



Passo a passo instalação TOTVS Agro Connector (Linux)

1. Instalação Docker Ubuntu

- Acessar a máquina do cliente.
- Os passos serão feitos de acordo com manual de instalação do link: <https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository>
- Executar os seguintes comandos no terminal:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

2. Pós instalação Docker Ubuntu

- Os passos serão feitos de acordo com manual de instalação do link: <https://docs.docker.com/engine/install/linux-postinstall/>
- Executar os seguintes comandos:

```
sudo groupadd docker
sudo usermod -aG docker $USER
```

MENSAGEM DE ALERTA: If you're running Linux in a virtual machine, it may be necessary to restart the virtual machine for changes to take effect/Se você está em uma máquina virtual Linux, é necessário reiniciá-la para as mudanças serem executadas.

Para isso, executar o seguinte comando:

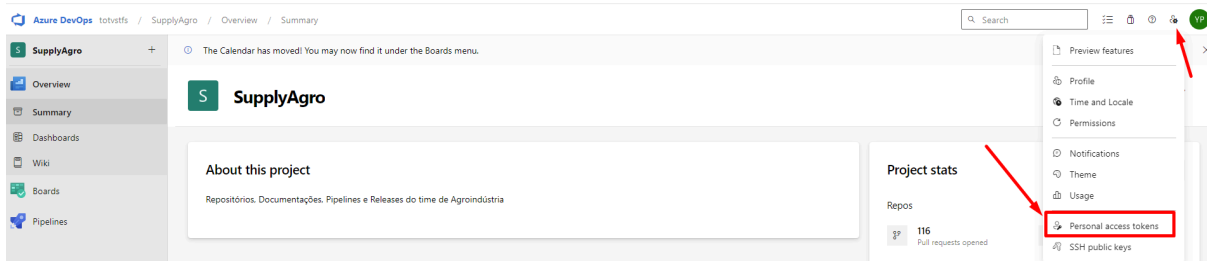
```
sudo reboot now
```

```
newgrp docker
docker run hello-world
sudo systemctl enable docker.service
sudo systemctl enable containerd.service
```

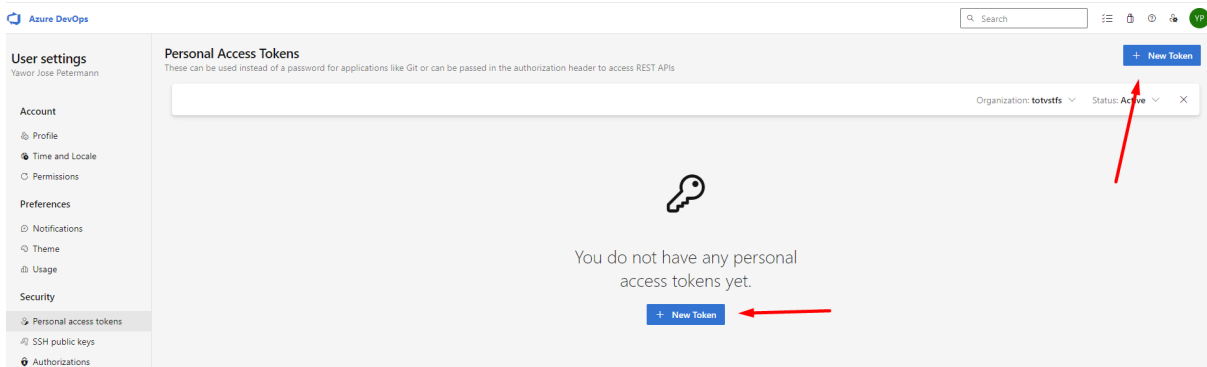


3. Criação Personal Access Token no Azure:

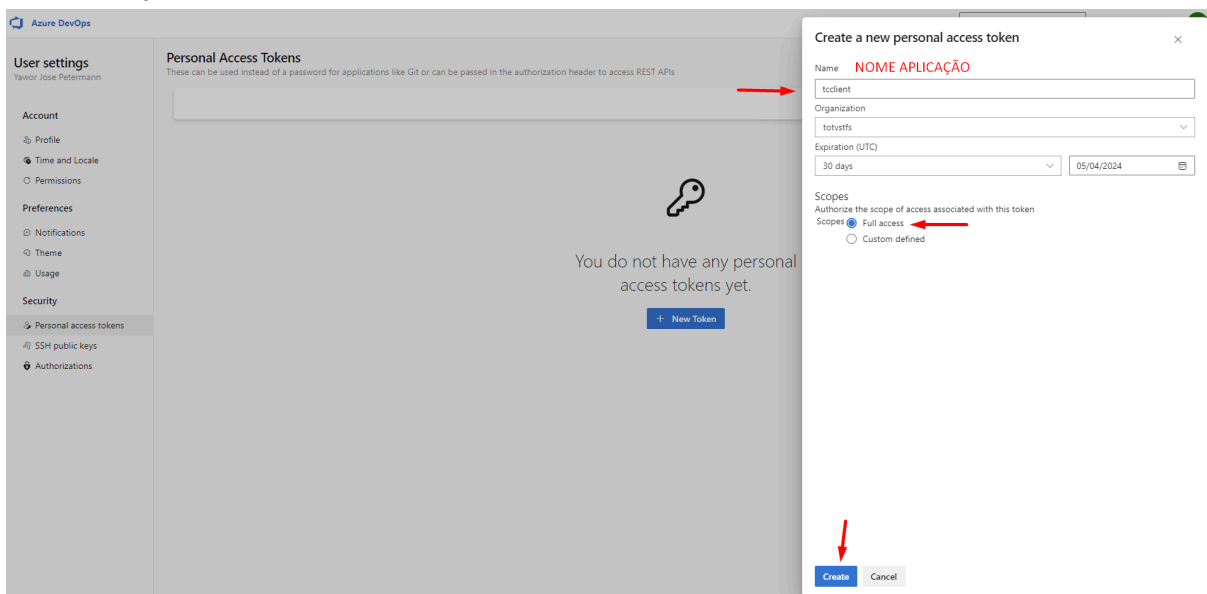
- Acessar o link <https://dev.azure.com/totvstfs/> > Projeto Supply Agro > User Settings > Personal access tokens.



- Para gerar um token, clique no botão +New Token, indicado na figura abaixo.



- Configurar conforme figura abaixo, preenchendo o campo Name com o nome da aplicação e no campo Scope marcar a opção Full Access. Após isso clique em Create para finalizar a criação do Personal Access Token.





- Após a criação, copie e anote a chave token conforme a figura abaixo, pois a mesma será reutilizada nos passos a frente.

Personal Access Tokens
These can be used instead of a password for applications like Git or can be passed in the authorization header to access REST APIs

Token name	Organization
<input type="checkbox"/> tcclient Full access	totvstfs

Success!
You have successfully added a new personal access token. Copy the token now!
tcclient token
7vzozcehbkgja6vovwj54

Warning - Make sure you copy the above token now. We don't store it and you will not be able to see it again.

4. Configuração Token no Docker (Linux)

- Rodar o comando abaixo para criação do azure agent deployment no ambiente docker (Linux).

EXEMPLO COMANDO AGENTE DEPLOYMENT:

```
sudo docker run --name "azure-agent" -d --restart always -e AZP_URL=[URL DO AZURE DEVOPS] -e AZP_DEPLOYMENT=true -e AZP_PROJECT_NAME=[PROJETO DO AZURE DEVOPS] -e AZP_DEPLOYMENTTT_GROUP_NAME=[NOME DO DEPLOYMENT GROUP] -e AZP_AGENT_NAME=[NOME DO AGENTE] -e AZP_TOKEN=[PERSONAL ACESS TOKEN] -v /var/run/docker.sock:/var/run/docker.sock -it docker.totvs.io/azure-devops-agents/agent-dind:latest
```

COMO VAI FICAR DE ACORDO COM SUAS CONFIGURAÇÕES:

```
sudo docker run --name "azure-agent" -d --restart always -e AZP_URL=https://totvstfs.visualstudio.com/ -e AZP_DEPLOYMENT=true -e AZP_PROJECT_NAME=SupplyAgro -e AZP_DEPLOYMENTTT_GROUP_NAME="TOTVS_TREINAMENTO" -e AZP_AGENT_NAME=MG-DEV -e AZP_TOKEN=cqe6cfdzpujvtmnrncsz24ufwviqkm23jxad73qgayzzv55vcicq -v /var/run/docker.sock:/var/run/docker.sock -it docker.totvs.io/azure-devops-agents/agent-dind:latest
```



Conferir no Azure DevOps > Pipelines > Deployments Groups, se o Nome do grupo que você criou (Ex:TOTVS TREINAMENTO) e o agente (Ex: MG_DEV) estão configurados, se sim vincular uma TAG para o agent pois o mesmo será o meio de ligação com a release da pipeline para que o deploy seja feito automaticamente.

Deployment groups: **TOTVS_TREINAMENTO** ← NOME DO GRUPO

Deployment group	Summary	Tags
ars-0lv	Release-201 / ADRS-PRD succeeded at 06/03/2024 11:04:10	ADRS
MG-DEV	Release-202 / MG-PRD succeeded at 06/03/2024 11:25:11	MG
rlr-prd	Release-204 / RLR-PRD succeeded at 06/03/2024 14:35:56	RLR
TCC-TFF	Release-200 / TFF-PRD succeeded at 06/03/2024 10:10:16	TFF

NOME DO AGENT

OBS: Após a finalização do processo de criação do agent para deployment, deverá ser revogado a chave de acesso (Token) criada no Terceiro Passo para meios de segurança.

Organization: totvstfs Status: Active

Manage tokens

Token name	Organization	Status	Expires on
tccient	totvstfs	Active	05/04/2024

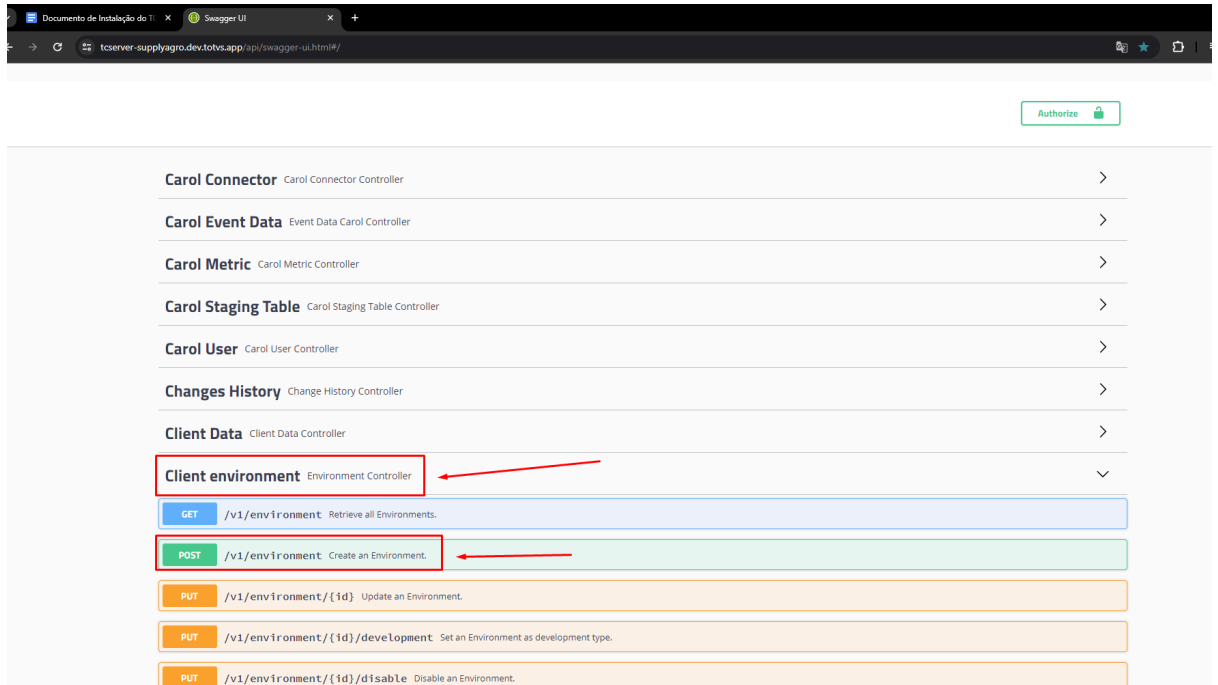
Revoke



5. Configuração Library no Azure

- Acessar o link:

<https://tserver-supplyagro.totvs.app/api/swagger-ui.html#/Client%20environment/saveUsingPOST> > Client Environment > POST - /v1/environment > Try it Out



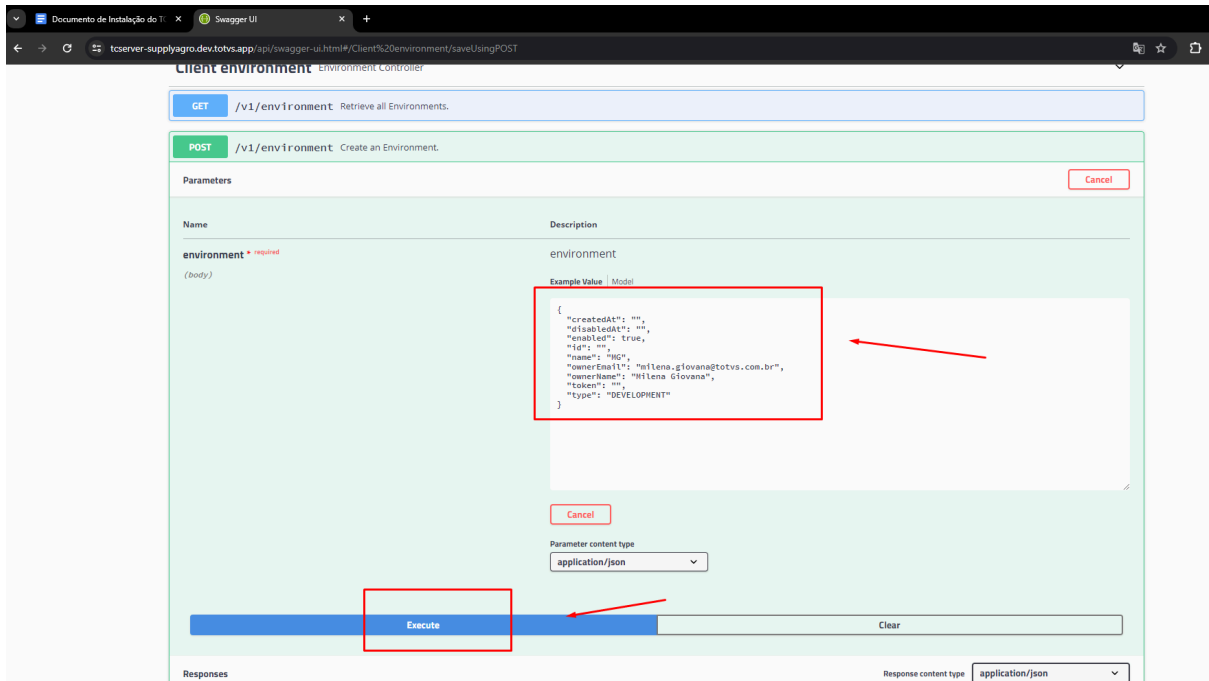
Configurar o JSON de acordo com os seus dados.

Exemplo:

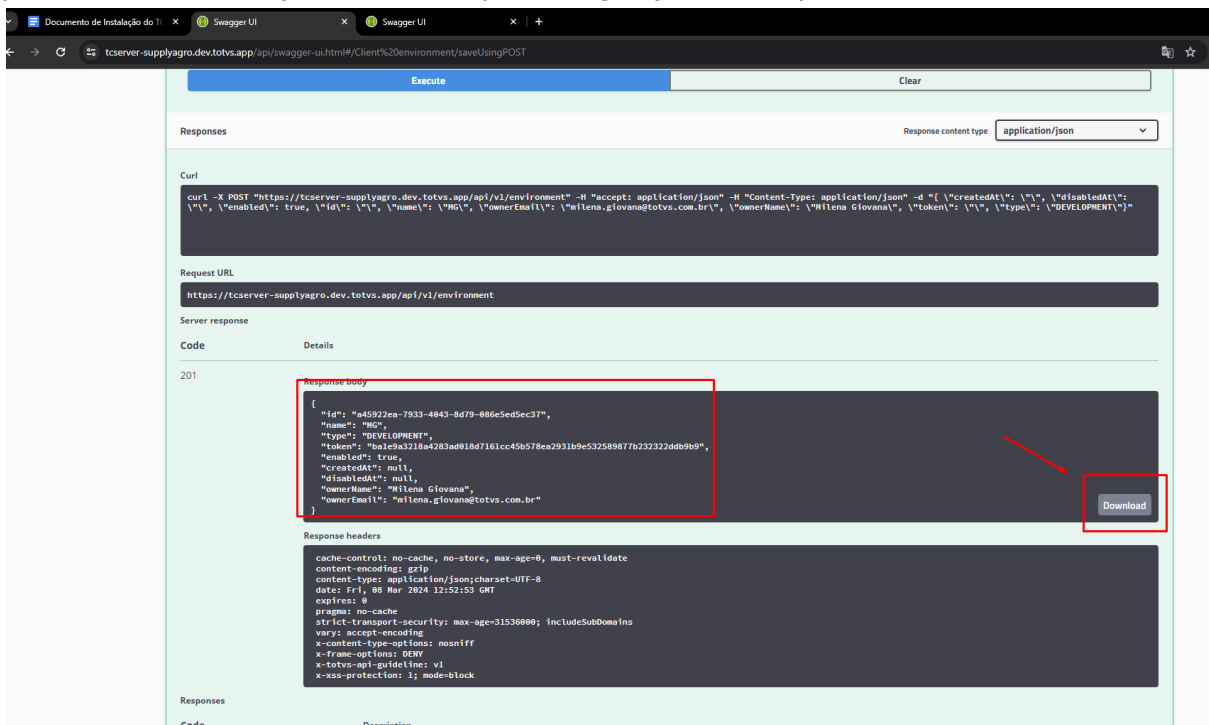
```
{
  "createdAt": "",
  "disabledAt": "",
  "enabled": true,
  "id": "",
  "name": "MG",
  "ownerEmail": "milena.giovana@totvs.com.br",
  "ownerName": "Milena Giovana",
  "token": "",
  "type": "DEVELOPMENT"
}
```



OBS: Como a instalação está sendo feita em ambiente de desenvolvimento, a tag type fica como DEVELOPMENT, no ambiente do cliente deve ser usado PRODUCTION.



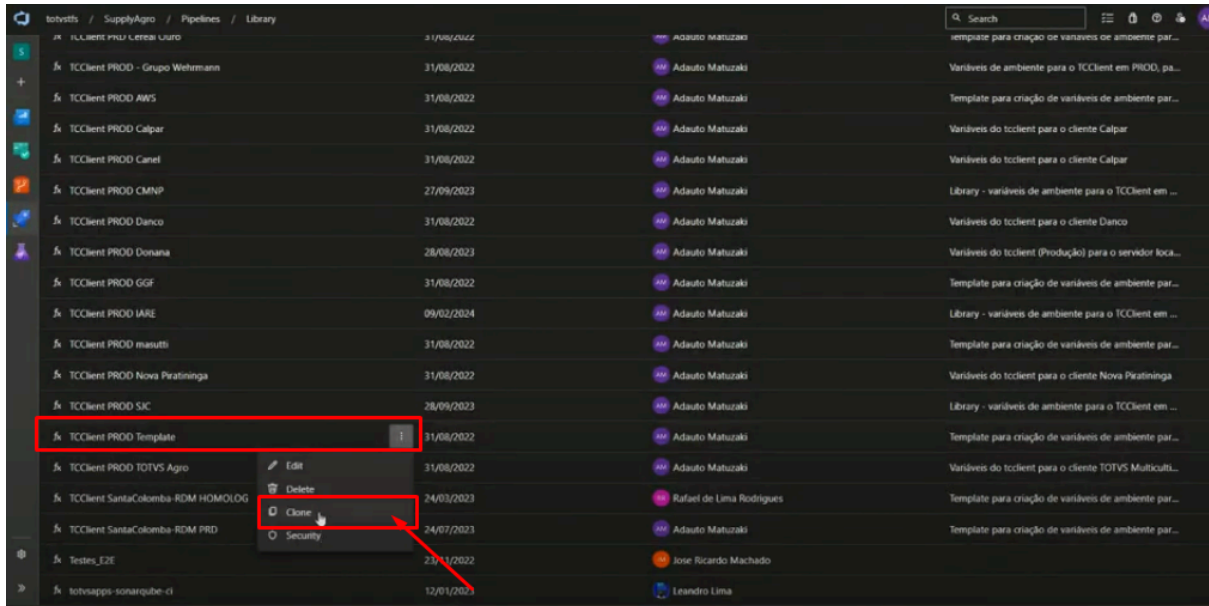
Clique em execute e no response body será retornado um JSON, que deverá ser feito o download pois conterá as informações necessárias para configuração da library no azure.





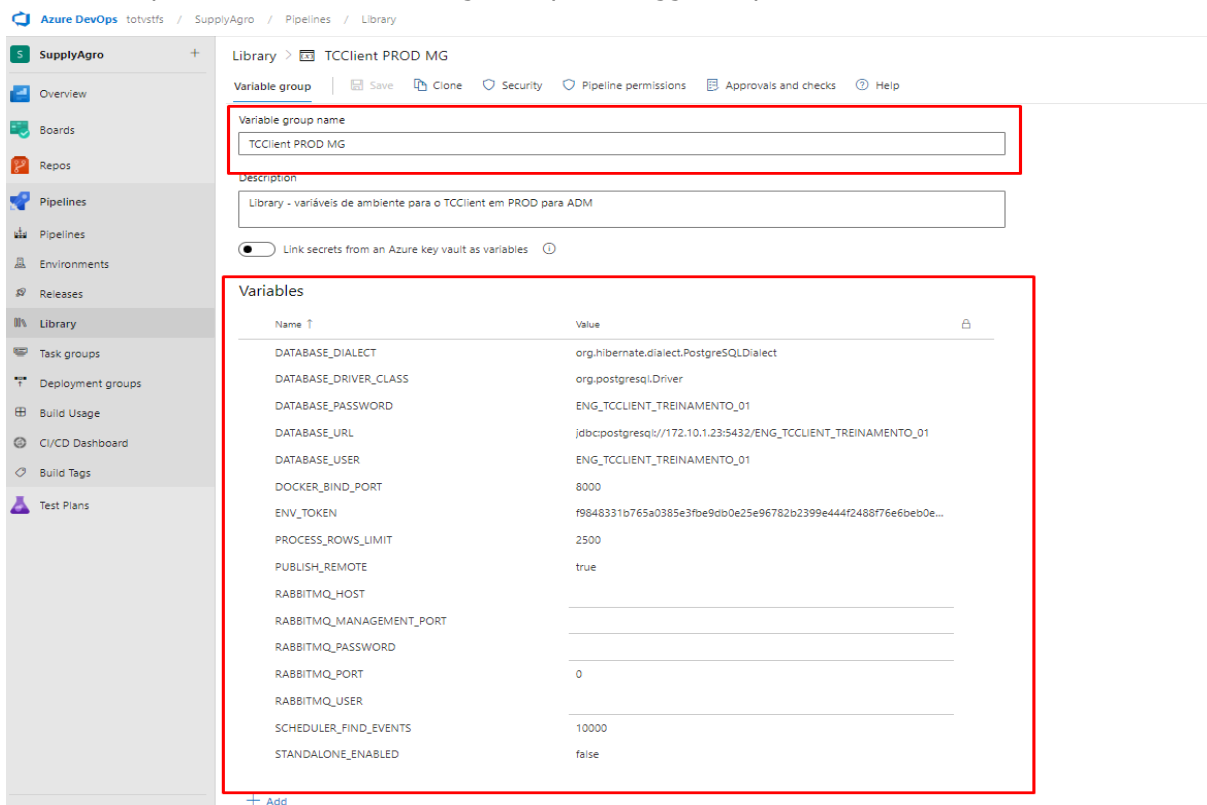
- No Azure > Projeto Supply Agro > Pipelines > Library

Para criação de uma nova library, há a opção de se criar uma nova, ou clonar a library TCClient PROD Template.



Ao clonar, realizar a configuração e renomear a variable group name e as variables de acordo com os dados do ambiente do cliente.

OBS: O campo ENV_TOKEN é o token gerado pelo swagger no passo acima.





6. Criação Pipeline Release no Azure

- Ainda no Azure > Projeto Supply Agro > Pipelines > Release, Procurar tcclient-api-CD-development, clicar em Edit.

The screenshot shows the Azure DevOps interface for the 'SupplyAgro' project. The left sidebar shows the 'Releases' section selected. The main area displays the 'tcclient-api-CD-development' pipeline with a list of releases. The 'Release-201' entry is highlighted, and the 'Edit' button is circled in red.

Clonar algum dos stage já criados para criar o seu.

The screenshot shows the Azure DevOps interface for the 'SupplyAgro' project, specifically the 'Stages' section for the 'tcclient-api-CD-development' pipeline. The 'Stages' tab is selected, and a list of stages is displayed. The 'ADRS-PRD' stage is highlighted, and the 'Clone' button is circled in red.



Ao clonar, clicar no ícone de raio e boneco.

Selecionar a trigger "After Release".



Mudar para o nome do seu stage.

The screenshot shows the Azure DevOps interface for a pipeline named 'tcclient-api-CD-development'. The pipeline is displayed as a sequence of tasks: 'Deploy - donana', 'Deploy - MATIV', 'Deploy - Benefica...', 'ADM-PRD', 'TFF-PRD', 'ADRS-PRD', and 'MG-PRD'. The 'MG-PRD' task is highlighted with a red box. A red arrow points from the 'MG-PRD' task to the 'Properties' panel on the right, where the 'Stage name' is set to 'MG-PRD' and is also highlighted with a red box.

Clicar em Save na aba superior direita.

The screenshot shows the same Azure DevOps interface as the previous one, but with a red box highlighting the 'Save' button in the top right corner of the interface. A red arrow points to the 'Save' button.



Vinculação da Tag do agent deployment ao stage criado na pipeline de release.

- Acessar a aba de pipeline de release > Tasks e selecionar o seu stage que foi criado:

The screenshot shows the Azure DevOps interface for a pipeline named 'tcclient-api-CD-development'. The 'Tasks' tab is selected, and a dropdown menu is open, showing a list of stages: 'Deploy - donana', 'Deploy - MATIV', 'Deploy - Beneficiamento...', 'ADM-PRD', 'TFF-PRD', 'ADRS-PRD', 'MG-PRD', and 'RLR-PRD'. The 'MG-PRD' stage is highlighted with a red box and a red arrow. The main view shows a sequence of tasks in the pipeline, each corresponding to a stage: 'Deploy - donana', 'Deploy - MATIV', 'Deploy - Beneficiamento...', 'ADM-PRD', 'TFF-PRD', 'ADRS-PRD', and 'MG-PRD'.

Clicar em deployment groups job e vincular sua tag no campo required tags.

The screenshot shows the configuration for a 'Deployment group job'. The 'Required tags' field is highlighted with a red box and a red arrow, showing the tag 'MG' entered. The configuration includes fields for 'Display name', 'Deployment targets', 'Deployment group', 'Required tags', 'Targets to deploy to in parallel', 'Timeout', 'Job cancel timeout', 'Artifact download', and 'Additional options'.



OBS: Se atentar quando clonar algum Stage já existente, se a Variables Groups onde está localizada sua tag não foi aplicada também ao stage que foi usado para clone.

Message: User does not have permission for linking the variable group 'TCClient PROD ADM (ID: 1096)' to the stage 'MG-PRD (1)'. Make sure user has 'User Role' on the variable group and retry the operation.

Name	Value
TCClient DEV Donana (18)	Scopes: Deploy - donana
TCClient DEV MATIV (19)	Scopes: Deploy - MATIV
TCClient HMG Beneficiamento (14)	Scopes: Deploy - Beneficiamento-HMG
TCClient PROD ADM (16)	Scopes: ADM-PRD-MG-PRD (1)
TCClient PROD TFF (16)	Scopes: TFF-PRD
TCClient PROD MG (16)	Scopes: MG-PRD
TCClient PROD RLR (16)	Scopes: RLR-PRD

Caso tenha sido aplicada, clique nos três pontos laterais do group variable e depois em change scope.

Message: User does not have permission for linking the variable group 'TCClient PROD ADM (ID: 1096)' to the stage 'MG-PRD (1)'. Make sure user has 'User Role' on the variable group and retry the operation.

Name	Value
TCClient DEV Donana (18)	Scopes: Deploy - donana
TCClient DEV MATIV (19)	Scopes: Deploy - MATIV
TCClient HMG Beneficiamento (14)	Scopes: Deploy - Beneficiamento-HMG
TCClient PROD ADM (16)	Scopes: ADM-PRD-MG-PRD (1)
TCClient PROD TFF (16)	Scopes: TFF-PRD
TCClient PROD MG (16)	Scopes: MG-PRD
TCClient PROD RLR (16)	Scopes: RLR-PRD

Change scope

Desvincular o seu stage com a variable groups criada.

Message: User does not have permission for linking the variable group 'TCClient PROD ADM (ID: 1096)' to the stage 'MG-PRD (1)'. Make sure user has 'User Role' on the variable group and retry the operation.

Name	Value
TCClient DEV Donana (18)	Scopes: Deploy - donana
TCClient DEV MATIV (19)	Scopes: Deploy - MATIV
TCClient HMG Beneficiamento (14)	Scopes: Deploy - Beneficiamento-HMG
TCClient PROD ADM (16)	Scopes: ADM-PRD-MG-PRD (1)
TCClient PROD TFF (16)	Scopes: TFF-PRD
TCClient PROD MG (16)	Scopes: MG-PRD
TCClient PROD RLR (16)	Scopes: RLR-PRD

Variable group scope

- ADM-PRD (1)
- MG-PRD
- RLR-PRD
- TFF-PRD
- ADM-PRD
- MG-PRD
- RLR-PRD
- MG-PRD (1) **DESVINCLAR**



7. Configuração Swagger ambiente linux

- No ambiente linux, acessar o link
http://127.0.0.1:8000/swagger-ui.html#/Schema%20Definitions/getAllUsingGET_7

Ir em Schema Definitions > GET > Clicar em Try it Out e Executar

Schema Definitions Schema Definition Controller

GET /api/v1/schema Retrieve all the schema definition

Parameters

Name	Description
fields	array[string] (query)
name	string (query)
order	array[string] (query)
page	integer(\$int32) (query)
pageSize	integer(\$int32) (query)
version	string (query)

GET /api/v1/schema Retrieve all the schema definition

Parameters

Name	Description
fields	array[string] (query) <input type="button" value="Add item"/>
name	string (query) <input type="text" value="name"/>
order	array[string] (query) <input type="button" value="Add item"/>
page	integer(\$int32) (query) <input type="text" value="page"/>
pageSize	integer(\$int32) (query) <input type="text" value="pageSize"/>
version	string (query) <input type="text" value="version"/>

Será necessário informar o login e senha de acesso na primeira vez.

Login: admin

Senha: 9ZnMp5PPh>;5v,Xr



Realizar o download do response body do get que foi executado.



- No ambiente linux, acessar o link
http://127.0.0.1:8000/swagger-ui.html#/Schema%20Definitions/getAllUsingGET_7

Ir em Product Connections > POST - /api/v1/productConnections > Clicar em Try it Out e modificar os dados de acordo com o banco de dados do cliente.

Exemplo:

```
{
  "dataBaseType": "ORACLE12C", (BASE DE DADOS DO CLIENTE)
  "enabled": true,
  "id": "",
  "password": "ENG_MC_TREINAMENTO_01", (SENHA DO BANCO DO CLIENTE)
  "productName": "PIMSMC", (NOME DO PRODUTO NA BASE DE DADOS DO CLIENTE)
  "productVersion": "12.1.2305", (VERSÃO DO PRODUTO NA BASE DE DADOS DO CLIENTE)
  "url": "jdbc:oracle:thin:@172.10.1.21:1521:DOURADO", (CAMINHO DE CONEXÃO COM O BANCO DE DADOS DO CLIENTE)
  "username": "ENG_MC_TREINAMENTO_01" (LOGIN NA BASE DE DADOS DO CLIENTE)
}
```

OBS: Para testar se os dados estão corretos do banco de dados, ainda no swagger, ir em Product Connections > POST - /api/v1/productConnections/connectionTest > Clicar em Try it Out, colar o json dos dados do banco do cliente e clicar em executar para testar se a conexão está correta, se o retorno no response body for 200, está ok e pode ser executado no POST acima.



POST /api/v1/productConnections/connectionTest Test a Product Connection

Parameters

Name Description

dto ^{required} (body)

Example Value | Model

```
{
  "databaseType": "ORACLE12C",
  "enabled": true,
  "url": "jdbc:oracle:thin:@127.0.0.1:1521:ORURADO",
  "password": "ENG_MC_TREINAMENTO_01",
  "productname": "PDMSC",
  "productversion": "12.1.2.305",
  "username": "ENG_MC_TREINAMENTO_01"
}
```

Parameter content type: application/json

Execute Clear

Responses

Response content type: */*

Curl

```
curl -X POST http://127.0.0.1:8000/api/v1/productConnections/connectionTest -H 'accept: */*' -H 'Content-Type: application/json' -d '{ "databaseType": "ORACLE12C", "enabled": true, "url": "jdbc:oracle:thin:@127.0.0.1:1521:ORURADO", "password": "ENG_MC_TREINAMENTO_01", "productname": "PDMSC", "productversion": "12.1.2.305", "username": "ENG_MC_TREINAMENTO_01" }'
```

Request URL

```
http://127.0.0.1:8000/api/v1/productConnections/connectionTest
```

Server response

Code: 200

Response body

```
{
  "code": "ProductConnection.success.connection",
  "message": "Conexão com o banco de dados realizada com sucesso!"
}
```

OBS-1: Ao realizar o Execute do POST - /api/v1/productConnections pegar e anotar o número do id que será retornado no response body pois o mesmo será necessário para configurar o schema.

POST /api/v1/productConnections

Parameters

Parameter content type: application/json

Execute Clear

Responses

Response content type: */*

Curl

```
curl -X POST http://127.0.0.1:8000/api/v1/productConnections -H 'accept: */*' -H 'Content-Type: application/json' -d '{ "databaseType": "ORACLE12C", "enabled": true, "url": "jdbc:oracle:thin:@127.0.0.1:1521:ORURADO", "password": "ENG_MC_TREINAMENTO_01", "productname": "PDMSC", "productversion": "12.1.2.305", "username": "ENG_MC_TREINAMENTO_01" }'
```

Request URL

```
http://127.0.0.1:8000/api/v1/productConnections
```

Server response

Code: 201

Response body

```
{
  "id": "0398088-6686-4286-9379-36639ff6a84e",
  "message": "sucesso!"
}
```

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-encoding: gzip
content-type: application/json;charset=UTF-8
date: Thu, 07 Mar 2024 15:39:16 GMT
etag: W/"1212305"
expires: 0
pragma: no-cache
server-timing: timeout=60
transfer-encoding: chunked
vary: accept-encoding
x-content-type-options: nosniff
x-frame-options: DENY
```



- Ainda na página do swagger, ir em Schema Definitions > GET > Try it Out, informar tamanho 2 no campo Pages e 100 no campo Pages Size e clicar em Executar.

The screenshot shows the Swagger UI interface. The 'Schema Definitions' section is expanded, showing the 'GET /api/v1/schema' endpoint. The 'Parameters' section is visible, with the 'page' field set to '2' and 'pageSize' set to '100'. The 'Execute' button is highlighted with a red box and an arrow pointing to it.

Ao clicar em Executar será carregado uma lista de esquemas no response body, realize o download deste arquivo.

The screenshot shows the Swagger UI interface displaying the response body of the 'GET /api/v1/schema' endpoint. The response body is a JSON array of schema definitions. The 'Download' button is highlighted with a red box and an arrow pointing to it.



Escolher dentro deste arquivo baixado, um schema que corresponda ao produto usado na sua tabela do banco de dados, no nosso caso foi usado o AnoAgricola que pertence ao produto PIMSMC.

```
{
  "id": "df3fce67-6653-42df-8bc6-682b60fcf997",
  "version": "12.1.27",
  "name": "AnoAgricola",
  "allTreeAttributes": [],
  "allRequiredAttributes": []
}
```

OBS-2: Os schemas são gerados automaticamente de acordo com arquivo abaixo, consultar a mesma para ver se o schema escolhido pertence ao produto do seu banco.

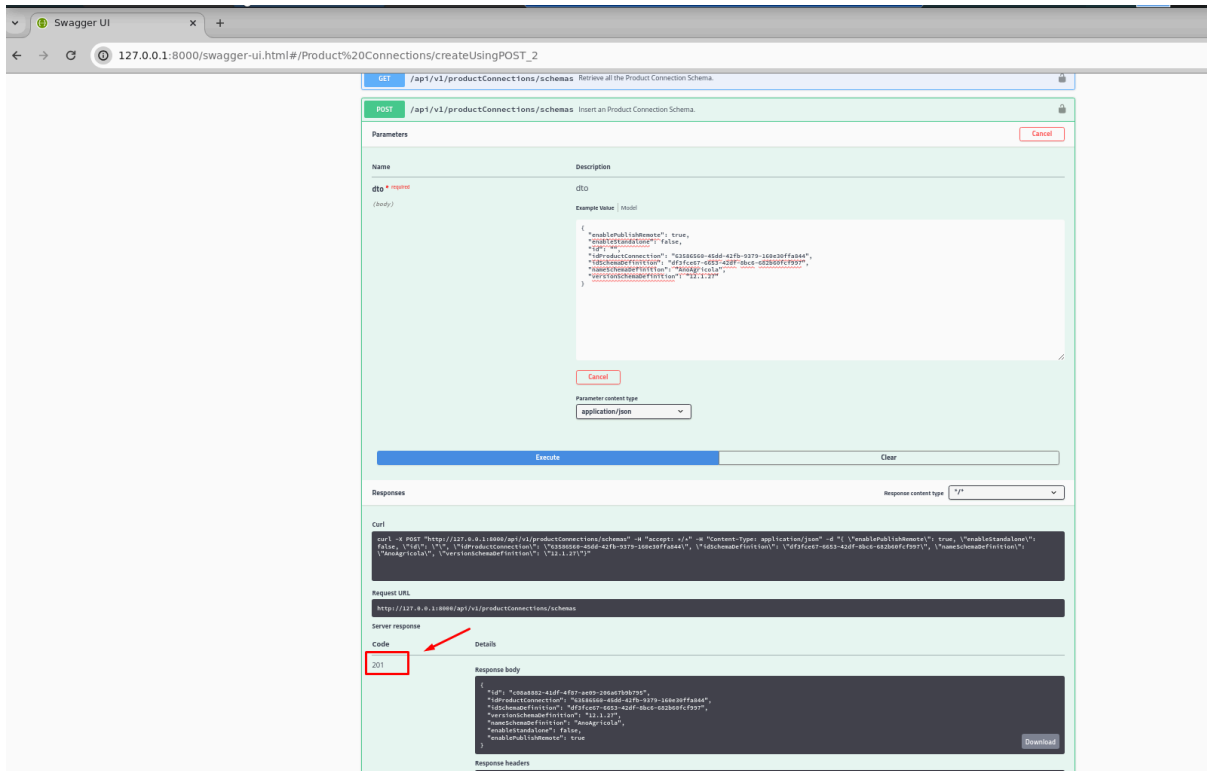
<https://docs.google.com/spreadsheets/d/1y5vVNuBzkdIHczfZyjtPEPKrb4QLYJsbTMSAJc5AbI/edit#gid=647426749>

- Ainda na página do swagger, ir em Product Connections > POST > /api/v1/productConnections/schemas > Try it Out, informar os dados do schema escolhido como mostrado na **OBS-2** e utilizar o id que foi retornado pela **OBS-1**. de acordo com o exemplo abaixo e clicar em Executar:

```
{
  "enablePublishRemote": true,
  "enableStandalone": false,
  "id": "",
  "idProductConnection": "63586560-45dd-42fb-9379-160e30ffa844",
  "idSchemaDefinition": "df3fce67-6653-42df-8bc6-682b60fcf997",
  "nameSchemaDefinition": "AnoAgricola",
  "versionSchemaDefinition": "12.1.27"
}
```

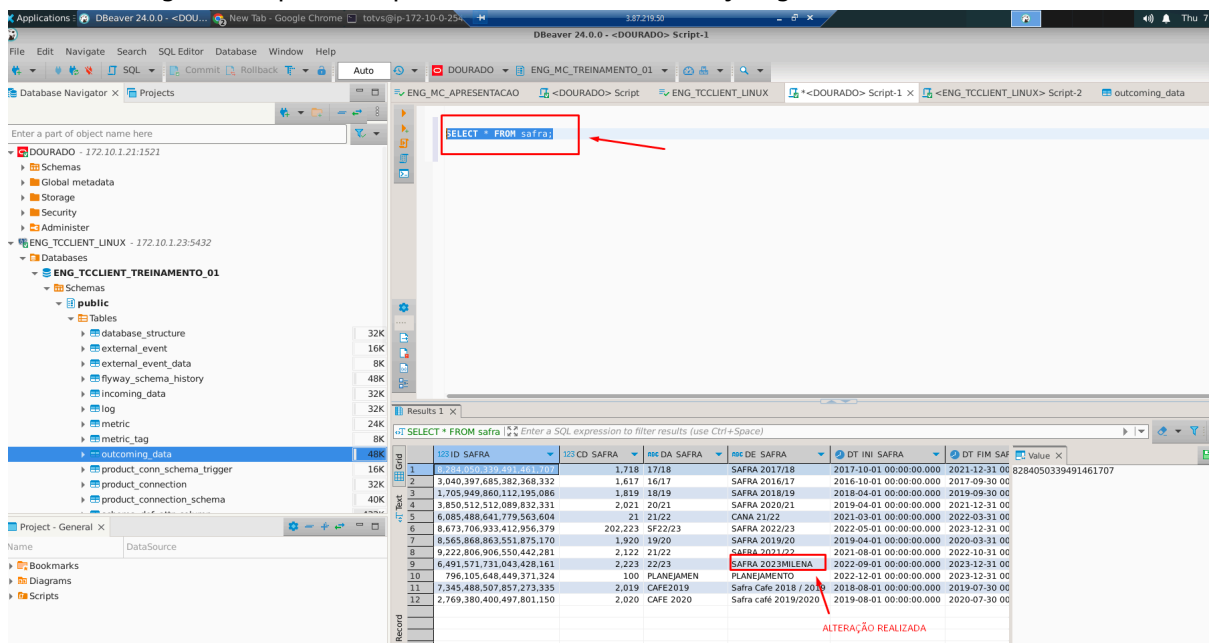


Após clicar em executar, será retornado um 201 no response body, sinalizando que o processo ocorreu ok e que o schema de AnoAgricola (Tabela SAFRAS) começou a ser monitorado pelo tcclient.



Sendo assim, o processo de configuração do totvs agro connector (tcclient) foi finalizado.

- Para simulação de um evento na conexão com o banco, acesse o ambiente do cliente, sua base de dados que no exemplo abaixo será o DBEave.
- Acesse a tabela do schema escolhido, que no exemplo utilizado na configuração foi o AnoAgricola e que corresponde a tabela SAFRAS e faça algum UPDATE nos dados.





- Acesse a tabela de eventos para verificar se sua alteração foi notificada como novo evento.

TABLE NAME	IDENTIFIER	ACTION	CREATED AT	PROCESSED AT	STATUS
TRA	ID_SAFRA=6491571731043428161	UPDATE	2024-03-07 19:55:18.000	2024-03-07 19:55:27.672	PROCESS
TRA	ID_SAFRA=6491571731043428161	UPDATE	2024-03-07 19:55:18.000	2024-03-07 19:55:27.773	PROCESS
TRA	ID_SAFRA=6491571731043428161	UPDATE	2024-03-07 19:55:18.000	2024-03-07 19:55:27.672	PROCESS
TRA	ID_SAFRA=6491571731043428161	UPDATE	2024-03-06 18:02:08.000	2024-03-06 18:02:13.153	PROCESS
TRA	ID_SAFRA=6491571731043428161	UPDATE	2024-03-06 18:02:08.000	2024-03-06 18:02:13.153	PROCESS

- Acesse a base do tclient para verificar quais foram as alterações realizadas no UPDATE da tabela SAFRAS.

Base TClient

ENG_TCCLIENT_TREINAMENTO_01

outcoming_data

SAFRA

Se as alterações foram evidenciadas, o evento foi lido e registrado e o processo foi concluído com sucesso.



LINKS Gravações de Configuração Adatao: WINDOWS-LINUX

- 1 - <https://drive.google.com/file/d/1xCHH-Dmd8QMbyrmNi1K8DUqk-d34A7G1/view>
- 2 - <https://drive.google.com/file/d/1YcCgPWFwca3vUfKTuJfyvFPyQyoZw1ER/view>
- 3 - <https://drive.google.com/file/d/1nnOh4HluoOmwr8Af2SRIDx9yuiPgR6oU/view>