

Creating Custom Jenkins X Pipelines



Andrew Morgan

INDEPENDENT

@mogronalol



Overview

Jenkins X pipeline overview

- Problems they solve
- YAML structure
- Build pack hierarchy

Overriding an application pipeline

Overriding an environment pipeline



Problems with Traditional CI/CD Pipelines



Not versioned



Problems with Traditional CI/CD Pipelines



Not versioned



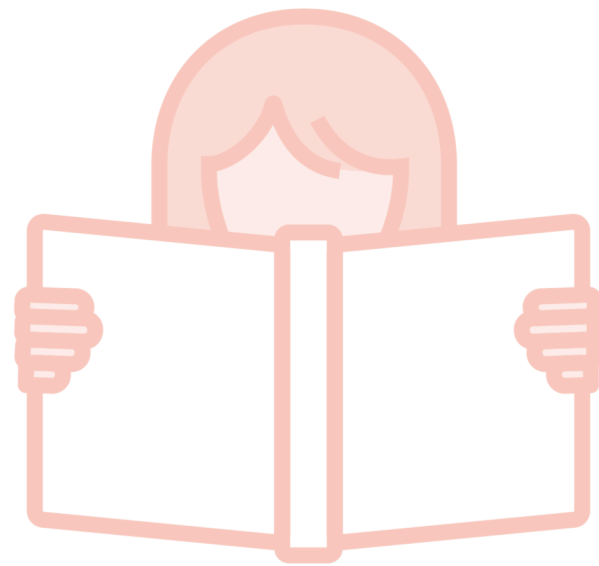
**Requires
knowledge of a
new scripting
language**



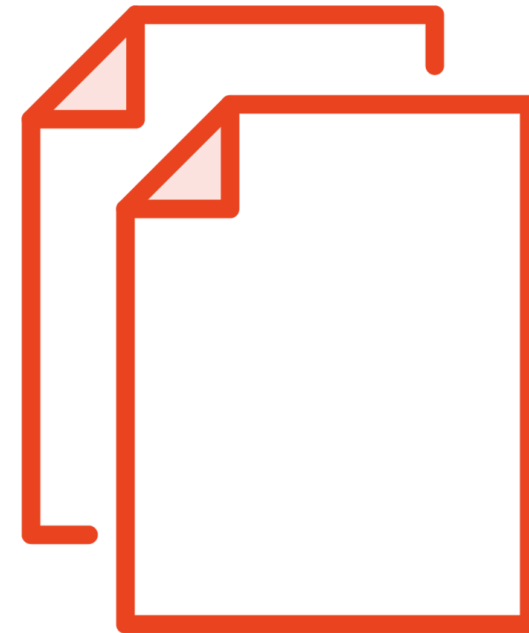
Problems with Traditional CI/CD Pipelines



Not versioned



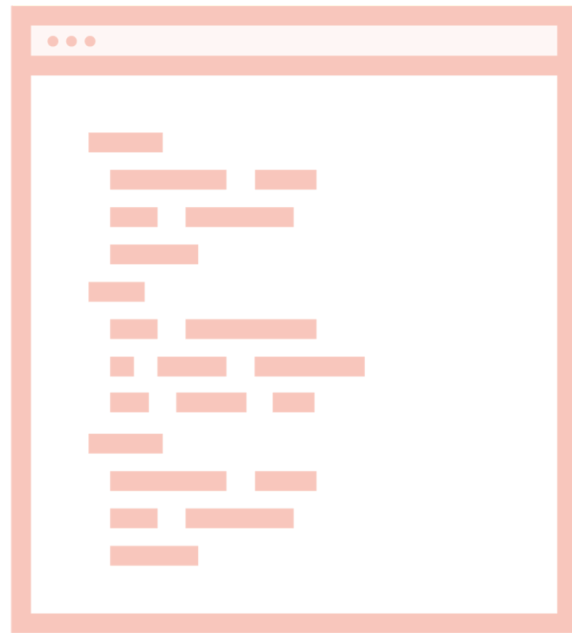
Requires
knowledge of a
new scripting
language



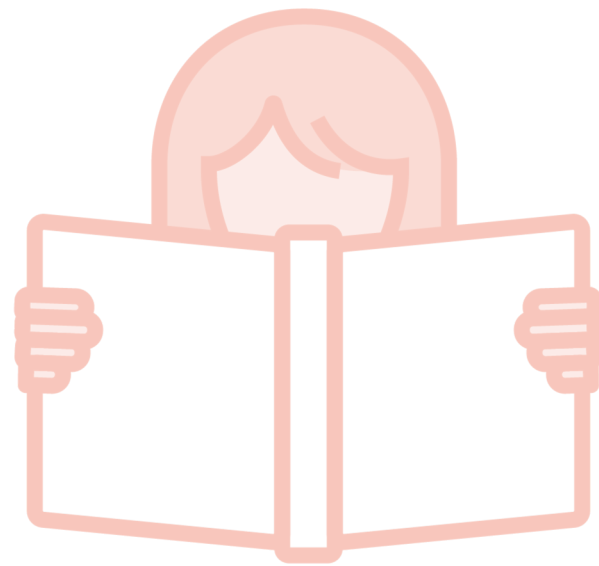
**Repeating
yourself across
projects**



Problems with Traditional CI/CD Pipelines



Not versioned



Requires
knowledge of a
new scripting
language



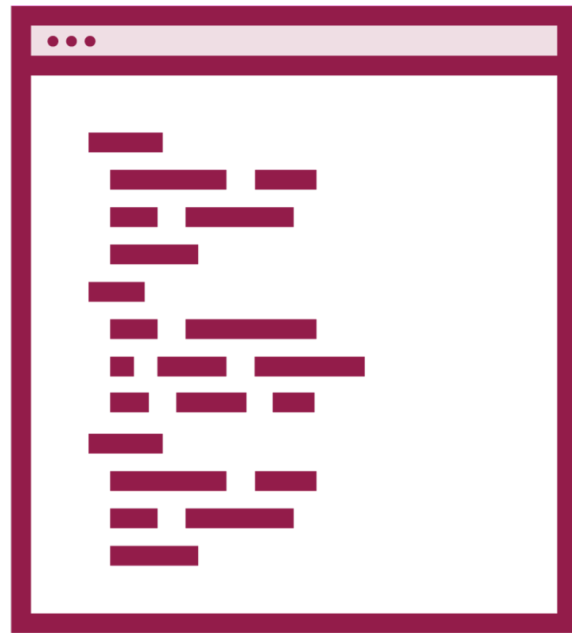
Repeating
yourself across
projects



**Depends on a
specific runtime**



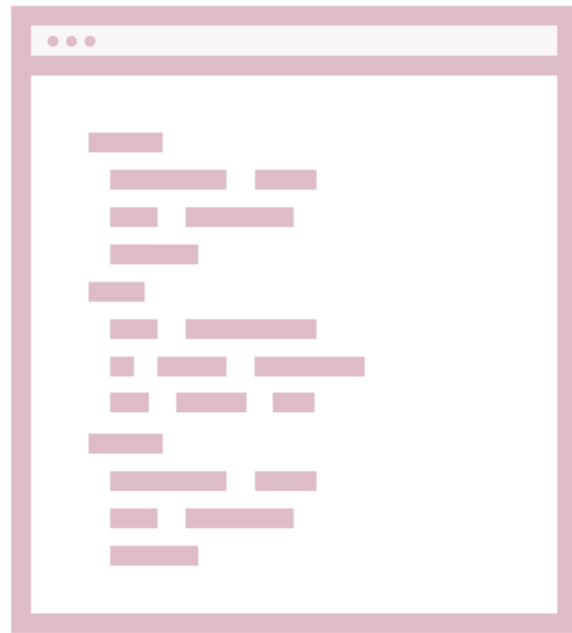
Jenkins X Pipelines Introduction



Stored in Git



Jenkins X Pipelines Introduction



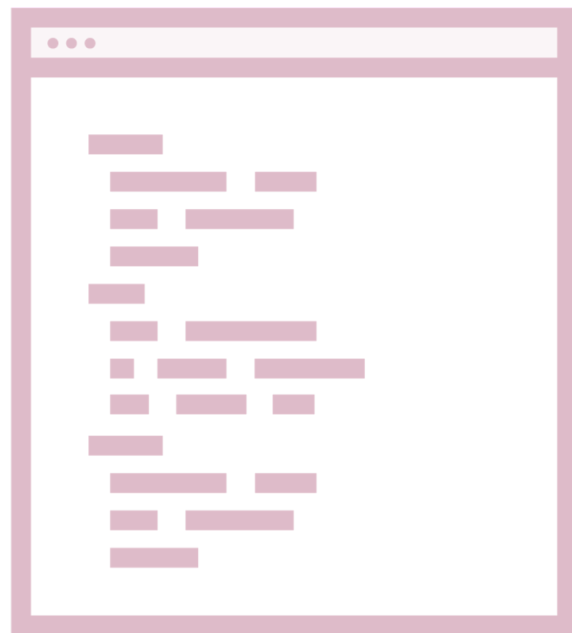
Stored in Git



**YAML file
structure with a
focus on
executables**



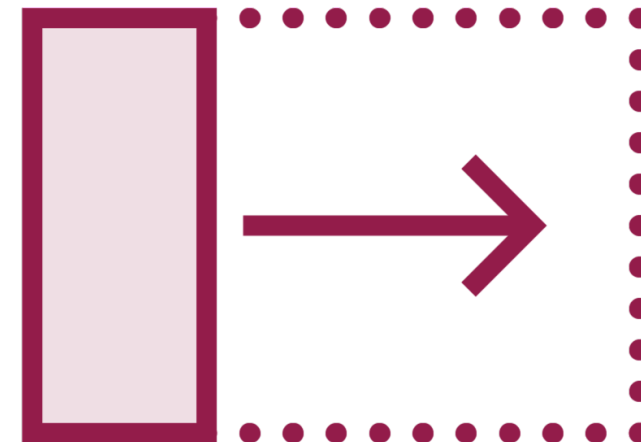
Jenkins X Pipelines Introduction



Stored in Git



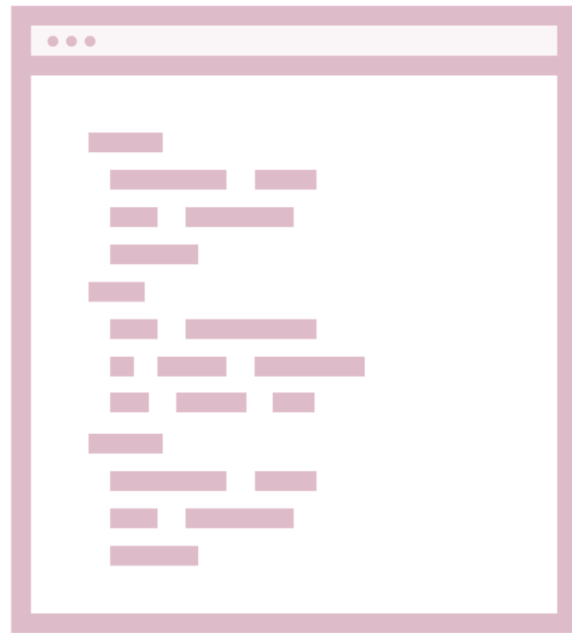
YAML file
structure with a
focus on
executables



**Supports
inheritance**



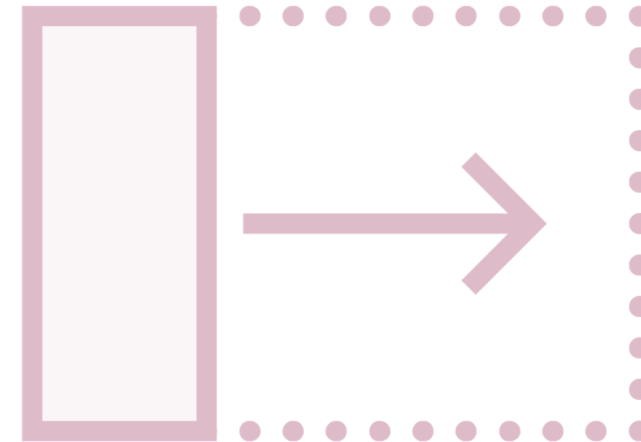
Jenkins X Pipelines Introduction



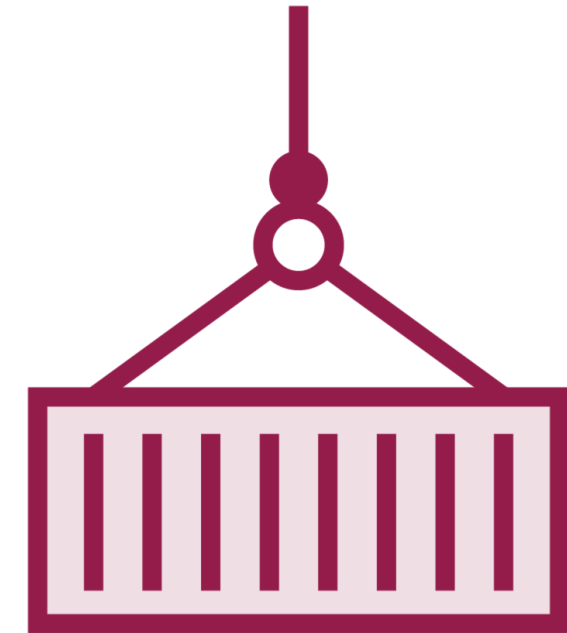
Stored in Git



YAML file
structure with a
focus on
executables



Supports
inheritance

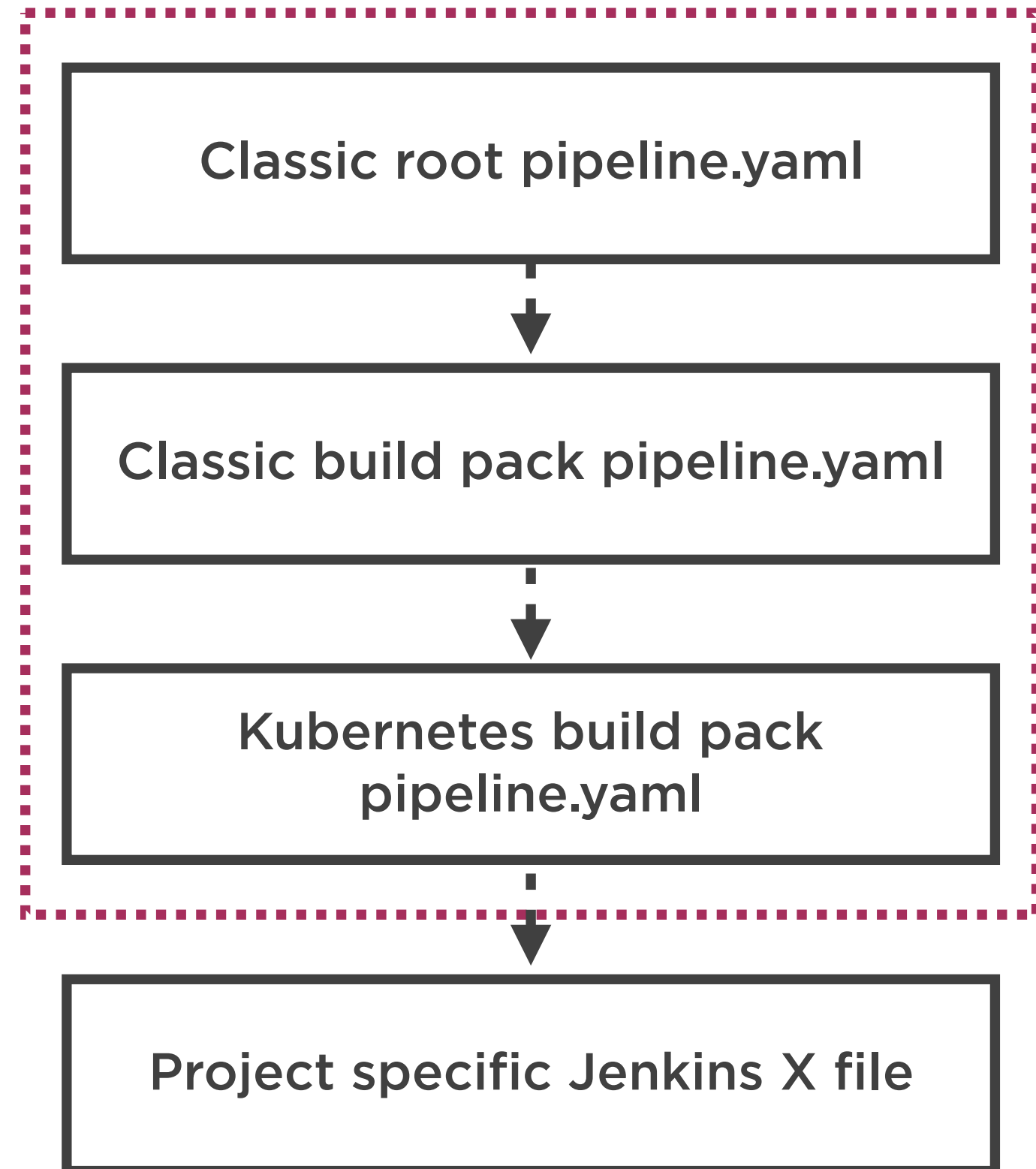


**Runs in
containers**



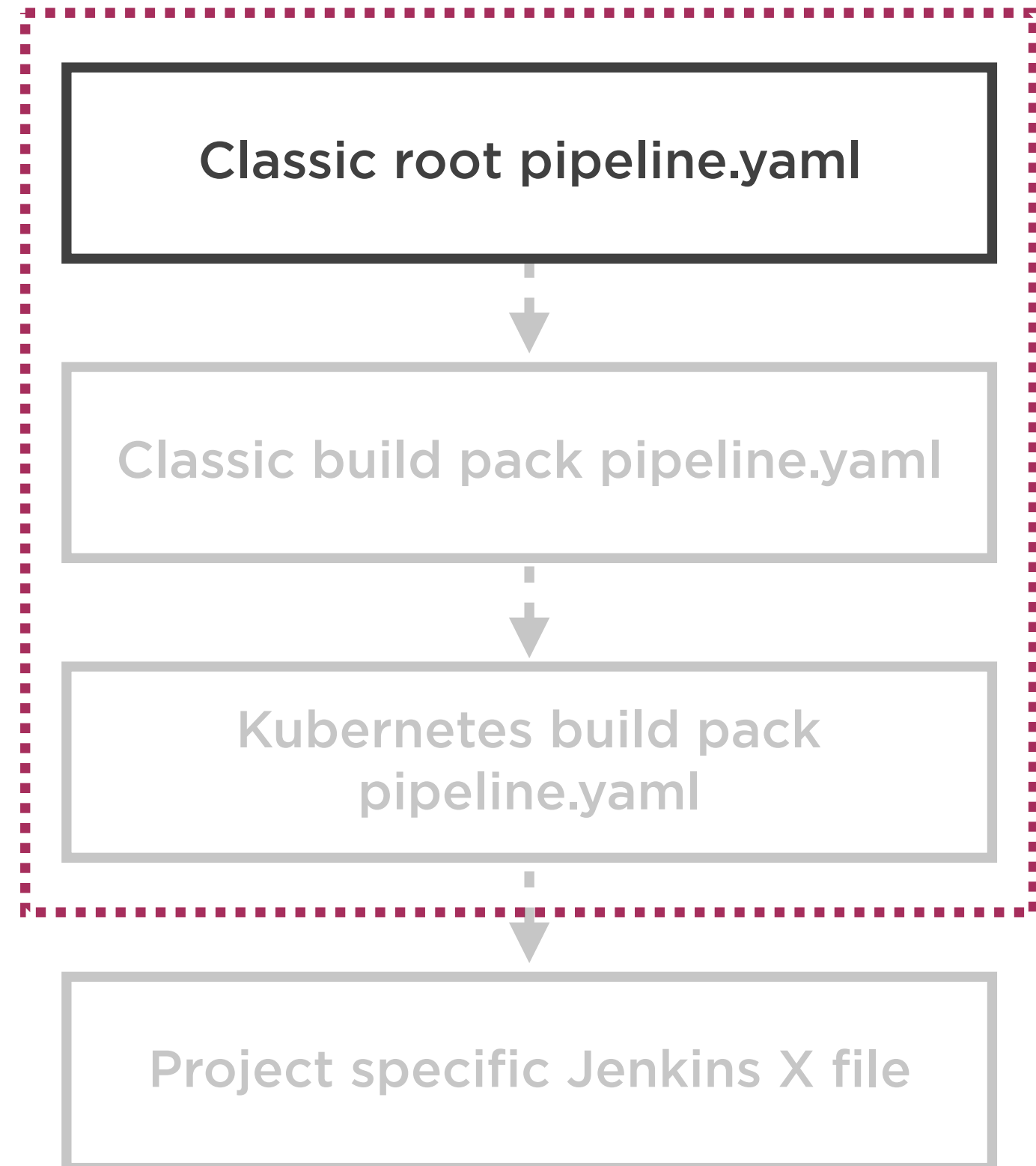
Jenkins X Pipeline Extension Model Hierarchy

Jenkins X Buildpacks



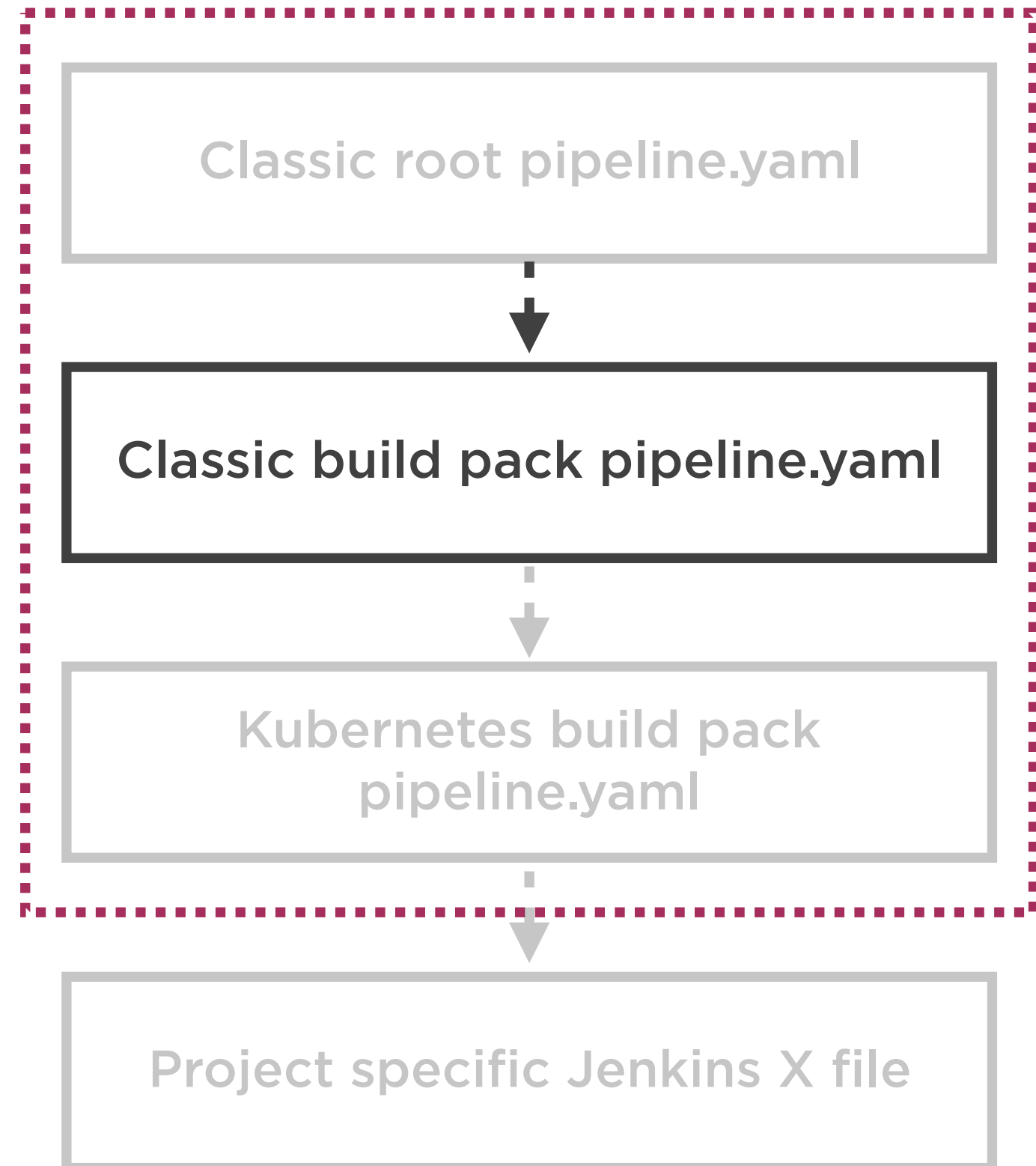
Jenkins X Pipeline Extension Model Hierarchy

Jenkins X Buildpacks



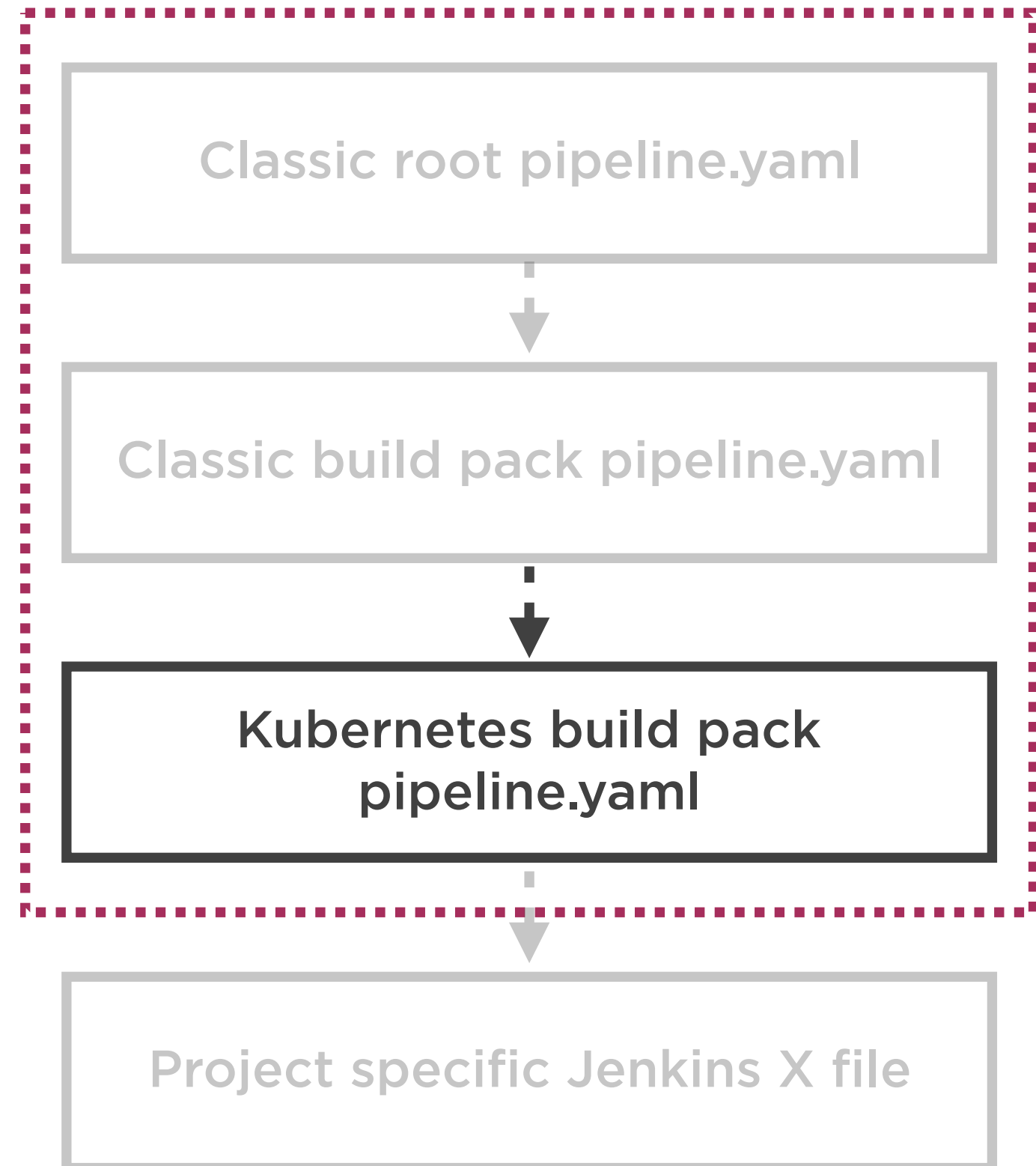
Jenkins X Pipeline Extension Model Hierarchy

Jenkins X Buildpacks



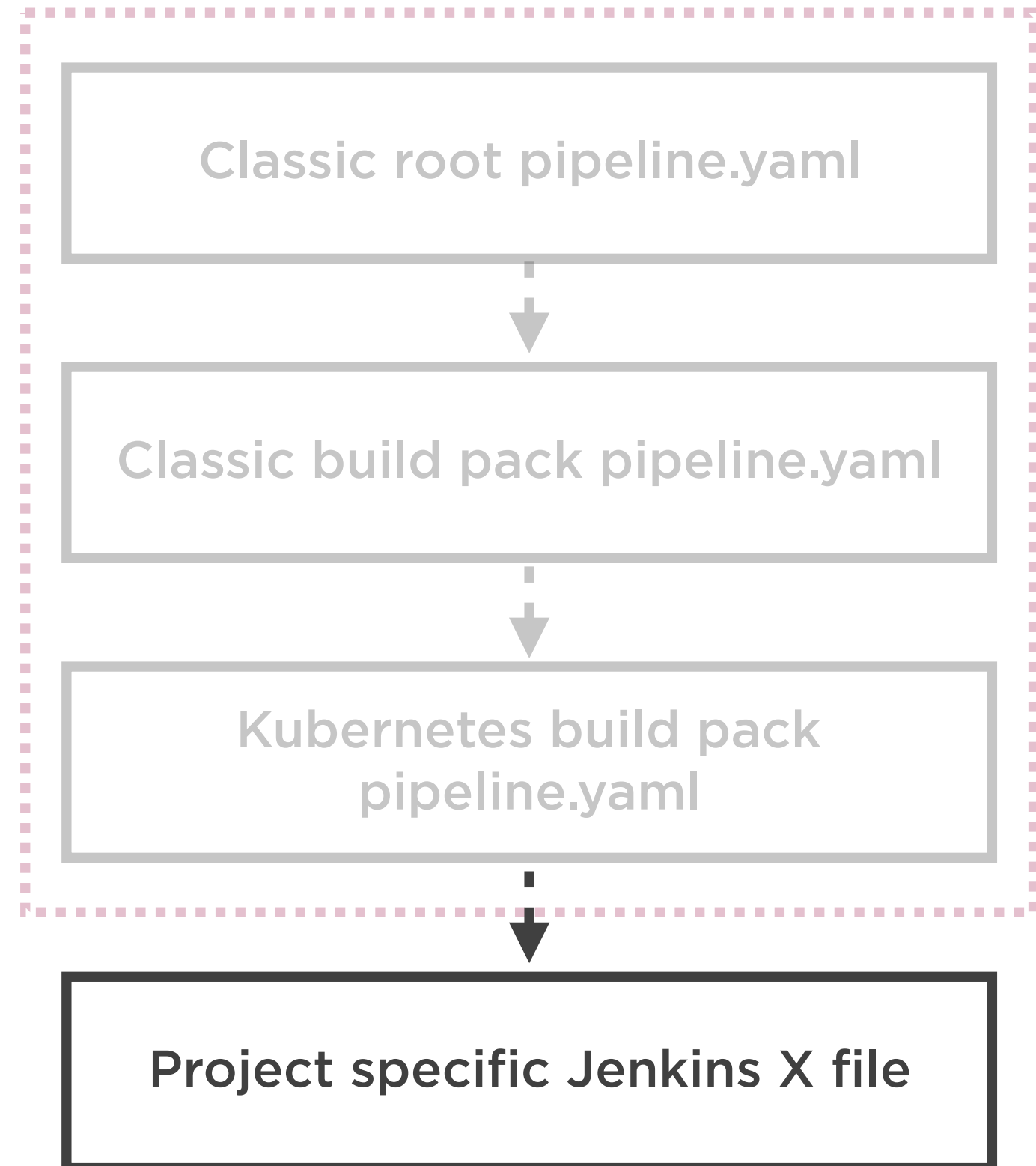
Jenkins X Pipeline Extension Model Hierarchy

Jenkins X Buildpacks



Jenkins X Pipeline Extension Model Hierarchy

Jenkins X Buildpacks



`buildPack: parent-buildpack`

◀ **Buildpack to inherit from**




```
buildPack: parent-buildpack  
pipelineConfig:
```

◀ Pipelines to override



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
  release:
  feature:
```

◀ Runs on pull request branches



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    release:
  feature:
```

◀ Run on the master branch



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
  release:
  feature:
```

◀ Merges to long running feature branches



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
    preBuild:
    build:
    postBuild:
    promote:
```

◀ Setup such as checking out code



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      preBuild:
        build:
          postBuild:
            promote:
```

◀ Steps to perform before build



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
    preBuild:
    build:
    postBuild:
    promote:
```

◀ Steps to perform during build



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
    preBuild:
    build:
    postBuild:
    promote:
```

◀ Steps to perform after build




```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
    preBuild:
    build:
    postBuild:
    promote:
```

◀ Promotions to environment



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
```

◀ List of commands to execute



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
        - name: step-first
          command: echo
          args:
            - first
        - name: step-second
          dir: /workspace/source
          command: make
          args: build
```

◀ Step name



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
        - name: step-first
          command: echo
          args:
            - first
        - name: step-second
          dir: /workspace/source
          command: make
          args: build
```

◀ Command to execute



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
        - name: step-first
          command: echo
        - args:
            - first
        - name: step-second
          dir: /workspace/source
          command: make
          args: build
```

◀ Remaining steps



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      preSteps:
        - name: step-first
          command: echo
        - args:
            - first
        - name: step-second
          dir: /workspace/source
          command: make
          args: build
```

◀ Remaining steps



```
buildPack: parent-buildpack
pipelineConfig:
  agent:
    image: "image/foo@version"
  env:
    - name: F00
      value: Some value
  containerOptions:
    resources:
      limits:
        cpu: 1
        memory: 256Mi
  pullRequest:
    setup:
      steps:
```

◀ Configure container for all pipelines



```
buildPack: parent-buildpack
pipelineConfig:
  agent:
    image: "image/foo@version"
  env:
    - name: F00
      value: Some value
  containerOptions:
    resources:
      limits:
        cpu: 1
        memory: 256Mi
  pullRequest:
    setup:
      steps:
```

◀ Set image




```
buildPack: parent-buildpack
pipelineConfig:
  agent:
    image: "image/foo@version"
  env:
    - name: F00
      value: Some value
  containerOptions:
    resources:
      limits:
        cpu: 1
        memory: 256Mi
  pullRequest:
    setup:
      steps:
```

◀ Environment variables



```
buildPack: parent-buildpack
pipelineConfig:
  agent:
    image: "image/foo@version"
  env:
    - name: F00
      value: Some value
  containerOptions:
    resources:
      limits:
        cpu: 1
        memory: 256Mi
  pullRequest:
    setup:
      steps:
```

◀ Configuration for container such as memory and CPU



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    agent:
      image: "image/foo@version"
    env:
      - name: F00
        value: Some value
    containerOptions:
      resources:
        limits:
          cpu: 1
          memory: 256Mi
  setup:
    steps:
```

◀ Applicable per pipeline



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      agent:
        image: "image/foo@version"
      env:
        - name: F00
          value: Some value
      containerOptions:
        resources:
          limits:
            cpu: 1
            memory: 256Mi
    steps:
```

◀ Applicable per lifecycle



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
        - agent:
            image: "image/foo"
          env:
            - name: F00
              value: Some value
          containerOptions:
            resources:
              limits:
                cpu: 1
                memory: 256Mi
```

◀ Applicable per step



buildPack: none

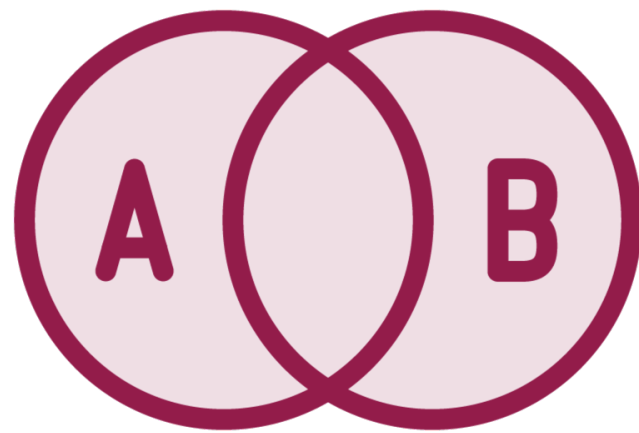
◀ **Fully custom pipeline**



A Jenkins X file contains pipelines,
each of which contains lifecycles,
each of which contains steps,
each of which execute commands



How Your Pipeline Is Generated

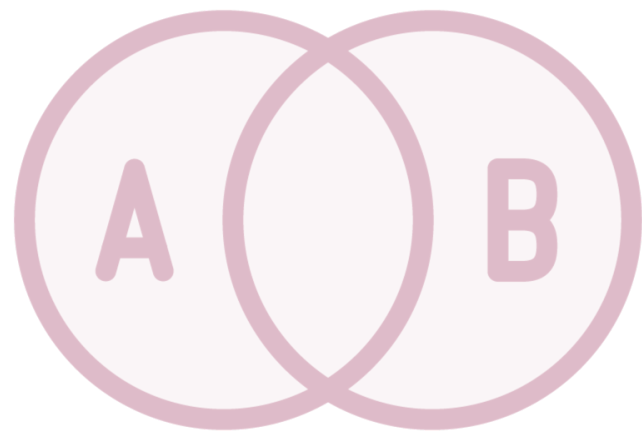


Meta Pipeline

Merges pipeline
files and creates
build pack pipeline

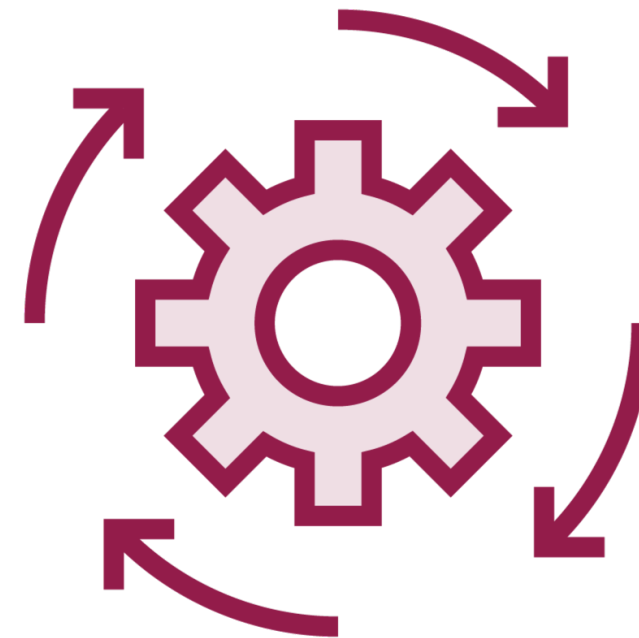


How Your Pipeline Is Generated



Meta Pipeline

Merges pipeline files and creates build pack pipeline



Build Pack Pipeline

Your actual pipeline



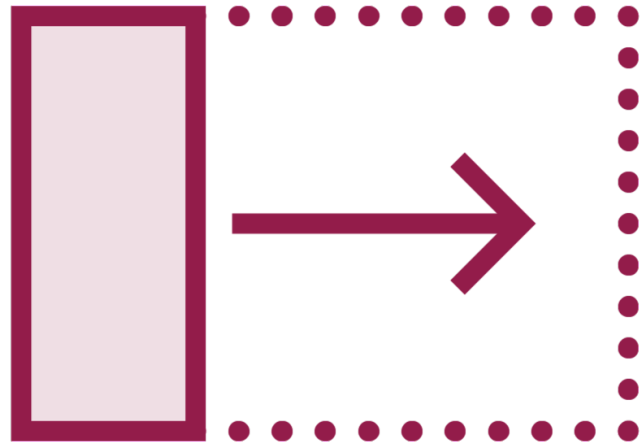
Demo

Analysis of our build pack hierarchy

**Customize an existing Jenkins X file with a
code coverage check**



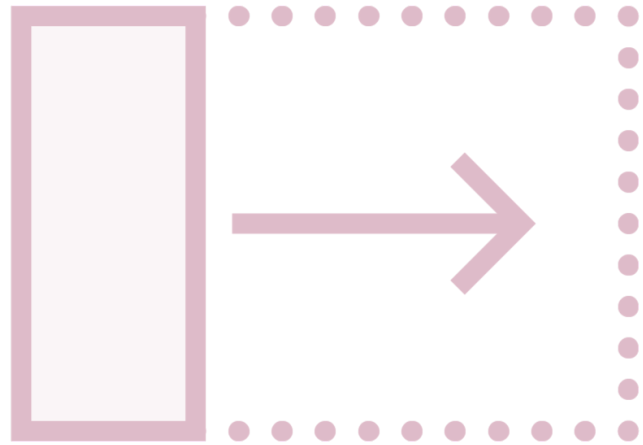
Environment Pipelines



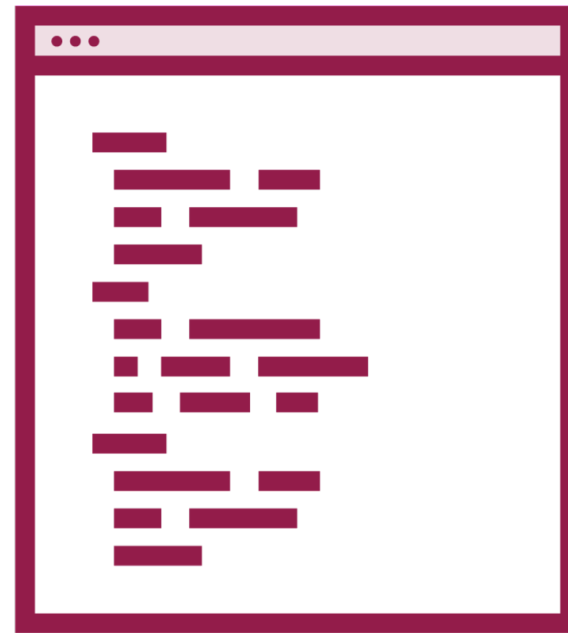
Inherit from
environment build
pack



Environment Pipelines



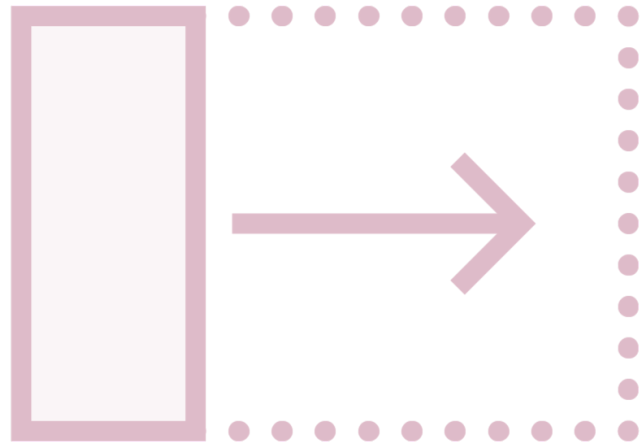
Inherit from
environment build
pack



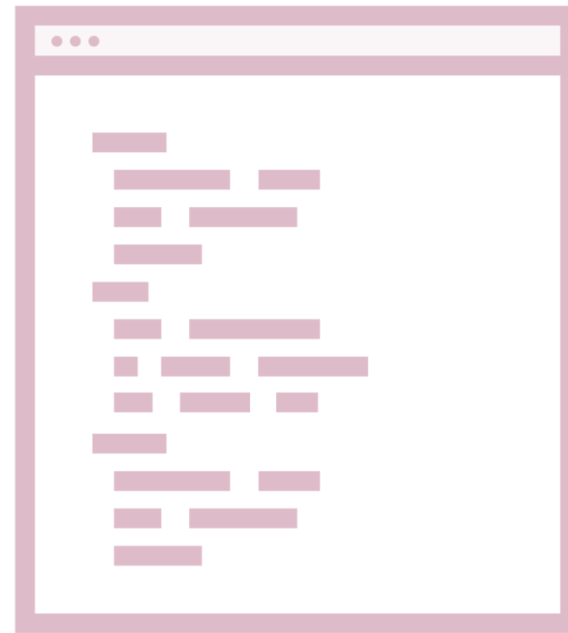
**Can be customised
just like build pack
pipelines**



Environment Pipelines



Inherit from
environment build
pack



Can be customised
just like build pack
pipelines



**A common use case
is acceptance
testing**



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
        - name: step-first
          command: echo
          args:
            - first
        - name: step-second
          command: echo
          args:
            - second
        - name: step-third
          command: echo
          args:
            - third
```



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
        - name: step-first
          command: echo
          args:
            - first
        - name: step-second
          command: echo
          args:
            - second
        - name: step-third
          command: echo
          args:
            - third
```

◀ echo first



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
        - name: step-first
          command: echo
          args:
            - first
        - name: step-second
          command: echo
          args:
            - second
        - name: step-third
          command: echo
          args:
            - third
```

◀ echo second




```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      steps:
        - name: step-first
          command: echo
          args:
            - first
        - name: step-second
          command: echo
          args:
            - second
        - name: step-third
          command: echo
          args:
            - third
```

◀ echo third



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      loop:
        variable: NUMBER
        values:
          - first
          - second
          - third
        steps:
          - name: step
            command: echo
          - args:
              - ${NUMBER}
```

◀ Allows steps to be run repeatedly in a loop



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      loop:
        variable: NUMBER
        values:
          - first
          - second
          - third
        steps:
          - name: step
            command: echo
          - args:
              - ${NUMBER}
```

- ◀ Set NUMBER environment variable for each value



```
buildPack: parent-buildpack
pipelineConfig:
  pullRequest:
    setup:
      loop:
        variable: NUMBER
        values:
          - first
          - second
          - third
        steps:
          - name: step
            command: echo
            args:
              - ${NUMBER}
```

- ◀ Repeat me multiple times, echoing the value of number each time



Demo

Analysis of our environment pipeline

Customize our pipeline to run end-to-end

Make use of loops



JX Commands Recap



```
jx step syntax validate pipeline
```

◀ Validate pipeline syntax



```
jx step syntax validate pipeline
```

```
jx step syntax effective
```

◀ Validate pipeline syntax

◀ Display full pipeline



Summary

Jenkins X pipeline definitions have a YAML structure

They make use of inheritance to reduce repetition

And are executed in containers

We can extend and override pipelines however we seem fit

