From Freestyle jobs to Pipeline, with JobDSL

Nicolaj Græsholt - @figaw on Twitter
DevOps Consultant at Eficode Praqma

About:nicolaj græsholt 😃

<DevOps Consultant>

❤️ Good **Documentation, Examples** and **Proof of Concepts**

DMs are **OPEN**, please be nice :) ⟶

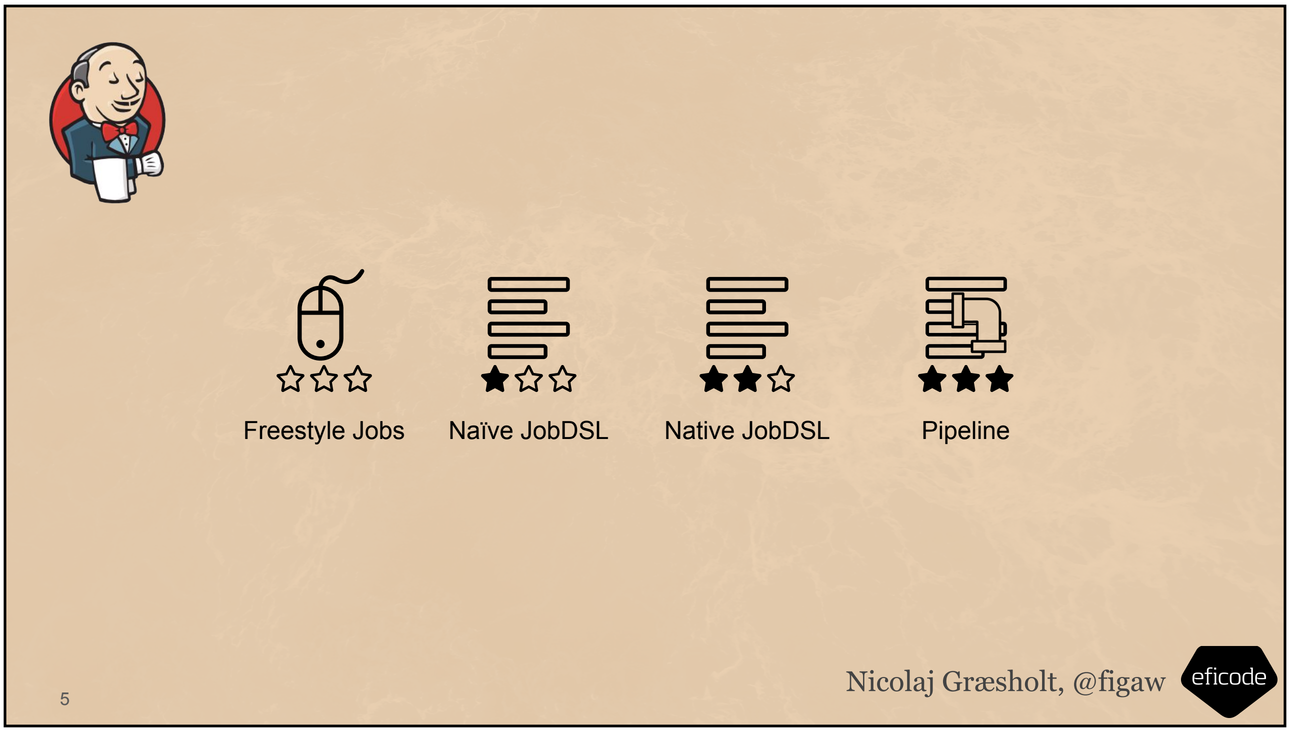
Source: https://www.artstation.com/artwork/1XmOe

DON'T PANIC

(Large, friendly letters)

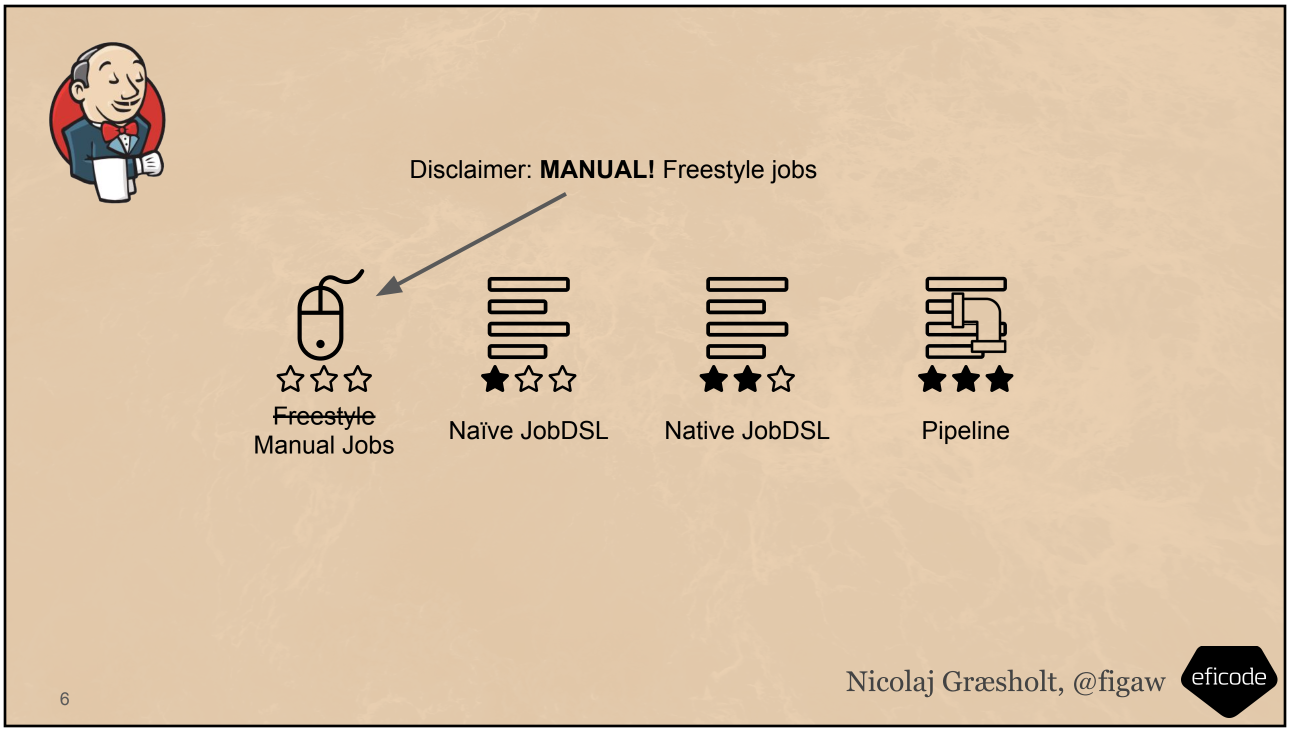
Everything You See Is What You Get

https://github.com/figaw/freestyle-to-pipeline-jenkins



☆☆☆	★☆☆	★★☆	★★★
Freestyle Jobs	Naïve JobDSL	Native JobDSL	Pipeline

Nicolaj Græsholt, @figaw eficode



Disclaimer: **MANUAL!** Freestyle jobs

☆☆☆	★☆☆	★★☆	★★★
~~Freestyle~~ Manual Jobs	Naïve JobDSL	Native JobDSL	Pipeline

Nicolaj Græsholt, @figaw eficode

Freestyle Jobs and the 5 Stages of Grief

“Kübler-Ross model”

😅 Denial - This is Great!

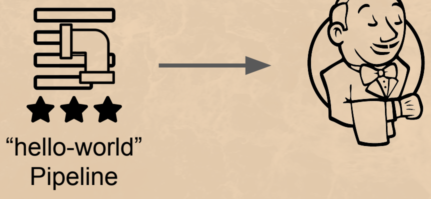
😡 Anger - Configuration Drift, Zero Reuse, Reconfiguration, Losing { Work, Time, Will to Continue.. }

🤔 Bargaining - Way of Working

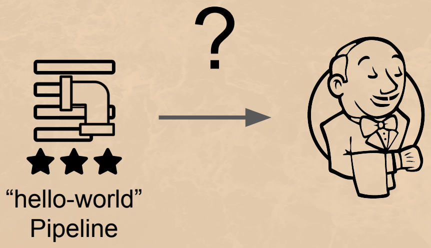
😭 Depression - This isn’t Working, colleagues and discipline (and that’s okay!)

😕 Acceptance - Manual Freestyle jobs aren’t *Great*

Goal 😃

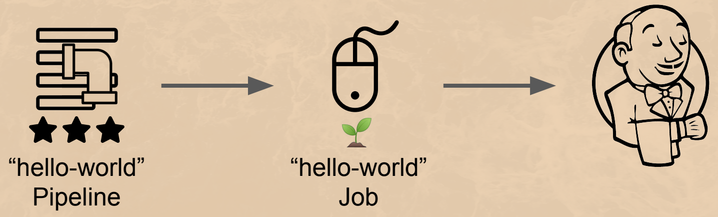


Goal 🤔

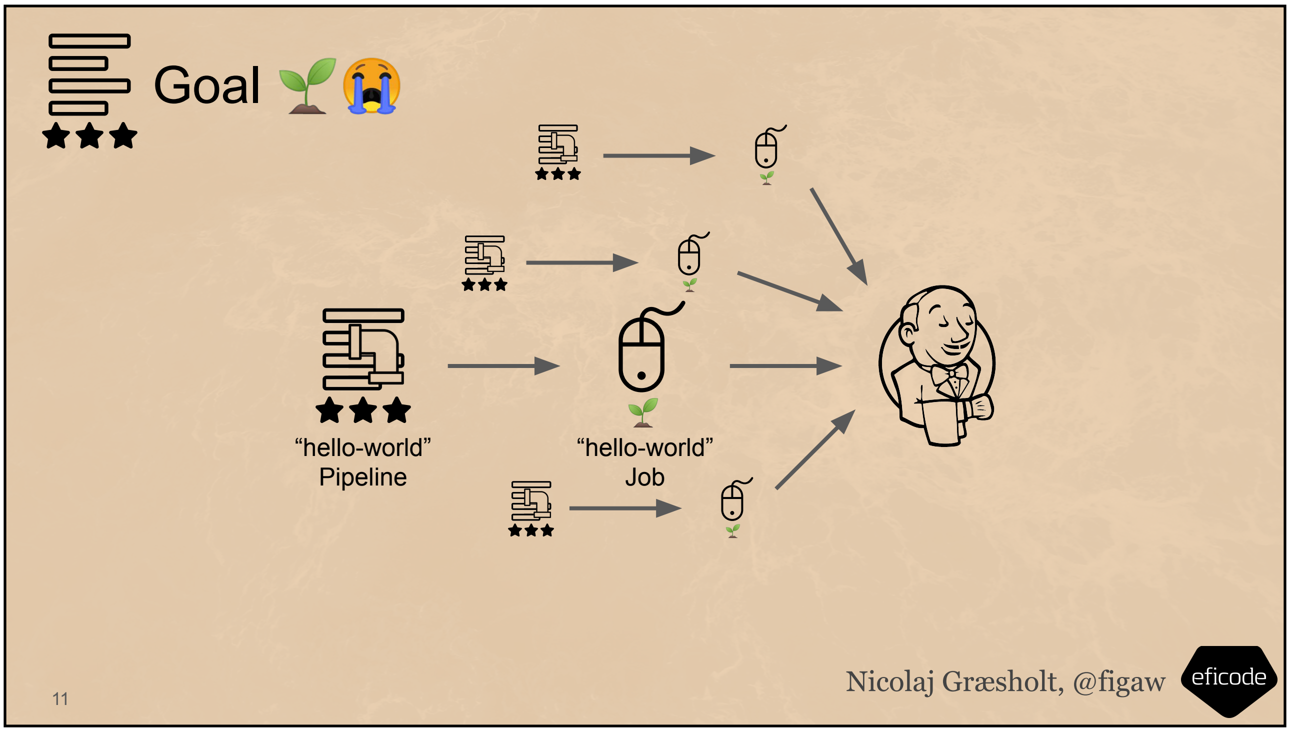


Nicolaj Græsholt, @figaw

Goal 🤔🌱



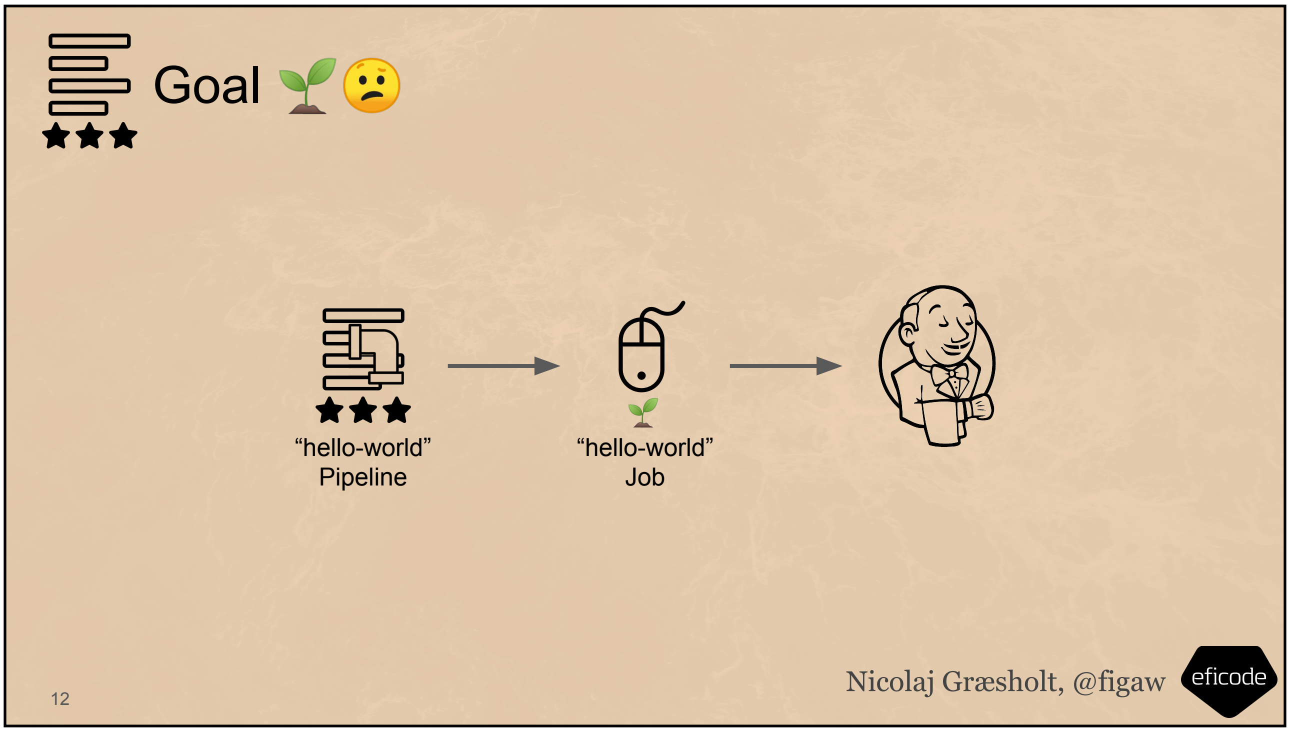
Nicolaj Græsholt, @figaw



Goal 🌱😭

Nicolaj Græsholt, @figaw

eficode

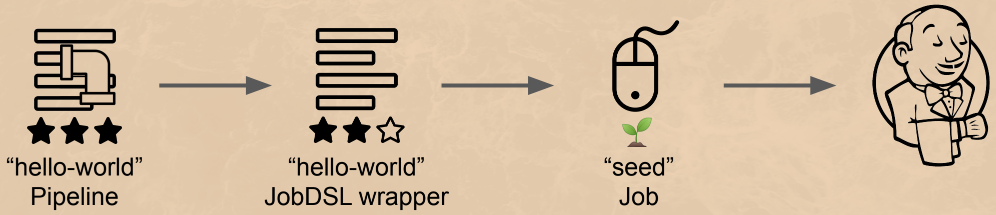


Goal 🌱😟

Nicolaj Græsholt, @figaw

eficode

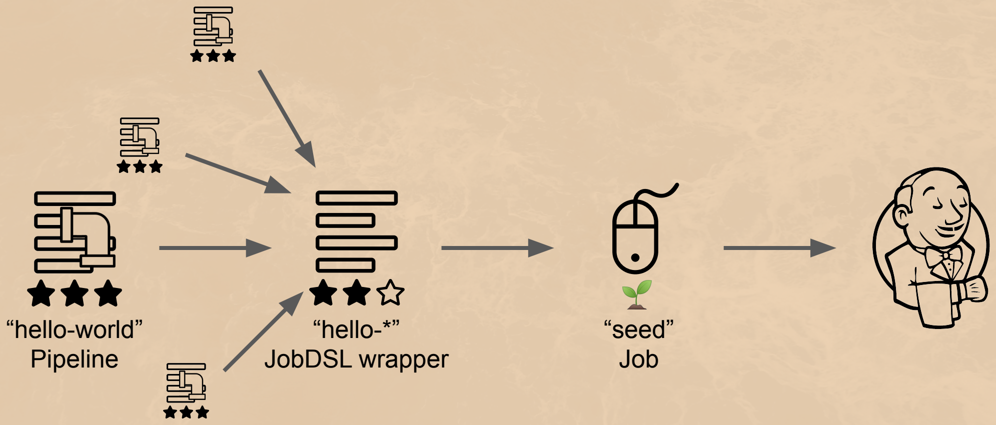
Goal 🌱🤔



Nicolaj Græsholt, @figaw

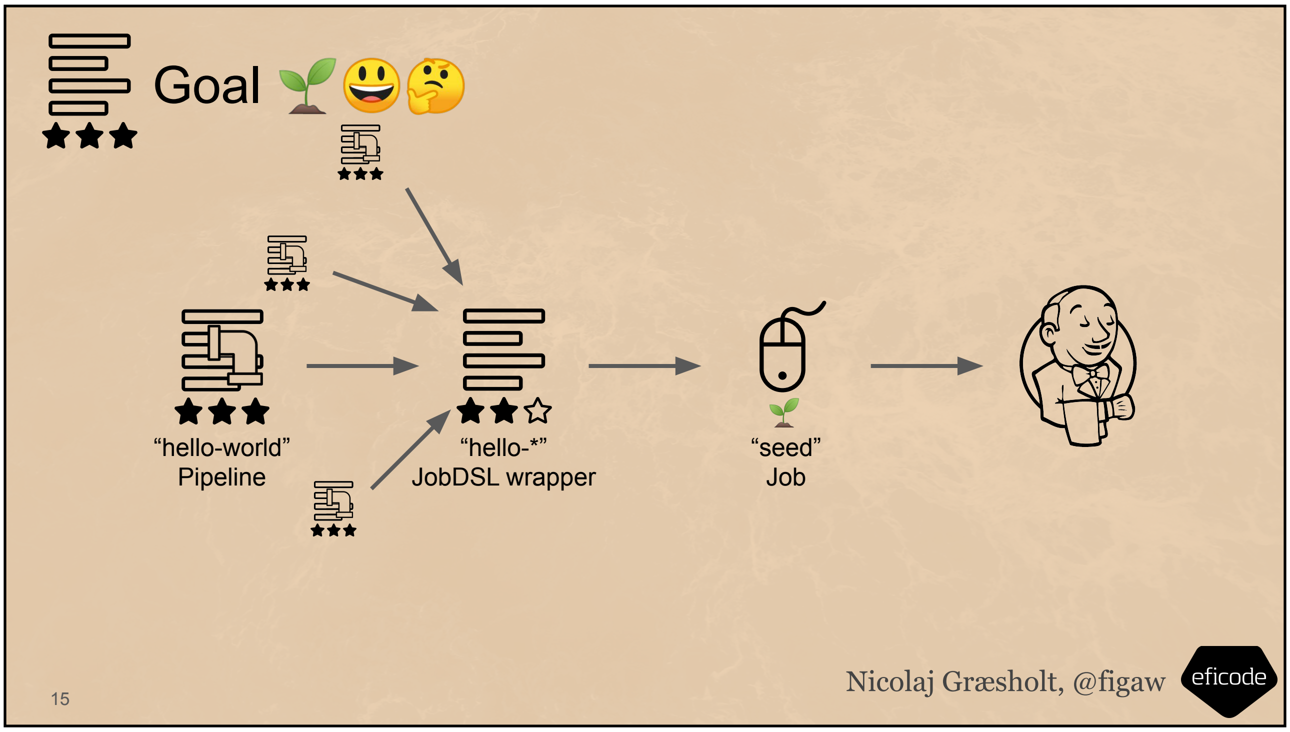
eficode

Goal 🌱😃

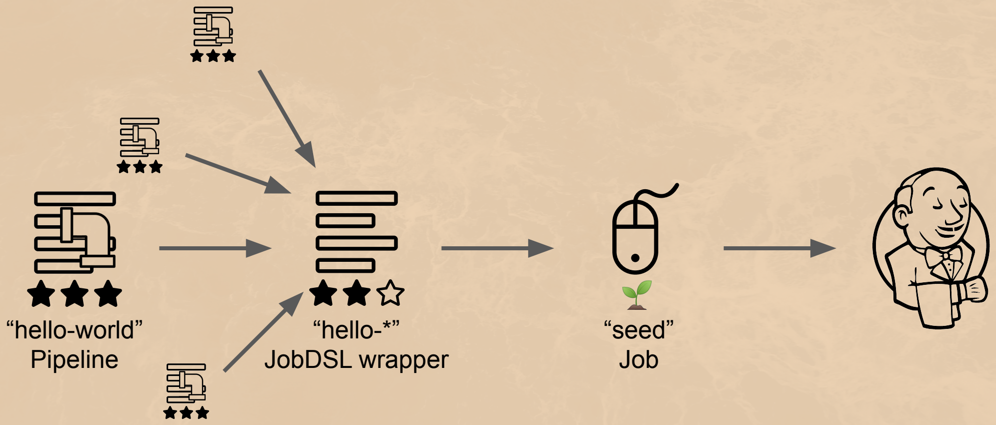


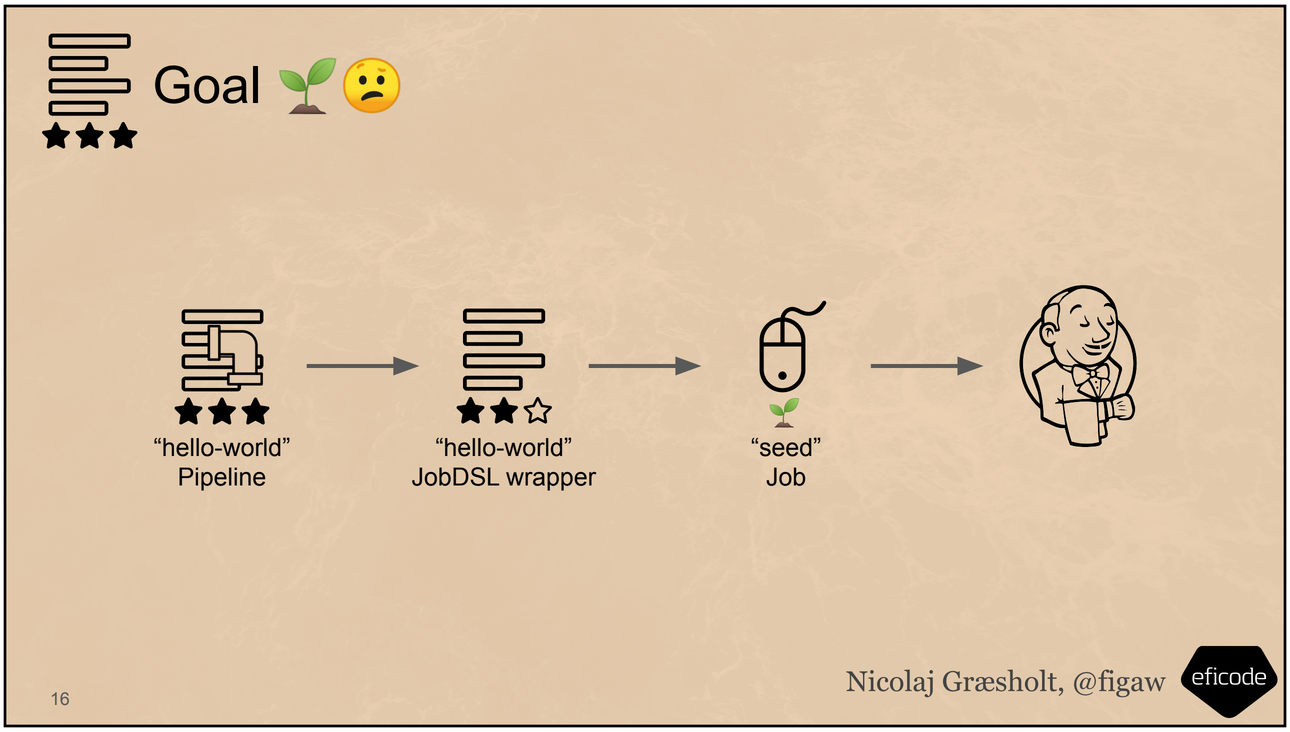
Nicolaj Græsholt, @figaw

eficode

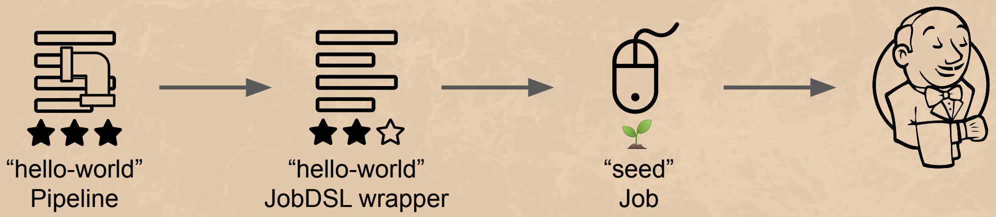


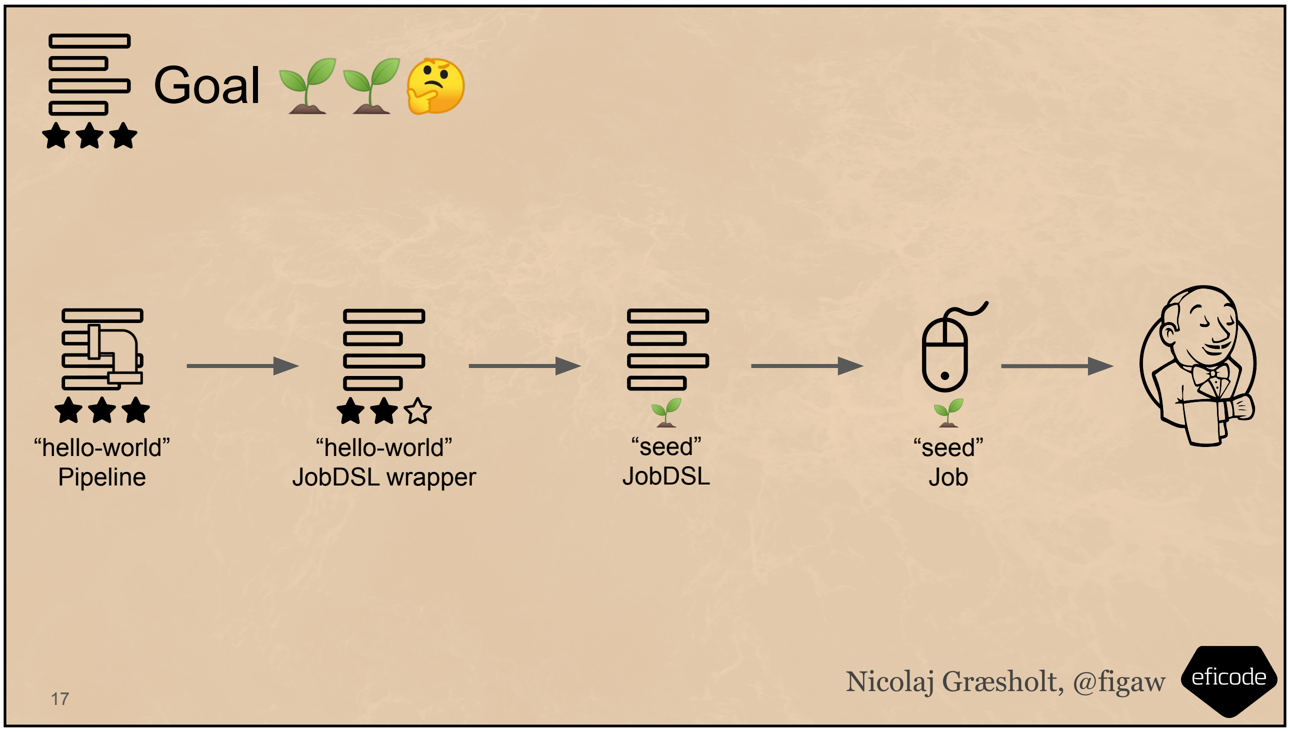
Goal 🌱😃🤔


"hello-world"
Pipeline
"hello-*"
JobDSL wrapper
"seed"
Job



Goal 🌱😟


"hello-world"
Pipeline
"hello-world"
JobDSL wrapper
"seed"
Job

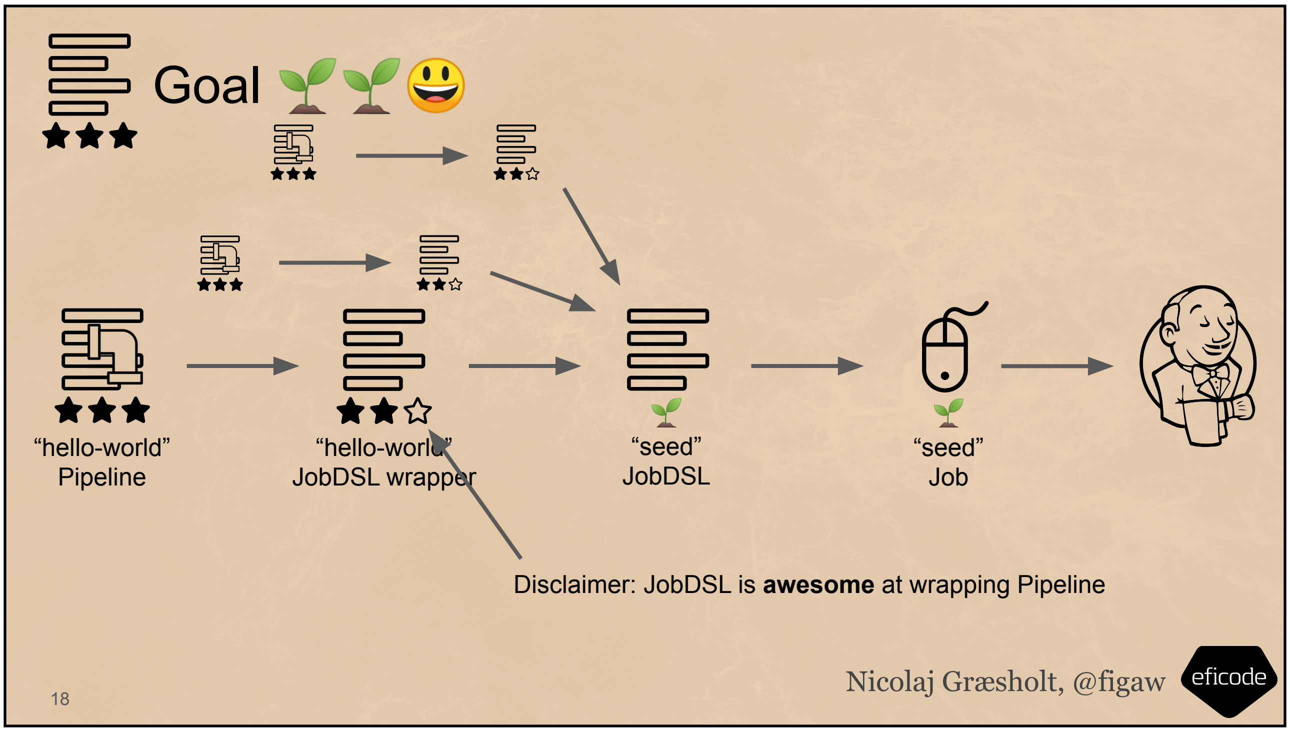


Goal 🌱🌱🤔

“hello-world” Pipeline → “hello-world” JobDSL wrapper → “seed” JobDSL → “seed” Job →

Nicolaj Græsholt, @figaw

eficode



Goal 🌱🌱😃

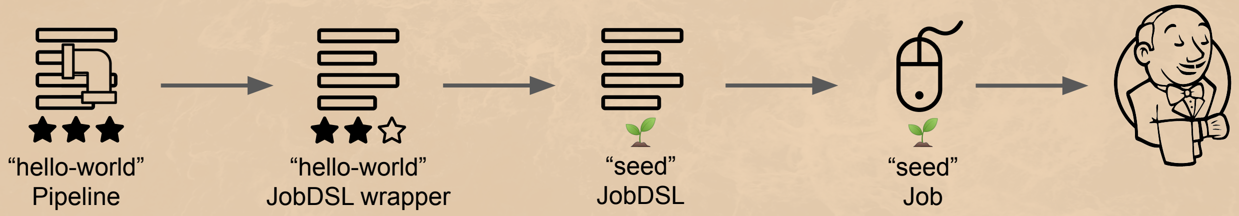
“hello-world” Pipeline → “hello-world” JobDSL wrapper → “seed” JobDSL → “seed” Job →

Disclaimer: JobDSL is **awesome** at wrapping Pipeline

Nicolaj Græsholt, @figaw

eficode

Goal 🌱🌱😃


“hello-world”
Pipeline
“hello-world”
JobDSL wrapper
“seed”
JobDSL
“seed”
Job

JobDSL

JobDSL

> **A Groovy DSL for Jenkins Jobs - Sweeeeet!**
>
> JobDSL Plugin, GitHub Repository

JobDSL crash-course 🤔

hello-world.groovy

```
job('hello-world') {
  steps {
    shell('echo Hello, World!')
  }
}
```

JobDSL crash-course 🤔

hello-world.groovy

```
job('hello-world') {
  steps {
    shell('echo Hello, World!')
  }
}
```

JobDSL crash-course 🤔

hello-world.groovy

```
job('hello-world') {
  steps {
    shell('echo Hello, World!')
  }
}
```

JobDSL crash-course 🤔

hello-world.groovy

```
job('hello-world') {
  steps {
    shell('echo Hello, World!')
  }
}
```

JobDSL crash-course 🤔

hello-world.groovy

```
job('hello-world') {
  steps {
    shell('echo Hello, World!')
  }
}
```

Demo Time: “hello-world”

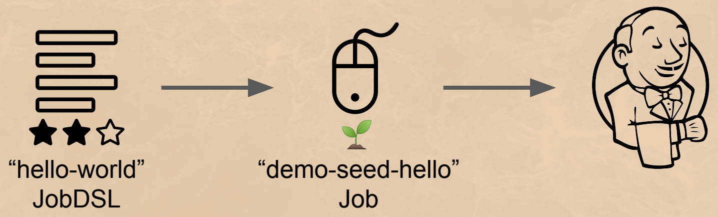
Will Show: JobDSL from slide works

hello-world.groovy

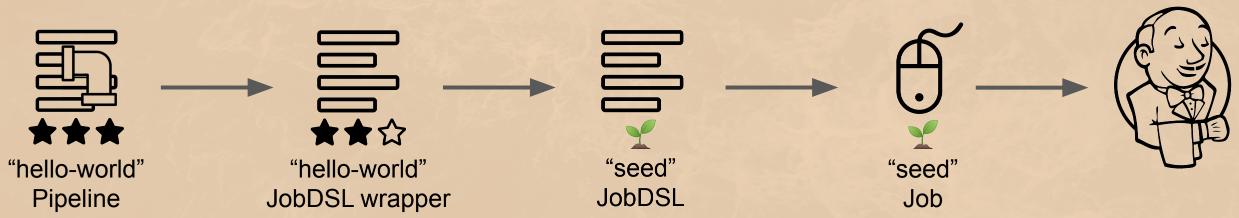
```
job('hello-world') {
  steps {
    shell('echo Hello, World!')
  }
}
```

Source: https://www.artstation.com/artwork/1XmOe

Basic JobDSL 😃



Goal 🌱🌱😃



JobDSL - Inline 🌱🤔

hello-world-seed-inline.groovy

```
job('seed-inline-job') {
  steps {
    dsl {
      text("""
        job('hello-world-seedet') {
          steps {
            shell('echo Hello, World!')
          }
        }""")
    }
  }
}
```

Demo Time: “seed-hello-world-inline” 🌱

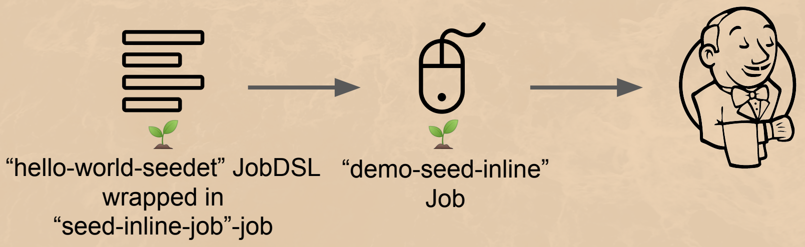
Will Show: JobDSL from slide works

seed-hello-world-inline.groovy

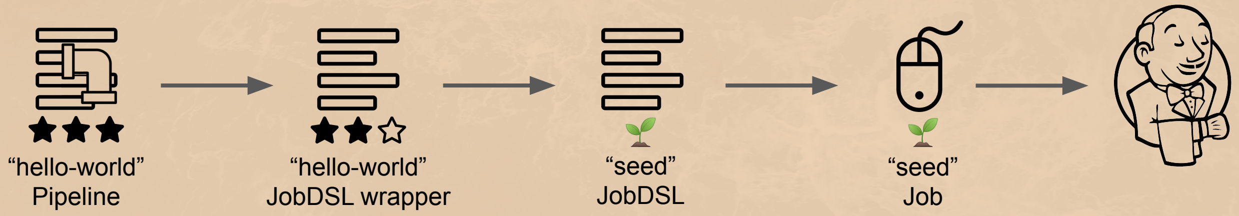
```
job('seed-inline-job') {
  steps {
    dsl {
      text("""
        job('hello-world-seedet') {
          steps {
            shell('echo Hello, World!')
          }
        }""")
    }
  }
}
```

Source: https://www.artstation.com/artwork/1XmOe

JobDSL - Inline🌱 😃


“hello-world-seedet” JobDSL wrapped in “seed-inline-job”-job
“demo-seed-inline” Job

Goal 🌱🌱😃



"Magic" and Caveats 🌱

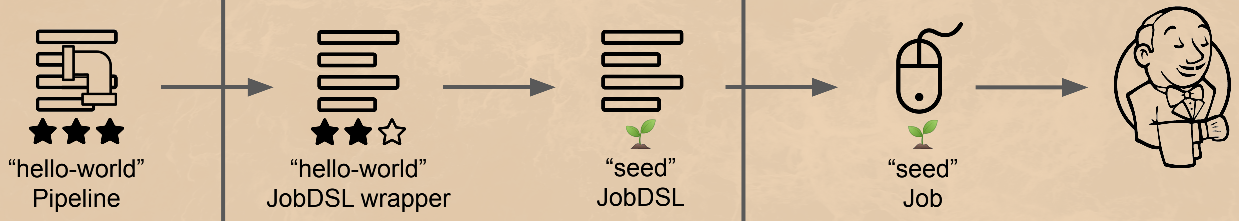
Demo:

- JobDSL script-security is off
- Formatting with `stripIndent`

```
job('seed-inline-job') {
  steps {
    dsl {
      text("""
        job('hello-world-seedet') {
          steps {
            shell('echo Hello, World!')
          }
        }""".stripIndent())
    }
  }
}
```

Source: https://www.artstation.com/artwork/1XmOe

Goal 🌱🌱😃



35

Nicolaj Græsholt, @figaw eficode

JobDSL - External 🌱😮

seed-external.groovy

```
job('seed-external-job') {
  // checkout git-repo
  steps {
    dsl {
      external('hello_external.groovy')
    }
  }
}
```

files

```
seed-external.groovy
hello_external.groovy
```

36

Nicolaj Græsholt, @figaw eficode

Demo Time: 🌱

Will (Not) Show: “seed-hello-world-external”

seed-external.groovy

```
job('seed-external-job') {
  // checkout git-repo
  steps {
    dsl {
      external('hello_external.groovy')
    }
  }
}
```

files

```
seed-external.groovy
hello_external.groovy
```

Source: https://www.artstation.com/artwork/1XmOe

JobDSL - External, Dynamic 🌱 😃

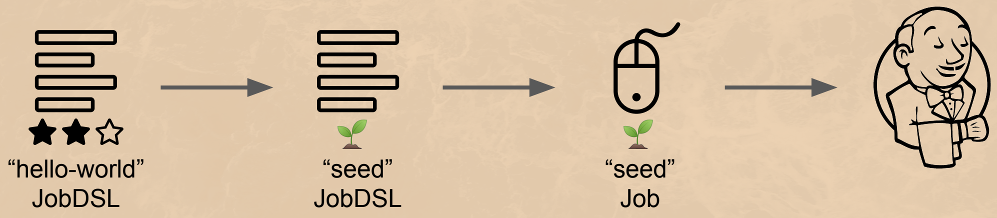
any-seed.groovy

```
job('seed-any-job') {
  // checkout git-repo
  steps {
    dsl {
      external('**/*.groovy')
    }
  }
}
```

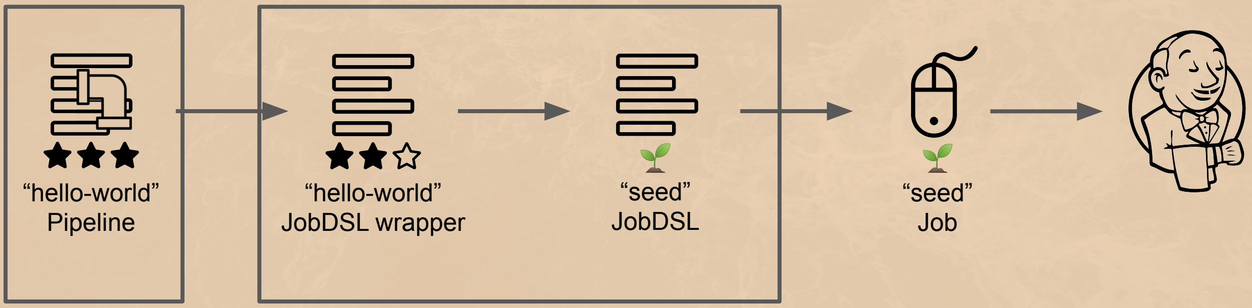
files

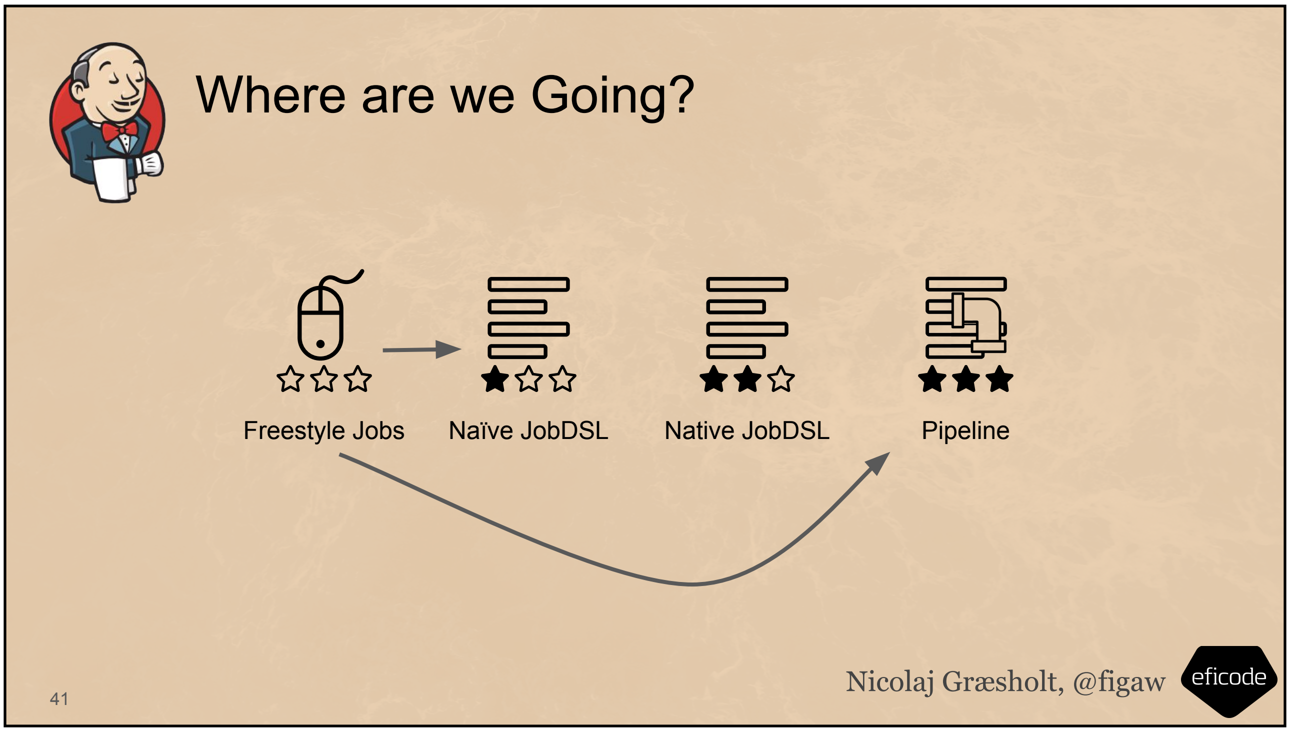
```
any-seed.groovy
hello_world.groovy
hello_external.groovy
...
hello/world_folder.groovy
```

JobDSL - External 🌱🌱😃



Goal 🌱🌱😃

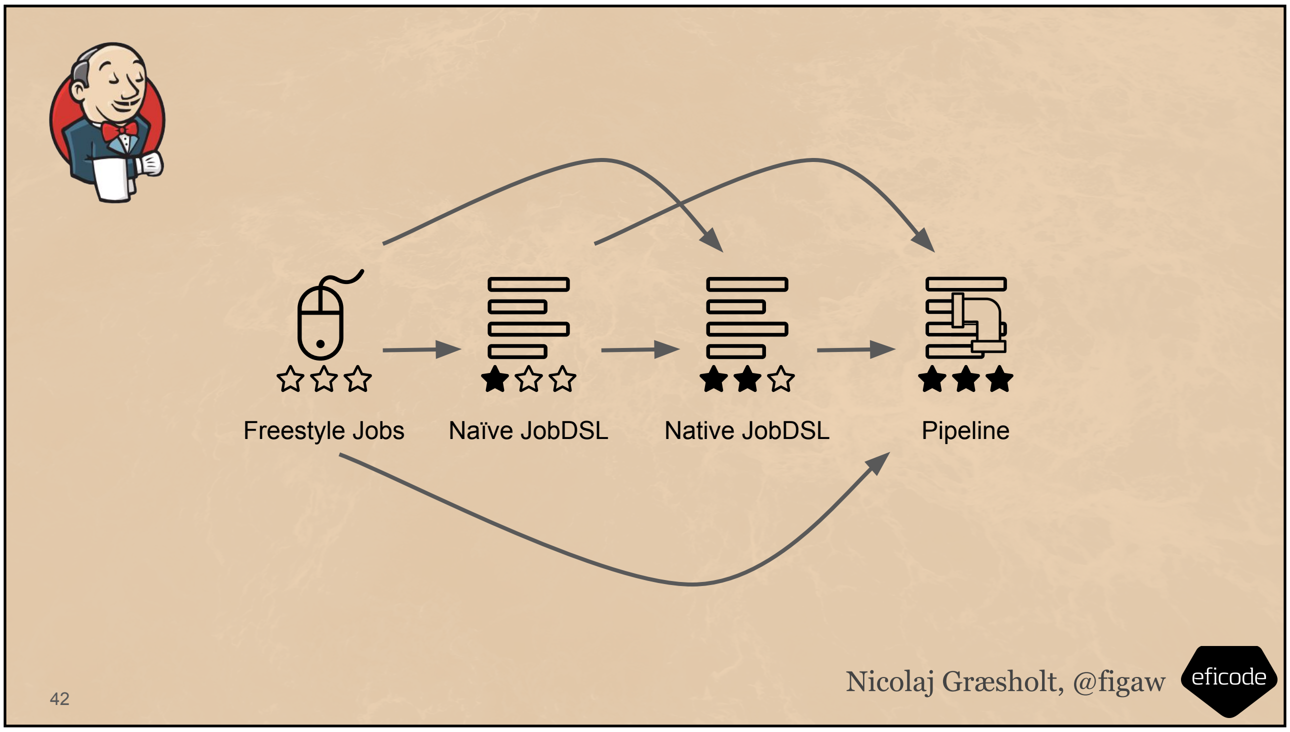




Where are we Going?

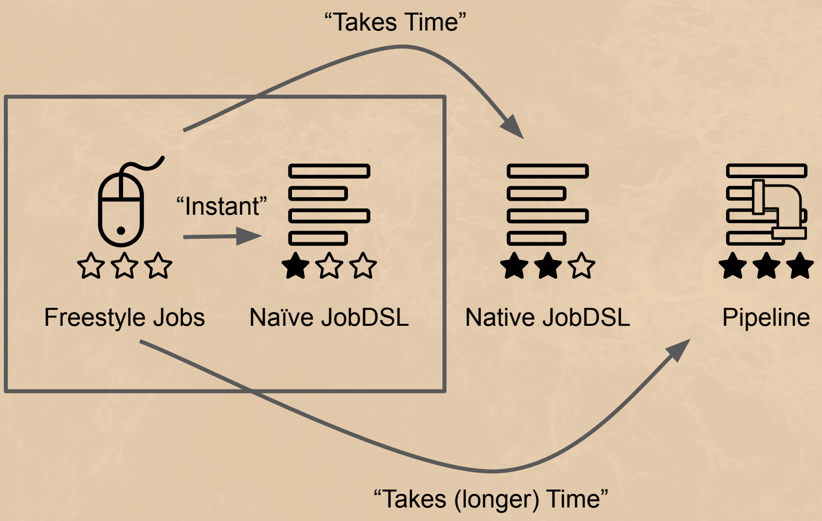
Nicolaj Græsholt, @figaw

eficode



Nicolaj Græsholt, @figaw

eficode

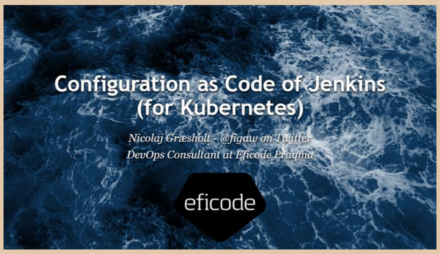


Nicolaj Græsholt, @figaw

eficode

Demo Time: "jobdsl" 😮

Will Show:
- A Jenkins job is already "as code" (XML)
- JobDSL to wrap XML
- JobDSL bootstrapping, for sanity check


Configuration as Code of Jenkins (for Kubernetes)

Nicolaj Græsholt - @figaw on Twitter
DevOps Consultant at Eficode Praqma

eficode

Source: https://www.artstation.com/artwork/1XmOe

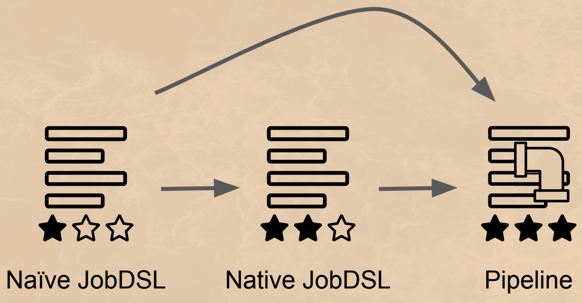
Nicolaj Græsholt, @figaw

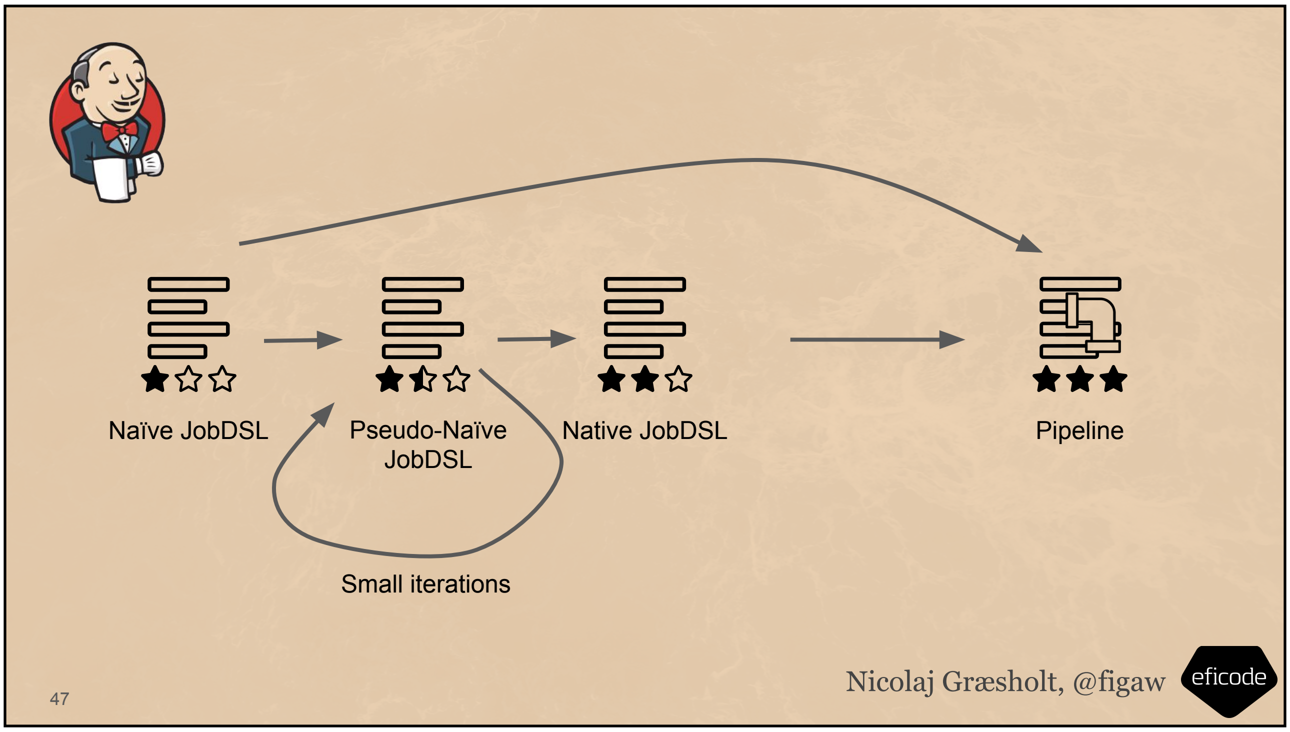
eficode

Is this cheating?

No:

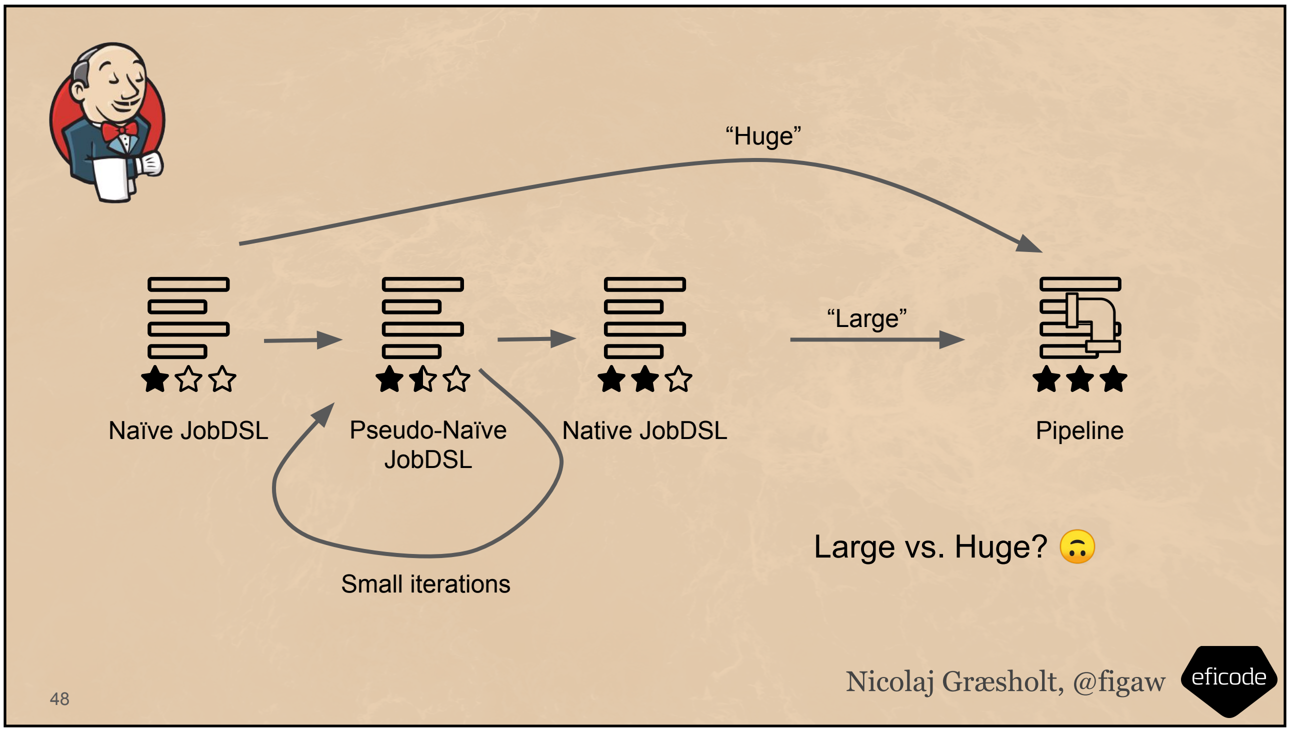
- Change job through the UI
- Persist changes with the generated config.xml, through Git
- Job configuration is under Version Control
- Easy reconfiguration with “dynamic” seed.groovy





Nicolaj Græsholt, @figaw

eficode

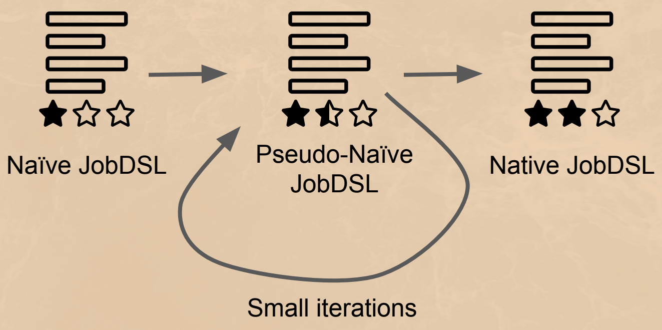


Large vs. Huge? 🙃

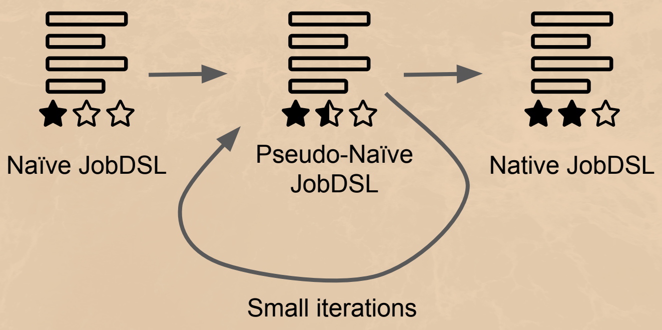
Nicolaj Græsholt, @figaw

eficode

CONFIGURE-BLOCK - **POWER**
UNFOLDING XML to JobDSL! - **ELBOW GREASE**



Crazy-tedious 😫
Very time consuming 😧
Super safe 😃



Demo Time: “jobdsl-iterative” 🌱

Will Show:

- Converting 6 lines of XML to
  4 lines of JobDSL with only 20 slides
- How to put an audience to sleep

Source: https://www.artstation.com/artwork/1XmOe

★☆☆ Naïve JobDSL

★★☆ Native JobDSL

“hello-world” in Naïve JobDSL

```
def jobconfig = """
<project>
  <description>This is my Job</description>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

(Continued)

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Identify

```
def jobconfig = """
<project>
  <description>This is my Job</description>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

(Continued)

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Extract

```
def jobdesc = """
  <description>This is my Job</description>
"""

def jobconfig = """
<project>
  <description>This is my Job</description>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

```
def jobdescnode =
  new XmlParser().parseText(jobdesc)

def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  configure { node ->
    node << jobdescnode
  }
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Nicolaj Græsholt, @figaw

eficode

Refactor

```
def jobdesc = """
  <description>This is my Job</description>
"""

def jobconfig = """
<project>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

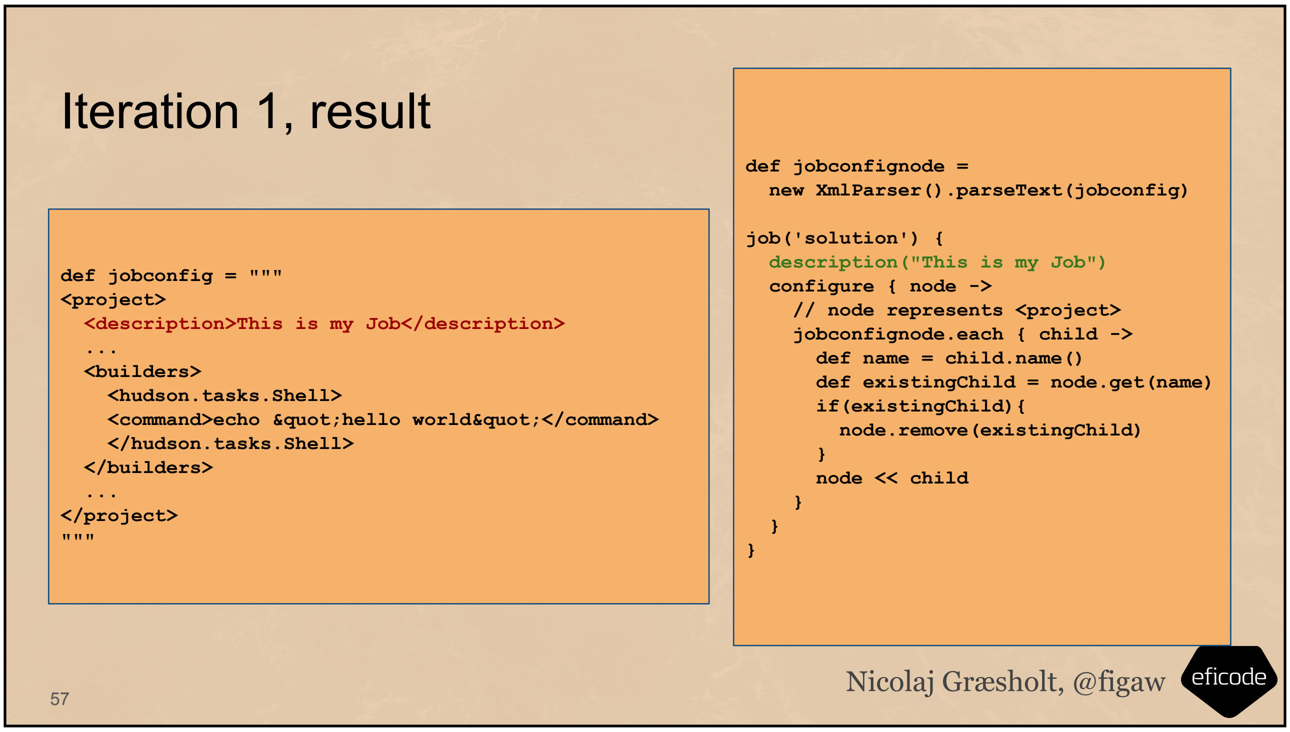
```
def jobdescnode =
  new XmlParser().parseText(jobdesc)

def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  configure { node ->
    node << jobdescnode
  }
  description("This is my Job")
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Nicolaj Græsholt, @figaw

eficode



Iteration 1, result

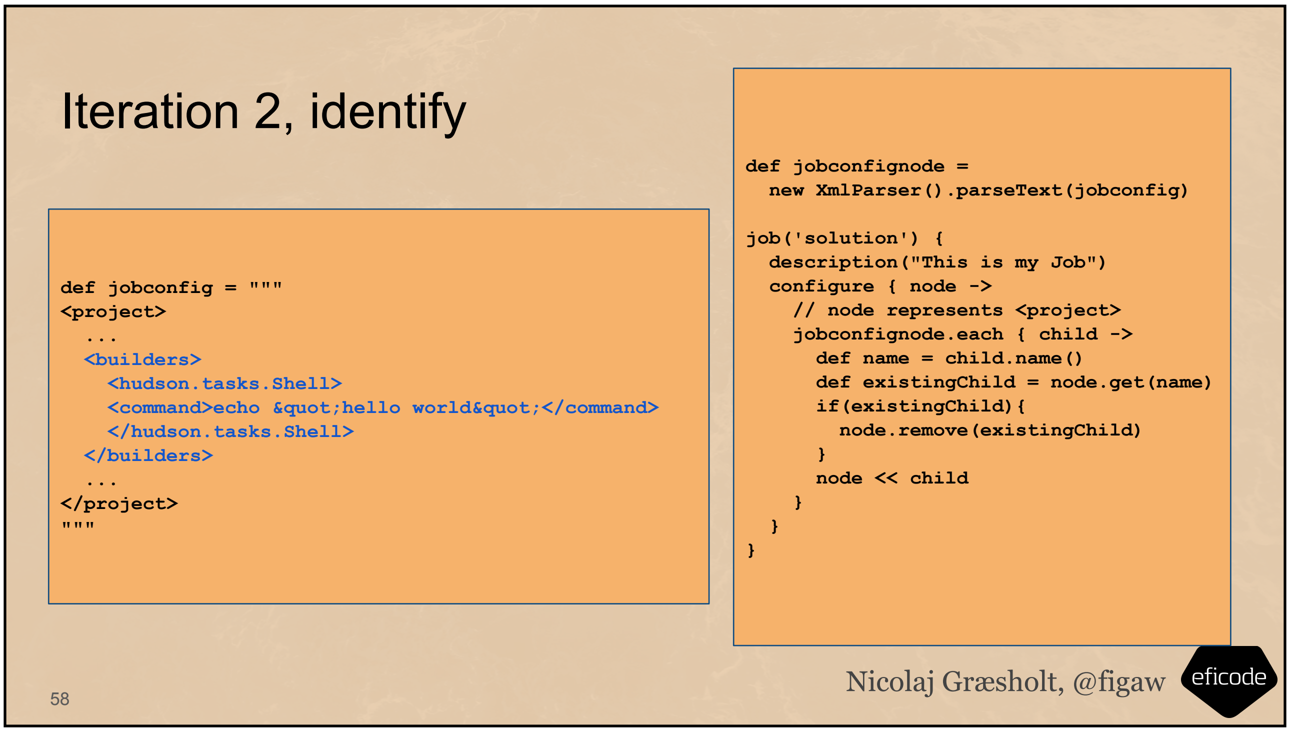
```
def jobconfig = """
<project>
  <description>This is my Job</description>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Nicolaj Græsholt, @figaw

eficode



Iteration 2, identify

```
def jobconfig = """
<project>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Nicolaj Græsholt, @figaw

eficode

Identify, Take a Deep Breath

```
def jobconfig = """
<project>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Nicolaj Græsholt, @figaw eficode

Omit details

```
def jobconfig = """
<project>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Nicolaj Græsholt, @figaw eficode

Omit details, move code

```
def jobconfig = """
<project>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

eficode

Okay, ready. Identify!

```
def jobconfig = """
<project>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

eficode

Extract

```
def jobsteps = """
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
"""

def jobconfig = """
<project>
  ...
  <builders>...</builders>
  ...
</project>
"""
```

```
def jobstepsnode =
  new XmlParser().parseText(jobsteps)

def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    node << jobstepsnode


  }
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

eficode

Unfold 1

```
def jobsteps = """
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
"""

def jobconfig = """
<project>
  ...
</project>
"""
```

```
def jobstepsnode =
  new XmlParser().parseText(jobsteps)

def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    node / builders << jobstepsnode


  }
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

eficode

Unfold 2

```
def jobsteps = """
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
"""

def jobconfig = """
<project>
  ...
</project>
"""
```

```
def jobstepsnode =
  new XmlParser().parseText(jobsteps)

def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    node / builders
      / 'hudson.tasks.Shell'
      << jobstepsnode
  }
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

eficode

Unfold 3

```
def jobsteps = """
    <command>echo &quot;hello world&quot;</command>
"""

def jobconfig = """
<project>
  ...
</project>
"""
```

```
def jobstepsnode =
  new XmlParser().parseText(jobsteps)

def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    node / builders
      / 'hudson.tasks.Shell' {
        command 'echo "hello world"'
      }
      << jobstepsnode
  }
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

eficode

Unfold 3 (clean)

```
def jobconfig = """
<project>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    node / builders
      / 'hudson.tasks.Shell' {
        command 'echo "hello world"'
      }
  }
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

eficode

Refactor

```
def jobconfig = """
<project>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  configure { node ->
    node / builders
      / 'hudson.tasks.Shell' {
        command 'echo "hello world"'
      }
  }
  steps {
    shell 'echo "hello world"'
  }
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

eficode

Iteration 2, result

```
def jobconfig = """
<project>
  ...
  <builders>
    <hudson.tasks.Shell>
    <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  steps {
    shell 'echo "hello world"'
  }
  ... // omitted configure-block
}
```

Nicolaj Græsholt, @figaw

Identify..

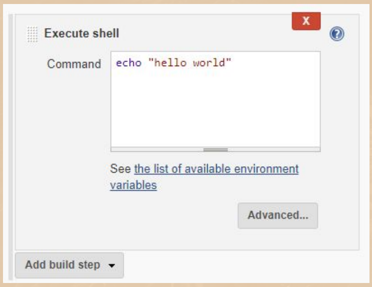
```
def jobconfig = """
<project>
  ...
</project>
"""
```

```
def jobconfignode =
  new XmlParser().parseText(jobconfig)

job('solution') {
  description("This is my Job")
  steps {
    shell 'echo "hello world"'
  }
  configure { node ->
    // node represents <project>
    jobconfignode.each { child ->
      def name = child.name()
      def existingChild = node.get(name)
      if(existingChild){
        node.remove(existingChild)
      }
      node << child
    }
  }
}
```

Nicolaj Græsholt, @figaw

Why Iterative?


Execute shell

Command
```
echo "hello world"
```
See the list of available environment variables

hello-world.groovy

```
job('hello-world') {
  steps {
    shell('echo Hello, World!')
  }
}
```

```
def jobconfig = """
<project>
  <actions/>
  <description></description>
  <keepDependencies>false</keepDependencies>
  <properties/>
  <scm class="hudson.scm.NullSCM"/>
  <canRoam>true</canRoam>
  <disabled>false</disabled>
  <blockBuildWhenDownstreamBuilding>false</blockBuildWhenDownstreamBuilding>
  <blockBuildWhenUpstreamBuilding>false</blockBuildWhenUpstreamBuilding>
  <triggers/>
  <concurrentBuild>false</concurrentBuild>
  <builders>
    <hudson.tasks.Shell>
      <command>echo &quot;hello world&quot;</command>
    </hudson.tasks.Shell>
  </builders>
  <publishers/>
  <buildWrappers/>
</project>
"""

def jobconfignode = new XmlParser().parseText(jobconfig)
job('hello-world') {
    configure { node ->
        // node represents <project>
        jobconfignode.each { child ->
          def name = child.name()
          def existingChild = node.get(name)
          if(existingChild){
            node.remove(existingChild)
          }
          node << child
        }
    }
}
```

JobDSL specification?

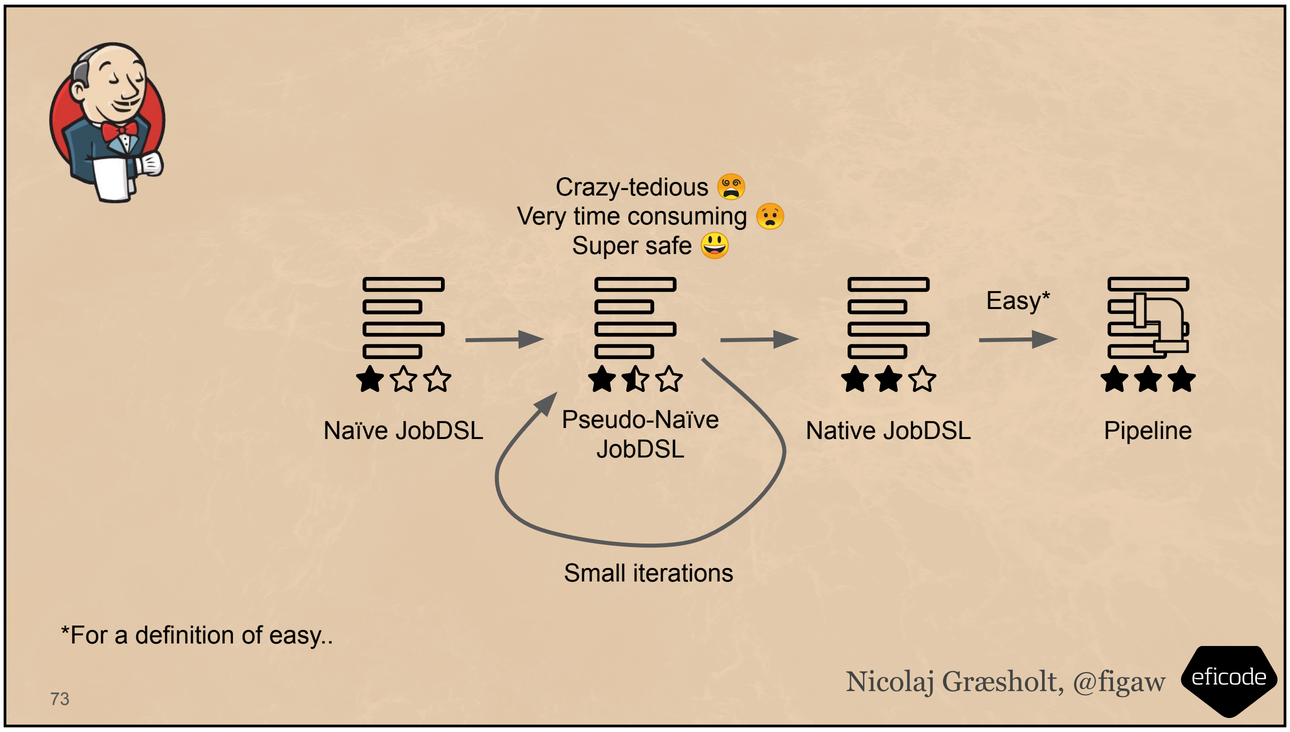
What can JobDSL do?

https://jenkinsci.github.io/job-dsl-plugin/#

(What can JobDSL do on MY Jenkins?)

https://github.com/jenkinsci/job-dsl-plugin#documentation

https://**your.jenkins.installation**/plugin/job-dsl/api-viewer/index.html

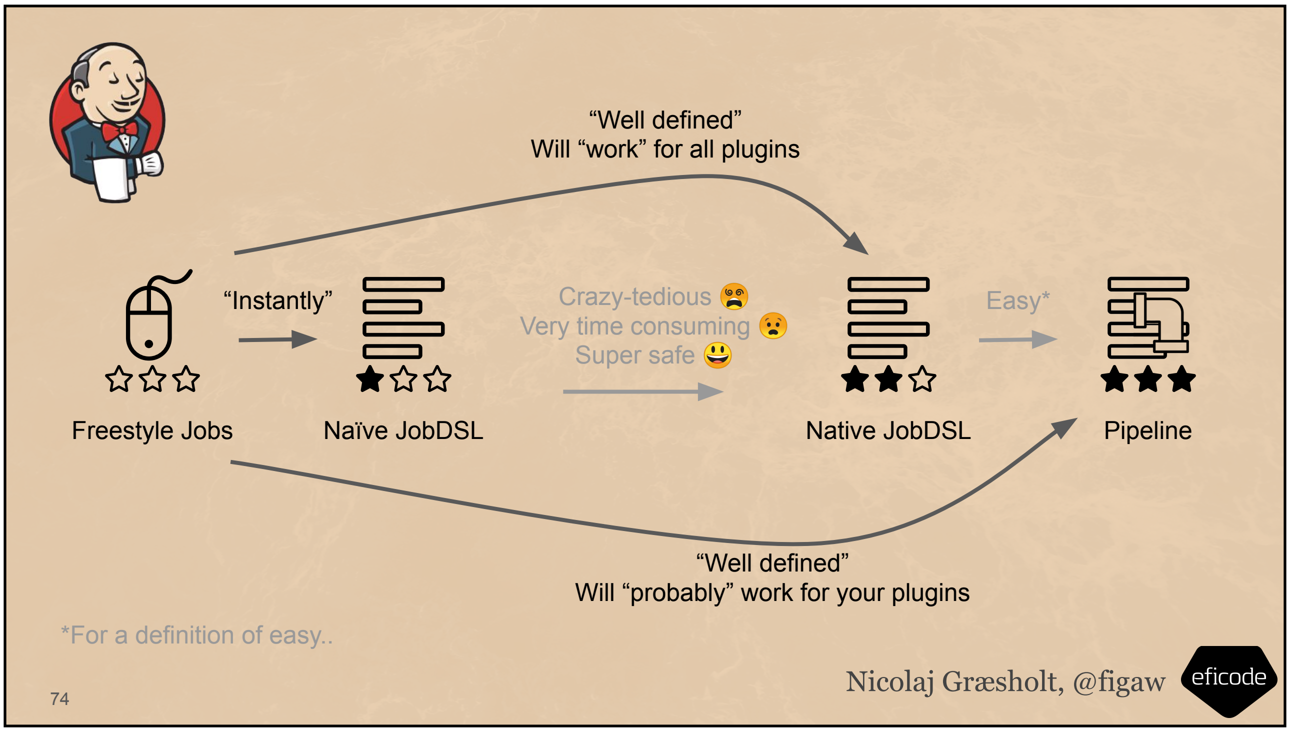


Crazy-tedious 😫
Very time consuming 😧
Super safe 😃

Naïve JobDSL ★☆☆ → Pseudo-Naïve JobDSL ★⯪☆ → Native JobDSL ★★☆ → Easy* → Pipeline ★★★

Small iterations

*For a definition of easy..



"Well defined"
Will "work" for all plugins

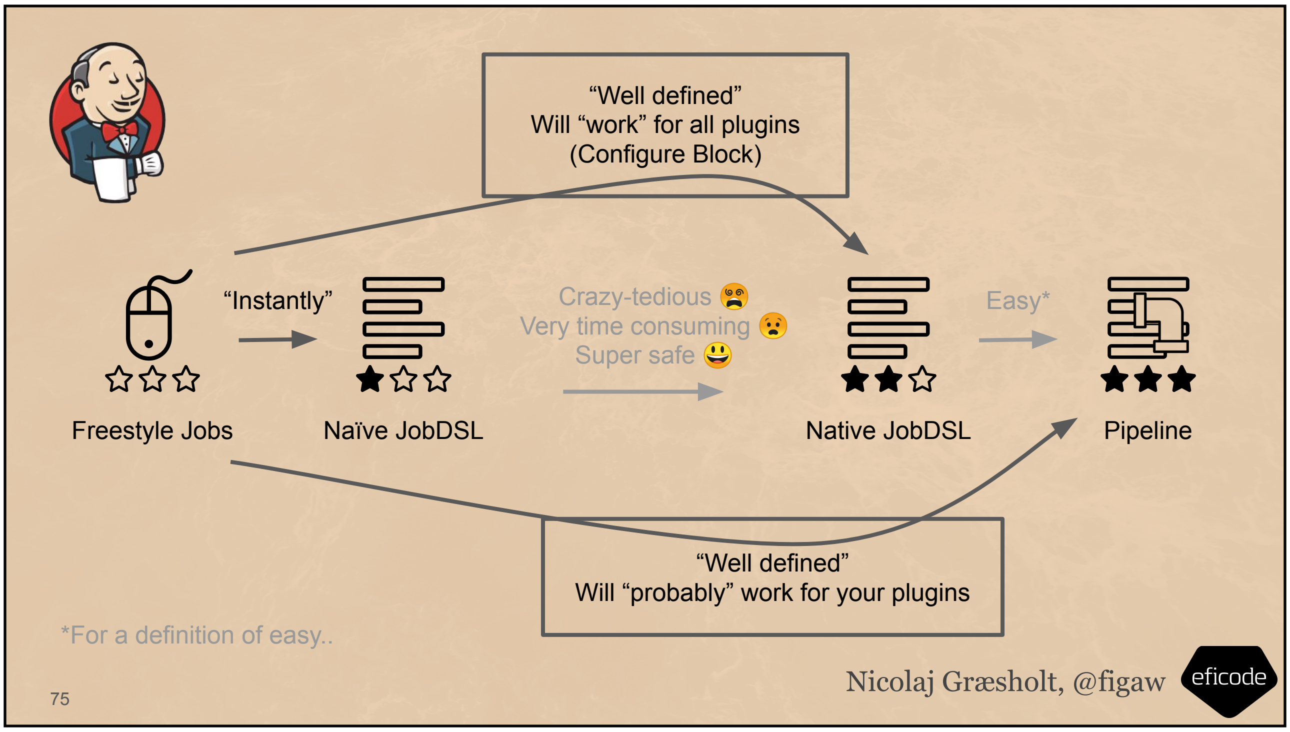
Freestyle Jobs ☆☆☆ → "Instantly" → Naïve JobDSL ★☆☆

Crazy-tedious 😫
Very time consuming 😧
Super safe 😃

Native JobDSL ★★☆ → Easy* → Pipeline ★★★

"Well defined"
Will "probably" work for your plugins

*For a definition of easy..

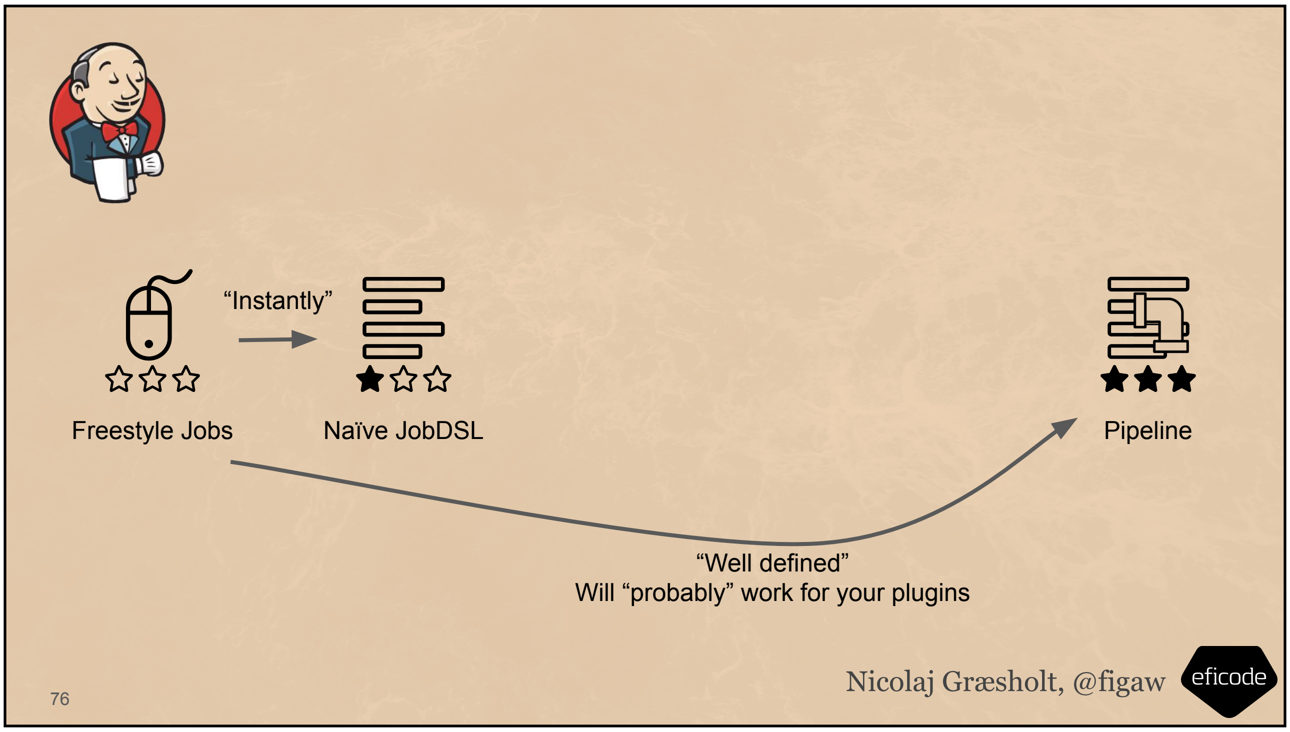


“Well defined”
Will “work” for all plugins
(Configure Block)

“Well defined”
Will “probably” work for your plugins

*For a definition of easy..

Nicolaj Græsholt, @figaw



“Well defined”
Will “probably” work for your plugins

Nicolaj Græsholt, @figaw

★★★
Pipeline

Why not stop at JobDSL?

★★★

Code
Durable
Pausable
Versatile
Extensible

https://www.jenkins.io/doc/book/pipeline/#why

Why not stop at JobDSL?

Code
Durable
Pausable
Versatile
Extensible

https://www.jenkins.io/doc/book/pipeline/#why

Pipeline crash-course🌱🌱😃

hello world pipeline

```
pipeline {
  agent any
  stages {
    stage('Hello, World!') {
      steps {
        sh 'echo "Hello, World!"'
      }
    }
  }
}
```

Pipeline crash-course🌱🌱😃

hello world pipeline

```
pipeline {
  agent any
  stages {
    stage('Hello, World!') {
      steps {
        sh 'echo "Hello, World!"'
      }
    }
  }
}
```

Pipeline crash-course🌱🌱😃

hello world pipeline

```
pipeline {
  agent any
  stages {
    stage('Hello, World!') {
      steps {
        sh 'echo "Hello, World!"'
      }
    }
  }
}
```

Pipeline crash-course🌱🌱😃

hello world pipeline

```
pipeline {
   agent any
   stages {
      stage('Hello, World!') {
         steps {
            sh 'echo "Hello, World!"'
         }
      }
   }
}
```

Pipeline crash-course🌱🌱😃

hello world pipeline

```
pipeline {
   agent any
   stages {
      stage('Hello, World!') {
         steps {
            sh 'echo "Hello, World!"'
         }
      }
   }
}
```

Pipeline crash-course🌱🌱😃

★★★

hello world pipeline

```
pipeline {
  agent any
  stages {
    stage('Hello, World!') {
      steps {
        sh 'echo "Hello, World!"'
      }
    }
  }
}
```

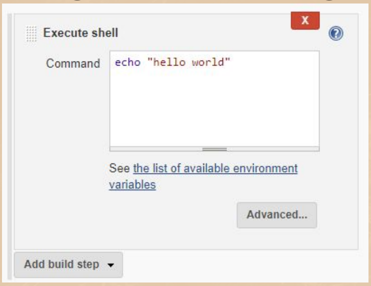
Pipeline crash-course🌱🌱😃

★★★

hello world pipeline

```
pipeline {
  agent any
  stages {
    stage('Hello, World!') {
      steps {
        sh 'echo "Hello, World!"'
      }
    }
  }
}
```

“Large” vs. “Huge”? 🙃


Execute shell

Command `echo "hello world"`

See the list of available environment variables

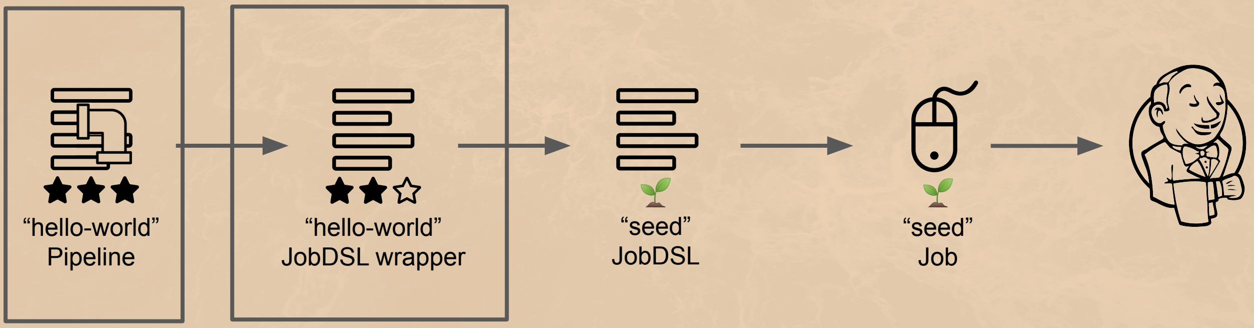
hello-world.groovy

```
job('hello-world') {
  steps {
    shell('echo Hello, World!')
  }
}
```

hello world pipeline

```
pipeline {
  agent any
  stages {
    stage('Hello, World!') {
      steps {
        sh 'echo "Hello, World!"'
      }
    }
  }
}
```

Goal 🌱🌱😃



JobDSL wrapper 😃

hello-pipeline.groovy

```
pipelineJob('hello-pipeline-inline') {
  definition { cps { script("""
     ...







  """) } }
}
```

https://jenkinsci.github.io/job-dsl-plugin/#path/pipelineJob-definition-cps-script

Nicolaj Græsholt, @figaw

JobDSL wrapper 😃

hello-pipeline.groovy

```
pipelineJob('hello-pipeline-inline') {
  definition { cps { script("""
    pipeline {
      agent any
      stages {
        stage('Hello, World!') {
          steps {
            sh 'echo "Hello, World!"'
          }
        }
      }
    }
  """) } }
}
```

https://jenkinsci.github.io/job-dsl-plugin/#path/pipelineJob-definition-cps-script

Nicolaj Græsholt, @figaw

Demo Time: “hello-pipeline-inline” 🌱

Will Show: JobDSL from slide works

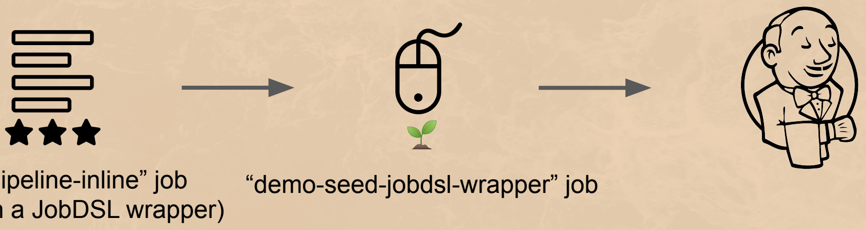
seed-hello-pipeline-inline.groovy

```
pipelineJob('hello-pipeline-inline') {
  definition { cps { script("""
    pipeline {
      agent any
      stages {
        stage('Hello, World!') {
          steps {
            sh 'echo "Hello, World!"'
          }
        }
      }
    }
  """) } }
}
```

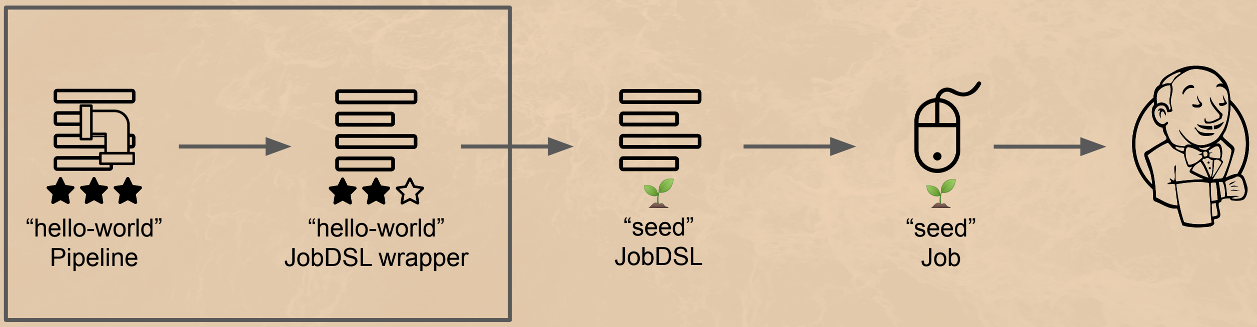
Source: https://www.artstation.com/artwork/1XmOe

Again, Disclaimer, Script security is off

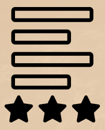
Goal 🌱🌱😃


“hello-pipeline-inline” job
(Pipeline in a JobDSL wrapper)
“demo-seed-jobdsl-wrapper” job

Goal 🌱🌱😃

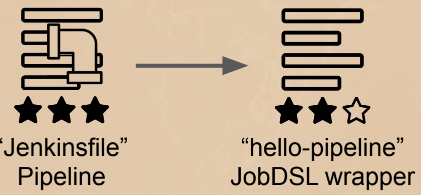


Pipeline 😃



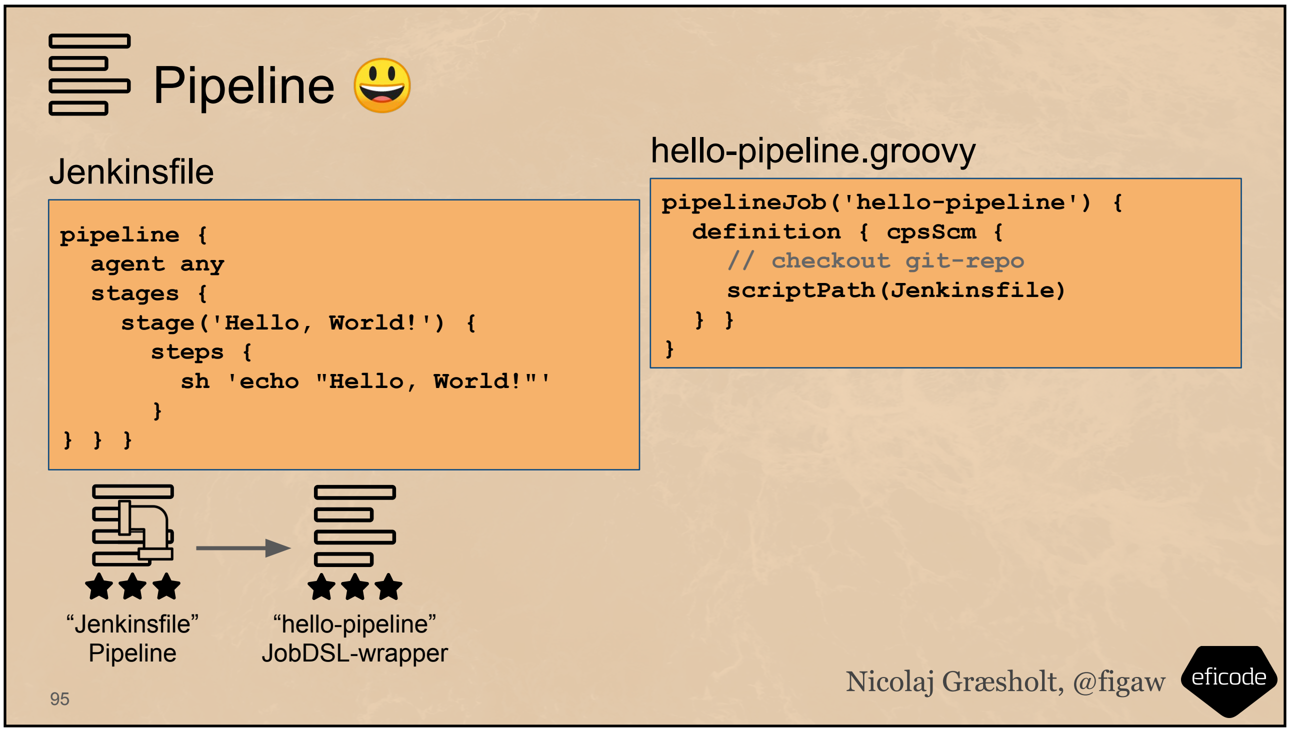
hello-pipeline-inline.groovy

```
pipelineJob('hello-pipeline-inline') {
  definition { cps { script("""
     ... // inline pipeline
  """) } }
}
```



hello-pipeline.groovy

```
pipelineJob('hello-pipeline') {
  definition { cpsScm {
    // checkout git-repo
    scriptPath(Jenkinsfile)
  } }
}
```



Pipeline 😃

Jenkinsfile

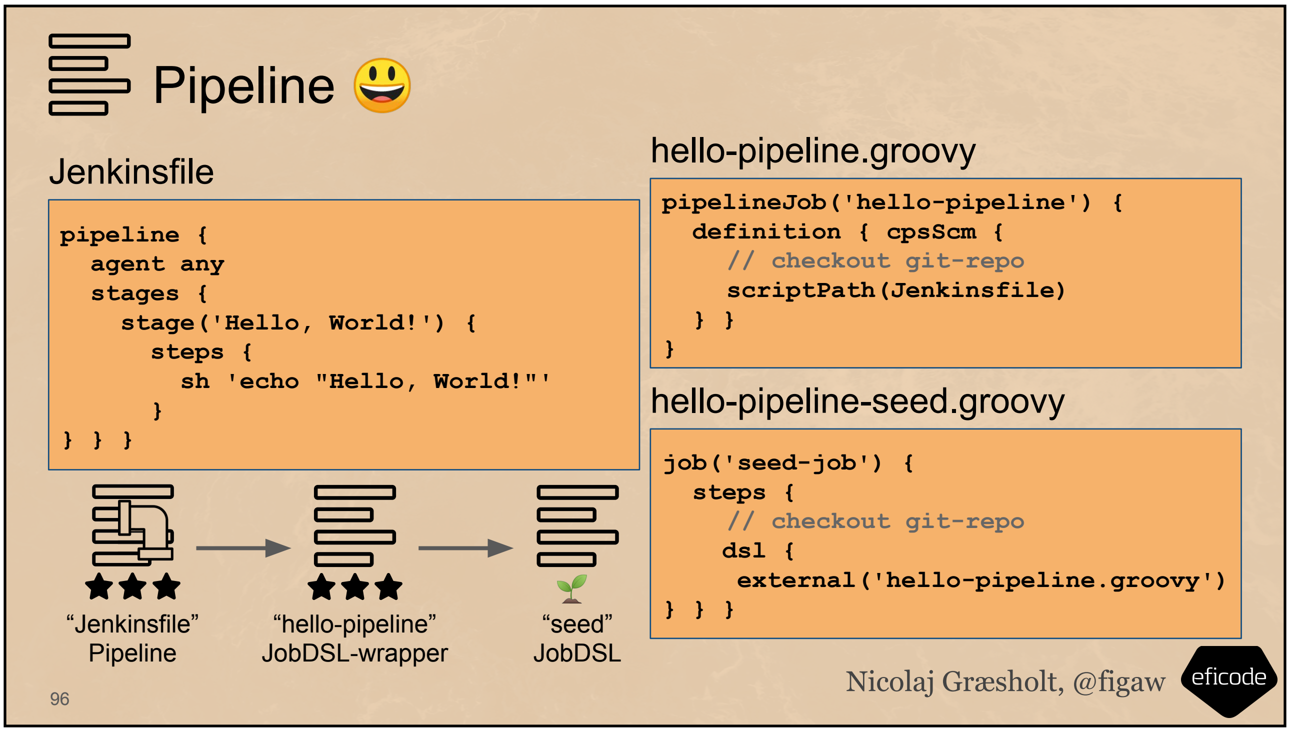
```
pipeline {
  agent any
  stages {
    stage('Hello, World!') {
      steps {
        sh 'echo "Hello, World!"'
      }
} } }
```

hello-pipeline.groovy

```
pipelineJob('hello-pipeline') {
  definition { cpsScm {
    // checkout git-repo
    scriptPath(Jenkinsfile)
  } }
}
```

★★★ "Jenkinsfile" Pipeline → ★★★ "hello-pipeline" JobDSL-wrapper

Nicolaj Græsholt, @figaw eficode



Pipeline 😃

Jenkinsfile

```
pipeline {
  agent any
  stages {
    stage('Hello, World!') {
      steps {
        sh 'echo "Hello, World!"'
      }
} } }
```

hello-pipeline.groovy

```
pipelineJob('hello-pipeline') {
  definition { cpsScm {
    // checkout git-repo
    scriptPath(Jenkinsfile)
  } }
}
```

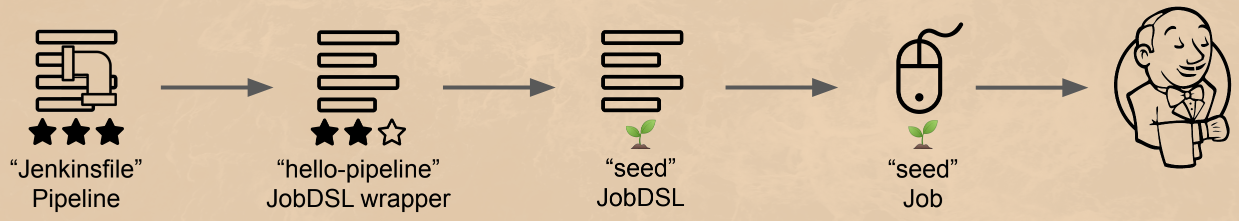
hello-pipeline-seed.groovy

```
job('seed-job') {
  steps {
    // checkout git-repo
    dsl {
     external('hello-pipeline.groovy')
} } }
```

★★★ "Jenkinsfile" Pipeline → ★★★ "hello-pipeline" JobDSL-wrapper → 🌱 "seed" JobDSL

Nicolaj Græsholt, @figaw eficode

Goal 🌱🌱😃



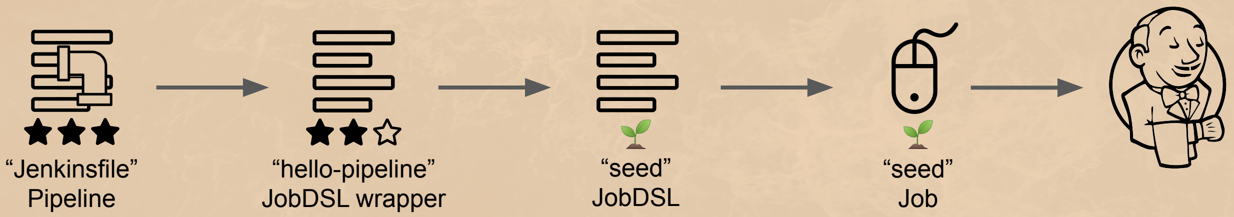
Freestyle Jobs and the 5 Stages of CasC

“Configuration as Code model”

- 😎 Confidence - **Who** changed **What**, **When**, from **Which Value**?
- 🤑 Value - Reuse through **Shared Libraries** and **Software Design Patterns**
- 😄 Relaxation - Automagical reconfiguration of jobs
- 🤗 Collaboration* - Working **With** my Colleagues
- 😃 Happiness - This **is** Great!

*not an actual emotion, but it should be.

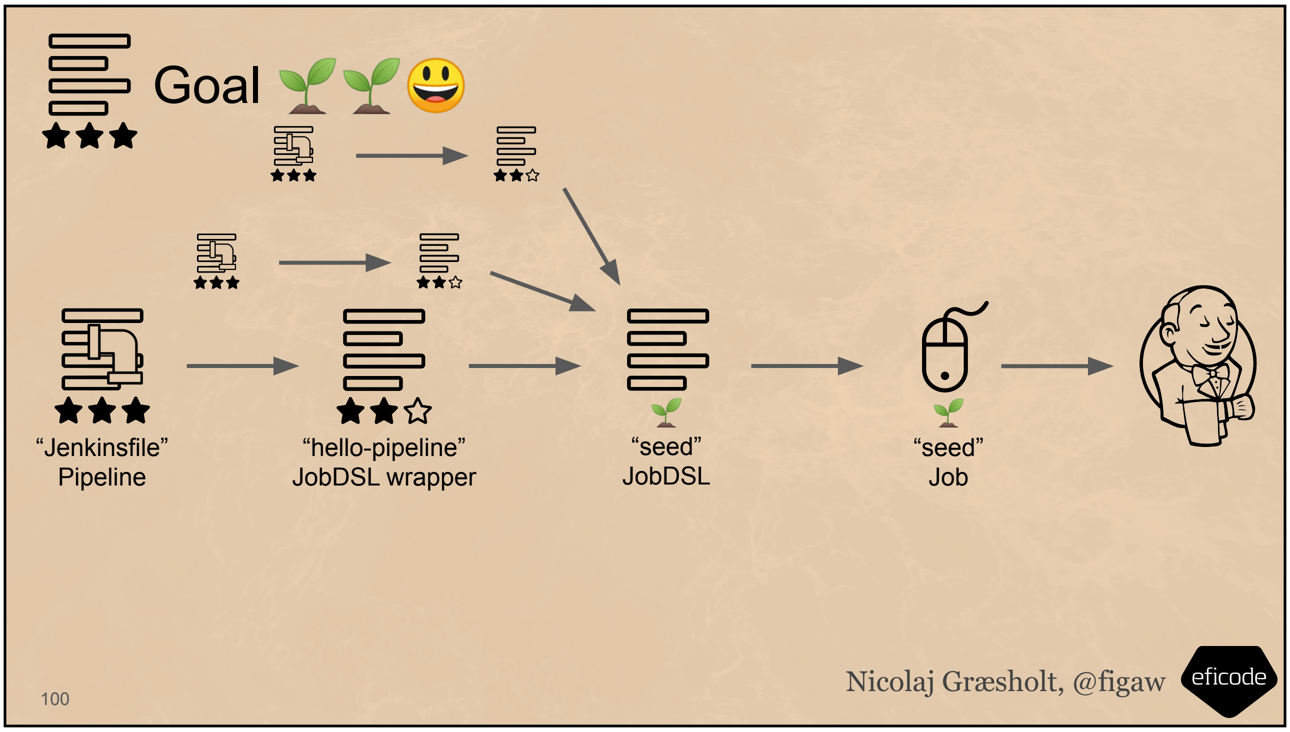
Goal 🌱🌱😃



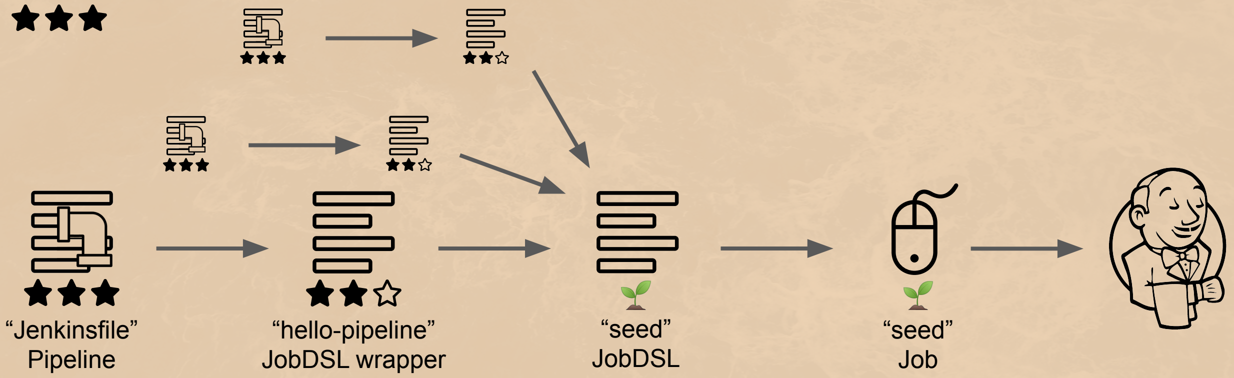
We Won, We Did It! 🍾

Nicolaj Græsholt, @figaw

eficode



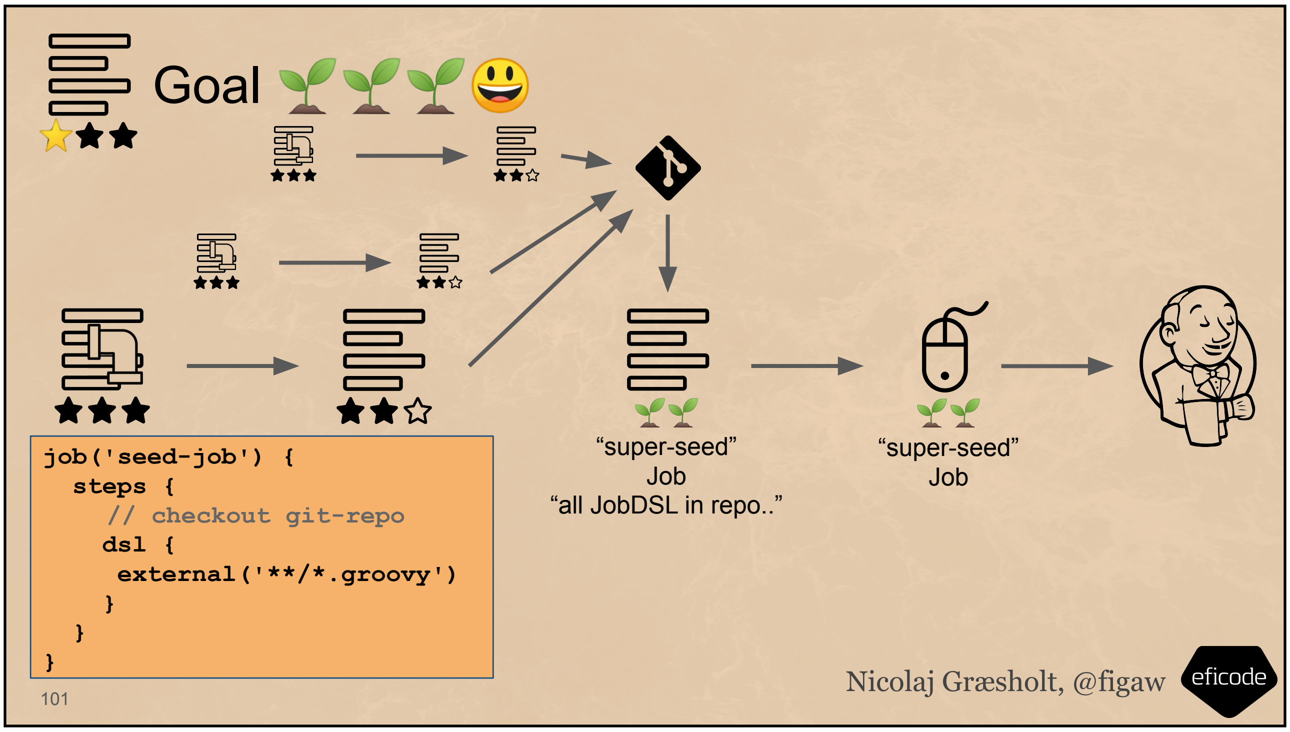
Goal 🌱🌱😃



Nicolaj Græsholt, @figaw

eficode

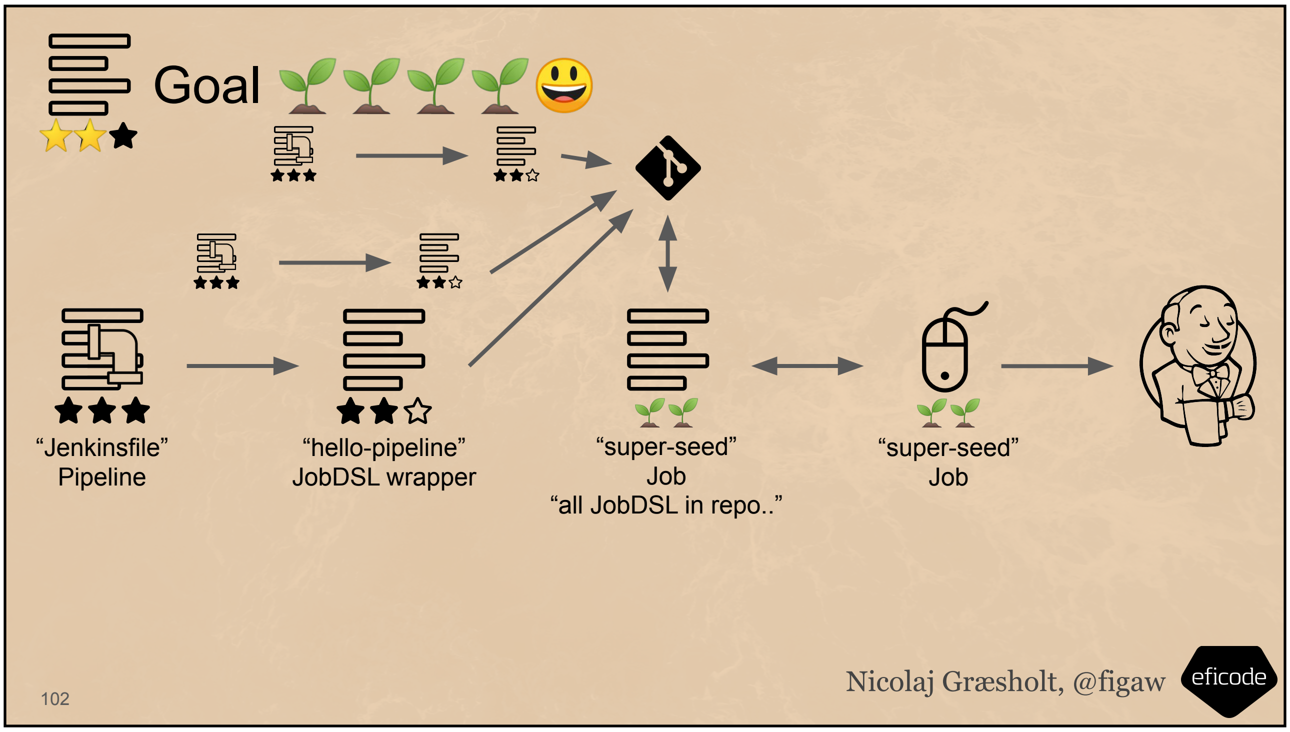
Goal 🌱🌱🌱😃



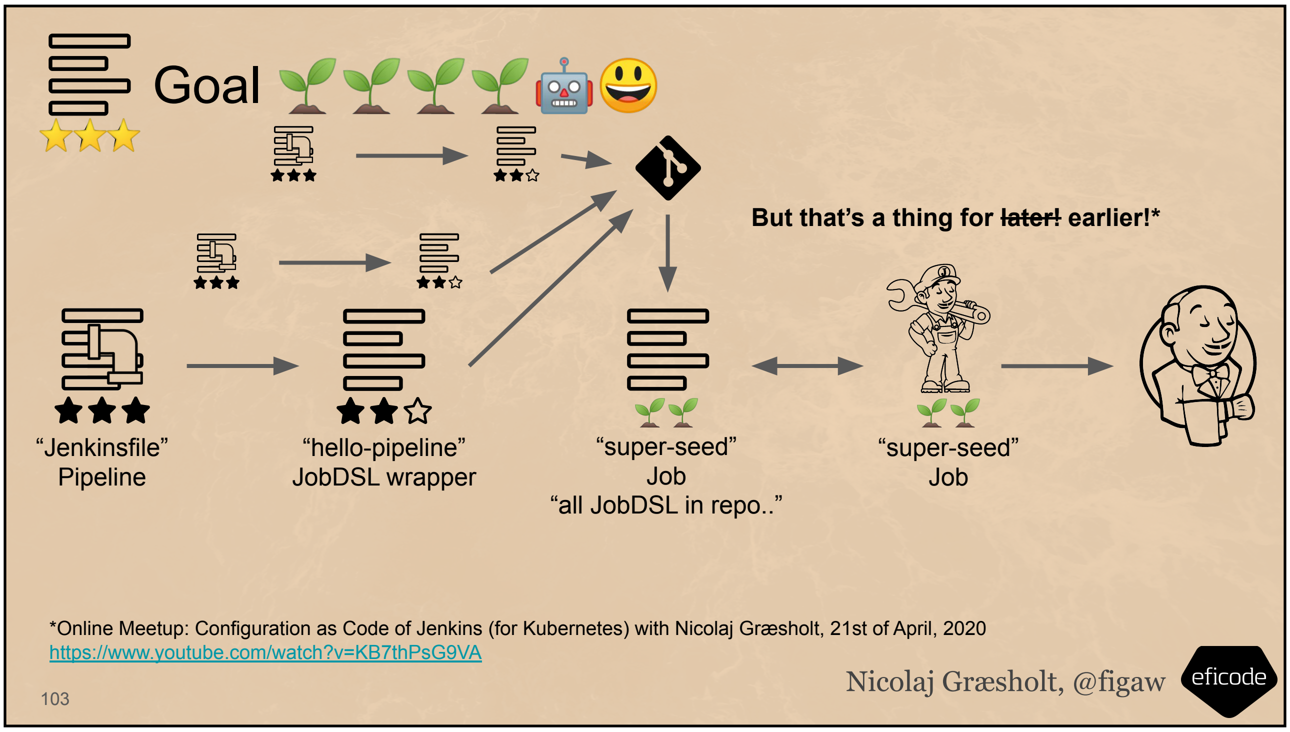
```
job('seed-job') {
  steps {
    // checkout git-repo
    dsl {
     external('**/*.groovy')
    }
  }
}
```

Nicolaj Græsholt, @figaw

Goal 🌱🌱🌱🌱😃



Nicolaj Græsholt, @figaw



Goal 🌱🌱🌱🌱🤖😃

★★★
“Jenkinsfile” Pipeline
★★☆
“hello-pipeline” JobDSL wrapper
“super-seed” Job “all JobDSL in repo..”
“super-seed” Job

But that’s a thing for ~~later!~~ earlier!*

*Online Meetup: Configuration as Code of Jenkins (for Kubernetes) with Nicolaj Græsholt, 21st of April, 2020
https://www.youtube.com/watch?v=KB7thPsG9VA

Nicolaj Græsholt, @figaw

THANK YOU!
QUESTIONS?

Nicolaj Græsholt - @figaw on Twitter
DevOps Consultant at Eficode Praqma