CI / CD @ Liveperson

Gidi Samuels

www.liveperson.com

July 16, 2014

#jenkinsconf

Lead the CI Team at Liveperson
3 years in Liveperson

Lead the CI Team at Liveperson
3 years in Liveperson

Agenda

- **Introducing Liveperson and its CI**
- **Creating CI framework with Apache Maven**
- **How we've made 200 developers write in puppet?**
- **Scaling from 4 CI flows to > 100 and making our developers' life easier**
- **Using Jenkins as a deployment tool**
- **Roadmap: Auto provisioning and containers**

Introducing Liveperson and its CI

SaaS platform for creation of meaningful connections through real-time engagement

Monitor web visitor's behavior
(Over 1.5 B visits each month)

Conduct behavioral ranking

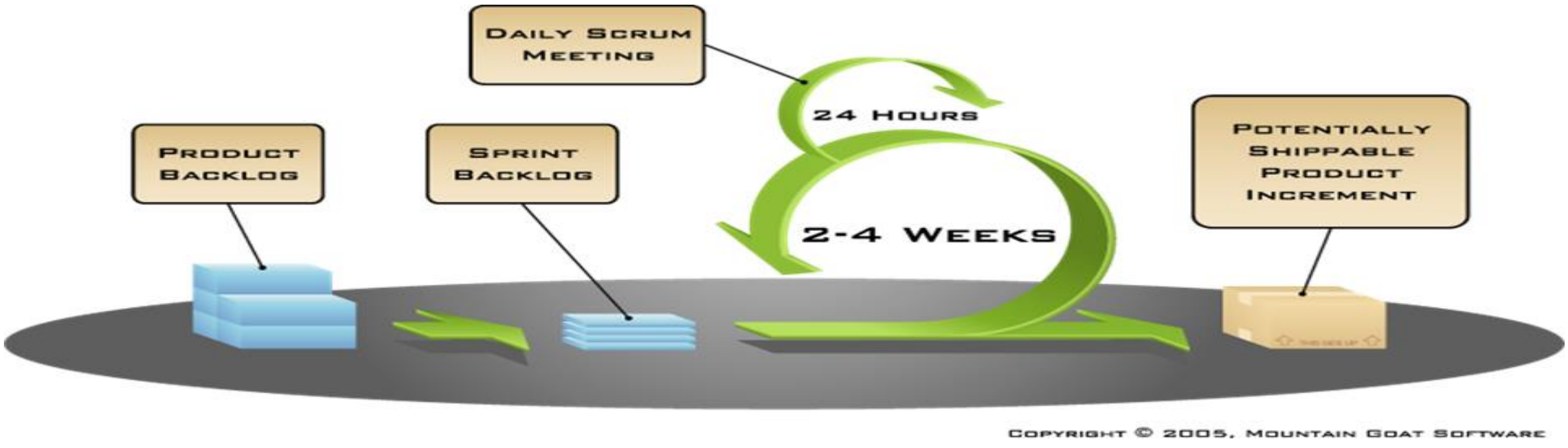
Provide the engagement platform
(Over 10 M chats each month)

SaaS & Cloud only

New LiveEngage platform - Simplify and Scale

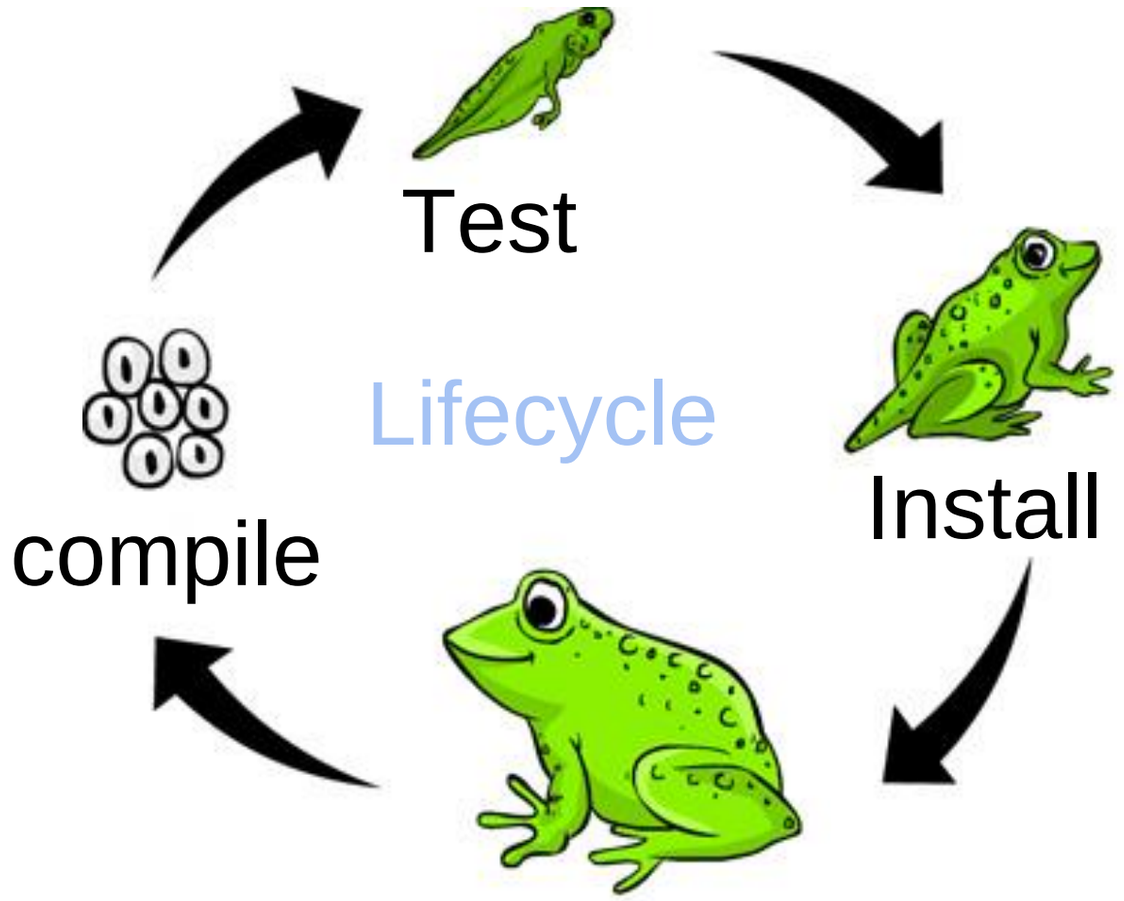
Numbers:

- 200 Developers
- 25 Scrum teams
- 500 Builds per day



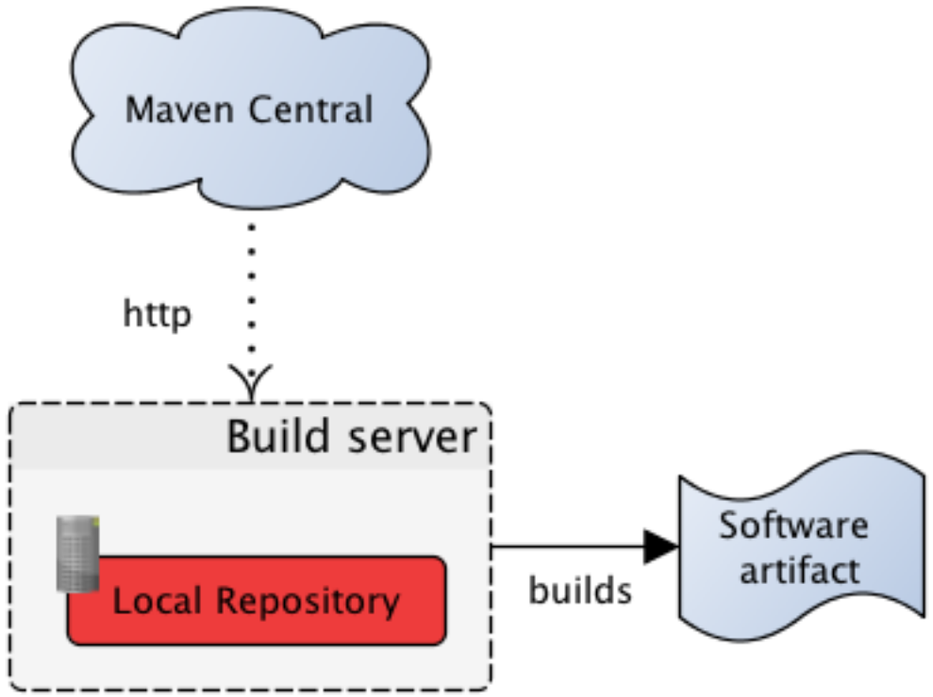
Creating CI framework with Apache Maven

Day 1 Build Config Tool



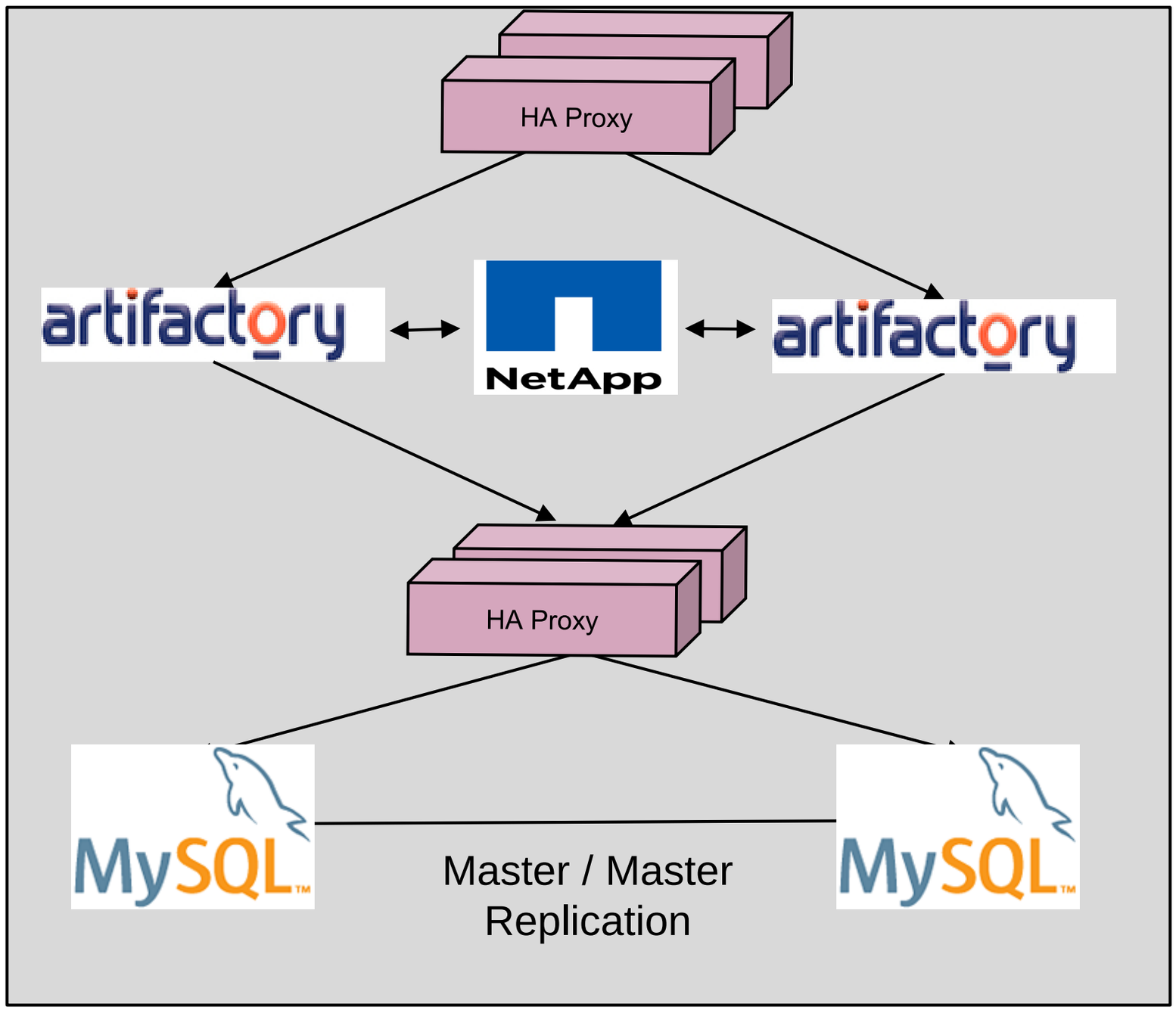
- Project management tool
- Well defined lifecycle
- Dependency management
- Makes the build process easy

Day 2 Maven Repository



- Started with single node
- Upgraded to High Availability
- Planning to add more instances for global development

Artifactory High Availability



Day 3 - TeamCity Pre-Test Commit

1

Get Code

2

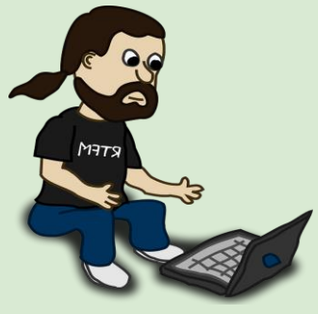
Pre-Test commit

Verify your change

3

4

Commit code



Subversion

git

Continuous Integration

Build Agent Pools

CLEAN

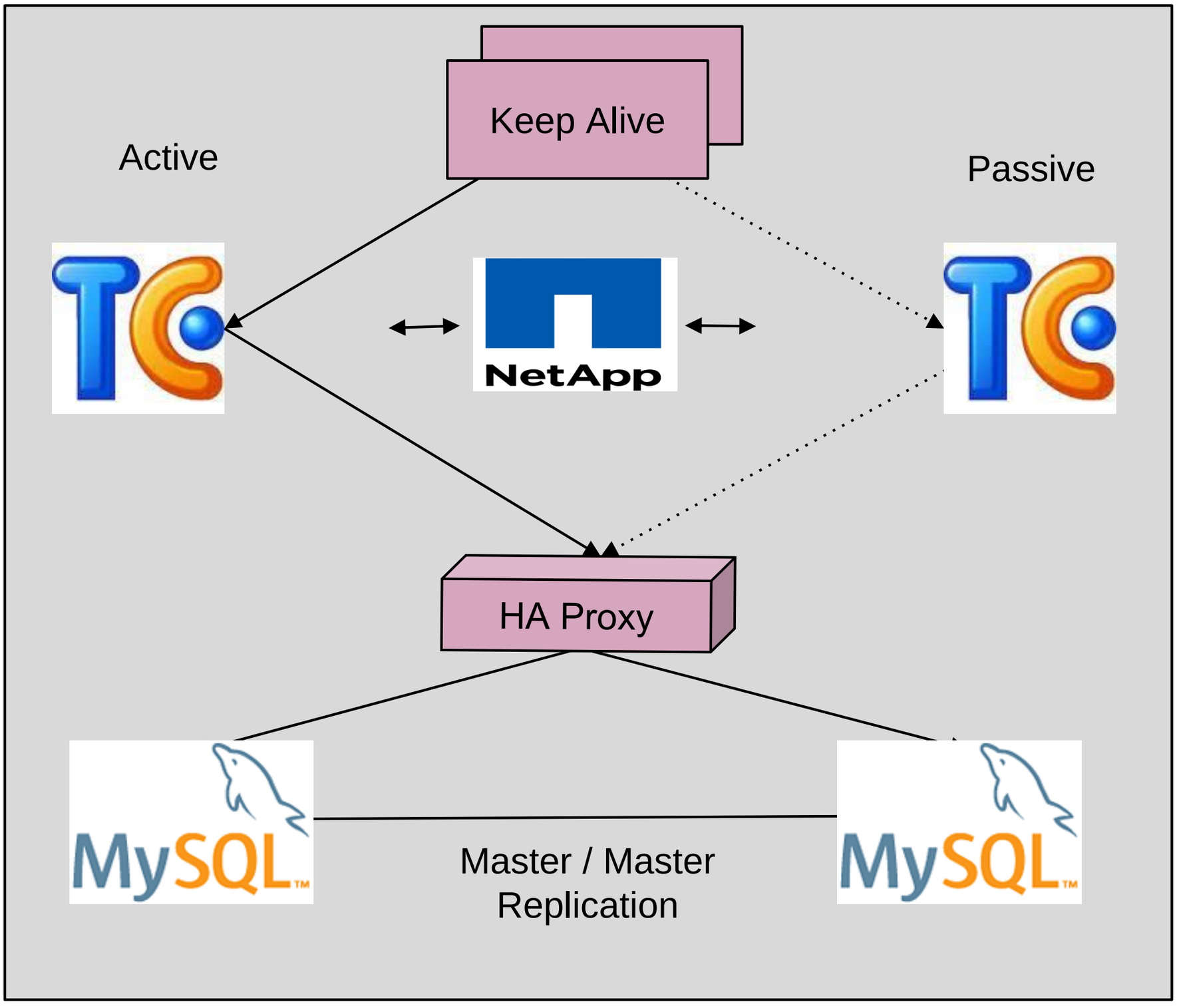
COMPILE

TEST

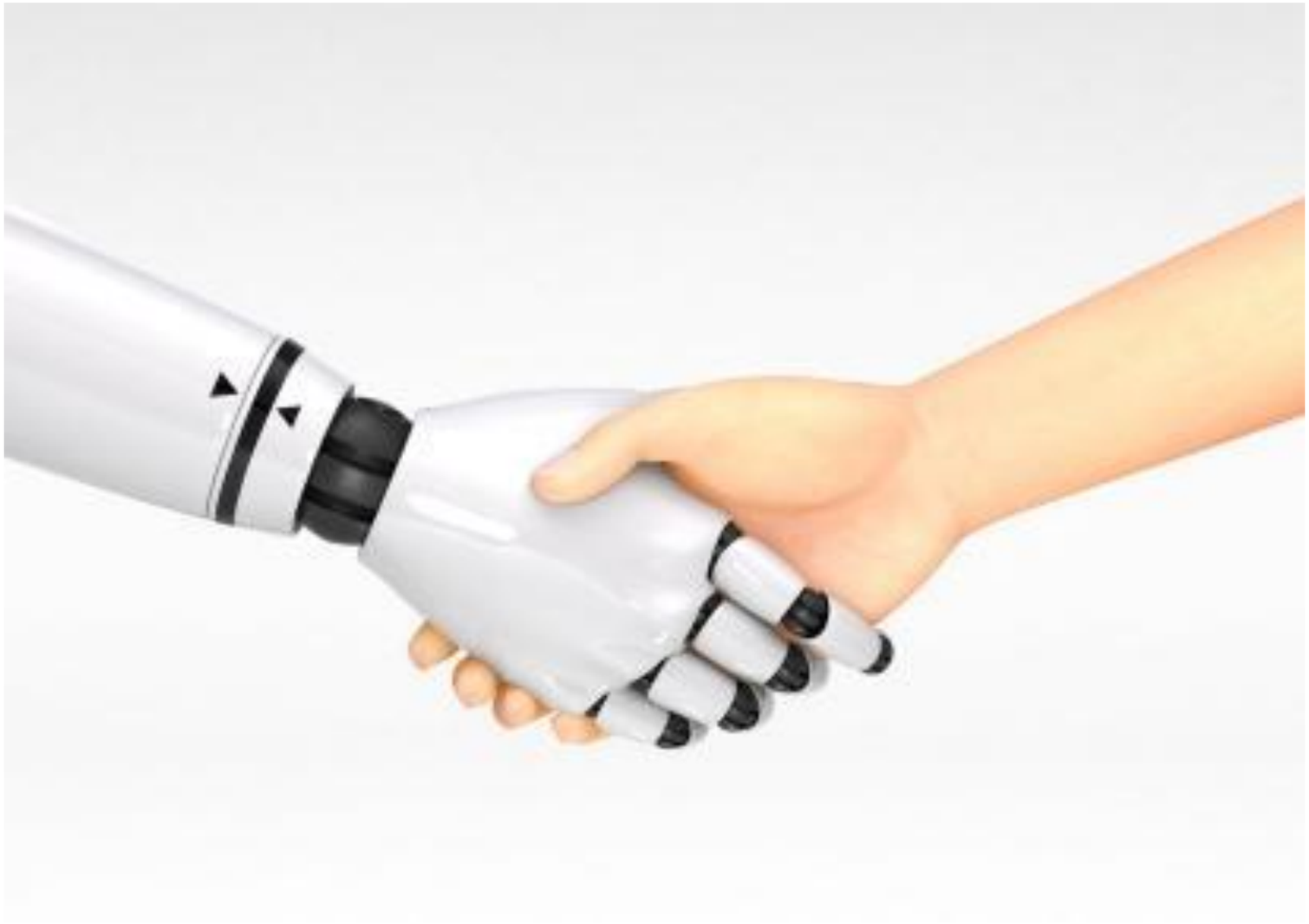
INSTALL

DEPLOY

TeamCity High Availability



We transformed our QA testes from manual to automate



Automation testing tools we use in our build

LP ADK

Measuring our coverage

Flow

Jacoco nightly activation profile	Jacoco is activated via nightly build
Compile services and attach sources	Use maven-source-plugin and attach sourcesfor the report
Deploy service on CI env	Deploy new rpm with jacocoagent enabled
Pre-integration	Extract all sources to one folder Reset jacoco output file
integration	Run your tests
Post-integration	Fetch jacoco output from server Generate report

Measuring our coverage

Flow

Jacoco nightly activation profile

Jacoco is activated via nightly build

Compile & attach s

Deploy env

Pre-inte

integration

Run your tests

Post-integration

Fetch jacoco output from server
Generate report

System test coverage report

Element	Missed Instructions	Cov.	Missed Branches	Cov.
com.liveperson.accountconfig.cdn.service.impl		62%		68%
com.liveperson.accountconfig.cdn.service.impl.cache		60%		54%
com.liveperson.accountconfig.cdn.service.impl.acclient		61%		58%
com.liveperson.accountconfig.cdn.management		55%		31%
com.liveperson.accountconfig.cdn.common.impl.monitoring		31%		50%
com.liveperson.accountconfig.cdn.facade.exception		33%		29%
com.liveperson.accountconfig.cdn.facade.impl		77%		50%
com.liveperson.accountconfig.cdn.service.impl.json		52%		30%
com.liveperson.accountconfig.cdn.common.impl.acclient		58%		75%
com.liveperson.accountconfig.cdn.application		78%		29%
com.liveperson.accountconfig.cdn.service.api		87%		65%
com.liveperson.accountconfig.cdn.common.impl.exception		55%		0%
com.liveperson.accountconfig.cdn.common.impl		67%		58%
Total	1,919 of 5,218	63%	145 of 325	55%

Measuring our coverage

Powered by

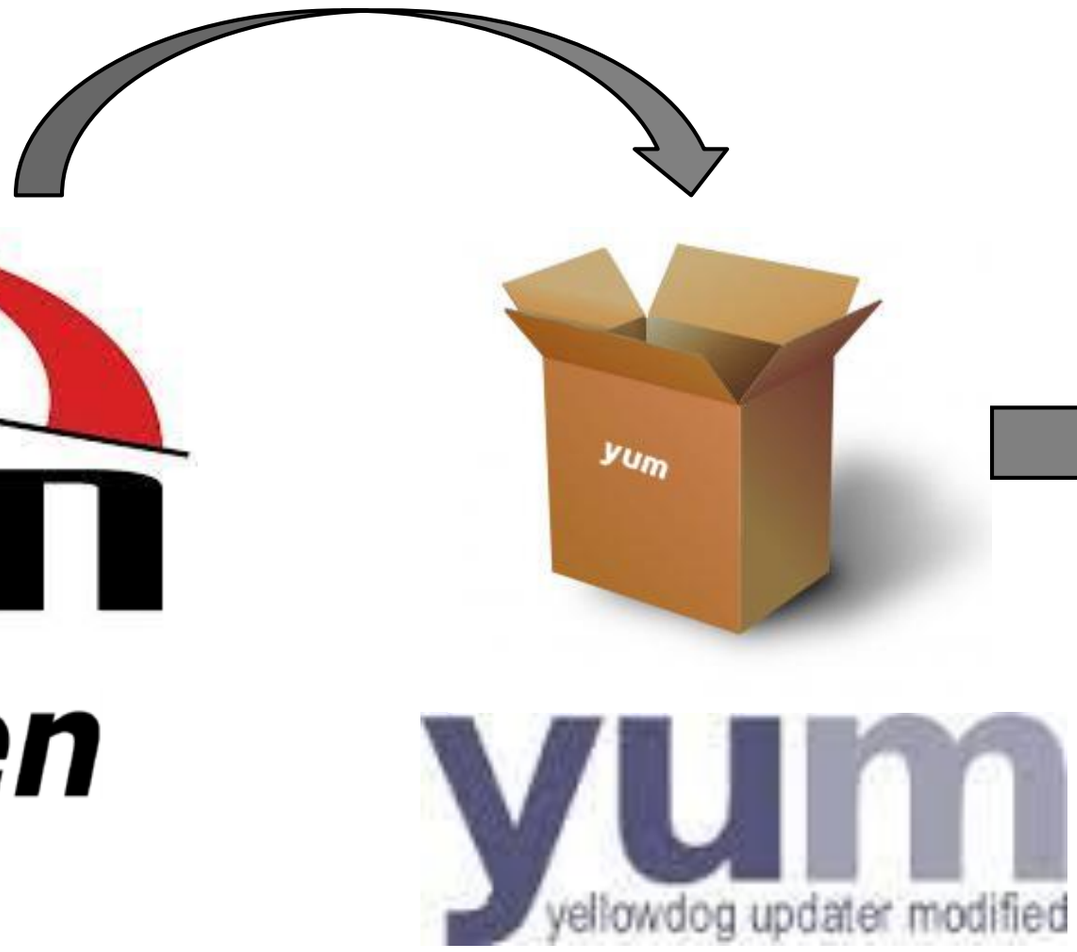
```
    for (ACClientTypeDefinition def : types.getTypes().values()) {

        AccountConfigClient accountConfigClient = null;

        if (!clientByPattern.containsKey(def.getCdnPatternType())) {
            accountConfigClient = acClientFactory.newAccountConfigClient(def);
            clientByPattern.put(def.getCdnPatternType(), accountConfigClient);
        }

        clientByType.put(getCacheName(def), accountConfigClient);
        accountConfigClient = clientByPattern.get(def.getCdnPatternType());

        try {
            registerType(def, accountConfigClient);
            log.info("Registered type: " + def.getParentType() + ", pattern:" + def.getCdnPatternType() + " to ac-client");
        } catch (ClientRegisterException e) {
            // failing to register a type will still allow other types to function
            log.error("Failed to register type: " + def.getParentType() + " to ac-client", e);
            // raise high severity snmp trap
            snmpMonitoringEventDispatcher.getMonitoringEventDispatcher().raiseEvent(new InternalACCDNErrorEvent(e, InternalACC
        }
    }
}
```

Post-integration

Fetch jacoco output from server
Generate report

Packaging

We use RPM as our packaging tool and repo



Production

Packaging

Included in maven-rpm-plugin

```
<groupId>org.codehaus.mojo</groupId>
<artifactId>rpm-maven-plugin</artifactId>
```

Standard configuration:

- projversion
- summary
- description
- mapping
- preinstallScriptlet
- postinstallScriptlet

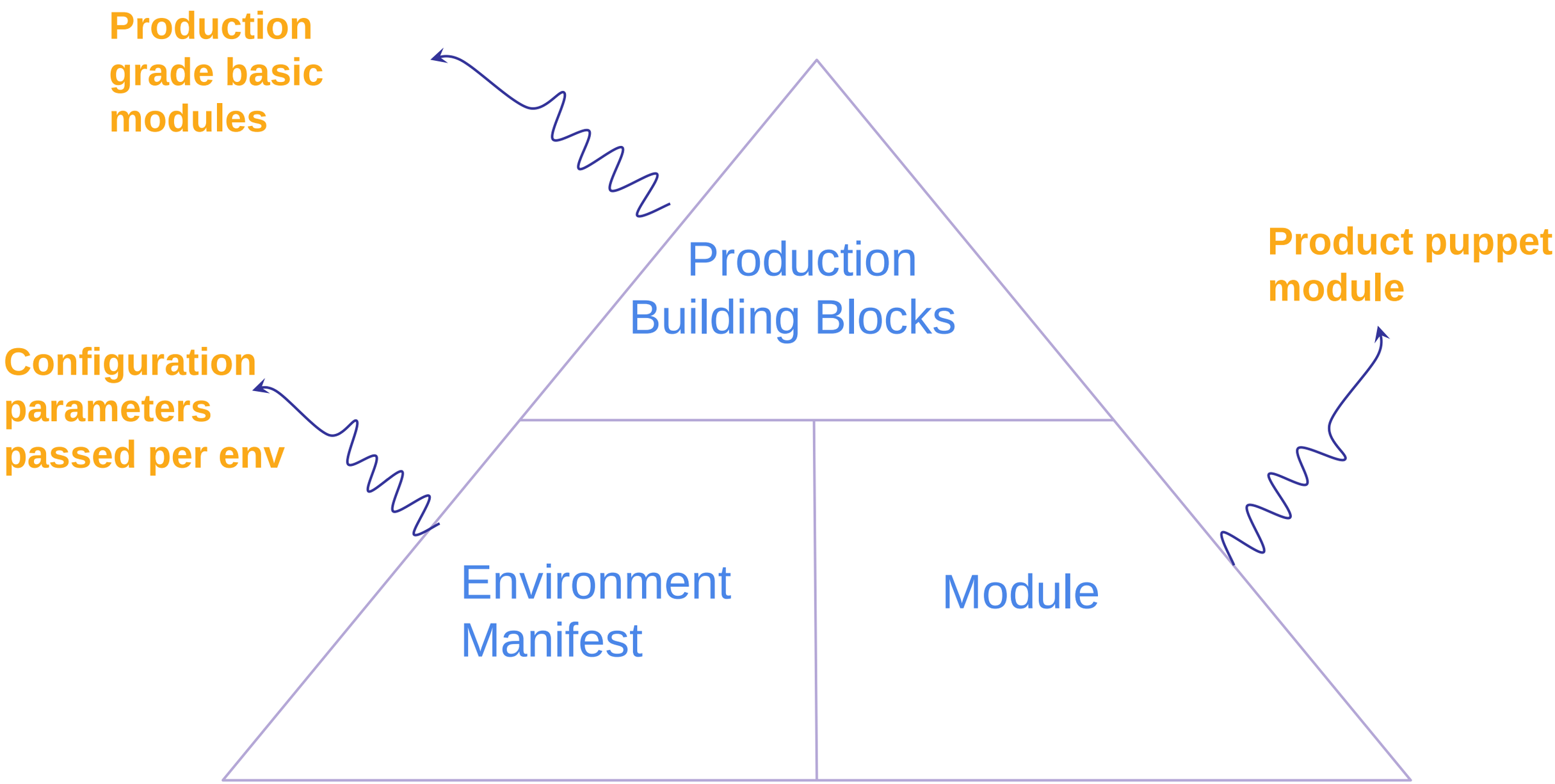
How we've made 200 developers write in puppet?

Configuration management

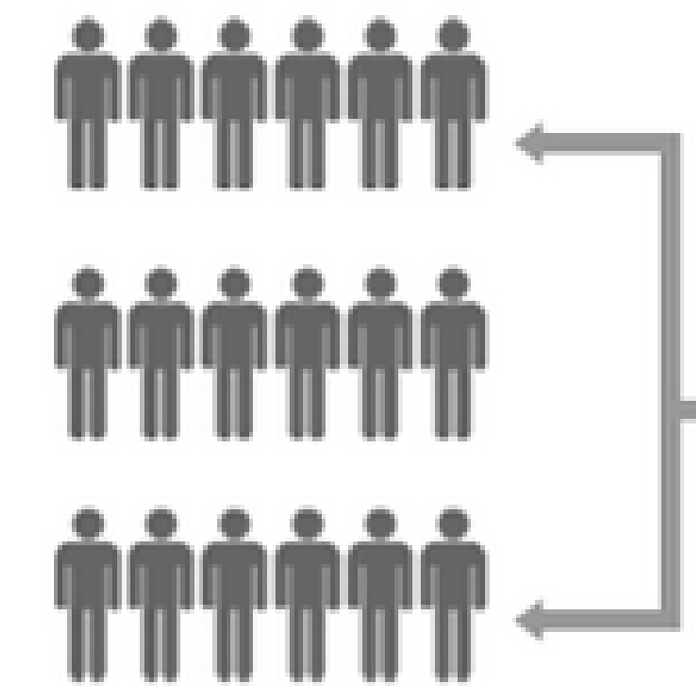
We deliver everything to production with PUPPET

- 200 developers program in Puppet to automate deployment
- Trained those 200 developers to write in Puppet
- We created standards in LP for deployment and configuration

Puppet configuration design



Configuration management



Development Teams

puppet labs

git

CD

Production

Mcollective

Scaling from 4 CI flows to > 100 and making our developers life easier

Step #1 Maven plugins

So how did we connect all these tools together ?
By extending Maven lifecycle with new maven plugins.

What is a plugin ?
Plugins are the central feature of Maven that allow for the reuse of common build logic across multiple projects.

Practice #1 Maven plugins

Practice #1 Maven plugins

Security analysis

During “nightly” builds and “release” build we added the following code analysis:

- Security vulnerability and regulation compliance

- Infrastructure vulnerability
- SaaS open source management tool

LIVEPERSON ?

How do i create a java project with Liveperson standard ?

- Where do I manage the code (SVN / GIT)
- What is LP project layout standard ?
- Which parentPom version to use ?
- How do I create the Puppet deployment Module?
- What about CI tools. How to create a build for my project ?

Step 2 Making our developers life easier

Java project

Stored in repository

RPM

Server to run on

Puppet module

Resources

LP ParentPom

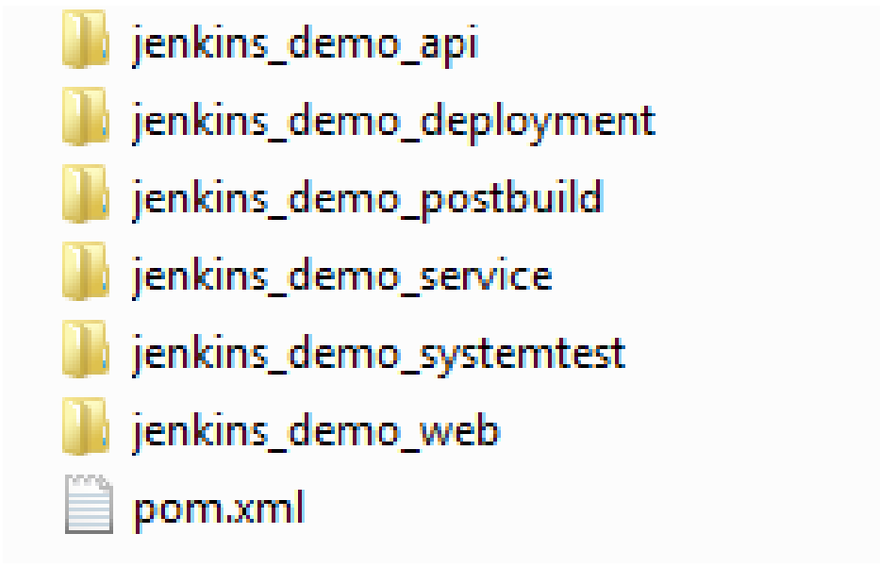
Promotion Tool

CI Build config

Code coverage

Security Analysis

The power of Maven Archetype



All together now

Commit

SVN

TeamCity

Jenkins

QA Servers

Production

E2E Servers

GDE Servers

Promotion Tool

compile

Unit tests

package

Install

System test

Code analysis

Security checks

Promote

maven

artifactory

JUnit.org

yum yellowdog updater modified

pulp

rpm

git

puppet labs

CI Servers Deploy

CHECKMARX

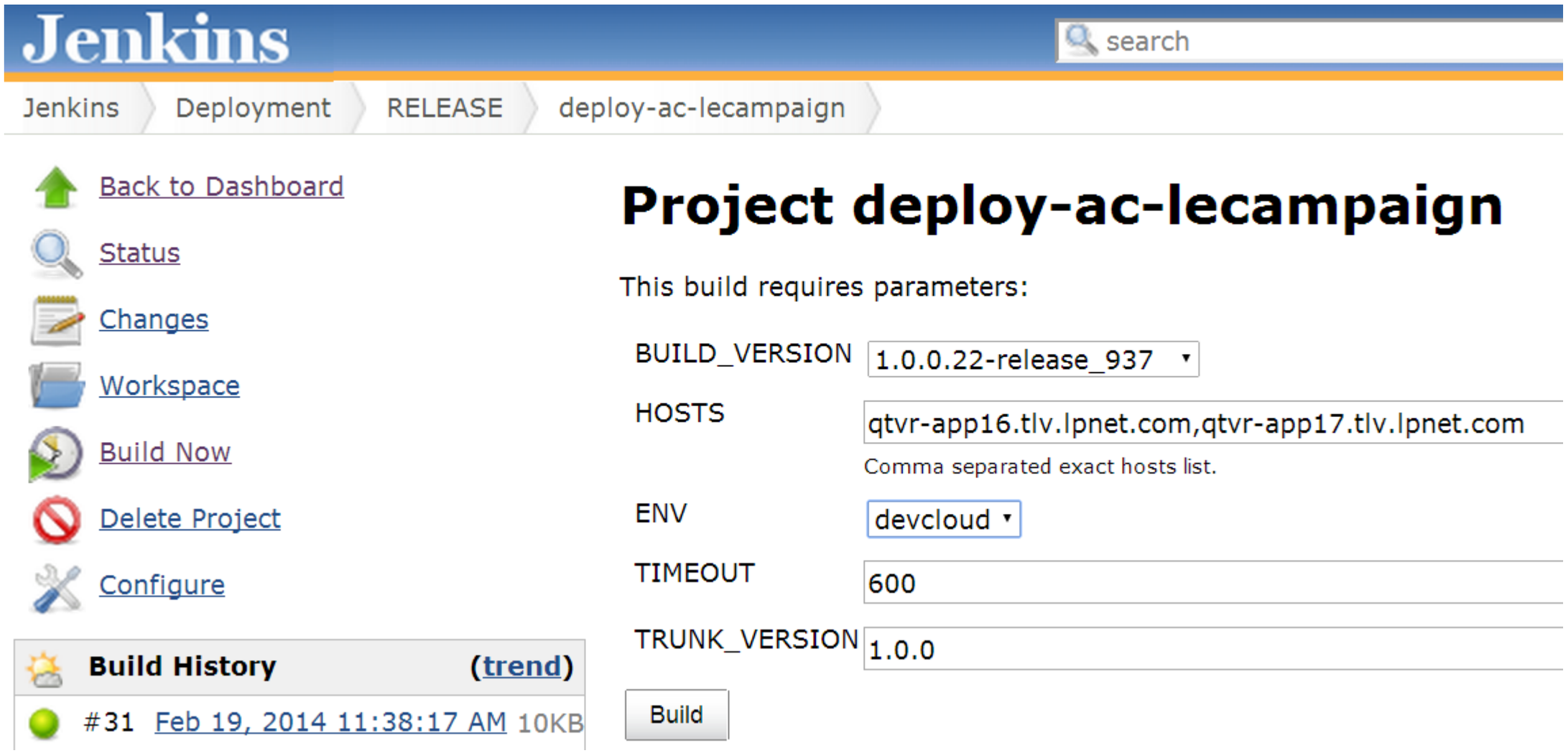
sonar

Nessus

WhiteSource

Selenium

Same deployment framework for CI / QA / PRODUCTION



Roadmap

- CI with Openstack on the fly
- Auto provisioning of full environments
- Moving with the same technology to Docker
- Support Global development
- GIT development

QA

We are hiring now for CI and Devops

Feel free to contact:gidis@liveperson.com
052 3914066

Thank You To Our Sponsors

Platinum

CloudBees

JFrog

Gold

BLACKDUCK

ravello systems

BlazeMeter™ THE LOAD TESTING CLOUD

PERFORCE

Silver

redhat