

# AI/ML Coding Round

## Problem:

Train a LLM that can answer queries about JFrog Pipelines' [native steps](https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-steps). When posed with a question like "How do I upload an artifact?" or "What step should I use for an Xray scan?", the model should list the appropriate native step(s) and provide an associated YAML for that step.

## Requirements

1. Data Collection: Acquire publicly available information on Native Steps from JFrog's website that contain information on native steps for building pipelines. Data that is not publicly accessible falls outside the scope of this coding challenge. (<https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-steps>)
2. Data Preprocessing: Process the text to make it suitable for training. This might involve tokenization, stemming, and other NLP techniques.
3. Model Training: Train a LLM on the (preprocessed) dataset. You can choose one of the freely available open source model like BERT or any other model available
4. Query Handling: Implement a function that takes a user query as input and returns the appropriate native step(s) and a sample YAML configuration.
5. YAML Generation: Implement a function that can generate a sample YAML configuration based on the identified native step(s).

---

@author - Midhun Kumar  
@email - [midhunkumar04@gmail.com](mailto:midhunkumar04@gmail.com)

---

## 1. Data scraping and preparation

### Data Understanding

1. I have started exploring the documentation sites given in the problem and noted the below points.
  - The site has multiple navigation links for various pipeline procedures.
  - Each discipline procedure has two main things, like instructions and yaml pipeline examples we can use them for training
  - If we can extract these links, we can use them to scrape the data from each document and use them for preparation.

## 1.1 Link Extraction

- Extracting the pipeline documentation links from following site

<https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-steps>

```
In [ ]: # importing libraries for link extraction
from bs4 import BeautifulSoup as soup
import requests
from pprint import pprint
```

```
In [ ]: # Link extraction function
def linkExtarction(url:str) -> list:
    """
    This function takes url:str as input and retrun list of link in the s
    """
    result = requests.get(url)
    scrapedData = soup(result.content)
    links = scrapedData.select('a')
    return [link['href'] for link in links]
```

```
In [ ]: # verifying the links
url = 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-s
links = linkExtarction(url)
pprint(links)
```

```
['https://jfrog.com/help/r/jfrog-pipelines-documentation',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/jfrog-pipelines',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-use-cas
es',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-concept
s',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-step-by
-step',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/see-it-live',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-quickst
art',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
hello-world',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
docker-build-and-push',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
release-to-edge-node',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
go-build',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
npm-build',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
maven-build',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
helm-blue-green-deploy',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/configuring-pipel
```

ines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-s-integrations',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-sources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-source-sync-recovery',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-s-node-pools',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-custom-vm-images',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-dynamic-nodes-on-kubernetes',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-s-static-nodes',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/sending-pipelines-nodes-agent-logs-to-logstash',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-pipeline-s',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/defining-a-pipeline',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-integrations',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/airbrake-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/artifactory-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/aws-keys-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/azure-keys-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/digital-ocean-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/bitbucket-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/bitbucket-server-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/distribution-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/docker-registry-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/file-server-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/generic-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/github-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/github-enterprise-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gitlab-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/google-cloud-integration',

'https://jfrog.com/help/r/jfrog-pipelines-documentation/jenkins-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jenkins-server-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jfrog-platform-access-token-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jira-integration',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/incoming-webhook-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/kubernetes-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/newrelic-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/outgoing-webhook-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pagerduty-events-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pem-key-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/slack-integration',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/smtp-credentials-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/ssh-key-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/viewing-resources-in-the-ui',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/advanced-usage-of-resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/aql',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/artifact',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/buildinfo',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/crontrigger',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/distributionrule',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/filespec',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gitrepo',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmchart',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/image',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/incomingwebhook',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/outgoingwebhook',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/propertybag',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/releasebundle',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/remotefile',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/vmcluster',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-steps',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/bash',

'https://jfrog.com/help/r/jfrog-pipelines-documentation/createreleasebundle',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/distributereleasebundle',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/dockerpush',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/dockerbuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gobuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gopublishbinary',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gopublishmodule',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gradlebuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmbbluegreencleanup',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmbbluegreendeploy',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmbbluegreenrole switch',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmdeploy',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmpublish',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jenkins',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/linuxvmdeploy',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/matrix',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/mvnbuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/npmbuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/npmpublish',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/prematrix',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/postmatrix',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/powershell',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/promotebuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/publishbuildinfo',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/signreleasebundle',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/triggerpipeline',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/uploadartifact',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/xrayscan',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-utility-functions',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/bump\_semver',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/replace\_envs',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/retry\_command',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get\_uuid',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/save\_artifact\_info',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/validate\_artifact',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/configure\_jfrog\_cli',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/use\_jfrog\_cli',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/check\_xray\_available',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/cleanup\_jfrog\_cli',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/set\_trigger\_payload'

ad',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get\_trigger\_paylo  
ad',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/end\_step',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/update\_run\_descri  
ption',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/set\_run\_name',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/source-control',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/compare\_git',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/update\_commit\_sta  
tus',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/test-reports',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/encryption',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/encrypt\_string',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/decrypt\_string',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/encrypt\_file',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/decrypt\_file',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/notifications',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/json',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/set\_payload',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/read\_json',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/replicate\_resourc  
e',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/write\_output',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/caching',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_cache\_files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/restore\_cache\_fil  
es',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/run-state-managem  
ent',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_run\_variables  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/export\_run\_variab  
les',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_run\_files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/restore\_run\_files  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/affinity-group-st  
ate-management',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_affinity\_grou  
p\_files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/restore\_affinity\_  
group\_files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-state-ma  
nagement',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_pipeline\_vari  
ables',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/export\_pipeline\_v  
ariables',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_pipeline\_file  
s',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/restore\_pipeline\_

```
files',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/step-properties',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/find_resource_variable',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get_integration_name',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get_resource_name',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get_resource_names',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get_affinity_group_step_names',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/find_step_configuration_value',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-environment-variables',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-templates',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/global-templates',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helloworld',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/goci',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gradleci',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/mavenci',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/npmci',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/promoteci',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/dockerbuildandpush',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmpublishanddeploy',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/artifactory-artifact-cleanup-template-1.40-and-higher',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-global-templates-to-create-a-pipeline-source-1.31-and-higher',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-global-templates-to-create-a-pipeline-source-1.30-and-lower',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/system-templates',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/local-templates',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-templates',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-tasks',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/build-publish',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/curate-artifact',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/email-notify',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/frogbot',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.1.0',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.0.3',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/go-build',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/go-security-check',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/go-static-check',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.1.0',
```

'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.0.2',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gradle-build',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jira-comment',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/mvn-build',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/npm-build',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/run-cypress',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-go',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.1.0',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.0.3',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-java',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-jfrog-cli',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-node',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-yarn',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/slack-notify',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/upload-file-to-s3',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/extending-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extension-expressions',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extension-resource-model',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extension-step-model',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extension-usage',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extensions-tutorial',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-extensions',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/working-with-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-history',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-run-logs',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/running-a-pipeline',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/breaking-your-pipelines-into-steps',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/triggering-pipelines-and-steps',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-stateful-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/inserting-secrets-in-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/running-multiple-steps-on-the-same-build-node',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/sending-notifications-from-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-jira-issues-from-pipelines',



'https://jfrog.com/help/r/jfrog-pipelines-documentation/sending-build-status-to-source-control',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/recording-unit-test-results',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/deploying-to-kubernetes-in-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setting-up-maven-builds-for-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/caching-step-runtimes',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-multibranch-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-jenkins-with-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-the-matrix-step',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setting-retention-policy',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/signed-pipelines',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setting-step-timeouts',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/speeding-up-pipeline-runs',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/triggering-a-run-with-custom-parameters',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/adding-build-badges',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/conditional-workflows',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/approval-gates',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-of-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/embedded-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/monitoring-pipelines-using-new-relic-apm',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-runtimes',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/choosing-your-runtime-image',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/running-steps-on-the-host',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/choosing-node-pools',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/runtime-images',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/t~ovaki5HJDlhSI8nUZZow',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/cvMvKDVfYgYz9CAEi5V0DQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/JuhAfy2o9I7WDuzDyPjI~w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/1yeQM4x8WIlvqW5PjuzmRA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/pqNZFdHirwvJ8kqjZPY46w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/CMyzVpgNkZRhMCAGBV3WUQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/CBbQUEleX0AdEK~sv3QSIg',

'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/ahbSstb2sYifVdbqlJb6pA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eeCPdY\_CZTb0tLPJzrgVaw',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/E5HPgKlVIduhL9KURYCrXg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/Ywv9eZVG2Q4SzRQ\_N8wxw',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/f39y1WcXkTQLusVmwu52uA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/OU140BJ\_4faErvHbd2e2Bg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/X1WU\_5tNPs67bGBUBcT8rw',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/ULtmnqkvN~70S8gReJZ55w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/P1nvfSf9oV0MY3jJGqMB\_g',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/s699hxjWCdbiUVf3JZ4NLA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/MD44zUcLMbxgNISASufU6w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/gvScchJe4eqc1uM4mf07Ew',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/50jvui6F1kvxnuzQWGAX\_A',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/440kpN\_tMlW1tYfk2qwhBg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/kCuBcVfDeWF6DgSXQWFLjA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/a0osLRoIFNB4e8I37qWbEQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/qjLtCJG9k5z3SIVWUyEMBQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/kohJX4jGwXj9k60unPZ2wA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/l~EJ8fnn2GqxVYHQ4FL1Cg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/IHsC7xFESYudBr8ySwl9\_w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/IU0nljLIKRF1wU0XmwAZnQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/9bmEZe6zWZ9mtGP6qMkEvA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/F8LwLNYebdJmDfy4L1jpNw',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/CzSwCXjPoLMY3gwig81pog',  
'https://pipelines.jfrog.io/ui/pipelines/myPipelines/viewPipelines',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_Generic',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/JuhAfy2o9I7WDuzDyPjI~w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/qjLtCJG9k5z3SIVWUyEMBQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-nativeN  
ativeSteps',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/JuhAfy2o9I7WDuzDyPjI~w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/gvScchJe4eqc1uM4mf07Ew',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/a0osLRoIFNB4e8I37qWbEQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/qjLtCJG9k5z3SIVWUyEMBQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/kohJX4jGwXj9k60unPZ2wA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/1yeQM4x8WIlvqW5PjuzmRA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/pqNZFdHirwvJ8kqjZPY46w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/CMyzVpgNkZRhMCAGBV3WUQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/CBbQUEleX0AdEK~sv3QSIg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/ahbSstb2sYifVdbqlJb6pA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eeCPdY\_CZTb0tLPJzrgVaw',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/E5HPgKlVIduhL9KURYCrXg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/ahbSstb2sYifVdbqlJb6pA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/Ywv9eZVG2Q4SzRQ\_N8wxw',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/f39y1WcXkTQLusVmwu52uA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/OU140BJ\_4faErvHbd2e2Bg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/X1WU\_5tNPs67bGBUBcT8rw',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/ULtmnqkvN~70S8gReJZ55w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/P1nvfSf9oV0MY3jJGqMB\_g',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/s699hxjWCdbiUVf3JZ4NLA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/50jvui6F1kvxnuzQWGAX\_A',

'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/440kpN\_tMlW1tYfk2qwhBg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/kCuBcVfDeWF6DgSXQWFLjA',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/l~EJ8fnn2GqxVYHQ4FL1Cg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/IHsC7xFESYudBr8ySwl9\_w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/IU0nljLIKRF1wU0XmWAZnQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/CzSwCXjPoLMY3gwig81pog',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-stepsGe  
nericandNativeSteps',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-Environ  
mentvariables',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-Runtime  
,  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-NodePoo  
lAssignment',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-Affinit  
yGroup',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-Queuing  
Priority',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-Timeout  
Duration',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-Integra  
tions',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-InputsT  
riggers',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/eJJdeVT9Cq2zK4TzJpEa0g?s  
ection=UUID-c9f2a58e-267d-94f7-b9df-1458c35267da\_id\_PipelinesSteps-Outputs  
,  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/7yJX~hEQVTb4T0BzmNbSVQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/CMyzVpgNkZRhMCAGBV3WUQ',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/CBbQUEleX0AdEK~sv3QSIg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/WSVzs0FbTFmdWbWseh45\_w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/UBYixse92Dy~zOMXKvQkVA',  
'urn:resource:component:362369',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/bYG9ddsey0Deno5UB6m1Qg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/JuhAfy2o9I7WDuzDyPjI~w',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/NDuFP707YmF1UrYCIbvUHg',  
'https://jfrog.com/help/r/2zMh0nF0k9FPLZNgd37bwg/PfQ\_oIQbrM7fhWxR1Hmstw',  
'https://jfrog.com/help/' ]

## Remarks:

- From the above results, we can see we have extracted all the links from the site.
- We can also observe that many links do not have proper naming conventions, so we are going to focus on only the links with proper naming so that we can use them to label the data.

```
In [ ]: '''
Filtering only required links
'''
import re # importing regex for string matching

def filter_links(links:list)->list:
    '''
    This function will take list of links and return filter list based on
    '''
    filterd_links = []
    for link in links:
        matchString = 'https://jfrog.com/help/r/jfrog-pipelines-documenta
        if re.search(matchString,link):
            filterd_links.append(link)
    return filterd_links
```

```
In [ ]: pprint(filter_links(links))

['https://jfrog.com/help/r/jfrog-pipelines-documentation/jfrog-pipelines',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-use-cas
es',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-concept
s',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-step-by
-step',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/see-it-live',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-quickst
art',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
hello-world',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
docker-build-and-push',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
release-to-edge-node',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
go-build',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
npm-build',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
maven-build',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-example-
helm-blue-green-deploy',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/configuring-pipel
ines',
 'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline
```

s-integrations',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-sources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-source-sync-recovery',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-s-node-pools',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-custom-vm-images',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-dynamic-nodes-on-kubernetes',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-s-static-nodes',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/sending-pipelines-nodes-agent-logs-to-logstash',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-pipeline-s',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/defining-a-pipeline',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-integrations',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/airbrake-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/artifactory-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/aws-keys-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/azure-keys-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/digital-ocean-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/bitbucket-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/bitbucket-server-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/distribution-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/docker-registry-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/file-server-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/generic-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/github-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/github-enterprise-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gitlab-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/google-cloud-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jenkins-integration',

'https://jfrog.com/help/r/jfrog-pipelines-documentation/jenkins-server-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jfrog-platform-access-token-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jira-integration',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/incoming-webhook-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/kubernetes-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/newrelic-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/outgoing-webhook-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pagerduty-events-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pem-key-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/slack-integration',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/smtp-credentials-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/ssh-key-integration',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/viewing-resources-in-the-ui',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/advanced-usage-of-resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/aql',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/artifact',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/buildinfo',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/crontrigger',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/distributionrule',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/filespec',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gitrepo',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmchart',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/image',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/incomingwebhook',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/outgoingwebhook',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/propertybag',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/releasebundle',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/remotefile',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/vmcluster',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-steps',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/bash',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/createreleasebundle',

'https://jfrog.com/help/r/jfrog-pipelines-documentation/distributerelease  
bundle',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/dockerpush',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/dockerbuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gobuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gopublishbinary',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gopublishmodule',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gradlebuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmbluegreenclea  
nup',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmbluegreendepl  
oy',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmbluegreenrole  
switch',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmdeploy',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helmpublish',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/jenkins',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/linuxvmdeploy',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/matrix',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/mvnbuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/npmbuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/npmublish',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/prematrix',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/postmatrix',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/powershell',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/promotebuild',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/publishbuildinfo'  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/signreleasebundle'  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/triggerpipeline',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/uploadartifact',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/xrayscan',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-utility  
-functions',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/bump\_semver',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/replace\_envs',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/retry\_command',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get\_uuid',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/save\_artifact\_inf  
o',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/validate\_artifact'  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/configure\_jfrog\_cli',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/use\_jfrog\_cli',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/check\_xray\_availa  
ble',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/cleanup\_jfrog\_cli'  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/set\_trigger\_paylo  
ad',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get\_trigger\_paylo

ad',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/end\_step',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/update\_run\_descri  
ption',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/set\_run\_name',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/source-control',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/compare\_git',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/update\_commit\_sta  
tus',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/test-reports',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/encryption',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/encrypt\_string',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/decrypt\_string',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/encrypt\_file',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/decrypt\_file',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/notifications',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/json',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/set\_payload',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/read\_json',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/resources',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/replicate\_resourc  
e',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/write\_output',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/caching',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_cache\_files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/restore\_cache\_fil  
es',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/run-state-managem  
ent',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_run\_variables  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/export\_run\_variab  
les',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_run\_files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/restore\_run\_files  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/affinity-group-st  
ate-management',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_affinity\_grou  
p\_files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/restore\_affinity\_  
group\_files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-state-ma  
nagement',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_pipeline\_vari  
ables',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/export\_pipeline\_v  
ariables',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/add\_pipeline\_file  
s',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/restore\_pipeline\_  
files',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/step-properties',



'https://jfrog.com/help/r/jfrog-pipelines-documentation/find\_resource\_variable',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get\_integration\_name',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get\_resource\_name',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get\_resource\_names',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/get\_affinity\_group\_step\_names',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/find\_step\_configuration\_value',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-environment-variables',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-templates',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/global-templates',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helloworld',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/goci',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gradleci',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/mavenci',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/npmci',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/promoteci',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/dockerbuildandpush',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/helpublishanddeploy',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/artifactory-artifact-cleanup-template-1.40-and-higher',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-global-templates-to-create-a-pipeline-source-1.31-and-higher',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-global-templates-to-create-a-pipeline-source-1.30-and-lower',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/system-templates',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/local-templates',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-templates',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-tasks',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/build-publish',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/curate-artifact',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/email-notify',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/frogbot',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.1.0',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.0.3',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/go-build',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/go-security-check',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/go-static-check',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.1.0',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.0.2',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/gradle-build',

'https://jfrog.com/help/r/jfrog-pipelines-documentation/jira-comment',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/mvn-build',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/npm-build',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/run-cypress',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-go',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.1.0',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/0.0.3',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-java',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-jfrog-cli',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-node',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setup-yarn',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/slack-notify',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/upload-file-to-s3',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/extending-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extension-expressions',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extension-resource-model',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extension-step-model',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extension-usage',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipelines-extensions-tutorial',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-pipeline-s-extensions',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/working-with-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-history',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-run-logs',  
,  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/running-a-pipeline',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/breaking-your-pipelines-into-steps',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/triggering-pipelines-and-steps',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-stateful-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/inserting-secrets-in-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/running-multiple-steps-on-the-same-build-node',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/sending-notifications-from-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-jira-issues-from-pipelines',  
'https://jfrog.com/help/r/jfrog-pipelines-documentation/sending-build-status-to-source-control',

```

'https://jfrog.com/help/r/jfrog-pipelines-documentation/recording-unit-test-results',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/deploying-to-kubernetes-in-pipelines',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setting-up-maven-builds-for-pipelines',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/caching-step-runtimes',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-multibranch-pipelines',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-jenkins-with-pipelines',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/using-the-matrix-step',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setting-retention-policy',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/signed-pipelines'
,
'https://jfrog.com/help/r/jfrog-pipelines-documentation/setting-step-timeouts',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/speeding-up-pipeline-runs',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/triggering-a-run-with-custom-parameters',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/adding-build-badges',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/conditional-workflows',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/approval-gates',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipeline-of-pipelines',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/embedded-pipelines',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/monitoring-pipelines-using-new-relic-apm',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/managing-runtimes',
,
'https://jfrog.com/help/r/jfrog-pipelines-documentation/choosing-your-runtime-image',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/running-steps-on-the-host',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/choosing-node-pools',
'https://jfrog.com/help/r/jfrog-pipelines-documentation/runtime-images']

```

Remarks:

- We have filtered all the required links

1.2 Now will work on data extraction for each link

```
In [ ]: '''
    Experimenting data extartcion
    '''

import html2text
url = 'https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-dy
# test = soup(responce.text, 'html.parser')
def htmlTotext(url:str) -> str:

    responce = requests.get(url)
    h = html2text.HTML2Text()
    h.body_width = 0
    h.ignore_links = True # to ignore the links
    h.ignore_images = True # to ignore the images
    h.ignore_tables = True # To remove table tags and ignore the table
    # h.ignore_emphasis =True
    h.mark_code = True # to mark if code snippets are present its importen
    h.single_line_break = True
    return str(h.handle(responce.text))

print(htmlTotext(url))
```

```
# Creating Dynamic Nodes on Kubernetes
## JFrog Pipelines Documentation
ft:sourceType
    Paligo
    * JFrog Pipelines
    * Pipelines Use Cases
    * Pipelines Concepts
    * Pipelines Step-By-Step
    * See it Live
    * Pipelines Quickstart
    * Pipeline Example: Hello World
    * Pipeline Example: Docker Build and Push
    * Pipeline Example: Release to Edge Node
    * Pipeline Example: Go Build
    * Pipeline Example: Npm Build
    * Pipeline Example: Maven Build
    * Pipeline Example: Helm Blue-Green Deploy
    * Configuring Pipelines
    * Managing Pipelines Integrations
    * Managing Pipeline Sources
    * Pipeline Source Sync Recovery
    * Managing Pipelines Node Pools
    * Creating Custom VM Images
    * Creating Dynamic Nodes on Kubernetes
    * Managing Pipelines Static Nodes
    * Sending Pipelines Nodes Agent Logs to Logstash
    * Creating Pipelines
    * Defining a Pipeline
    * Pipelines Integrations
    * Airbrake Integration
    * Artifactory Integration
    * AWS Keys Integration
```

- \* Azure Keys Integration
- \* Digital Ocean Integration
- \* Bitbucket Integration
- \* Bitbucket Server Integration
- \* Distribution Integration
- \* Docker Registry Integration
- \* File Server Integration
- \* Generic Integration
- \* GitHub Integration
- \* GitHub Enterprise Integration
- \* GitLab Integration
- \* Google Cloud Integration
- \* Jenkins Integration
- \* Jenkins Server Integration
- \* JFrog Platform Access Token Integration
- \* Jira Integration
- \* Incoming Webhook Integration
- \* Kubernetes Integration
- \* NewRelic Integration
- \* Outgoing Webhook Integration
- \* PagerDuty Events Integration
- \* PEM Key Integration
- \* Slack Integration
- \* SMTP Credentials Integration
- \* SSH Key Integration
- \* Pipelines Resources
- \* Creating Resources
- \* Using Resources
- \* Viewing Resources in the UI
- \* Advanced Usage of Resources
- \* Aql
- \* Artifact
- \* BuildInfo
- \* CronTrigger
- \* DistributionRule
- \* FileSpec
- \* GitRepo
- \* HelmChart
- \* Image
- \* IncomingWebhook
- \* OutgoingWebhook
- \* PropertyBag
- \* ReleaseBundle
- \* RemoteFile
- \* VmCluster
- \* Pipelines Steps
- \* Bash
- \* CreateReleaseBundle
- \* DistributeReleaseBundle
- \* DockerPush
- \* DockerBuild
- \* GoBuild
- \* GoPublishBinary

- \* GoPublishModule
- \* GradleBuild
- \* HelmBlueGreenCleanup
- \* HelmBlueGreenDeploy
- \* HelmBlueGreenRoleSwitch
- \* HelmDeploy
- \* HelmPublish
- \* Jenkins
- \* LinuxVMDeploy
- \* Matrix
- \* MvnBuild
- \* NpmBuild
- \* NpmPublish
- \* PreMatrix
- \* PostMatrix
- \* PowerShell
- \* PromoteBuild
- \* PublishBuildInfo
- \* SignReleaseBundle
- \* TriggerPipeline
- \* UploadArtifact
- \* XrayScan
- \* Pipelines Utility Functions
- \* bump\_semver
- \* replace\_envs
- \* retry\_command
- \* get\_uuid
- \* save\_artifact\_info
- \* validate\_artifact
- \* configure\_jfrog\_cli
- \* use\_jfrog\_cli
- \* check\_xray\_available
- \* cleanup\_jfrog\_cli
- \* set\_trigger\_payload
- \* get\_trigger\_payload
- \* end\_step
- \* update\_run\_description
- \* set\_run\_name
- \* Source Control
- \* compare\_git
- \* update\_commit\_status
- \* Test Reports
- \* Encryption
- \* encrypt\_string
- \* decrypt\_string
- \* encrypt\_file
- \* decrypt\_file
- \* Notifications
- \* JSON
- \* set\_payload
- \* read\_json
- \* Resources
- \* replicate\_resource

- \* write\_output
- \* Caching
- \* add\_cache\_files
- \* restore\_cache\_files
- \* Run State Management
- \* add\_run\_variables
- \* export\_run\_variables
- \* add\_run\_files
- \* restore\_run\_files
- \* Affinity Group State Management
- \* add\_affinity\_group\_files
- \* restore\_affinity\_group\_files
- \* Pipeline State Management
- \* add\_pipeline\_variables
- \* export\_pipeline\_variables
- \* add\_pipeline\_files
- \* restore\_pipeline\_files
- \* Step Properties
- \* find\_resource\_variable
- \* get\_integration\_name
- \* get\_resource\_name
- \* get\_resource\_names
- \* get\_affinity\_group\_step\_names
- \* find\_step\_configuration\_value
- \* Pipelines Environment Variables
- \* Pipelines Templates
- \* Global Templates
- \* HelloWorld
- \* GoCI
- \* GradleCI
- \* MavenCI
- \* NpmCI
- \* PromoteCI
- \* DockerBuildAndPush
- \* HelmPublishAndDeploy
- \* Artifactory Artifact Cleanup Template (1.40 and higher)
- \* Using Global Templates to Create a Pipeline Source (1.31 and higher)
- \* Using Global Templates to Create a Pipeline Source (1.30 and lower)
- \* System Templates
- \* Local Templates
- \* Managing Pipelines Templates
- \* Pipelines Tasks
- \* build-publish
- \* curate-artifact
- \* email-notify
- \* frogbot
- \* 0.1.0
- \* 0.0.3
- \* go-build
- \* go-security-check
- \* go-static-check
- \* 0.1.0
- \* 0.0.2

- \* gradle-build
- \* jira-comment
- \* mvn-build
- \* npm-build
- \* run-cypress
- \* setup-go
- \* 0.1.0
- \* 0.0.3
- \* setup-java
- \* setup-jfrog-cli
- \* setup-node
- \* setup-yarn
- \* slack-notify
- \* upload-file-to-s3
- \* Extending Pipelines
- \* Pipelines Extension Expressions
- \* Pipelines Extension Resource Model
- \* Pipelines Extension Step Model
- \* Pipelines Extension Usage
- \* Pipelines Extensions Tutorial
- \* Managing Pipelines Extensions
- \* Working with Pipelines
- \* Pipeline History
- \* Pipeline Run Logs
- \* Running a Pipeline
- \* Using Pipelines
- \* Breaking Your Pipelines into Steps
- \* Triggering Pipelines and Steps
- \* Creating Stateful Pipelines
- \* Inserting Secrets in Pipelines
- \* Running Multiple Steps on the Same Build Node
- \* Sending Notifications from Pipelines
- \* Creating Jira Issues From Pipelines
- \* Sending Build Status to Source Control
- \* Recording Unit Test Results
- \* Deploying to Kubernetes in Pipelines
- \* Setting Up Maven Builds for Pipelines
- \* Caching Step Runtimes
- \* Creating Multibranch Pipelines
- \* Using Jenkins With Pipelines
- \* Using the Matrix Step
- \* Setting Retention Policy
- \* Signed Pipelines
- \* Setting Step Timeouts
- \* Speeding up Pipeline Runs
- \* Triggering a Run with Custom Parameters
- \* Adding Build Badges
- \* Conditional Workflows
- \* Approval Gates
- \* Pipeline of Pipelines
- \* Embedded Pipelines
- \* Monitoring Pipelines using New Relic APM
- \* Managing Runtimes



- \* Choosing your Runtime Image
- \* Running Steps on the Host
- \* Choosing Node Pools
- \* Runtime Images

This tutorial explains how to specify a kubeconfig for a Kubernetes Integration to authenticate to a self-hosted Kubernetes cluster for a dynamic node pool. You can use a cloud provider solution like EKS, GKE, or AKS, or a self-hosted Kubernetes solution.

This tutorial assumes that you have working knowledge of Docker and Kubernetes and understand the following concepts:

- \* Self hosting on GCP
- \* Self hosting on AWS
- \* kubeconfig files
- \* Configuring Service Accounts

#### ##### Configure a Kubernetes Service Account

You must configure a service account in Kubernetes to provide an identity for the build node processes that Pipelines will dynamically control. This procedure will use your personal account to create the service account. Make sure your personal account has permissions to do this.

#### ##### Verify Access to the Cluster

First, make sure you can authenticate yourself to the cluster. This means you have a kubeconfig file that uses your personal account. You can verify this by running this command on your local machine and you should see the file listed.

[code]

```
ls -al $HOME/.kube
```

[/code]

**\*\*Author a service account spec\*\***

To create a service account on Kubernetes, you can leverage `kubectl` and a service account spec. Create a YML file similar to the one below:

**\*\*pipelines\_k8s\_sa.yml\*\***

[code]

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: pipelines-k8s-pool    # <-- any name you'd like
  namespace: jfrog           # <-- the cluster namespace
```

[/code]

#### ##### Create the service account

You can create a service account by running the following command:

[code]

```
kubectl apply -f pipelines_k8s_sa.yaml
```

[/code]

#### ##### Manually Create a Long-lived API Token for a ServiceAccount

##### ### Note

Perform these steps if you are using Kubernetes 1.22 or higher, as these versions do not automatically create long-lived tokens. Versions of Kubernetes before v1.22 automatically created long term credentials for accessing the Kubernetes API.

If you want to obtain an API token for a ServiceAccount, you create a new Secret with a special annotation, `kubernetes.io/service-account.name`.

[code]

```
kubectl apply -f - <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: pipelines-k8s-pool
  annotations:
    kubernetes.io/service-account.name: pipelines-k8s-pool
type: kubernetes.io/service-account-token
EOF
```

[/code]

If you view the Secret using:

```
`kubectl get secret/pipelines-k8s-pool -o yaml`
```

you can see that the Secret now contains an API token for the "pipelines-k8s-pool" ServiceAccount.

Because of the annotation you set, the control plane automatically generates a token for that ServiceAccounts, and stores them into the associated Secret. The control plane also cleans up tokens for deleted ServiceAccounts

```
`.
`kubectl describe secrets/pipelines-k8s-pool`
```

The output is similar to this:

[code]

```

Name:      pipelines-k8s-pool
Namespace: default
Labels:    <none>
Annotations: kubernetes.io/service-account.name: pipelines-k8s-pool
             kubernetes.io/service-account.uid: da68f9c6-9d26-11e7-b84e-002dc52800da

```

```

Type:      kubernetes.io/service-account-token

```

```

Data
====

```

```

ca.crt:      1338 bytes
namespace:    7 bytes
token:        ...

```

[/code]

When you delete a ServiceAccount that has an associated Secret, the Kubernetes control plane automatically cleans up the long-lived token from that Secret.

##### Get Tokens and IP from Kubernetes

Once the service account has been created, you will need to retrieve some key information from Kubernetes in order to configure it through a kubeconfig.

##### Fetch the name of the secrets used by the service account

This can be found by running the following command:

[code]

```
kubectl describe serviceAccounts pipelines-k8s-pool
```

[/code]

```
**output**
```

[code]

```
Image pull secrets: <none>
Mountable secrets:  pipelines-k8s-pool-token-h6pdj
Tokens:             pipelines-k8s-pool-token-h6pdj
```

```

[code]
    kubectl config view --flatten --minify > cluster-cert.txt
    cat cluster-cert.txt
[/code]
**output**
[code]
    apiVersion: v1
    clusters:
    - cluster:
        certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk
1JSURDekNDQWZPZ0F3SUJBZ0lRZmo4VVMxNXpuaGRVbG15a3AvSVFqekFOQmdrcWhraUc5dzBC
QVFzRkFEQXYKTVMwd0t3WURWUWFERXlSaVl6RTB0eVlV5WkxwMk9UTTFMVFEhWldFdE9HTm1PUz
FrWmpSak5tUXlZem4TVRndwIaGN0TVRnd05EQTVNVGd6TVRReVdoY05Nak13TkRBNE1Ua3pN
VFF5V2pBdk1TMHdld1lEVlFRREV5UmlZekUwCk56VXlZeTAyT1RNMUxUUTFaV0V0T0d0aU9TMW
taa1JqTm1ReVl6Z3hNVGd3Z2dFaU1BMEdDU3FHU0liM0RRRUlKQVVFVQUE0SUJEd0F3Z2dFS0Fv
SUJBUURIVHFPV0ZXL09odDFTbDBjeUZXOGl5WUZPZHFN1lrRVFHa3E3enkzMAppUEQydUZyNj
RPRXRPOtDVR0Z0SVFyMkpxcGQ2UWdtQVNPMTlNUklkb3c4eUowTE5Ycm1jT2tv0UtMVy96UtdU
ClI0ZWp1VD1lclUNwUGR4b0Z1TnRtWGVuQ3g5dFdHNXdBV0JvU05reForTC9RN2ZpSutWU01SSn
hsQVJsWll4TFQKZ1hMam1HMnp3WGVFem5lL0tsdE14NU5neGs3U1NUQkRvRzhYR1NVRzhpUWZD
NGYzTk4zUET3Wk92SEtRc0MyZAo0ajVyc3IwazNuT1lwWDFwWnBYUmp0cTBRZTF0RzNMVE9nVW
lmZjJHq1BNZ1htVndtejJzd2xPb24wcldlRERKcMpqNGVqdjNrbDRRMXA2WXJBYNQ1RXYzeFVM
K1BTT2R0SlhadTFGWWREZHZyQWdNQkFBR2pJekFoTUE0R0ExVWQKRHdFQi93UUVBd0lDQkRBUE
JnTlZlUk1CQWY4RUJUURBUUgvtUEwR0NTcUdTSWlZRFFFQkN3VUFBNElCQVFCQwpHWd0R043
SHJpV2JlOUZtZFFGWFIXdjNLb0ZMd2o0NmxtTmtMVEphQ0ZUT3dzaVdJcXlIejUrZ2xIa0gwZ1
B2ClBDMlF2RmtDMXhieThBUWtlQy9PM2xXOC9IRmpMQVZQS3BtNnFoQytwK0J5R0pFS1BVTzVP
bDB0UkRDNjR2K0cKUXdMcTNNYnVPMdMmYVVLbzNMUWxFcXlWUFBiMwYzRUM3QytUamFlM0FZd2
VDUDN0dHJMdBVZV2NtU2VSK3F4TQpoaVRTalNpVXdleEY4cVV2SmM3dS9UWTFVVDNUd0hRR1dI
Q0J2YktDWHZvaU9VTjBKa0dHZXJ3VmJGd2tK0HdxCKdsZW40Q2RjOXRjVU1J1dm1hVGVCaklIYU
ZZdmIxejMyVWJDVjRTWUowa3dpbHE5RGJxNmNDUEI3Nj1wY0o1Kzkkb2cxbHVYYXZzQnYySwdN
a1EwL24KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
        server: https://35.203.181.169
        name: gke_jfrog-200320_us-west1-a_cluster
    contexts:
    - context:
        cluster: gke_jfrog-200320_us-west1-a_cluster
        user: gke_jfrog-200320_us-west1-a_cluster
        name: gke_jfrog-200320_us-west1-a_cluster
    current-context: gke_jfrog-200320_us-west1-a_cluster
    kind: Config
    preferences: {}
    users:
    - name: gke_jfrog-200320_us-west1-a_cluster
      user:
        auth-provider:
          config:
            access-token: ya29.Gl2YBba5duRR8Zb6DekAdjPtPGepx9Em3gX1LAhJuYz
q1G4XpYwXTS_wF4cieZ8qztMhB35lFJC-DJR6xcB02oXXkiZvWk5hH4YAw1FPrfsZWG57x43xC
rl6cvHAp40
            cmd-args: config config-helper --format=json
            cmd-path: /Users/ambarish/google-cloud-sdk/bin/gcloud
            expiry: 2018-04-09T20:35:02Z
            expiry-key: '{.credential.token_expiry}'
            token-key: '{.credential.access_token}'

```

name: gcp

[/code]

Copy and save two pieces of information from here:

- \* `certificate-authority-data`
- \* `server`

#### ##### Configuring Permissions in Kubernetes

Kubernetes includes a number of resources, including roles and role bindings that can be used to break your cluster into namespaces and limiting access to namespaced resources to specific accounts.

This section provides information about defining permissions in Kubernetes using roles and role binding.

#### ##### Creating a Role

A Role sets permissions within a particular namespace, which must be specified when creating a Role. Each Role has a `rules` section to define the resources that the rules apply to and the allowed operations, which are required for service account users to run builds within Kubernetes.

For example, the following example creates a Role in the `jfrog` namespace, which will allow read/write access to all resources in the namespace:

[code]

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: jfrog
  name: pipelines-builder-role
rules:
- apiGroups: [ "", "apps" ]
  resources: ["persistentvolumeclaims", "secrets", "pods", "secrets", "configmaps", "deployments", "deployments/scale", "services"]
  verbs:
    - get
    - list
    - watch
    - create
    - update
    - patch
    - delete
```

[/code]

#### ##### Creating a Role Binding

The service account that was created in the previous section can now be given the Role that was created earlier using a RoleBinding in the `jfrog` namespace:

[code]

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: jfrog-builder-rb
  namespace: jfrog
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: pipelines-builder-role
```

```
subjects:
  - kind: ServiceAccount
    name: pipelines-k8s-pool
    namespace: jfrog
```

[/code]

##### Add a Kubernetes Administration Integration

You must add a Kubernetes integration as an administration integration:

- \* From the JFrog Platform **Administration** module go to **Pipelines | Integrations**.
- \* Click **Add an Integration**.
- \* In the resulting **Add New Integration** display, click the **\_Integration Type\_** field and select Kubernetes from the dropdown list.
- \* Enter a **\_Name\_** for the Kubernetes integration
- \* Paste in a kubeconfig specification as described below
- \* Click **Create** to finish adding the Kubernetes integration

##### Specify a kubeconfig

From the steps in the prior sections, you should have the following pieces of information:

- \* <token>
- \* <certificate-authority-data>
- \* <server>

The kubeconfig specification you paste into the **\_Kube Config\_** setting should follow this format:

[code]

```
  apiVersion: v1
  kind: Config
  users:
    - name: pipelines-k8s-pool
# <--- Your service account name
    user:
      token: <token>
  clusters:
    - cluster:
        certificate-authority-data: <certificate-authority-data>
        server: <server>
        name: self-hosted-cluster
  contexts:
    - context:
        cluster: self-hosted-cluster
        user: pipelines-k8s-pool
# <--- Your service account name
    namespace: jfrog
# <--- The namespace you defined
    name: pipelines_k8s_context
    current-context: pipelines_k8s_context
```

[/code]

##### Create a Dynamic Node Pool

Once you have successfully added the Kubernetes administration integration, you can add a dynamic node pool that uses it.

[Back to home page](#)

Remarks:

- From above observations we have removed many unwanted lines and data by using various `html2text` options
- We can also see output format is `markdown` its usefull when we show the output in GenAi as markdown
- We also added `code block` in output which will useful when we print and extarct from bot output
- Still we need to remove some data like sidebar data and some common unwanted lines( `back to homepage, ## JFrog Pipelines Documentation` etc ) from data
- From various links have verified first 250 lines and last line are same in all the docuemnt hence we can stripe them

```
In [ ]: data = htmlTotext(url)
data = ('\n').join(data.split('\n')[250:-2])
```

```
In [ ]: data = data + f'\n\n Document url for reference - {url}' # we can append
print(data)
```

This tutorial explains how to specify a kubeconfig for a Kubernetes Integration to authenticate to a self-hosted Kubernetes cluster for a dynamic node pool. You can use a cloud provider solution like EKS, GKE, or AKS, or a self-hosted Kubernetes solution.

This tutorial assumes that you have working knowledge of Docker and Kubernetes and understand the following concepts:

- \* Self hosting on GCP
- \* Self hosting on AWS
- \* kubeconfig files
- \* Configuring Service Accounts

#### ##### Configure a Kubernetes Service Account

You must configure a service account in Kubernetes to provide an identity for the build node processes that Pipelines will dynamically control.

This procedure will use your personal account to create the service account. Make sure your personal account has permissions to do this.

#### ##### Verify Access to the Cluster

First, make sure you can authenticate yourself to the cluster. This means you have a kubeconfig file that uses your personal account. You can verify this by running this command on your local machine and you should see the file listed.

[code]

```
ls -al $HOME/.kube
```

[/code]

**\*\*Author a service account spec\*\***

To create a service account on Kubernetes, you can leverage `kubectl` and

a service account spec. Create a YML file similar to the one below:

```
**pipelines_k8s_sa.yml**
```

```
[code]
```

```
  apiVersion: v1
  kind: ServiceAccount
  metadata:
    name: pipelines-k8s-pool    # <-- any name you'd like
    namespace: jfrog           # <-- the cluster namespace
```

```
[/code]
```

##### Create the service account

You can create a service account by running the following command:

```
[code]
```

```
  kubectl apply -f pipelines_k8s_sa.yaml
```

```
[/code]
```

##### Manually Create a Long-lived API Token for a ServiceAccount

### Note

Perform these steps if you are using Kubernetes 1.22 or higher, as these versions do not automatically create long-lived tokens. Versions of Kubernetes before v1.22 automatically created long term credentials for accessing the Kubernetes API.

If you want to obtain an API token for a ServiceAccount, you create a new Secret with a special annotation, `kubernetes.io/service-account.name`.

```
[code]
```

```
  kubectl apply -f - <<EOF
  apiVersion: v1
  kind: Secret
  metadata:
    name: pipelines-k8s-pool
    annotations:
      kubernetes.io/service-account.name: pipelines-k8s-pool
  type: kubernetes.io/service-account-token
  EOF
```

```
[/code]
```

If you view the Secret using:

```
`kubectl get secret/pipelines-k8s-pool -o yaml`
```

you can see that the Secret now contains an API token for the "`pipelines-k8s-pool`" ServiceAccount.

Because of the annotation you set, the control plane automatically generates a token for that ServiceAccounts, and stores them into the associated Secret. The control plane also cleans up tokens for deleted ServiceAccounts

```
·
`kubectl describe secrets/pipelines-k8s-pool`
```

The output is similar to this:

```
[code]
```

```
  Name:      pipelines-k8s-pool
  Namespace: default
  Labels:    <none>
  Annotations: kubernetes.io/service-account.name: pipelines-k8s-pool
               kubernetes.io/service-account.uid: da68f9c6-9d26-11e7-
b84e-002dc52800da
```

```
  Type:      kubernetes.io/service-account-token
```



```
Data
====
ca.crt:      1338 bytes
namespace:   7 bytes
token:       ...
```

```
[/code]
```

When you delete a ServiceAccount that has an associated Secret, the Kubernetes control plane automatically cleans up the long-lived token from that Secret.

##### Get Tokens and IP from Kubernetes

Once the service account has been created, you will need to retrieve some key information from Kubernetes in order to configure it through a kubeconfig.

##### Fetch the name of the secrets used by the service account

This can be found by running the following command:

```
[code]
```

```
kubectl describe serviceAccounts pipelines-k8s-pool
```

```
[/code]
```

```
**output**
```

```
[code]
```

```
Name:      pipelines-k8s-pool
Namespace:  jfrog
Labels:     <none>
Annotations: <none>
```

```
Image pull secrets: <none>
```

```
Mountable secrets: pipelines-k8s-pool-token-h6pdj
```

```
Tokens:     pipelines-k8s-pool-token-h6pdj
```

```
[/code]
```

Note the `Mountable secrets` string. This is the name of the secret that holds the token, and will be used in the next step. Or if you are on a later version of k8s, describe the secret you created in the last step.

##### Fetch the token from the secret

Using the `Mountable secrets` string, you can get the token used by the service account. Run the following command to extract this information:

```
[code]
```

```
kubectl describe secrets pipelines-k8s-pool-token-h6pdj
```

```
[/code]
```

```
**output**
```

```
[code]
```

```
Name:      pipelines-k8s-pool
Namespace:  jfrog
Labels:     <none>
Annotations:  kubernetes.io/service-account.name=pipelines-k8s-pool
               kubernetes.io/service-account.uid=c2117d8e-3c2d-11e8-9ccd-4201
```

```
0a8a012f
```

```
Type:      kubernetes.io/service-account-token
```

```
Data
```

```
====
```

```
ca.crt:      1115 bytes
```

```
namespace:   7 bytes
```

```
server: https://35.203.181.169
name: gke_jfrog-200320_us-west1-a_cluster
contexts:
```

```

- context:
  cluster: gke_jfrog-200320_us-west1-a_cluster
  user: gke_jfrog-200320_us-west1-a_cluster
  name: gke_jfrog-200320_us-west1-a_cluster
current-context: gke_jfrog-200320_us-west1-a_cluster
kind: Config
preferences: {}
users:
- name: gke_jfrog-200320_us-west1-a_cluster
  user:
    auth-provider:
      config:
        access-token: ya29.Gl2YBba5duRR8Zb6DekAdjPtPGepx9Em3gX1LAhJuYz
q1G4XpYwXTS_wF4cieZ8qztMhB35lFJC-DJR6xcB02oXXkiZvWk5hH4YAw1FPrfsZWG57x43xC
rl6cvHAp40
        cmd-args: config config-helper --format=json
        cmd-path: /Users/ambarish/google-cloud-sdk/bin/gcloud
        expiry: 2018-04-09T20:35:02Z
        expiry-key: '{.credential.token_expiry}'
        token-key: '{.credential.access_token}'
      name: gcp

```

[/code]

Copy and save two pieces of information from here:

```

* `certificate-authority-data`
* `server`

```

#### ##### Configuring Permissions in Kubernetes

Kubernetes includes a number of resources, including roles and role bindings that can be used to break your cluster into namespaces and limiting access to namespaced resources to specific accounts.

This section provides information about defining permissions in Kubernetes using roles and role binding.

#### ##### Creating a Role

A Role sets permissions within a particular namespace, which must be specified when creating a Role. Each Role has a `rules` section to define the resources that the rules apply to and the allowed operations, which are required for service account users to run builds within Kubernetes.

For example, the following example creates a Role in the `jfrog` namespace, which will allow read/write access to all resources in the namespace:

[code]

```

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: jfrog
  name: pipelines-builder-role
rules:
- apiGroups: [ "", "apps" ]
  resources: [ "persistentvolumeclaims", "secrets", "pods", "secrets", "configmaps", "deployments", "deployments/scale", "services" ]
  verbs:
  - get
  - list

```

- watch
- create
- update
- patch
- delete

[/code]

#### ##### Creating a Role Binding

The service account that was created in the previous section can now be given the Role that was created earlier using a RoleBinding in the `jfrog` namespace:

[code]

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: jfrog-builder-rb
  namespace: jfrog
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: pipelines-builder-role
subjects:
  - kind: ServiceAccount
    name: pipelines-k8s-pool
    namespace: jfrog
```

[/code]

#### ##### Add a Kubernetes Administration Integration

You must add a Kubernetes integration as an administration integration:

- \* From the JFrog Platform **Administration** module go to **Pipelines | Integrations**.
- \* Click **Add an Integration**.
- \* In the resulting **Add New Integration** display, click the **\_Integration Type\_** field and select Kubernetes from the dropdown list.
- \* Enter a **\_Name\_** for the Kubernetes integration
- \* Paste in a kubeconfig specification as described below
- \* Click **Create** to finish adding the Kubernetes integration

#### ##### Specify a kubeconfig

From the steps in the prior sections, you should have the following pieces of information:

- \* <token>
- \* <certificate-authority-data>
- \* <server>

The kubeconfig specification you paste into the **\_Kube Config\_** setting should follow this format:

[code]

```
apiVersion: v1
kind: Config
users:
  - name: pipelines-k8s-pool
# <-- Your service account name
```

```

    user:
      token: <token>
clusters:
- cluster:
    certificate-authority-data: <certificate-authority-data>
    server: <server>
    name: self-hosted-cluster
contexts:
- context:
    cluster: self-hosted-cluster
    user: pipelines-k8s-pool
# <-- Your service account name
    namespace: jfrog
# <-- The namespace you defined
    name: pipelines_k8s_context
    current-context: pipelines_k8s_context
[/code]
##### Create a Dynamic Node Pool
Once you have successfully added the Kubernetes administration integration
, you can add a dynamic node pool that uses it.

```

Document url for reference – <https://jfrog.com/help/r/jfrog-pipelines-documentation/creating-dynamic-nodes-on-kubernetes>

Remarks: Able to successfully extract the data for our desired format now we can append the data from each link and prepare the data for training

```

In [ ]: '''
final Function to concat all the data
'''

import pandas as pd # for dataframe creation and convert to csv
from tqdm import tqdm
pd.set_option('display.max_colwidth',None)

def dataOrchestrator(url:str) -> pd.DataFrame:
    '''
    This function will take the parent portal URL and extrcat links and d
    and return pandas DataFareme
    '''
    dataframe = pd.DataFrame(columns=['Title', 'PipelineProcess'])
    all_links = tqdm(linkExtarction(url), desc=f'Extrating all the links
    pipeline_links = tqdm(filter_links(all_links), desc=f'Filtering only p
    for link in tqdm(pipeline_links, desc=f'Data extraction in progress...
        temp_data = htmlTotext(link)
        data = ('\n').join(temp_data.split('\n')[250:-2]) # striping first
        data = data + f'\n\n Document url for reference – {url}' # Adding
        templist = [[link[55:], data]]
        dataframe = dataframe._append(pd.DataFrame(templist, columns=['Ti
    return dataframe

```

```
In [ ]: # Finale data extraction
parentUrl = 'https://jfrog.com/help/r/jfrog-pipelines-documentation/pipel
final_data = dataOrchestrator(parentUrl)
final_data
```

```
Extracting all the links from - https://jfrog.com/help/r/jfrog-pipelines-do
documentation/pipelines-steps: 100%|██████████| 329/329 [00:00<00:00, 249264
.09it/s]
Filtering only pipeline links: 100%|██████████| 244/244 [04:33<00:00, 1.12
s/it]
Data extraction in progress...: 100%|██████████| 244/244 [04:33<00:00, 1.
12s/it]
```

```
Out [ ]:
```

	Title	PipelineProcess
--	-------	-----------------

0	jfrog-pipelines	JFrog Pipelines offers JFrog Platform customer...
1	pipelines-use-cases	Let's explore some of the most common ways to ...
2	pipelines-concepts	Before learning how to use Pipelines, here are...
3	pipelines-step-by-step	After you have a Pipelines installation workin...
4	see-it-live	Have we piqued your interest? Ready to see som...
...	...	...
239	managing-runtimes	Every step in your pipeline executes on a buil...
240	choosing-your-runtime-image	By default, your steps run inside a container ...
241	running-steps-on-the-host	When you need to execute your step directly on...
242	choosing-node-pools	A pipeline can, if necessary, control through ...
243	runtime-images	A runtime image is the Docker image for the co...

244 rows × 2 columns

```
In [ ]: final_data.iloc[[0,78,101,234]]
```

```
Out [ ]:
```

	Title	Pi
--	-------	----

JFrog Pipelines offers JFrog Platform customers three vital capabilities: automation (CI/CD), workflow and tool orchestration, and the optimal JFrog toolset functionality in use. Consistent with JFrog's customer-centric philosophy, Pipelines is enterprise-ready and universal.

Automation

A pipeline is an event-driven automated workflow for executing DevOps activities (CI, deployments, infrastructure provisioning, etc). It is a sequence of interdependent **steps** which execute discrete functions on **resources**, which hold the information needed to execute (files, pairs, etc).

Developers can create pipelines easily with a simple declarative language. While each step in a pipeline executes in a stateful environment, Pipelines provides facilities to manage state and step context in the workflow so that all dependent steps can access the information from upstream steps in order to execute. This helps coordinate activities across diverse DevOps tools and teams without custom DIY scripts.

0

jfrog-  
pipelines

configured for a variety of scenarios, including:

- \* Continuous Integration applications
- \* Continuous Delivery workflows that connect all your DevOps activities across tools and functional silos
- \* Automate IT Ops infrastructure provisioning, security patching, and image building

and running with JFrog Pipelines

In this section, you will find information started whether you are a new user or an existing user.

- \* If you do subscription, get started with trial subscription of the JFrog Platform
- \* If you are a new user, get started with the onboarding via Pipelines.Onboarding Best Practices: JFrog Pipelines

Fe Pipelines as Code

Define your automated workflow through code, using a specific language in a YAML file of key-value pairs that you can create with your favorite text editor.

Real Time Visibility

JFrog Pipeline pipeline definition as an interactive diagram, helping you to see the flow of their inter-dependencies, as well as view the success record of any pipeline performed.

Universal

Connect your pipeline automation to your repositories in a version control system (such as GitHub or GitLab) to automatically trigger execution on any new submission (commit) of a repository. Connect to other popular tools through your credentials for storage, notifications, orchestration and more through a library of integrations.

Integration with Artifactory

JFrog Pipelines is designed to be used with built-in directives for pushing artifacts, performing builds, scanning artifacts, information, image scanning, and build promotion.

Integration with JFrog Platform

JFrog Pipelines is designed as an integral part of the JFrog Platform, including scanning artifacts/builds through Xray, the creation and delivery of bundles through JFrog Distribution, for a complete end-to-end SDLC from commit to production runtime.

Security First

Fine-grained permissions and access control limit who can access workflows. Centralized, encrypted credentials and keys help ensure secrets stay safe.

Ready to Use

Manage multiple execution nodes using a single installation of JFrog Pipelines, which automatically distribute Pipeline execution across them for scale and performance.

Watch the Screencast

Document url for reference - <https://jfrog.com/pipelines-documentation/>

The **DockerBuild** native step performs a build to produce a Docker image from a Dockerfile in a GitRepo source repository resource.

In the step configuration, you must provide the name ( ``dockerFileName`` ) and directory ( ``dockerFileLocation`` ) of the Dockerfile that contains the command to be processed by a build step, as well as the name ( ``dockerImageName`` ) and tag ( ``dockerImageTag`` ) of the resulting image. The image is built on the build node, and information about that image is stored in the run state.

To build a Docker image that relies on a base image:

1. Define the base image as an Image resource, with ``baseImage`` set to ``true``.
2. Specify the Image resource as one of the ``inputResources`` of the DockerBuild step.

To include artifacts in the Docker image that are built by the GitRepo source repository:

1. Define a FileSpec resource that specifies the artifacts to include from Artifactory.
2. Specify the FileSpec resource as one of the ``inputResources`` of the DockerBuild step.

Proper usage of the DockerBuild and DockerPush steps must be assigned to an ``affinityGroup`` to share state. If this is not done, the output of DockerBuild is not available for DockerPush. For more information on using ``affinityGroup``, see the Docker Build and Push quickstart demonstrates the definition of a pipeline using the DockerBuild and DockerPush native steps to build a single Docker image, push it to Artifactory, and then publish the BuildInfo.

YAML Schema

The schema for DockerBuild native step is as follows:

```

DockerBuild
  pipelines:
    - name: <string>
      steps:
        - name: <string>
          type: DockerBuild
          configuration:
            #inherits all the tags from bash
            affinityGroup: <string>
            dockerFileLocation: <string>
            dockerFileName: <string>
            dockerImageName: <string>
            dockerImageTag: <string>
            baseImage: <boolean>
            inputResources:
              - <string>

```

## 78      dockerbuild

```

<string>\n dockerImageTag: <string>\n dockerOptions: <string>\n \n in
name: <artifactory or docker registry integration> # required\n \n input
name: <GitRepo resource> # required, git repository containing your
name: <Image resource> # optional base image\n - name: <FileSpe
optional\n \n execution:\n onStart:\n - echo "Preparing for work..." \n c
echo "Job well done!"\n onFailure:\n - echo "uh oh, something \
onComplete: #always\n - echo "Cleaning up some stuff"\n[/
Tags\n##### name\nAn alphanumeric string (underscores are p
identifies the step.\n##### type\nMust be `DockerBuild
type.\n##### configuration\nSpecifies all configuration selections
execution environment. This step inherits the Bash/ PowerShell step
tags, including these pertinenttags:\nTag\n **
usage**\nRequired/Optional \n`affinityGroup`\nMust specify an affinity
that is the same as specified in a subsequent DockerPush s
\n`inputResources`\nMust specify:\n * a GitRepo resource (the
Dockerfile)\n\n\nOptionally, you may also specify:\n * One or more Imag
pull base images used in the build or to trigger this build.\n * One or
resources that specify what files to include in the build context.
automatically copied to `dockerFileLocation`.\n\n\nRequired/Optiona
these tags can be defined to support the step's native operation:\n###
from Bash\nAll native steps derive from the Bash step. This means that
the same base set of tags from Bash, while native steps have their
tags as well that support the step's particular function. So it's importan
with the Bash step definition, since it's the core of the definit
steps.\nTag\n **Description of usage**\nReq
\n`dockerFileLocation`\nDirectory containing the Dockerfile, which is th
Docker build configuration. This file is also used as the context for the
The path provided should be relative to the root of the input GitRepo re
location is provided, the default is the root of the GitRepo repositi
\n`dockerFileName`\nName of the Dockerfile.\nRequired \n`dockerImag
name of the Docker image to create. This can be set using environme
triggering a run using parameters.\nRequired \n`dockerImageTag`\n
Docker image to create. This can be set using environment variables
run using parameters.\nRequired \n`dockerOptions`\nAdditional c
docker build command.\nOptional \n##### execution\nDeclares colle
command sequences to perform for pre- and post-execution p
**Description of usage**\nRequired/Optional \n`onStart`\nCommand:
advance of the native operation\nOptional \n`onSuccess`\nCommands
successful completion\nOptional \n`onFailure`\nCommands to exe
completion\nOptional \n`onComplete`\nCommands to e
completion\nOptional \n\nThe actions performed for the `onExecu
inherent to this step type and may not be overridden.\n##### E
following examples use a GoLang Git repository represented by a Gitl
named `gosvc_app` to create a Docker image that is published to Ar
assume that an Artifactory integration named `MyArtifactory` has bee
that the Artifactory instance has a Docker repositi
`docker.artprod.company`.\n * These examples require an Artifactory li
a GitHub Integration.GitHub Integration\n * The Pipelines DSL for a sim
available in this repository in the JFrog GitHub account.\n * For a fu
Pipeline Example: Docker Build and Push.\n\n\nThe following resource
support these examples. Not all of these resources
examples.\n##### Resources\n[code]\n resources:\n # Appl
repository\n - name: gosvc_app\n type: GitRepo\n configuration:
myGithub\n path: myuser/myrepo # replace with your repository name
include: master\n \n # Docker image in an Artifactory repositi
base_image\n type: Image\n configuration:\n registry: n
sourceRepository: docker-local # replace with your repository name\

```



```

docker.artprod.mycompany.com/baseimage\n imageTag: latest\n autoF
Files in an Artifactory repository\n - name: icon_files\n ty
configuration:\n sourceArtifactory: myArtifactory\n pattern: my-loc
images/\n target: icons/\n[/code]\n##### Build a Docker image
repository\nThis example builds a Docker image to a Docker registry
The tag for the image is set to the pipeline's run number.\n[/code]\r
name: demo_pipeline\n steps:\n - name: bld_image\n type:
configuration:\n dockerFileLocation: .\n dockerFileName:
dockerImageName: docker.artprod.mycompany.com/gosvc # replace
qualified Docker registry/image name\n dockerImageTag: [Ma
Error]{run_number}\n dockerOptions: --build-arg build_number_env_v
Processing Error}{run_number}\n inputResources:\n - name: gosvc_
base_image\n integrations:\n - name: MyArtifactory\n[/code]\n##### I
image with files outside the current path\nThis example demonstr
Docker image that includes files outside of the current path. It pulls ic
in an Artifactory repository for integration art named `my-local-repo`
that the Dockerfile has a command that will include the files in `/i
image.\n[/code]\n pipelines:\n - name: demo_pipeline\n steps:\n - name
type: DockerBuild\n configuration:\n dockerFileLocation: .\n do
Dockerfile\n dockerImageName: docker.artprod.mycompany.com/g
with your fully qualified Docker registry/image name\n do
${run_number}\n inputResources:\n - name: gosvc_app\n - nam
integrations:\n - name: MyArtifactory\n[/code]\n##### How it Works\n
the DockerBuild native step in a pipeline, it performs the followin
the background:\n * cp (if there is a FileSpec input, copy those files to
cloned GitRepo input)\n * docker build\n * add_run_variables (add se
that are later used when pushing the Docker image or publishing build
rt build-collect-env (collect environment information to be later publis
build info)\n * add_run_files (save information collected for buil
Document url for reference - https://jfrog.com/help/r/jf
documentation/p

```

The **UploadArtifact** native step uploads artifacts to Artifactory. Or also publish build information to Artifactory and trigger Xray scans.\nTh the JFrog CLI to upload an artifact to Artifactory. The file(s) may be FileSpec, if already in Artifactory, or RemoteFile or GitRepo input.\nSchema\nThe YAML schema for UploadArtifact native step i

```

UploadArtifact\n[/code]\n pipelines: \n - name: <string>\n st
<string>\n type: UploadArtifact\n configuration:\n targetPath: <string>
sourcePath: <string> #optional\n properties: <string> #optional\n regE
#optional\n flat: <boolean> #optional\n module: <string> #optional\n
#optional\n recursive: <boolean> #optional\n dryRun: <boolean>
symlinks: <boolean> #optional\n explode: <boolean> #optional\n exclus
#optional\n includeDirs: <boolean> #optional\n syncDeletes: <string>
forceXrayScan: <boolean> #optional\n failOnScan: <boolean> #
autoPublishBuildInfo: <boolean> #optional\n inputResources:\n - name:
- name: artifactoryFileSpec \n - name: myRemoteFile \n outputResour
myFileSpec\n - name: myBuildInfo\n integrations:\n - name: m
execution:\n onStart:\n - echo "Preparing for work..." \n onSuccess:\n -
done!" \n onFailure:\n - echo "uh oh, something went wrong" \n
#always\n - echo "Cleaning up some stuff" \n \n[/code]\n##### `
name\nAn alphanumeric string (underscores are permitted) tha
step.\n##### type\nMust be `UploadArtifact` for this step t
configuration\nSpecifies all configuration selections for the ste
environment. This step inherits the Bash/ PowerShell step confi
including these pertinent tags:\nTag\n Description of usage\nReq
\n`integrations`\nMust specify an Artifactory Integrati

```

**101** uploadartifact

`\n`inputResources`\n`May specify a GitRepo, FileSpec, or Remot containing the file(s) to be uploaded. One of each type may be specif  
`\n`outputResources`\n`Must specify a BuildInfo resource if ``autoPubli`  
`set as `true`.\n`If ``JFROG_CLI_BUILD_NAME`` or ``JFROG_CLI_BUIL`  
`set as an environment variable for the pipeline or the step, that name :`  
`is used for the output BuildInfo. Otherwise, the default `bu`  
``buildNumber` are `pipeline_name`and`run_number`.\n`May also spe  
`resource to be updated with the pattern and properties of`  
`Artifact.\n`May be required \nIn addition, these tags can be defined  
`step's native operation:\nTag\n **Description of usage**\nReq`  
`\ntargetPath\n`Path to upload the files, including repository na  
`\n`sourcePath`\n`Files to upload. If this is a relative path pattern, it is  
`root of a GitRepo/FileSpec/RemoteFile input.\n`Default is ``*`` when ``regl`  
`and `.*` when `regExp` is `true`.\n`Optional \n`properties`\nSemi-cc  
`properties for the uploaded artifact`  
``myFirstProperty=one;mySecondProperty=two`.\n`Properties ``pipelines`  
``pipelines_run_number`, `pipelines_step_id`, `pipelines_pi  
`pipelines_step_url`, `pipelines_step_type`, and `pipelines_step_plat  
be added.\nOptional \n`regExp`\nWhen set as `true`, regular expressio  
other parameters, such as `sourcePath`, instead of wildcards. Expressio  
parentheses.\nDefault is `false`.\nOptional \n`flat`\nWhen set as `true`,  
files are flattened, removing the directory structure.\nDefault is `fal  
\n`module`\nA module name for the Build Info.\n`Option  
``distribution/component/architecture` for Debian packages. If th`  
`component, or architecture includes a / it must be double-escaped`  
``distribution/my\\\\component/architecture` for a `m`  
`component.\n`Optional \n`recursive`\nWhen set as ``false``, do r  
`matches in subdirectories.\n`Default is `true``.\nOptional \n`dryRun`  
``true`, nothing is uploaded.\n`Default is ``false``.\nOptional \n`symlinks`  
``true`, symlinks matching the other criteria are upload`  
``false`.\n`Optional \n`explode`\nWhen set as ``true`` and the uploader  
`archive, the archive is expanded.\n`Default is ``false``.\nOptional \n`exclu  
`colon separated patterns to exclude.\n`Optional \n`includeDirs`\nWhen  
`empty directories matching the criteria are uploaded.\n`Default is ``fal`  
`\n`syncDeletes`\n`A path under which to delete any e  
`Artifactory.\n`Optional \n`forceXrayScan`\nWhen set as ``true``, force:  
`after publishing to Artifactory.\n`Default is ``false``.\nOptional \n`failOr  
`set as `true`, and when the Xray Policy Rule Fail Build checkbox is ch`  
`Xray scan will result in a failure of the step.Creating Xray Policies and I`  
`is `true`.\n`Optional \n`autoPublishBuildInfo`\nWhen set as ``true``, p  
`info to Artifactory.\n`Default is ``false``.\nOptional \n##### execu  
`collections of shell command sequences to perform for pre- and p`  
`phases:\nTag\n **Description of usage**\nRequired/Optional\n`onStart`  
`to execute in advance of the native operation\n`Optional \n`onSuccess  
`to execute on successful completion\n`Optional \n`onFailure`\nCommarr  
`on failed completion\n`Optional \n`onComplete`\nCommands to e  
`completion\n`Optional \n`The actions performed for the `onExecu  
`inherent to this step type and may not be overridden.\n`#### Note\|  
``onStart`, `onSuccess`, `onFailure`, and `onComplete` are reser`  
`Using these keywords in any other context in your execution scri`  
`unexpected behavior.\n`##### Examples\nThe following examples show  
`which a UploadArtifact step can be configured.\n`##### Uploading  
`Another Repository using a FileSpec Resource\n`The most  
`UploadArtifact. Uses all default values. This step will download the file`  
`FileSpec and upload it to the location in `targetPath`. The optional o`  
`resource will be updated with the `targetPath` and the default prope`  
`the uploaded artifact.\n **UploadArtifact**\n[code]\n pipelir`

```

uploadArtifactPipeline\n steps:\n - name: uploadArtifactStep\n type: UploadArtifact\n configuration:\n targetPath: my-repository/myDirectory/myFile.txt\n in\n name: myArtifactoryIntegration\n inputResources:\n - name: myInput\n outputResources:\n - name: myOutputFileSpec\n \n[/code]\n#####\n Artifact from a RemoteFile Resource\n\nIn this example, the input is\n resource. Otherwise, this is very similar to the previous example with\n downloads a file that is then uploaded and an optional FileSpec output\n the uploaded file.\n **UploadArtifact**\n[/code]\n\npipeline\nuploadArtifactPipeline\n steps:\n - name: uploadArtifactStep\n type: UploadArtifact\n configuration:\n targetPath: my-repository/myDirectory/myFile.txt\n in\n name: myArtifactoryIntegration\n inputResources:\n - name: myInput\n outputResources:\n - name: myOutputFileSpec\n \n[/code]\n#####\n Info and Trigger Xray Scan\n\nIn this example, build info is published\n UploadArtifact step and an Xray scan is triggered.\n **UploadArtifact\n pipelines: \n - name: uploadArtifactPipeline\n steps:\n - name: upload\n type: UploadArtifact\n configuration:\n targetPath: my-repository/myDirectory/myFile.txt\n autoPublishBuildInfo: true\n forceXray\n integrations:\n - name: myArtifactoryIntegration\n inputResources:\n myFileSpec\n outputResources:\n - name: myBuildInfo\n \n[/code]\n\nWorks\n\nWhen you use the **UploadArtifact** native step in a pipeline, it\n following functions in the background:\n * jfrog rt config (configure JFrog\n integration listed in the yaml)\n * jfrog rt use (configure JFrog CLI to\n for the integration listed in the yaml)\n * mkdir (create a directory to use\n relative paths in the following actions)\n * cp (copy the FileSpec, local\n GitRepo files to the new directory, limit one of each input type)\n * uploadArtifact (upload the Artifact)\n * write_output (update the FileSpec output resource\n uploaded pattern and properties)\n * add_run_variables (save information\n for future steps to reference)\n * jfrog rt build-collect-env (collect\n environment, preparing for build publish)\n * jfrog rt build-publish (publish\n only if autoPublishBuildInfo is true)\n * write_output (update the Build\n resource with the published name/number)\n * jfrog rt build-scan (if for\n true)\n * add_run_files (adds build info to run state)\n\n\n\n Do\n reference - https://jfrog.com/help/r/jfrog-pipelines-documentation/p

```

Conditional workflow in Pipelines enables you to choose if a step executes based on certain conditions set for the previous upstream step. This means when a workflow reaches a conditional step, it can choose different workflow paths based on the step's status. This provides more flexibility in the execution of a pipeline.

#### Note

Conditional workflow can be applied to a single step.

##### Step Status Conditional Workflow

With the `status` conditional workflow, you can configure a step to execute only if an input step's status in the current run, is satisfied. You can configure any number of statuses for a step.

**\*\*YAML Schema\*\***

```

steps:\n - name: <step_name>\n type: <step_type>\n configuration:\n allowFailure: boolean #optional\n inputStatuses:\n - <step_name>\n status:\n - <terminal_status>\n - <terminal_status>\n[/code]

```

#### Note

It is important to note that the status of an input step in the current run only is considered for conditional workflow. If a step is not part of the current run, it is always assumed that the condition for that step is met.

##### Adding Conditional Workflow for Steps

To add a conditional workflow for a step:

1. In the `inputSteps` section of a step, add the status of the upstream step as a property.
2. Add any of these values: `success`, `failure`, `cancelled`, `skipped`, `unstable`, `timeout`.

#### Note

These values are in lowercase and use the same spelling as shown above. If a step fails from this will cause the pipeline source sync to fail.

**\*\*Example\*\*** : In the following example, step\_B has only one status: `success`. Step\_C has multiple statuses: `skipped`, `cancelled`.

```

- name: step_A\n type: Bash\n configuration:\n allowFailure: boolean #optional\n inputSteps:\n - name: step_

```

234

conditional-  
workflows

```

success\n - name: step_C\n status:\n - failure\n - skipped\n - cancelled
**allowFailure**\n **Optional** : If you do not want a particular step to
the final status of the run, you can add `allowFailure: true` to the `
section of that step. When this option is used, even when a step fails
the final status of the run is not affected.\nFor example, a pipeline cont
S1 and S2: \n * **Scenario 1** : Step S2 is a cleanup step and its statu
The overall run status should be determined by S1's status and S2's sta
ignored. In this case, add `allowFailure: true` to S2, since this is purely a
and only S1's status should be taken into consideration.\n * **Scenario
has been configured to fail as part of the workflow. However S2 runs
and the run is not to be considered a failure. The run's final status shou
status since S1's status does not interrupt the flow. In this case, add
true` to S1 since S1's failure is a known possibility and expected, and t
affect the final status of the run.\n\n\nFor more examples, see
Examples.\n#### Run Variable Conditional Workflow\nCreate a cond
the values of `add_run_variables` environment variable, so that
skipped based on dynamically set variables before it gets assigned to
Note\nWhen using a `condition`, boolean values must be enclosed
**Examples** :\n * `condition: 'trigger == "true"'`\n * `condition:
'true'`\n * `condition: trigger == 'true'`\n * `condition: trigger == "tr
Note\nPipelines environment variables cannot be added as a cond
Environment Variables\n **YAML Schema**\n[code]\n st
<step_name>\n type: <step_type>\n execution:\n onExecute:\n - add_
'key=value'\n - name: <step_name>\n type: <step_type>\n configuratio
'key == value' // Any logical boolean expression that results
inputSteps:\n - name: <step_name>\n[/code]\n **Example**\n[code]\r
name: Example\n steps:\n - name: step1\n type: Bash\n execution:\n (
echo 'step1'\n - add_run_variables 'var1=1'\n - name: step2\n
configuration:\n condition: 'var1 == 1' // Any logical boolean expression
a boolean\n inputSteps:\n - name: step1\n execution:\n onExe
'success'\n[/code]\n#### Environment Variables Conditional Workl
conditional workflow based on environment variables defined in the
section of your pipelines YAML file. The step executes when the declar
met.\n **Example**\n[code]\n pipelines:\n - name: myPipelines\n c
environmentVariables:\n readOnly:\n new_env:\n default: 1\n allowC
steps:\n - name: step1\n type: Bash\n configuration:\n environme
new_env:\n default: 2\n #allowCustom: true\n condition: new_env == 2\n
onExecute:\n - echo [Math Processing Error]res_my_app_repo_res
./execute.sh\n - popd\n[/code]\n **Example 2 - newVersionOn
resources:\n - name: new_resource\n type: PropertyBag\n c
runNumber: 0\n \n pipelines:\n - name: pipeline_01\n steps:\n - name
type: Bash\n configuration:\n outputResources:\n - name: n
execution:\n onExecute:\n #- write_output new_resource runN
[Processing Error]\n \n pipelines: \n - name: simple_jenkins_demo\n st
jenkins\n type: Jenkins\n configuration:\n inputResources:\n - nam
jenkinsJobName: testPipeline\n integrations:\n - name: myJenkir
step_A\n type: Bash\n configuration:\n inputSteps:\n - name: jenkins
failure\n - error\n - timeout\n execution:\n onExecute:\n - echo "Executi

step_jenkins_statusName == " failure "]; then echo " Do something
step_jenkins_statusName == "error" ]; then echo "Do something else";
simple_conditional_B\n type: Bash\n configuration:\n inputSteps:\n - n
status:\n - failure\n - error\n execution:\n onExecute:\n - ec
simple_conditional_B"\n - printenv\n \n[/code]\n#### allowFailure Exa
Example 1\nStep1 is configured for success and step2 for failure. Step
run when step1 fails and the final status of the run is success.\n[code]\r
name: PIPE_9455_Workflow_03\n steps:\n - name: step1\n type: Bash\n

```

```

onExecute:\n - echo 'step1'\n - name: step2\n type: Bash\n co
allowFailure: true\n inputSteps:\n - name: step1\n status:\n - succe
failure\n - timeout\n execution:\n onExecute:\n - echo 'su
1\n[/code]\n##### Example 2\nStep1 is configured for failure and step
Step2 is allowed to run when step1 fails and the final statu
success.\n[/code]\n pipelines:\n - name: PIPE_9455_Workflow_05\n st
step1\n type: Bash\n configuration:\n allowFailure: true\n execution:\n (
echo 'step1'\n - exit 1\n - name: step2\n type: Bash\n configuration:\n i
name: step1\n status:\n - success\n - error\n - failure\n - timeout\n
onExecute:\n - echo 'success'\n[/code]\n##### Example 3\nStep1 is
success and step2 for failure. When triggered, the final statu
failure.\n[/code]\n pipelines:\n - name: PIPE_9455_Workflow_03\n st
step1\n type: Bash\n configuration:\n allowFailure: true\n execution:\n (
echo 'step1'\n - name: step2\n type: Bash\n configuration:\n inputSt
step1\n status:\n - success\n - error\n - failure\n - timeout\n
onExecute:\n - echo 'failure'\n - exit 1\n[/code]\n\n Document url f
https://jfrog.com/help/r/jfrog-pipelines-documentation/p

```

```

In [ ]: # We have extarcted the data and we can save as CSV
final_data.to_csv('./jFrog_pipeline.csv')

```

We have extarcted our data and we can use `jFrog_pipeline.csv` for training