

本文由 [简悦 SimpRead](#) 转码，原文地址 [kaiwu.lagou.com](#)

上一讲我们讨论了在开发时提升编码效率的相关工具。在项目开发过程中，我们可以编写自定义的代码片段缩写规则来提升个人的编码效率。那么，如果想要把这些规则分享给团队内的其他成员或自己的其他电脑设备时又该怎么做呢？带着这个问题，我们就进入到今天要聊的主题：**云开发**（Cloud Development）。

和之前介绍的适用于个人的开发提效工具不同，云开发的优势主要是在团队效率的提升方面。在这一讲里，我会介绍云开发和普通开发模式的区别，目前典型的云开发产品，以及云开发模式的应用场景。下面我就进入到第一个问题：什么是云开发。

## 软件开发环境的对比

在具体介绍什么是云开发之前，让我们先来看下普通的日常开发流程是什么样的。

### 个人电脑开发环境

通常我们都使用个人电脑作为日常的开发环境，对应的开发流程通常是：

1. **基础环境准备**：在个人电脑中准备开发环境所需设施，下载安装开发所需各种**应用程序**（以前端为例：IDE、Node、Git、Yarn.....），调试各种**配置文件**（.bashrc、npmrc.....），安装必要 **IDE 插件**并调试 **IDE 配置项**（UI、编码格式、Snippets.....）等。
2. 下载代码：将项目源代码从代码仓库（例如 Git Repo）中下载到个人电脑的开发目录下。
3. 安装项目依赖。
4. 运行开发服务。
5. 编码调试。
6. 执行任务（Lint 检查、格式化检查、单元测试等）。

上述开发流程的流畅度一定程度上依赖于所使用电脑的硬件配置，因此程序员往往需要高性能配置的个人电脑。

### 远程开发

远程开发是将开发环境部署到远程服务器，然后通过个人电脑的 IDE（Integrated Development Environment，集成开发环境）进行远程连接来进行开发的方式（例如通过 VS Code 中的 [Remote SSH 插件](#)）。和传统的个人电脑开发环境相比，远程开发模式的优点主要在于：

1. 由远程的开发服务器来承载项目数据存储和运行计算的需求，从而解放了对个人电脑资源的占用和对性能的要求。
2. 由于个人电脑只提供了用户操作界面和远程连接的能力，因此大大减少了访问设备变更对于项目开发的影响，例如在更换新电脑或在家使用不同电脑设备的情况下，只需要安装 IDE 和少量配置，就能快速获得一致的开发体验，而无须再重复进行基础环境的准备。

远程开发的主要问题在于：

1. 需要申请单独的开发机资源。
2. 新申请的开发机也需要和个人电脑环境一样，人工进行基础环境的准备工作。
3. 将开发机单独用于远程开发，在资源分配上可能存在资源利用不充分的问题。

### 云开发

云开发模式是在上述远程开发模式的基础之上发展而来的，将开发环境托管，由远程开发服务器变更为云服务。个人电脑通过 IDE 或云服务提供的浏览器界面访问云端工作区进行开发。云开发模式在继承远程开发模式优点的基础上，更能提升效率的原因在于：

1. 通过容器化技术，将开发环境所需基础设施（应用程序、配置文件、IDE 插件、IDE 设定项等）集成到基础镜像中，大大**提升开发环境准备的效率**。同时，同样的基础环境也避免了相同项目不同开发集成环境导致的环境差异类问题。
2. 通过服务化的云开发平台，**简化使用流程**，解决个人使用远程开发时可能遇到的技术困难，使得刚入职的新人也能够快速上手。
3. 对于团队而言，能够**提升团队协作效率**。云开发模式有利于流程规范的统一，有利于团队成员共享开发工具，同时支持多人访问相同开发环境，有助于结对编程等协作流程。
4. 对于公司而言，使用弹性化的云端容器环境有利于资源利用率的提升和硬件资产成本的降低。

## 典型云开发产品介绍

以下表格是一些已经推出的云开发产品，感兴趣的话，你可以根据自己所接触过的云厂商来进一步了解。

产品	厂商	基础 IDE	IDE 类型	代码托管方式
VS Codespace	微软	VS Code	Web/VS/VSC	云端（Asure） / 自维护
Gitpod	Eclipse	Theia	Web/Desktop	云端 / 自维护（限制用户数量）
CloudIDE	阿里云	KAITIAN IDE	Web	云端
Cloud Studio	Coding.net（腾讯云）	VS Code	Web	云端（5 个工作空间）
Cloud9	AWS	Cloud9	Web	云端（AWS）

本课时重点介绍的是以 VS Codespace 为代表的云开发产品，以及以 Theia 为代表的 WebIDE 框架。

### 微软：Visual Studio Codespace

[Visual Studio Codespace](#)（以下简称 **Codespace**），是微软 VS Code 团队 2019 年推出的云开发环境产品，该产品的特点是：

- 支持三种访问客户端：VS Code，Visual Studio IDE，以及 Web。
- 提供收费的云托管（Azure）环境与免费的自维护环境两种服务方式（仍需要注册 Azure 账号来访问）。
- 内置多人协作工具 Live Share 和 AI 智能代码提示功能 IntelliCode。
- 自定义环境基础配置，可[定制化](#)开发环境基础设施，例如 Linux 版本、工具、端口、变量、IDE 插件等。
- 自定义[个性化配置](#)，定制环境中各类配置文件，例如 .bashrc，.editorconfig 等。

### Eclipse: Theia

[Eclipse Theia](#)（以下简称 **Theia**），是 Eclipse 基金会推出的 VS Code 的替代产品，它的定位是以 NodeJS 和 TS 为技术栈开发的云端和桌面端的 IDE 基础框架，于 2017 年启动，2018 年发布了对应的 Web 端 IDE 产品 [Gitpod](#)。

Theia 和 VS Code 的技术相同点有：

- 编辑器核心都基于 [Monaco Editor](#)。
- 都支持 Language Server Protocol（LSP）。
- 都支持 Debug Adapter Protocol（DAP）。
- 都支持 VS Code 的插件体系。

官方总结，与 VS Code 相比，Theia 的不同之处在于：

- 架构上更模块化，更易于自定义。
- 从一开始就被设计成同时运行于桌面和云端。
- 由厂商中立的开源基金会开发维护。
- 开发独立的 WebIDE 是云开发产品的**首选**，目前 VS Code 并未开源功能完整的 Web 版本（目前开源的 Web 版本仅可用于预览功能），但 Theia 有开源可定制化的版本。

## 云开发模式的技术要素

一般来说，云开发模式依赖的技术要素主要有三个方面：**WebIDE**，**容器化**，以及能够**对接其他云服务**。

### WebIDE

继 VS Code 2019 年发布 Codespace 后，Eclipse 基金会于 2020 年初也发布了 Theia 1.0 版本。WebIDE 在功能体验上已达到和桌面 IDE 相同的水平（尽管在初始化阶段会有不同程度的额外耗时）。同时，WebIDE 还具有以下优点：

- 便于**平台化定制**：在团队使用时可通过定制 WebIDE 来实现**通用的功能扩展和升级**，而无须变更团队成员的桌面 IDE（例如，使用微信开发者工具软件的同学，在工具发布新版本时需要各自处理升级，而 Web 版则无须如此）。
- **流程体验上更平滑**：虽然基本使用仍然是打开一个包含源代码的工作空间容器进行开发，但是通过和代码仓库以及 CI/CD 工具的对接，可以在很多流程节点上做到平滑的体验（例如，测试环境下修复 Bug，可以通过工具，在查找到对应的提交版本后点击进入到 IDE 界面进行修复、测试和提交，相比于原先需要线下操作的流程而言，效率会上升一个台阶）。

### 容器化

容器化以往在服务部署中应用较多。在云开发中的用途主要有：

- 为每个用户的每个项目创建**独立的工作空间**。
- 基于容器化的分层结构，可以方便地在基础环境、项目、用户等维度做镜像继承，便于团队成员维护相同项目时**提升环境创建效率**。
- 相比个人虚拟机，有利于**提升资源利用率**，同时环境搭建更便捷。

### 云服务对接

在一些云厂商的云开发产品中，除了容器化工作空间和 WebIDE 之外，也包含了与其他上下游服务的对接。例如在阿里云的 CloudIDE 产品中，就包含了一键部署等功能。而在自研的体系内，也可通过类似的方式将各个环节的工作流程进行串联，从而形成整体工作流程的效率提升。

## 云开发的效率提升应用场景

当我们以团队的方式来实践云开发时，能够找到一些效率优化的切入点，下面仅列举一些代表性的应用场景。

### 项目篇

1. **加速创建新项目**：在云开发模式下，可以将包含依赖安装的项目模板存储为镜像，开发者选择镜像并创建容器后即可直接预览效果或进入开发，免去下载模板与安装依赖的时间。
2. **项目依赖版本统一**：npm 依赖包在不同环境下安装时，版本自动升级的问题常常对开发测试造成影响（尽管可以通过“npm ci”等命令锁定版本，但在实际业务中普及率并不高，这个问题在部署效率篇中会再次谈到）。而在云开发模式下，可以将 node\_modules 依赖目录（或 Yarn 的 .pnp 目录）与依赖版本做关联，存储为独立镜像，供开发、测试、部署使用。在相应的流程中就可以免去安装依赖，以达到各环境下依赖版本的统一管理，同时也提升了各环境的处理效率。

## 工具篇

1. **开箱即用的开发环境**：在开发环境维度上，通过云开发模式，可以将开发所需的不同基础环境以及各种应用程序制作成开发环境镜像，供开发者自由选择。刚入职的新人无须花费大量时间去学习和安装调试项目的开发环境，真正达到开箱即用的效果。
2. **自定义辅助工具的快速共享和共建**：前端工具的共享不再局限于各自安装 npm 包的方式，通用的配置、公共的依赖、针对特定项目类型的代码片段、emmet 缩写等，一切能想到的辅助工具，都可以在云平台的模式下快速落地，集成到各开发者的工作空间中。同时，对于工具的作者来说，在云平台的方式下也更容易获取到安装使用量等数据反馈，让优秀的工具得以呈现和传播。

## 流程篇

1. **连接代码仓库与开发环境**：云开发的模式可以缩短代码仓库与开发环境的距离，通过项目与开发环境的配置关联，可以从代码仓库的任意 commit 直连创建云端工作空间或进入已有工作空间。
2. **连接 Pipeline 与开发环境**：在构建部署的过程中，遇到构建问题或其他测试流程问题的情况时，可以通过对应的提交信息，直连创建临时修复用途的项目工作空间，以便快速调试和修复部署。

## 使用云开发的注意点

尽管随着 WebIDE 的兴起，越来越多的云开发产品开始呈现，但是作为一种新兴的工作模式，在尝试规模化使用前还是需要考虑到可能出现的一些问题：

1. **代码安全问题**：代码安全是首先需要考虑的问题。通常在代码仓库中我们会设置具体项目的访问权限，云开发模式下的镜像与空间访问设计上也应当注意对这部分权限的验证。此外，对于公司内部的项目，在使用云开发模式时应当首选**支持内部部署**的云服务或**搭建自维护**的云服务，而非将代码上传到外部云空间中。
2. **服务搭建与维护**：要在团队内使用云开发的功能，需要考虑服务搭建的方式和成本。对于大厂而言，云服务资源和技术建设比较丰富，搭建自维护的云开发服务可以提供更多灵活的功能；而对于中小规模的技术团队而言，购买使用一些支持内部部署的现有云开发服务是更好的选择。
3. **服务降级与备份**：由于云开发模式下将开发环境与工作代码都存储于云端，需要考虑当云端服务异常时的降级策略。例如是否有独立的环境镜像可供下载后离线使用，以及工作空间内的暂存代码是否有备份，可供独立下载使用。

## 总结

这一课时我们先介绍了云开发的概念，以及相比于现在的开发方式，它能解决哪些方面的问题。然后一起了解了几款有代表性的云产品，其中需要重点关注的是 VS Code 系的 Codespace 产品。此外，如果你对定制 WebIDE 感兴趣，从 Theia 入手会是比较好的选择。

在介绍完产品后，我们又讨论了云开发这种模式的一般技术要素，以及使用它所能带来的几个比较明确的效率提升场景。最后还有几个新技术对应的风险点，在真正尝试选择云开发方案前需要被考虑到。

今天的**课后思考题**是：希望你能实际体验课中讲到的一些产品，可以在课后讨论中分享你使用后的感受。