上海商業銀行股份有限公司
新核心系統 INFRA
軟硬體設備建置專案服務

OCP CI/CD
Jenkins pipeline 初版腳本

台灣國際商業機器股份有限公司

OCP Application CI/CD Jenkins 腳本 Sample

*OpenShift 相關設定，請參考文件: Jenkins 導入 openshift-client 外掛。

*Trivy 相關設定，請參考文件: Trivy 離線版安裝。

腳本範例如下：

```
pipeline {
   agent any
   tools {
      oc 'OpenShift CLI'
   }
   environment {
      GIT_REPO_URL = 'https://gitlab.brobridge.lab/ocp-demo/vue-cicd.git'
      GIT_REPO_SSH_URL = 'git@gitlab.brobridge.lab:ocp-demo/vue-cicd.git'
      CONTAINER_REGISTRY = 'hb.k8sbridge.com'
      CONTAINER_PROJECT = 'scsb'
      CONTAINER_NAME = 'demo-vue'
      CICDAPISERVICE_URL = 'http://jenkins.brobridge.lab:5001'
      OPENSHIFT_SERVER = 'lab-ocp'
      OPENSHIFT_PROJECT = 'dev'
      APP_NAME = 'demo-vue'
   }
   stages {
      stage('Declarative: Checkout SCM') { // for display purposes
         steps {
            echo '=============================='
            echo 'Declarative: Checkout  SCM'
            echo '==============================='
            script {
               // 使用 withCredentials 來取得 Git 的帳號密碼
               withCredentials([gitUsernamePassword(credentialsId:
'79206b5c-2aa3-471c-a6fb-36b25eb3cd7e', gitToolName: 'git-tool')]) {
                  // 使用 GitSCM 來 checkout
                  checkout([
                     $class: 'GitSCM',
                     branches: [[name: 'dev']],
                     userRemoteConfigs: [[url: "${GIT_REPO_URL}",
credentialsId: '79206b5c-2aa3-471c-a6fb-36b25eb3cd7e']]
                  ])
                  // 取得 Git 的 commit hash
```

```
                        env.GIT_COMMIT = sh(script: 'git rev-parse HEAD', 
returnStdout: true).trim()
                        // 取得 Git 的 commit 簡短 hash
                        env.GIT_COMMIT_SHORT = sh(script: "git rev-parse --
short ${GIT_COMMIT}", returnStdout: true).trim()
                    }
                    // 設定 IMAGE 的版本
                    env.APP_VERSION = env.GIT_COMMIT_SHORT
                }
                echo "GIT_COMMIT: ${env.GIT_COMMIT}"
                echo "APP_VERSION: ${env.APP_VERSION}"
            }
        }
        stage('Build') {
            steps {
                echo '================================'
                echo 'Build'
                echo '================================='
                // 透過 Jenkins 的 withCredentials 來取得 Harbor 的帳號密碼
                withCredentials([usernamePassword(credentialsId: 'e16751f7-
c6b3-489d-9640-ea70719d02d2', passwordVariable: 'HARBOR_PASSWORD', 
usernameVariable: 'HARBOR_USER')]) {
                    // podman login
                    sh "echo '$HARBOR_PASSWORD' | podman login 
${CONTAINER_REGISTRY} -v -u ${HARBOR_USER} --password-stdin"
                }
                // 直接在 Jenkins 系統上執行指令
                script {
                    // 執行 podman build 指令
                    sh " podman build -t 
${CONTAINER_REGISTRY}/${CONTAINER_PROJECT}/${CONTAINER_NAME}:${APP_VERSION
} ."
                    env.HASH = sh(returnStdout: true, script: "podman inspect
-f '{{.Digest}}' 
${CONTAINER_REGISTRY}/${CONTAINER_PROJECT}/${CONTAINER_NAME}:${APP_VERSION
}").trim()
                }
                println(env.HASH)
                sh "podman logout ${CONTAINER_REGISTRY}"
            }
        }
        stage('Publish to harbor') {
            steps {
                echo '================================'
                echo 'Publish to harbor'
                echo '================================='
                // 透過 Jenkins 的 withCredentials 來取得 Harbor 的帳號密碼
                withCredentials([usernamePassword(credentialsId: 'e16751f7-
c6b3-489d-9640-ea70719d02d2', passwordVariable: 'HARBOR_PASSWORD',
usernameVariable: 'HARBOR_USER')]) {
                    // podman login
                    sh "echo '$HARBOR_PASSWORD' | podman login
${CONTAINER_REGISTRY} -v -u ${HARBOR_USER} --password-stdin"
                }
                script {
                    // 取得 RepoDigest
                    def repoDigest = sh(returnStdout: true, script: "podman
inspect -f '{{index .RepoDigests 0}}'
${CONTAINER_REGISTRY}/${CONTAINER_PROJECT}/${CONTAINER_NAME}:${APP_VERSION
}").trim()
                    //  使用 podman push RepoDigest
                    sh "podman push ${repoDigest}"
                    // push image 到 harbor
                    sh "podman push
${CONTAINER_REGISTRY}/${CONTAINER_PROJECT}/${CONTAINER_NAME}:${APP_VERSION
}"
                    withCredentials([file(credentialsId: 'cosign-private-
key', variable: 'COSIGN_PRIVATE_KEY'), string(credentialsId: 'cosign-
password', variable: 'COSIGN_PASSWORD')]) {
                        println(repoDigest)
                        // 使用 cosign 簽名
                        sh "cosign sign --key '${COSIGN_PRIVATE_KEY}' --tlog-
upload=false --allow-insecure-registry=true ${repoDigest}"
                    }
                    sh "podman logout ${CONTAINER_REGISTRY}"
                }
            }
        }
        stage('Publish YAML to Nexus') {
            steps {
                script {
                    // 指定上傳的檔案
                    def yamlFiles = findFiles(glob: 'ocp-yaml/*.yaml')
                    if (yamlFiles.length > 0) {
                        yamlFiles.each { file ->
                            echo "Uploading ${file.name} to Nexus..."
                            // 上傳到 Nexus
                            // credentialsId: Nexus 的帳號密碼
                            // nexusUrl: Nexus 的 URL
                            // nexusVersion: Nexus 的版本
                            // protocol: 使用的協定
                            // repository: 上傳的 repository
                            // groupId: 上傳的 groupId
                            // version: 上傳的版本
                            // artifacts: 上傳的檔案
                            nexusArtifactUploader(
                                credentialsId: '5a189b4b-734a-454f-b132-
ae5fe0e33136',
                                nexusUrl: 'nexus.brobridge.lab',
                                nexusVersion: 'nexus3',
                                protocol: 'https',
                                repository: 'ocp-yaml-repository',
                                groupId: 'com.yourcompany.yaml',
                                version: '1.0.0',
                                artifacts: [
                                    [artifactId: file.name.replace('.yaml', ''),
                                    classifier: '',
                                    file: file.path,
                                    type: 'yaml']
                                ]
                            )
                        }
                    } else {
                        error 'No YAML files found in ocp-yaml directory.'
                    }
                }
            }
        }
        stage('Download YAMLs from Nexus') {
            steps {
                script {
                    // 從 Nexus 下載檔案
                    // 因 Nexus 上傳檔案需指定版本，所以下載時也需指定版本
                    def filesToDownload = [
                        'https://nexus.brobridge.lab/repository/ocp-yaml-
repository/com/yourcompany/yaml/deployment/1.0.0/deployment-1.0.0.yaml',
                        'https://nexus.brobridge.lab/repository/ocp-yaml-
repository/com/yourcompany/yaml/route/1.0.0/route-1.0.0.yaml',
                        'https://nexus.brobridge.lab/repository/ocp-yaml-
repository/com/yourcompany/yaml/service/1.0.0/service-1.0.0.yaml'
                    ]
                    // credentialsId: Nexus 的帳號密碼
                    def credentialsId = '5a189b4b-734a-454f-b132-ae5fe0e33136'
                    // 下載的目錄
                    def downloadDirectory = 'downloaded-files'
                    // 建立目錄
                    sh "mkdir -p ${downloadDirectory}"
                    withCredentials([usernamePassword(credentialsId:
credentialsId, passwordVariable: 'NEXUS_PASSWORD', usernameVariable:
'NEXUS_USERNAME')]) {
                        // 下載檔案
                        filesToDownload.each { fileUrl ->
                            // 取得檔案名稱
                            def fileName = fileUrl.tokenize('/').last()
                            // 下載的檔案路徑
                            def destinationFile =
"${downloadDirectory}/${fileName}"
                            echo "Downloading ${fileUrl} to
${destinationFile}..."
                            // 使用 curl 下載檔案
                            sh " curl -u $NEXUS_USERNAME:$NEXUS_PASSWORD -o
${destinationFile} ${fileUrl}"
                        }
                    }
                    echo "All files downloaded to ${downloadDirectory}."
                }
            }
        }
        stage('Trivy scan image') {
            steps {
                echo '================================'
                echo 'Trivy scan'
                echo '================================'
                sh 'mkdir -p reports'
                // 使用 Trivy 進行掃描 image
                // --skip-db-update: 跳過更新 Trivy 的資料庫
                // --skip-java-db-update: 跳過更新 Java 的資料庫
                // --offline-scan: 離線掃描
                // --skip-check-update: 跳過檢查更新
                // --format template: 輸出格式為 template
                // --template '@/usr/local/share/trivy/templates/html.tpl': 使
用 HTML 的格式
                // template 要有讀的權限
                // -o reports/trivy-image-report.html: 輸出檔案為 reports/trivy-
image-report.html
                sh "trivy image --skip-db-update \
                    --skip-java-db-update \
                    --offline-scan \
                    --skip-check-update \
                    --format template \
                    --template '@/usr/local/share/trivy/templates/html.tpl' \
                    -o reports/trivy-image-report.html \
${CONTAINER_REGISTRY}/${CONTAINER_PROJECT}/${CONTAINER_NAME}:${APP_VERSION
}"
                // 上傳報告到 Jenkins
                publishHTML([
                    allowMissing: true,
                    alwaysLinkToLastBuild: false,
                    keepAll: true,
                    reportDir: 'reports',
                    reportFiles: 'trivy-image-report.html',
                    reportName: 'Trivy Image Report',
                    reportTitles: 'Trivy Image Report',
                    useWrapperFileDirectly: true
                ])
            }
        }
        stage('Trivy scan kubernetes YAML') {
            steps {
                script {
                    // 使用 Trivy 進行掃描 kubernetes YAML
                    // yaml 內容中有 {{HARBOR_URL}} 的地方會被取代成
harbor.example.com/demo-vue:latest
                    // 但 {{HARBOR_URL}} 並不是真實的 URL，所以掃描時會有錯誤，所以需
要先取代成真實的 URL，掃描完後再還原
                    sh "sed -i 's#{{HARBOR_URL}}#harbor.example.com/demo-
vue:latest#g' ocp-yaml/deployment.yaml"
                    // --skip-check-update: 跳過檢查更新
                    // --severity HIGH,CRITICAL: 只顯示 HIGH 和 CRITICAL 的問題
                    // --format template: 輸出格式為 template
                    // --template
'@/usr/local/share/trivy/templates/html.tpl': 使用 HTML 的格式
                    // template 要有讀的權限
                    // -o reports/report_yaml.html: 輸出檔案為
reports/report_yaml.html
                    sh "trivy config --format template \
                        --skip-check-update \
                        --severity HIGH,CRITICAL \
                        --template
'@/usr/local/share/trivy/templates/html.tpl' \
                        -o reports/report_yaml.html \
                        ./ocp-yaml/"
                    sh 'git checkout -- ocp-yaml/deployment.yaml'
                }
```

```
                // 上傳報告到 Jenkins
                publishHTML([
                    allowMissing: true,
                    alwaysLinkToLastBuild: false,
                    keepAll: true,
                    reportDir: 'reports',
                    reportFiles: 'report_yaml.html',
                    reportName: 'Trivy YAML Report',
                    reportTitles: 'Trivy YAML Report',
                    useWrapperFileDirectly: true
                ])
            }
        }
        stage('Deploy to OCP') {
            steps {
                echo '================================'
                echo 'Deploy to OCP'
                echo '================================='
                // 修改 deployment.yaml 的 IMAGE 版本
                script {
                    // 參照 sed 's#{{HARBOR_URL}}#hb.k8sbridge.com/scsb/demo-
vue:dev#g' deployment.yaml
                    sh "sed -i
's#{{HARBOR_URL}}#${CONTAINER_REGISTRY}/${CONTAINER_PROJECT}/${CONTAINER_N
AME}:${APP_VERSION}#g' ocp-yaml/deployment.yaml"
                }
                script {
                    // 指定 Openshift Cluster Server，需先在系統設定 Openshift
Cluster Server
                    openshift.withCluster(OPENSHIFT_SERVER) {
                        // 指定 Openshift Project 名稱，需先有針對該 Project 的權限
                        openshift.withProject(OPENSHIFT_PROJECT) {
                            //openshift.verbose() // 開啓 verbose 模式
                            // 檢查是否存在 deployments 和 pods
                            deploymentsExist =
openshift.selector('deployments', "${APP_NAME}").exists()
                            podsExist = openshift.selector('pods', [app:
"${APP_NAME}"]).exists()
                            //openshift.verbose(false) // 關閉 verbose 模式
                            // 部署應用程式
                            deployment = openshift.raw('apply', '-f', 'ocp-
yaml/deployment.yaml')
                            deployment = openshift.raw('apply', '-f', 'ocp-
yaml/service.yaml')
```

```
                            deployment = openshift.raw('apply', '-f', 'ocp-
yaml/route.yaml')
                        }
                    }
                }
            }
        }
        //印出 Route URL
        stage('Print Route URL') {
            steps {
                script {
                    // 指定 Openshift Cluster Server
                    openshift.withCluster(OPENSHIFT_SERVER) {
                        // 指定 Openshift Project 名稱
                        openshift.withProject(OPENSHIFT_PROJECT) {
                            // 取得 Route URL
                            routeURL = openshift.raw('get', 'route',
"${APP_NAME}", '-o', 'jsonpath={.spec.host}')
                            echo "routeURL: ${routeURL.out}"
                        }
                    }
                }
            }
        }
    }
}
```