

## 一、基本信息

电子邮箱: 928981943@qq.com

时间	院校	学历	专业
2012.9~2015.6	中国矿业大学	硕士	计算机应用技术
2008.9~2012.6	南通大学	本科	计算机科学与技术

时间	公司	岗位	工作内容
2019.3~	贝壳	运维开发	DevOps工具平台研发
2018.1~2019.3	平安金服（银科中心）	运维开发	DevOps工具平台研发
2017.6~2018.1	沪江教育	运维开发	DevOps工具平台研发
2015.7~2017.6	安硕信息	软件工程师	1. 后端应用开发 2. 兼职项目运维 3. 平台运维管理 4. 运维架构调研 5. 性能测试与优化

1. 熟悉Linux操作系统，熟悉DNS、NFS等基础服务搭建配置
2. 熟悉Docker、Kubernetes，了解Istio
3. 熟悉ELK日志分析技术栈，熟悉Logback、winston等日志框架
4. 熟悉MySQL、Oracle与Redis、Mongodb等数据库应用
5. 熟悉Git、Gitlab、Maven、Nexus3与Jenkins等
6. 熟悉Go、Python、Java、Ruby、Rails
7. 熟悉Django、Django REST framework等框架
8. 熟悉vue.js、element-ui
9. 熟悉Jmeter，了解LoadRunner性能测试工具
10. 熟悉nginx、jetty、Nagios等服务器工具
11. 熟悉阿里云服务，了解CloudStack、Zstack等
12. 熟悉SaltStack、Ansible等配管工具

## 五、项目经验

### 生产机房迁移（2020.4~2020.8）

- 设计并验证生产机房迁移方案：业务应用迁移、DB迁移协作、中间件迁移、流量切分方案、数据同步方案、快速回滚方案
- 制定迁移操作标准流程，组织培训和跟踪，帮助SRE处理迁移过程问题
- 开发迁移工具（嵌入发布平台），支持迁移过程中的双集群部署，简化迁移流程，减少手工操作，避免误操作。

### 应用发布平台（2019.3至今）

#### （1）持续构建管理：

- 打通Jenkins交互，实现Jenkins Pipeline的标准化与定制化设计
- 打通Sonar、MavenAnalysis（内部平台），实现代码质量分析与依赖分析

#### （2）持续部署管理

- 部署配置版本化管理：通过DeployTemplate模型管理发布相关的配置信息，标准化部署配置
- 部署配置个性化支持：通过全局、应用组、应用三级管理，对skywalking、eureka等启动配置个性化支持，管理配置生效作用域。
- 资源规格与容量管理：采用资源套餐，合理分配资源，区分“长久配置”与“活动配置”，支持活动的临时资源扩充以及事后回收。
- 多种发布策略支持：滚动发布、蓝绿发布、灰度发布，灰度发布按照灰度规则区分发布配置，如基于源IP区分流量、基于版本权重区分流量。
- 多种部署模式：OncePod、KubeIngress和Istio。
- 平滑升级：为了解决eureka客户端数据时效问题，采用lifestyle与watcher方案，对eureka的客户端数据实时刷新，避免因发布、重启等情况导致数据不一致问题。
- 内置标准规范：基于公司内部规范，在启动参数、启动任务调度上内置架构标准、测试任务，如支持应用启动后根据环境触发自动化测试任务。

#### （3）K8S管理平台

- 远程终端：Stark WebShell支持常规的运维命令，支持开发线上排障，webshell命令按照环境、用户角色划分权限。
- 虚拟资源池：基于k8s污点容忍机制，将资源隔离分配，支持重要应用的独占式使用资源，目前支持BU、AppGroup两类划分。
- 域名管理：与域名管理平台对接，自动创建、更新Ingress、VirtualService。

### CMDB开发（2018.10~2019.3）

（1）功能：与Agent交互，收集汇总资产数据，建立各类资源关联关系；对外提供查询、聚合接口，支持ODB关联查询；向第三方系统推送更新数据（支持推拉两种模式）。

（2）技术栈：Go语言开发，基于go-restful框架，ZooKeeper作为服务发现组件，Redis作为缓存组件，MongoDB作数据持久化，RocketMQ作为消息中间件。构建代码提交审查规范

## DBTools工具开发（2018.4~2018.9）

（1）功能：慢查询管理；SQL语法规则校验；Web端SQL操作和脚本执行；DB操作权限授权和管理；对接发布系统执行DB脚本；定时任务与消息推送。

（2）技术栈：

- 前端：vue.js + element.js
- 后端：Python语言，Django+Django RESTful框架，Django celery、Django crontab
- SQL校验器：Go语言，Go net/http包，tidb-sql-parser包。

## 工单系统

（1）功能：自定义工单模板和申请审批流，工单申请审批交互流程，提供节点查询、催促等功能。

（2）技术栈：前端采用Vue.js+ElementUI组件，后端采用Django+Django Restful框架，后端采用mongodb做数据持久化。

## 基于GitLab源码修改，实现CICD联动

（1）功能：丰富GitLab Project创建页面，增添若干必填项；GitLab作为数据源，同步创建Jenkins、Sonar等后续服务Job，实现CICD的服务联动。

（2）技术栈：Ruby开发语言，使用Ruby on Rail框架。

## 代码质量分析配置管理

（1）基于Git hook机制，利用Ruby实现代码格式检查、提交信息检查。利用Google Checkstyle对Java代码检查，利用Eslint对JS代码检查，制定了代码提交Commit Message规范，区分feat、fix、refactor等场景，规范了代码提交规范。

（2）实现了hook 脚本自动化推送更新机制

（3）基于Git fork模式与Gitlab权限管理，建立代码提交、审查流程。

## Java应用的压力测试与性能调优

（1）利用Jmeter工具对Java应用API进行高并发压力测试，生成Dashboard Report分析。

（2）对Java代码进行重构，采用三层缓存方式（即本地内存—Redis缓存—MySQL存储），提高数据读取效率。

（3）对MySQL数据表进行优化，通过适当增加索引，提高检索效率。

## 基于ELK框架与阿里云NAS服务搭建分布式日志分析服务

（1）利用logback框架收集java日志，利用winston框架收集node日志

（2）利用阿里云NAS服务实现分布式日志的统一收集与管理

（3）利用Logstash file Input插件收集日志文件，利用Logstash json、mutate、geoip等Filter插件实现日志记录的清洗，利用Logstash Elasticsearch Output插件将格式化日志转存到Elasticsearch。

（4）利用Elasticsearch实现日志的检索、分析，借助Kibana工具实现图形化展示。

## 六、团队建设

### 优化SRE团队工作方式，提高团队效率

- 推动OnCall与业务培训机制落地，改善团队服务质量，缩短服务响应时间，业务团队反馈问题能够得到及时处理、持续跟踪，改善团队间交互，避免推诿踢皮球，问题由Oncaller持续跟踪反馈；同时，基于内部的业务培训，Oncaller可以处理60%的业务问题，非Oncaller人员可以更专注于自身的领域事务，并有充足的时间处理剩余的40%问题，并深度剖析和事后总结，避免相似问题重复出现。
- 事后总结与技术手册的编写培训，让文档活起来。通过逼文档、学文档、奖励文档等手段，让团队成员接受、乐于编写文档，让知识可分享、问题可讨论，让文档真正成为技术参考，让新人更快融入，让事故成为故事（经验）。
- 通过Oncall、Backup、内部培训等机制，解决领域知识的单点问题，如ES只有一个人懂，离了他谁也搞不定，既可以让领域专家可以放心休假，也避免了人员流失引发的被动局面。这些手段的目的是“知识共享、责任分担”，并非是要替代某一人，如果目的不纯，很难推动实施。

### SRE成员培养，提升个人能力，赋能团队

- 对初级工程师帮带，有目的的分配任务，逐步提升个人能力，将一个几近于PIP的小白，培养成能够独立承担80%的SRE业务的中级工程师，实现员工与公司的双赢。
- 高级工程师结对钻研，Istio小组共同调研和探索，实现了基于Istio的灰度发布，既提高了团队成员的技术素养，又很好的满足了业务团队需求。