

# 构建基于LIGHT的WEB项目

ver 0.3

——ACHILLES实践经验分享

无线中心/网站研发部 Allen Guo

Since 2016/12/10

[guojunlong@tuniu.com](mailto:guojunlong@tuniu.com)

# 创建HELLO WORLD项目

```
$> cnpm set registry http://10.40.189.154:7001  
$> cnpm i -g @tuniu/light-cli  
$> light-cli init --template=clean <Project Name>  
$> cd <Project Name>  
$> npm run local.start  
#(window系统执行 npm run win.start)
```

访问：<http://localhost:5000/>

# 无线**NODE.JS**项目代号

途牛PC站

Achilles

[ə'kɪlɪz]

<希神>阿喀琉斯（希腊勇士）

途牛M站

Ares

[ɛrɪz]

<希神>阿瑞斯，战神

# ACHILLES—启动

svn co <http://boy.tuniu.com/svn/MOB/ACHILLES/branch> achilles

cd achilles

cnpm i

npm run local.start # for win32 : npm run win.start

npm run status # 监控控制台日志

访问：<http://localhost:4006/>

# ACHILLES—最佳实践I

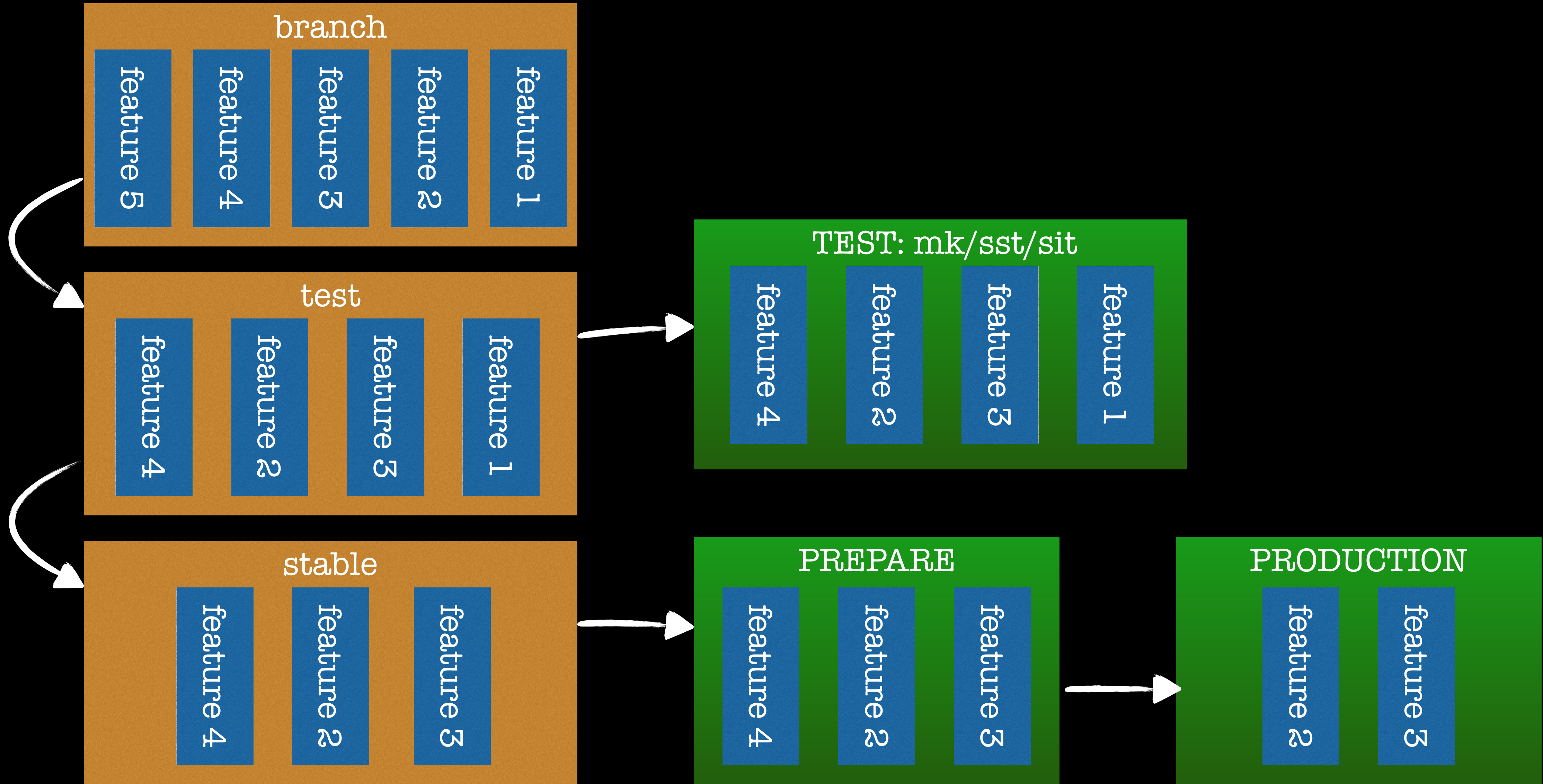
- ① Code Style遵从: <http://wiki.tuniu.org/pages/viewpage.action?pageId=65683301>
- ② 每个文件都应该有注释, 标注文件的作者, 多个作者用逗号分隔
- ③ 总是尽可能使用默认约定而不用配置
- ④ 可重用的常量应考虑配置化
- ⑤ 模块里的导出函数或类应总是保持与模块文件名一致 (区分大小写)
- ⑥ 建议总是为Web Api提供Swagger测试文档.

# ACHILLES—最佳实践II

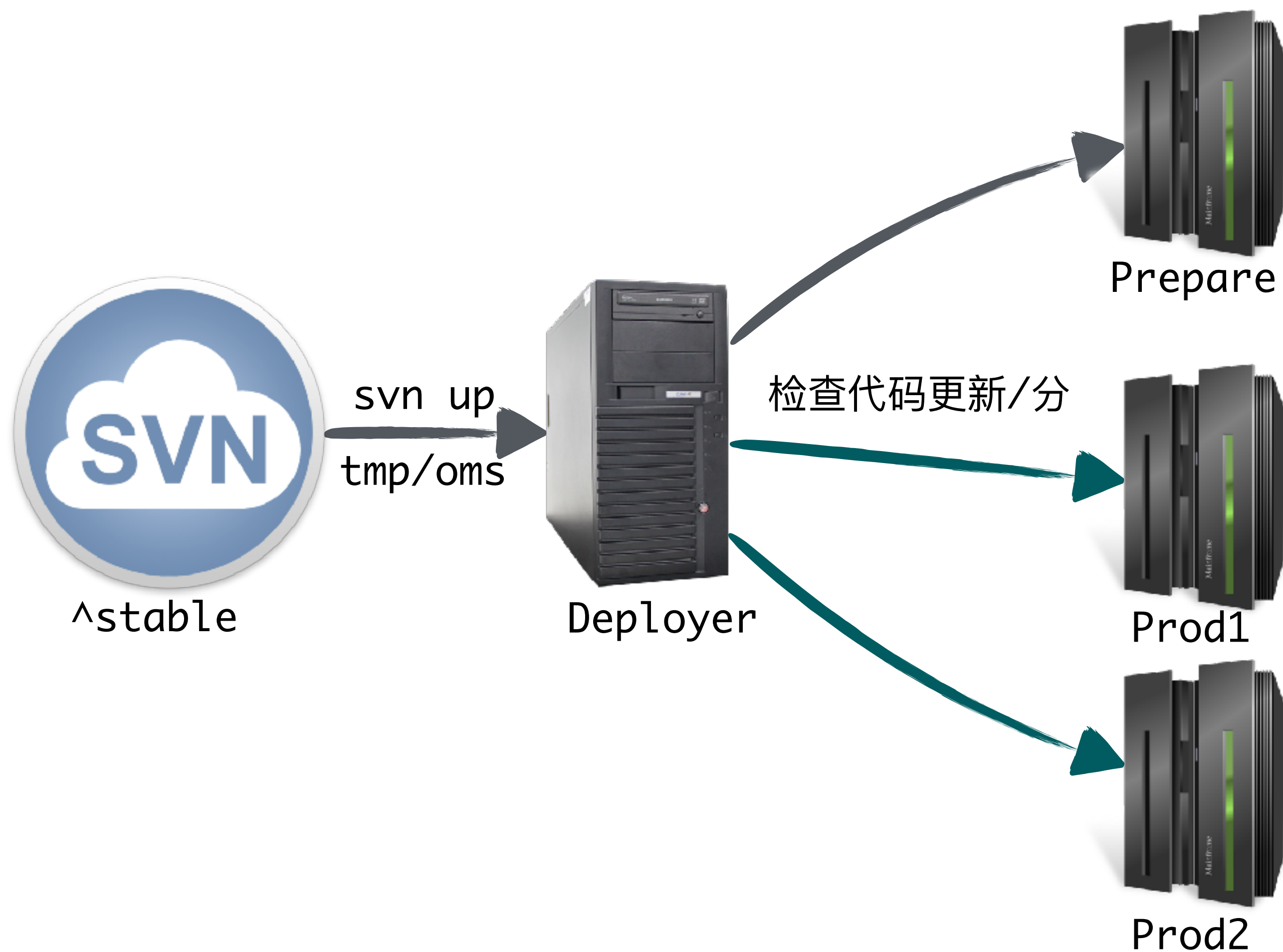
- ① 没有环境依赖的Model调用应使用importModel，从来不要用使用require
- ② Generator风格的Controller总是受欢迎的
- ③ Model方法的返回如果是Promise，方法名应总是以大写P结尾
- ④ Model方法要么抛出异常处理错误，要么使用错误码处理错误，不应两者皆有
- ⑤ 如果异常不需要二次处理，从来不要捕获
- ⑥ 倘若捕获异常返回框架处理，需设置合适的http状态码后调用next(err)
- ⑦ 不需重用的业务逻辑可以直接写在Controller里，不必一定要创建Model方法



# 测试&发布流程



# 发布机制



- ① oms发布代码到Deployer
- ② 目标服务器每一分钟检查一次 Deployer上的项目代码，如果有变更，就同步到自己
- ③ 如果项目根目录package.json有变化，执行cnpm i
- ④ 最后重新启动Node服务

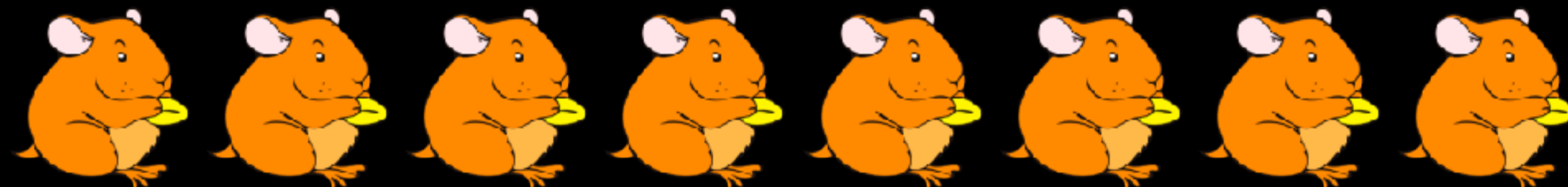


# 发布流程缺陷

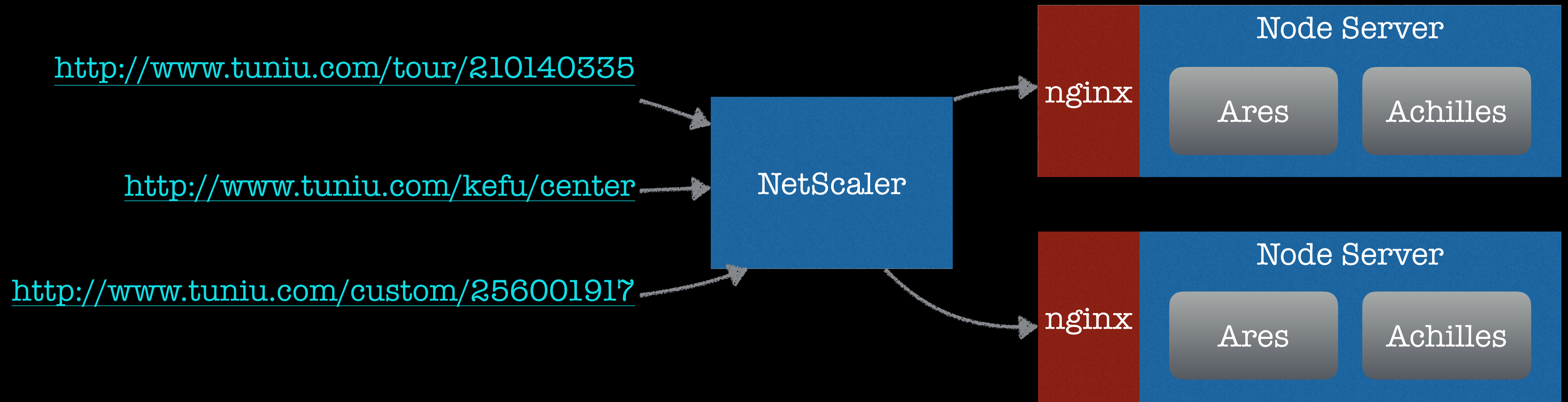
- ① Node.js启动瞬时问题
- ② 未正确使用合并和复兴合并，无法追溯历史
- ③ cherry-pick合并阻碍，然后直接文本diff修改stable上代码，一段时间过去stable与branch偏离越来越远
- ④ stable分支不能精确反应为生产服务器上的代码
- ⑤ 前面3个问题导致：系统重构、升级风险重重

# 更好的发布模型

- ① 代码一致性：正确使用SVN长短分支与合并
- ② 快照：使用SVN TAG发布和回滚
- ③ 灰度发布：一组一组机器发布
- ④ 串行：svn合并、代码发布顺序进行



# 服务器架构



- ❶ Xeon(R) CPU E5-2620 2.4Ghz; 12核(超线程)
- ❷ 64G Memory
- ❸ 每应用4个Node进程

# ACHILLES—基准测试环境

## mk (test分支)

- host配置: 172.31.3.91 [www.mk.tuniu.org](http://www.mk.tuniu.org)
- <http://www.mk.tuniu.org:4006>

## sst (test分支)

- host配置: 10.10.30.147 [www.ord.tuniu-sst.org](http://www.ord.tuniu-sst.org)
- <http://www.ord.tuniu-sst.org:4006>

## sit (test分支)

- <http://www.ord.tuniu-sst.org:4006>

## prepare (stable分支)

- <http://w-p.tuniu.com>

## production (stable分支)

- <http://www.tuniu.com>



# 背后的坑





# 坑1：现象

- ① 一次周五晚上迭代发布后，系统运行12小时，早上起来服务器挂了，错误日志无；
- ② 怀疑发布代码有问题(恰好改了底层代码)，怀疑发布不完全再发布，10个小时后又挂了，错误日志无；
- ③ 怀疑tmp的svn up遗漏，checkout整个项目再发布，5分钟挂了，错误日志无。

# 坑1：CODES BUG

```
'use strict';

let str = '<br/>'          早餐后自由活动，于指定时间集合自行办理退房手续。';
str += '<br/>'              <br/>              <br/>              <br/>';
str += '                  <br/>'              <br/>';
str += '                  <br/>'              <br/>';
str += '                  根据船班时间，自行前往暹粒机场，返
回中国。<br/>';
str += '如需送机服务，需增加280/每单。<br/>';

let r = String(str).replace(/(^(\s*<br[\s\|]*>|(\s*<br[\s\|]*>\s*)+$)/igm, '');
console.log(r);
```

# 坑2：现象

- ❶ 一次周五晚上迭代发布后，有人报告页面头部时有时无。因为过一会现象全无，没有理会。24小时后问题再次反复出现
- ❷ 发现后台有一条奇怪日志：一个写死的变量(常量)产生未定义错误
- ❸ 每次访问同一条跟团游线路详情页，内容都不同。

# 坑2：CODES BUG

## BEFORE

```
module.exports = function(defaultHandler) {  
  let handler = _merge({}, defaultHandler);  
  return {  
    tuniuRender: function(req, res, next) {  
      .....  
    }  
  }  
}
```

## AFTER

```
module.exports = function(defaultHandler) {  
  return {  
    tuniuRender: function(req, res, next) {  
      let handler = _merge({}, defaultHandler);  
      .....  
    }  
  }  
}
```

# 坑3：现象

- ❶ Ares网站访问时快时慢
- ❷ 确定Redis连接不上导致，但Ares代码N月未改，初步甩锅给运维
- ❸ 运维发现大量的Redis连接，重新启动服务器剔除无效连接，但不承认是他们的锅
- ❹ 当晚Redis服务器又收到Node端数千个连接



# 坑3：CODES BUG

## BEFORE

```
let ioredis = null;  
function StorageRedis(redisConf) {  
    ioredis = new IORedis(redisConf);  
}  
module.exports = StorageRedis;
```

## AFTER

```
let ioredis = null;  
function StorageRedis(redisConf) {  
    if (!ioredis) {  
        ioredis = new IORedis(redisConf);  
    }  
}  
module.exports = StorageRedis;
```

Achilles和Ares部署在同台Node服务器

**DEPPER 坑... ..**

# 参考资源

Node.js知识

<http://wiki.tuniu.org/pages/viewpage.action?pageId=65991227>

@tuniu/remote-client

<http://wiki.tuniu.org/pages/viewpage.action?pageId=68871311>

无线Node.js项目

<http://wiki.tuniu.org/pages/viewpage.action?pageId=67466108>

Achilles快速开始

<http://wiki.tuniu.org/pages/viewpage.action?pageId=67449334>

无线前端规范2016版

<http://wiki.tuniu.org/pages/viewpage.action?pageId=65683301>

**THE END**