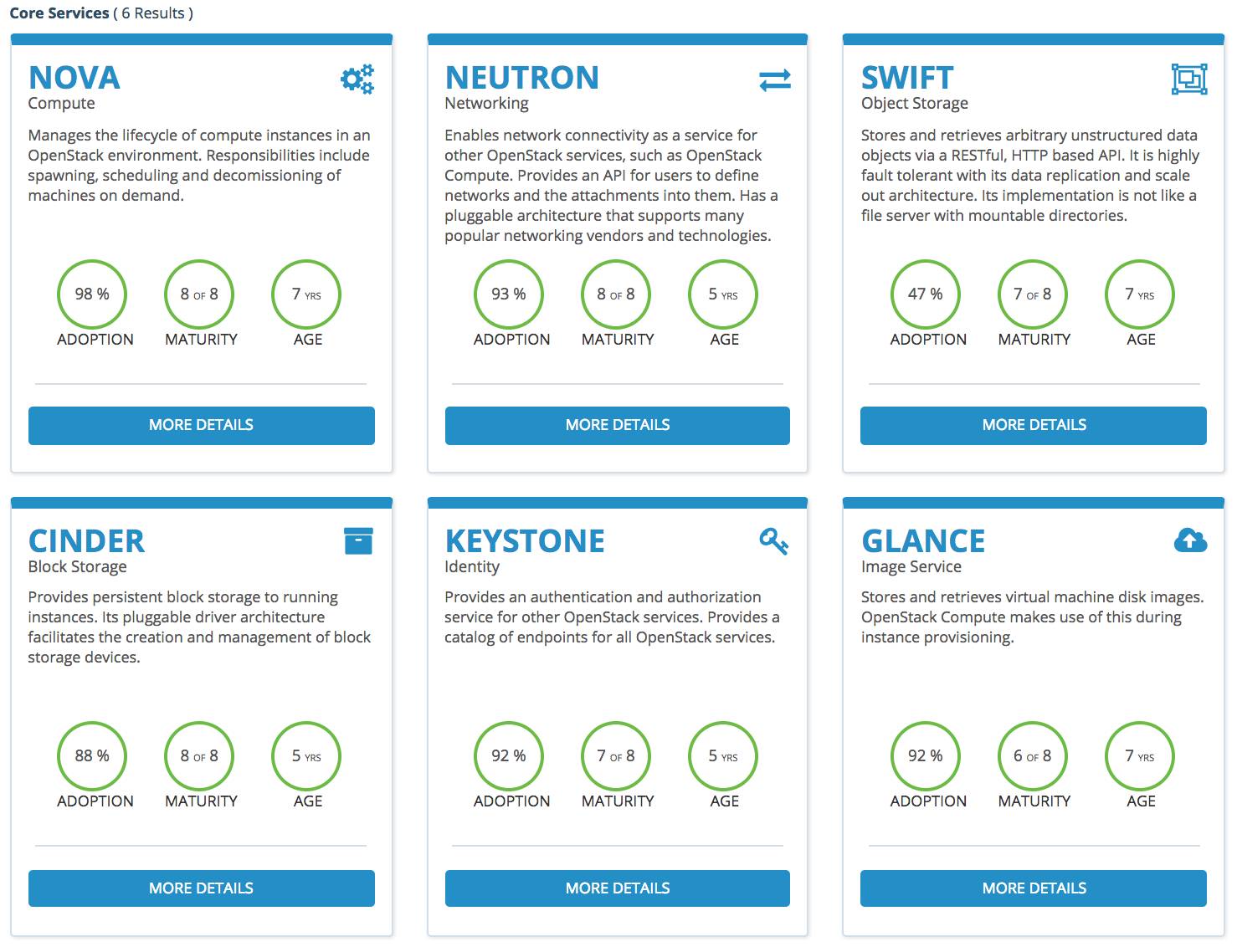


作为 OpenStack 的核心教程，我们已经到了最后总结的部分。

OpenStack 目前已经有好几十个模块，本教程讨论的是最最重要的核心模块：Keystone，Nova，Glance，Cinder 和 Neutron。请大家看下图：



此图截自 https://www.openstack.org/software/project-navigator/，这是 OpenStack 官方定义的 6 个 Core Service。每个模块都会从三个维度来衡量：

1. ADOPTION - 采用度
2. MATURITY - 成熟度
3. AGE - 年龄

ADOPTION 和 MATURITY 是最关键的两个指标。

我们先来看 ADOPTION。模块的 ADOPTION 如果能达到 90%，基本上可以说只要使用 OpenStack 就会部署此模块。

MATURITY 8 为满分，如果是 7、8 分的 MATURITY，那么这个模块的架构就非常稳定，功能也很完善了，应该说达到了上生产的水平。

其实 CloudMan 在考虑教程覆盖的技术范围时也是参考了这张图，采用度和成熟度都高的模块才是我们应该重点研究的技术。大家可能已经注意到，在 6 个 Core Service 中，Swift 没有被纳入到此教程。Swift 是对象存储，成熟度很高，在互联网行业中应用非常广泛，但在传统应用领域使用还不多。OpenStack 目前主要还是用于私有云，所以 ADOPTION 方面受到了限制，还不到 50%。这也是 CloudMan 暂时没有将它放到本教程的主要原因。

我们所处是一个知识更新迭代非常快的行业，如果两三年不学习可能就会有被淘汰的危险。学习新技术是 IT 从业者日常工作的一部分，既然选择做这行就必须要面对这个的现实。

对于如何学习新技术，CloudMan 有一些个人的建议，借这个机会跟大家交流一下。

**哪些技术应该优先学习？**

我们的时间和精力都是有限的，而新技术新知识是很多的，这时候就必须考虑一个效能问题：**用最少的时间掌握最有价值的技能**。

哪些技能最有价值呢？

我的 **第一个判断标准是：选择那些未来必将流行和大规模使用技能**。比如 OpenStack，目前已经是 IaaS 事实上的标准，社区活跃，又有大厂商积极参与，更重要的是越来越多重量级企业开始部署和实践 OpenStack，发展势头强劲。另一方面 OpenStack 的技术人才缺口很大，稀缺性决定了价值，所以我们看到了人才市场对 OpenStack 的青睐。

好，假设我们都同意 OpenStack 是应该学习的技能了（不然你也不会看这个文章了 :-），那么接下来的问题是 OpenStack 模块这么多，我要怎样学呢？是都学吗？按什么顺序学呢？这就是我的 **第二个判断标准：学那些 ADOPTION 和 MATURITY 都高的技能**。

ADOPTION 高意味着市场有需求，学了就能用。MATURITY 高意味着技术架构已经成熟和稳定，学了之后的在相当长时间内都有效，不过时。下面找两个模块具体分析一下。

先来看 Swift，前面说了，这个技术互联网企业用得多但传统企业目前应用有限，如果你就职于传统企业，花时间学习 Swift 肯定不是一件划算的投资。当然，如果你已经明确知道自己公司会用对象存储，那则是另外一回事了。

另一个模块是 Ceilometer，它属于 OpenStack optional service。请先看下图：



Ceilometer 的主要作用是收集 OpenStack 的性能数据和事件，对于运维还是非常重要的，所以我们看到它的 ADOPTION 还可以，62 %。不过成熟度太低。事实也是如此，几乎 OpenStack 的每个版本 Ceilometer 架构和代码都有很多变化，虽然是一个很有前景的模块，但我们如果花太多时间去研究，可能半年之后又得花大力气重新学习一遍，因为可能已经过时了。对于 OpenStack 的大部分技术人员，CloudMan 并不建议花太多精力在 Ceilometer 上。当然如果你的职责是要实现 OpenStack 的性能监控或者计量计费，Ceilometer 应该是绕不过去的坎。

在 optional service 有一个 ADOPTION 和 MATURITY 都还可以的模块 - Heat。这个模块是用来做编排的，其实很有价值，比如可以实现虚拟机的 auto scaling。CloudMan 如果要更新教程，Heat 肯定不会错过。

**如何学习？**

简单的说就是：**理论联系实际**。

要掌握 IT 技术说难吧，确实，像 OpenStack 这种体系结构大而复杂的，是有难度。但从另一方面看，如果轻松就能学会，我们技术人员的价值也就不大了。我没有看到哪个公司愿意花大价钱雇一个 VMware 管理员。

对于 IT 技术，最好的学习方法是边学边实践。刚开始搞不太清楚原理和底层细节没关系，先装上跑起来，能运行一个最小案例，先建立起感性认识和正反馈很重要。然后再根据文档从架构到细节，从原理到实现一步一步探索。这也是 CloudMan 在教程中尽量遵循的方法。每一章都设计了大量的实践操作环节，通过截图和日志帮助大家理解各个技术要点。在教程的编写过程为了其中一个实验我会尝试不同的场景和边界条件，最后呈现给大家的是最典型的场景。非常鼓励大家跟着做实验，说实话，仅仅看一遍教程我不觉得会有多大收获，当然如果你的目的只是了解 OpenStack 能干些什么则另当别论。

**如何使用这个教程？**

本教程对读者应该会有两个作用：

1. 初学者可以按照章节顺序系统地学习 OpenStack，并通过教程中的实验掌握 OpenStack 的理论知识和实操技能。
2. 有经验的运维人员可以将本教程当做参考材料，在实际工作中有针对性地查看相关知识点。

**继续学习的建议**

如果大家想更深入地学习 OpenStack，我的建议是：首先确保掌握本教程的知识点，因为这是基础。掌握的含义是理论能够联系实践。能部署，会操作，能看日志，会排查问题。之后嘛？我相信当你做到前面这点后就已经具备了自学其他模块或者更深入学习某个 core service 的能力了。

**对新教程的想法**

OpenStack 这个教程暂时告一段落了，之所以是“暂时”，是因为目前还没有别的模块满足我们前面的那两个选择条件。当有新模块满足条件后，CloudMan 会继续更新这个教程，目前看，比较有希望的是 Heat 和 Swift。再次说明，这并不是说其他模块就不值得学习，只是作为教程内容不适合，因为教程是面向大多数人的。如果你自己有工作或兴趣上的需求，当然可以自学。就拿 CloudMan 来说，有的客户需要计费功能，我也花了不少时间去研究 Cloudkitty 模块。

另外，核心模块中如果有一些新的有价值特性，我也会择机更新教程。

除此之外，还有什么技术满足那两个选择条件呢？如果你足够敏锐，相信会给出这个答案 - **容器**。

容器是云计算的另一大趋势，而且围绕 Docker 建立的容器生态越来越成熟，所以 CloudMan 下一个教程就叫 **《每天5分钟玩转容器技术》**。在这个教程中会涉及容器生态系统中最核心的几个方面：容器、镜像、网络、存储、编排引擎、管理平台和基于容器的 PaaS。

不过在开始这个新教程之前，CloudMan 还想跟大家分享一些个人这几年在 OpenStack 项目实践中总结和积累的实用技能和经验，作为对教程的补充。这个部分会有 10 篇左右，每篇一个技能或案例。

公众号还是每周 1、3、5 更新，感谢大家过去一年的陪伴，2017 我们继续携手前行。