iOS 雲端服務開發

第五週

課程大綱

- 自動化作業： **grunt**
- 發佈服務至**AWS**
  - **forever**
- 效能測試 & 系統狀態監控
- 與**iOS App**合作
  - **Documentation**
  - **Testing Server (通常是CI Server)**

Cluster

- 一個node.js執行緒只會用到一個cpu core
- 最佳化系統資源運用 ex: 共用port
- zero downtime upgrading
  - http://blog.evantahler.com/blog/production-deployment-with-node-js-clusters.html

Cluster

- cluster.isMaster / isWorker
- cluster.fork()
- Worker
  - id
  - process

Cluster

```
var cluster = require('cluster');
var os = require('os');

var cores = os.cpus().length;

var workers = {};

if (cluster.isMaster) {
    cluster.on('exit', function(worker) {
        delete workers[worker.process.pid];
        worker = cluster.fork();
        workers[worker.process.pid] = worker;
    });

    for (var i = 0; i < cores; i++) {
        var worker = cluster.fork();
        workers[worker.process.pid] = worker;
    }
} else {
    var app = require('./app');
    app.listen(app.get('port'), function() {
        console.log("HTTP Server listening to port: " + app.get('port'));
    });
}

process.on('SIGTERM', function() {
    for (var pid in workers) {
        process.kill(pid);
    }
    process.exit(0);
});
```

config tips

- mailer.example.json

```
{
    "smtpOption": {
        "service": "Gmail",
        "auth": {
            "user": "",
            "pass": ""
        }
    },
    "mailOption": {
        "subject": "MessageBoard Confirmation Mail",
        "template": "confirmation",
        "generateTextFromHTML": true
    }
}
```

- 複製到mailer.json 並修改，mailer.json放到gitignore, 不要commit，repo裡面永遠只有範本

Grunt

- 自動化
  - minification/uglify
  - unit testing
  - linting
  - deploy
  - merge
  - shell command

JSHint

- http://www.jshint.com/
- 檢查js檔是否有潛在錯誤

JSHint Coding Style

- http://www.jshint.com/hack/
- 使用tab縮排
- **if, for, while** 後面接一個空白
- 匿名函式需要一個空白: **function () {};**
- 具名函式不需要: **function foo() {};**

JSHint Coding Style

- 變數/屬性指派之縮排

```
// Good
var next = token.peak();
var prev = token.peak(-1);
var cur  = token.current;

var scope = {
  name:   "(global)",
  parent: parentScope,
  vars:   [],
  uses:   []
};

// Bad
var cur         = token.current;
var isSemicolon = cur.isPunctuator(";");
```

JSHint Coding Style

- 一次宣告一個變數，但沒有指派值的變數可以一起宣告

```
var token = tokens.find(index);
var scope = scopes.current;
var next, prev, cur;
```

- 使用 === 以及 !==
- 使用分號結尾

安裝 Grunt

grunt指令

```
#grunt命令列
~ $ npm install -g grunt-cli
```

grunt套件

```
#先更新系統
~ $ npm install grunt --save-dev
```

package.json

```
"name": "MessageBoard",
  "version": "0.0.1",
  "dependencies": {
    ...
  },
  "devDependencies": {
    "grunt": "~0.4.1",
    "grunt-contrib-uglify": "~0.2.2",
    "grunt-contrib-jshint": "~0.6.0",
    "grunt-contrib-concat": "~0.3.0",
    "grunt-contrib-watch": "~0.4.4",
    "grunt-mocha-test": "~0.5.0",
    "grunt-shell": "~0.3.0"
  }
}
```

- **devDependencies:**
  npm install --production
  時不會安裝（僅在**develop**環境時安裝）

Gruntfile.js

- **wrapper**
- 專案與任務設定
- **grunt**任務插件載入
- 自定任務

Gruntfile.js

```
#Gruntfile.js

module.exports = function(grunt) {
  // Project configuration.
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    uglify: {
      options: {
        banner: '/*! <%= pkg.name %> <%= grunt.template.today("yyyy-mm-
dd") %> */\n'
      },
      build: {
        src: 'src/<%= pkg.name %>.js',
        dest: 'build/<%= pkg.name %>.min.js'
      }
    }
  });

  // Load the plugin that provides the "uglify" task.
  grunt.loadNpmTasks('grunt-contrib-uglify');

  // Default task(s).
  grunt.registerTask('default', ['uglify']);
};
```

task

target

任務設定

```
grunt.initConfig({
  concat: {
    options: { /* task-level options, not a target */ },
    foo: {
      options: { /*target-level options */ },
      // concat task "foo" target options and files go here.
    },
    bar: {
      // concat task "bar" target options and files go here.
    }
  }
});
```

- **grunt concat 會執行concat.foo & concat.bar**
- **grunt concat:foo 僅執行concat.foo**

任務檔案設定

```
grunt.initConfig({
  concat: {
    foo: {
      files: {
        'dest/a.js': ['src/aa.js', 'src/aaa.js'],
        'dest/a1.js': ['src/aa1.js', 'src/aaa1.js'],
      },
    },
    bar: {
      files: [
        {src: ['src/bb.js', 'src/bbb.js'], dest: 'dest/b/', nonull: true},
        {src: ['src/bb1.js', 'src/bbb1.js'], dest: 'dest/b1/', filter: 'isFile'}
      ]
    }
  }
});
```

任務檔案設定

- *: "/" 以外的字元
- ?: 非 "/" 的單一字元
- **: 任何字元，包括 "/"
- {}: 比對裡面的任何一個條件，以逗號分隔。ex: {a,b}
- !: not

任務檔案設定

```
{src: 'foo/this.js', dest: ...}
{src: ['foo/this.js', 'foo/that.js', 'foo/the-other.js'], dest: ...}
{src: 'foo/th*.js', dest: ...}

{src: 'foo/{a,b}*.js', dest: ...}
{src: ['foo/a*.js', 'foo/b*.js'], dest: ...}

{src: ['foo/*.js'], dest: ...}
{src: ['foo/bar.js', 'foo/*.js'], dest: ...}

{src: ['foo/*.js', '!foo/bar.js'], dest: ...}
{src: ['foo/*.js', '!foo/bar.js', 'foo/bar.js'], dest: ...}

{src: ['src/<%= basename %>.js'], dest: 'build/<%= basename %>.min.js'}
{src: ['foo/*.js', '<%= jshint.all.src %>'], dest: ...}
```

動態任務檔案

- **expand: true**
- **cwd:** 設定來源檔根目錄
- **src:** 來源檔**pattern**
- **dest:** 目的檔案/目錄
- **ext:** 目的檔案的附檔名

動態任務檔案

```
grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    uglify: {
        dynamic_mappings: {
            files: [{
                    expand: true, // Enable dynamic expansion.
                    cwd: 'src/', // Src matches are relative to this path.
                    src: ['**/*.js'], // Actual pattern(s) to match.
                    dest: 'build/', // Destination path prefix.
                    ext: '.min.js', // Dest filepaths will have this extension.
                }
            ]
        },
        static_mappings: {
            files: [/*static file array*/]
        }
    }
})
```

動態任務檔案

- 注意：如果要針對**target**設定動態檔案，方式又不一樣

```
grunt.initConfig({
    uglify: {
        build: {
            expand: true, // Enable dynamic expansion.
            cwd: 'src/', // Src matches are relative to this path.
            src: ['**/*.js'], // Actual pattern(s) to match.
            dest: 'build/', // Destination path prefix.
            ext: '.min.js' // Dest filepaths will have this extension.
        }
    }
})
```

變數

- <%= key %>：搜尋config[key]

```
grunt.initConfig({
  concat: {
    sample: {
      options: {
        banner: '/* <%= baz %> */\n',   // '/* abcde */\n'
      },
      src: ['<%= qux %>', 'baz/*.js'],  // [['foo/*.js', 'bar/*.js'],
'baz/*.js']
      dest: 'build/<%= baz %>.js',      // 'build/abcde.js'
    },
  },
  // Arbitrary properties used in task configuration templates.
  foo: 'c',
  bar: 'b<%= foo %>d', // 'bcd'
  baz: 'a<%= bar %>e', // 'abcde'
  qux: ['foo/*.js', 'bar/*.js'],
});
```

載入外部設定

```
grunt.initConfig({
  pkg: grunt.file.readJSON('package.json'),
  uglify: {
    options: {
      banner: '/*! <%= pkg.name %> <%= grunt.template.today("yyyy-mm-dd") %> */\n'
    },
    dist: {
      src: 'src/<%= pkg.name %>.js',
      dest: 'dist/<%= pkg.name %>.min.js'
    }
  }
});
```

常用的任務插件

- grunt-contrib-uglify
- grunt-contrib-concat
- grunt-contrib-jshint
- grunt-contrib-watch
- grunt-contrib-clean
- grunt-shell
- grunt-mocha-test

Grunt for MessageBoard

- 需要的任務： 測試、建置、發佈、監控
- 安裝 **grunt-cli**
- 安裝 **grunt** 及 其他**task plugin:**
  grunt-contrib-uglify, grunt-mocha-test, grunt-contrib-concat, grunt-contrib-jshint, grunt-contrib-watch, grunt-shell
- **Gruntfile.js**

Gruntfile

```
grunt.loadNpmTasks('grunt-contrib-uglify');
grunt.loadNpmTasks('grunt-contrib-jshint');
grunt.loadNpmTasks('grunt-mocha-test');
grunt.loadNpmTasks('grunt-contrib-watch');
grunt.loadNpmTasks('grunt-contrib-clean');
grunt.loadNpmTasks('grunt-shell');

grunt.registerTask('test', ['jshint', 'mochaTest']);
grunt.registerTask('build', ['jshint', 'mochaTest', 'clean:build', 'uglify']);
grunt.registerTask('deploy', ['build', 'shell:uninstall', 'shell:install']);
grunt.registerTask('default', ['test']);
```

Gruntfile

```
mochaTest: {
    test: {
        options: {
            reporter: 'spec'
        },
        src: ['test/**/*.js']
    }
},

jshint: {
    files: ['gruntfile.js', 'src/**/*.js', 'test/**/*.js'],
    options: {
        // options here to override JSHint defaults
        globals: {
            console: true,
            module: true
        }
  }
}
```

Gruntfile

```
clean: {
    build: ["build/"]
},

uglify: {
    options: {
        banner: '/*! <%= pkg.name %> <%= grunt.template.today("dd-mm-
yyyy") %> */\n'
    },
    build: {
        expand: true, // Enable dynamic expansion.
        cwd: 'src/', // Src matches are relative to this path.
        src: ['**/*.js'], // Actual pattern(s) to match.
        dest: 'build/', // Destination path prefix.
        ext: '.js' // Dest filepaths will have this extension.
    }
},
```

Gruntfile

```
    shell: {
        install: {
            options: {
                stdout:true,
                stderr: true,
                failOnError: true
            },
            command: [
                'mkdir -p <%= baseDir %>',
                'cp -rf build config public views package.json <%= baseDir %>',
                'cd <%= baseDir %>',
                'npm install --production'
            ].join('&&')
        },
        uninstall: {
            options: {
                stdout:true,
                stderr: true,
                failOnError: true
            },
            command: 'rm -rf <%= baseDir %>'
        }
    },

    watch: {
        files: ['<%= jshint.files %>'],
        tasks: ['test']
    },
```

AWS

- 登入

```
#ubuntu
~ $ ssh -i <cert.pem> -l ubuntu/ec2user <ip>
```

- 安裝所需軟體

```
#Git
sudo apt-get install git git-core
```

- nvm參考week1講義,
- http://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/

Run our project

- 登入

```
~ $ git clone https://github.com/finfin/CloudAppDev.git
```

```
~ $ cd CloudAppDev/week5/messageBoard
~ $ npm install
~ $ grunt build
~ $ node build/cluster.js
```

- 登出之後就不會執行了怎麼辦？

Forever

- 安裝

```
~ $ sudo npm install forever -g
```

```
~ $ forever my-script.js
~ $ forever start my-script.js

~ $ forever list
~ $ forever stop [name]
~ $ forever stopall
~ $ forever restart [name]
```

UPSTART

```
#!upstart
description "myApp server"
author      "Fin Chen"

#開機執行/關機關閉
start on startup
stop on shutdown

# Restart when job dies
respawn

# Give up restart after 5 respawns in 60 seconds
respawn limit 5 60

#設定執行環境參數
env NODE_ENV=production

script
    logger "start running myApp..." #log到syslog
    chdir /data/apps/myApps #切換至App目錄，主要是避免執行node時的目錄抓取問題
    exec /usr/bin/node /data/apps/myApp/lib/app.js > /data/apps/myApp/log/
app.log 2>&1
end script
```

UPSTART

- /etc/init/#{jobName}.conf
- sudo start/stop #{jobName}

伺服器效能測試

- http://httpd.apache.org/docs/2.2/programs/ab.html

```
#osx, linux server幾乎都有內建
~ $ ab [options] url
```

```
~$ ab -n 10000 -c 10 http://www.example.com/index.aspx
~$ ab -n 10000 -c 10 -k http://www.example.com/index.aspx

# export csv
ab -e output.csv -n 10000 -c 10 http://www.example.com/
index.aspx
```

伺服器效能測試

```
➜ ~  ab -n 5000 -c 5 http://127.0.0.1:3000/
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests


Server Software:
Server Hostname:        127.0.0.1
Server Port:            3000

Document Path:          /
Document Length:        3057 bytes

Concurrency Level:      5
Time taken for tests:   128.581 seconds
Complete requests:      5000
Failed requests:        0
Write errors:           0
```

```
Concurrency Level:      5
Time taken for tests:   128.581 seconds
Complete requests:      5000
Failed requests:        0
Write errors:           0
Total transferred:      16778092 bytes
HTML transferred:       15285000 bytes
Requests per second:    38.89 [#/sec] (mean)
Time per request:       128.581 [ms] (mean)
Time per request:       25.716 [ms] (mean, across all concurrent requests)
Transfer rate:          127.43 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.1      0       2
Processing:    85  128  20.4    123     327
Waiting:       85  128  20.4    123     326
Total:         85  129  20.4    123     327

Percentage of the requests served within a certain time (ms)
  50%    123
  66%    128
  75%    132
  80%    136
  90%    159
  95%    170
  98%    182
  99%    195
 100%    327 (longest request)
```

效能測試

- 排除頻寬的限制
  - 頻寬花錢就可以解決，且與程式本身無關
- 循序漸進
  - 測試系統在各種狀態下的反映

系統狀態監控 nodetime

- https://nodetime.com/
- 註冊

```
~ $ npm install nodetime
```

```
require('nodetime').profile({
    accountKey: '180dd7176a33ba89aaa6149e4f434a01e6a4b044',
    appName: 'Message Board'
});
```

- https://nodetime.com/apps

API Documentation

- **Server/Client遵照文件開發並測試**
- 減少互相等待時間
- **http://weblog.bocoup.com/documenting-your-api/**

API Documentation

- Title
- URL
- Method
- URL Query
- Data (HTTP Body)
- Success Response
- Error Response
- Sample Call
- Note

API Documentation

操作功能	操作URL	Method	Query參數	Data	response type	Success Response	Error Response	說明
書籍清單	/api/materials/	GET			JSON	[{"id": $id, "productID", $productID, "title":$title, "description":$description, "thumbnail": $thumbnail_url, "thumbnail_big": $thumbnail_big_url, "price": $price, "tag": [$tags...], "type": $type, "modifiedTime": $modifiedTime}...]		
書籍清單 by Group	/api/materials/	GET	tag=#{tag}		JSON	[{"id": $id, "productID", $productID, "title":$title, "description":$description, "thumbnail": $thumbnail_url, "thumbnail_big": $thumbnail_big_url, "price": $price, "tag": [$tags...], "type": $type, "modifiedTime": $modifiedTime}...]		
書籍檔案下載	/api/materials/:id	GET	receipt=#{receipt}		attachment file	Content-disposition="attachment; filename=#{material.filename}"	HTTP Status: 404, id not exist	
						Content-type=#{material.type}	HTTP Status: 403, receipt invalid	
							HTTP Status: 400, id do not match	

API Doc - MessageBoard

- POST /apiLogin
- GET /apiLogout
- GET /api/messages
- POST /api/messages
- GET /api/mymessages
- DELETE /api/messages/:id
- GET /api/users/:name

- **Title:** 列出訊息
- **URL:** /api/messages
- **Method:** GET
- **URL Query:**
  name=$user, page=$page, size=$size
- **Data:** n/a
- **Success:**

```
[
  {
    "user": "fin",
    "text": " dddddd",
    "type": "public",
    "_id": "51cdc21a81a7420000000001",
    "date": "2013-06-28T17:04:26.418Z"
  }, ...
]
```

- **Error:** 500

功能測試 curl

```
~ $ curl -v [url]
```

```
~ $ curl -d "username=fin&password=1234" http://localhost:3000/login
```

```
~ $ curl -c cookie.txt -d "username=fin&password=1234" http://
localhost:3000/login
```

```
~ $ curl -b cookie.txt -d "text=howdy&type=public" http://localhost:
3000/messages
```

```
~ $ curl -b cookie.txt -d "_method=DELETE" http://localhost:3000/
messages/:id
```

Continuos Integration

- 主要用於測試用途：自動化與非自動化
- 自動化：**testing**、**code coverage**
- 可當成**testing server**
- 通常設定成每日固定從**repo**抓取最新的程式碼來執行

Continuos Integration

- Jenkins

```
sudo apt-get install jenkins
su - jenkins
git config --global user.name "username"
git config --global user.email "email"
```

- 設定檔在/etc/default/jenkins

```
HTTP_PORT=8080 #改成你要的port
```

- jenkins有自己的user，請確保兩邊的開發版本是相同的

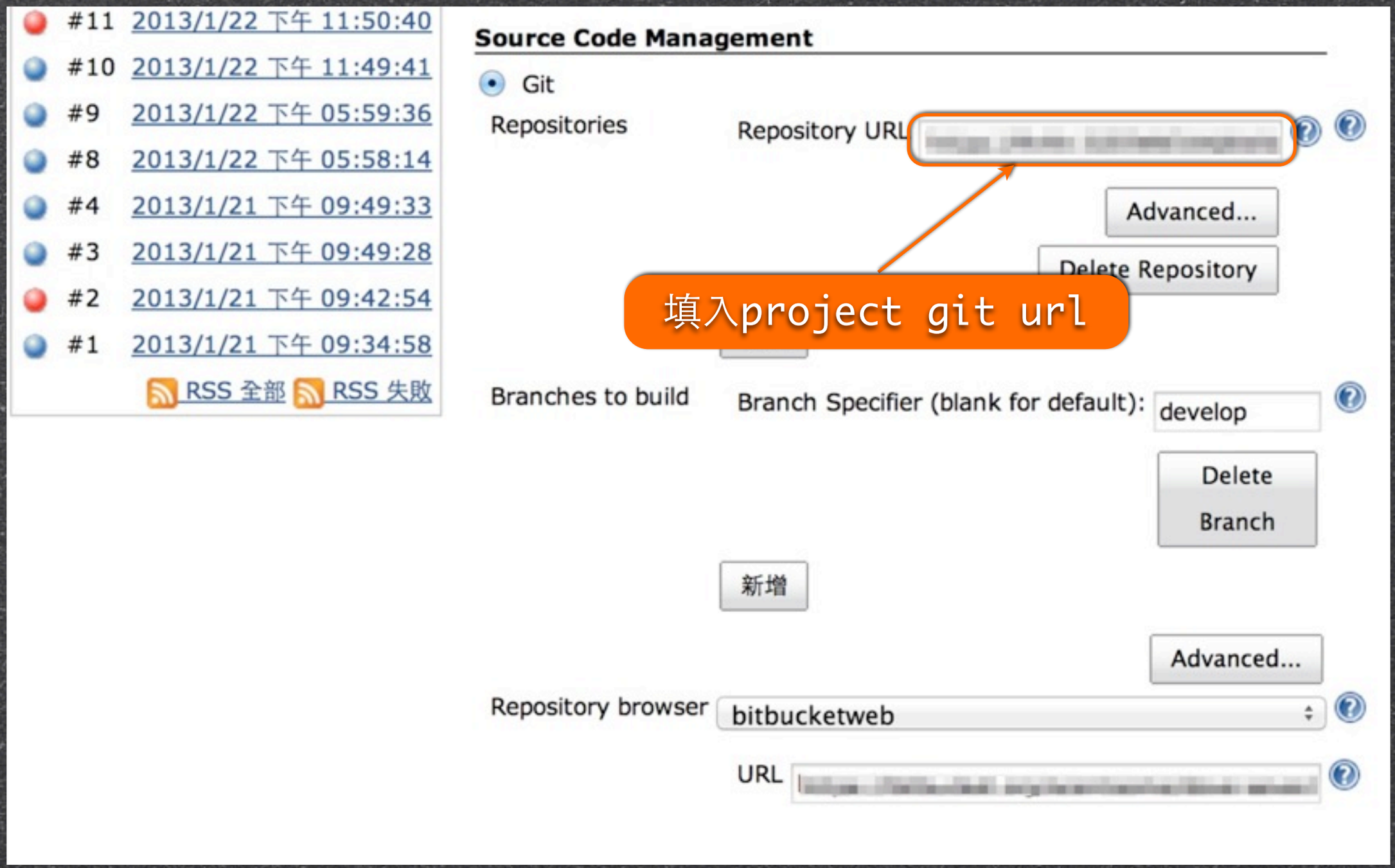
Continuos Integration

- http://server_ip:8080

Continuos Integration

- 管理Jenkins -> 管理插件 -> 有效的
- 搜尋(⌘+f) "Git Plugin" -> 勾選 -> 安裝
- New Job -> 輸入Job名稱 -> Build a free-style software project

CI Job設定


#11 2013/1/22 下午 11:50:40
#10 2013/1/22 下午 11:49:41
#9 2013/1/22 下午 05:59:36
#8 2013/1/22 下午 05:58:14
#4 2013/1/21 下午 09:49:33
#3 2013/1/21 下午 09:49:28
#2 2013/1/21 下午 09:42:54
#1 2013/1/21 下午 09:34:58
RSS 全部
RSS 失敗
Source Code Management
Git
Repositories
Repository URL
Advanced...
Delete Repository
填入project git url
Branches to build
Branch Specifier (blank for default):
develop
Delete Branch
新增
Advanced...
Repository browser
bitbucketweb
URL

CI Job設定

Build Triggers(使用Cron語法)

- [ ] Build after other projects are built
- [ ] Trigger builds remotely (e.g., from scripts)
- [x] Build periodically

Schedule

```
35 4 * * *
```

- [x] Poll SCM

Schedule

```
25 4 * * *
```

Build

Execute shell

命令

```
cd week5/messageBoard
npm install
grunt -v deploy
cd /data/apps/MessageBoard
forever stop build/cluster.js || true
forever start -o out.log -e err.log -a build/cluster.js
```

Resources

- http://nodejs.tw/ http://cnodejs.org/
- https://www.facebook.com/NodeJS.tw
- http://book.nodejs.tw/
- http://www.nodebeginner.org/
- http://visionmedia.github.io/masteringnode/

Resources by me

- http://rettamkrad.blogspot.tw/
- https://www.facebook.com/thingsaboutwebdev