部署方案说明及优缺点对比

一、原有部署方式与一键部署方式对比

对于现有部署的方式是通过使用 SVN 下载到最新代码后使用本地 IDE 编译成可运行包发送到 SVN 中间机器，然后发起上线请求，再由运维拉取代码把相应的 IP 等修改成线上地址后发布部署。我们暂时把这种部署方式设定为（1），方便后面比较使用。

新系统的一键部署方式则把这些手工流程全部程序化，程序自动把程序从版本控制服务器机器下载下来后－>自动编译成可运行的包－>自动把可运行包发送到指定位置、并自动重启相关服务，无需手工操作其中任何一个环节。只需要第一次部署的时候编写好一键部署脚本即可，我们暂时把这种部署方式设定为(2)，方便后面比较使用。

从两种部署上线方式，通俗的对比一下两种方式的优缺点：

1. （1）在整个上线流程里有过多的人工操作情况，容易操作失误，人工成本过大。每次上线持续的周期过长。工作效率偏低，
2. （2）在整个上线流程里基本上全靠程序操作。一键部署。周期短。少人工操作干预。每次上线持续时间较短。工作效率高。
3. 目前主流的方式基本上都是 jenkins 图形化一键部署，我们提供的是脚本一键部署，可以通过（2）轻松过渡到 jenkins

上。未来向大型互联网公司甚至 BAT 看齐，提供便利。

4. 我们在提供功能的前提下，也同时希望能把比较先进，便于操作的技术带到贵公司。技术的革新是能带动一定的生产力的，（2）相比于（1）来说可以降低工作操作成本，提高生产力，也同时能为未来新技术提供基础平台，（2）在我们公司看为为持续集成做的一个大铺垫，持续集成在以前是一个趋势，但是在现在的这个互联网环境下 这个已经成为一个必备的高效工具。

下面我们从技术层面对比优缺点：

角度 / 方案	原有部署方式	一键部署方式	优缺点
采用的版本控制技术	SVN	GIT	Git 在开发多分支协同工作时，相比 SVN 有很大的优势，1. 分支操作简单。轻量，不需要克隆代码完成代码分支的工作，2. 有完整成熟的

			工作流模式,轻松建立分支，完成多需求协同开发，在分支合并上拥有智能合并，可以降低出错率。对于一些改动很大的操作也有完整的追踪。很适合多人团队协同工作。
构建工具	无	MAVEN	MAVEN 是一款目前主流且稳定的构建工具，他可以把项目通过一个简单的命令构建成可执行的包，同时他具有跨平

			台、跨 IDE 的优点，无论开发者使用的是 Eclipse、MyEclipse、Intellij 等 都可以做到完美兼容、开发。他比 Ant 更好用（不了解 Ant 的可以百度），无需你去找各种第三方插件。只需要一个配置就可以完成你想要的构建。总结：这是一个不可忽视的构建工具。
操作方式	人工	**自定义脚本**	我们通过自定

			义的脚本把这些人工需要执行的步骤，全部程序化为了一个简单的小脚本，通过执行这个小脚本就可以完成以前需要注意执行顺序、执行步骤的人工操作。让部署更简单、更方便。更让运维的工作更轻松，更精准。
配置修改	提交上线后，运维人工修改	**Host 模式。＋Prod 线上配置**	我们把 ip 化为一个个的（伪）内网域名。通过 host 配置直接指向后。也无需

			让开发知道 ip。就可以完成部署。便于运维和开发独立运营。通过也避免开发知道线上 IP。Prod 配置是一套适应线上的配置。Debug 配置用作测试环境。Prod 配置用于线上配置。配置独立。便于多环境协同测试开发。甚至可以添加灰度、预线上配置、添加灰度环境。当前比较主流的模式，比较成熟的方案

根据我上面从通俗层面和技术层面的刨晰。大家不难看出持续集成的魅力。为什么很多大型互联网公司都会是 使用持续集成的方案，可以在细节上每个公司都不太一样。但是大都殊途同归。我们这个方案不能说是最好的。以我对现在的情况的了解，这绝对是对贵公司的一个技术进步和革新。也是贵公司走向大型互联网公司甚至是 BAT 的一个必经过程。

二、部署中提到的内网访问 iptable 限制方式优点介绍

其实在我整理部署文档中遇到了很多的困扰。我认为 小机房、线上环境咱们公司都是有的、我认为 小机房和线上环境的互通是有专线部署的，我认为 很多很多认为，在初步讨论回复中 我也自以为的确定是存在的。但是结合现状发现我过于理想化了。So，结合目前的情况。最终确定阿里云部署所有服务。也是 ok 的。但是 iptable 方式是我坚持的。下面我说一下坚持的理由：

iptables 我们称之为网络层控制。算是 linux 下的防火墙。其实在实现内网访问控制有很多方案，比如 nginx 层控制、后端程序控制等等、我们就从 iptables 层和 nginx 层两款控制来对比说明其中的优缺点，这是我坚持的原因所在：

1. iptables 防火墙是网络层控制。顾名思义。采取 iptables 控制后，在网络层就会隔断请求进入。那么一个请求在执行第一次握手机制就会得到无法访问的回复，可以避免很多。

试图进入系统、试图攻击系统的一些垃圾请求。可以防御一些一般的攻击手段。大家可以去百度 iptables 可以得到防 ddos、cc 攻击等很多帖子、那么这就很能说明防火墙控制的重要性。咱们有两套程序一个是对公网开放的 一个是对内网开放的、在对内网开放的机器上配置 iptables 可以有效防止只对内网服务的功能 不流露到外网。使用 iptables 控制时 也同时间接节省了流量、风险程度等。

2. nginx 层控制、也能做到限制外网访问。那么我不建议使用的原因就在于。请求已经通过网络同意进入到 nginx 层了。说明流量已经进入到系统 。如果采用一些攻击手段。也不是不可能在 nginx 发现一些目前未发现的严重 bug 造成内网程序暴露的风险。这样大大提高了承受风险的几率、不利于节约流量、 同时也加大了真正的内网用户访问系统出现访问不到的几率。
3. 那么这两种方案 在 nginx 层控制可以减少使用一台虚拟机。但是相应的也增加了危险系数。也许这个系数是比较小。但是 风险如果能好的规避，或者降低。那么为什么去选择试图侥幸呢。

三、目的

对于我们公司来说。其实项目能顺利交付、顺序完工应该是我们

最大的需要。其实我们完全可以不写这个方案、我们只要建议到就够了。但是为什么还要去写这个文档。当然是希望能得到认可、希望能对贵公司有所帮助，这点容易理解。我们需要的是长期合作。从这个项目价格、技术使用等 我们相信懂的人一定会懂。我们的诚意。甚至本来一键部署都是为了更好而附送的。那么我们现在只是遵从我们的内心去做。去写这些。只希望能换来贵公司的一个认可。一个可以得到未来更多合作机会的一个契机吧。

我曾经问过一个朋友，做生意最重要的是什么，我朋友告诉我没有什么比利益更重要，然而对于这句话我并不认可、我认为做生意做重要的是心、诚心、真心、信心。我相信我们服务过的客户今天不晓得、明天不晓得、后天一定晓得。我们是站在客户的位置去看、去听、去做。也只需这样 足够了。

我们是一支专业做定制化系统、优化方案的公司（硬广），希望能用我们的专业让世界和平（神经病啊）。