

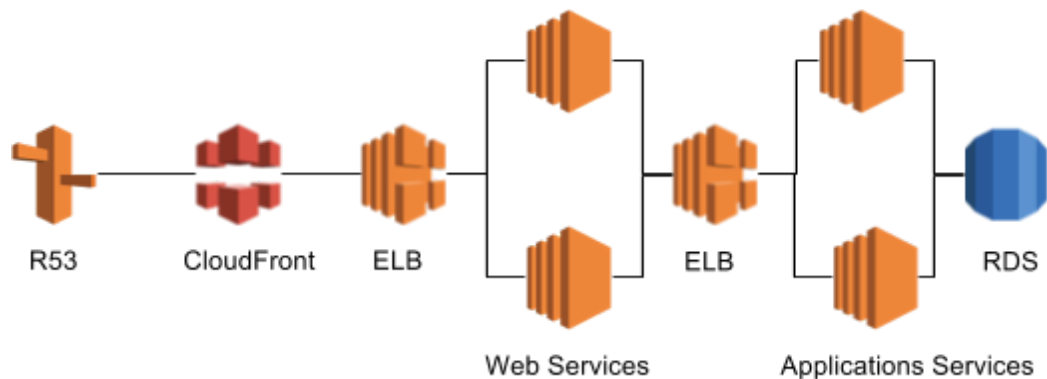
用例名称:

Common Web Application (Based on AWS)

系统描述:

1. 通过AWS R53进行用户域名请求的服务
2. CloudFront作为CDN在理用户最近的位置来分发静态、流式数据以及动态的数据
3. HTTP请求首先会被AWS ELB进行处理, 自动负载均衡至后端的Web Services
4. Web Services会将请求转发至Application Services进行处理, AWS ELB会将请求进行负载均衡
5. AWS RDS作为后端数据库被Application Services调用

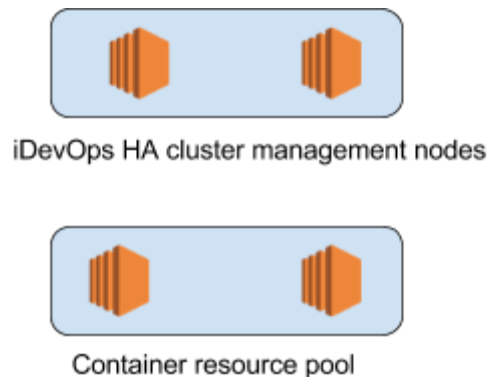
架构图 (Before):



架构图 (After):



部署图:



平台接入步骤之开发测试环境搭建：

Prerequisite:

1. DevOps平台部署完成，容器化资源池分配完毕
2. 管理员登录平台，进入“应用管理”，点击“添加新应用”，新建应用“App1”
3. 设置应用基本信息，为应用进行团队设置并明确成员角色
4. “开发测试环境”中为应用进行线下的资源配置，分配可以使用的计算节点
5. “发布环境”中为应用进行线上发布的资源配置，分配可以使用的计算节点

Steps:

1. 应用开发成员登录，在“我的应用”页面点击进入新建的应用
2. 环境类别选择“开发测试”，点击“新建环境”，在弹出框内输入新建环境的名称“App1_env_tom”，点击确定
3. 切换环境至“开发测试环境”，并下拉框选择“App1_env_tom”
4. 切换Tab至“部署”页，点击“添加组件”按钮
5. **（考虑到依赖设置，添加后端再前端）**
6. （创建后端服务）输入组件名称“web-backend”，输入镜像地址，指定初始版本号“1.0”；设定内存“最小保证”：4x128，“最大限制”：6x128；高可用设置副本数：2；设置环境变量（如果需要）；在“服务扩展绑定”中绑定测试所用的AWS RDS的endpoint（endpoint信息会通过环境变量设置入生成的容器实例中）
7. 服务配置中设定web-backend的开放端口信息，如“容器端口：48082”，“主机端口：8080”，“协议：TCP”，IP绑定中选择多于一个已分配的nodes的IP保证HA即可，如“172.17.10.22, 172.17.10.23”。
8. 完成web-backend组件设置，点击保存退出
9. （创建前端服务）输入组件名称“web-front”，输入镜像地址，指定初始版本号“1.0”；设定内存“最小保证”：2x128，“最大限制”：4x128；高可用设置副本数：2；设置环境变量（如果需要）；在“依赖设置”中添加组件“web-backend”
10. 服务配置中设定web-front的开放端口信息，如“容器端口：48082”，“主机端口：6080”，“协议：TCP”，IP绑定中选择多于一个已分配的nodes的IP保证HA即可，如“172.17.10.22, 172.17.10.23”。
11. 完成web-front组件设置，点击保存退出
12. 在应用“App1_env_tom”的部署页，点击“部署”按钮，返回“状态”页面，等待当前状态变为“创建完成”

Post actions:

1. QA进行开发测试环境的验证

平台接入步骤之线上发布环境：

Prerequisite：

1. 应用资源（线上线外）均已分配完成
2. 应用的团队成员以及角色均已分配完毕，包括“团队成员”与“负责人”
3. 已发布的开发测试环境通过QA验证
4. **在相应环境的构建日志页获取可以进行正式发布的BuildNum**

Steps：

1. 在“我的应用”页面点击进入需要进行发布的应用
2. 环境类别选择“正式发布”，并进入“部署”页
3. **点击“部署”，给新的“正式发布”环境命名：“App1_env_release1”，并输入Prerequisite中获取的BuildNum，点击确认**

4. 系统提示由于架构变更需要先进行同步并修改响应的组件设置，点击“确认”会进行架构同步。之后会看到部署页内所以组件均已同步，但此时尚未进行真正的产品发布。注意此时线上组件的配置信息尚未写入DB，只有在真正触发“部署”后才会进行写入。
5. 进入“部署”->“web-backend”的组件设置页面，此时配置中“版本号”，“镜像地址”以及脚本设置均为只读无法变更
6. 设定内存“最小保证”：4x128，“最大限制”：6x128；高可用设置副本数：2；设置环境变量（如果需要）；在“服务扩展绑定”中绑定线上发布用的AWS RDS的endpoint（endpoint信息会通过环境变量设置入生成的容器实例中）
7. 服务配置中设定web-backend的开放端口信息，如“容器端口：48082”，“主机端口：8080”，“协议：TCP”，IP绑定中选择多于一个已分配的nodes的IP保证HA即可，如“10.17.10.22, 10.17.10.23”。
8. 完成web-backend组件设置，点击保存退出
9. 进入“部署”->“web-backend”的组件设置页面，此时配置中“版本号”，“镜像地址”以及脚本设置均为只读无法变更
10. 设定内存“最小保证”：2x128，“最大限制”：4x128；高可用设置副本数：2；设置环境变量（如果需要）；在“依赖设置”中添加组件“web-backend”
11. 服务配置中设定web-front的开放端口信息，如“容器端口：48082”，“主机端口：6080”，“协议：TCP”，IP绑定中选择多于一个已分配的nodes的IP保证HA即可，如“10.17.10.22, 10.17.10.23”。
12. 完成web-front组件设置，点击保存退出
13. 点击“部署”，选择“正式发布”环境：“App1_env_release1”，并输入Prerequisite中获取的BuildNum，点击确认
14. 应用的“负责人”会在“我的审批”中看到本次发布的信息，点击“批准”则触发发布
15. 切换“状态”页面，等待当前状态显示为“创建完毕”

Post actions:

1. 设置ELB指向“10.17.10.22, 10.17.10.23”，端口“6080”
2. QA对线上发布环境进行验证

接入后持续发布步骤：

Prerequisite：

1. 环境类别“正式发布”中的环境“App1_env_release1”的运行状态为“创建完成”或“更新完成”
2. 组件“web-backend”和“web-front”需要进行版本变更
3. 测试人员在“开发测试”中的环境“App1_env_tom”中进行变更组件以及测试

Steps:

1. 选择“App1”并进入“App1_env_tom”环境
2. 选择“部署”页，选择“web_front”组件进行“设置”
3. 在镜像地址中输入新的镜像版本，如“v2”，设置版本号为：1.1。保存退出。
4. 选择“部署”页，选择“web_backend”组件进行“设置”
5. 在镜像地址中输入新的镜像版本，如“v2”，设置版本号为：1.1。保存退出。
6. 点击“部署”，等待状态变更为“更新完成”后进行测试。并获取BuildNum。
7. 选择环境类别“正式发布”，选择环境“App1_env_release1”
8. 进入“部署”页，点击部署按钮，输入获取的BuildNum，点击确定。
9. 应用的“负责人”会在“我的审批”中看到本次发布的信息，点击“批准”则触发发布
10. 切换“状态”页面，等待当前状态显示为“创建完毕”

Post actions:

1. QA对线上发布环境进行验证。

Complex Web Application

Cross regions service

Big data interaction

Simple online game