onlyoffice能够解决什么问题？

能够解决系统的在线预览和编辑文档的问题（使用较多的是对word文件的预览和编辑）。

结合zip方式实现嵌入式word（已在工程建设审批系统engineering中测试出一套完整方法，具体用于生产还得持续完善）。

结合旧版fileCenter实现文件的预览和保存。

## 支持onlyoffice的浏览器有哪些？

谷歌浏览器，ie浏览器（10版本或者是10版本以上）；

## 使用onlyoffice的好处？

适用范围更广，之前旧版本的activex插件的方式支持ie浏览器，onlyoffice能够覆盖其无法支持的谷歌浏览器。

不需要依赖activex插件（更加轻量）。

使用docker进行管理，更方便的移植部署，且不对环境有依赖（docker的优点）。

## Onlyoffice的不足之处？

不支持低版本的ie浏览器（9版本以及以下版本均不支持）。

# 1 准备环境

## 1.1 安装Linux服务器

安装Linux服务器（当前文档使用的是Centos7版本），需要内核版本为3.10以上，查看内核版本命令：cat /proc/version，如下图1.1：

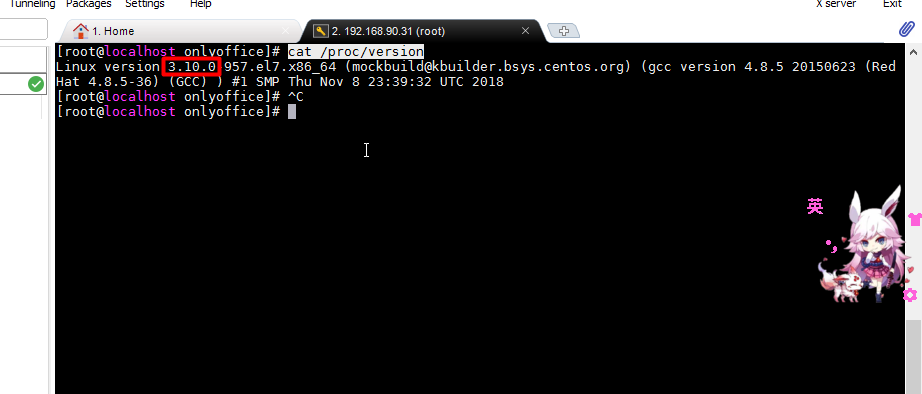


图 1.1

## 1.2 安装docker开源容器引擎

安装docker开源容器引擎。分为在线安装，以及离线安装。

### 1.2.1 在线安装

参考网址：<https://www.runoob.com/docker/centos-docker-install.html>

前提条件：系统为64位、系统内核版本为 3.10 以上

步骤如下：

使用 yum 安装（CentOS 7下）

Docker 要求 CentOS 系统的内核版本高于 3.10 ，通过 **uname -r**命令查看你当前的内核版本

[root@runoob ~]# uname -r

https://www.runoob.com/wp-content/uploads/2016/05/docker08.png

安装 Docker

从 2017 年 3 月开始 docker 在原来的基础上分为两个分支版本: Docker CE 和 Docker EE。

Docker CE 即社区免费版，Docker EE 即企业版，强调安全，但需付费使用。

本文介绍 Docker CE 的安装使用。

移除旧的版本：

$ sudo yum remove docker \

docker-client \

docker-client-latest \

docker-common \

docker-latest \

docker-latest-logrotate \

docker-logrotate \

docker-selinux \

docker-engine-selinux \

docker-engine

安装一些必要的系统工具：

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

添加软件源信息：

sudo yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

更新 yum 缓存：

sudo yum makecache fast

安装 Docker-ce：

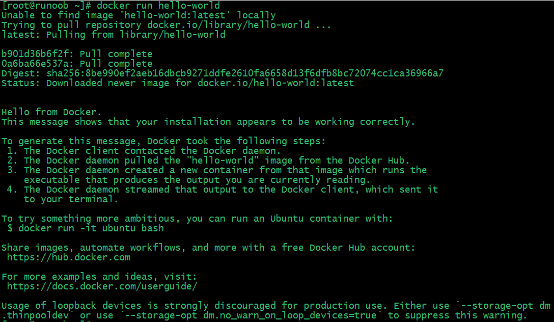
sudo yum -y install docker-ce

启动 Docker 后台服务

sudo systemctl start docker

测试运行 hello-world

[root@runoob ~]# docker run hello-world



由于本地没有hello-world这个镜像，所以会下载一个hello-world的镜像，并在容器内运行。

### 1.2.2 离线安装docker

参考网址：https://www.cnblogs.com/luoSteel/p/10038954.html

将文件：docker-18.06.1-ce.tgz（文件见附件）上传至Linux服务器（当前文档为Centos7版本）。

解压压缩文件夹，命令：tar -xvf docker-18.06.1-ce.tgz。

将解压出来的docker文件内容移动到 /usr/bin/ 目录下，命令：cp docker/\* /usr/bin/。

将docker注册为service，命令：vi /etc/systemd/system/docker.service。

将附件中docker.service.txt中的文本内容加到docker.service中并保存

#添加文件权限并启动docker，命令：chmod +x /etc/systemd/system/docker.service。

#重载unit配置文件，命令：systemctl daemon-reload。

#启动Docker，命令：systemctl start docker。

#设置开机自启，命令：systemctl enable docker.service。

#查看Docker状态，命令：systemctl status docker。

#查看Docker版本，命令：docker version。

当你看到Client以及Server的时候代表安装完成，如下图1.2



图 1.2

# 2 docker中运行onlyoffice镜像

将附件中的onlyoffice-gtmap-1.0.1.tar（文件见附件）上传至Linux服务器。

Docker将压缩文件读取为镜像，命令：docker load -i onlyoffice-gtmap-1.0.1.tar。

Docker运行onlyoffice镜像，命令：docker run -d -p 9000:80 onlyoffice/gtmap:1.0.1。80为容器中的端口，映射到Linux服务器的9000端口，对应的Linux服务器需要开放9000端口的访问权限。

查看容器运行日志，命令：docker logs -f -t --tail 行数 容器名。

# 3 onlyoffice使用

## 3.1 查看功能

以打开html页面的方式使用onlyoffice的查看功能。对应的html页面模板见附件onlyoffice.html。

以下为提供的详细的前台页面配置参数：

1. **<**@com.html title="onlyoffice"**>**
2. **<style>**
3. html,body{height:100%;}
4. **</style>**
6. **<div** id="placeholder"**></div>**
7. **<script** type="text/javascript" src="http://172.24.24.89:9000/web-apps/apps/api/documents/api.js"**></script>**
8. **<script** type="text/javascript"**>**
9. //ps：172.24.24.89为Linux服务器的ip；9000为Linux服务器映射出的端口；web-apps/apps/api/documents/api.js为固定写法
10. var config = {
11. //文档的类型：text为word；spreadsheet为Excle；presentation为ppt
12. documentType: 'text',
13. //文档的详细信息
14. "document": {
15. //页面标题栏内容，默认为：南京国图
16. "title": "南京国图",
17. //工程中文档的地址【ps：需要onlyoffice容器所在的Linux服务器能够访问到172.24.24.98这个地址，不能写127.0.0.1地址】
18. "url": "http://172.24.24.98:9011/resp/2c9898e26b01c3f7016b0bcafddc0065/docx",
19. //文档后缀名，word为doc
20. "fileType": "doc",
21. //key用作docker中的onlyoffice刷新标识，key值为uuid；234sd78wd77dfs778dfae7882为随机数
22. "key": "234sd78wd77dfs778dfae7882"
23. },
24. "editorConfig": {
25. //文档的操作类型：view：预览；edit：编辑
26. "mode": 'edit',
27. //语言：默认为zh-CN
28. "lang": "zh-CN",
29. //保存方法回调的地址
30. "callbackUrl": "http://192.168.90.140/engineering/onlyoffice/save\_file?nodeId=testid"
31. }
32. };
33. var docEditor = new DocsAPI.DocEditor("placeholder",config);
34. **</script>**
35. **</**@com.html**>**

效果如下，图3.1。



图 3.1

## 3.2 保存功能

需要设置的前台html页面为editorConfig的callbackUrl属性。该属性定义了onlyoffice点击保存之后回调的方法（系统内的方法），以下为举例内容。

前台配置的回调地址参数（详见3.1中“详细的前台页面配置参数”中的“callbackUrl”参数）：

1. http://192.168.90.140/engineering/onlyoffice/save\_file?nodeId=testid

对应后台的处理方法：

1. @RequestMapping("/save\_file")
2. @ResponseBody
3. **public** **void** saveFile(HttpServletRequest request, HttpServletResponse response) **throws** IOException {
4. NodeService nodeService = (NodeService) Container.getBean("NodeService");
5. FileService fileService = (FileService) Container.getBean("FileService");
6. //获取回调方法的参数
7. Integer nodeId = Integer.parseInt(request.getParameter("nodeId"));
8. //获取文件节点
9. Node fileNode = nodeService.getNode(nodeId);
10. //获取文件父节点
11. Node parentNode = nodeService.getParentNode(nodeId);
12. //使用writer作为返回值使用
13. PrintWriter writer = response.getWriter();
14. String body = "";
15. **try** {
16. Scanner scanner = **new** Scanner(request.getInputStream());
17. scanner.useDelimiter("\\A");
18. body = scanner.hasNext() ? scanner.next() : "";
19. scanner.close();
20. } **catch** (Exception scannerException) {
21. writer.write("get request.getInputStream error:" + scannerException.getMessage());
22. **return**;
23. }
24. **if** (body.isEmpty()) {
25. writer.write("empty request.getInputStream");
26. **return**;
27. }
28. JSONObject jsonObj = JSON.parseObject(body);
29. **int** status = (Integer) jsonObj.get("status");
30. **int** saveStatus = 0;
31. //依据status返回值，进行保存文件
32. **if** ((2 == status) || (6 == status)) {
33. //if (false) {
34. String downloadUri = (String) jsonObj.get("url");
36. URL url = **new** URL(downloadUri);
37. HttpURLConnection connection = (HttpURLConnection) url.openConnection();
38. //获取onlyoffice上的缓存文件，获取为InputStream格式的数据
39. InputStream stream = connection.getInputStream();
40. //调用老版本文件中心中的文件上传方法
41. fileService.uploadFile(stream, parentNode.getId(), fileNode.getName(), **null**, **true**, **false**);
42. connection.disconnect();
43. }
44. writer.write("{\"error\":" + saveStatus + "}");
45. }