随着微服务架构的普及，线上服务越来越多，随之而来的就是部署越来越频繁；随着互联网行业的兴旺，产品迭代的频率也是越来越快，服务上线速度逐步提升。有上线、有部署，就有风险。有风险，就对业务有影响，然后就有了一系列减少这种风险的部署方案：蓝绿部署、金丝雀发布（灰度发布），也有适应产品迭代频率的AB测试。

本文主要是简单解释下这几个概念，帮助自己理解。研发中心的同学务必都要搞清楚这些概念，方便后续的工作推进。

英语好的同学可以参考：

<https://blog.christianposta.com/deploy/blue-green-deployments-a-b-testing-and-canary-releases/>

**蓝绿部署**

蓝绿部署的目的是减少发布时的中断时间、能够快速撤回发布。

It’s basically a technique for releasing your application in a predictable manner with an goal of reducing any downtime associated with a release. It’s a quick way to prime your app before releasing, and also quickly roll back if you find issues.

蓝绿部署中，一共有两套系统：一套是正在提供服务系统，标记为“绿色”；另一套是准备发布的系统，标记为“蓝色”。两套系统都是功能完善的，并且正在运行的系统，只是系统版本和对外服务情况不同。

最初，没有任何系统，没有蓝绿之分。

然后，第一套系统开发完成，直接上线，这个过程只有一个系统，也没有蓝绿之分。

后来，开发了新版本，要用新版本替换线上的旧版本，在线上的系统之外，搭建了一个使用新版本代码的全新系统。 这时候，一共有两套系统在运行，正在对外提供服务的老系统是绿色系统，新部署的系统是蓝色系统。

蓝绿部署

蓝色系统不对外提供服务，用来做啥？

用来做发布前测试，测试过程中发现任何问题，可以直接在蓝色系统上修改，不干扰用户正在使用的系统。（注意，两套系统没有耦合的时候才能百分百保证不干扰）

蓝色系统经过反复的测试、修改、验证，确定达到上线标准之后，直接将用户切换到蓝色系统：

蓝绿部署

切换后的一段时间内，依旧是蓝绿两套系统并存，但是用户访问的已经是蓝色系统。这段时间内观察蓝色系统（新系统）工作状态，如果出现问题，直接切换回绿色系统。

当确信对外提供服务的蓝色系统工作正常，不对外提供服务的绿色系统已经不再需要的时候，蓝色系统正式成为对外提供服务系统，成为新的绿色系统。 原先的绿色系统可以销毁，将资源释放出来，用于部署下一个蓝色系统。

蓝绿部署只是上线策略中的一种，它不是可以应对所有情况的万能方案。 蓝绿部署能够简单快捷实施的前提假设是目标系统是非常内聚的，如果目标系统相当复杂，那么如何切换、两套系统的数据是否需要以及如何同步等，都需要仔细考虑。

[BlueGreenDeployment](https://martinfowler.com/bliki/BlueGreenDeployment.html)中给出的一张图特别形象：

蓝绿部署示意图

**金丝雀发布（灰度发布）**

金丝雀发布（Canary）也是一种发布策略，和国内常说的灰度发布是同一类策略。

蓝绿部署是准备两套系统，在两套系统之间进行切换，金丝雀策略是只有一套系统，逐渐替换这套系统。

金丝雀发布

譬如说，目标系统是一组无状态的Web服务器，但是数量非常多，假设有一万台。

这时候，蓝绿部署就不能用了，因为你不可能申请一万台服务器专门用来部署蓝色系统（在蓝绿部署的定义中，蓝色的系统要能够承接所有访问）。

可以想到的一个方法是：

只准备几台服务器，在上面部署新版本的系统并测试验证。测试通过之后，担心出现意外，还不敢立即更新所有的服务器。 先将线上的一万台服务器中的10台更新为最新的系统，然后观察验证。确认没有异常之后，再将剩余的所有服务器更新。

这个方法就是金丝雀发布。

实际操作中还可以做更多控制，譬如说，给最初更新的10台服务器设置较低的权重、控制发送给这10台服务器的请求数，然后逐渐提高权重、增加请求数。

这个控制叫做“流量切分”，既可以用于金丝雀发布，也可以用于后面的A/B测试。

蓝绿部署和金丝雀发布是两种发布策略，都不是万能的。有时候两者都可以使用，有时候只能用其中一种。

上面的例子中可以用金丝雀，不能用蓝绿，那么什么时候可以用蓝绿，不能用金丝雀呢？整个系统只有一台服务器的时候。

**A/B测试**

首先需要明确的是，A/B测试和蓝绿部署以及金丝雀，完全是两回事。

蓝绿部署和金丝雀是发布策略，目标是确保新上线的系统稳定，关注的是新系统的BUG、隐患。

A/B测试是效果测试，同一时间有多个版本的服务对外服务，这些服务都是经过足够测试，达到了上线标准的服务，有差异但是没有新旧之分（它们上线时可能采用了蓝绿部署的方式）。

A/B测试关注的是不同版本的服务的实际效果，譬如说转化率、订单情况等。

A/B测试时，线上同时运行多个版本的服务，这些服务通常会有一些体验上的差异，譬如说页面样式、颜色、操作流程不同。相关人员通过分析各个版本服务的实际效果，选出效果最好的版本。

a/b测试

在A/B测试中，需要能够控制流量的分配，譬如说，为A版本分配10%的流量，为B版本分配10%的流量，为C版本分配80%的流量。

**总结**

这里总结一下：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 特点 | 优势 | 劣势 |
| 蓝绿部署 | 同时存在两个集群，两个集群中只有一个集群真正提供服务，另外一个集群测试、验证或待命 | 服务文档，版本回退简单，适用于各种场景的升级，大版本不兼容升级的或迭代兼容升级 | 浪费硬件资源，需要同时有两个集群，如果集群比较大，比如有1000个节点，这种方式几乎不可用 |
| 金丝雀部署 | 逐点部署，逐步替换线上服务 | 小步快跑，快速迭代 | 只能适用于兼容迭代的方式，如果是大版本不兼容的场景，就没办法使用这种方式了 |

AB测试和上面两个不是一个范畴，不做比较。但是需要说明的一点，AB测试可以采用上面两种部署方式的手法。

**参考**

[Blue-green Deployments, A/B Testing, and Canary Releases](http://blog.christianposta.com/deploy/blue-green-deployments-a-b-testing-and-canary-releases/)

[BlueGreenDeployment](https://martinfowler.com/bliki/BlueGreenDeployment.html)