DistributeReleaseBundle

Overview

The **DistributeReleaseBundle** native step triggers the distribution of a Release Bundle to an Artifactory Edge Node. This step requires a signed release bundle and one or more distribution rules to successfully execute.

Page Contents

YAML Schema

The YAML schema for DistributeReleaseBundle native step is as follows:

DistributeReleaseBundle

```
pipelines:
  - name:    <string>
    steps:
      - name: my_distribute
        type: DistributeReleaseBundle
        configuration:
                #inherits all the tags from bash; https://www.jfrog.com/confluence/display/JFROG/Bash
          dryRun: <boolean>                # optional
          inputResources:
            - name: my_releaseBundle      # one ReleaseBundle is required
              trigger: false
            - name: my_distributionRule   # one DistributionRule is required
              trigger: false    # default true
          outputResources:
            - name: my_releaseBundleOutput # one ReleaseBundle is optional

        execution:
          onStart:
            - echo "Preparing for work..."
          onSuccess:
            - echo "Job well done!"
          onFailure:
            - echo "uh oh, something went wrong"
          onComplete: #always
            - echo "Cleaning up some stuff"
```

Tags

name

An alphanumeric string (underscores are permitted) that identifies the step.

type

Must be `DistributeReleaseBundle` for this step type.

configuration

Specifies all configuration selections for the step's execution environment. This step inherits the Bash/PowerShell step configuration tags, including these pertinent tags:

Tag	Description of usage	Required /Optional
`inputResources`	Must specify a ReleaseBundle resource and one DistributionRule resource.	Required
`outputResources`	May specify a ReleaseBundle resource to be updated with the `name` and `version` of the input ReleaseBundle.	Optional

In addition, these tags can be defined to support the step's native operation:

> **Tags derived from Bash**
>
> All native steps derive from the Bash step. This means that all steps share the same base set of tags from Bash, while native steps have their own additional tags as well that support the step's particular function. So it's important to be familiar with the Bash step definition, since it's the core of the definition of all other steps.

Tag	Description of usage	Required /Optional
`dryRun`	Controls whether this should be a dry run to test if the release bundle can distribute to the Edge nodes matching the distribution rule. The default is true.	Optional

execution

Declares collections of shell command sequences to perform for pre- and post-execution phases:

Tag	Description of usage	Required/Optional
`onStart`	Commands to execute in advance of the native operation	Optional
`onSuccess`	Commands to execute on successful completion	Optional
`onFailure`	Commands to execute on failed completion	Optional
`onComplete`	Commands to execute on any completion	Optional

The actions performed for the `onExecute` phase are inherent to this step type and may not be overridden.

Examples

The following examples show how to configure a DistributeReleaseBundle step to distribute or for a dry run.

Distribute Input Release Bundle Edge Node

Distributes the input release bundle to the edge nodes defined in the distribution rule.

- This example requires an Artifactory Integration and a Distribution Integration.
- The Pipelines DSL for this example is available in this repository in the JFrog GitHub account.

DistributeReleaseBundle

```
template: true   # required for local templates
```

```
valuesFilePath: ./values.yml

resources:
  # Build info of first build to bundle
  - name: gosvc_promoted_build_info
    type: BuildInfo
    configuration:
      sourceArtifactory: {{ .Values.myArtifactoryIntegration }}
      buildName: svc_build
      buildNumber: 1

  # Build info of second build to bundle
  - name: appl_promoted_build_info
    type: BuildInfo
    configuration:
      sourceArtifactory: {{ .Values.demoArtifactoryIntegration }}
      buildName: backend_build
      buildNumber: 1

  # Release bundle
  - name: release_bundle
    type: ReleaseBundle
    configuration:
      sourceDistribution: {{ .Values.distributionIntegration }}
      name: demo_rb
      version: v1.0.0

  # Signed version of the same release bundle
  - name: signed_bundle
    type: ReleaseBundle
    configuration:
      sourceDistribution: {{ .Values.distributionIntegration }}
      name: demo_rb
      version: v1.0.0

  # Distribution rules
  - name: distribution_rules
    type: DistributionRule
    configuration:
      sourceDistribution: {{ .Values.distributionIntegration }}
      serviceName: "*"
      siteName: "*"
      cityName: "*"
      countryCodes:
        - "CN"
        - "GB"

pipelines:
  - name: demo_release_mgmt
    steps:
      - name: bundle
        type: CreateReleaseBundle
        configuration:
          releaseBundleName: demo_rb
          releaseBundleVersion: v1.0.${run_number}
          dryRun: false
          sign: false
          description: "some random test description"
          inputResources:
            - name: gosvc_promoted_build_info
              trigger: true
            - name: appl_promoted_build_info
              trigger: true
          outputResources:
            - name: release_bundle
          releaseNotes:
            syntax: markdown
            content: |
              ## Heading
                * Bullet
                * Points
```

```
    - name: sign
      type: SignReleaseBundle
      configuration:
        inputResources:
          - name: release_bundle
        outputResources:
          - name: signed_bundle

    - name: distribute
      type: DistributeReleaseBundle
      configuration:
        dryRun: false
        inputResources:
          - name: signed_bundle
          - name: distribution_rules
```

Trigger a Dry Run

Triggers a dry run of the distribution.

DistributeReleaseBundle

```
pipelines:
  - name: distributeReleaseBundlePipeline
    steps:
      - name: distributeReleaseBundleDryRun
        type: DistributeReleaseBundle
        configuration:
          dryRun: true
          inputResources:
            - name: myReleaseBundle
            - name: myInputDistributionRule
```

Triggers Distribution and Updates Output Resource

Triggers a distribution and updates the output resource with the name and version of the input.

DistributeReleaseBundle

```
pipelines:
  - name: distributeReleaseBundlePipeline
    steps:
      - name: distributeReleaseBundleDryRun
        type: DistributeReleaseBundle
        configuration:
          dryRun: false
          inputResources:
            - name: myReleaseBundle
            - name: myInputDistributionRule
          outputResources:
            - name: myOutputReleaseBundle
```

Same Step for Dry Runs and to Distribute

In this example, the same step is used for both dry runs and to distribute the release bundle to Edge Nodes. The dry_run variable may be set in the pipeline configuration section or step configuration or added as a run variable by an earlier step in the pipeline using add_run_variable.

DistributeReleaseBundle

```
pipelines:
  - name: distributeReleaseBundlePipeline
    steps:
      - name: distributeReleaseBundleStep
        type: DistributeReleaseBundle
        configuration:
          dryRun: ${dry_run}
          inputResources:
            - name: myReleaseBundle
            - name: myInputDistributionRule
```

How it Works

When you use the **DistributeReleaseBundle** native step in a pipeline, it performs the following functions in the background:

- Create the distribution payload (the JSON object that will be in the request to Distribution)
- curl $distributionUrl/api/v1/distribution/$releaseBundleName/$releaseBundleVersion (send the distribution or dry run payload to Distribution)
- curl $distributionUrl/api/v1/release_bundle/$releaseBundleName/$releaseBundleVersion/distribution/$trackerId (if not a dry run, using the tracker ID returned by Distribution, check if the distribution is complete)
- write_output (update the output ReleaseBundle resource)