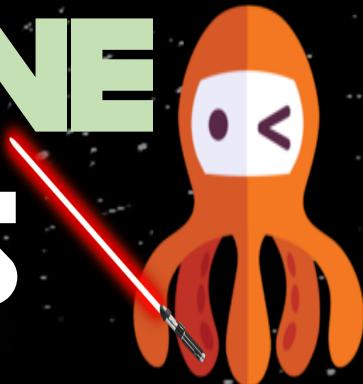




THE JENKINS PIPELINE FORCE AWAKENS

REINFORCED BY TIKAL KNOWLEDGE

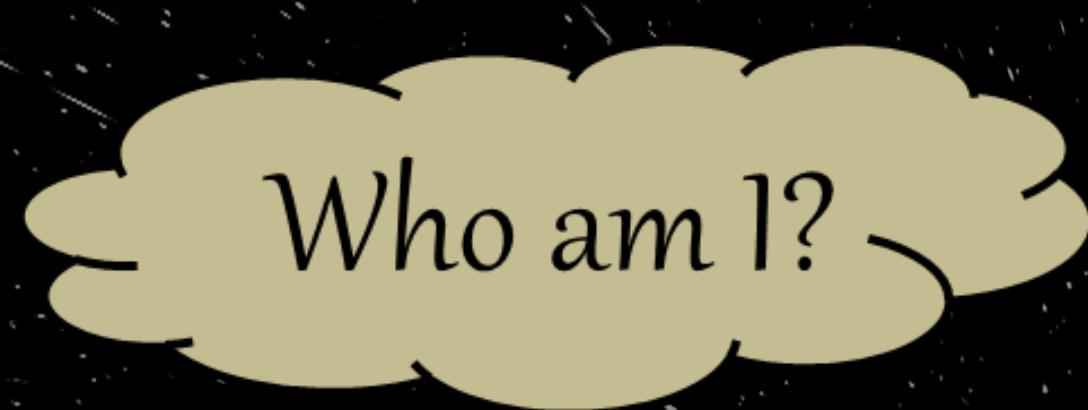


JENKINS WORKSHOP EPISODE:
PRACTICE JENKINS SHARED LIBRARIES



 YORAM MICHAELI





Who am I?

Yoram Michaeli

Tikal Knowledge

DevOps Tech Leader

JEDI MASTER



Episode JUC MMXVIII

**While using Jenkins Pipeline
is spreading into the universe,
the dark side habits of Jenkinsfile
copy-paste and Stack Overflow addiction
still rules.**

**Using shared libraries and programming
capabilities is a better approach and this
presentation will explain and show you how to
use it.**

May the Jenkins force be with you...

Agenda

- The use case
- Pure Jenkinsfile implementation
- What is a shared library?
- Steps examples
- Flow implementation using shared library
- The Tikal reference

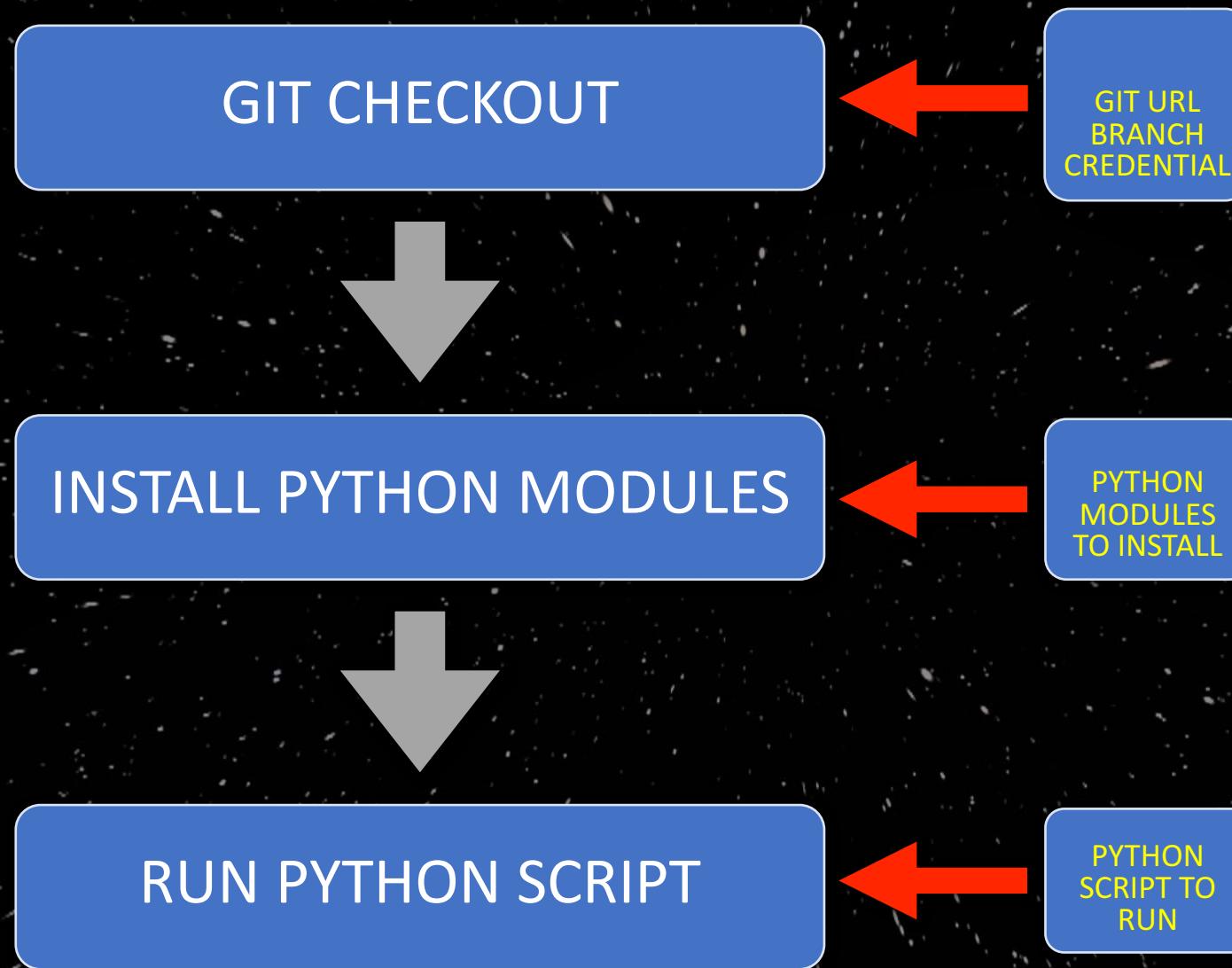


STAR WARS
EPISODE II
ATTACK OF THE CLONES

**THE
USE CASE**



THE USE CASE





PIPELINE IMPLEMENTATION



PIPELINE IMPLEMENTATION

tikal-pipelib-touchstone

examples

pipelineForce

genericPythonScriptRunner

Jenkinsfile

```
parameters
{
    string(name: 'REPOSITORY_URL',
           description: 'URL to GIT repository')
    string(name: 'GIT_REPO_CRED',
           description: 'Credential name of GIT user')
    string(name: 'BRANCH',
           defaultValue: 'master',
           description: 'Name of branch')
    string(name: 'REQUIREMENTS_FILE',
           description: 'Relative path to the requirements file')
    string(name: 'PYTHON_SCRIPT_FILE',
           description: 'Relative path to the python script')
}
```

Parameters

```
agent
{
    docker
    {
        image 'python:2.7'
        args "-u root"
    }
}
```

Run environment

```
pipeline
{
    parameters
    {
        ...
    }
    agent
    {
        ...
    }
    stages
    {
        stage('Setup')
        {
            ...
        }
        stage('Install requirements')
        {
            ...
        }
        stage('Run script')
        {
            ...
        }
    }
}
```



stage('Setup')

steps

{

checkout([

```
    $class: 'GitSCM',
    branches: [[name: BRANCH]],
    userRemoteConfigs: [[url: "${REPOSITORY_URL}",
                        credentialsId: "${GIT_REPO_CRED}"]]
```

])

}

stage('Install requirements')

{

steps

{

```
    sh "pip install -r ${REQUIREMENTS_FILE}"
```

}

stage('Run script')

{

steps

{

script

{

sh

```
"python ./${PYTHON_SCRIPT_FILE}"
```

}

Code checkout

Requirements

Script run



BY
SA

PIPELINE COPY-PASTE IMPLEMENTATION

tikal-pipelib-touchstone > examples > pipelineForce > genericPythonScriptRunnerClient > Jenkinsfile-client

environment

```
{  
    REPOSITORY_URL='https://github.com/tikalk/pipelineForceCode.git'  
    GIT_REPO_CRED=''  
    BRANCH='master'  
    REQUIREMENTS_FILE='groovy/resources/genericPythonRunner/requirements.txt'  
    PYTHON_SCRIPT_FILE='groovy/src/genericPythonRunner/testGenericPythonRunner.py'  
}
```

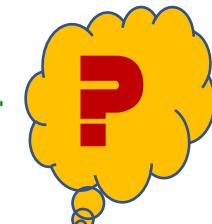
Environment



- COPY-PASTE
- MAINTENANCE

```
pipeline  
{  
    environment  
    {  
        ...  
    }  
    agent  
    {  
        ...  
    }  
    stages  
    {  
        stage('Setup')  
        {  
            ...  
        }  
        stage('Script run')  
        {  
            ...  
        }  
        stage('Run script')  
        {  
            ...  
        }  
    }  
}
```

```
pipeline  
{  
    environment  
    {  
        REPOSITORY_URL='https://github.com/tikalk/pipelineForceCode.git'  
        GIT_REPO_CRED=''  
        BRANCH='master'  
        REQUIREMENTS_FILE='groovy/resources/genericPythonRunner/requirements.txt'  
        PYTHON_SCRIPT_FILE='groovy/src/genericPythonRunner/testGenericPythonRunner.py'  
    }  
    agent  
    {  
        docker  
        {  
            image 'python:2.7'  
            args "-u root"  
        }  
    }  
    stages  
    {  
        stage('Setup')  
        {  
            steps  
            {  
                checkout([  
                    $class: 'GitSCM',  
                    branches: [[name: BRANCH]],  
                    userRemoteConfigs: [[url: "${REPOSITORY_URL}",  
                        credentialsId: "${GIT_REPO_CRED}"]]  
                ])  
            }  
        }  
        stage('Install requirements')  
        {  
            steps  
            {  
                sh "pip install -r ${REQUIREMENTS_FILE}"  
            }  
        }  
        stage('Run script')  
        {  
            steps  
            {  
                script  
                {  
                    sh "python ./${PYTHON_SCRIPT_FILE}"  
                }  
            }  
        }  
    }  
}
```



```

pipeline
{
    agent any
    stages
    {
        stage('Run pipelineForceCode Python script')
        {
            steps
            {
                build job: 'TPL_testGenericPythonRunner',
                parameters:
                    [string(name: 'REPOSITORY_URL',
                           value: 'https://github.com/tikalk/pipelineForceCode.git'),
                     string(name: 'GIT_REPO_CRED',
                           value: ''),
                     string(name: 'BRANCH',
                           value: 'master'),
                     string(name: 'REQUIREMENTS_FILE',
                           value: 'groovy/resources/genericPythonRunner/requirements.txt'),
                     string(name: 'PYTHON_SCRIPT_FILE',
                           value: 'groovy/src/genericPythonRunner/testGenericPythonRunner.py')]

            }
        }
    }
}

```

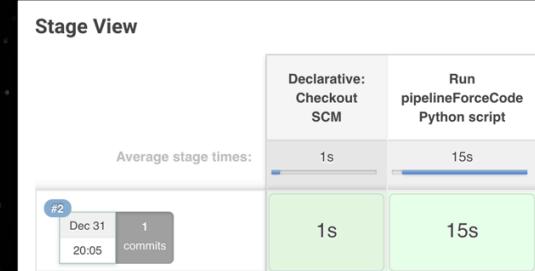


- COPY-PASTE
- MAINTENANCE
- FOLLOW

Run job with parameters

UPSTREAM & DOWNSTREAM

TPL_testGenericPythonRunnerClient



Scheduling project: [TPL_testGenericPythonRunner](#)
Starting building: [TPL_testGenericPythonRunner #](#)



TPL_testGenericPythonRunner

Stage View

	Declarative: Checkout SCM	Setup	Script run
Average stage times:	1s	733ms	6s
#22 Dec 31 20:05 1 commits	887ms	669ms	6s
#21 Dec 31 20:02 1 commits	932ms	715ms	6s
#20 Dec 31 18:02 1 commits	1s	815ms	7s

Started by upstream project "TPL testGenericPythonRunnerClient"

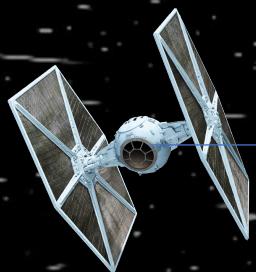
STARWARS
EPISODE IV
A NEW HOPE

JENKINS SHARED LIBRARY



JENKINS SHARED LIBRARY

Shared Library repository



```
tikal-pipelib /opt/tikal/repos/tikal-pipelib
  src
    resources
    tpl
      ci
        gradle
          tplGradleCiPipeline.groovy
        maven
          tplArtifactoryCiPipeline.groovy
          tplBaseCiPipeline.groovy
      python
        runners
          tplBaseGenericPythonRunner.groovy
          tplBaseGenericPythonRunnerWithSteps.groovy
        utils
      vars
        README.md
        tplAdvancedSlackNotification.groovy
        tplBlockDirectManualJobTriggering.groovy
        tplClearWorkspaceAsRoot.groovy
        tplEcho.groovy
        tplFindStringInBuildLog.groovy
        tplGetBuildUserEmailAddress.groovy
        tplGetBuildUserFullName.groovy
        tplGetBuildUserId.groovy
        tplGetChangersList.groovy
        tplGetChangesList.groovy
        tplGetJobLatestSuccessfulBuildNumber.groovy
        tplGetTestSummary.groovy
        tplInDir.groovy
        tplIsBuildStartedByUser.groovy
        tplKeepBuildForever.groovy
        tplPythonRequirements.groovy
        tplPythonScriptRun.groovy
        tplRepositoryDirectoryCheckout.groovy
        tplSetJobEnabledStatus.groovy
        tplSetJobNextBuildNumber.groovy
        tplSetRandomBuildStatus.groovy
```

JENKINS SHARED LIBRARY

Jenkins server configuration form

Manage Jenkins

Configure System
Configure global settings and paths.

Library

Name

tikal-pipelib

Default version

master

Currently maps to revision: 24dc12607ad6

Load implicitly

Allow default version to be overridden

Retrieval method

Modern SCM

Source Code Management

Git

Project Repository

<https://github.com/tikalk/tikal-pipelib.git>



JENKINS SHARED LIBRARY REFERENCE

Option 1: Jenkinsfile with indirect library reference

```
@Library("tikal-pipelib") _  
  
@Library("tikal-pipelib@TheForceAwakens") _
```

Option 2: Jenkinsfile with direct library reference

```
library identifier: 'custom-lib@master', retriever: modernSCM(  
  
[$class: 'GitSCMSource',  
  
remote: 'git@git.mycorp.com:my-jenkins-utils.git',  
  
credentialsId: 'my-private-key'])
```

Option 3: Jenkinsfile with implicit library reference

STAR WARS
EPISODE VI
RETURN OF THE JEDI

**JENKINS
STEPS**



```
def call(String repositoryUrl, String branch, String credentialsId, String relativPath=".") {  
    dir(relativPath)  
    {  
        timeout(time: 5, unit: 'MINUTES')  
        {  
            checkout [  
                $class: 'GitSCM', branches: [[name: branch]],  
                userRemoteConfigs: [[url: repositoryUrl ,credentialsId:credentialsId]]  
            ]  
        }  
    }  
}
```

Code checkout

/vars/tplRepositoryDirectoryCheckout.groovy

```
def call(String requirementsFilePath) {  
    File file = new File( env.WORKSPACE + "/" + requirementsFilePath )  
    if( file.exists() && file.isFile() ) {  
        sh "pip install -r ${pythonRequirements}"  
    }  
}
```

Requirements

/vars/tplPythonRequirements.groovy

```
def call(String scriptFilePath) {  
    File file = new File( env.WORKSPACE + "/" + scriptFilePath )  
    if( file.exists() && file.isFile() ) {  
        sh "python ./${scriptFilePath}"  
    }  
}
```

Script run

/vars/tplPythonScriptRun.groovy



STEPS USAGE

```
@Library("tikal-pipelib")
```

```
pipeline
{
    parameters
    {
        string(name: 'REPOSITORY_URL',
               description: 'URL to GIT repository')
        string(name: 'GIT_REPO_CRED',
               description: 'Credential name of GIT user')
        string(name: 'BRANCH',
               defaultValue: 'master',
               description: 'Name of branch')
        string(name: 'REQUIREMENTS_FILE',
               description: 'Relative path to the requirements file')
        string(name: 'PYTHON_SCRIPT_FILE',
               description: 'Relative path to the python script')
    }
    agent
    {
        docker
        {
            image 'python:2.7'
            args "-u root"
        }
    }
}
```

```
stages
{
    stage('Setup')
    {
        steps
        {
            tplRepositoryDirectoryCheckout(REPOSITORY_URL)
        }
    }
    stage('Script run')
    {
        steps
        {
            tplPythonRequirements(REQUIREMENTS_FILE)
        }
    }
    stage('Run script')
    {
        steps
        {
            tplPythonScriptRun(PYTHON_SCRIPT_FILE)
        }
    }
}
```

```
def call(String repositoryUrl, String branch, String credentialsId, String relativPath="") {
    dir(relativPath)
    {
        timeout(time: 5, unit: 'MINUTES')
        {
            checkout([
                $class: 'GitSCM', branches: [[name: branch]],
                userRemoteConfigs: [[url: repositoryUrl ,credentialsId:credentialsId]]
            ])
        }
    }
}
```

```
def call(String requirementsFilePath) {
    File file = new File( env.WORKSPACE + "/" + requirementsFilePath )
    if( file.exists() && file.isFile() ) {
        sh "pip install -r ${pythonRequirements}"
    }
}
```

```
def call(String scriptFilePath) {
    File file = new File( env.WORKSPACE + "/" + scriptFilePath )
    if( file.exists() && file.isFile() ) {
        sh "python ./${scriptFilePath}"
    }
}
```



STAR WARS
EPISODE VII
THE FORCE AWAKENS

FLOW IMPLEMENTATION



FLOW IMPLEMENTATION BASE CLASS

```
abstract class tplBaseGenericPythonRunner implements Serializable {
```

```
    void run() {
        script.timestamps()
        runImpl()
    }
}
```

Abstract base
class

Entry point

```
        void runImpl() {
            runStage( name: 'Setup', this.&initParams)
            runStage( name: 'Checkout', this.&checkout)
            runStage( name: 'Requirements', this.&requirements)
            runStage( name: 'Script Run', this.&scriptRun)
        }
    }
```

Flow stages
definition

```
    void checkout() {
        script.git credentialsId: gitCredentialsId, url: gitRepoUrl, branch: gitBranch
    }
}
```

Code checkout

```
    void requirements(){
        commonRequirements()
        additionalGroovySetup()
        script.echo "Install Python requirements file '${pythonRequirements}'"
        def file = new File( script.env.WORKSPACE + "/" + pythonRequirements )
        if( file.exists() && file.isFile() ) {
            script.sh "pip install -r ${pythonRequirements}"
        }
    }
}
```

Requirements

```
    void scriptRun(){
        script.echo "Run Python script '${pythonScript}'"
        script.sh "python ./${pythonScript}"
    }
}
```

Script run

FLOW IMPLEMENTATION

EXTENDED CLASS

tikal-pipelib > src > tpl > python > runners > tplGenericPythonRunner.groovy

```
import tpl.python.tplBaseGenericPythonRunner

class tplGenericPythonRunner extends tplBaseGenericPythonRunner {

    tplGenericPythonRunner(script) {
        super(script)
    }
}
```

Extended class

```
@Override
void initParams()
{
    gitCredentialsId = script.env.GIT_TOKEN
    gitRepoUrl = script.env.REPOSITORY
    gitBranch = script.env.BRANCH
    pythonRequirements = script.env.REQUIREMENTS_FILE
    pythonScript = script.env.PYTHON_SCRIPT_FILE
    groovySetupScript = script.params.GROOVY_SCRIPT
}
```

Overwrite parameters

FLOW IMPLEMENTATION

JENKINSFILE

tikal-pipelib-touchstone > examples > pipelineForce > genericPythonScriptRunner > Jenkinsfile-withLibrary

```
@Library("tikal-pipelib") _  
import tpl.python.runners.tplGenericPythonRunner
```

Class import

```
stage('Python script run')  
{  
    steps  
    {  
        script  
        {  
            docker.image("python:2.7").inside('-u root')  
            {  
                def runner = new tplGenericPythonRunner(this)  
                runner.run()  
            }  
        }  
    }  
}
```

Class usage



IMPLEMENT USING LIBRARY & STEPS

```
void checkout() {  
    script.tplRepositoryDirectoryCheckout(gitRepoUrl,gitBranch,gitCredentialsId)  
}  
  
void requirements(){  
    commonRequirements()  
    additionalGroovySetup()  
    script.echo "Install Python requirements file '${pythonRequirements}'"  
    script.tplPythonRequirements(pythonRequirements)  
}  
  
void scriptRun(){  
    script.echo "Run Python script '${pythonScript}'"  
    script.tplPythonScriptRun(pythonScript)  
}
```





A SPARK OF HOPE







SHARED LIBRARY

tikal-pipelib

<https://github.com/tikalk/tikal-pipelib>

SHARED EXPERIENCE

tikal-pipelib-touchstone

<https://github.com/tikalk/tikal-pipelib-touchstone>

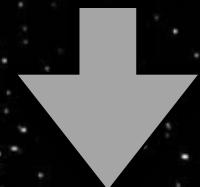
STAR WARS

RETURN OF THE JEDI

THE RETURN OF THE WORKSHOP



BUILD PREPARATIONS



DOCKER BUILD



PUBLISH IMAGE



Workshop

- Make sure the base framework works
- Implement the steps in a library
- Implement the flow in a library
- Merge the steps into the flow



**THANK YOU &
GOOD LUCK!**

