

Edinburgh JAM

November 2017

Corporate Sponsor

HAYS
Recruiting experts
in Digital Technology



Welcome



Introductions



Organizers

- John Pikoulas

- jpikoulas@gmail.com



- Lee Varga

- lee.varga@hays.com
 - @LSVarga & @HaysDigitalEH
 - Picture
-



Topics

A brief Introduction to Jenkins
& Pipelines(Scripting &
Declarative)

- A Brief Introduction to Jenkins
 - Intro and Jenkins Conf 2017 news
 - Jenkins Pipelines
 - Early day orchestration
 - Scripting pipelines
 - Declarative pipelines
 - Jenkins Disaster Recovery ready environment topology
 - What capabilities Jenkins add to your organization
 - Next session teaser
-

Intro and Jenkins Conf 2017 news

- Blue ocean pipeline, adding Infrastructure as Code.
- Resume pipeline after a Jenkins restart
- Pipeline reusability in an organization
- All Jenkins technologies and plugins can be accessed through the pipeline
- Automate entire organizations not just individual projects
- Scaling Jenkins with Kubernetes



Pipelines

- In the beginning, there were no pipelines
- Job orchestration achieved by hand crafting 'Build other projects' and 'Post build actions' in each Jenkins job.

```
build( "job1" )  
build( "job2" )  
build( "job3" )
```
- Build Flow Plugin. Manage job orchestration plugin.
- DSL plugin. The DSL allows the definition of a job, and then offers a useful set of functions to configure common Jenkins items. The script is groovy code.
- DSL pipelines use to create scripting pipelines, more flexible but can get complicated to maintain.

Pipelines

- Do Pipelines replace job-dsl?
- Declarative is more restrictive through syntax
- Scripted limited only by Groovy syntax limitations
- Durability, to survive Jenkins restarts
- Pausing for user input
- Parallelism built in
- Pipeline snippet generator to help you build pipelines



Jenkins Disaster Recovery

- Jenkins Disaster Recovery - SCM Sync Plugin
- Back Up the latest Jenkins artifact version to Git or filesystem
- In conjunction with Docker, Jenkins complete recovery in less than 10 minutes.

SCM Sync configuration

SCM

None

Subversion

Git

Repository URL

Never bother me with commit messages

Display SCM Sync Status

Commit message pattern

Manual synchronization includes

Reload config from SCM [Reload](#)

Capabilities added to your organization

- Ability to work across silos
 - Dev teams owners of the entire tech stack, including processes and prod support
- Increased alignment with the business
 - Deploy small and often - quick requirements to implementation turnaround
 - Zero downtime deployments
- Easier to implement Governance framework across organization

Jenkins Pipeline Demo

```
// A Declarative Pipeline is defined within a 'pipeline' block.
pipeline {

    // agent defines where the pipeline will run
    // if you have a slave defined by a label you can use that - e.g. java8 or postgres or whatever
    agent {

        // This also could have been 'agent any' - that has the same meaning.
        label ""

        // Other possible built-in agent types are 'agent none', for not running the
        // top-level on any agent (which results in you needing to specify agents on
        // each stage and do explicit checkouts of scm in those stages), 'docker',
        // and 'dockerfile'.
    }

    // The tools directive allows you to automatically install tools configured in
    // Jenkins - note that it doesn't work inside Docker containers currently.
    // tools {
    //     jdk "jdk8"
    //     maven "mvn3.3.8"
    // }

    environment {

        // Environment variable identifiers need to be both valid bash variable
        // identifiers and valid Groovy variable identifiers. If you use an invalid
        // identifier, you'll get an error at validation time.
        // Right now, you can't do more complicated Groovy expressions or nesting of

        // other env vars in environment variable values, but that will be possible
        // when https://issues.jenkins-ci.org/browse/JENKINS-41748 is merged and
        // released.
        FOO = "BAR"
    }

    stages {

        // At least one stage is required.
        stage("first stage") {

            // Every stage must have a steps block containing at least one step.
            steps {

                // You can use steps that take another block of steps as an argument.
                sh "sleep 10"

                echo "We're not doing anything particularly special here."

                echo "Hello World etc."

                echo "FOO is ${FOO}"

                sh "sleep 5"
            }

            // Post can be used both on individual stages and for the entire build.
            post {
                success {
                    echo "Only when we haven't failed running the first stage"
                }
            }
        }
    }
}
```



Jenkins Pipeline Demo

Build History [trend](#)

find X

- #2 23-Nov-2017 21:21
- #1 23-Nov-2017 20:40

[RSS for all](#) [RSS for failures](#)

Pipeline Jenkins_pipeline

[add description](#)
[Disable Project](#)

[Recent Changes](#)

Stage View

Average stage times:
(Average full run time: ~35s)

	Declarative: Checkout SCM	first stage	second stage	third stage	testing	Declarative: Post Actions
#2	2s	15s	11s	129ms	167ms	2s
Nov 23 21:21 1 commits	3s	15s	11s	147ms	154ms	2s
#1	2s	15s	11s	111ms	180ms	2s
Nov 23 20:40 No Changes						

Permalinks

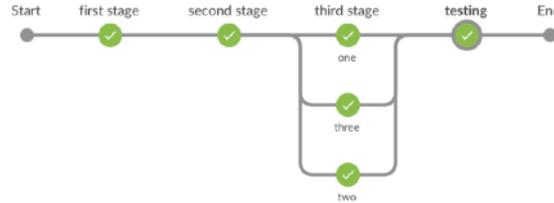
- [Last build \(#2\). 1 day 1 hr ago](#)
- [Last stable build \(#2\). 1 day 1 hr ago](#)
- [Last successful build \(#2\). 1 day 1 hr ago](#)
- [Last completed build \(#2\). 1 day 1 hr ago](#)

Jenkins Pipeline Demo

✓ Jenkins_pipeline 2

Pipeline Changes Tests Artifacts ↻ ⚙️ 📄 Login ✕

Branch: — 37s Changes by noreply
Commit: — 4 days ago Started by user john



Steps testing

✓ > Testing the chrome browser — Print Message	<1s
✓ > Testing the firefox browser — Print Message	<1s
✓ > Testing the vivaldi browser — Print Message	<1s
✓ > Testing the safari browser — Print Message	<1s
✓ > Testing the tor browser — Print Message	<1s
✓ > Build 2 of the Jenkins_pipeline job has passed. — Mail	1s
✓ > echo Yay for this, sending an email... — Shell Script	<1s
✓ > Recursively delete the current directory from the workspace	<1s



Future talk teaser



—

Recent Events

 Jenkins JAM meetups. Providing best Jenkins practices, news and updates.

<https://www.meetup.com/Edinburgh-Jenkins-Area-Meetup>

meetup

 DEV-OPS workshops. A series of in-depth workshops targeting different dev-ops technologies and practices.

 https://www.youtube.com/channel/UCSNpyys87aAXG5_ZwGPIq3Q

 <http://www.automateditsolutions.com>

 YouTube^{GB}

 <http://www.automateditsolutions.com/register/>