

建设一个运维外挂的最佳实践

Shell脚本 eryajf 2年前 (2018-07-22) 3412°C 已收录 4个评论

本文预计阅读时间 17 分钟

1，由来。

运维外挂，听上去貌似是一个比较让人摸不着头脑的一个词儿，究竟背后包含了什么不为人知的事实真相呢，今天我就来分享一下个人在公司使用的方方面面最佳实践。

最早看到这个词儿，是在一篇公众号文章中提到的，其中对运维外挂解释如下：

什么是运维外挂? 无论在哪里,一行命令” 武器” 下载到手边。
当你登录一台新主机 # `curl -s xabc.io/v|sh` 一行命令创建自己习惯的 vimrc 环境,网络版的类 alias 用法,这就是运维外挂,加速你的操作,建设你企业的共享部署资源,节约你宝贵的时间。

文章目录

[隐藏]

- 1，由来。
- 2，初衷。
- 3，实战。
 - 1，安装nginx。
 - 2，配置nginx。
 - 3，准备原料。
- 4，讲解。
- 5，升级。

看到其中的安装代码，好像看懂了，毕竟以前在安装一些服务的时候，大概是见过这种安装形式的。但当我打算认真学习一下这个外挂是怎么建成的时候发现，教程中依赖于lua所写的一个脚本，而自己对lua又不甚熟悉，因此只得放弃这里的方案，自己另外再寻找方案。

2，初衷。

先说下为什么我第一眼看到这个东西就立刻产生兴趣了吧。

其实今天这个繁杂而又浮躁的时代中，每天我们浏览大量的对自己没有任何意义与帮助的信息，对很多或许重要或许优秀的信息，大多也都蜻蜓点水一样就那么擦肩而过了，或许正如您在看这篇文章一般的，有的人觉得非常好，正适合我的需求，我也要搞一套起来，有的人就觉得soso，上下一翻腾，就又跳转到其他网页去了。一切都由天定，如果你在这篇文章看到了这里，就是缘分。

额，一下没控制住，就扯远了。

说回正题，其实日常工作当中，无论是内部的测试环境，还是预发以及线上的环境，因为时常会有变动，因此一些高频率部署的软件，可能每次在新建一个虚拟机之后，都要先去把软件包拷过来，然后再进行安装，就算已经有了安装脚本来让效率增加，但是对于安装一个诸如jdk环境的包，我们都知道这个包还挺大的，可能你还要在你的本机上找半天（如果没有做好软件归类汇总的话），然后还要花时间等待传输，最后再进行安装。这个过程虽然不至于多么痛苦，但是总是不够最佳。

还有就是我们安装完软件之后，往往，都会在主机上留下一堆乱七八糟的安装材料（安装包啦，解压过的文件啦，等等），占用空间是一点，还有就是显得比较凌乱。

因此这不是我们安装常用软件的最佳实践。



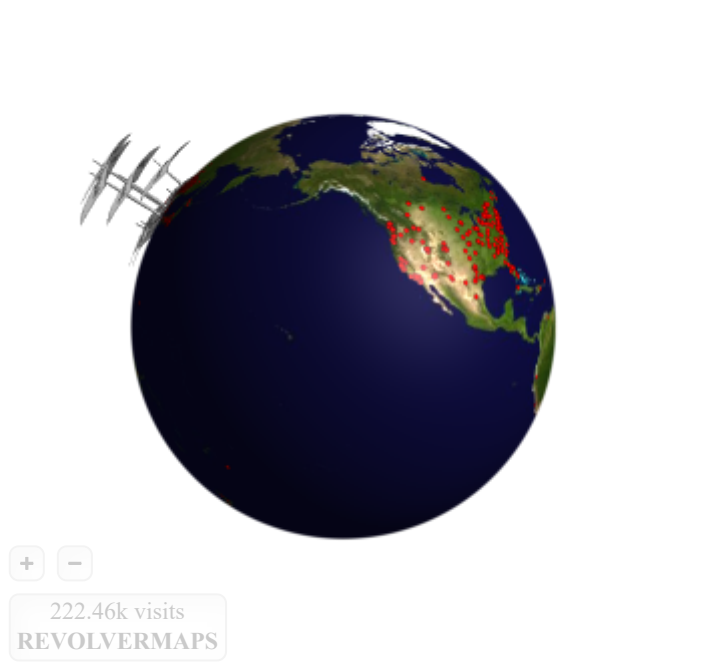
郑智化-我这样的男人



微信扫码，订阅本站！




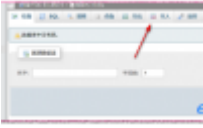
地球访问



不期而遇

所有文章

- 

<二>zabbix实战应用-zabbix-agent配置
2018-05-24 3评论
- 

wordpress整站迁移操作
2018-05-15 0

反馈
建议

我们想要的，就是方便简单的一键安装，安装完之后只有软件的安装目录，其他的一律没有。今天的[运维外挂](#)，就是来满足这个需求的。

3，实战。

当我们了解到那种安装方式之后，大概就能明白，用curl下载一个脚本，然后sh这个脚本进行安装的操作，把这些脚本放进nginx，大概再合适不过了。

于是在内网找到一台没有跑nginx服务的主机。

1，安装nginx。

2，配置nginx。

配置信息大概如下：

```
1.  server {
2.      listen      80;
3.      server_name test.install.com;
4.      charset utf-8;
5.      location / {
6.          root    /usr/local/nginx/html/install;
7.          autoindex      on;
8.          autoindex_exact_size    off;
9.          autoindex_localtime    on;
10.     }
11. }
```

3，准备原料。

在刚才nginx当中定义好的install目录下，放置各个软件的安装脚本，用不同的名字区别开，然后可以另外创建一个pack的目录，专门存放安装包。

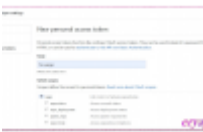
别的还没弄，就先弄了几个常用的软件安装脚本了。

nginx

```
1.  #!/bin/bash
2.  set -e
3.  ip=192.168.10.10
4.
5.  nginx(){
6.      dir=`pwd`
7.      [ -d /usr/local/nginx ] && echo "已经安装nginx" && exit 0
8.      yum install wget gcc gcc-c++ pcre pcre-devel zlib zlib-devel openssl openssl-devel -y
9.      wget $ip/pack/nginx-1.9.3-1.x86_64.rpm
10.     yum -y install nginx-1.9.3-1.x86_64.rpm
11.     ln -s /usr/local/nginx/sbin/* /usr/local/sbin
12.     rm -rf $dir/nginx*
13. }
14. nginx
```

jdk

```
1.  #!/bin/bash
2.  set -e
3.  ip=192.168.10.10
4.
5.  jdk8(){
6.      dir=`pwd`
7.      yum -y install wget && wget $ip/pack/jdk.tar.gz
8.      tar xf jdk.tar.gz && mv jdk1.8.0_144 /usr/local/jdk8
9.      echo 'JAVA_HOME=/usr/local/jdk8' >> /etc/profile
10.     echo 'PATH=$PATH:$JAVA_HOME/bin' >> /etc/profile
11.     echo 'export PATH' >> /etc/profile
12.     echo "进入root目录,运行 source /etc/profile 命令正式完成！"
13.     rm -rf $dir/jdk*
14. }
15. jdk
```



使用PicGo+GitHub自建免费图床

2019-07-16 0



Logrotate入门了解及生产实践

2020-04-28 0



nginx通过四层代理实现端口转发

2018-11-08 0

牧童遥指（友链）

abcdocker	utools
一路向北	不死鸟-分享为王
五福吧	儒释道多元文化讲堂
北慕城南博客	十年之约
吾悟屋-运维艺术	周立的博客
山山仙人	张志明博客
张馆长	懒得勤快
收复台湾	散尽浮华
月与灯依旧	朋友-肖硕
朋友-韦勤耕	朱双印个人日志
柒's blog	清晨博客
熊哥club	爱运维
菜鸟博客	闲不住的人

标签云

Jenkins (45) k8s (39) 系统优化 (25)

elk (22) k8s集群 (21) kubernetes (20)

迎刃而解 (16) docker (16) nginx (16)

nexus (13) 效率工具 (11) 我的爷爷 (11)

Git (10) elasticsearch (10) rancher (9)

prometheus (8) pipeline (8) 坑啊坑 (7)

消息队列 (7) tomcat (6) 术业专攻 (6)

云计算 (6) shell脚本 (6) maven (6)

golang (6) openLDAP (6) kvm (5)

zabbix (5) java (5) 影视分享 (5) kafka (5)

流水线 (5) 插件 (4) 配置文件 (4) wiki (4)

磁盘 (4) VMware (4) PHP (4)

Dockerfile (4) filebeat (4) RabbitMQ (4)

MongoDB (4) 数据库 (4) ansible (4)

CMDB (3) wordpress (3) node (3)

开源项目 (3) Python (3) MySQL (3)

镜像 (3) kibana (3) 中间件 (3) logstash (3)

回滚 (3) 代码 (2) 钉钉 (2) Android (2)

自动化运维 (2) sftp (2) iptables (2)

yum (2) 招聘面试 (2) 年终总结 (2)

进程管理 (2) systemd (2) Inmp (2) Mac (2)

反馈
建议

tomcat

```
1.  #!/bin/bash
2.  set -e
3.  ip=192.168.10.10
4.
5.  tomcat(){
6.      dir=`pwd`
7.      yum -y install wget && wget $ip/pack/apache-tomcat-8.0.47.tar.gz
8.      tar xf apache-tomcat-8.0.47.tar.gz && mv apache-tomcat-8.0.47 /usr/local/tomcat
9.      rm -rf $dir/*tomcat*
10. }
11. tomcat
```

maven

```
1.  #!/bin/bash
2.  set -e
3.  ip=192.168.10.10
4.
5.  maven(){
6.      dir=`pwd`
7.      yum -y install wget && wget $ip/pack/maven.tar.gz
8.      tar -xf maven.tar.gz && mv maven /usr/local/maven
9.      rm -rf $dir/maven.tar.gz
10.     echo 'MAVEN_HOME=/usr/local/maven' >> /etc/profile
11.     echo 'PATH=$PATH:$JAVA_HOME/bin:$MAVEN_HOME/bin' >> /etc/profile
12.     echo "进入root目录,运行 source /etc/profile 命令正式完成！"
13. }
14. maven
```

mysql

```
1.  #!/bin/bash
2.  set -e
3.  ip=192.168.10.10
4.
5.  mysql(){
6.      yum -y install wget
7.      wget $ip/pack/mysql-5.6.16.tar.gz && echo "获取mysql安装包"
8.      [ ! -f $dir/mysql-5.6.16.tar.gz ] && exit 1
9.      mqnu=`cat /etc/passwd | grep mysql |wc -l`
10.     if [ $mqnu -ne 1 ];then
11.         echo "mysql用户不存在,新建用户"
12.         groupadd mysql
13.         useradd -g mysql -s /sbin/nologin mysql
14.     else
15.         echo "mysql已经存在"
16.     fi
17.
18.     yum install gcc gcc-c++ autoconf automake zlib* libxml* ncurses-devel libtool-ltdl-devel* make cmake -y
19.
20.     [ ! -d /usr/local/mysql/data ] && mkdir -p /usr/local/mysql/data && chown -R mysql:mysql /usr/local/mysql
21.     echo "开始编译安装！！"
22.     tar -xf mysql-5.6.16.tar.gz && cd mysql-5.6.16 && cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -DMYSQL_DATADIR=/usr/local/mysql/data -DWITH_MYISAM_STORAGE_ENGINE=1 -DWITH_INNOBASE_STORAGE_ENGINE=1 -DWITH_MEMORY_STORAGE_ENGINE=1 -DWITH_READLINE=1 -DMYSQL_UNIX_ADDR=/var/lib/mysql/mysql.sock -DMYSQL_TCP_PORT=3306 -DENABLED_LOCAL_INFILE=1 -DWITH_PARTITION_STORAGE_ENGINE=1 -DEXTRA_CHARSETS=all -DDEFAULT_CHARSET=utf8 -DDEFAULT_COLLATION=utf8_general_ci && make && make install
23.     echo "注册为服务！！"
24.     cd /usr/local/mysql/scripts && ./mysql_install_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data
25.     cd /usr/local/mysql/support-files && cp mysql.server /etc/rc.d/init.d/mysql && yes | cp my-default.cnf /etc/my.cnf && chkconfig --add mysql && chkconfig mysql on && service mysql start
26.
```

- 监控告警 (2) openVPN (2) ETCD (2)
- TCP (2) 阿里云 (1) Windows (1)
- 项目管理 (1) redis (1) pinpoint (1)
- supervisor (1) sonar (1) rsyslog (1)
- 微服务 (1) 我的父亲 (1) 读书相关 (1)
- 工作感悟 (1) 宝贝宝贝 (1) 随感随想 (1)
- 图床建设 (1) vscode (1) CentOS7 (1)
- 小技巧 (1) 闲言碎语 (1) CPU (1) gradle (1)
- 安卓 (1) 抓包 (1) neovim (1)
- OpenResty (1) gorm (1) Vue (1)
- element-ui (1) 实战经验 (1)

最新评论

点此去留言板

- daimei20小时前说：回复可见哈哈哈哈哈
- PonyZebra2天前说：虽然听说很简陋，但是还是想试一试
- alian2天前说：申请友链：名称：小爱博客地址：https://www.51ceo.wang
- alian2天前说：申请友链：
- alian2天前说：名称：小爱博客地址：https://www.51ceo.wang描述：分享i
- aimleft3天前说：测试回复可见是否可用！
- XX3天前说：嘿嘿
- kkkk3天前说：谢谢，很好用，感谢
- yuxiaotao4天前说：我来看看密码
- coolwo7天前说：龙哥牛逼

网站统计

文章总数：321 篇 评论数目：1715 条

建站日期：2017-08-13 运行天数：1111 天

标签总数：104 个 用户总数：1712 个

最后更新：2020-8-20

文章归档

- 2020年八月 (2) 2020年七月 (7)
- 2020年六月 (6) 2020年五月 (5)
- 2020年四月 (2) 2020年三月 (9)

反馈

建议

```
27.     echo 'PATH=/usr/local/mysql/bin:$PATH' >> /etc/profile
28.     echo 'export PATH' >> /etc/profile
29.
30.     rm -rf $dir/mysql*
31. }
32. mysql
```

node

```
1.  #!/bin/bash
2.  set -e
3.  ip=192.168.10.10
4.
5.  node(){
6.      dir=`pwd`
7.      yum -y install wget && wget $ip/pack/node-v10.6.0-linux-x64.tar.xz
8.      tar -xf node-v10.6.0-linux-x64.tar.xz && mv node-v10.6.0-linux-x64
/usr/local/node
9.      rm -rf $dir/*node*
10.      echo 'NODE=/usr/local/node' >> /etc/profile
11.      echo 'PATH=$PATH:$NODE/bin' >> /etc/profile
12.      echo "进入root目录,运行 source /etc/profile 加载并可使用 node -v检验安
装！ "
13.  }
14.  node
```

如果仔细看脚本的话，大概能看出，我们直接在当前主机下的web根目录下的pack目录下放的安装包，先下载到本地，然后进行对应的安装操作，最后再将安装包之类的删除，真正做到来无影去无踪。武林绝学莫过于此！

4，讲解。

因为上边的脚本我们都是直接命名为jdk, tomcat, nginx, node，，，因此我们在使用的时候可以直接用如下操作来进行软件的安装。

```
1.  curl 192.168.10.10/nginx | sh
2.  curl 192.168.10.10/jdk | sh
3.  curl 192.168.10.10/tomcat | sh
4.  curl 192.168.10.10/maven | sh
5.  curl 192.168.10.10/mysql | sh
6.  curl 192.168.10.10/node | sh
```

无论你是在什么地方，只用执行对应的命令，即可直接进行对应的软件安装咯。当然如果可能的话，这里的ip也可以替换成域名。

但是经过测试，发现这个操作适用于CentOS-7的系统，在6的系统上应用的话，就行不通咯。

还有，诸上一些操作只是一个小小的抛砖引玉，您可以根据自己的实际需求，定制出更多便捷的安装脚本。

5，升级。

你以为这样就完了吗，一开始我也觉得就这么完了，就这么用了，已经挺好的了，直到在我打算优化一下脚本时看到了如下的一个神器。

名称叫做菜单选择脚本，在虚拟机上执行一遍之后，脑子一转，就开始了我的外挂升级之旅。

根据对菜单选择脚本的演练以及观察，最后摸清了其规律，于是最终优化版脚本如下（脚本已放置在个人Github中，点击可跳转查看）：

🐱 运维外挂脚本

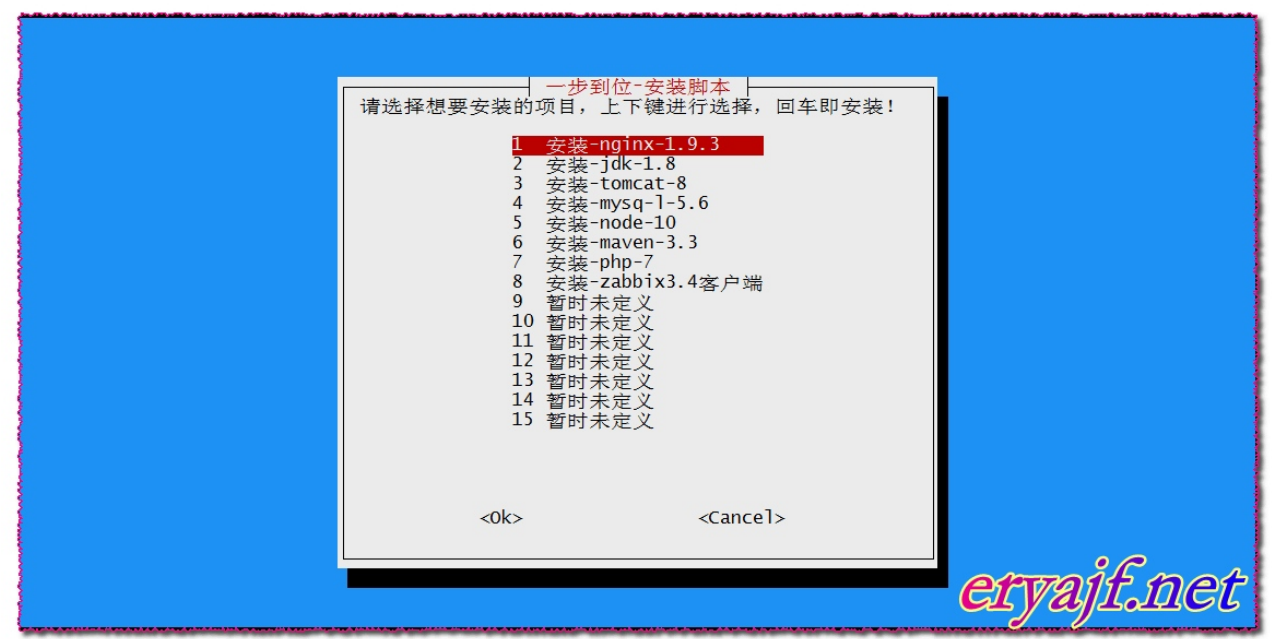
我把这个脚本的名字定义为a，添加上执行权限，最终安装效果如下图所示：

2020年二月 (3)	2020年一月 (3)
2019年十二月 (10)	2019年十一月 (4)
2019年十月 (6)	2019年九月 (5)
2019年八月 (12)	2019年七月 (16)
2019年六月 (8)	2019年五月 (6)
2019年四月 (7)	2019年三月 (8)
2019年二月 (15)	2019年一月 (9)
2018年十二月 (39)	2018年十一月 (7)
2018年十月 (20)	2018年九月 (9)
2018年八月 (15)	2018年七月 (14)
2018年六月 (28)	2018年五月 (22)
2018年四月 (11)	2018年三月 (13)

文章日历

2020年八月						
一	二	三	四	五	六	日
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						
« 7月						

curl 192.168.10.10/a | sh



如此一来，以后安装软件的时候，只需要执行一条命令，通过上下键进行选择，然后回车即可安装对应的软件啦。


最后，单独放出那个菜单选择脚本，并做一些简单的个人理解：

菜单选择脚本：

```
1.  #!/bin/bash
2.  OPTION=$(whiptail --title "Test Menu Dialog" --menu "Choose your option"
3.  15 60 4 \
4.  "1" "Grilled Spicy Sausage" \
5.  "2" "Grilled Halloumi Cheese" \
6.  "3" "Charcoaled Chicken Wings" \
7.  "4" "Fried Aubergine" 3>&1 1>&2 2>&3)
8.  exitstatus=$?
9.  if [ $exitstatus = 0 ]; then
10.     echo "Your chosen option:" $OPTION
11. else
12.     echo "You chose Cancel."
```

- 1，菜单选择脚本的成立依赖于 `whiptail` 这个工具，这好像是一个很强大的工具，日后可以多多了解一波。
- 2，从演练的执行当中我们看出， `OPTION` 是由下边 `1234` 条内容组成，当我们选择了某一个选项之后， `OPTION` 的值是对应的 `1234`，我们进行操作的时候，只用取到它的这个值即可展开各种操作啦。
- 3， `title` 以及 `menu` 都是可以自定义的，后边的三个数字是定义的选择框的长宽以及背景的大小的，可根据自己实际需求进行调整。
- 4， `3>&1 1>&2 2>&3` 这里应该是一个固定格式，照着写即可，不必纠结过多。
- 5，下边的执行内容，原文是 `if` 的判断，我也就接着用if了，并没有采取 `case` 语句，因为目前这种操作来看，都差不多。

如此，一个基本的所谓的运维安装外挂应该算是相对完善的制作完成了，当然脚本里边其实还应该再加一下是否已经安装过软件，如果系统已安装就取消新安装的操作，这些稍候进行一个完善！



扫码订阅本站，第一时间获得更新

微信扫描二维码，订阅我们网站的动态，另外不定时发送WordPress小技巧，你可以随时退订，欢迎订阅哦~

二丫讲梵，版权所有 | 如未注明，均为原创 | 本网站采用BY-NC-SA协议进行授权，转载请注明建设一个运维外挂的最佳实践！