

# Jenkins

Guia de implementação

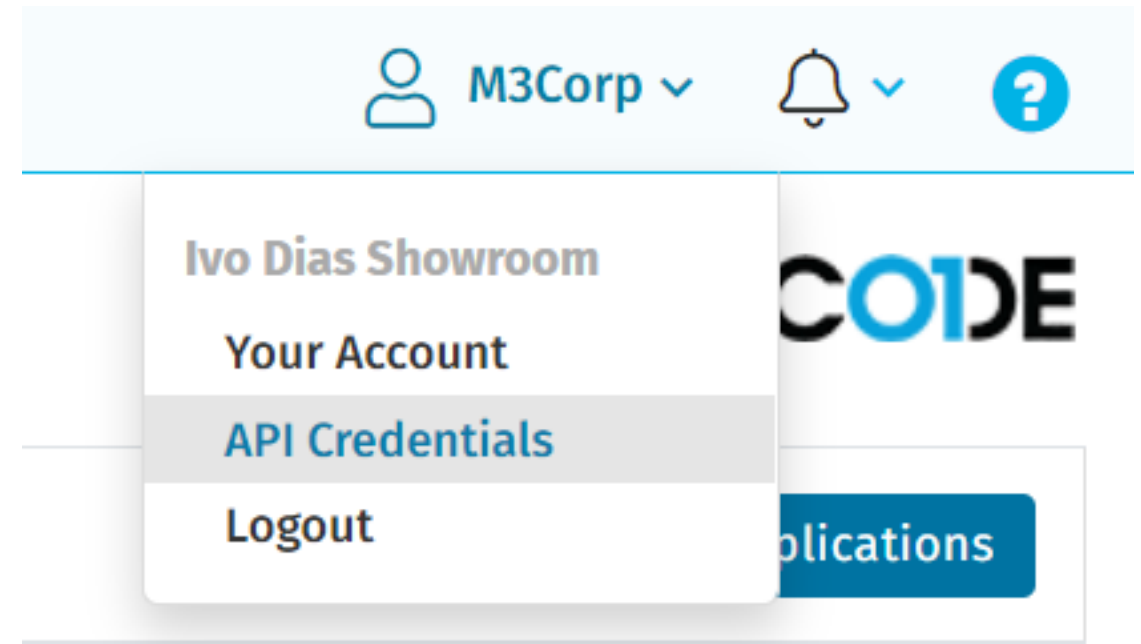
VERACODE



# Credenciais API



- Nosso primeiro passo é obter as credenciais no portal da Veracode
- O recomendado é a criação de um usuário API específico para essas integrações
- Conforme a imagem ao lado, precisamos apenas clicar no nosso usuário no canto superior esquerdo e selecionamos a opção de credenciais



# Credenciais API



- Vamos precisar do ID e da Secret Key para fazer as integrações
- Por padrão, essas credenciais duram 1 ano, mas conforme a imagem é possível revoga-las a qualquer momento

## Credentials Details

[Generate API Credentials](#)[Revoke API Credentials](#)

**ID:**

\*\*\*\*\*

**Secret Key:**

\*\*\*\*\*

**Created:** 13 Jul 2021 @ 9:21 am EDT

**Expires:** 13 Jul 2022 @ 9:21 am EDT

# Armazenando credenciais API



- Dentro do seu projeto no Jenkins, iremos clicar no “Pipeline Syntax”
- Em **Sample Step** iremos abrir a lista e clicar em **“withCredentials: Bind credentials to variables”**
- Nos campos “Username/Password Variables” colocar o nome das variáveis que serão utilizadas para serem chamadas no pipeline
- Na seção “bindings”, clique na opção **Add > Username and Password(separated)**

Steps

Sample Step **withCredentials: Bind credentials to variables**

Secret values are masked on a best-effort basis to prevent accidental disclosure. See the inline help for details and usage guidelines.

Bindings

**Username and password (separated)**

Username Variable

Password Variable

Credentials **ddd33d978c971b0f014340e06dfc9bf/\*\*\*\*\* (API ID/Key fr** **Add**

**Delete**

**Add**



# Armazenando credenciais API

- No campo **Username** colocar a **API ID** da Veracode
- No campo **Password** colocar a **API SECRET KEY** da Veracode
- Informar um “ID” qualquer apenas para identificação do Jenkins
- Description é um campo opcional para informações caso desejar.

The screenshot shows the 'Jenkins Credentials Provider: Jenkins' interface. Under the 'Add Credentials' section, the following fields are visible:

- Domain:** Global credentials (unrestricted)
- Kind:** Username with password
- Scope:** Global (Jenkins, nodes, items, all child items, etc)
- Username:** veracode\_username
- Password:** (masked with dots)
- ID:** veracode-credentials
- Description:** Veracode API ID and key

At the bottom, there are 'Add' and 'Cancel' buttons.

# Obtendo o Plug-in

- Navegamos até a parte de “Gerenciar Plugins”
- Pesquisar na aba “Disponíveis” por Veracode e instalar o Veracode Scan
- Após instalação, checar na mesma página de “Gerenciar Plugins” o Veracode Scan instalado

## System Configuration



Configurar o sistema  
Configurar opções globais e caminhos



Global Tool Configuration  
Configure tools, their locations and automatic installers.



Gerenciar plugins  
Adiciona, remove, desabilita e habilita plugins que podem incrementar as funcionalidades do Jenkins.  
🔴 Atualizações disponíveis



Gerenciar nós  
Adiciona, remove, controla e monitora o vários nós

veracode					
Atualizações	Disponíveis	Instalados	Avançado		
Habilitar	Nome ↓	Versão	Versão anterior instalada	Desinstalar	
<input checked="" type="checkbox"/>	JAXB plugin JAXB packaging for more transparent Java 9+ compatibility	2.3.0.1		<button>Desinstalar</button>	
<input checked="" type="checkbox"/>	Oracle Java SE Development Kit Installer Plugin Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	1.5		<button>Desinstalar</button>	
<input checked="" type="checkbox"/>	Struts Library plugin for DSL plugins that need names for Jenkins objects.	1.23		<button>Desinstalar</button>	
<input checked="" type="checkbox"/>	Veracode Scan The official, fully supported Veracode plugin for Jenkins. To learn more about this plugin, please go to the Veracode Help Center.	21.9.16.0		<button>Desinstalar</button>	

# Veracode Upload and Scan



- Novamente dentro do projeto, iremos navegar até o Pipeline Syntax
- Agora na lista do “Sample Step”, iremos selecionar veracode: Upload and Scan with Veracode Pipeline

## Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

## Steps

### Sample Step

veracode: Upload and Scan with Veracode Pipeline

veracode

Application Name

# Veracode Upload and Scan



Iniciaremos a configuração, informando:

1. Qual é o nome do Perfil de Aplicação que o scan estará associado na Veracode.
2. Se queremos criar o Perfil da Aplicação, caso ele ainda não exista na Veracode.
3. Se queremos associar essa aplicação a um Time, para restringir os acessos de quem poderá visualizar os resultados para essa aplicação,
4. Informar qual é o nível de criticidade dessa aplicação para o seu negócio

The screenshot shows the Veracode configuration interface with the following fields and options:

- Application Name:** A text input field containing "Java-App-Project-1", which is highlighted with a red rectangular box.
- Create Application:** A checkbox that is checked, with the label "Create Application".
- Team Name:** A text input field containing "team1".
- Business Criticality:** A dropdown menu with the following options: "Very High", "High", "Medium", "Low", and "Very Low". The "Very High" option is selected and highlighted in blue. A red arrow points to this dropdown menu.





# Veracode Upload and Scan

1. Caso deseje realizar um SAST - Sandbox Scan, selecione o checkbox “**create Sandbox**” e todo o job então será referente ao Sandbox criado
2. Em “**Scan Name**” podemos passar variáveis ambiente do Jenkins conforme o exemplo ao lado
3. Em “**Include Filepaths Pattern**” iremos informar o PATH exato de onde está a nossa aplicação empacotada após compilação (.jar, .war, .zip, .apk, etc...)

No exemplo ao lado, incluímos **\*\*/\*\*** para que seja pesquisado em todo diretório do projeto Jenkins o arquivo **.WAR**

☐ Create Sandbox ?

Scan Name ?

`${BUILD_TIMESTAMP} - ${BUILD_NUMBER}`

Upload ?

Include Filepaths Pattern

`**/*.war`

Enter the filepaths of the files to upload for scanning, represented as a comma-separated list of ant-style include patterns relative to the job's workspace root directory.

Patterns are case-sensitive. Patterns that include commas because they denote filepaths that contain commas need to replace the commas with a wildcard character.

If no filepaths are provided, all files in the job's workspace root directory are included.

See <http://ant.apache.org/manual/dirtasks.html> for more info.

Exclude Filepaths Pattern ?

# Veracode Upload and Scan



1. Se desejar que a esteira **aguarde** o resultado e análise pela Veracode completar, clique no checkbox **“Wait for Scan to Complete”**.

**Detalhe:** essa opção pode quebrar sua esteira dependendo do resultado final e da política de segurança atrelada a aplicação

2. Caso queira **Deletar** scans que não foram completados anteriormente por algum motivo, selecione o checkbox **Delete Incomplete Scan**
3. Informe o nome das variáveis que criamos anteriormente no **“Credentials Binding”**
4. Selecione o checkbox **“Fail Job”** se quiser quebrar a esteira se algum problema ocorrer no upload and scan da Veracode,

The screenshot shows the configuration interface for the Veracode Jenkins plugin. It includes several sections with checkboxes and input fields:

- Wait for Scan to Complete:** A checked checkbox with a help icon. Below it is a text field for "Maximum Wait Time (in minutes)" containing the value "60".
- Delete Incomplete Scan:** A checked checkbox with a help icon.
- API ID:** A text field containing the placeholder "VERACODE\_API\_ID".
- API Key:** A text field containing the placeholder "VERACODE\_API\_KEY".
- Fail Job:** A checked checkbox with a help icon. Below it is a detailed instruction: "Select the checkbox if you want the entire Jenkins job to fail if the upload and scan with Veracode action fails. If you do not select this option and the upload and scan with Veracode action fails, the Jenkins job completes and the failure is logged, but you do not receive any notification of the failure."

# Veracode Upload and Scan



1. Selecionar a opção “**Debug**” caso queira receber mais informações durante step da Veracode
2. Por fim, clique em **Generate Pipeline Script** para que possamos integrar o script dentro do JenkinsFile

Copy Output Remote Files to Controller

☐ When a remote machine performs the build, the output files are copied to controller (not recommended).

Debug

☒ Run in debug mode.

Select the checkbox to display additional information in the console output window.

☐ Connect using proxy

**Generate Pipeline Script**

```
veracode applicationName: "", canFailJob: true, criticality: 'VeryHigh', deleteIncompleteScan: true, fileNamePattern: "", replacementPattern: "", sandboxName: "", scanExcludesPattern: "", scanIncludesPattern: "", scanName: '${BUILD_TIMESTAMP} - ${BUILD_NUMBER}', teams: "", timeout: 60, uploadIncludesPattern: '**/*.war', vid: 'VERACODE_API_ID', vkey: 'VERACODE_API_KEY', waitForScan: true
```



# Veracode Upload and Scan

Dentro do seu JenkinsFile, todo script gerado pelo plugin deve estar dentro do Credentials Binding “withCredentials”

Exemplo de script **SAST Policy Scan** integrado no JenkinsFile:

```
stage(' Veracode SAST - Policy Scan'){
    withCredentials([usernamePassword(credentialsId: 'veracode-credentials', passwordVariable: 'VERACODE_API_KEY', usernameVariable:
'VERACODE_API_ID')]) {

        veracode applicationName: 'Java-App-Project-1',
        canFailJob: true,
        createProfile: true,
        criticality: 'VeryHigh',
        deleteIncompleteScan: true,
        scanName: '${BUILD_TIMESTAMP} - ${BUILD_NUMBER}',
        timeout: 60,
        uploadIncludesPattern: '**/*.war',
        vid: '${VERACODE_API_ID}',
        vkey: '${VERACODE_API_KEY}',
        waitForScan: true
    }
}
```

# Mais informações



- Para mais detalhes e review dessas informações na documentação oficial, acessar:

[https://docs.veracode.com/r/t\\_configure\\_jenkins](https://docs.veracode.com/r/t_configure_jenkins)

- Demo utilizando o Plugin com um projeto Freestyle do Jenkins

<https://www.youtube.com/watch?v=7cBJYtNqe-w>



# Sem plugin

Caso não queira instalar o Plugin da Veracode no seu Jenkins, podemos trabalhar com o **API Wrapper**. Ele nos auxilia em diversas automações, inclusive no Upload and Scan que vimos anteriormente.

Para fazer o download basta seguir conforme esse exemplo:

```
stage('Veracode - Download Java Wrapper') {  
    sh 'rm -rf veracode-wrapper-api.jar'  
    sh 'wget -q -O veracode-wrapper-api.jar https://search.maven.org/remotecontent?filepath=com/veracode/vosp/api/wrappers/vosp-api-wrappers-java/"${VERACODE_WRAPPER_VERSION}"/vosp-api-wrappers-java-"${VERACODE_WRAPPER_VERSION} ".jar'  
    sh 'ls -l | grep veracode-wrapper-api.jar'  
}
```



# Sem plugin

Para automatizar a análise SAST, basta chamar o API Wrapper e informar os parâmetros desejados conforme exemplo abaixo:

```
stage('Veracode SAST - Policy Scan') {  
    withCredentials([usernamePassword(credentialsId: 'veracode_credentials', passwordVariable: 'VERACODE_KEY', usernameVariable:  
'VERACODE_ID')]) {  
        sh (""" java -jar veracode-wrapper-api.jar \  
            -vid "${VERACODE_ID}" \  
            -vkey "${VERACODE_KEY}" \  
            -action uploadandscan \  
            -appname "Java-App-Project-1" \  
            -createprofile false \  
            -filepath /target/SNAPSHOT.war \  
            -deleteincompletescan true \  
            -scanpollinginterval 60 \  
            -version "${BUILD_TIMESTAMP} - ${BUILD_NUMBER}" \  
            -scantimeout 60  
            """)  
    }  
}
```

# Mais informações




- Para mais detalhes sobre o API Wrapper acessar os links abaixo:

Sobre os Wrappers: [https://docs.veracode.com/r/c\\_about\\_wrappers](https://docs.veracode.com/r/c_about_wrappers)

Java API Wrapper: [https://docs.veracode.com/r/t\\_working\\_with\\_java\\_wrapper](https://docs.veracode.com/r/t_working_with_java_wrapper)

Parâmetros do Upload and Scan: [https://docs.veracode.com/r/r\\_uploadandscan?section=r\\_uploadandscan](https://docs.veracode.com/r/r_uploadandscan?section=r_uploadandscan)





Entre em contato com a nossa  
equipe para mais detalhes:  
[prevendas@m3corp.com.br](mailto:prevendas@m3corp.com.br)

SECURING THE SOFTWARE  
THAT POWERS YOUR WORLD

---

**VERAC**CODE