

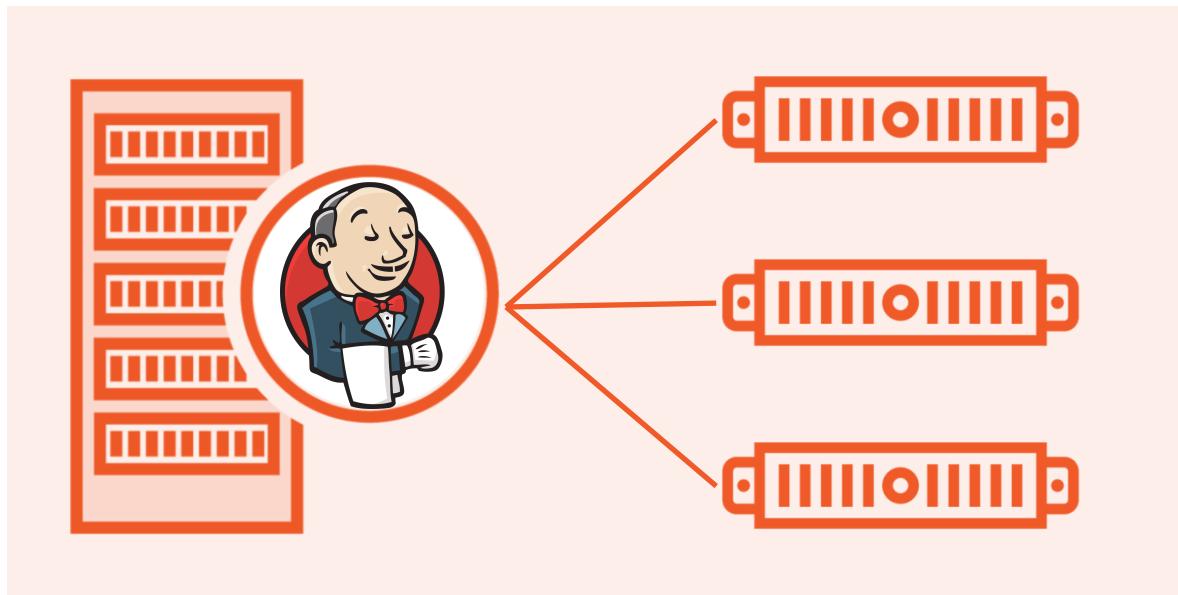
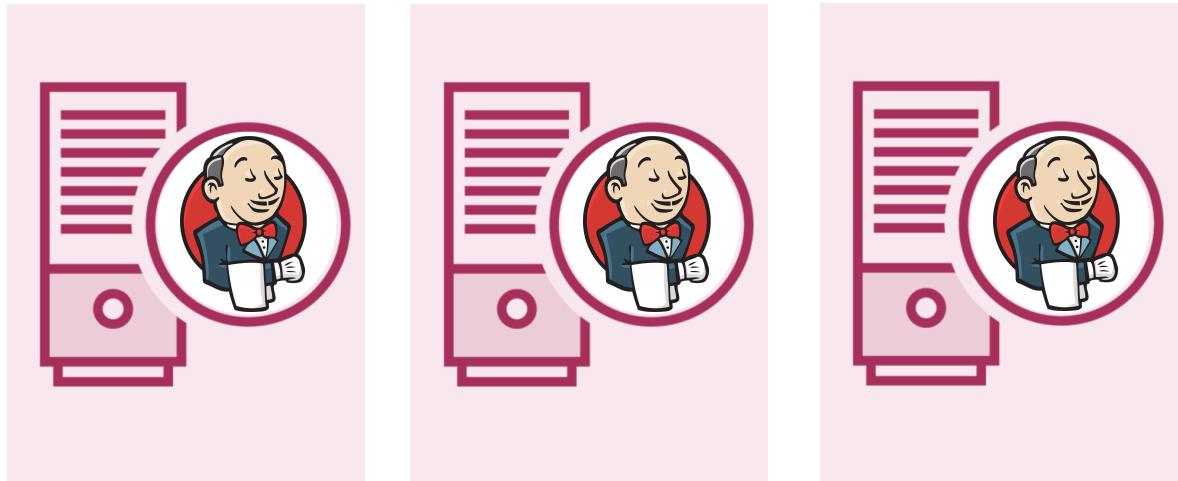
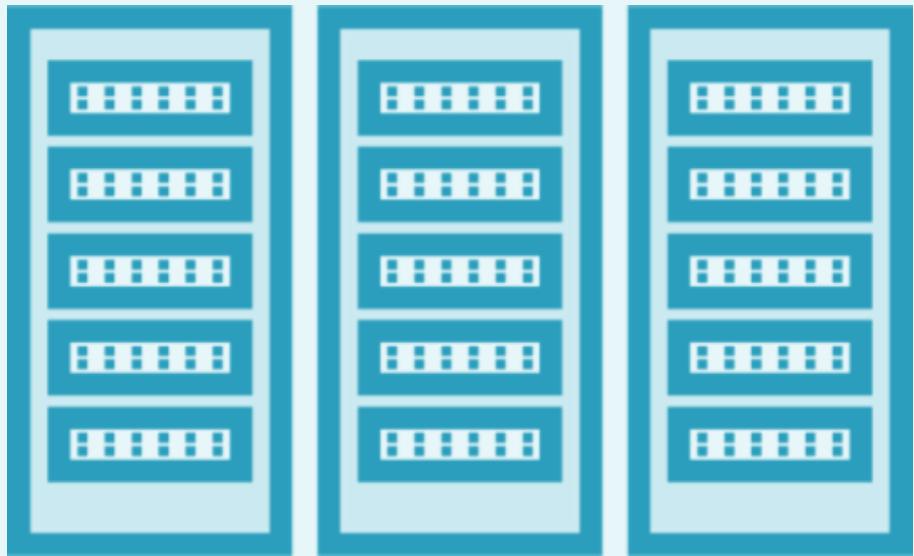
Using Pipelines to Support Your Workflow

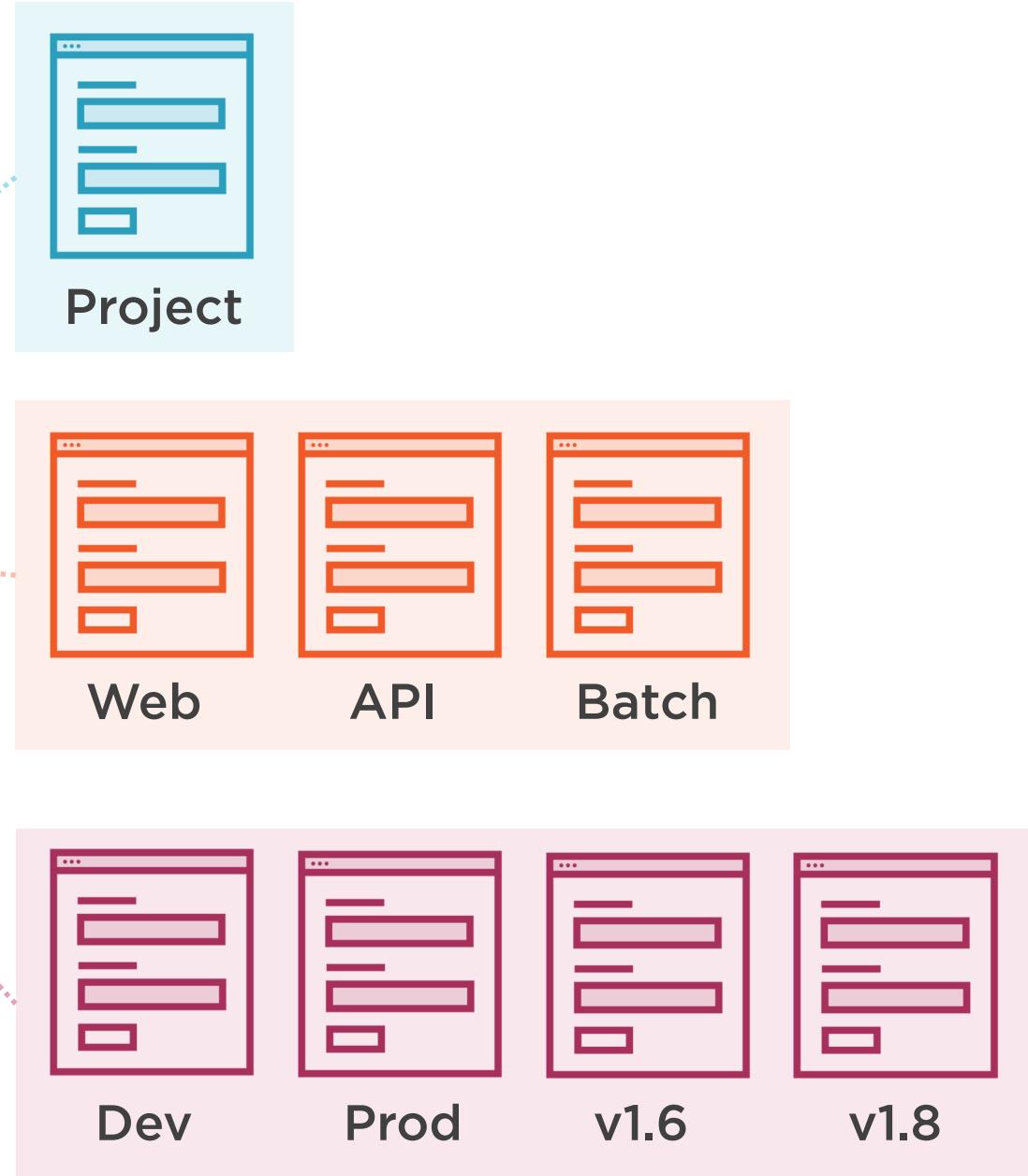
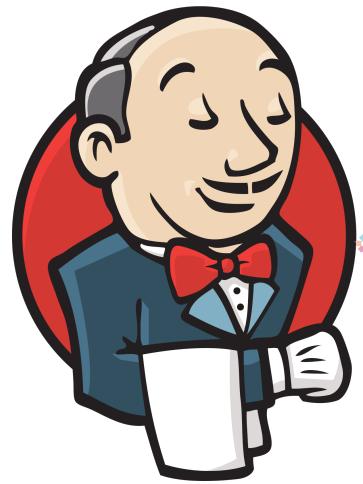


Elton Stoneman

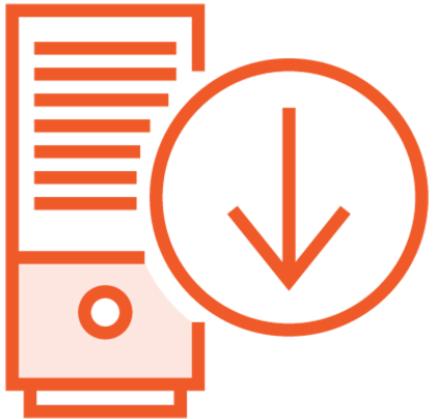
CONSULTANT & TRAINER

@EltonStoneman | blog.sixeyed.com





Modern Jenkins: Minimizing Complexity



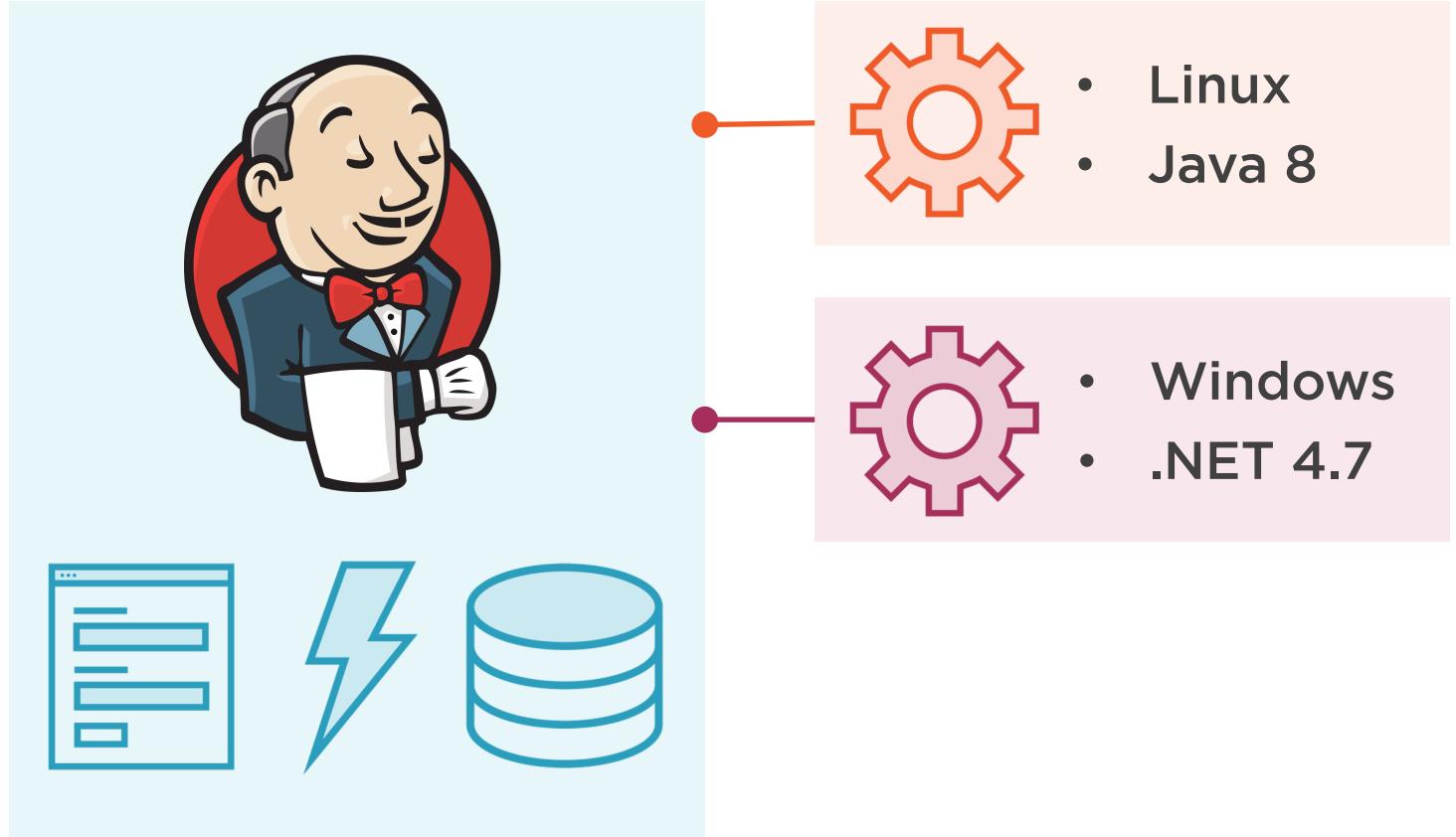
Configuring Jenkins

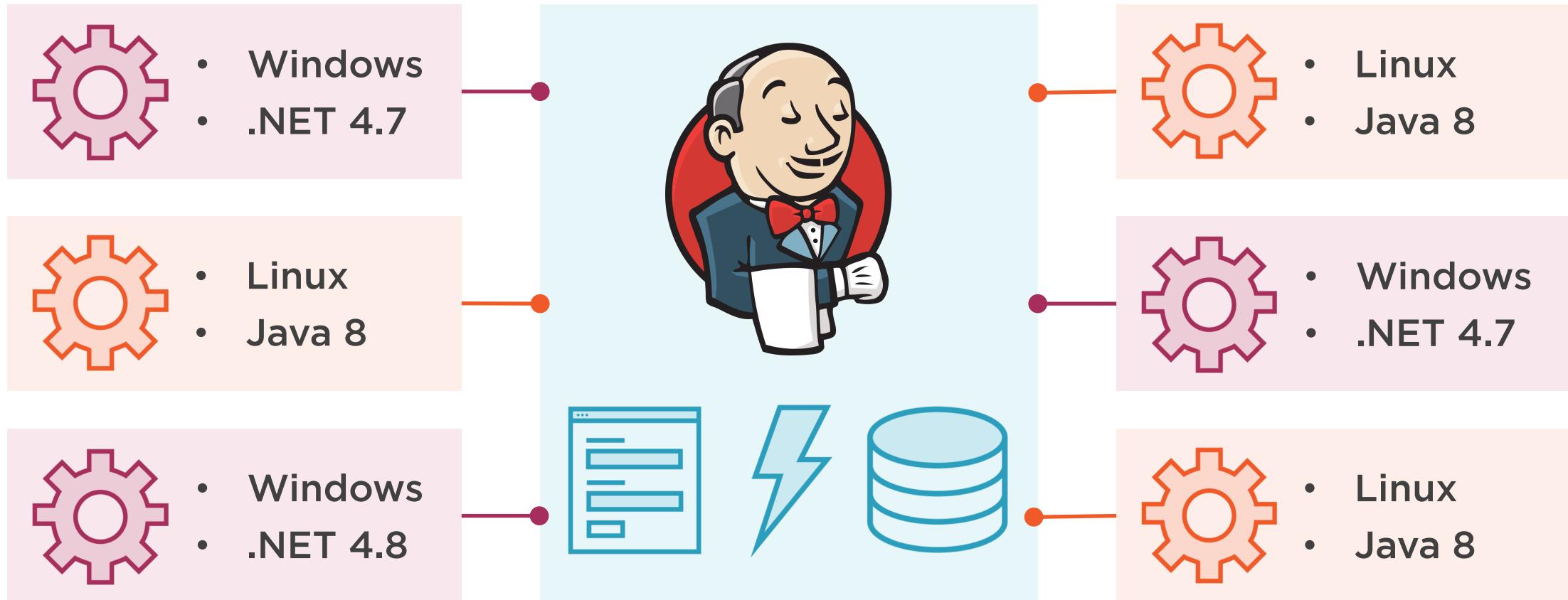


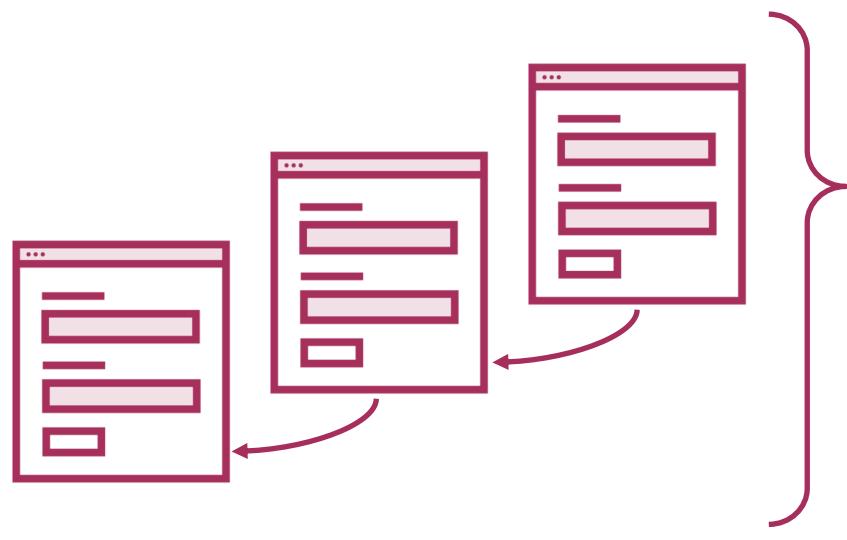
Creating build jobs

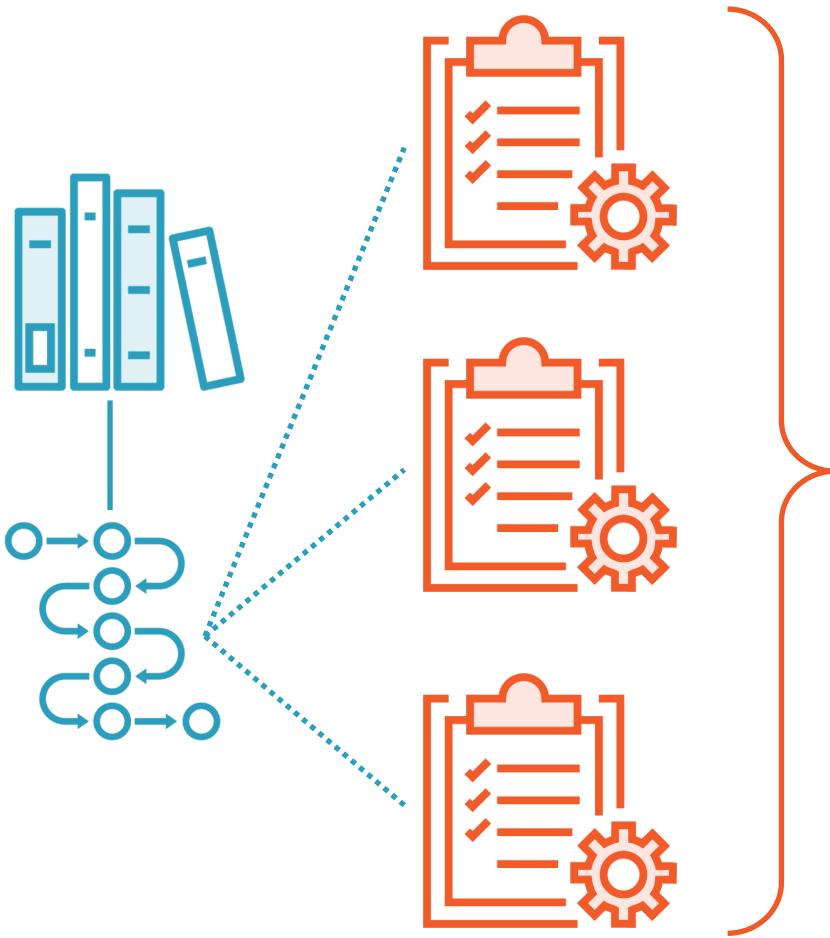


Maintaining jobs

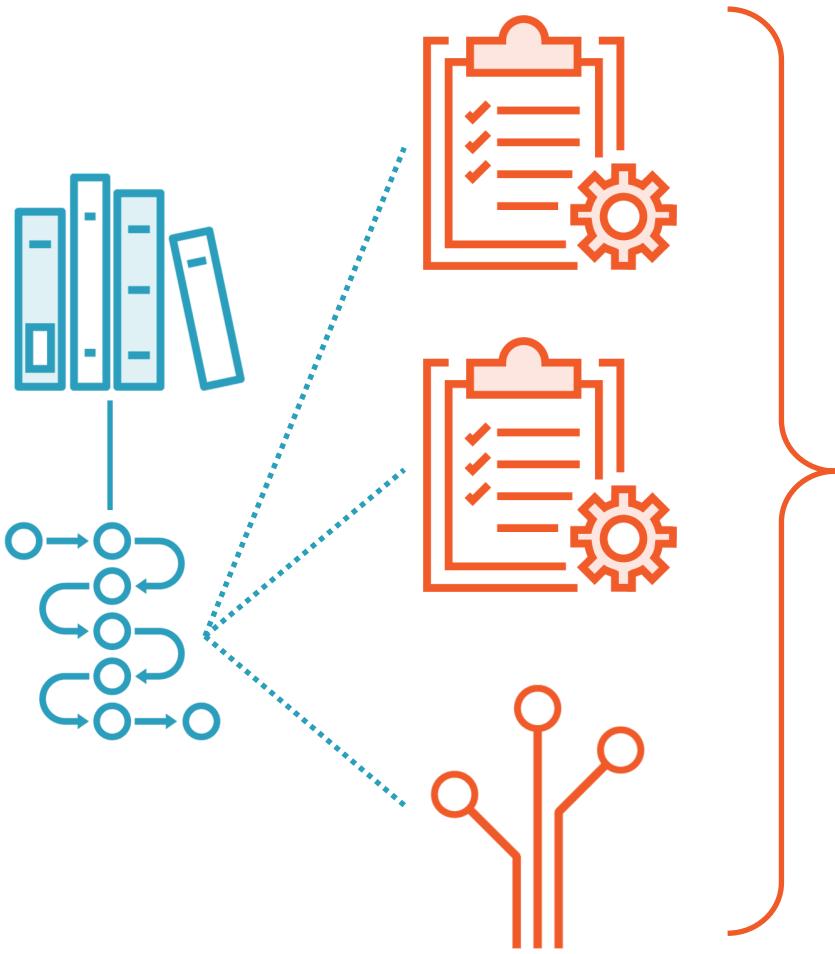






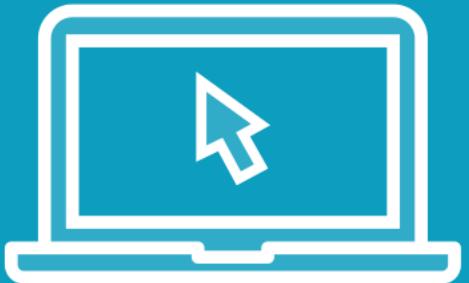


- Linux
- Java 8
- Windows .NET 4.7
- Windows .NET 4.8



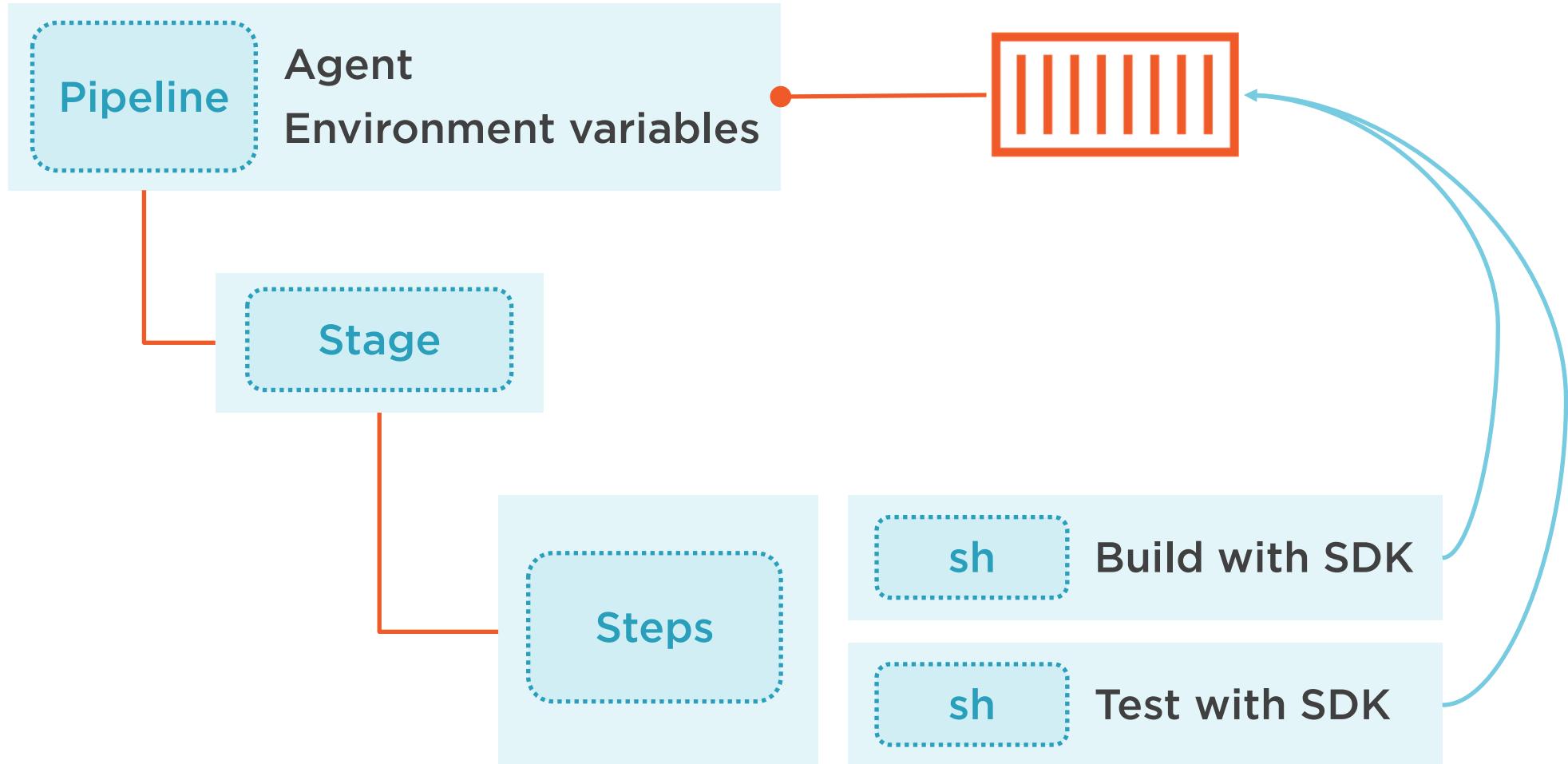
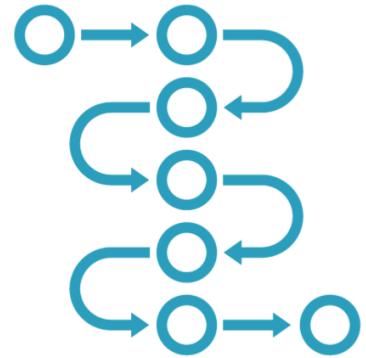
- Linux
 - Java 8
- Windows
 - .NET 4.7
- Windows
 - .NET 4.8

Demo



Powering pipelines with Docker

- Containers as build agents
- Customizing the build container
- Using the Docker Pipeline plugin



Jenkinsfile

```
pipeline {

    agent  {

        docker {

            image 'mcr.microsoft.com/dotnet/core/sdk:3.1.101'

        }
    }

    stages {

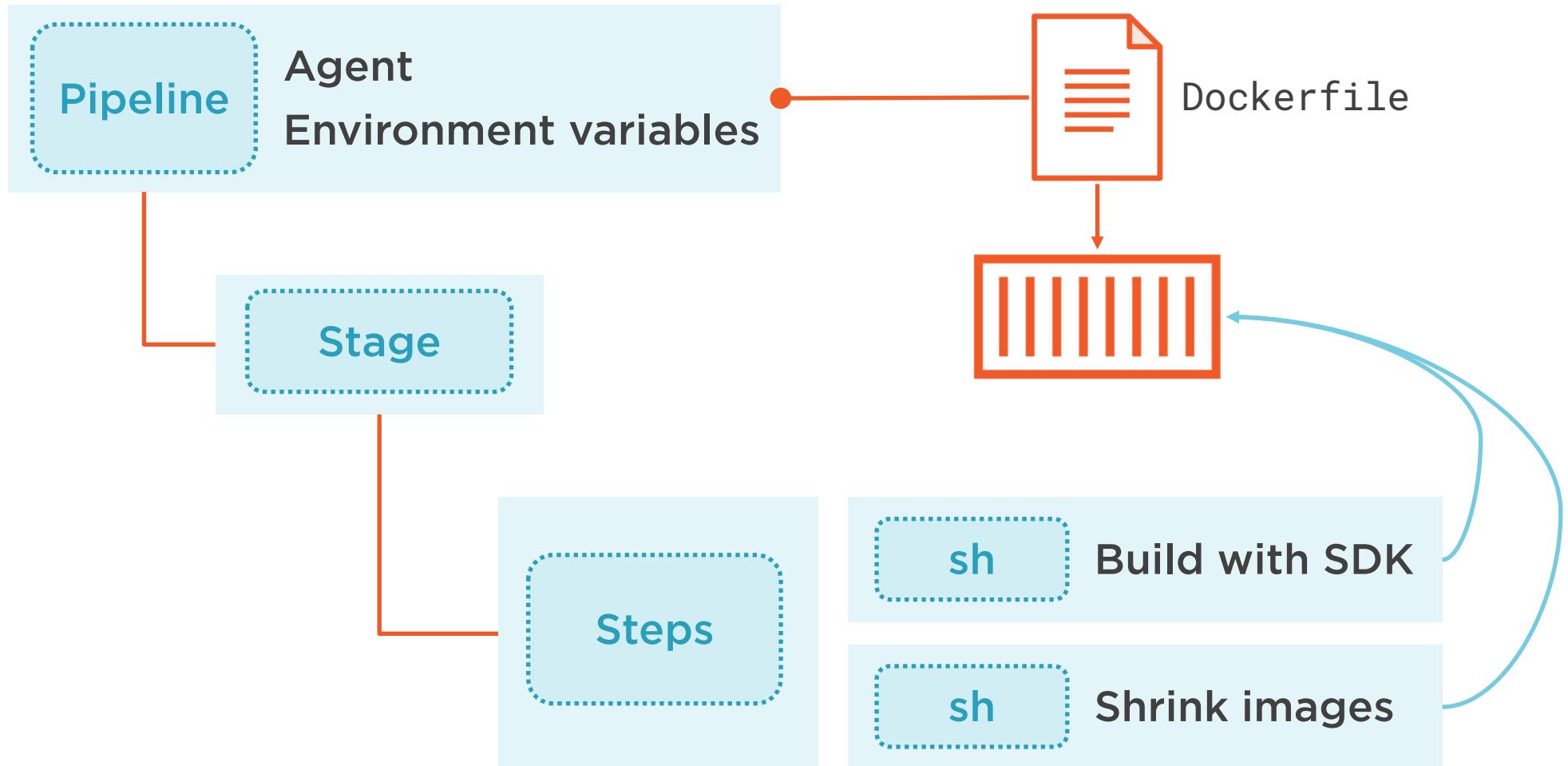
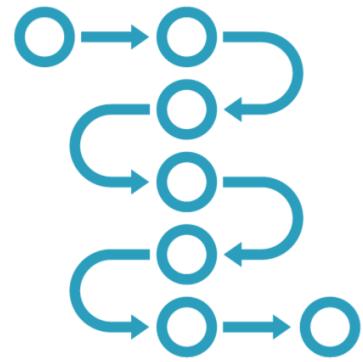
        stage('Build') {

            steps {

                sh 'dotnet build "$WORKSPACE/m4/src/Pi.Web/Pi.Web.csproj"'

            }
        }

        // ...
    }
}
```



Jenkinsfile

```
pipeline {

    agent  {

        dockerfile {

            dir 'm4'

            filename 'Dockerfile.sdk'

        }

    }

    stages {

        stage('Build') {

            steps {

                sh 'dotnet build "$WORKSPACE/m4/src/Pi.Web/Pi.Web.csproj"'

                // ...

            }

        }

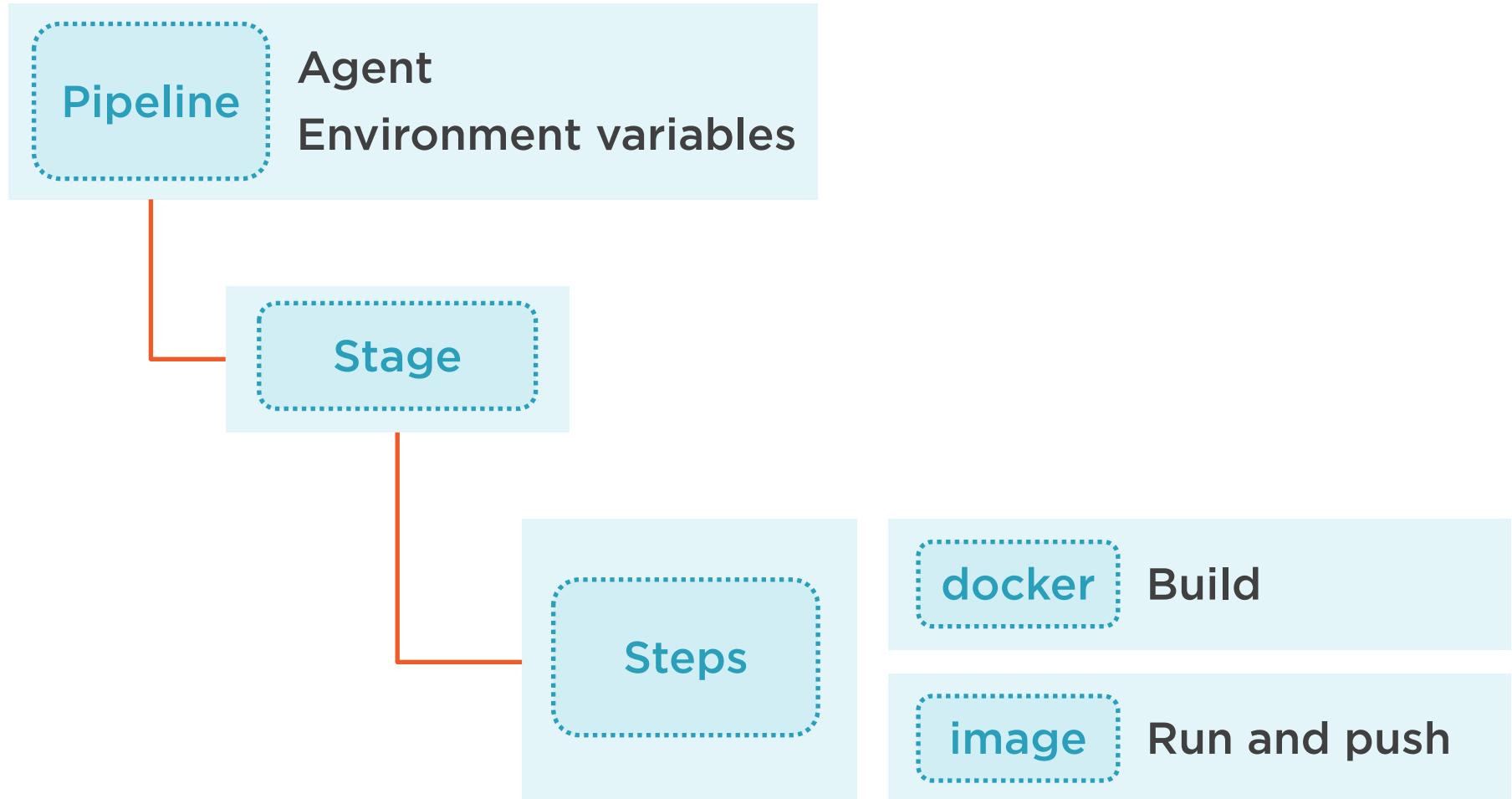
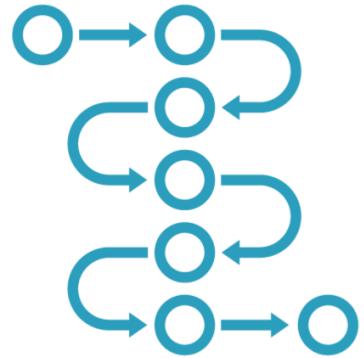
    }

}
```

Dockerfile.sdk

```
FROM mcr.microsoft.com/dotnet/core/sdk:3.1.101 AS builder
```

```
ENV PS_MODULE=m4
```



Jenkinsfile

```
pipeline {  
    agent any  
    stages {  
        stage('Build') {  
            steps {  
                script {  
                    image = docker.build("sixeyed/pi:psod-pipelines",  
                                         "--pull -f m4/Dockerfile m4")  
                }  
                // ...  
            }  
        }  
    }  
}
```

Dockerfile

```
FROM mcr.microsoft.com/dotnet/core/sdk:3.1.101 AS builder

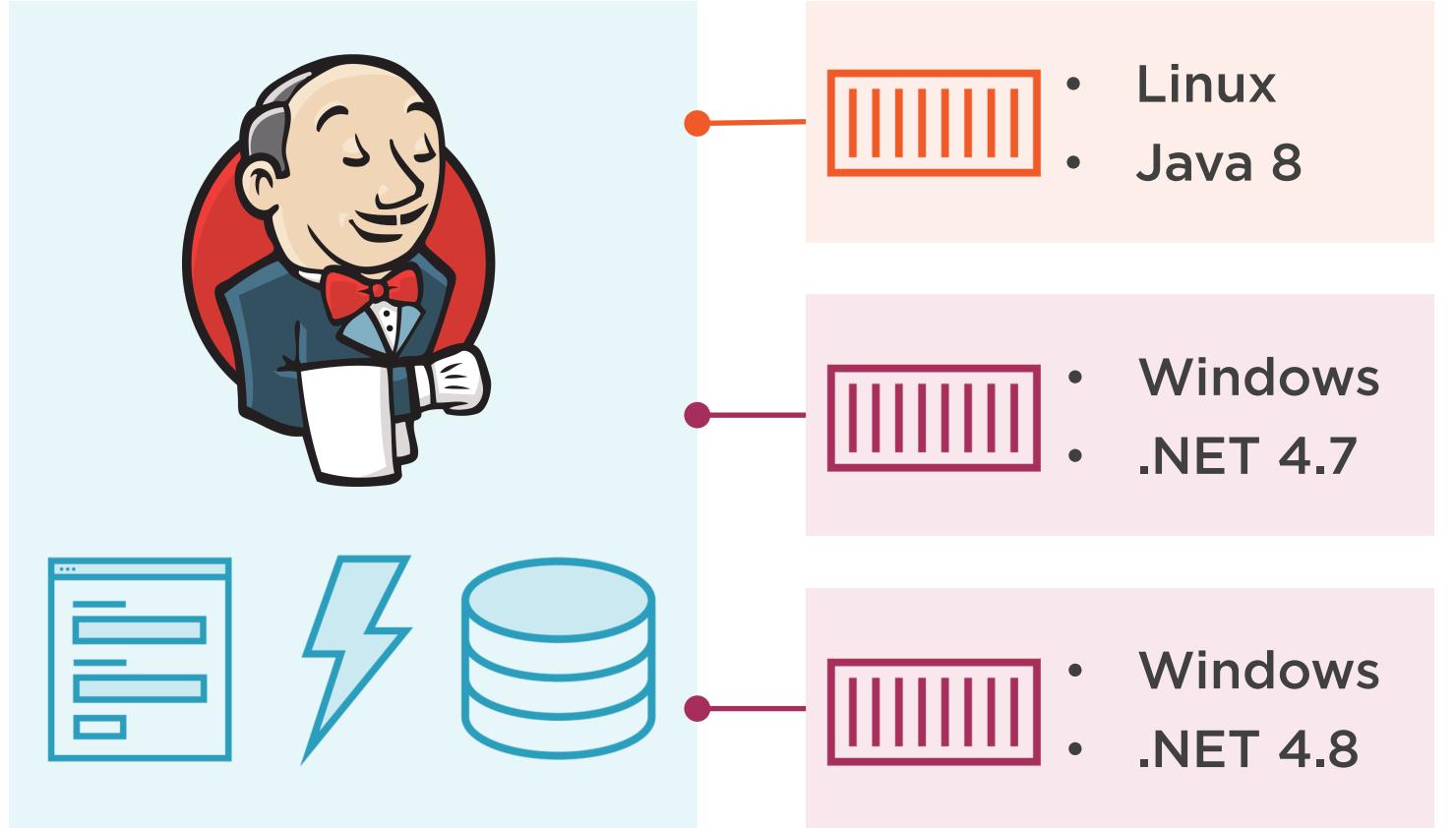
WORKDIR /src

COPY src/Pi.Math/Pi.Math.csproj ./Pi.Math/
COPY src/Pi.Runtime/Pi.Runtime.csproj ./Pi.Runtime/
COPY src/Pi.Web/Pi.Web.csproj ./Pi.Web/

WORKDIR /src/Pi.Web

RUN dotnet restore

# ...
```

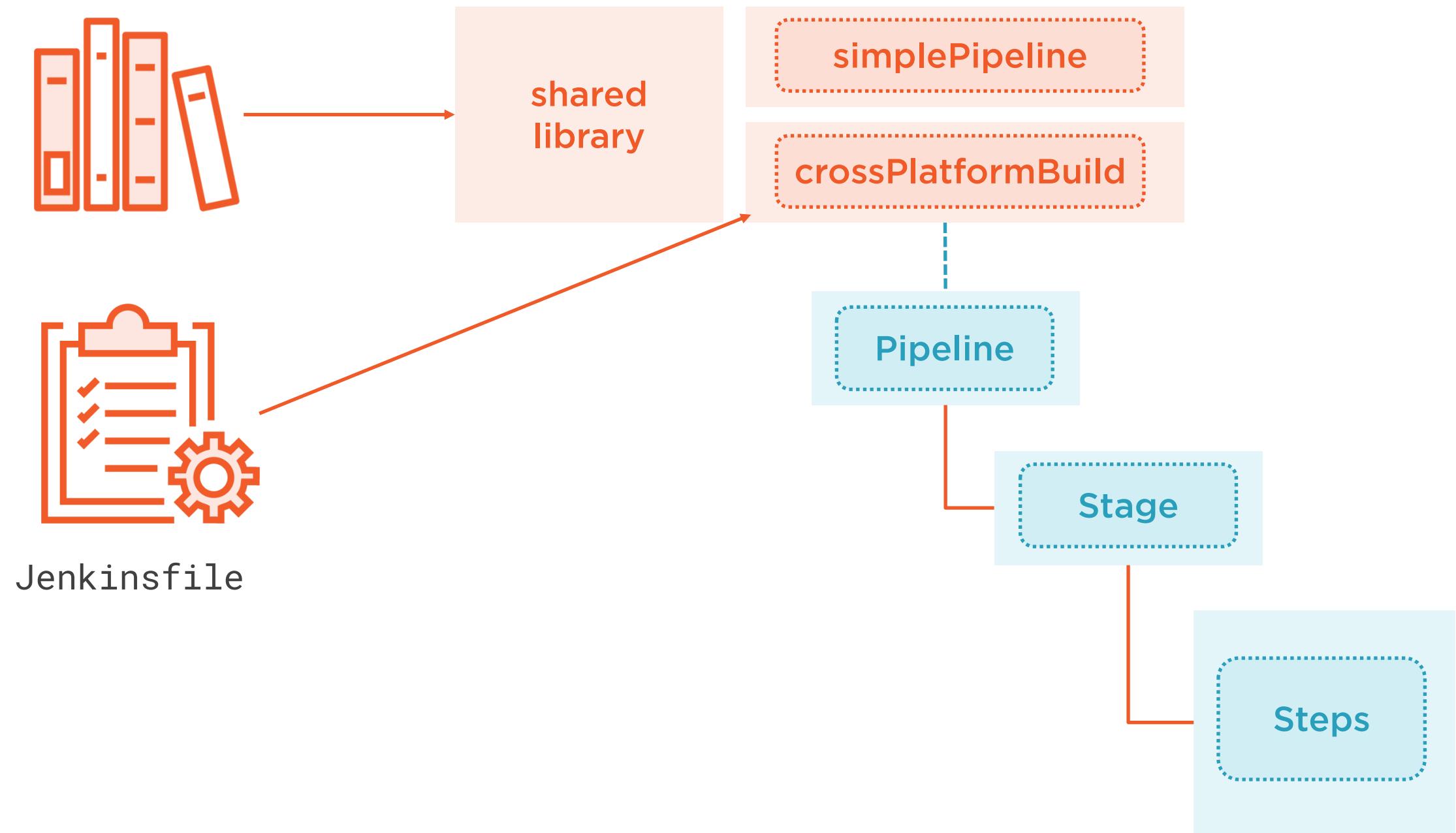


Demo



Shared pipelines and the Job DSL plugin

- Writing pipelines in shared libraries
- A cross-platform shared pipeline
- Generating pipeline jobs with DSL



simplePipeline.groovy

```
def call() {  
  
    pipeline {  
  
        agent any  
  
        environment {  
            MODULE='m4'  
        }  
  
        stages {  
            // ...  
        }  
    }  
}
```

Jenkinsfile

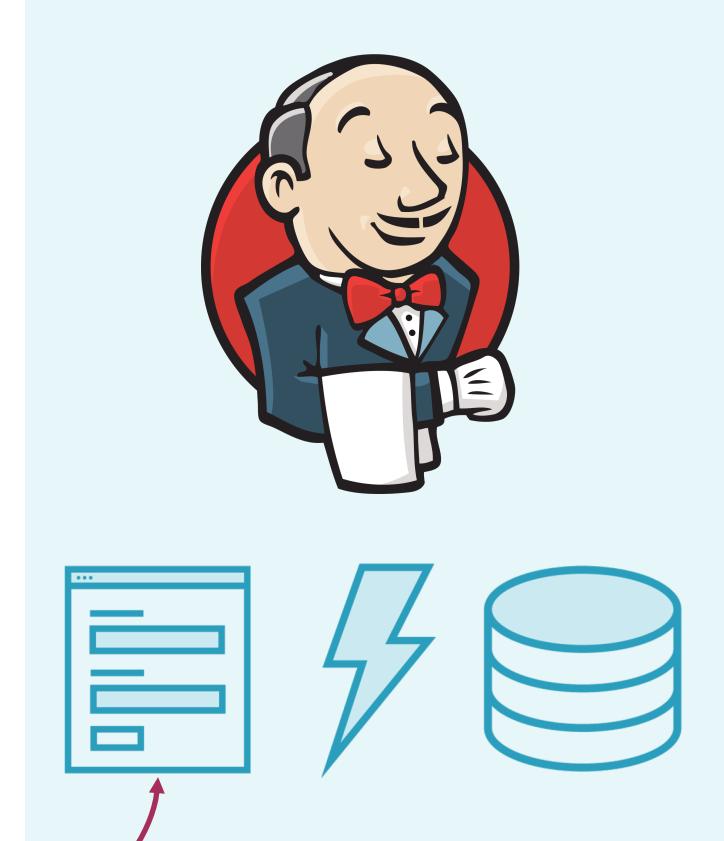
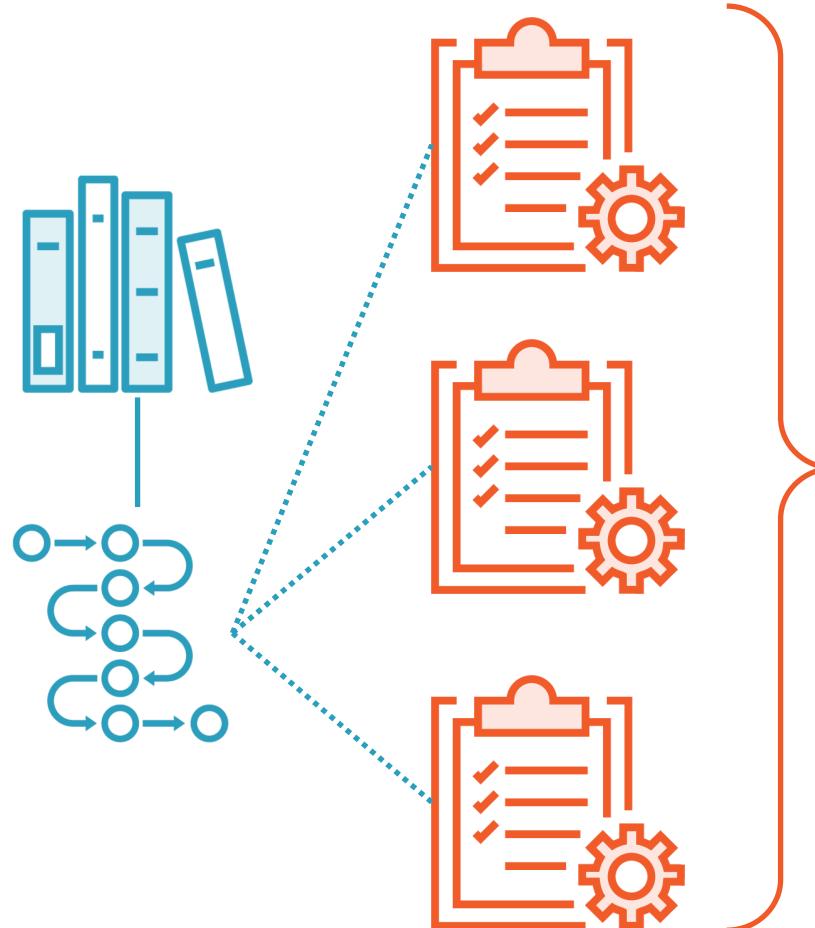
```
library identifier: 'jenkins-  
pipeline-demo-library@master',  
        retriever: modernSCM([...])  
  
simplePipeline()
```

crossPlatformBuild.groovy

```
def call(Map config) {  
  
    pipeline {  
  
        agent any  
  
        environment {  
  
            REPO_NAME = "${config.repoName}"  
  
            DOCKER_LINUX_SERVER = '...'  
  
            DOCKER_WINDOWS_SERVER = '...'  
  
        }  
  
        stages {  
  
            // ...  
        }  
    }  
}
```

Jenkinsfile

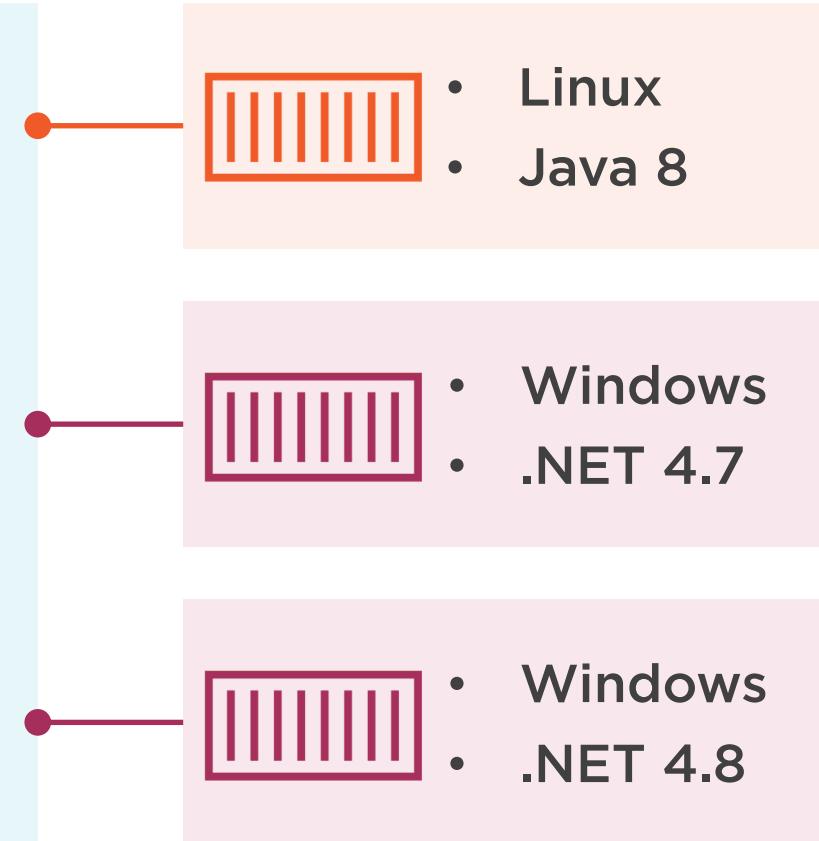
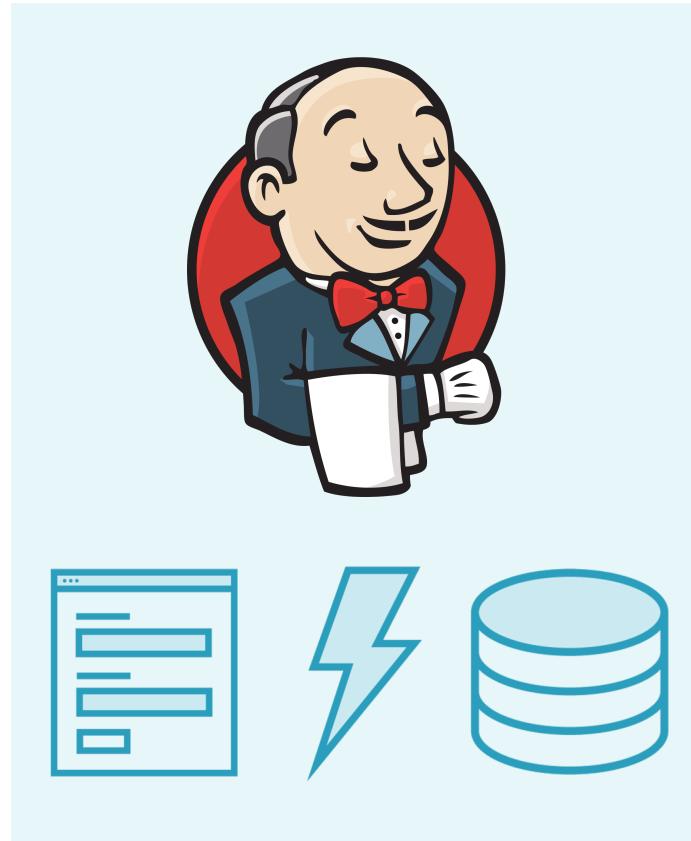
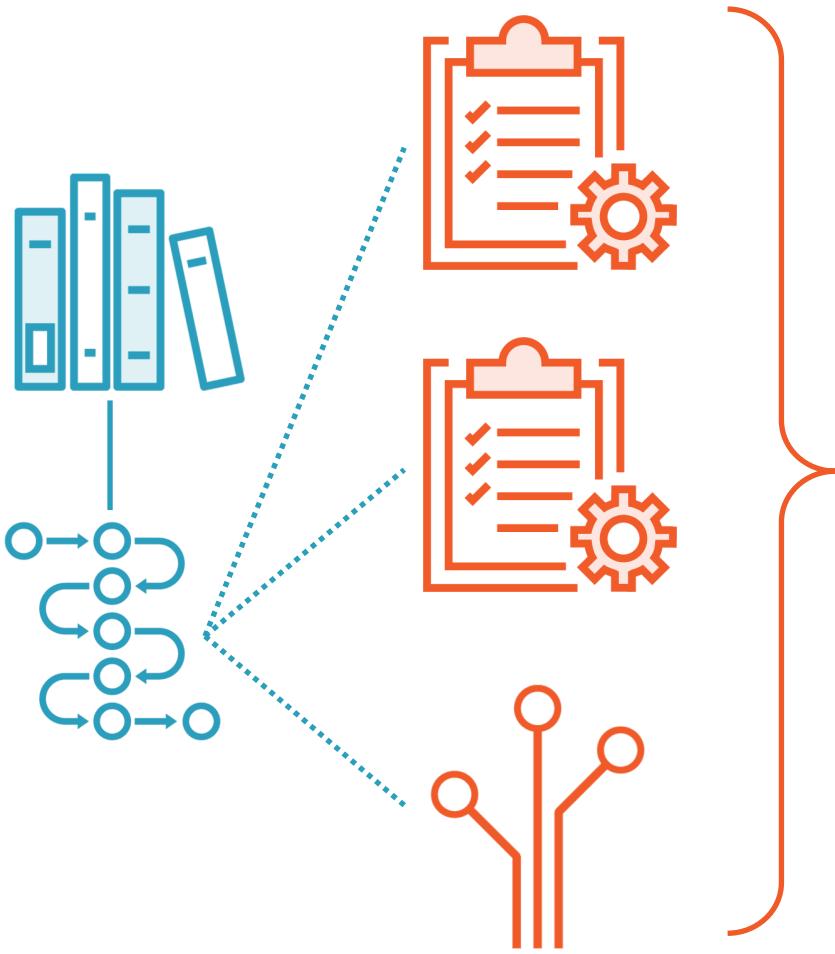
```
library identifier: 'jenkins-pipeline-demo-library@master',  
        retriever: modernSCM([...])  
  
crossPlatformBuild  
  
    repoName: 'sixeyed/pi',  
    linuxContext: 'm4',  
    windowsContext: 'm4'
```



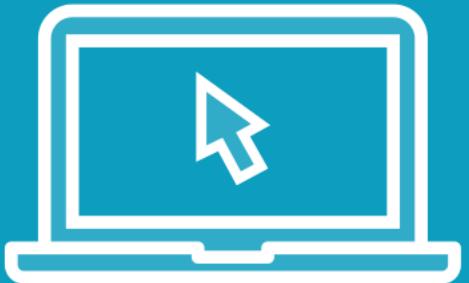
job_dsl.groovy

```
def pipelines = [
    [name: 'm4/demo2-1', scriptPath: 'm4/demo2/2.1/Jenkinsfile'],
    [name: 'm4/demo2-2', scriptPath: 'm4/demo2/2.2/Jenkinsfile']
]

for(p in pipelines) {
    pipelineJob("${p.name}") {
        definition {
            cpsScm {
                scm { // ... }
                scriptPath("${p.scriptPath}")
            }
        }
    }
}
```

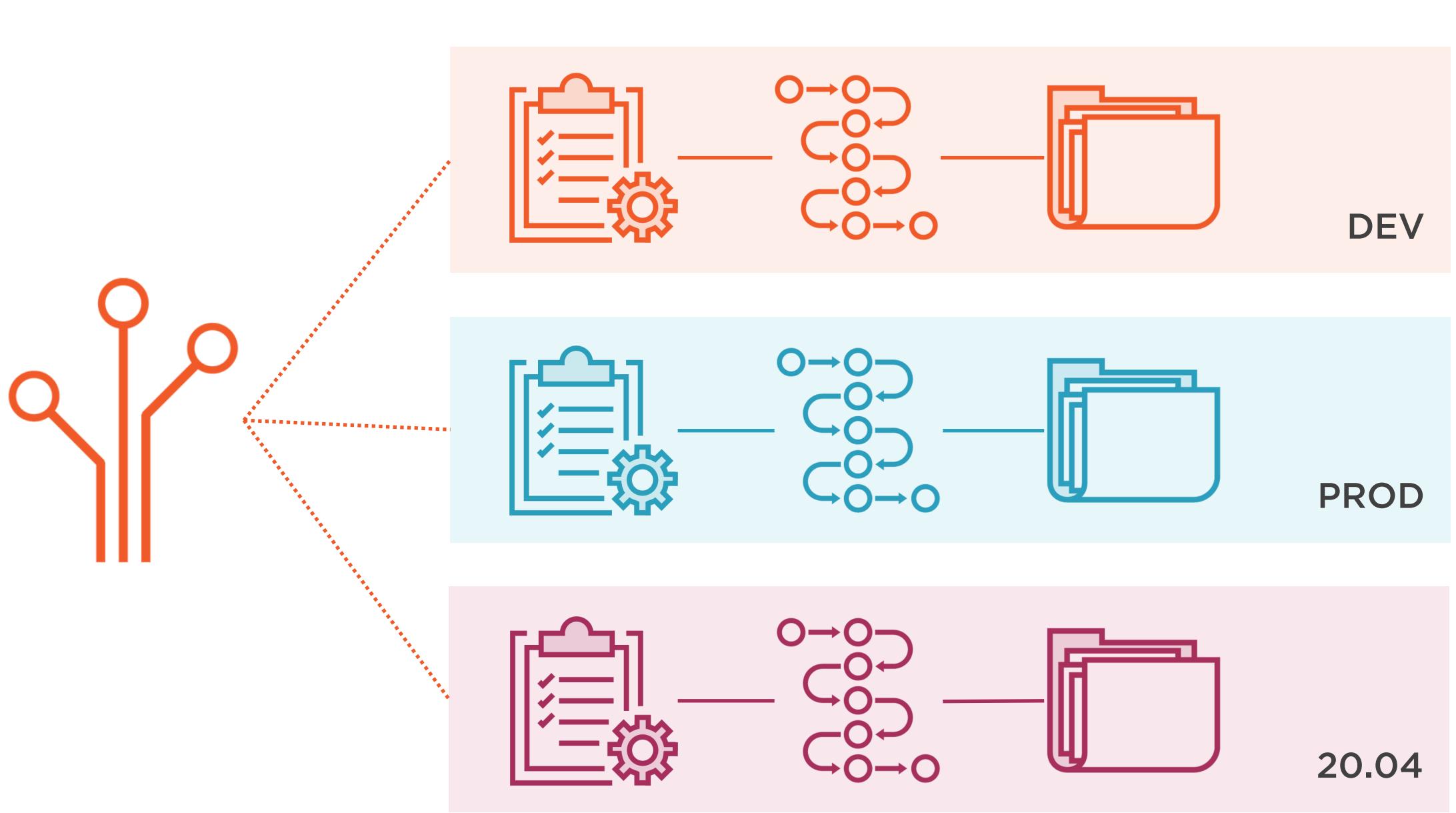


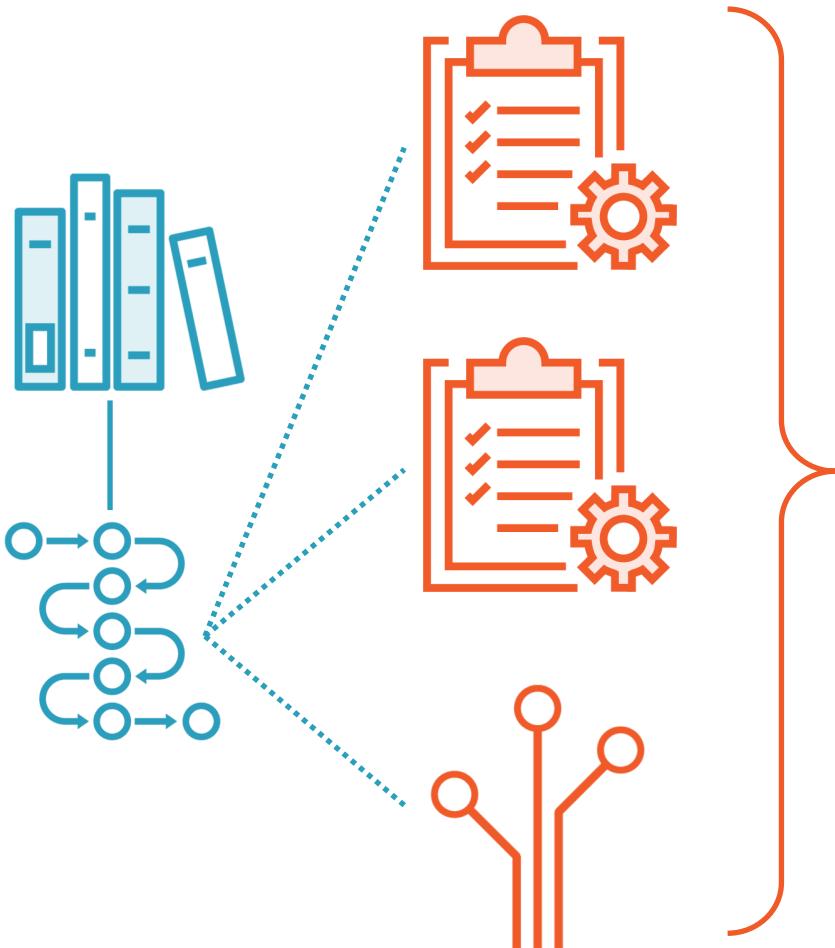
Demo



Using multi-branch pipelines

- Creating a multi-branch job
- Evolving branch builds
- Removing branches





- Linux
- Java 8
- Windows
- .NET 4.7
- Windows
- .NET 4.8

Summary



Containers as build agents

- Official or custom SDK images
- Non-Docker projects

Templating with shared pipelines

- Pipelines in shared libraries
- Abstracting complex builds
- Automation with the Job DSL

Multi-branch pipelines

- One job for branch builds
- Pipeline snapshot per branch

We're Done!



So...

- Please leave a rating
- Follow @EltonStoneman on Twitter
- Check out blog.sixeyed.com
- Watch my other courses ☺