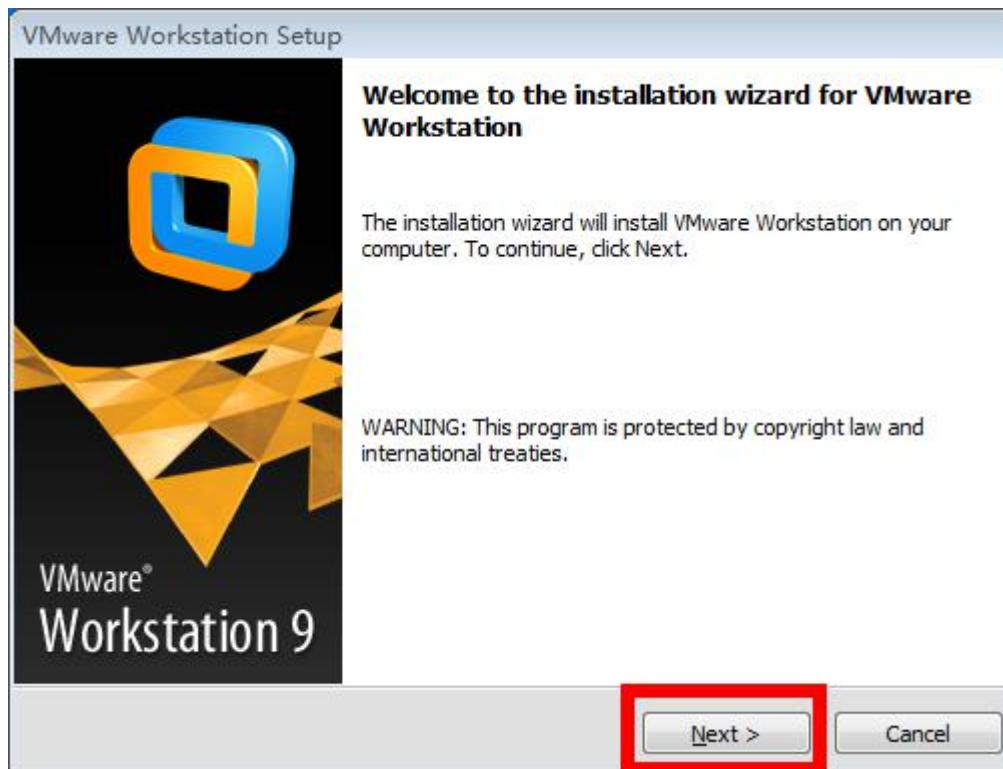


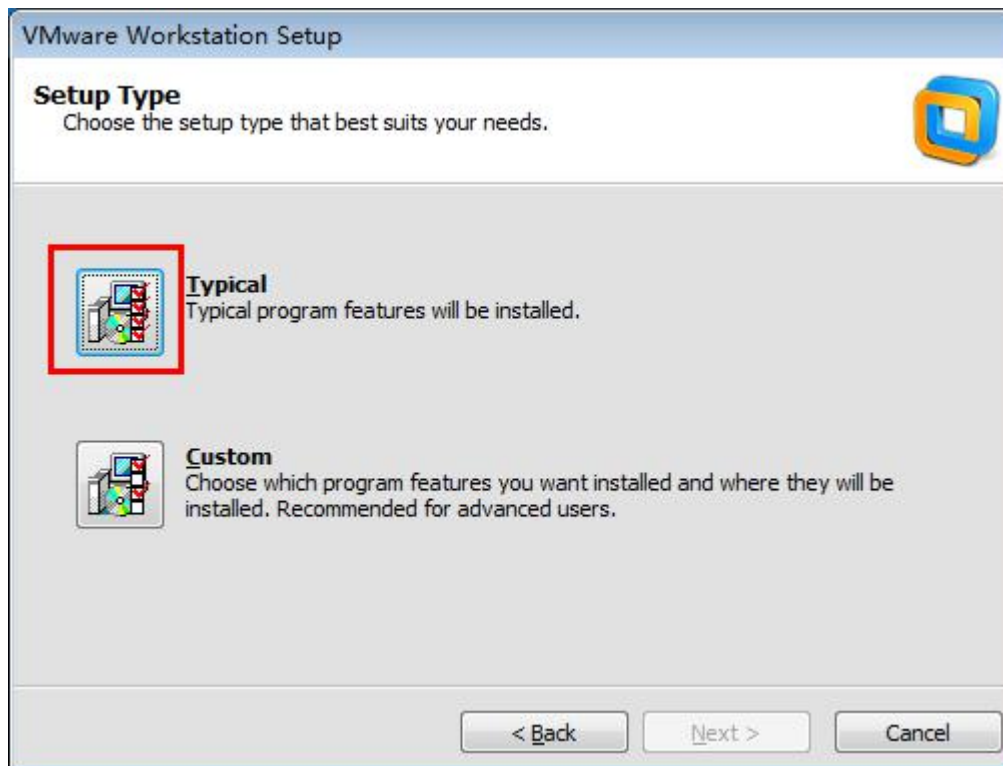
## 1、VMware workstation 安装：

1 双击 VMware-workstation-full-9.0.0-812388.exe

2 点击 next 进行安装



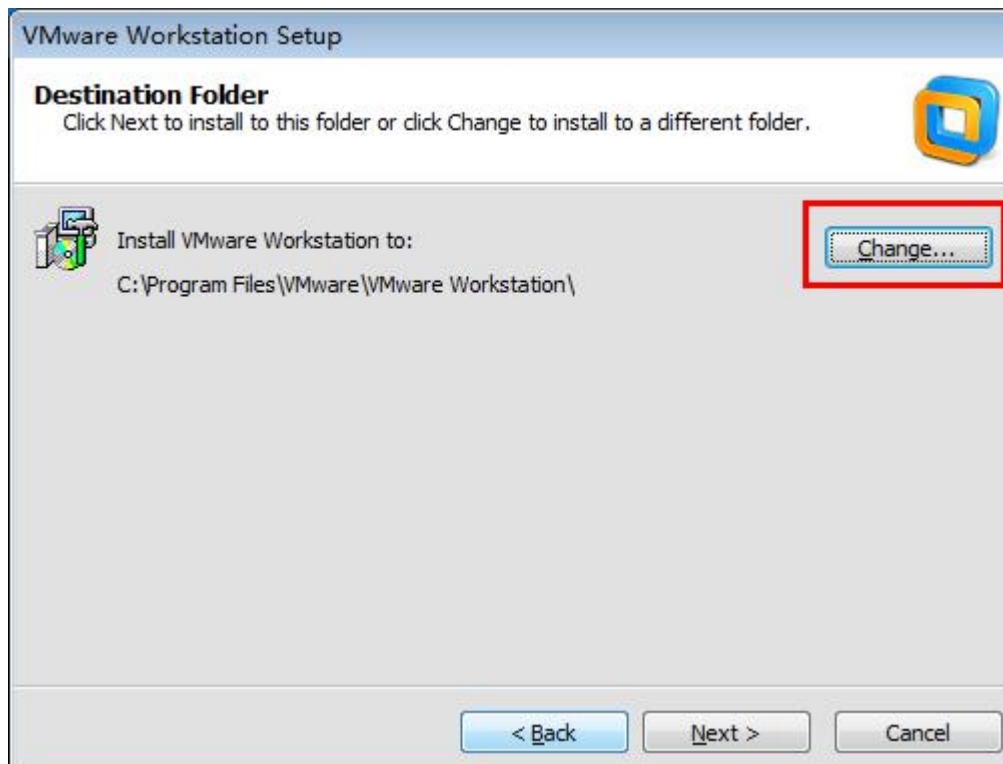
3.选择安装方式



Typical: 典型安装

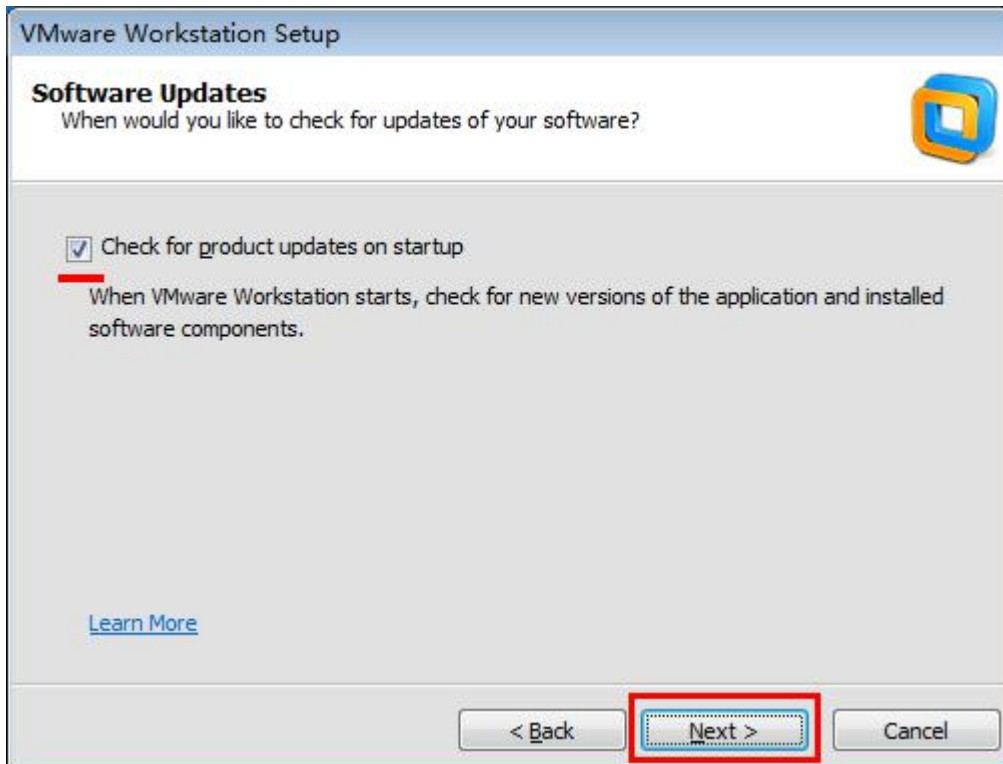
Custom: 自定义安装

#### 4.选择程序安装位置



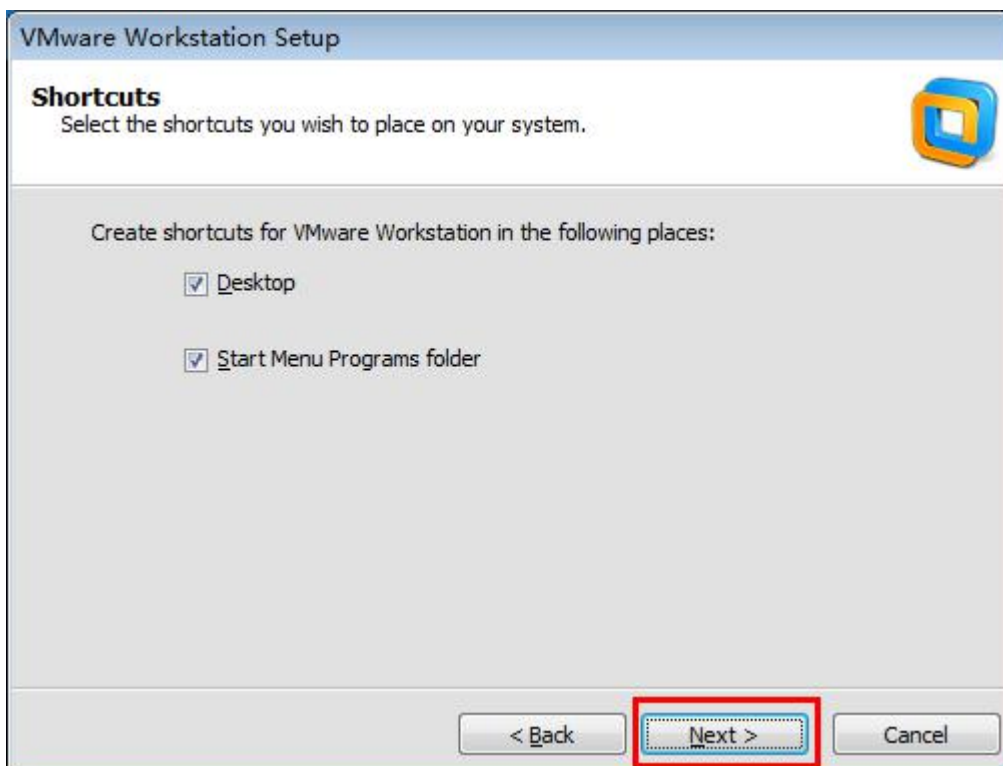
点击 change 选择程序安装位置，然后点击 next

#### 5.选择是否自动检测更新



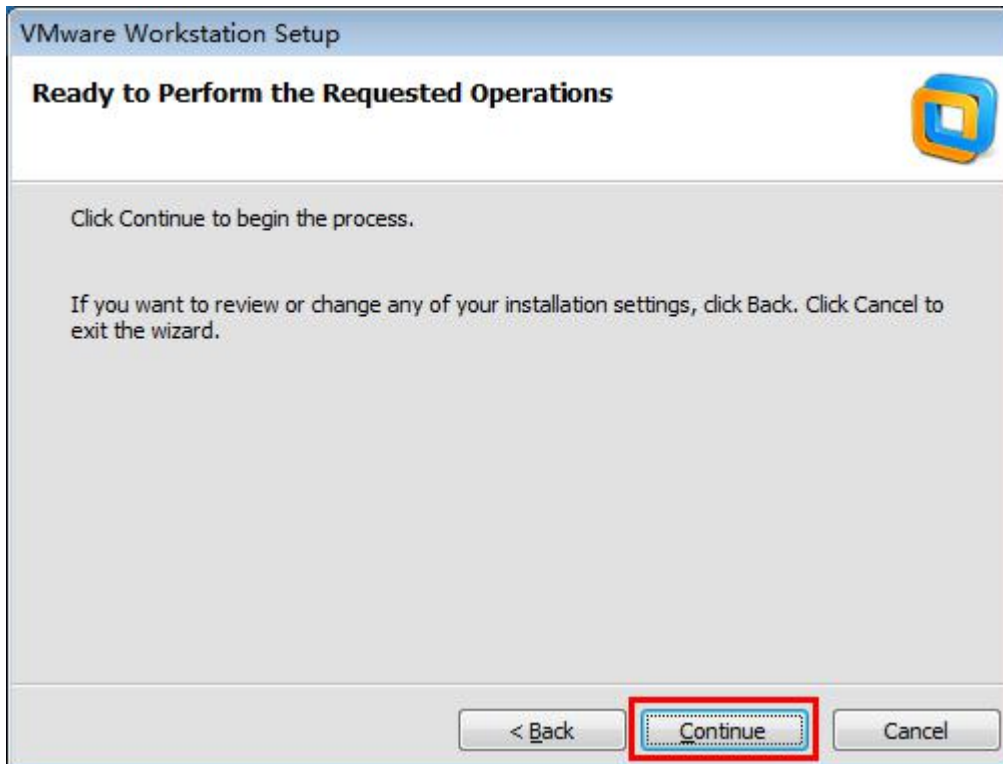
如勾选，有新版时会提示你跟新版本，点 next 进行下一步

## 6.创建快捷方式



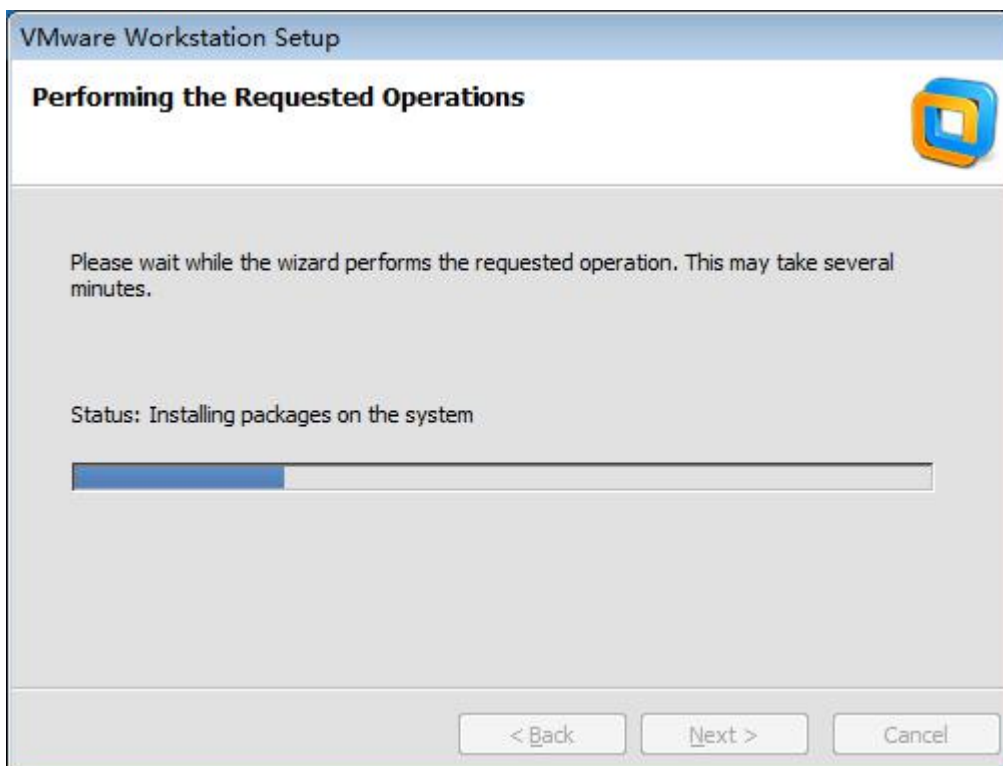
选择后点击 next

## 7.配置完成，开始安装程序

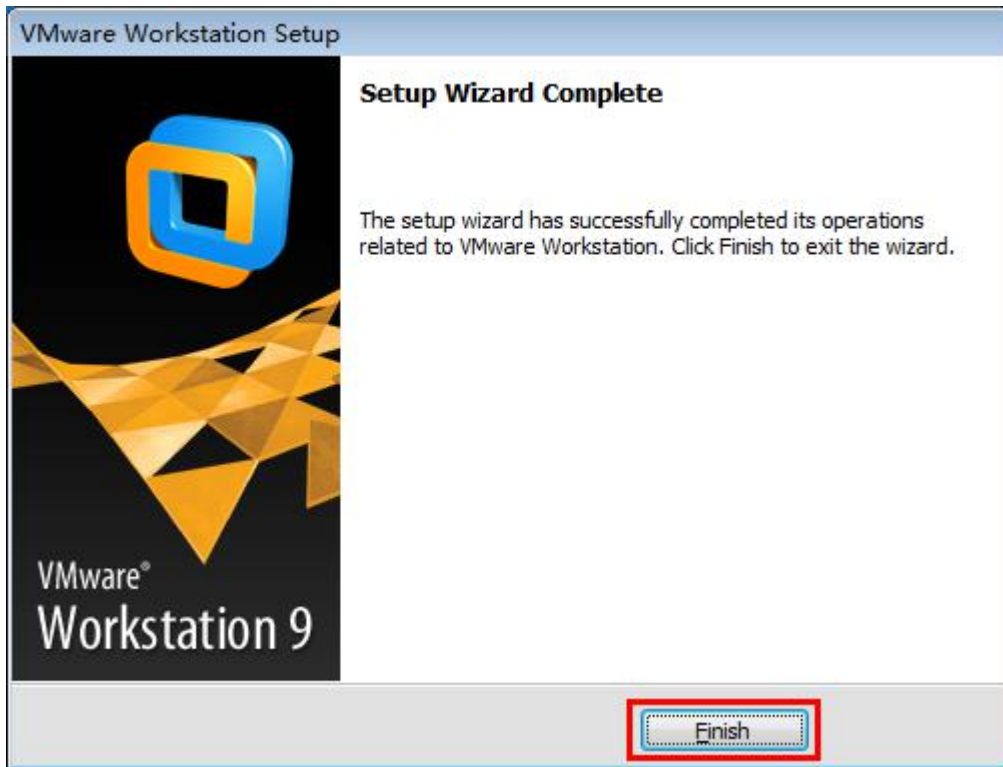


点击 continue

#### 8.开始安装虚拟原件



#### 9.完成



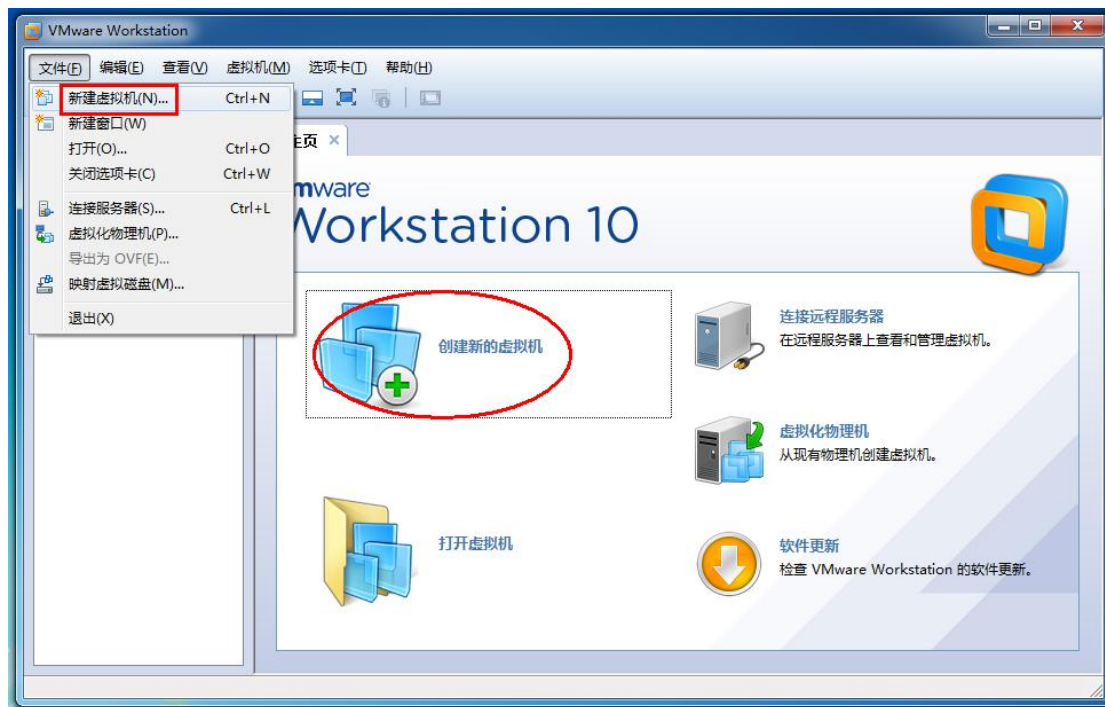
点击 finish 完成安装

## 2、使用 VMware 安装 CentOS:

### 安装步骤

第 1 步：首先安装 VMware Workstation，只需点下一步即可，安装过程略

第 2 步：文件 → 新建虚拟机 或 直接点击 创建新的虚拟机 图标



第 3 步：选择 典型（推荐）→ 下一步



第 4 步：稍后安装操作系统



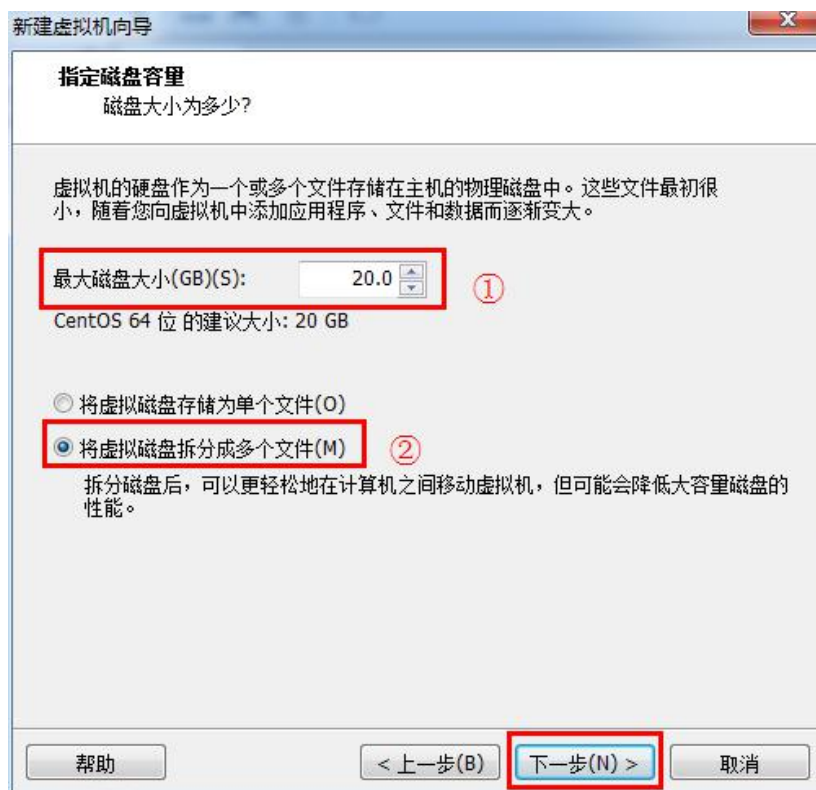
## 第 5 步：选择操作系统和版本



## 第 6 步：输入虚拟机名称和安装路径

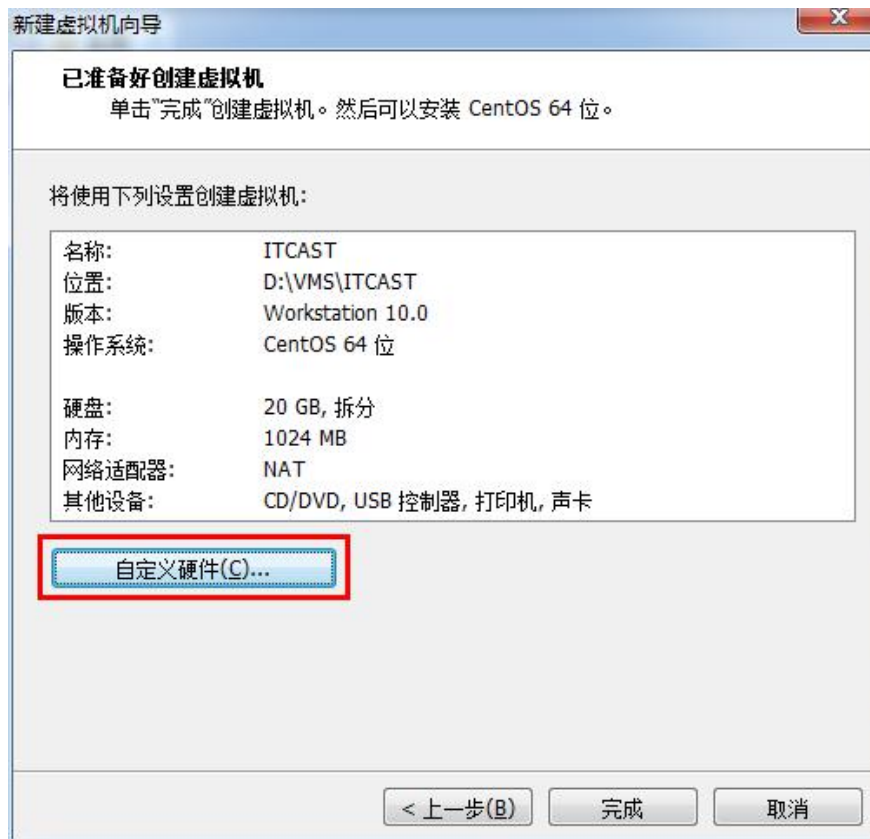


## 第 7 步：设置磁盘大小

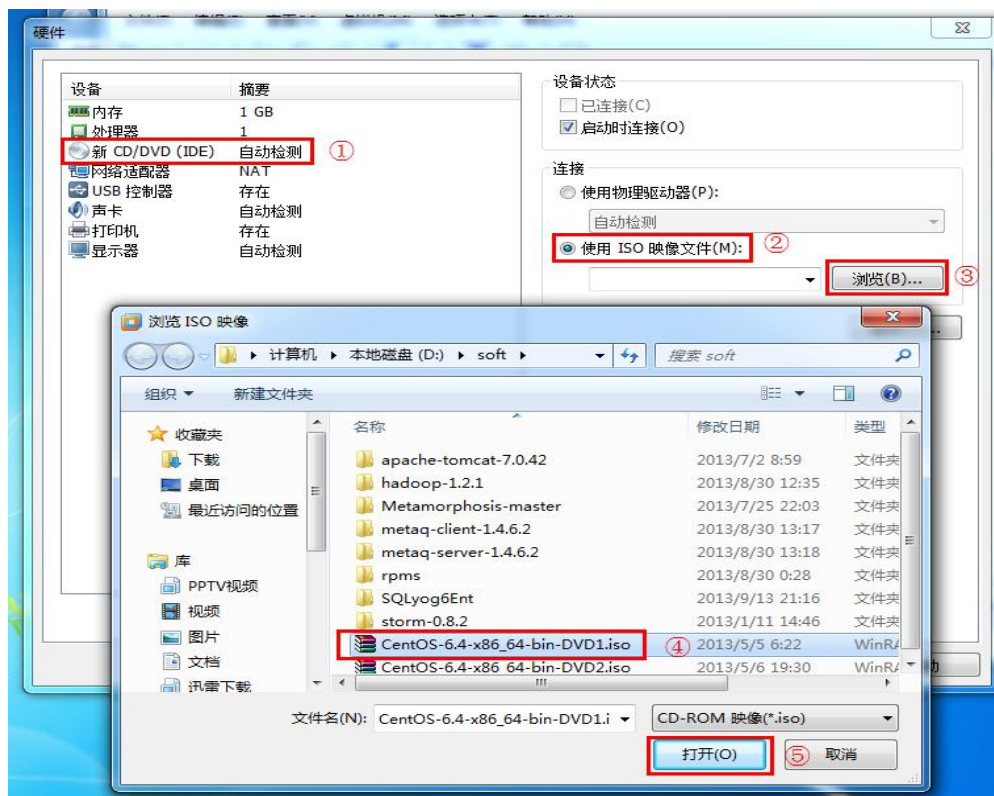


## 第 8 步：自定义硬件





## 第 9 步：选择 CentOS 安装镜像文件



## 第 10 步：点击完成



## 3、在 Linux 系统中安装 JDK 并完成环境变量配置

**【步骤一】：**上传 JDK 到 Linux 的服务器。

- \* 上传 JDK

- \* 卸载 open-JDK

```
java -version
```

```
rpm -qa | grep java
```

```
rpm -e --nodeps java-1.6.0-openjdk-1.6.0.35-1.13.7.1.el6_6.i686
```

```
rpm -e --nodeps java-1.7.0-openjdk-1.7.0.79-2.5.5.4.el6.i686
```

**【步骤二】：**在 Linux 服务器上安装 JDK.

- \* 通常将软件安装到/usr/local

- \* 直接解压就可以

```
tar -xvf jdk.tar.gz -C 目标路径
```

```
tar -xvf jdk-7u71-linux-i586.tar.gz -C ./jdk
```

**【步骤三】：**配置 JDK 的环境变量.

配置环境变量：

- ① vi /etc/profile

- ② 在末尾行添加

```
#set java environment
JAVA_HOME=/usr/local/jdk/jdk1.7.0_71
CLASSPATH=.:$JAVA_HOME/lib/tools.jar
PATH=$JAVA_HOME/bin:$PATH
export JAVA_HOME CLASSPATH PATH
保存退出
③source /etc/profile 使更改的配置立即生效
```

## 4、安装 Nginx 服务器并启动

### 4.1 环境准备

- (1) 需要安装 gcc 的环境: `yum install gcc-c++`
- (2) 安装第三方的开发包 **PCRE**: `yum install -y pcre pcre-devel`
- (3) 安装 **zlib**: `yum install -y zlib zlib-devel`
- (4) 安装 OpenSSL: `yum install -y openssl openssl-devel`

### 4.2 Nginx 下载

官方网站下载 nginx: <http://nginx.org/>

### 4.3 Nginx 安装

第一步: 把 nginx 的源码包 `nginx-1.8.0.tar.gz` 上传到 linux 系统

第二步: 解压缩

```
tar zxvf nginx-1.8.0.tar.gz
```

第三步: 进入 `nginx-1.8.0` 目录 使用 `configure` 命令创建一 `makeFile` 文件。

```
./configure \
--prefix=/usr/local/nginx \
--pid-path=/var/run/nginx/nginx.pid \
--lock-path=/var/lock/nginx.lock \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--with-http_gzip_static_module \
--http-client-body-temp-path=/var/temp/nginx/client \
--http-proxy-temp-path=/var/temp/nginx/proxy \
--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \
--http-uwsgi-temp-path=/var/temp/nginx/uwsgi \
```

```
--http-scgi-temp-path=/var/temp/nginx/scgi
```

```
[root@localhost nginx-1.8.0]# ll
total 660
drwxr-xr-x 6 1001 1001 4096 Sep  1 02:44 auto
-rw-r--r-- 1 1001 1001 249124 Apr 21 2015 CHANGES
-rw-r--r-- 1 1001 1001 379021 Apr 21 2015 CHANGES.ru
drwxr-xr-x 2 1001 1001 4096 Sep  1 02:44 conf
-rwxr-xr-x 1 1001 1001 2478 Apr 21 2015 configure
drwxr-xr-x 4 1001 1001 4096 Sep  1 02:44 contrib
drwxr-xr-x 2 1001 1001 4096 Sep  1 02:44 html
-rw-r--r-- 1 1001 1001 1397 Apr 21 2015 LICENSE
-rw-r--r-- 1 root root 345 Sep  1 02:45 Makefile
drwxr-xr-x 2 1001 1001 4096 Sep  1 02:44 man
drwxr-xr-x 3 root root 4096 Sep  1 02:45 objs
-rw-r--r-- 1 1001 1001 49 Apr 21 2015 README
drwxr-xr-x 8 1001 1001 4096 Sep  1 02:44 src
```

第四步：编译

```
make
```

第五步：安装

```
make install
```

## 4.4 Nginx 启动与访问

注意：启动 nginx 之前，上边将临时文件目录指定为/var/temp/nginx/client，需要在/var 下创建此目录

```
mkdir /var/temp/nginx/client -p
```

进入到 Nginx 目录下的 sbin 目录

```
cd /usr/local/nginx/sbin
```

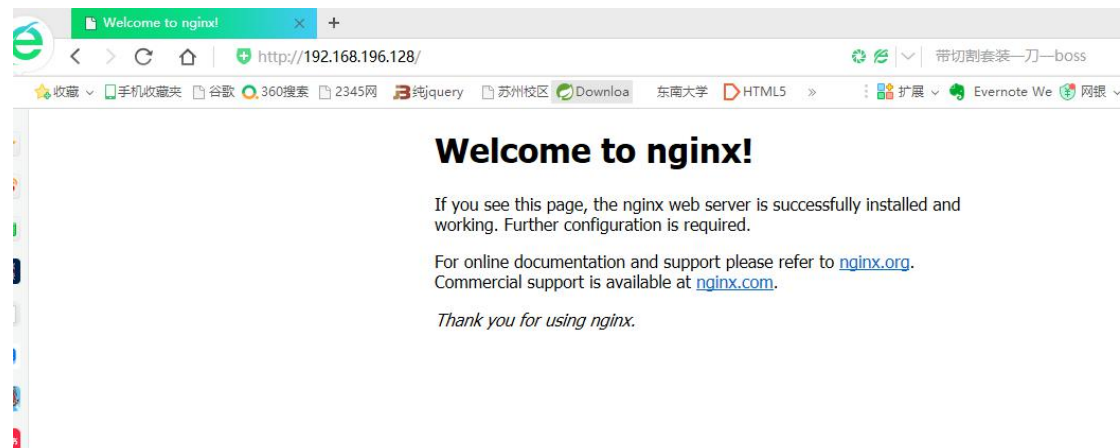
输入命令启动 Nginx

```
./nginx
```

启动后查看进程

```
ps aux|grep nginx
```

地址栏输入虚拟机的 IP 即可访问（默认为 80 端口）



关闭 nginx:

```
./nginx -s stop
```

或者

```
./nginx -s quit
```

 保存后退出

重启 nginx:

1、先关闭后启动。

2、刷新配置文件:

```
./nginx -s reload
```

## 5、将京东首页部署至 Springboot 项目，使用 Nginx 实现负载均衡配置,并启动三个不同的 web 服务器。

在 nginx 的配置文件中配置一下内容

```
upstream tomcat-jd{
    server 192.168.196.128:8080;
    server 192.168.196.128:8081;
    server 192.168.196.128:8082;
}
server {
    listen      80; #默认端口号
    server_name localhost; #域名或 ip

    location / {
        #root    index; #默认访问资源的目录
        proxy_pass http://tomcat-jd;
        index    index.html index.htm; #默认访问的资源名称
    }
}
```

在虚拟机中同时开启三个 tomcat,修改 tomcat 的端口号分别为 8080, 8081, 8082

将京东首页项目以 ROOT 命名打成 war 包，上传到三个 tomcat 中的 webapps 中。

浏览器输入 192.168.196.128 访问 nginx,随机访问不同的 tomcat.

```
[root@lj tomcat]# tomcat1/bin/startup.sh
J$ing CATALINA_BASE:   /usr/local/tomcat/tomcat1
J$ing CATALINA_HOME:   /usr/local/tomcat/tomcat1
J$ing CATALINA_TMPDIR: /usr/local/tomcat/tomcat1/temp
J$ing JRE_HOME:        /usr/local/jdk/jdk1.7.0_71
J$ing CLASSPATH:       /usr/local/tomcat/tomcat1/bin/bootstrap.jar:/usr/local/tomcat/tomcat1/bin/tomcat-juli.jar
tomcat started.
[root@lj tomcat]# tomcat2/bin/startup.sh
J$ing CATALINA_BASE:   /usr/local/tomcat/tomcat2
J$ing CATALINA_HOME:   /usr/local/tomcat/tomcat2
J$ing CATALINA_TMPDIR: /usr/local/tomcat/tomcat2/temp
J$ing JRE_HOME:        /usr/local/jdk/jdk1.7.0_71
J$ing CLASSPATH:       /usr/local/tomcat/tomcat2/bin/bootstrap.jar:/usr/local/tomcat/tomcat2/bin/tomcat-juli.jar
tomcat started.
[root@lj tomcat]# tomcat3/bin/startup.sh
J$ing CATALINA_BASE:   /usr/local/tomcat/tomcat3
J$ing CATALINA_HOME:   /usr/local/tomcat/tomcat3
J$ing CATALINA_TMPDIR: /usr/local/tomcat/tomcat3/temp
J$ing JRE_HOME:        /usr/local/jdk/jdk1.7.0_71
J$ing CLASSPATH:       /usr/local/tomcat/tomcat3/bin/bootstrap.jar:/usr/local/tomcat/tomcat3/bin/tomcat-juli.jar
tomcat started.
```





