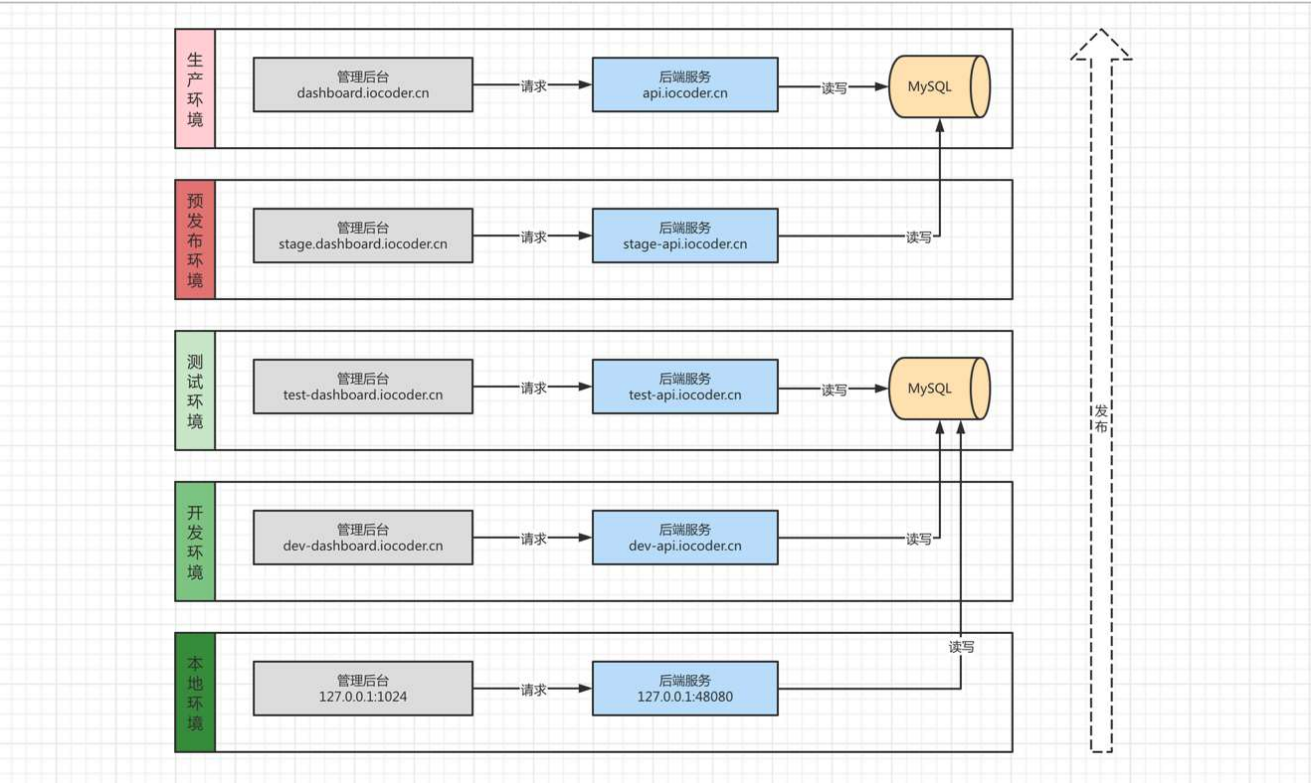


🔗 开发环境

在系统开发的经典模型，一般会分成 2 类 5 种环境：

- 【线下】本地环境(local)、开发环境(dev)、测试环境(test)
- 【线上】预发布环境(stage)、生产环境(prod)



- 每个环境、每个项目使用独立的二级域名
- 线下、线上各一套 MySQL 数据库，多个环境共享使用
- 每个环境对应一个配置文件，后端使用 `application-{env}.yaml` 文件，前端使用 `.env.{env}` 文件

友情提示：项目中暂时没有 test、stage、production 等环境的配置，需要自己创建。

另外，本文的 MySQL 数据库是基础设施的“泛指”，包括 Redis 缓存、MQ 消息队列，都需要线上线下独立。

1. 本地环境

后端工程师使用 `application-local.yaml` 配置文件，在本地电脑启动后端服务，连接线下 MySQL 数据库。考虑到不影响 dev、test 环境，会配置禁用定时任务、MQ 集群消费的执

行。

前端工程师也会在本地电脑启动前端服务，一般不使用 `.env.local` 配置文件，而是使用 `.env.dev` 配置文件，访问 dev 环境的后端服务。如果需要和后端进行本地联调，可以使用 `.env.local` 配置文件。

2. 开发环境

dev 环境的用户是前端工程师、后端工程师，主要用于前后端的联调、又或者功能开发完后的自测。

一些公司可能不提供 dev 环境，直接使用 test 环境，适合团队规模较小的团队，可以降低服务器的成本。

不过，测试工程师可能比较反感 dev 和 test 环境不隔离，因为他们是按照测试用例，一轮一轮的进行验收。这个时候，如果前端或者后端工程师部署了 test 环境，“破坏”了他当前轮次的验收。

疑问：开发环境可以使用独立的 MySQL 数据库吗？

当然是可以的，提供更好的环境隔离性，避免开发阶段产生过多的脏数据，影响 test 环境的验收。

不过呢，这也带来额外的成本，部署程序到 test 环境时，需要做一次数据库的同步。

3. 测试环境

test 环境的用户是产品经理、测试工程师，主要用于他们的功能验收。

考虑到 test 环境的稳定性，一般建议由测试工程师使用 Jenkins 等工具，完成该环境的部署。具体的原因，上面 dev 环境已经解释了。

疑问：如果需要并行验收多个功能，怎么办？

并行验收多个功能时候，对应不同的 Git 分支，需要搭建多套测试环境。

4. 预发布环境

stage 环境的用户是产品经理、测试工程师，连接线上 MySQL 数据库，基于真实的数据，进行功能的全回归测试。

因为数据更加真实，且更具多样性，所以往往也会测试出较多的 Bug。比较好的解决方案，是将线上数据库定期脱敏，导入线下数据库。

考虑到 stage 环境的安全性，一般由技术经理、运维工程师进行部署。

一些公司可能不提供 stage 环境，直接上线到 production 环境，风险非常高，容易产生较多报错。

5. 生产环境

production 环境的用户是真实用户，即线上环境。一般发布上线时，会进行核心功能的快速测试，避免主流程存在问题。

考虑到 production 环境的问题排查效率，会给技术核心开放 MySQL 数据库的读权限。

[← 地区 & IP 库](#)

[Linux 部署→](#)



Theme by [Vdoing](#) | Copyright © 2019-2024 芋道源码 | MIT License