Qualys IaC Security Integration with Jenkins

In the existing Continuous Integration and Continuous Deployment (CICD) environment, the security scans are conducted on cloud resources after deployment. As a result, you secure your cloud resources post-deployment to respective Cloud accounts.

With an introduction of the Infrastructure as Code (IaC) security feature by Qualys CloudView, you can now secure your IaC templates before the cloud resources are deployed in your cloud environments. The IaC Security feature will help you shift cloud security and compliance posture to the left, allowing evaluation of cloud resources for misconfigurations much early during the development phase.

CloudView offers integration with Jenkins to scan and secure your IaC templates using the Jenkins pipeline job. It continuously verifies security misconfigurations against CloudView controls and displays the misconfigurations for each run. With a continuous visibility of the security posture of your IaC Templates at Jenkins pipeline you can plan for remediation to stay secure post deployment.

For supported templates, other integrations, and features of Cloud IaC Security, refer to CloudView User Guide and CloudView API User Guide.

Scanning IaC Templates at Jenkins

The Jenkins integration allows you to perform IaC scans using pipeline job. We provide you with a pipeline job and options that you can configure to run based on various triggers.

You can perform an IaC scan on either of the following:

- the entire git repository.

- only the templates that were newly added / updates to the branch.

The results are generated on the build console that provides you with proactive visibility into the security of your IaC templates residing in Git repositories.

Pre-requisite

- Ensure that you have a valid docker pipeline plugin installed.

- Ensure to configure environment variables used in the pipeline script before you run the pipeline job in Jenkins. For more info, refer Configure Environment Variables.

- To auto-trigger a Jenkins pipeline job, ensure that you install a specific Source Code Management (SCM) plugin, e.g., Bitbucket plugin, Bitbucket Server Integration. For auto-trigger, the pipeline job must contain a Jenkins file.

- Docker must be installed on the Jenkins agent node.

- Ensure that you have a valid Qualys CloudView Security Assessment app subscription.

Let us see the quick workflow:

Configure Environment Variables

Configure Git Repositories

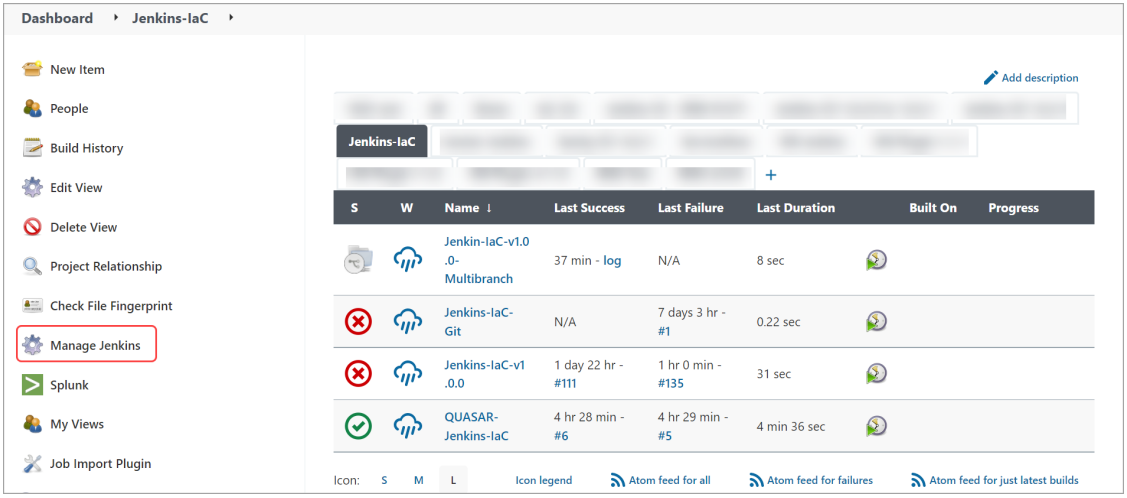
Configure Pipeline Job

View Scan Output

Configure Environment Variables

To add new environment variables,

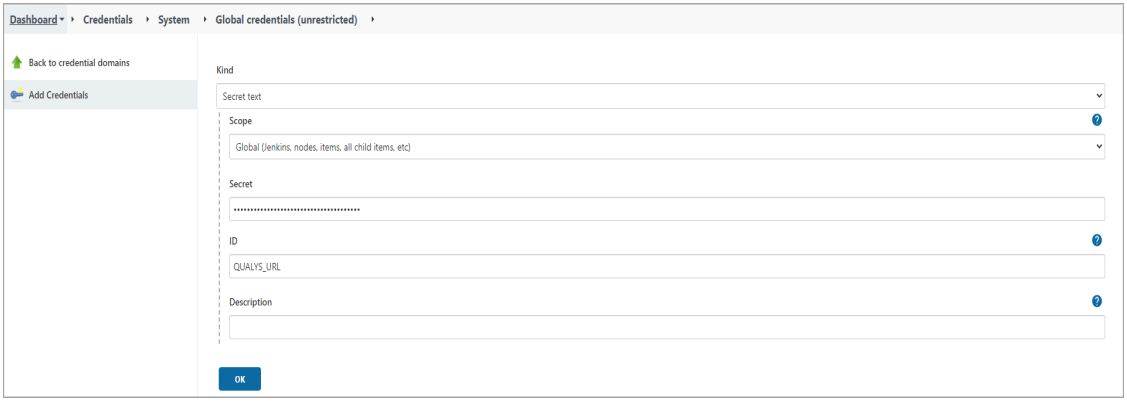
1. On the Jenkins console, go to **Manage Jenkins** > **Manage Credentials**.



2. Click any row and go to **Add Credentials**.

3. Select the **Secret Text** option from the **Kind** drop-down menu and enter the required **Secret** and **ID**.

Note: In the **Secret** field, add the actual values for URL, username, and password, and in the **ID** field, add variable names such as QUALYS_URL, QUALYS_USERNAME, and QUALYS_-PASSWORD to identify the secrets. Use these variable names (ID) in the script and not the actual values.



4. Click **OK**.

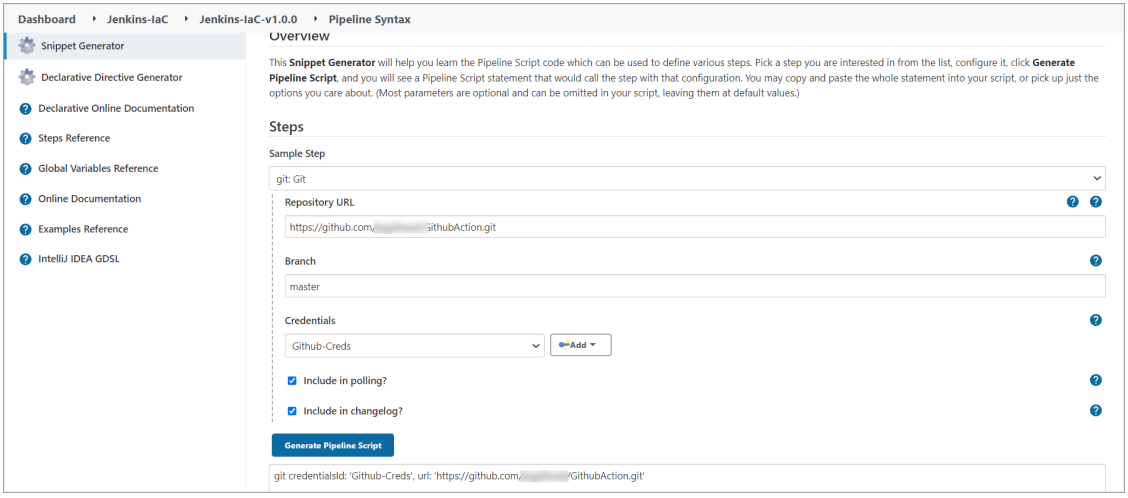
The newly added credentials appear in the **Dashboard** > **Credentials** list.

Variable	Description
QUALYS_URL	Qualys platform URL. To know about your Qualys platform URL, click here.
QUALYS_USERNAME	Qualys username
QUALYS_PASSWORD	Qualys password

Configure Git Repositories

To configure Git repositories,

1. Select the Jenkins Pipeline Project and click **Configure**.
2. Scroll to the end and click **Pipeline Syntax**.
3. Select **git: Git** from the drop-down menu.
4. Add **Repository URL**, **Branch**, and **Credentials** in the respective fields.
5. Click **Generate Pipeline Script**.
6. Copy this generated pipeline script and use it while configuring the pipeline job.



Configure Pipeline Job

You can use the Jenkins pipeline job to scan and secure the IaC templates.

1. Create a Jenkins pipeline project and place the required script in the pipeline project.

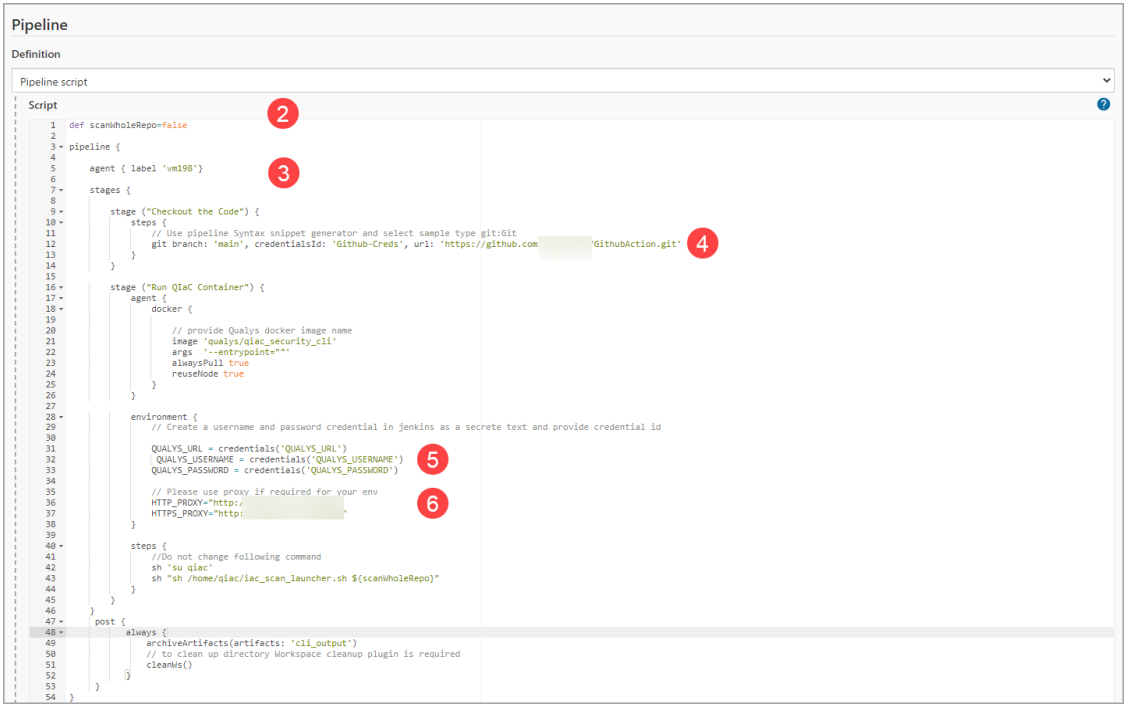
2. If you want to scan the entire repository, set the value for **scanWholeRepo** as True. If you want to scan only the changed / newly added files, set the value for **scanWholeRepo** as False.

3. To run this job on the required agent, add the agent details in the script and click **Save**.

4. Paste the generated pipeline script copied earlier from step 6 in Configure Git Repositories.

5. Add the environment variables created in step 3 in Configure Environment Variables.

6. If you are connected to a proxy server, mention the HTTP Proxy details in the script.



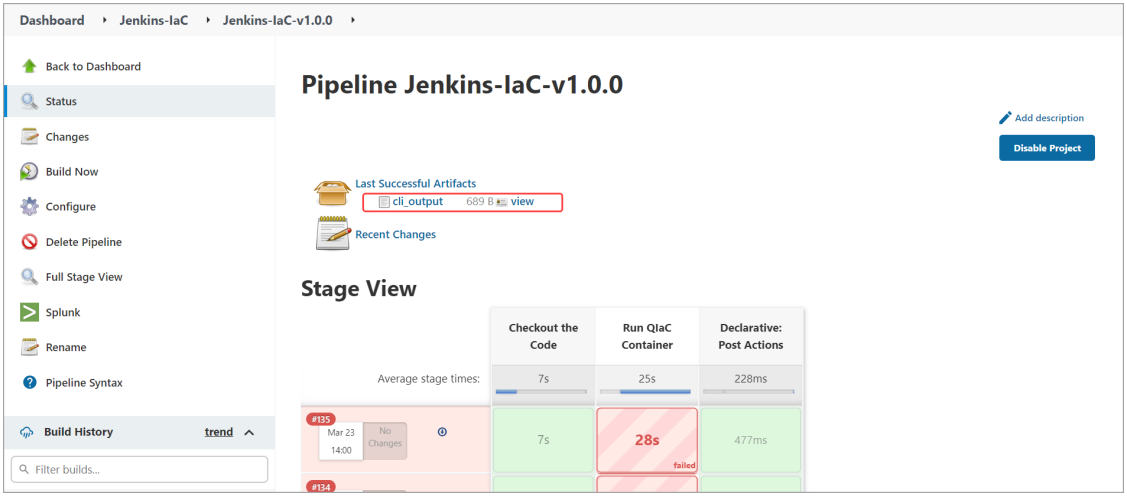
Sample Script

```
def scanWholeRepo=false
pipeline {
    agent { label 'vm198'}
    stages {
        stage ("Checkout the Code") {
            steps {
               // Use pipeline Syntax snippet generator and select sample
type git:Git
                git branch: 'main', credentialsId: 'Github-Creds', url:
'https://github.com/xxxxxx/GithubAction.git'
            }
        }
        stage ("Run QIaC Container") {
            agent {
                docker {
                    // provide Qualys docker image name
                    image 'qualys/qiac_security_cli'
                    args  '--entrypoint=""'
                    alwaysPull true
                    reuseNode true
                }
            }
            environment {
               // Create a username and password credential in jenkins as
a secrete text and provide credential id
                QUALYS_URL = credentials('QUALYS_URL')
                 QUALYS_USERNAME = credentials('QUALYS_USERNAME')
                QUALYS_PASSWORD = credentials('QUALYS_PASSWORD')
                // Please use proxy if required for your env
                HTTP_PROXY="http://xx.xxx.xx.xx:xxxx"
                HTTPS_PROXY="http://xx.xxx.xx.xx:xxxx"
            }
            steps {
                //Do not change following command
                sh 'su qiac'
                sh "sh /home/qiac/iac_scan_launcher.sh ${scanWholeRepo}"
            }
        }
    }
     post {
           always {
               archiveArtifacts(artifacts: 'cli_output')
             // to clean up directory Workspace cleanup plugin is required
               cleanWs()
           }
     }
}
```

View Scan Output

At the end of the job, the Jenkins pipeline creates the artifact file.

Go to **Status** and click **view** to view the scan report for a selected pipeline job.



To view the scan report in detail, go to **Console Output**.