

目录

1.ABS 定位.....	3
2.ABS 是互联网运维体系的第一站:	3
3.ABS 的核心价值:	3
4.ABS 的实施时间表:	3
5.ABS 的系统角色职责:	4
6.系统介绍.....	5
6.1 编译种类及其说明-按编译目标.....	5
①Daily Building:	6
②Private Buiding:	6
③Module Building	6
④System Building	6
⑤Test Building.....	6
⑥Module Release	6
⑦System Release.....	6
⑧Smoke Release.....	7
⑧Product Release	7
6.2 编译种类及其说明-按执行时间	7
①Check-In Building:	7
②Schedule Buiding:	7
③Manual Building	7

- 文档属性

属性	内容
项目名称:	Taobao ABS (AutoBuildSystem) 开源
文档主标题:	ABS 总体介绍
文档副标题:	
文档编号:	
文档版本号:	1.0
版本日期:	2010 年 9 月 20 日
文档状态:	Active
文档提供:	Taobao
文档作者:	ShiLong<shilong@taobao.com>

- 文档变更过程

版本	修正日期	修正人	描述
1.0	2010 年 10 月 20 日	士龙	新文件

1.ABS 定位

专注编译，支持日编译和发布管理

2.ABS 是互联网运维体系的第一站：

A:保证开发/测试/生产环境一致性。解决一个 build 和打成 rpm 并上传到 yum 服务器的问题。在 test 包就上传到 yum 的 test 目录，测试可以生产，就到 current 目录，基础软件会有 stable 。

B: 以 ABS 为轴心，驱动代码 review, 安全 review, DBA review, AutoTest 等动作和 build 的无缝集成。

C: 和自动部署集成，自动部署可能是其他团队来做，我们去推。

包管理服务器 (yum server) 大面积推广后，继续 config server 和自动部署系统。

1) 先期包管理起来以后，可以直接批量的 yum install。包依赖通过 rpm 包解决。

2) 对应用进行批量配置安装部署。以产品的某个模块（同样功能的一组机器）为单位进行自动部署。每个模块为一个自动部署条目，模块后面对应的机器是同一个 nodegroup，服务器到 nodegroup 的对应关系都在目前重新整理的 opsdb 里面的。自动部署主要是考虑非 java 部署部分。

Config server: 一个配置服务，即通过它集中维护配置信息，可以动态从这个服务上获取配置信息。一个功能有 1 组机器，它们上线后，根据配置，自动到 config server 上注册，注册信息有，具体什么功能，现在可以提供服务等等，应用就调用这组功能的服务器。

Deploy server: 对相同部署逻辑的机器定义为一宏，将部署过程程式化后应用到宏上，当宏中增加机器时会自动部署上。

3.ABS 的核心价值：

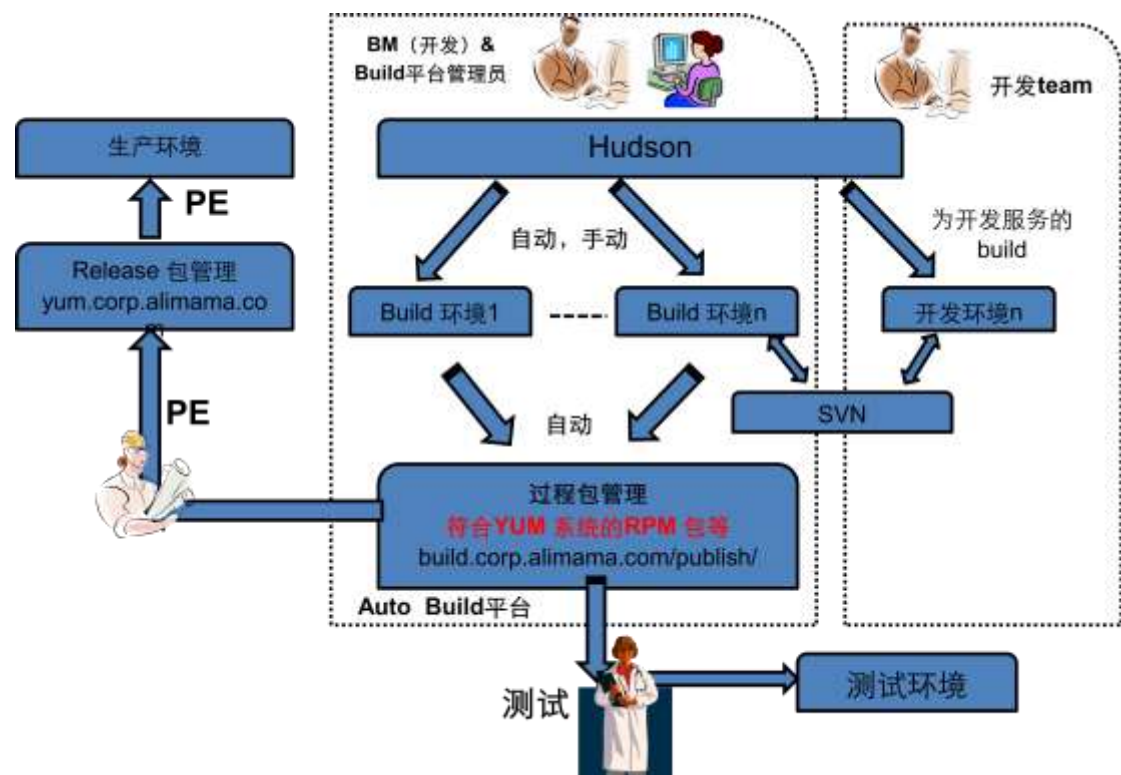
1) 通过 ABS 推广，进行 OPS 基础环境整理，统一基础包，统一开发/测试/生产环境一致性，利用 RPM 解决包依赖，进行大规模集群部署，安装。

2) 以 PE 为核心，通过 PE 来规范 QA, DEV, 同时将开发过程中需要自动化串联的事情通过 ABS 驱动完成，如代码 review, unit test, auto test 等等。

3) 角色职责分工明确，规范执行力很强，ABS 迫使将 dev/qa/pe 分开，迫使代码结构垂直化。

4.ABS 的实施时间表：

5.ABS 的系统角色职责:



角色与职责:

角色	职责
组织级配管	维护 Hudson 编译环境; 设置权限; 对编译环境提供支持; 负责 Hudson 的培训实施推广工作。
开发人员 DEV	运用 Hudson 进行 check-in build。 通过平台将代码提交给 BM。
配管代表	配合组织级配管进行 ABS 推广工作。 整理配置库。 督促团队成员按照配管约定开展工作。
上线负责人 BM/Build Master	运用 Hudson 进行 Test Building, 通过平台自动生成测试包放置到约定地点, 提交给 QA 人员执行测试; 整理上线步骤, 将 QA 验证过的安装指南, 通过平台自动生成发布包放置到约 定地点, 供 PE 分发。
测试人员 QA	到约定地点获取测试包执行测试; 部署维护测试环境;
上线执行人 PE	到约定地点获取发布包执行分发, 根据 BM 提供的经过 QA 验证过的安装指 南进行预发布环境部署以及线上分发。

6.系统介绍

6.1 编译种类及其说明-按编译目标

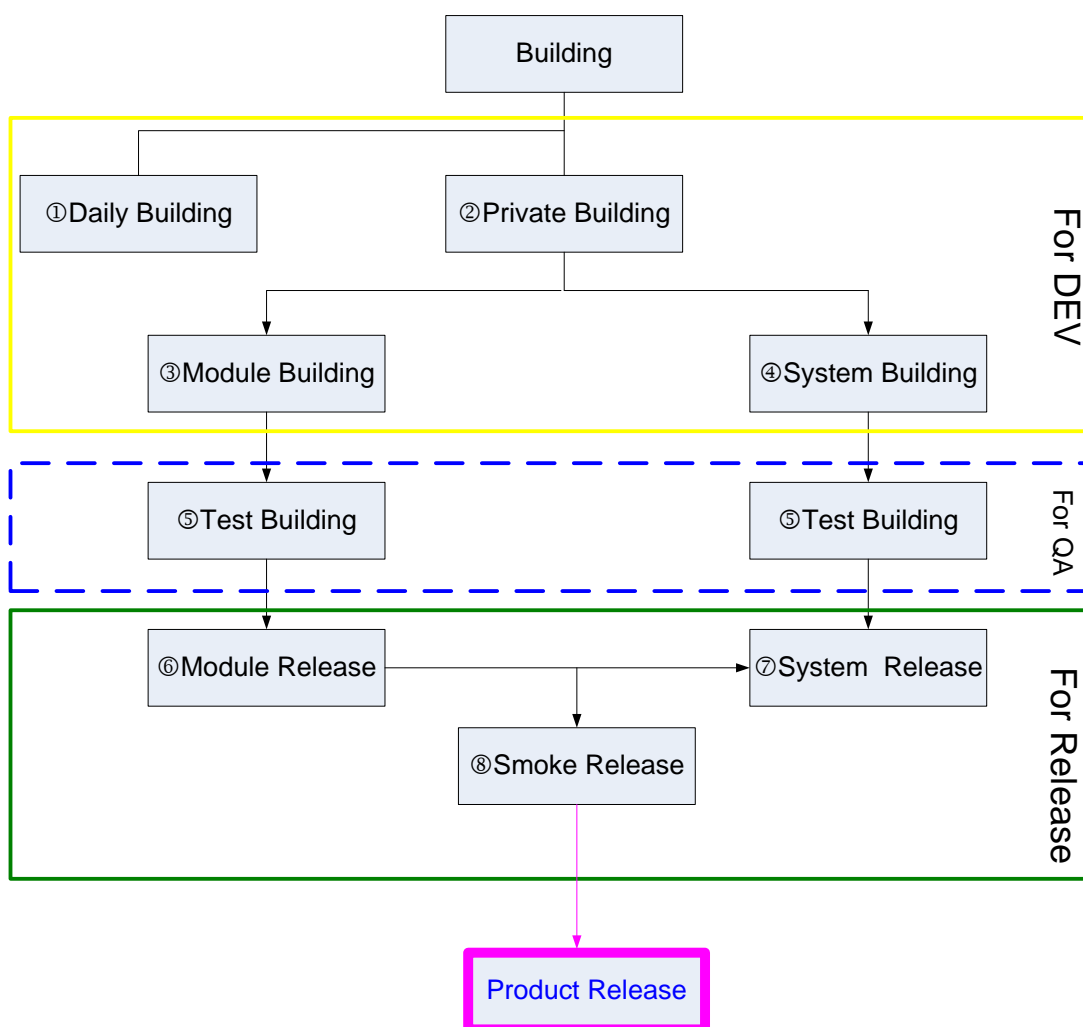
开发组可以分成 3 种情况：

做基础模块的小组如基础平台开发，SDS 开发团队；

中间件的如算法，引擎；

做应用层的小组如前端。

根据不同的分发对象，编译需要分成以下几种类型的编译，如图：



①Daily Building:

为了检查代码的完整性和一致性，同时将项目的公用组件以产出物的方式进行及时共享，同时可做到代码精细权限控制的目的。项目进行每日构建，构建的时间可以根据实际情况确，各研发人员需要在此前将每天的代码提交到 SVN。每日由自动 Build 平台进行编译。

②Private Buiding:

开发进行程序调试进行的编译或者提交测试前进行自测的编译可以通过 Private Building 进行。PrivateBuilding 又可以根据不同的产品发布要求分为：Module Building 和 System Building.

③Module Building

编译的范围是其中一个 Module，从功能的角度上颗粒度是到模块，产出物可能是 dll, lib,.a 等文件，也可以是 exe 文件。Module Building 完成后不需要有安装包。

④System Building

和 Module Building 相对应的一种编译类型，编译需要对整个系统进行全编译，产出物是 exe 或者 war 文件。.

⑤Test Building

开发自测通过需要提交给内容 QA 的编译称作 Test Building. 现场紧急 BUG 修改或者已发布版本临时修改需要回溯到此编译所对应的分支上进行。

⑥Module Release

和 Module Building 相对应的发布，不需要有全量编译。

⑦System Release

和 System Building 相对应，为全量发布。

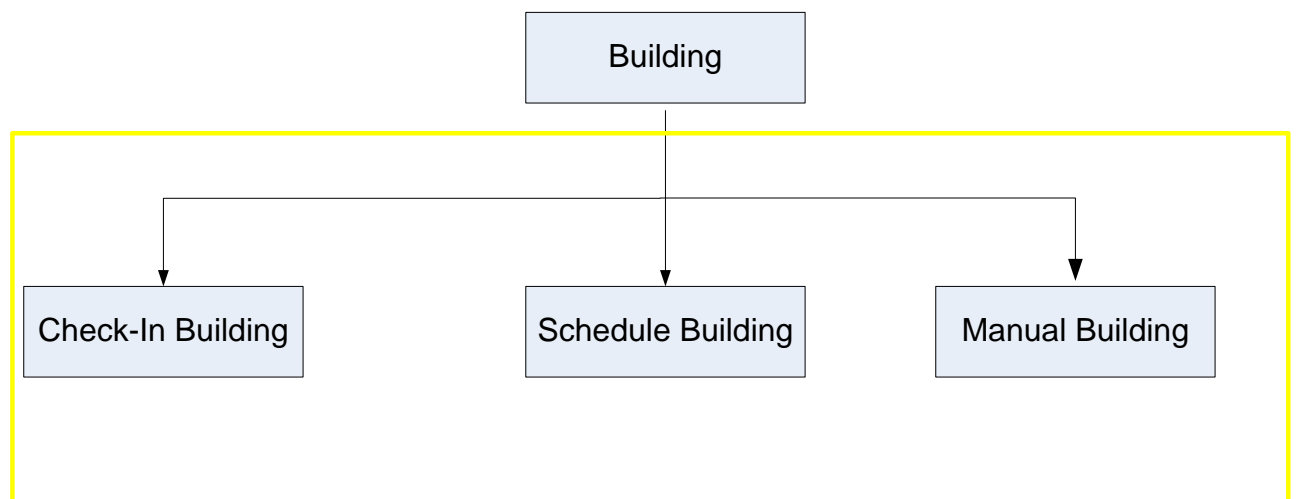
⑧Smoke Release

预发布环境上的发布，叫 Smoke Release。

⑧Product Release

PE 线上正式分发，叫 Product Release。

6.2 编译种类及其说明-按执行时间



①Check-In Building:

在代码提交至配置库时自动进行编译，从而拦截编译类型的 BUG。

②Schedule Buiding:

定时运行的编译，根据实际情况定制。

③Manual Building

人工强制编译，按需执行，通常为开发提交测试时的编译，或者测试通过后提交 PE 的编译。

