Build Trust in Your Build-to-Deployment Flow!

Fred Simon, JFrog

About me

- ✓ Fred Simon
- ✓ Chief Architect @JFrog
- ✓ Trying to tell machine to behave for the last 20 years
- ✓ From Consulting to DevOps in the last 13 years

Agenda

- ✓ The cloud silver bullet
- ✓ The right tool for the job
- ✓ Binaries all the way
- ✓ The black magic
  of releasing

LOTS OF LIVE
DEMOS AHEAD!

The New Silver Bullet

EVERYTHING *aaS

What’s good about *aaS?

✓ *aaS features Continuous Delivery

Continuous Delivery FTW

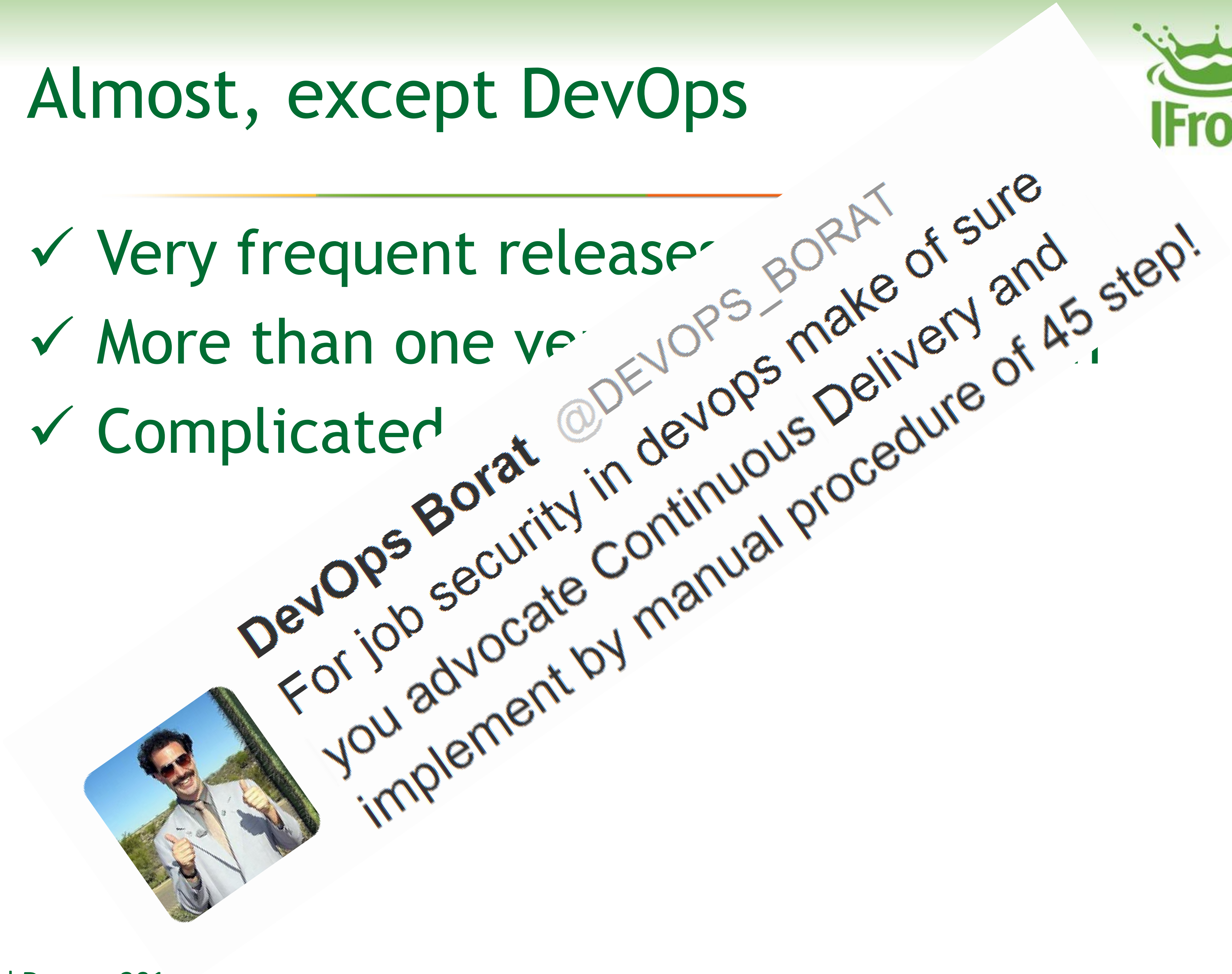
- ✓ Advantages for the user:
  - > Always on the latest version
- ✓ Advantages for the ISV:
  - > Agile
  - > Rapid feedback
  - > Users are the best beta-testers
  - > No long-term support
- ✓ Everybody Wins?

Almost, except DevOps

- ✓ Very frequent releases
- ✓ More than one version in production
- ✓ Complicated access levels

Almost, except DevOps

- ✓ Very frequent releases
- ✓ More than one ve
- ✓ Complicated



DevOps Borat @DEVOPS_BORAT

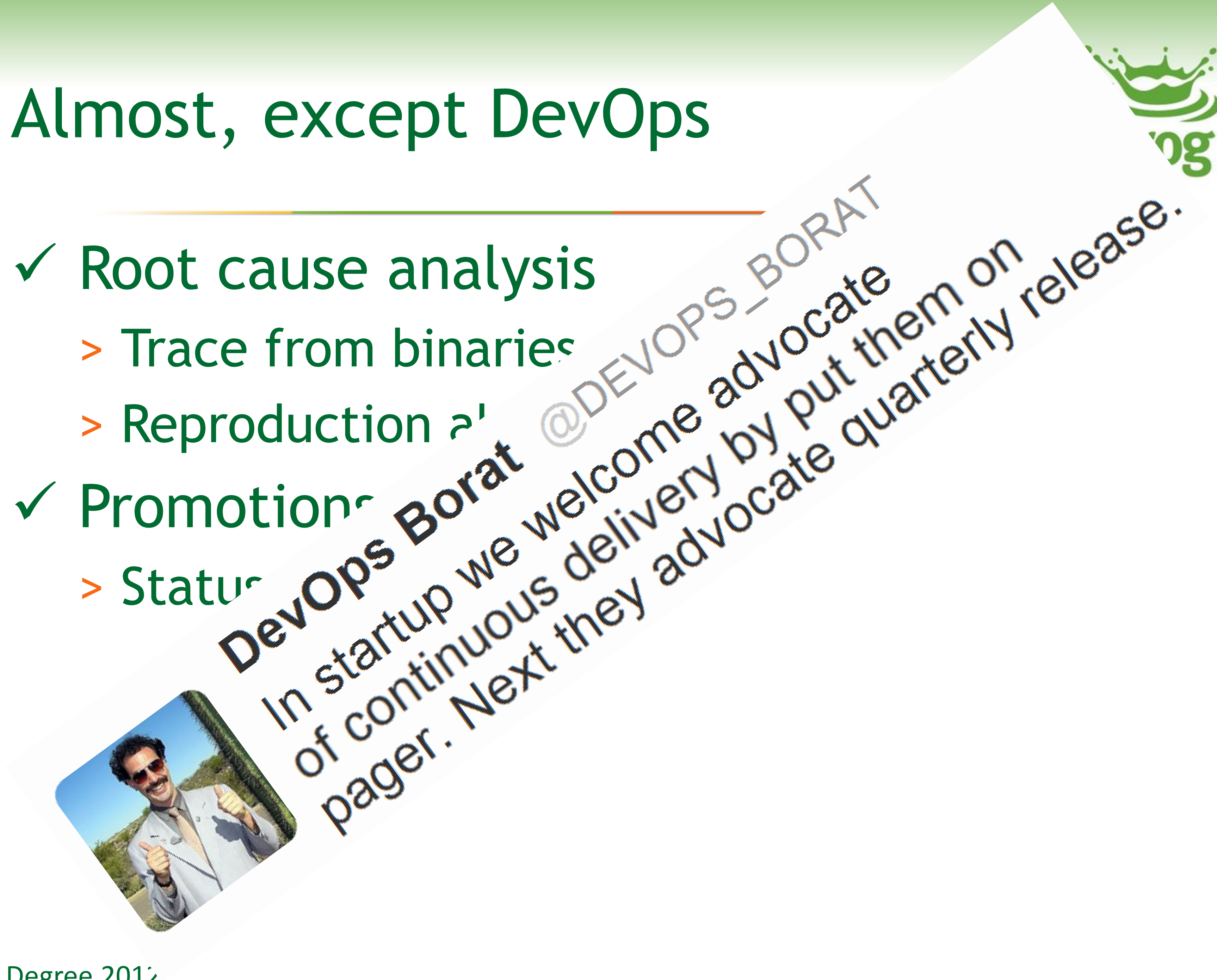
For job security in devops make of sure you advocate Continuous Delivery and implement by manual procedure of 45 step!

Almost, except DevOps

- ✓ Root cause analysis
  - > Trace from binaries to sources
  - > Reproduction abilities
- ✓ Promotions
  - > Status changes

Almost, except DevOps

- ✓ Root cause analysis
  - > Trace from binaries
  - > Reproduction a
- ✓ Promotion
  - > Statu



DevOps Borat @DEVOPS_BORAT

In startup we welcome advocate of continuous delivery by put them on pager. Next they advocate quarterly release.

Sounds Familiar?

- ✓ Agile principles applied for DevOps
- ✓ We have good tooling for Agile development
  - > Version Control
  - > Unit Testing (and coverage)
  - > Build Servers
  - > Hot Swap tools
- ✓ What's up with tooling for Agile DevOps?

Agile Tooling for DevOps Checklist

- ✓ Versioning
- ✓ Access control
- ✓ Traceability
- ✓ Promotions
- ✓ Tags and annotations
- ✓ Search

How Do I Know?

- ✓ JFrog SaaS offering
  - > Artifactory Online
  - > Gradle, Grails, SpringSource, Typesafe, Jenkins, etc.
- ✓ We build, release and eat our own dog food
  - > Continuously

The Right Tool for the Job

HERE COMES BINARY REPOSITORY

Here Comes Binary Repository

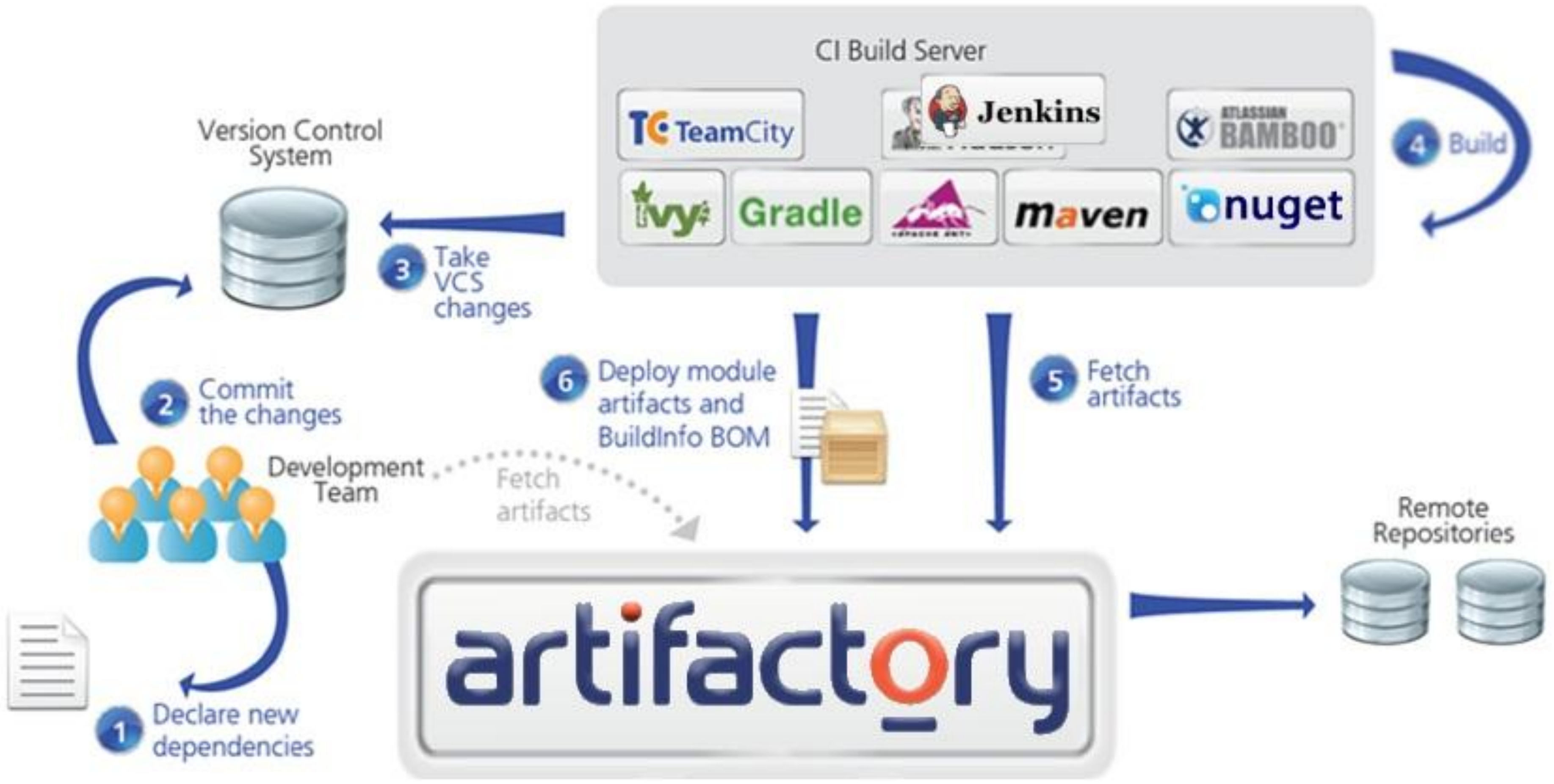
- ✓ E.g. Artifactory
- ✓ Main feature – Smart Storage
- ✓ Much more than passive storage
- ✓ Critical for CI and ALM

Tooling Chain

--- +++ git
SUBVERSION
PERFORCE

ivy
maven
yum
yellowdog updater modified
nuget

Artifactory in DevOps Ecosystem



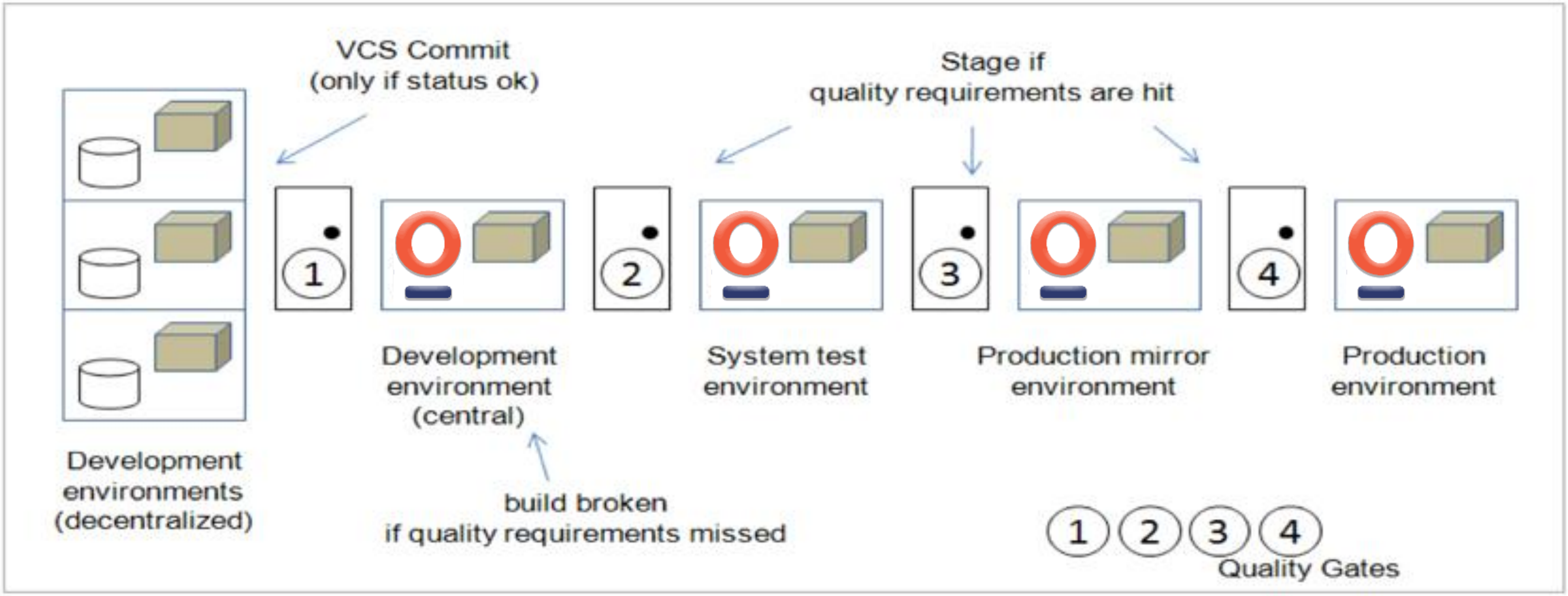
Meet Artifactory

DEMO TIME!

Binaries All the Way

- ✓ From some point product in your lifecycle, all you care about is binaries
- ✓ Lots of things to do after the software is built

The Release Pipeline


VCS Commit
(only if status ok)
Stage if
quality requirements are hit
1
2
3
4
Development environments (decentralized)
Development environment (central)
System test environment
Production mirror environment
Production environment
build broken
if quality requirements missed
1 2 3 4
Quality Gates

Source: Agile ALM, Michael Hüttermann, Manning Publications Co.

Passing the software to QA

- ✓ Different access rights
- ✓ Different physical location
- ✓ Ability to annotate

Staging and Preproduction

- ✓ Replication of Production environment
  - > Lock versions of dependencies and artifacts
- ✓ Allow access to set of users

Going to Production

- ✓ Convert staging binaries to production
- ✓ Allow public access
- ✓ Change settings
- ✓ Tag

Traceability

- ✓ Binaries should be traceable in every stage
  - > Sources
  - > Dependencies
  - > Environment details
  - > Tags
- ✓ Where's the information?
  - > Version Control System
  - > Build Server

JFrog

artifactory

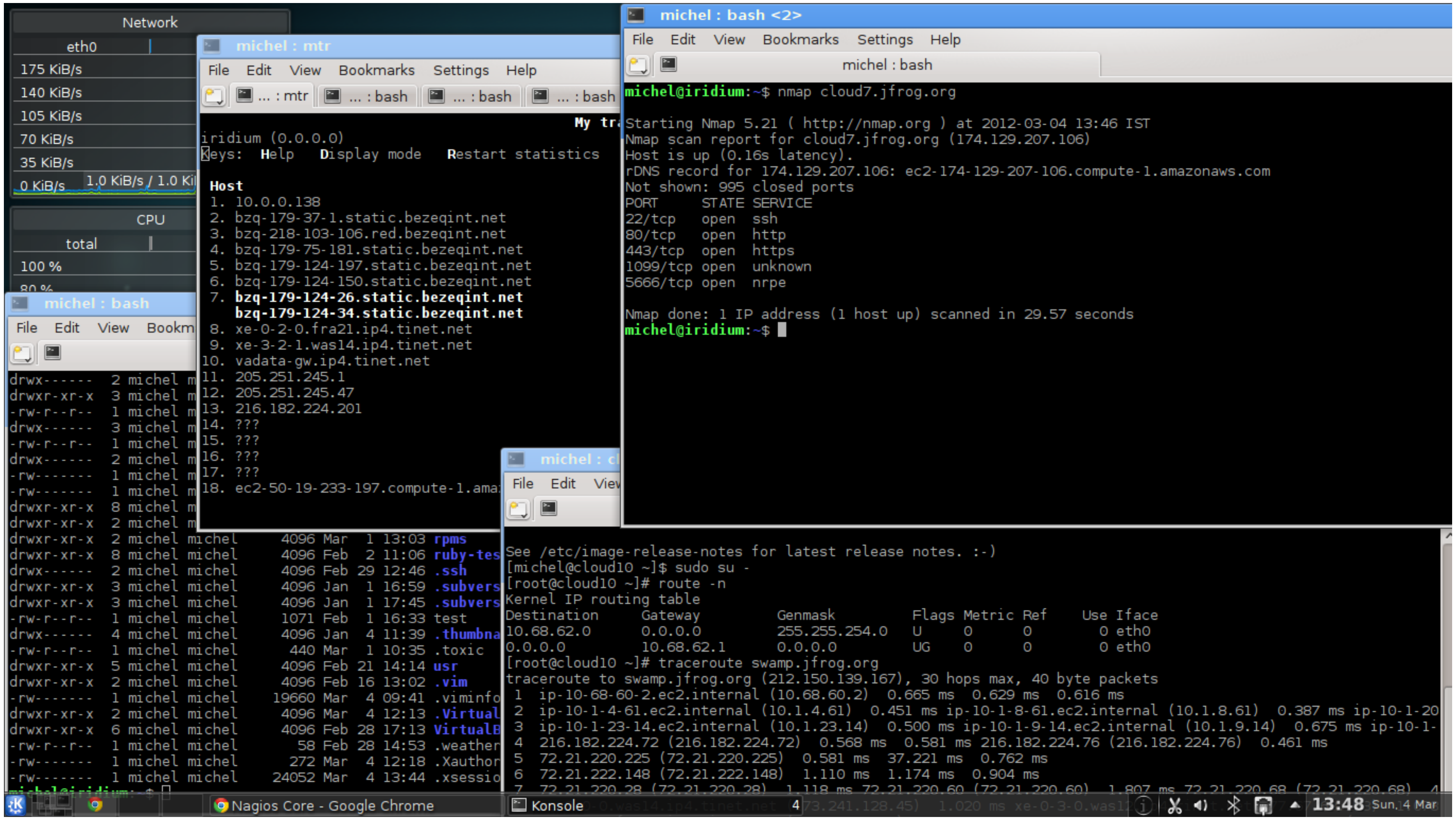
DevOps

WHAT MY FRIENDS THINK I DO

What Others Think I Do

What I Think I Do

What I Really Do



```
My tra
iridium (0.0.0.0)
Keys:  Help   Display mode   Restart statistics

 Host
 1. 10.0.0.138
 2. bzq-179-37-1.static.bezeqint.net
 3. bzq-218-103-106.red.bezeqint.net
 4. bzq-179-75-181.static.bezeqint.net
 5. bzq-179-124-197.static.bezeqint.net
 6. bzq-179-124-150.static.bezeqint.net
 7. bzq-179-124-26.static.bezeqint.net
    bzq-179-124-34.static.bezeqint.net
 8. xe-0-2-0.fra21.ip4.tinet.net
 9. xe-3-2-1.was14.ip4.tinet.net
10. vadata-gw.ip4.tinet.net
11. 205.251.245.1
12. 205.251.245.47
13. 216.182.224.201
14. ???
15. ???
16. ???
17. ???
18. ec2-50-19-233-197.compute-1.ama
```

```
michel@iridium:~$ nmap cloud7.jfrog.org

Starting Nmap 5.21 ( http://nmap.org ) at 2012-03-04 13:46 IST
Nmap scan report for cloud7.jfrog.org (174.129.207.106)
Host is up (0.16s latency).
rDNS record for 174.129.207.106: ec2-174-129-207-106.compute-1.amazonaws.com
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
1099/tcp open  unknown
5666/tcp open  nrpe

Nmap done: 1 IP address (1 host up) scanned in 29.57 seconds
michel@iridium:~$ 
```

```
drwx------  2 michel m
drwxr-xr-x  3 michel m
-rw-r--r--  1 michel m
drwx------  3 michel m
-rw-r--r--  1 michel m
drwx------  2 michel m
-rw-------  1 michel m
-rw-------  1 michel m
drwxr-xr-x  8 michel m
drwxr-xr-x  2 michel m
drwxr-xr-x  2 michel michel     4096 Mar  1 13:03 rpms
drwxr-xr-x  8 michel michel     4096 Feb  2 11:06 ruby-tes
drwx------  2 michel michel     4096 Feb 29 12:46 .ssh
drwxr-xr-x  3 michel michel     4096 Jan  1 16:59 .subvers
drwxr-xr-x  3 michel michel     4096 Jan  1 17:45 .subvers
-rw-r--r--  1 michel michel     1071 Feb  1 16:33 test
drwx------  4 michel michel     4096 Jan  4 11:39 .thumbna
-rw-r--r--  1 michel michel      440 Mar  1 10:35 .toxic
drwxr-xr-x  5 michel michel     4096 Feb 21 14:14 usr
drwxr-xr-x  2 michel michel     4096 Feb 16 13:02 .vim
-rw-------  1 michel michel    19660 Mar  4 09:41 .viminfo
drwxr-xr-x  2 michel michel     4096 Mar  4 12:13 .Virtual
drwxr-xr-x  6 michel michel     4096 Feb 28 17:13 VirtualB
-rw-r--r--  1 michel michel       58 Feb 28 14:53 .weather
-rw-------  1 michel michel      272 Mar  4 12:18 .Xauthor
-rw-------  1 michel michel    24052 Mar  4 13:44 .xsessio
michel@iridium:~$ 
```

```
See /etc/image-release-notes for latest release notes. :-)
[michel@cloud10 ~]$ sudo su -
[root@cloud10 ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.68.62.0      0.0.0.0         255.255.254.0   U     0      0        0 eth0
0.0.0.0         10.68.62.1      0.0.0.0         UG    0      0        0 eth0
[root@cloud10 ~]# traceroute swamp.jfrog.org
traceroute to swamp.jfrog.org (212.150.139.167), 30 hops max, 40 byte packets
 1  ip-10-68-60-2.ec2.internal (10.68.60.2)  0.665 ms  0.629 ms  0.616 ms
 2  ip-10-1-4-61.ec2.internal (10.1.4.61)  0.451 ms ip-10-1-8-61.ec2.internal (10.1.8.61)  0.387 ms ip-10-1-20
 3  ip-10-1-23-14.ec2.internal (10.1.23.14)  0.500 ms ip-10-1-9-14.ec2.internal (10.1.9.14)  0.675 ms ip-10-1-
 4  216.182.224.72 (216.182.224.72)  0.568 ms  0.581 ms 216.182.224.76 (216.182.224.76)  0.461 ms
 5  72.21.220.225 (72.21.220.225)  0.581 ms  37.221 ms  0.762 ms
 6  72.21.222.148 (72.21.222.148)  1.110 ms  1.174 ms  0.904 ms
 7  72.21.220.28 (72.21.220.28)  1.118 ms 72.21.220.60 (72.21.220.60)  1.807 ms 72.21.220.68 (72.21.220.68)  4
```

What I should Do

Target: Automation

- ✓ It's impossible to release frequently with manual procedures
  - > While maintaining quality
- ✓ Use your binaries storage to release

Put your repository to work

THE MAGIC OF RELEASE

Release With Release Candidate

- ✓ Your next build is release-candidate
- ✓ Once successfully built and tested, click the button
  - > Automatic versions switch
    - › From integration to release
  - > Right place to put your binaries
    - › Move from Staging to Public
  - > Automatic VCS tagging for the release

Release With Release Candidate

- ✓ Process:
  1. Produce and build snapshots until satisfied
  2. Once satisfied, build release candidate
  3. Stage RC, check and verify
  4. Once checked, release

Release With Artifactory: Mechanics

- ✓ : The Artifactory Jenkins Plugin
  - > Gathers build information
  - > Uploads artifacts in bulk
  - > Uploads build information
  - > Provides bi-directional linking
- ✓ Release Management
  - > Changes versions in build script
  - > Allows to Choose repository to deploy to
  - > Creates a VCS tag/branch

Release With Release Candidate

DEMO TIME!

OOTB Release Management

- ✓ Pros
  - > Out of the box
  - > Supports the “by the book” release cycle
  - > Supports majority of the tools
- ✓ Cons
  - > Not flexible
  - > Not extensible
  - > May not suite your case

We Know: We Don't Know Better

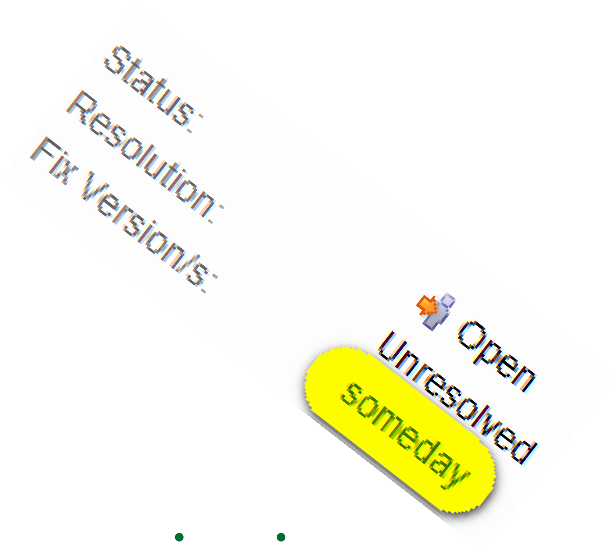
- ✓ YMMV (great deal)
- ✓ Write your own release logic
- ✓ Pre and post build deploy hooks

Flexible Release

- ✓ Code your release strategy
  - > Versioning scheme
  - > VCS (tagging, branching, commit comments)
  - > Promotion hook (copy/move, comments, status)
- ✓ Available by REST

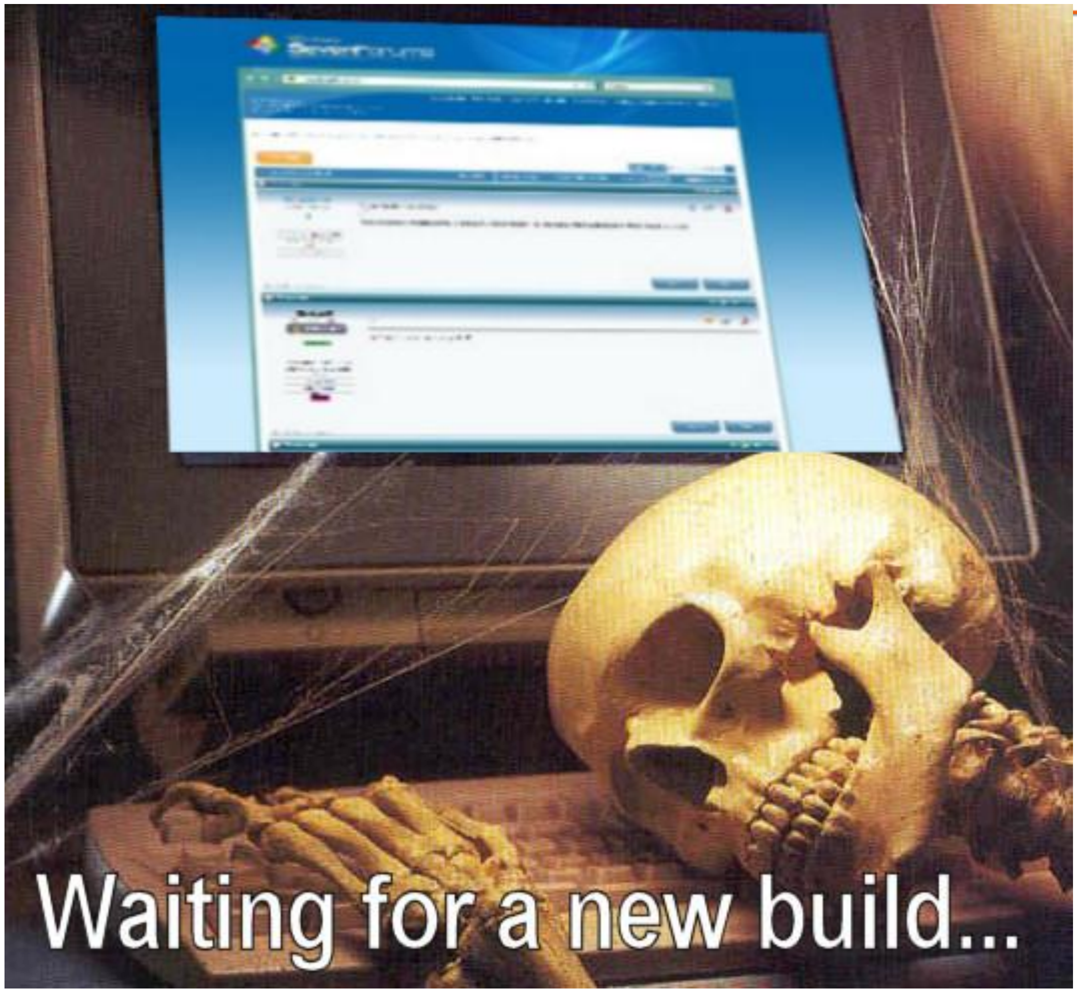
Controlling Versioning Scheme

- ✓ Classic versioning scheme:
  - > Release version
    - › 2.0.3
  - > Integration version
    - › 2.0.4-SNAPSHOT
- ✓ YMMV
  - > Write your own strategy for versioning
  - > Dynamic Groovy code



Status: Open
Resolution: Unresolved
Fix Version/s: someday

Example: Promotion of Snapshots



- ✓ Sometimes the build takes long time…
- ✓ But that’s the silly reason

Release With Release Candidate

✓ Process:

1. Produce and build snapshots until satisfied
2. Once satisfied, build release candidate
3. Stage RC, check and verify
4. Once checked, release

Can you spot the problem?

Release With Release Candidate

✓ Process:

1. Produce and build snapshots until satisfied
2. **Once satisfied, build release candidate**
3. Stage RC, check and verify
4. Once checked, release

Redundant build

Release With Release Candidate

- ✓ Lots of things can go wrong during one more build
- ✓ If we won’t build it, we won’t screw it
- ✓ Process:
  1. Produce and build snapshots until satisfied
  2. When satisfied, check and verify
  3. Once checked, release

Example: Promotion of Snapshots

- ✓ Choose existing build to become a release
- ✓ Using REST API without build server
- ✓ Invoke Promotion plugin
  - > Convert to next version
  - > Tag, branch, etc.
  - > Promote (copy/move)

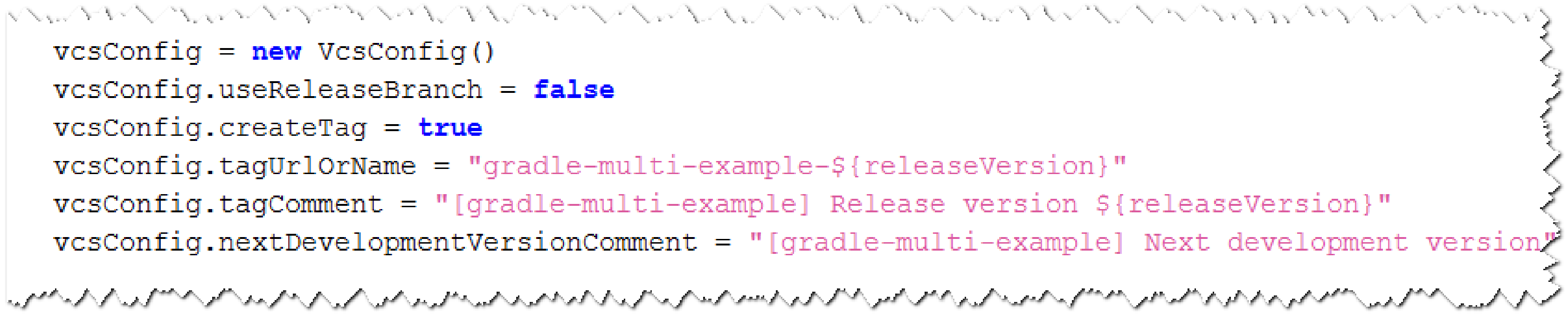
Plugin What?

CODE TIME!

Plugin Code

- ✓ Groovy goodness
- ✓ Executed directly in Artifactory
- ✓ Uses Public API
  - > Search for artifacts
  - > Search for builds
  - > Copy/Move artifacts
  - > Manipulate files
    - › E.g. change versions in descriptors

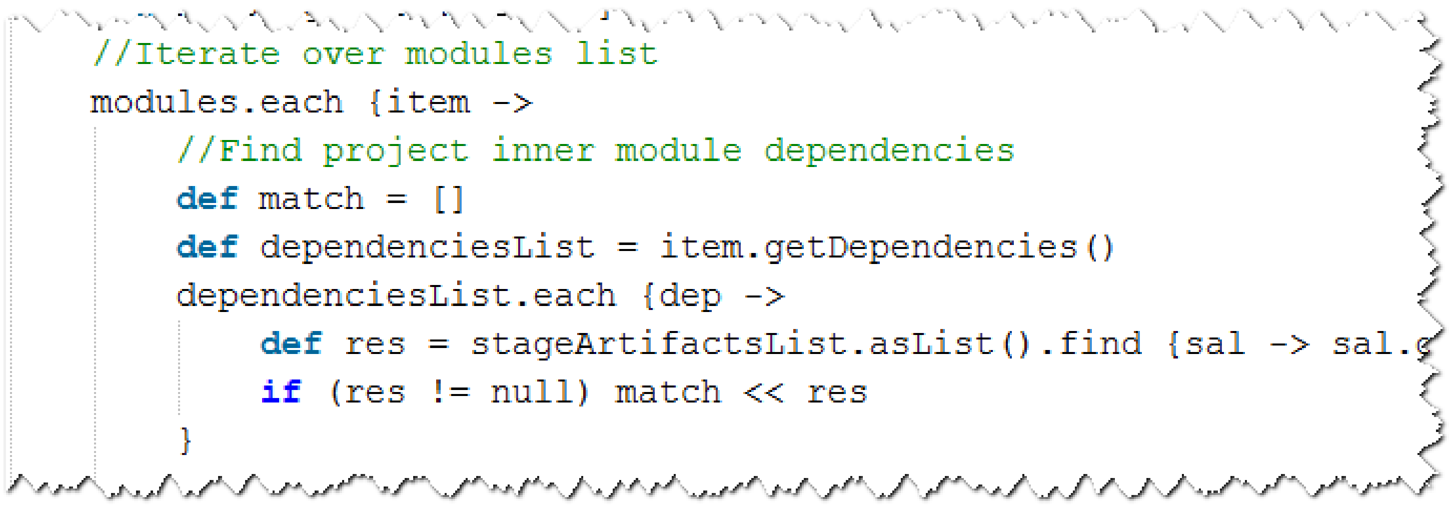
Plugin Code



```
vcsConfig = new VcsConfig()
vcsConfig.useReleaseBranch = false
vcsConfig.createTag = true
vcsConfig.tagUrlOrName = "gradle-multi-example-${releaseVersion}"
vcsConfig.tagComment = "[gradle-multi-example] Release version ${releaseVersion}"
vcsConfig.nextDevelopmentVersionComment = "[gradle-multi-example] Next development version"
```

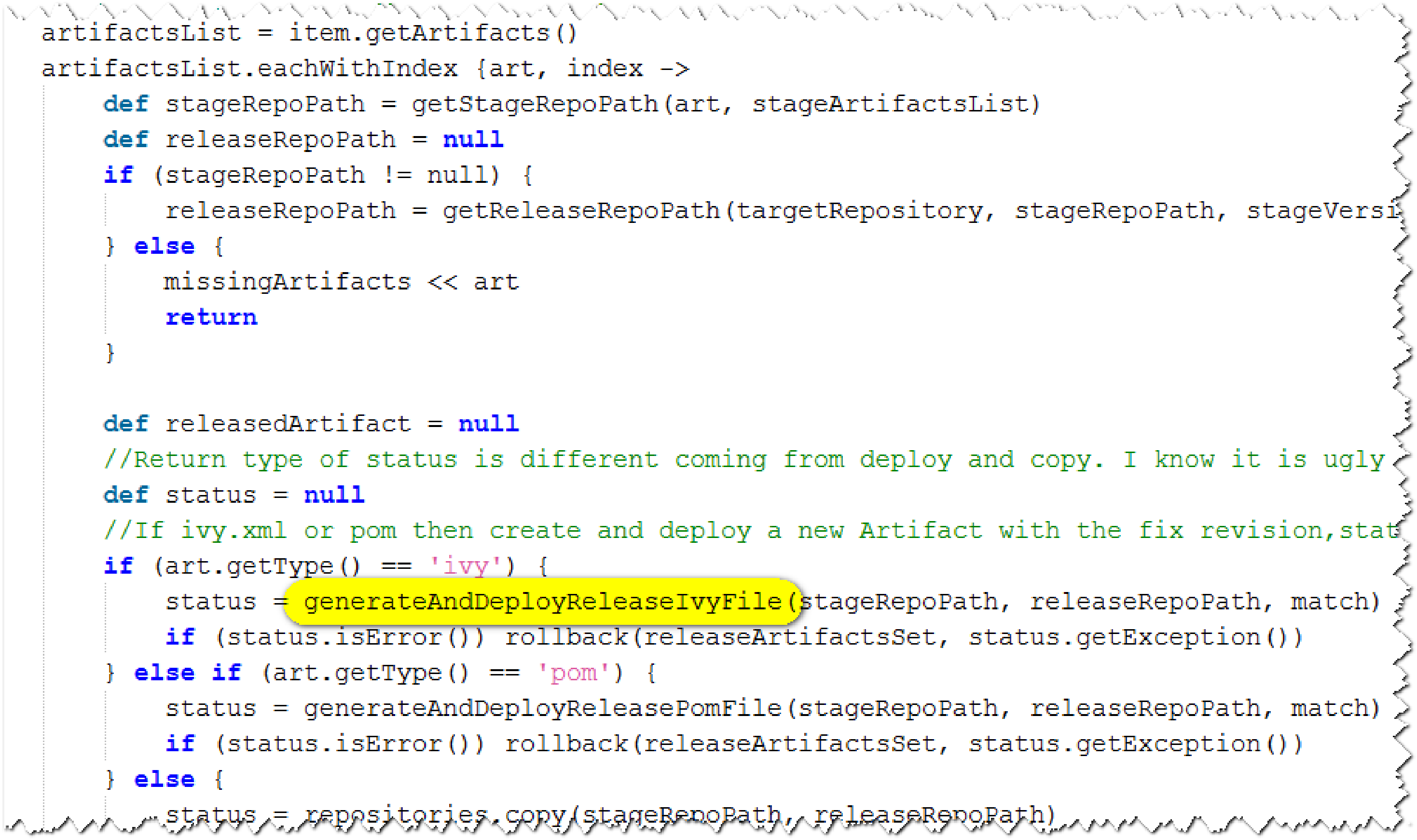
✓ Manipulating Version Control Systems

Plugin Code


```
//Iterate over modules list
modules.each {item ->
    //Find project inner module dependencies
    def match = []
    def dependenciesList = item.getDependencies()
    dependenciesList.each {dep ->
        def res = stageArtifactsList.asList().find {sal -> sal.g
        if (res != null) match << res
    }
```

✓ Manipulating BuildInfo object

Plugin Code



```
artifactsList = item.getArtifacts()
artifactsList.eachWithIndex {art, index ->
    def stageRepoPath = getStageRepoPath(art, stageArtifactsList)
    def releaseRepoPath = null
    if (stageRepoPath != null) {
        releaseRepoPath = getReleaseRepoPath(targetRepository, stageRepoPath, stageVersi
    } else {
        missingArtifacts << art
        return
    }

    def releasedArtifact = null
    //Return type of status is different coming from deploy and copy. I know it is ugly
    def status = null
    //If ivy.xml or pom then create and deploy a new Artifact with the fix revision,stat
    if (art.getType() == 'ivy') {
        status = generateAndDeployReleaseIvyFile(stageRepoPath, releaseRepoPath, match)
        if (status.isError()) rollback(releaseArtifactsSet, status.getException())
    } else if (art.getType() == 'pom') {
        status = generateAndDeployReleasePomFile(stageRepoPath, releaseRepoPath, match)
        if (status.isError()) rollback(releaseArtifactsSet, status.getException())
    } else {
        status = repositories.copy(stageRepoPath, releaseRepoPath)
```

✓ Creating and replacing artifacts

Calling REST API With CURL

```
http://repo-demo:8080/artifactory/api/plugins/build/promote/snapshotToRelease/gradle-multi-example/1?params=snapExp=d14|targetRepository=gradle-release-local
```

Calling REST API With CURL

```
http://repo-demo:8080/
artifactory/api/plugins/
build/promote/snapshotToRelease/
gradle-multi-example/1?
params=snapExp=d14|
targetRepository=gradle-release-
local
```

Calling REST API With CURL

```
http://repo-demo:8080/
artifactory/api/plugins/
build/promote/snapshotToRelease/
gradle-multi-example/1?
params=snapExp=d14|
targetRepository=gradle-release-
local
```

Artifactory Server

Calling REST API With CURL

```
http://repo-demo:8080/
artifactory/api/plugins/
build/promote/snapshotToRelease/
gradle-multi-example/1?
params=snapExp=d14|
targetRepository=gradle-release-
local
```

Artifactory Server

Plugins API

Calling REST API With CURL

```
http://repo-demo:8080/
artifactory/api/plugins/
build/promote/snapshotTo
gradle-multi-example/1?
params=snapExp=d14|
targetRepository=gradle-release-
local
```

Artifactory Server

Plugins API

Plugin Name

Calling REST API With CURL

```
http://repo-demo:8080/
artifactory/api/plugins/
build/promote/snapshotToF
gradle-multi-example/1?
params=snapExp=d14|
targetRepository=gradle-release-
local
```

Artifactory Server

Plugins API

Plugin Name

Build Name and Number

Calling REST API With CURL

```
http://repo-demo:8080/
artifactory/api/plugins/
build/promote/snapshotTo
gradle-multi-example/1?
params=snapExp=d14|
targetRepository=gradle-release-
local
```

Artifactory Server

Plugins API

Plugin Name

Build Name and Number

Param: Versioning Scheme

Calling REST API With CURL

```
http://repo-demo:8080/
artifactory/api/plugins/
build/promote/snapshotToR
gradle-multi-example/1?
params=snapExp=d14|
targetRepository=gradle-release-
local
```

Artifactory Server

Plugins API

Plugin Name

Build Name and Number

Param: Versioning Scheme

Target repository for release

Recap: Promotion of Snapshots

- ✓ Choose existing build to become a release
- ✓ Using REST API without build server
- ✓ Invoke Promotion plugin
  - > Convert to next version
  - > Tag, branch, etc.
  - > Promote (copy/move)

Release by Snapshot Promotion

DEMO TIME!

4 Commandments of DevOps

- ✓ Automate everything
- ✓ Version everything
- ✓ Trace everything
- ✓ Report/Log/Feed back everything

Designed by Jessica Allen on Dribbble.com

4 Commandments of DevOps

- ✓ Automate everything
- Version everything
- ✓ Trace everything
- ✓ Report/Log/Feedback everything

Designed by Jessica Allen on Dribbble.com

Thank You!