

你好，我是秦粤，欢迎你和我一起学Serverless。说起Serverless这个词啊，我估计你应该不陌生，就算你没详细了解过，但我确定你肯定在过去几年时间里听别人说过。去年我在参加[GMTC全球大前端技术大会](#)的时候，也惊讶地发现国内几个大公司都已经成熟的应用案例了，所以我当时就感叹说：“Serverless终于要飞入寻常百姓家了。”

三个问题

作为一名Serverless的拥趸者，过去几年时间里，我总是喜欢向朋友和同事推荐Serverless技术。不过，在“推销”的过程中，他们经常会问我一些问题，那在今天的开篇词里，我就来统一回答下这些共性的问题吧，我估计你也会问到。

问题一：说来说去，到底Serverless要解决什么问题？

我不知道你有没有算过你们公司每年在服务器上的开销，反正创业之前我是没算过，总觉得这钱算了也没必要，还是应该多花心思在怎么挣钱上，但后面当我真正自己开公司之后，才知道柴米贵。

咱们就拿自己部署一套博客来说吧，常见的Node.js MVC架构，需要购买云服务商的Linux虚拟机、RDS关系型数据库，做得好的话还要购买Redis缓存、负载均衡、CDN等等，再专业一点，可能还会考虑容灾和备份，这么算下来一年最小开销都在1万元左右。但如果你用Serverless的话，这个成本可以直接降到1000元以下。

Serverless是对运维体系的极端抽象，就像iPhone当年颠覆诺基亚一样，它给应用开发和部署提供了一个极简模型。这种高度抽象的模型，可以让一个零运维经验的人，几分钟就部署一个Web应用上线，并对外提供服务。也就是说，你根本不需要再学习怎么在Linux上装Web服务器，怎么配置负载均衡等等这些繁琐的偏运维方向的工作。

所以，你要问我Serverless解决了什么问题，一句话总结就是它可以帮你省钱、省力气。

问题二：为什么阿里巴巴、腾讯这样的公司都在关注Serverless？

首先，Serverless可以有效降低企业中中长尾应用的运营成本。中长尾应用就是那些每天大部分时间都没有流量或者有很少流量的应用，你可以想想你们公司是不是也有很多。这一点我特别有感触，尤其是企业在落地微服务架构后，一些边缘的微服务被调用的概率其实很低。而这个时候，我们往往又很难通过人工来控制中长尾应用，因为这里面不少应用还是被强依赖的，不可以直接下线处理。Serverless之前，这些中长尾应用至少要独占1台虚拟机；现在有了Serverless的极速冷启动特性，企业就可以节省这部分开销。

其次，Serverless可以提高研发效能。我们专栏会讲到Serverless应用架构的设计，其中，SF（Serverless For Frontend）可以让前端同学自行负责数据接口的编排，微服务BaaS化则让我们的后端同学更加关注领域设计。可以说，这是一个颠覆性的变化，它能够进一步放大前端工程师的价值。

最后我想说Serverless作为一门新兴技术，未来的想象空间很大。我看到有创业公司用FaaS来做基础设施编排和云服务编排；也有外包公司利用Serverless应用架构的快速迭代能力，提升开发效率，降低出错率，还可以给自己沉淀领域的解决方案；还有包括风险投资方也在逐渐开始关注Serverless领域，毕竟这这也是一个新的风口。我讲大企业的使用方式只是希望你一些灵感，不想过多限制你的想象，Serverless的疆域边界还在等你去扩展。

这里是GMTC会议上几个大公司的分享资料，你感兴趣的话，可以先看看。

- 阿里跨境供应链前端架构演进与 Serverless 实践
- Serverless 前端工程化落地与实践
- 从前端和云厂商的视角看 Serverless 与未来的开发生态

问题三：Serverless对前端工程师来说会有什么机遇？为什么我们要学习Serverless？

相对其他工种而言，Serverless给前端工程师带来的机遇更大，它能让前端工程师也享受到云服务的红利。如果说Node.js语言的出现解放了一波前端工程师的生产力，那Node.js+Serverless又可以进一步激发前端工程师的创造力。口说无凭，到时候你看完咱们专栏里的例子就能体会到这句话的意思了。

另外，我觉得学习Serverless是成为云开发者的最适合的切入点。无论你是零基础，还是资深服务端运维，都可以从Serverless上学习到现代服务端运维体系的重要思想（我在第一节课程就会讲）。

回答完这几个问题，我想你对Serverless已经有了初步了解。接下来我也不想绕弯子，我就再和你聊聊我准备怎么给你讲这门课吧。

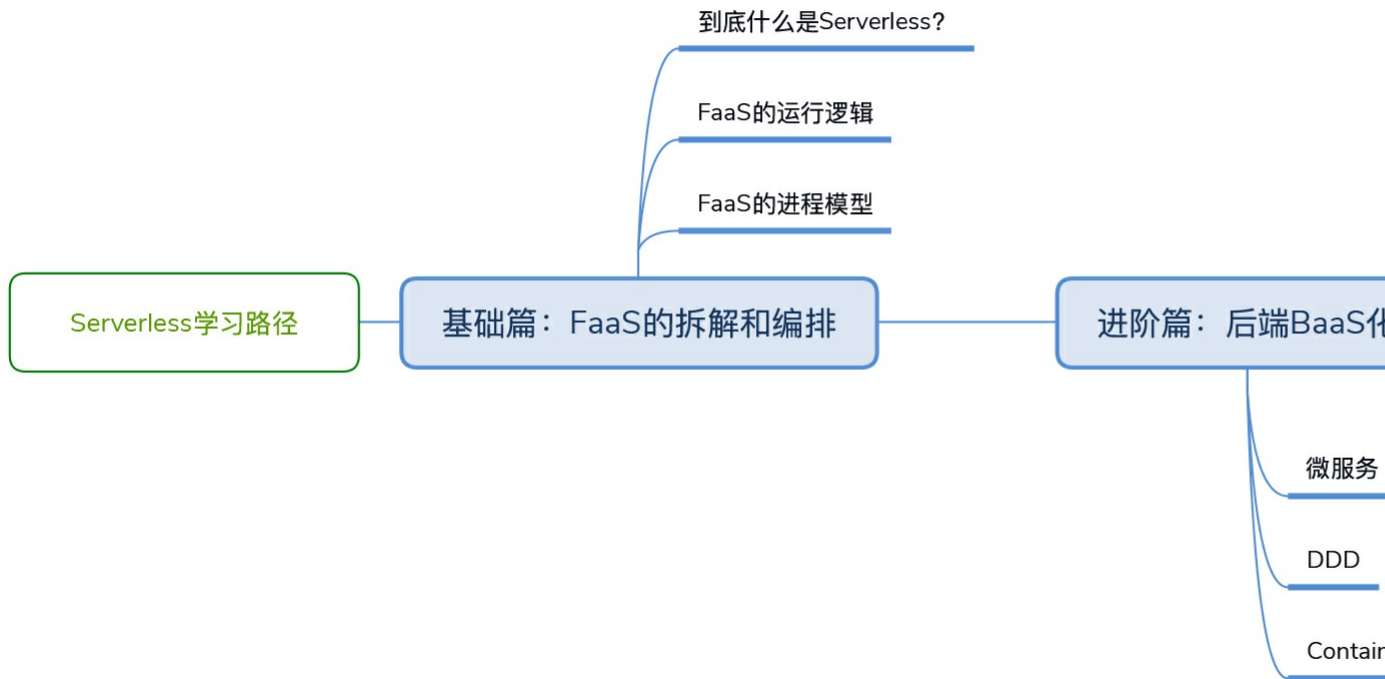
课程设计

基础篇，我会继续带你理解Serverless要解决什么问题，以及Serverless的边界和定义。搞清楚来龙去脉，我们会进入动手环节，我会通过一个例子来给你讲解Serverless引擎盖下的工作原理，以及FaaS的一些应用场景。

进阶篇，我们将一起学习FaaS的后端解决方案BaaS，以及我们自己现有的后端应用如何BaaS化。为了更好地展现Serverless的发展历程和背后的思考，我也为你准备了一个基于Node.js的待办任务的Web应用，你要做好准备，这里我会给你布置很多动手作业。

GitHub地址：<https://github.com/pusongyang/todoist-backend>

实战篇，我会通过Google开源的Kubernetes向你演示本地化Serverless环境如何搭建，并根据我的经验，和你聊聊Serverless架构应该如何选型，以及目前Serverless开发的最佳实践。



最后，我再介绍一下我自己吧。我叫蒲松洋，秦粤是我的花名。2006年从电子科技大学毕业后，我就进入了UT斯达康（现在这家公司已经谢幕，它是小灵通的主要生产厂商）做通讯相关的工作，当时的职位是PHP前端工程师。2013年，我跳槽加入百度，从PHP前端工程师转为了Node.js前端工程师。2015年开始又和朋友折腾创业，用Node.js做智能家居IoT。2016年底，创业没成，我又回到了国内某一线互联网公司，负责Node.js应用治理和Node.js微服务架构设计。

我在用Node.js做微服务时发现，微服务本身提出了很多理念，但微服务在服务端运维却缺少给力的支撑平台，后来我们就试着用容器集群搭建了自己的Container Serverless。结果证明它不但可以支撑Node.js微服务运维，还可以提高Node.js中长尾应用的资源利用率。再后面，大家都意识到了Serverless带来的便利性，于是我们团队也就整体参与到了公司Serverless整体建设中了。

在研究并落地Serverless技术的过程中，我发现国外的Serverless开源社区其实比国内更加活跃。自从2014年AWS推出第一个FaaS服务Lambda后，国外的很多公司都在积极推进Serverless的生态发展，并且开始占领高地，制定了很多的规范。而放眼国内，目前还只有为数不多的大型互联网公司在重点跟进，其他人基本上只是在观望或者看热闹。

因此，我也特别希望通过这个专栏能够带你真正理解Serverless，并让你尽快享受到技术的红利。Serverless肯定是未来云计算发展的重点方向，作为工程师，特别是前端工程师，我们应该思考的是如何抓住这波机遇，如何利用FaaS+各种创意，组合碰撞出各种化学反应，去为公司、为自己创造更大的价值。

以上就是今天的全部内容。有关Serverless，不知道你的看法是怎样的？如果你有什么疑惑，或者在实践中遇到了哪些困难？欢迎在留言区中提出，我们共同探讨。

你好，我是秦粤，欢迎你和我一起学习交流Serverless。说起Serverless这个词啊，我估计你应该不陌生，就算你没详细了解过，但我确定你肯定在过去几年时间里听别人说过。去年我在参加GMTC全球大前端技术大会的时候，也惊讶地发现国内几个大公司都已经有了成熟的应用案例了，所以我当时就感叹说：“Serverless终于要飞入寻常百姓家了。”

三个问题

作为一名Serverless的拥趸者，过去几年时间里，我总是喜欢向朋友和同事推荐Serverless技术。不过，在“推销”的过程中，他们经常会问我一些问题，那在今天的开篇词里，我就来统一回答下这些共性的问题吧，我估计你也会问到。

问题一：说来说去，到底Serverless要解决什么问题？

我不知道你有没有算过你们公司每年在服务器上的开销，反正在创业之前我是没算过，总觉得这钱算了也没必要，还是应该多花心思在怎么挣钱上，但后面当我真正自己开公司之后，才知道柴米贵。

咱们就拿自己部署一套博客来说吧，常见的Node.js MVC架构，需要购买云服务商的Linux虚拟机、RDS关系型数据库，做得好的话还要购买Redis缓存、负载均衡、CDN等等，再专业一点，可能还会考虑容灾和备份，这么算下来一年最小开销都在1万元左右。但如果你用Serverless的话，这个成本可以直接降到1000元以下。

Serverless是对运维体系的极端抽象，就像iPhone当年颠覆诺基亚一样，它给应用开发和部署提供了一个极简模型。这种高度抽象的模型，可以让一个零运维经验的人，几分钟就部署一个Web应用上线，并对外提供服务。也就是说，你根本不需要再学习怎么在Linux上装Web服务器，怎么配置负载均衡等等这些繁琐的偏运维方向的工作。

所以，你要问我Serverless解决了什么问题，一句话总结就是它可以帮你省钱、省力气。

问题二：为什么阿里巴巴、腾讯这样的公司都在关注Serverless？

首先，Serverless可以有效降低企业中中长尾应用的运营成本。中长尾应用就是那些每天大部分时间都没有流量或者有很少流量的应用，你可以想想你们公司是不是也有很多。这一点我特别有感触，尤其是企业在落地微服务架构后，一些边缘的微服务被调用的概率其实很低。而这个时候，我们往往又很难通过人工来控制中长尾应用，因为这里面不少应用还是被强依赖的，不可以直接下线处理。Serverless之前，这些中长尾应用至少要独占1台虚拟机；现在有了Serverless的极速冷启动特性，企业就可以节省这部分开销。

其次，Serverless可以提高研发效能。我们专栏会讲到Serverless应用架构的设计，其中，SFF（Serverless For Frontend）可以让前端同学自行负责数据接口的编排，微服务BaaS化则让我们的后端同学更加关注领域设计。可以说，这是一个颠覆性的变化，它能够进一步放大前端工程师的价值。

最后我想说Serverless作为一门新兴技术，未来的想象空间很大。我看到有创业公司用FaaS来做基础设施编排和云服务编排；也有外包公司利用Serverless应用架构的快速迭代能力，提升开发效率，降低出错率，还可以给自己沉淀领域的解决方案；还有包括风险投资方也在逐渐开始关注Serverless领域，毕竟这也是一个新的风口。我讲大企业的使用方式只是希望你一些灵感，不想过多限制你的想象，Serverless的疆域边界还在等你去扩展。

这里是GMTC会议上几个大公司的分享资料，你感兴趣的话，可以先看看。

- [阿里跨境供应链前端架构演进与 Serverless 实践](#)
- [Serverless 前端工程化落地与实践](#)
- [从前端和云厂商的视角看 Serverless 与未来的开发生态](#)

问题三：Serverless对前端工程师来说会有什么机遇？为什么我们要学习Serverless？

相对其他工种而言，Serverless给前端工程师带来的机遇更大，它能让前端工程师也享受到云服务的红利。如果说Node.js语言的出现解放了一波前端工程师的生产力，那Node.js+Serverless又可以进一步激发前端工程师的创造力。口说无凭，到时候你看完咱们专栏里的例子就能体会到这句话的意思了。

另外，我觉得学习Serverless是成为云开发者的最适合的切入点。无论你是零基础，还是资深服务端运维，都可以从Serverless上学习到现代服务端运维体系的重要思想（我在第一节课就会讲）。

回答完这几个问题，我想你对Serverless已经有了初步了解。接下来我也不想绕弯子，我就再和你聊聊我准备怎么给你讲这门课吧。

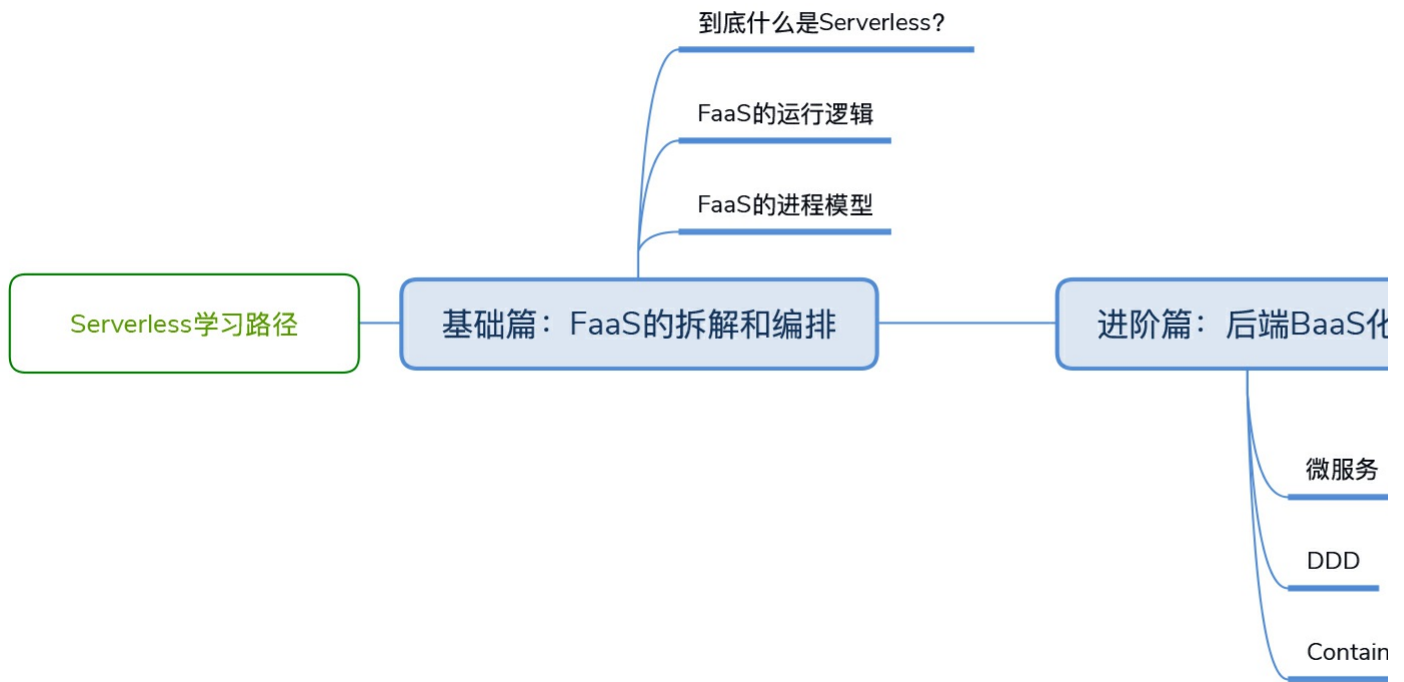
课程设计

基础篇，我会继续带你理解Serverless要解决什么问题，以及Serverless的边界和定义。搞清楚了来龙去脉，我们会进入动手环节，我会通过一个例子来给你讲解Serverless引擎盖下的工作原理，以及FaaS的一些应用场景。

进阶篇，我们将一起学习FaaS的后端解决方案BaaS，以及我们自己现有的后端应用如何BaaS化。为了更好地展现Serverless的发展历程和背后的思考，我也为你准备了一个基于Node.js的待办任务的Web应用，你要做好准备，这里我会给你布置很多动手作业。

GitHub地址：<https://github.com/pusongyang/todolist-backend>

实战篇，我会通过Google开源的Kubernetes向你演示本地化Serverless环境如何搭建，并根据我的经验，和你聊聊Serverless架构应该如何选型，以及目前Serverless开发的最佳实践。



最后，我再来介绍一下我自己吧。我叫蒲松洋，秦粤是我的花名。2006年从电子科技大学毕业后，我就进入了UT斯达康（现在这公司已经谢幕，它是小灵通的主要生产厂商）做通讯相关的工作，当时的职位是PHP前端工程师。2013年，我跳槽加入百度，从PHP前端工程师转为了Node.js前端工程师。2015年开始又和朋友折腾创业，用Node.js做智能家居IoT。2016年底，创业没成，我又回到了国内某一线互联网公司，负责Node.js应用治理和Node.js微服务架构设计。

我在用Node.js做微服务时发现，微服务本身提出了很多理念，但微服务在服务端运维却缺少给力的支撑平台，后来我们就试着用容器集群搭建了自己的Container Serverless。结果证明它不但可以支撑Node.js微服务运维，还可以提高Node.js中长尾应用的资源利用率。再后面，大家都意识到了Serverless带来的便利性，于是我们团队也就整体参与到了公司Serverless整体建设中了。

在研究并落地Serverless技术的过程中，我发现国外的Serverless开源社区其实比国内更加活跃。自从2014年AWS推出第一个FaaS服务Lambda后，国外的很多公司都在积极推进Serverless的生态发展，并且开始占领高地，制定了很多的规范。而放眼国内，目前还只有为数不多的大型互联网公司在重点跟进，其他人基本上只是在观望或者看热闹。

因此，我也特别希望通过这个专栏能够带你真正理解Serverless，并让你尽快享受到技术的红利。Serverless肯定是未来云计算发展的重点方向，作为工程师，特别是前端工程师，我们应该思考的是如何抓住这波机遇，如何利用FaaS+各种创意，组合碰撞出各种化学反应，去为公司、为自己创造更大的价值。

以上就是今天的全部内容。有关Serverless，不知道你的看法是怎样的？如果你有什么疑惑，或者在实践中遇到了哪些困难？欢迎在留言区中提出，我们共同探讨。

你好，我是秦粤，欢迎你和我一起学Serverless。说起Serverless这个词啊，我估计你应该不陌生，就算你没详细了解过，但我确定你肯定在过去几年时间里听别人说过。去年我在参加[GMTC全球大前端技术大会](#)的时候，也惊讶地发现国内几个大公司都已经有了成熟的应用案例了，所以我当时就感叹说：“Serverless终于要飞入寻常百姓家了。”

三个问题

作为一名Serverless的拥趸者，过去几年时间里，我总是喜欢向朋友和同事推荐Serverless技术。不过，在“推销”的过程中，他们经常会问我一些问题，那在今天的开篇词里，我就来统一回答下这些共性的问题吧，我估计你也会问到。

问题一：说来说去，到底Serverless要解决什么问题？

我不知道你有没有算过你们公司每年在服务器上的开销，反正在创业之前我是没算过，总觉得这钱算了也没必要，还是应该多花心思在怎么挣钱上，但后面当我真正自己开公司之后，才知道柴米贵。

咱们就拿自己部署一套博客来说吧，常见的Node.js MVC架构，需要购买云服务商的Linux虚拟机、RDS关系型数据库，做得好的话还要购买Redis缓存、负载均衡、CDN等等，再专业一点，可能还会考虑容灾和备份，这么算下来一年最小开销都在1万元左右。但如果你用Serverless的话，这个成本可以直接降到1000元以下。

Serverless是对运维体系的极端抽象，就像iPhone当年颠覆诺基亚一样，它给应用开发和部署提供了一个极简模型。这种高度抽象的模型，可以让一个零运维经验的人，几分钟就部署一个Web应用上线，并对外提供服务。也就是说，你根本不需要再学习怎么在Linux上装Web服务器，怎么配置负载均衡等等这些繁琐的偏运维方向的工作。

所以，你要问我Serverless解决了什么问题，一句话总结就是它可以帮你省钱、省力气。

问题二：为什么阿里巴巴、腾讯这样的公司都在关注Serverless？

首先，Serverless可以有效降低企业中中长尾应用的运营成本。中长尾应用就是那些每天大部分时间都没有流量或者有很少流量的应用，你可以想想你们公司是不是也有很多。这一点我特别有感触，尤其是企业在落地微服务架构后，一些边缘的微服务被调用的概率其实很低。而这个时候，我们往往又很难通过人工来控制中长尾应用，因为这里面不少应用还是被强依赖的，不可以直接下线处理。Serverless之前，这些中长尾应用至少要独占1台虚拟机；现在有了Serverless的极速冷启动特性，企业就可以节省这部分开销。

其次，Serverless可以提高研发效能。我们专栏会讲到Serverless应用架构的设计，其中，SFF（Serverless For Frontend）可以让前端同学自行负责数据接口的编排，微服务BaaS化则让我们的后端同学更加关注领域设计。可以说，这是一个颠覆性的变化，它能够进一步放大前端工程师的价值。

最后我想说Serverless作为一门新兴技术，未来的想象空间很大。我看到有创业公司用FaaS来做基础设施编排和云服务编排；也有外包公司利用Serverless应用架构的快速迭代能力，提升开发效率，降低出错率，还可以给自己沉淀领域的解决方案；还有包括风险投资方也在逐渐开始关注Serverless领域，毕竟这也是一个新的风口。我讲大企业的使用方式只是希望给你一些灵感，不想过多限制你的想象，Serverless的疆域边界还在等你去扩展。

这里是GMTC会议上几个大公司的分享资料，你感兴趣的话，可以先看看。

- [阿里跨境供应链前端架构演进与 Serverless 实践](#)
- [Serverless 前端工程化落地与实践](#)
- [从前端和云厂商的视角看 Serverless 与未来的开发生态](#)

问题三：Serverless对前端工程师来说会有什么机遇？为什么我们要学习Serverless？

相对其他工种而言，Serverless给前端工程师带来的机遇更大，它能让前端工程师也享受到云服务的红利。如果说Node.js语言的出现解放了一波前端工程师的生产力，那Node.js+Serverless又可以进一步激发前端工程师的创造力。口说无凭，到时候你看完咱们专栏里的例子就能体会到这句话的意思了。

另外，我觉得学习Serverless是成为云开发者的最适合的切入点。无论你是零基础，还是资深服务端运维，都可以从Serverless上学习到现代服务端运维体系的重要思想（我在第一节就会讲）。

回答完这几个问题，我想你对Serverless已经有了初步了解。接下来我也不想绕弯子，我就再和你聊聊我准备怎么给你讲这门课吧。

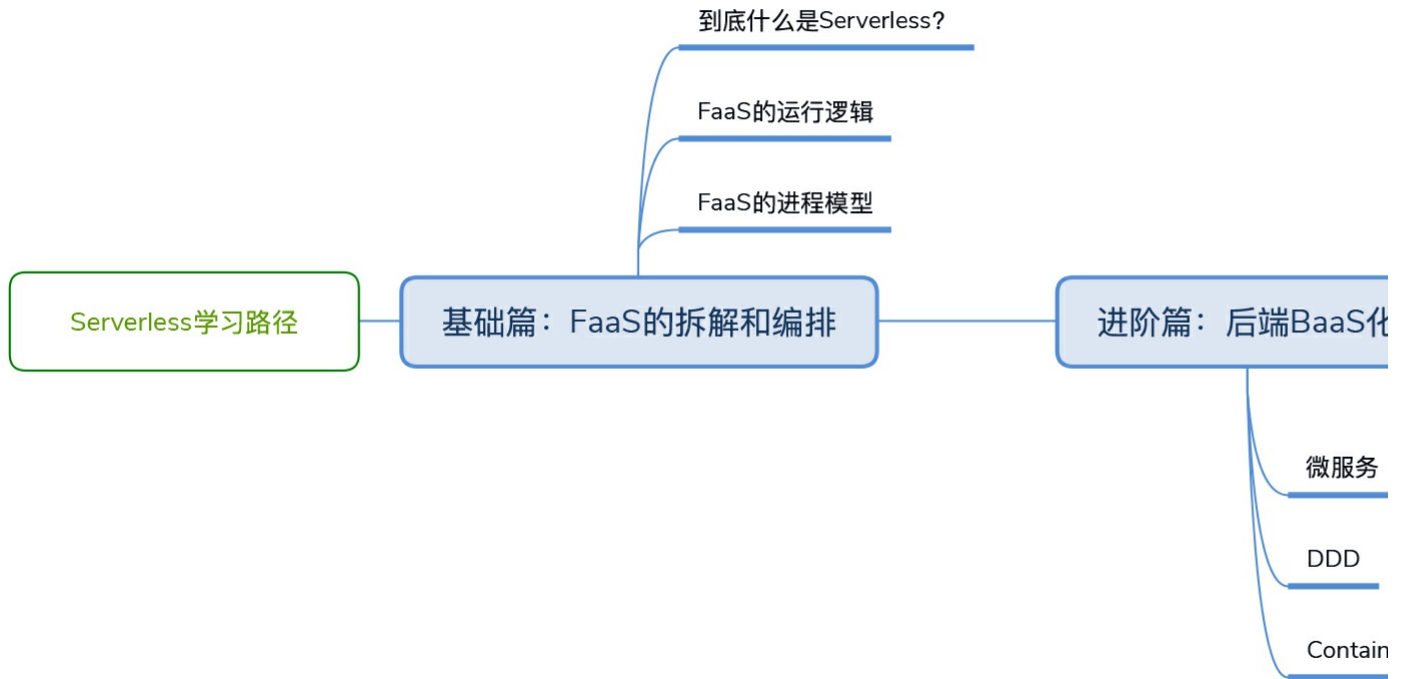
课程设计

基础篇，我会继续带你理解Serverless要解决什么问题，以及Serverless的边界和定义。搞清楚来了龙去脉，我们会进入动手环节，我会通过一个例子来给你讲解Serverless引擎盖下的工作原理，以及FaaS的一些应用场景。

进阶篇，我们将一起学习FaaS的后端解决方案BaaS，以及我们自己现有的后端应用如何BaaS化。为了更好地展现Serverless的发展历程和背后的思考，我也为你准备了一个基于Node.js的待办任务的Web应用，你要做好准备，这里我会给你布置很多动手作业。

GitHub地址：<https://github.com/pusongyang/todoist-backend>

实战篇，我会通过Google开源的Kubernetes向你演示本地化Serverless环境如何搭建，并根据我的经验，和你聊聊Serverless架构应该如何选型，以及目前Serverless开发的最佳实践。



最后，我再介绍一下我自己吧。我叫蒲松洋，秦粤是我的花名。2006年从电子科技大学毕业后，我就进入了UT斯达康（现在这家公司已经谢幕，它是小灵通的主要生产厂商）做通讯相关的工作，当时的职位是PHP前端工程师。2013年，我跳槽加入百度，从PHP前端工程师转为了Node.js前端工程师。2015年开始又和朋友折腾创业，用Node.js做智能家居IoT。2016年底，创业没成，我又回到了国内某一线互联网公司，负责Node.js应用治理和Node.js微服务架构设计。

我在用Node.js做微服务时发现，微服务本身提出了很多理念，但微服务在服务端运维却缺少给力的支撑平台，后来我们就试着用容器集群搭建了自己的Container Serverless。结果证明它不但可以支撑Node.js微服务运维，还可以提高Node.js中长尾应用的资源利用率。再后面，大家都意识到了Serverless带来的便利性，于是我们团队也就整体参与到了公司Serverless整体建设中了。

在研究并落地Serverless技术的过程中，我发现国外的Serverless开源社区其实比国内更加活跃。自从2014年AWS推出第一个FaaS服务Lambda后，国外的很多公司都在积极推进Serverless的生态发展，并且开始占领高地，制定了很多的规范。而放眼国内，目前还只有为数不多的大型互联网公司在重点跟进，其他人基本上只是在观望或者看热闹。

因此，我也特别希望通过这个专栏能够带你真正理解Serverless，并让你尽快享受到技术的红利。Serverless肯定是未来云计算发展的重点方向，作为工程师，特别是前端工程师，我们应该思考的是如何抓住这波机遇，如何利用FaaS+各种创意，组合碰撞出各种化学反应，去为公司、为自己创造更大的价值。

以上就是今天的全部内容。有关Serverless，不知道你的看法是怎样的？如果你有什么疑惑，或者在实践中遇到了哪些困难？欢迎在留言区中提出，我们共同探讨。

你好，我是秦粤，欢迎你和我一起學習Serverless。说起Serverless这个词啊，我估计你应该不陌生，就算你没详细了解过，但我确定你肯定在过去几年时间里听别人说过。去年我在参加GMTC全球大前端技术大会的时候，也惊讶地发现国内几个大公司都已经有了成熟的应用案例了，所以我当时就感叹说：“Serverless终于要飞入寻常百姓家了。”

三个问题

作为一名Serverless的拥趸者，过去几年时间里，我总是喜欢向朋友和同事推荐Serverless技术。不过，在“推销”的过程中，他们经常会问我一些问题，那在今天的开篇词里，我就来统一回答下这些共性的问题吧，我估计你也会问到。

问题一：说来说去，到底Serverless要解决什么问题？

我不知道你有没有算过你们公司每年在服务器上的开销，反正在创业之前我是没算过，总觉得这钱算了也没必要，还是应该多花心思在怎么挣钱上，但后面当我真正自己开公司之后，才知道柴米贵。

咱们就拿自己部署一套博客来说吧，常见的Node.js MVC架构，需要购买云服务商的Linux虚拟机、RDS关系型数据库，做得好的话还要购买Redis缓存、负载均衡、CDN等等，再专业一点，可能还会考虑容灾和备份，这么算下来一年最小开销都在1万元左右。但如果你用Serverless的话，这个成本可以直接降到1000元以下。

Serverless是对运维体系的极端抽象，就像iPhone当年颠覆诺基亚一样，它给应用开发和部署提供了一个极简模型。这种高度抽象的模型，可以让一个零运维经验的人，几分钟就部署一个Web应用上线，并对外提供服务。也就是说，你根本不需要再学习怎么在Linux上装Web服务器，怎么配置负载均衡等等这些繁琐的偏运维方向的工作。

所以，你要问我Serverless解决了什么问题，一句话总结就是它可以帮你省钱、省力气。

问题二：为什么阿里巴巴、腾讯这样的公司都在关注Serverless？

首先，Serverless可以有效降低企业中中长尾应用的运营成本。中长尾应用就是那些每天大部分时间都没有流量或者有很少流量的应用，你可以想想你们公司是不是也有很多。这一点我特别有感触，尤其是企业在落地微服务架构后，一些边缘的微服务被调用的概率其实很低。而这个时候，我们往往又很难通过人工来控制中长尾应用，因为这里面不少应用还是被强依赖的，不可以直接下线处理。Serverless之前，这些中长尾应用至少要独占1台虚拟机；现在有了Serverless的极速冷启动特性，企业就可以节省这部分开销。

其次，Serverless可以提高研发效能。我们专栏会讲到Serverless应用架构的设计，其中，SFF（Serverless For Frontend）可以让前端同学自行负责数据接口的编排，微服务BaaS化则让我们的后端同学更加关注领域设计。可以说，这是一个颠覆性的变化，它能够进一步放大前端工程师的价值。

最后我想说Serverless作为一门新兴技术，未来的想象空间很大。我看到有创业公司用FaaS来做基础设施编排和云服务编排；也有外包公司利用Serverless应用架构的快速迭代能力，提升开发效率，降低出错率，还可以给自己沉淀领域的解决方案；还有包括风险投资方也在逐渐开始关注Serverless领域，毕竟这也是一个新的风口。我讲大企业的使用方式只是希望给你一些灵感，不想过多限制你的想象，Serverless的疆域边界还在等你去扩展。

这里是GMTC会议上几个大公司的分享资料，你感兴趣的话，可以先看看。

- [阿里跨境供应链前端架构演进与 Serverless 实践](#)

- [Serverless 前端工程化落地与实践](#)
- [从前端和云厂商的视角看 Serverless 与未来的开发生态](#)

问题三：Serverless对前端工程师来说会有什么机遇？为什么我们要学习Serverless？

相对其他工种而言，Serverless给前端工程师带来的机遇更大，它能让前端工程师也享受到云服务的红利。如果说Node.js语言的出现解放了一波前端工程师的生产力，那Node.js+Serverless又可以进一步激发前端工程师的创造力。口说无凭，到时候你看完咱们专栏里的例子就能体会到这句话的意思了。

另外，我觉得学习Serverless是成为云开发者的最适合的切入点。无论你是零基础，还是资深服务端运维，都可以从Serverless上学习到现代服务端运维体系的重要思想（我在第一节就会讲）。

回答完这几个问题，我想你对Serverless已经有了初步了解。接下来我也不想绕弯子，我就再和你聊聊我准备怎么给你讲这门课吧。

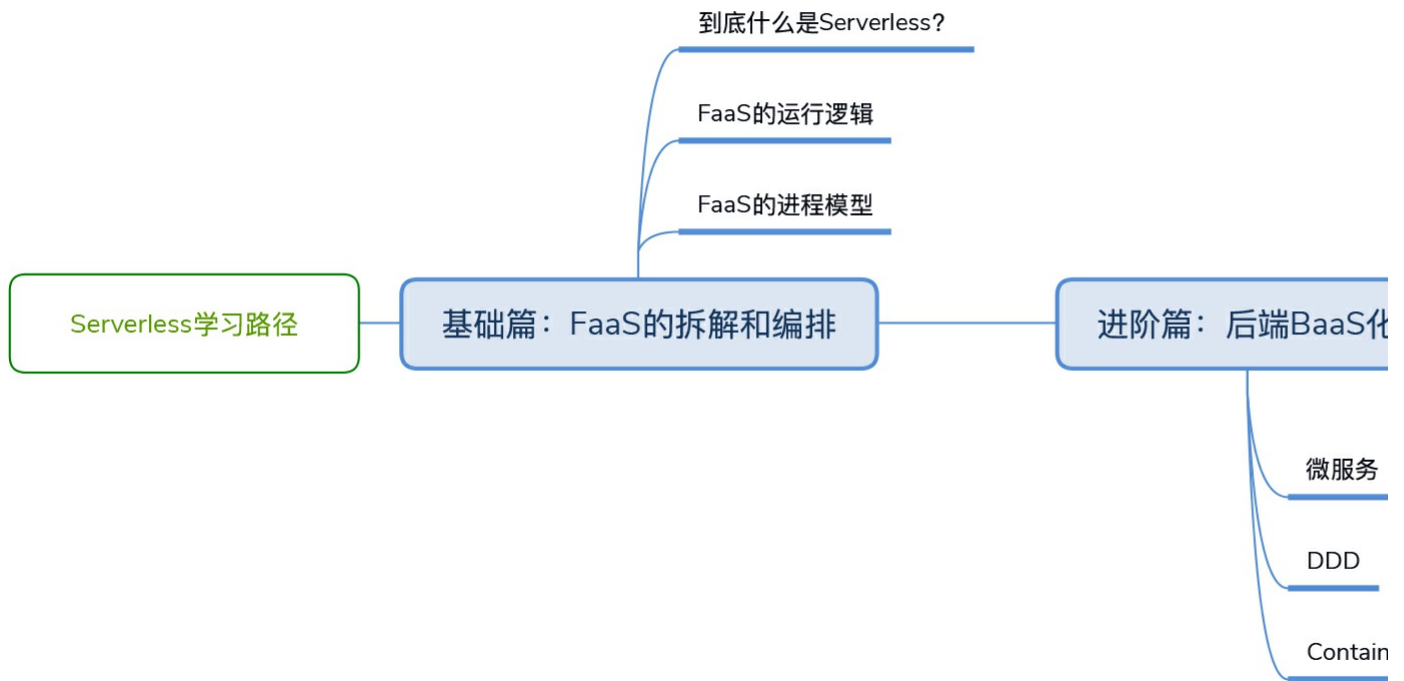
课程设计

基础篇，我会继续带你理解Serverless要解决什么问题，以及Serverless的边界和定义。搞清楚了来龙去脉，我们会进入动手环节，我会通过一个例子来给你讲解Serverless引擎盖下的工作原理，以及FaaS的一些应用场景。

进阶篇，我们将一起学习FaaS的后端解决方案BaaS，以及我们自己现有的后端应用如何BaaS化。为了更好地展现Serverless的发展历程和背后的思考，我也为你准备了一个基于Node.js的待办任务的Web应用，你要做好准备，这里我会给你布置很多动手作业。

GitHub地址：<https://github.com/pusongyang/todoist-backend>

实战篇，我会通过Google开源的Kubernetes向你演示本地化Serverless环境如何搭建，并根据我的经验，和你聊聊Serverless架构应该如何选型，以及目前Serverless开发的最佳实践。



最后，我再来介绍下我自己吧。我叫蒲松洋，秦粤是我的花名。2006年从电子科技大学毕业后，我就进入了UT斯达康（现在这家公司已经谢幕，它是小灵通的主要生产厂商）做通讯相关的工作，当时的职位是PHP前端工程师。2013年，我跳槽加入百度，从PHP前端工程师转为了Node.js前端工程师。2015年开始又和朋友折腾创业，用Node.js做智能家居IoT。2016年底，创业没成，我又回到了国内某一线互联网公司，负责Node.js应用治理和Node.js微服务架构设计。

我在用Node.js做微服务时发现，微服务本身提出了很多理念，但微服务在服务端运维却缺少给力的支撑平台，后来我们就试着用容器集群搭建了自己的Container Serverless。结果证明它不但可以支撑Node.js微服务运维，还可以提高Node.js中长尾应用的资源利用率。再后面，大家都意识到了Serverless带来的便利性，于是我们团队也就整体参与到了公司Serverless整体建设中了。

在研究并落地Serverless技术的过程中，我发现国外的Serverless开源社区其实比国内更加活跃。自从2014年AWS推出第一个FaaS服务Lambda后，国外的很多公司都在积极推进Serverless的生态发展，并且开始占领高地，制定了很多的规范。而放眼国内，目前还只有为数不多的大型互联网公司在重点跟进，其他人基本上只是在观望或者看热闹。

因此，我也特别希望通过这个专栏能够带你真正理解Serverless，并让你尽快享受到技术的红利。Serverless肯定是未来云计算发展的重点方向，作为工程师，特别是前端工程师，我们应该思考的是如何抓住这波机遇，如何利用FaaS+各种创意，组合碰撞出各种化学反应，去为公司、为自己创造更大的价值。

以上就是今天的全部内容。有关Serverless，不知道你的看法是怎样的？如果你有什么疑惑，或者在实践中遇到了哪些困难？欢迎在留言区中提出，我们共同探讨。

你好，我是秦粤，欢迎你和我一起学Serverless。说起Serverless这个词啊，我估计你应该不陌生，就算你没详细了解过，但我确定你肯定在过去几年时间里听别人说过。去年我在参加GMTC全球大前端技术大会的时候，也惊讶地发现国内几个大公司都已经有了成熟的应用案例了，所以我当时就感叹说：“Serverless终于要飞入寻常百姓家了。”

三个问题

作为一名Serverless的拥趸者，过去几年时间里，我总是喜欢向朋友和同事推荐Serverless技术。不过，在“推销”的过程中，他们经常会问我一些问题，那在今天的开篇词里，我就来统一回答下这些共性的问题吧，我估计你也会问到。

问题一：说来说去，到底Serverless要解决什么问题？

我不知道你有没有算过你们公司每年在服务上的开销，反正在创业之前我是没算过，总觉得这钱算了也没必要，还是应该多花心思在怎么挣钱上，但后面当我真正自己开公司之后，才知道柴米贵。

咱们就拿自己部署一套博客来说吧，常见的Node.js MVC架构，需要购买云服务商的Linux虚拟机、RDS关系型数据库，做得好的话还要购买Redis缓存、负载均衡、CDN等等，再专业一点，可能还会考虑容灾和备份，这么算下来一年最小开销都在1万元左右。但如果你用Serverless的话，这个成本可以直接降到1000元以下。

Serverless是对运维体系的极端抽象，就像iPhone当年颠覆诺基亚一样，它给应用开发和部署提供了一个极简模型。这种高度抽象的模型，可以让一个零运维经验的人，几分钟就部署一个Web应用上线，并对外提供服务。也就是说，你根本不需要再学习怎么在Linux上装Web服务器，怎么配置负载均衡等等这些繁琐的偏运维方向的工作。

所以，你要问我Serverless解决了什么问题，一句话总结就是它可以帮你省钱、省力气。

问题二：为什么阿里巴巴、腾讯这样的公司都在关注Serverless？

首先，Serverless可以有效降低企业中中长尾应用的运营成本。中长尾应用就是那些每天大部分时间都没有流量或者有很少流量的应用，你可以想想你们公司是不是也有很多。这一点我特别有感触，尤其是企业在落地微服务架构后，一些边缘的微服务被调用的概率其实很低。而这个时候，我们往往又很难通过人工来控制中长尾应用，因为这里面不少应用还是被强依赖的，不可以直接下线处理。

Serverless之前，这些中长尾应用至少要独占1台虚拟机；现在有了Serverless的极速冷启动特性，企业就可以节省这部分开销。

其次，Serverless可以提高研发效能。我们专栏会讲到Serverless应用架构的设计，其中，SFF（Serverless For Frontend）可以让前端同学自行负责数据接口的编排，微服务BaaS化则让我们的后端同学更加关注领域设计。可以说，这是一个颠覆性的变化，它能够进一步放大前端工程师的价值。

最后我想说Serverless作为一门新兴技术，未来的想象空间很大。我看到有创业公司用FaaS来做基础设施编排和云服务编排；也有外包公司利用Serverless应用架构的快速迭代能力，提升开发效率，降低出错率，还可以给自己沉淀领域的解决方案；还有包括风险投资方也在逐渐开始关注Serverless领域，毕竟这也是一个新的风口。我讲大企业的使用方式只是希望给你一些灵感，不想过多限制你的想象，Serverless的疆域边界还在等你去扩展。

这里是GMTC会议上几个大公司的分享资料，你感兴趣的话，可以先看看。

- [阿里跨境供应链前端架构演进与 Serverless 实践](#)
- [Serverless 前端工程化落地与实践](#)
- [从前端和云厂商的视角看 Serverless 与未来的开发生态](#)

问题三：Serverless对前端工程师来说会有什么机遇？为什么我们要学习Serverless？

相对其他工种而言，Serverless给前端工程师带来的机遇更大，它能让前端工程师也享受到云服务的红利。如果说Node.js语言的出现解放了一波前端工程师的生产力，那Node.js+Serverless又可以进一步激发前端工程师的创造力。口说无凭，到时候你看完咱们专栏里的例子就能体会到这句话的意思了。

另外，我觉得学习Serverless是成为云开发者的最适合的切入点。无论你是零基础，还是资深服务端运维，都可以从Serverless上学习到现代服务端运维体系的重要思想（我在第一节就会讲）。

回答完这几个问题，我想你对Serverless已经有了初步了解。接下来我也不想绕弯子，我就再和你聊聊我准备怎么给你讲这门课吧。

课程设计

基础篇，我会继续带你理解Serverless要解决什么问题，以及Serverless的边界和定义。搞清楚来了龙去脉，我们会进入动手环节，我会通过一个例子来给你讲解Serverless引擎盖下的工作原理，以及FaaS的一些应用场景。

进阶篇，我们将一起学习FaaS的后端解决方案BaaS，以及我们自己现有的后端应用如何BaaS化。为了更好地展现Serverless的发展历程和背后的思考，我也为你准备了一个基于Node.js的待办任务的Web应用，你要做好准备，这里我会给你布置很多动手作业。

GitHub地址：<https://github.com/pusongyang/todolist-backend>

实战篇，我会通过Google开源的Kubernetes向你演示本地化Serverless环境如何搭建，并根据我的经验，和你聊聊Serverless架构应该如何选型，以及目前Serverless开发的最佳实践。



最后，我再来介绍下我自己吧。我叫蒲松洋，秦粤是我的花名。2006年从电子科技大学毕业后，我就进入了UT斯达康（现在这公司已经谢幕，它是小灵通的主要生产厂商）做通讯相关的工作，当时的职位是PHP前端工程师。2013年，我跳槽加入百度，从PHP前端工程师转为了Node.js前端工程师。2015年开始又和朋友折腾创业，用Node.js做智能家居IoT。2016年底，创业没成，我又回到了国内某一线互联网公司，负责Node.js应用治理和Node.js微服务架构设计。

我在用Node.js做微服务时发现，微服务本身提出了很多理念，但微服务在服务端运维却缺少给力的支撑平台，后来我们就试着用容器集群搭建了自己的Container Serverless。结果证明它不但可以支撑Node.js微服务运维，还可以提高Node.js中长尾应用的资源利用率。再后面，大家都意识到了Serverless带来的便利性，于是我们团队也就整体参与到了公司Serverless整体建设中了。

在研究并落地Serverless技术的过程中，我发现国外的Serverless开源社区其实比国内更加活跃。自从2014年AWS推出第一个FaaS服务Lambda后，国外的很多公司都在积极推进Serverless的生态发展，并且开始占领高地，制定了很多的规范。而放眼国内，目前还只有为数不多的大型互联网公司在重点跟进，其他人基本上只是在观望或者看热闹。

因此，我也特别希望通过这个专栏能够带你真正理解Serverless，并让你尽快享受到技术的红利。Serverless肯定是未来云计算发展的重点方向，作为工程师，特别是前端工程师，我们应该思考的是如何抓住这波机遇，如何利用FaaS+各种创意，组合碰撞出各种化学反应，去为公司、为自己创造更大的价值。

以上就是今天的全部内容。有关Serverless，不知道你的看法是怎样的？如果你有什么疑惑，或者在实践中遇到了哪些困难？欢迎在留言区中提出，我们共同探讨。

你好，我是秦粤，欢迎你和我一起学Serverless。说起Serverless这个词啊，我估计你应该不陌生，就算你没详细了解过，但我确定你肯定在过去几年时间里听别人说过。去年我在参加GMTC全球前端技术大会的时候，也惊讶地发现国内几个大公司都已经有了成熟的应用案例了，所以我当时就感叹说：“Serverless终于要飞入寻常百姓家了。”

三个问题

作为一名Serverless的拥趸者，过去几年时间里，我总是喜欢向朋友和同事推荐Serverless技术。不过，在“推销”的过程中，他们经常会问我一些问题，那在今天的开篇词里，我就来统一回答下这些共性的问题吧，我估计你也会问到。

问题一：说来说去，到底Serverless要解决什么问题？

我不知道你有没有算过你们公司每年在服务器上的开销，反正在创业之前我是没算过，总觉得这钱算了也没必要，还是应该多花心思在怎么挣钱上，但后面当我真正自己开公司之后，才知道柴米贵。

咱们就拿自己部署一套博客来说吧，常见的Node.js MVC架构，需要购买云服务商的Linux虚拟机、RDS关系型数据库，做得好的话还要购买Redis缓存、负载均衡、CDN等等，再专业一点，可能还会考虑容灾和备份，这么算下来一年最小开销都在1万元左右。但如果你用Serverless的话，这个成本可以直接降到1000元以下。

Serverless是对运维体系的极端抽象，就像iPhone当年颠覆诺基亚一样，它给应用开发和部署提供了一个极简模型。这种高度抽象的模型，可以让一个零运维经验的人，几分钟就部署一个Web应用上线，并对外提供服务。也就是说，你根本不需要再学习怎么在Linux上装Web服务器，怎么配置负载均衡等等这些繁琐的偏运维方向的工作。

所以，你要问我Serverless解决了什么问题，一句话总结就是它可以帮你省钱、省力气。

问题二：为什么阿里巴巴、腾讯这样的公司都在关注Serverless？

首先，Serverless可以有效降低企业中中长尾应用的运营成本。中长尾应用就是那些每天大部分时间都没有流量或者有很少流量的应用，你可以想想你们公司是不是也有很多。这一点我特别有感触，尤其是企业在落地微服务架构后，一些边缘的微服务被调用的概率其实很低。而这个时候，我们往往又很难通过人工来控制中长尾应用，因为这里面不少应用还是被强依赖的，不可以直接下线处理。Serverless之前，这些中长尾应用至少要独占1台虚拟机；现在有了Serverless的极速冷启动特性，企业就可以节省这部分开销。

其次，Serverless可以提高研发效能。我们专栏会讲到Serverless应用架构的设计，其中，SF（Serverless For Frontend）可以让前端同学自行负责数据接口的编排，微服务BaaS化则让我们的后端同学更加关注领域设计。可以说，这是一个颠覆性的变化，它能够进一步放大前端工程师的价值。

最后我想说Serverless作为一门新兴技术，未来的想象空间很大。我看到有创业公司用FaaS来做基础设施编排和云服务编排；也有外包公司利用Serverless应用架构的快速迭代能力，提升开发效率，降低出错率，还可以给自己沉淀领域的解决方案；还有包括风险投资方也在逐渐开始关注Serverless领域，毕竟这这也是一个新的风口。我讲大企业的使用方式只是希望你一些灵感，不想过多限制你的想象，Serverless的疆域边界还在等你去扩展。

这里是GMTC会议上几个大公司的分享资料，你感兴趣的话，可以先看看。

- [阿里跨境供应链前端架构演进与 Serverless 实践](#)
- [Serverless 前端工程化落地与实践](#)
- [从前端和云厂商的视角看 Serverless 与未来的开发生态](#)

问题三：Serverless对前端工程师来说会有什么机遇？为什么我们要学习Serverless？

相对其他工种而言，Serverless给前端工程师带来的机遇更大，它能让前端工程师也享受到云服务的红利。如果说Node.js语言的出现解放了一波前端工程师的生产力，那Node.js+Serverless又可以进一步激发前端工程师的创造力。口说无凭，到时候你看完咱们专栏里的例子就能体会到这句话的意思了。

另外，我觉得学习Serverless是成为云开发者的最适合的切入点。无论你是零基础，还是资深服务端运维，都可以从Serverless上学习到现代服务端运维体系的重要思想（我在第一节课程就会讲）。

回答完这几个问题，我想你对Serverless已经有了初步了解。接下来我也不想绕弯子，我就再和你聊聊我准备怎么给你讲这门课吧。

课程设计

基础篇，我会继续带你理解Serverless要解决什么问题，以及Serverless的边界和定义。搞清楚了来龙去脉，我们会进入动手环节，我会通过一个例子来给你讲解Serverless引擎盖下的工作原理，以及FaaS的一些应用场景。

进阶篇，我们将一起学习FaaS的后端解决方案BaaS，以及我们自己现有的后端应用如何BaaS化。为了更好地展现Serverless的发展历程和背后的思考，我也为你准备了一个基于Node.js的待办任务的Web应用，你要做好准备，这里我会给你布置很多动手作业。

GitHub地址：<https://github.com/pusongyang/todoist-backend>

实战篇，我会通过Google开源的Kubernetes向你演示本地化Serverless环境如何搭建，并根据我的经验，和你聊聊Serverless架构应该如何选型，以及目前Serverless开发的最佳实践。



最后，我再来介绍下我自己吧。我叫蒲松洋，秦粤是我的花名。2006年从电子科技大学毕业后，我就进入了UT斯达康（现在在这公司已经谢幕，它是小灵通的主要生产厂商）做通讯相关的工作，当时的职位是PHP前端工程师。2013年，我跳槽加入百度，从PHP前端工程师转为了Node.js前端工程师。2015年开始又和朋友折腾创业，用Node.js做智能家居IoT。2016年底，创业没成，我又回到了国内某一线互联网公司，负责Node.js应用治理和Node.js微服务架构设计。

我在用Node.js做微服务时发现，微服务本身提出了很多理念，但微服务在服务端运维却缺少给力的支撑平台，后来我们就试着用容器集群搭建了自己的Container Serverless。结果证明它不但可以支撑Node.js微服务运维，还可以提高Node.js中长尾应用的资源利用率。再后面，大家都意识到了Serverless带来的便利性，于是我们团队也就整体参与到了公司Serverless整体建设中了。

在研究并落地Serverless技术的过程中，我发现国外的Serverless开源社区其实比国内更加活跃。自从2014年AWS推出第一个FaaS服务Lambda后，国外的很多公司都在积极推进Serverless的生态发展，并且开始占领高地，制定了很多的规范。而放眼国内，目前还只有为数不多的大型互联网公司在重点跟进，其他人基本上只是在观望或者看热闹。

因此，我也特别希望通过这个专栏能够带你真正理解Serverless，并让你尽快享受到技术的红利。Serverless肯定是未来云计算发展的重点方向，作为工程师，特别是前端工程师，我们应该思考的是如何抓住这波机遇，如何利用FaaS+各种创意，组合碰撞出各种化学反应，去为公司、为自己创造更大的价值。

以上就是今天的全部内容。有关Serverless，不知道你的看法是怎样的？如果你有什么疑惑，或者在实践中遇到了哪些困难？欢迎在留言区中提出，我们共同探讨。

你好，我是秦粤，欢迎你和我一起学Serverless。说起Serverless这个词啊，我估计你应该不陌生，就算你没详细了解过，但我确定你肯定在过去几年时间里听别人说过。去年我在参加GMTC全球前端技

木大会的时候，也惊讶地发现国内几个大公司都已经有成熟的应用案例了，所以我当时就感叹说：“Serverless终于要飞入寻常百姓家了。”

三个问题

作为一名Serverless的拥趸者，过去几年时间里，我总是喜欢向朋友和同事推荐Serverless技术。不过，在“推销”的过程中，他们经常会问我一些问题，那在今天的开篇词里，我就来统一回答下这些共性的问题吧，我估计你也会问到。

问题一：说来说去，到底Serverless要解决什么问题？

我不知道你有没有算过你们公司每年在服务器上的开销，反正创业之前我是没算过，总觉得这钱算了也没必要，还是应该多花心思在怎么挣钱上，但后面当我真正自己开公司之后，才知道柴米贵。

咱们就拿自己部署一套博客来说吧，常见的Node.js MVC架构，需要购买云服务商的Linux虚拟机、RDS关系型数据库，做得好的话还要购买Redis缓存、负载均衡、CDN等等，再专业一点，可能还会考虑容灾和备份，这么算下来一年最小开销都在1万元左右。但如果你用Serverless的话，这个成本可以直接降到1000元以下。

Serverless是对运维体系的极端抽象，就像iPhone当年颠覆诺基亚一样，它给应用开发和部署提供了一个极简模型。这种高度抽象的模型，可以让一个零运维经验的人，几分钟就部署一个Web应用上线，并对外提供服务。也就是说，你根本不需要再学习怎么在Linux上装Web服务器，怎么配置负载均衡等等这些繁琐的偏运维方向的工作。

所以，你要问我Serverless解决了什么问题，一句话总结就是它可以帮你省钱、省力气。

问题二：为什么阿里巴巴、腾讯这样的公司都在关注Serverless？

首先，Serverless可以有效降低企业中中长尾应用的运营成本。中长尾应用就是那些每天大部分时间都没有流量或者有很少流量的应用，你可以想想你们公司是不是也有很多。这一点我特别有感触，尤其是企业在落地微服务架构后，一些边缘的微服务被调用的概率其实很低。而这个时候，我们往往又很难通过人工来控制中长尾应用，因为这里面不少应用还是被强依赖的，不可以直接下线处理。Serverless之前，这些中长尾应用至少要独占1台虚拟机；现在有了Serverless的极速冷启动特性，企业就可以节省这部分开销。

其次，Serverless可以提高研发效能。我们专栏会讲到Serverless应用架构的设计，其中，SFF（Serverless For Frontend）可以让前端同学自行负责数据接口的编排，微服务BaaS化则让我们的后端同学更加关注领域设计。可以说，这是一个颠覆性的变化，它能够进一步放大前端工程师的价值。

最后我想说Serverless作为一门新兴技术，未来的想象空间很大。我看到有创业公司用FaaS来做基础设施编排和云服务编排；也有外包公司利用Serverless应用架构的快速迭代能力，提升开发效率，降低出错率，还可以给自己沉淀领域的解决方案；还有包括风险投资方也在逐渐开始关注Serverless领域，毕竟这也是一个新的风口。我讲大企业的使用方式只是希望你一些灵感，不想过多限制你的想象，Serverless的疆域边界还在等你去扩展。

这里是GMTC会议上几个大公司的分享资料，你感兴趣的话，可以先看看。

- [阿里跨境供应链前端架构演进与 Serverless 实践](#)
- [Serverless 前端工程化落地与实践](#)
- [从前端和云厂商的视角看 Serverless 与未来的开发生态](#)

问题三：Serverless对前端工程师来说会有什么机遇？为什么我们要学习Serverless？

相对其他工种而言，Serverless给前端工程师带来的机遇更大，它能让前端工程师也享受到云服务的红利。如果说Node.js语言的出现解放了一波前端工程师的生产力，那Node.js+Serverless又可以进一步激发前端工程师的创造力。口说无凭，到时候你看咱们专栏里的例子就能体会到这句话的意思了。

另外，我觉得学习Serverless是成为云开发者的最适合的切入点。无论你是零基础，还是资深服务端运维，都可以从Serverless上学习到现代服务端运维体系的重要思想（我在第一节课就会讲）。

回答完这几个问题，我想你对Serverless已经有了初步了解。接下来我也不想绕弯子，我就再和你聊聊我准备怎么给你讲这门课吧。

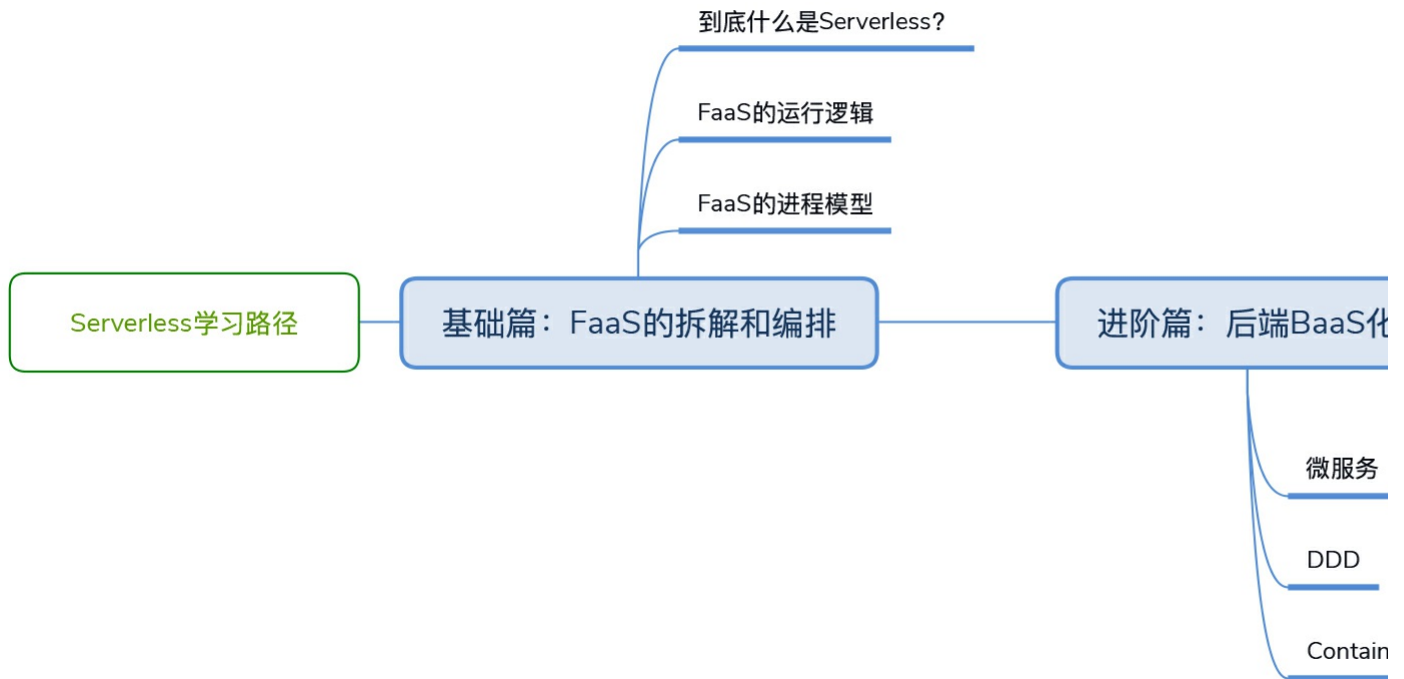
课程设计

基础篇，我会继续带你理解Serverless要解决什么问题，以及Serverless的边界和定义。搞清楚了来龙去脉，我们会进入动手环节，我会通过一个例子来给你讲解Serverless引擎盖下的工作原理，以及FaaS的一些应用场景。

进阶篇，我们将一起学习FaaS的后端解决方案BaaS，以及我们自己现有的后端应用如何BaaS化。为了更好地展现Serverless的发展历程和背后的思考，我也为你准备了一个基于Node.js的待办任务的Web应用，你要做好准备，这里我会给你布置很多动手作业。

GitHub地址：<https://github.com/pusongyang/todolist-backend>

实战篇，我会通过Google开源的Kubernetes向你演示本地化Serverless环境如何搭建，并根据我的经验，和你聊聊Serverless架构应该如何选型，以及目前Serverless开发的最佳实践。



最后，我再来介绍下我自己吧。我叫蒲松洋，秦粤是我的花名。2006年从电子科技大学毕业后，我就进入了UT斯达康（现在这家公司已经谢幕，它是小灵通的主要生产厂商）做通讯相关的工作，当时的职位是PHP前端工程师。2013年，我跳槽加入百度，从PHP前端工程师转为了Node.js前端工程师。2015年开始又和朋友折腾创业，用Node.js做智能家居IoT。2016年底，创业没成，我又回到了国内某一线互联网公司，负责Node.js应用治理和Node.js微服务架构设计。

我在用Node.js做微服务时发现，微服务本身提出了很多理念，但微服务在服务端运维却缺少给力的支撑平台，后来我们就试着用容器集群搭建了自己的Container Serverless。结果证明它不但可以支撑Node.js微服务运维，还可以提高Node.js中长尾应用的资源利用率。再后面，大家都意识到了Serverless带来的便利性，于是我们团队也就整体参与到了公司Serverless整体建设中了。

在研究并落地Serverless技术的过程中，我发现国外的Serverless开源社区其实比国内更加活跃。自从2014年AWS推出第一个FaaS服务Lambda后，国外的很多公司都在积极推进Serverless的生态发展，并且开始占领高地，制定了很多的规范。而放眼国内，目前还只有为数不多的大型互联网公司在重点跟进，其他人基本上只是在观望或者看热闹。

因此，我也特别希望通过这个专栏能够带你真正理解Serverless，并让你尽快享受到技术的红利。Serverless肯定是未来云计算发展的重点方向，作为工程师，特别是前端工程师，我们应该思考的是如何抓住这波机遇，如何利用FaaS+各种创意，组合碰撞出各种化学反应，去为公司、为自己创造更大的价值。

以上就是今天的全部内容。有关Serverless，不知道你的看法是怎样的？如果你有什么疑惑，或者在实践中遇到了哪些困难？欢迎在留言区中提出，我们共同探讨。

你好，我是秦粤，欢迎你和我一起學習Serverless。说起Serverless这个词啊，我估计你应该不陌生，就算你没详细了解过，但我确定你肯定在过去几年时间里听别人说过。去年我在参加GMTC全球大前端技术大会的时候，也惊讶地发现国内几个大公司都已经有了成熟的应用案例了，所以我当时就感叹说：“Serverless终于要飞入寻常百姓家了。”

三个问题

作为一名Serverless的拥趸者，过去几年时间里，我总是喜欢向朋友和同事推荐Serverless技术。不过，在“推销”的过程中，他们经常会问我一些问题，那在今天的开篇词里，我就来统一回答下这些共性的问题吧，我估计你也会问到。

问题一：说来说去，到底Serverless要解决什么问题？

我不知道你有没有算过你们公司每年在服务器上的开销，反正创业之前我是没算过，总觉得这钱算了也没必要，还是应该多花心思在怎么挣钱上，但后面当我真正自己开公司之后，才知道柴米贵。

咱们就拿自己部署一套博客来说吧，常见的Node.js MVC架构，需要购买云服务商的Linux虚拟机、RDS关系型数据库，做得好的话还要购买Redis缓存、负载均衡、CDN等等，再专业一点，可能还会考虑容灾和备份，这么算下来一年最小开销都在1万元左右。但如果你用Serverless的话，这个成本可以直接降到1000元以下。

Serverless是对运维体系的极端抽象，就像iPhone当年颠覆诺基亚一样，它给应用开发和部署提供了一个极简模型。这种高度抽象的模型，可以让一个零运维经验的人，几分钟就部署一个Web应用上线，并对外提供服务。也就是说，你根本不需要再学习怎么在Linux上装Web服务器，怎么配置负载均衡等等这些繁琐的偏运维方向的工作。

所以，你要问我Serverless解决了什么问题，一句话总结就是它可以帮你省钱、省力气。

问题二：为什么阿里巴巴、腾讯这样的公司都在关注Serverless？

首先，Serverless可以有效降低企业中中长尾应用的运营成本。中长尾应用就是那些每天大部分时间都没有流量或者有很少流量的应用，你可以想想你们公司是不是也有很多。这一点我特别有感触，尤其是企业在落地微服务架构后，一些边缘的微服务被调用的概率其实很低。而这个时候，我们往往又很难通过人工来控制中长尾应用，因为这里面不少应用还是被强依赖的，不可以直接下线处理。Serverless之前，这些中长尾应用至少要独占1台虚拟机；现在有了Serverless的极速冷启动特性，企业就可以节省这部分开销。

其次，Serverless可以提高研发效能。我们专栏会讲到Serverless应用架构的设计，其中，SFF（Serverless For Frontend）可以让前端同学自行负责数据接口的编排，微服务BaaS化则让我们的后端同学更加关注领域设计。可以说，这是一个颠覆性的变化，它能够进一步放大前端工程师的价值。

最后我想说Serverless作为一门新兴技术，未来的想象空间很大。我看到有创业公司用FaaS来做基础设施编排和云服务编排；也有外包公司利用Serverless应用架构的快速迭代能力，提升开发效率，降低出错率，还可以给自己沉淀领域的解决方案；还有包括风险投资方也在逐渐开始关注Serverless领域，毕竟这也是一个新的风口。我讲大企业的使用方式只是希望你一些灵感，不想过多限制你的想象，Serverless的疆域边界还在等你去扩展。

这里是GMTC会议上几个大公司的分享资料，你感兴趣的话，可以先看看。

- [阿里跨境供应链前端架构演进与 Serverless 实践](#)
- [Serverless 前端工程化落地与实践](#)
- [从前端和云厂商的视角看 Serverless 与未来的开发生态](#)

问题三：Serverless对前端工程师来说会有什么机遇？为什么我们要学习Serverless？

相对其他工种而言，Serverless给前端工程师带来的机遇更大，它能让前端工程师也享受到云服务的红利。如果说Node.js语言的出现解放了一波前端工程师的生产力，那Node.js+Serverless又可以进一步激发前端工程师的创造力。口说无凭，到时候你看完咱们专栏里的例子就能体会到这句话的意思了。

另外，我觉得学习Serverless是成为云开发者的最适合的切入点。无论你是零基础，还是资深服务端运维，都可以从Serverless上学习到现代服务端运维体系的重要思想（我在第一节课就会讲）。

回答完这几个问题，我想你对Serverless已经有了初步了解。接下来我也不想绕弯子，我就再和你聊聊我准备怎么给你讲这门课吧。

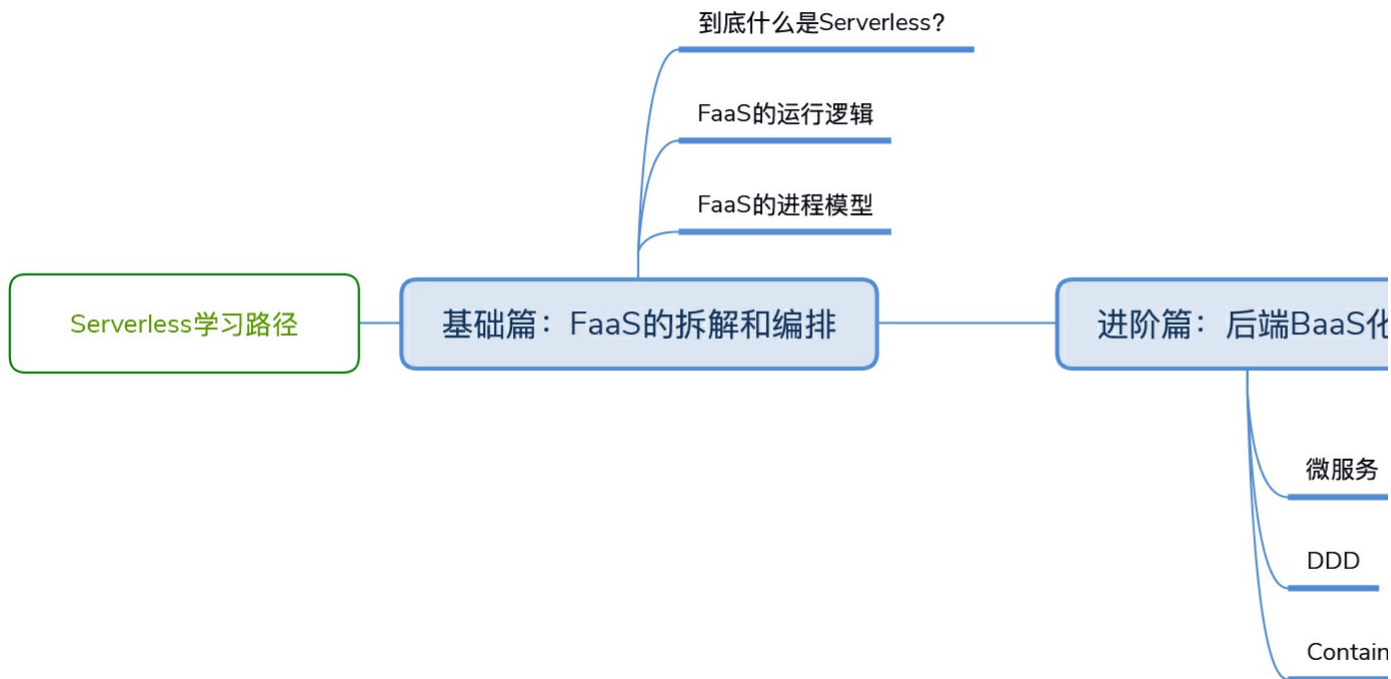
课程设计

基础篇，我会继续带你理解Serverless要解决什么问题，以及Serverless的边界和定义。搞清楚了来龙去脉，我们会进入动手环节，我会通过一个例子来给你讲解Serverless引擎盖下的工作原理，以及FaaS的一些应用场景。

进阶篇，我们将一起学习FaaS的后端解决方案BaaS，以及我们自己现有的后端应用如何BaaS化。为了更好地展现Serverless的发展历程和背后的思考，我也为你准备了一个基于Node.js的待办任务的Web应用，你要做好准备，这里我会给你布置很多动手作业。

GitHub地址：<https://github.com/pusongyang/todolist-backend>

实战篇，我会通过Google开源的Kubernetes向你演示本地化Serverless环境如何搭建，并根据我的经验，和你聊聊Serverless架构应该如何选型，以及目前Serverless开发的最佳实践。



最后，我再来介绍下我自己吧。我叫蒲松洋，秦粤是我的花名。2006年从电子科技大学毕业后，我就进入了UT斯达康（现在这家公司已经谢幕，它是小灵通的主要生产厂商）做通讯相关的工作，当时的职位是PHP前端工程师。2013年，我跳槽加入百度，从PHP前端工程师转为了Node.js前端工程师。2015年开始又和朋友折腾创业，用Node.js做智能家居IoT。2016年底，创业没成，我又回到了国内某一线互联网公司，负责Node.js应用治理和Node.js微服务架构设计。

我在用Node.js做微服务时发现，微服务本身提出了很多理念，但微服务在服务端运维却缺少给力的支撑平台，后来我们就试着用容器集群搭建了自己的Container Serverless。结果证明它不但可以支撑Node.js微服务运维，还可以提高Node.js中长尾应用的资源利用率。再后面，大家都意识到了Serverless带来的便利性，于是我们团队也就整体参与到了公司Serverless整体建设中了。

在研究并落地Serverless技术的过程中，我发现国外的Serverless开源社区其实比国内更加活跃。自从2014年AWS推出第一个FaaS服务Lambda后，国外的很多公司都在积极推进Serverless的生态发展，并且开始占领高地，制定了很多的规范。而放眼国内，目前还只有为数不多的大型互联网公司在重点跟进，其他人基本上只是在观望或者看热闹。

因此，我也特别希望通过这个专栏能够带你真正理解Serverless，并让你尽快享受到技术的红利。Serverless肯定是未来云计算发展的重点方向，作为工程师，特别是前端工程师，我们应该思考的是如何抓住这波机遇，如何利用FaaS+各种创意，组合碰撞出各种化学反应，去为公司、为自己创造更大的价值。

以上就是今天的全部内容。有关Serverless，不知道你的看法是怎样的？如果你有什么疑惑，或者在实践中遇到了哪些困难？欢迎在留言区中提出，我们共同探讨。