



Generalitat de Catalunya
**Centre de Telecomunicacions
i Tecnologies de la Informació**

Guia de desplegament d' APIs

Octubre 2024

Índex de Continguts

1	Introducció.....	3
2	Passos previs al desplegament	3
2.1	Permisos	3
2.2	Instal·lació de programari	6
2.2.1	Git	6
2.2.2	Client VPN	6
2.2.3	API Connect Toolkit	7
2.2.4	Docker.....	7
2.3	Configuració de l' entorn local.....	8
2.3.1	VPN	8
2.3.2	API Connect Toolkit i Local Testing Environments (LTE)	10
2.3.3	Gitlab	11
3	Desplegament de APIs	15
3.1	Consideracions prèvies dins del CTTI	15
3.2	Creació de fitxers necessaris	16
3.2.1	ACA (Arxiu generat pel desenvolupador)	16
3.2.2	ACS (Arxiu generat per SIC)	20
3.2.3	ACD (Arxiu generat per SIC).....	21
3.2.4	ACI (Arxiu generat per SIC)	21
3.3	Pujada de carpetes i fitxers al repositori.....	22
3.4	Desplegament al Pipeline.....	25
3.4.1	Check de validació per als plans i seguretat.....	25
3.4.2	Passos per executar el pipeline	26
3.4.3	Consultar logs	34
4	Enllaços d' interès	36

1 Introducció

El present document ha estat elaborat amb la finalitat d' oferir als proveïdors que usen el servei d' API Manager del CTTI d' un conjunt d' instruccions detallades que ajudaran els proveïdors a procedir amb el desplegament d' APIs dels seus projectes.

2 Passos previs al desplegament

Abans de començar amb el desplegament d' APIs, es llisten una sèrie d' accions que s' han de demanar i realitzar per tal de poder dur a terme un desplegament sense dificultats.

2.1 Permisos

Per poder usar les eines i serveis necessaris per al desplegament d' APIs es requereix demanar una sèrie de permisos, que es detallen a continuació:

- **GICAR:** És l'Identity provider de CTTI. Cal tenir donat d' alta un usuari a GICAR ja que les eines a usar estan integrades amb aquest IDP.

La gestió de l'alta, baixa, modificació dels usuaris/passwords a GICAR es duu a terme a través del departament de Gestió d'identitats (PMO) dins de CTTI. Cada proveïdor tindrà una persona assignada del PMO, que s'encarregarà de gestionar i redirigir totes les sol·licituds o tràmits a realitzar relacionats amb GICAR.

Els mecanismes de sol·licitud d'alta a GICAR per a cadascun dels proveïdors es gestionaran de forma independent entre el proveïdor i la persona del PMO assignada a ell. Havent-hi d'aquesta forma mecanismes de sol·licitud diferents per a cada proveïdor.

Per exemple, per a Accenture com a proveïdor, el mecanisme de sol·licitud a seguir seria el següent:

- *L'alta d'usuari a GICAR es demana manant un correu a PMO.HPSpain@accenture.com , adjuntant **el nom i els cognoms, el DNI i el número mòbil**.
Un cop demanat el permís, se li manarà un SMS detallant els següents passos a seguir i la necessitat de canviar la contrasenya.*

**** És recomanat especificar en el correu que es necessita tant credencials GICAR PRE com per a PRO**, per poder accedir tant a l' entorn de Reproducció com el de Producció de GICAR. Com a nota, comentar que per lloar-se als portals d'API Manager, només es fan servir les credencials de PRO de GICAR, tot i que les APIs en PRE sí que poden tenir configurats proveïdors OAuth que usin usuaris de GICAR PRE.**

- **VPN:** Necessària per accedir als recursos interns de CTTI, com ara el Gitlab, Jenkins..., per poder realitzar els desenvolupaments per cadascun dels proveïdors.

Cada proveïdor tindrà un conjunt de VPNs assignades que difereixen de la resta de proveïdors.

La gestió de l'alta, baixa, modificació dels accessos a la VPN es duu a terme igual que l'accés a GICAR, a través de la persona de l'equip de Gestió d'identitats (PMO) dins de CTTI assignada a cada proveïdor, la qual s'encarregarà de gestionar i redirigir totes les sol·licituds o tràmits a realitzar relacionats amb l'accés a les VPNs corresponents.

- **JIRA:** Plataforma que es fa servir per realitzar peticions de suport, incidències, dubtes...
 - **** S' haurà de demanar al més aviat possible per poder demanar els permisos restants. ****
 - L'alta a JIRA es demana manant el portal **PAUTIC** a la secció de **Gestió accés d'usuaris**, sol·licitant accés a peticionar a **ACOAPIM** per poder obrir tiquets d'acompanyament de tot allò relacionat amb API Manager i **ACOCLDSIC** per poder sol·licitar al SIC la configuració de les pipelins:

- **Gestió accés d'usuaris**



El GDI podrà sol·licitar l'alta, la baixa i la modificació d'usuaris en l'aplicació. En el cas que se sol·liciti una petició relacionada amb un projecte JIRA serà necessari adjuntar un correu o document amb la validació del responsable del projecte. Qualsevol dubte que es tingui, es recomana contactar amb l'equip responsable via portal PAUTIC o via correu cstd.ctti@gencat.cat.

- **IBM API Connect:** Plataforma per on estan desplegades les APIs i els productes relacionats amb el CTTI.
 - **Requereix accés a JIRA****
 - Per sol·licitar el permís a l'IBM API Connect, s'obre un tiquet de JIRA a la següent url: <https://cstd-ctti.atlassian.net/jira/software/c/projects/ACOAPIM/issues> S'introdueixen les credencials de GICAR i creem una petició nova sol·licitant l'accés corresponent:



Crear

Los campos obligatorios están marcados con un asterisco *

Proyecto *

Servei d'acompanyament APIM (ACOAPIM)

Tipo de incidencia *

Acompañamiento

[Más información sobre los tipos de incidencias](#)

Estado

Nou

Este es el estado inicial en el momento de la creación

Organisme / Projecte Afectat

Ninguno

(Migrated on 12 Jul 2024 13:59 UTC)

Crear otra

Cancelar **Crear**

A continuació, s'haurien d'emplenar com a mínim els següents camps recomanats:

- **Organisme/Projecte Afectat:** Indicar el nom del projecte/organisme.
 - **Si no localitzes el projecte en el desplegable, introdueix-lo aquí:** Indicar el nom del projecte a desplegar, si no es troba en el desplegable.
 - **Resum:** Concepte de la petició, per exemple: "Sol·licitud d'accés a IBM API Connect".
 - **Descripció:** Descripció detallada de la sol·licitud.
 - **Codi de servei:** codi del servei per al qual es demana l'acompanyament.
- **Gitlab:** És l'eina que es farà servir com a repositori de tots els fitxers relacionats amb el projecte.
 - Un cop donat d'alta el projecte en l'inventari, es crearà al seu torn de forma automàtica el grup de Gitlab, assignant el rol de Release Manager a un membre de l'equip qui s'encarrega de crear les carpetes necessàries dins del repositori i de donar permisos a la resta del grup.
 - Per a consultes sobre la gestió de permisos, el funcionament de Gitlab..., accedir a la següent url: [Autoservei d'usuaris \(gencat.cat\)](https://autoservei.diusuaris.gencat.cat).

2.2 Instal·lació de programari

2.2.1 Git

Git és un sistema de control de versions distribuït que farem servir en aquest guia per establir connexió i operar a Gitlab.

Nota: Cal esmentar que existeixen altres eines de Git disponibles, com Sourcetree, Tortoise Git, Fork..., però en aquest guia es farà servir Git com a exemple.

Els passos per a la seva instal·lació són:

1. Entrar a <https://git-scm.com/> i donar-li a la secció Downloads.
2. Escollir la versió que ens correspongui segons el nostre sistema operatiu. En aquest cas, triem 64-bit Git for Windows Setup.

Downloads



Download for Windows

Click here to download the latest (2.42.0) 64-bit version of **Git for Windows**. This is the most recent **maintained build**. It was released **about 2 months ago**, on 2023-08-30.

Other Git for Windows downloads

Standalone Installer

32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

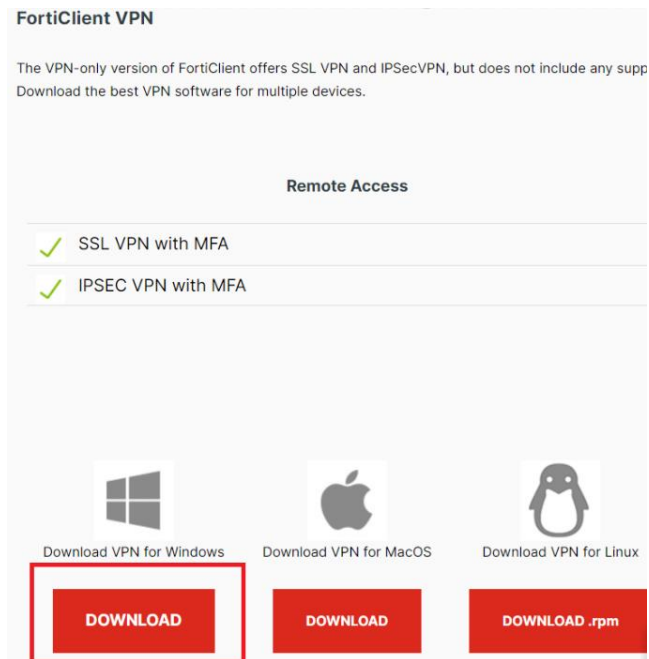
3. Procedim a instal·lar el programari amb totes les configuracions posades en **predeterminades**.

2.2.2 Client VPN

La VPN és l'eina que farem servir per connectar-nos amb la xarxa interna de CTTI. Per a això, és necessari descarregar-se un client de VPN, com, per exemple, FortiClient VPN, Anyconnect o OpenVPN.

Com a exemple, per descarregar el client de FortiClient VPN, els passos a seguir per a la seva instal·lació serien les següents:

1. Accedir a <https://www.fortinet.com/support/product-downloads> i descarregar el **FortiClient VPN**.



2. Després, s'executaria el fitxer "FortiClientVPNOnlineInstaller.exe" per instal·lar el programari. Posteriorment, s'hauria de configurar per poder establir la connexió amb els entorns de CTTI.

2.2.3 API Connect Toolkit

API Connect Toolkit és l'eina facilitada per IBM perquè els desenvolupadors puguin desenvolupar les seves pròpies APIs en el seu entorn local, que compta amb les principals funcionalitats que té l'API Manager. Tota la informació sobre la seva instal·lació i configuració es troba més endavant en el [punt 2.3.2](#).

2.2.4 Docker

Docker és una plataforma de programari que ens permet crear un entorn local de prova per testejar l'API que es desenvolupi. Els passos per a la seva instal·lació són:

1. Accedim a la pàgina oficial de Docker i li donem al botó "Docker Desktop for Windows" per procedir a descarregar-lo.



2. Procedim amb la instal·lació del programari amb totes les configuracions posades en **predeterminades**.

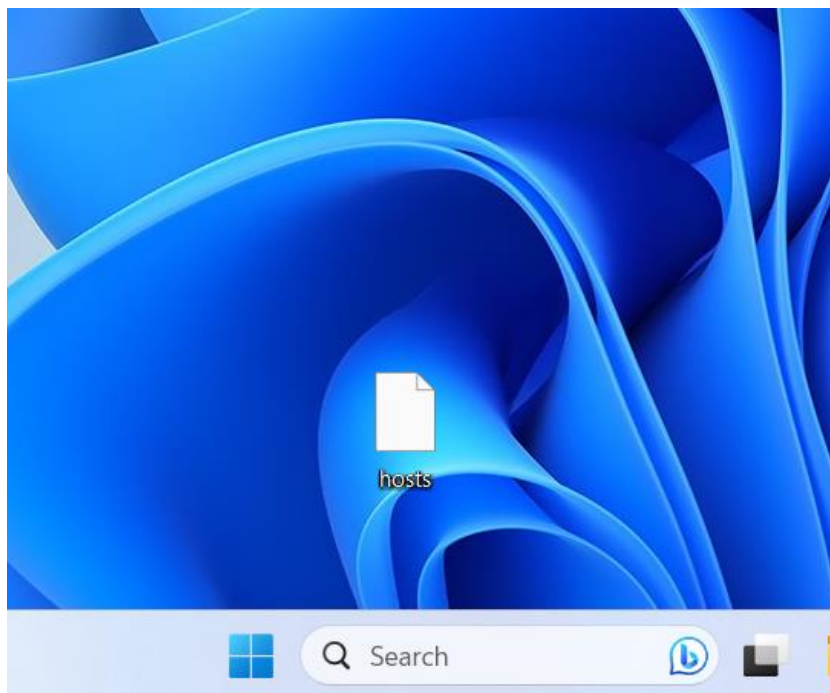
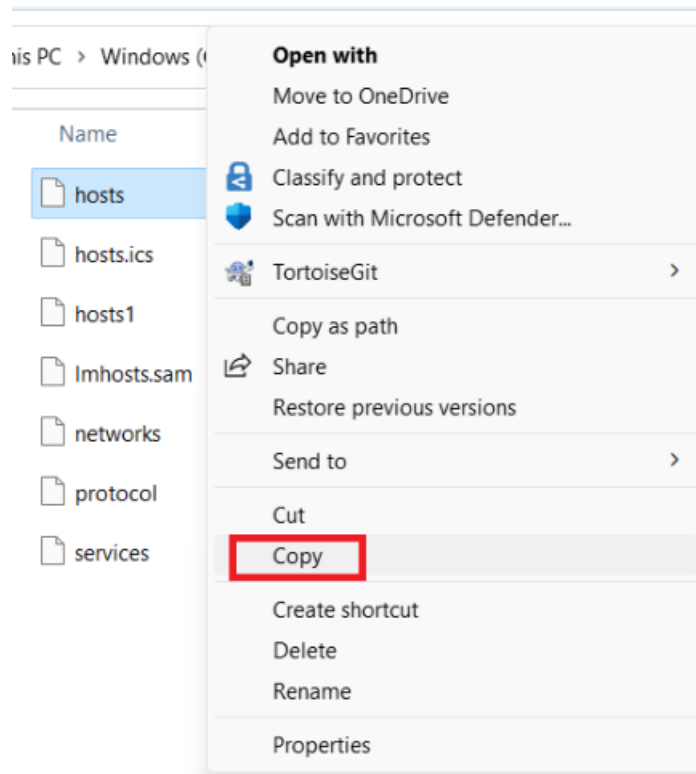
2.3 Configuració de l'entorn local

2.3.1 VPN

Tenint en compte que a cada proveïdor se li assigna per part de CTTI un conjunt de VPNs diferents a qualsevol altre proveïdor, el client VPN a usar i la seva configuració haurà de ser indicada i proveïda per la persona del PMO assignada a cada proveïdor.

Com a primer pas, és possible que s'hagi de configurar l'arxiu **hosts**, on es relacionen les adreces IP dels servidors o llocs web amb els seus noms de domini, de manera que puguem accedir als recursos interns de CTTI. Per a això, es llisten els següents passos a seguir per configurar-lo amb les IPs necessàries per al treball en CTTI:

1. Obrim l'explorador d'arxius i ens dirigim a la següent ruta:
C:\Windows\System32\drivers\etc
2. Localitzem l'arxiu **hosts** i li donem click dret per realitzar una còpia, la qual enganxem en qualsevol altra ruta (en l'escriptori preferiblement).



3. Obriem l'arxiu amb un editor de text (bloc de notes) i **hi afegim** les següents IPs dins l'arxiu després de l'última línia comentada (els comentaris dins del fitxer de host es marquen amb #)

```
# localhost name resolution is handled within DNS itself.  
#          127.0.0.1 localhost  
#          ::1 localhost
```

Les IPs a afegir són la següent:

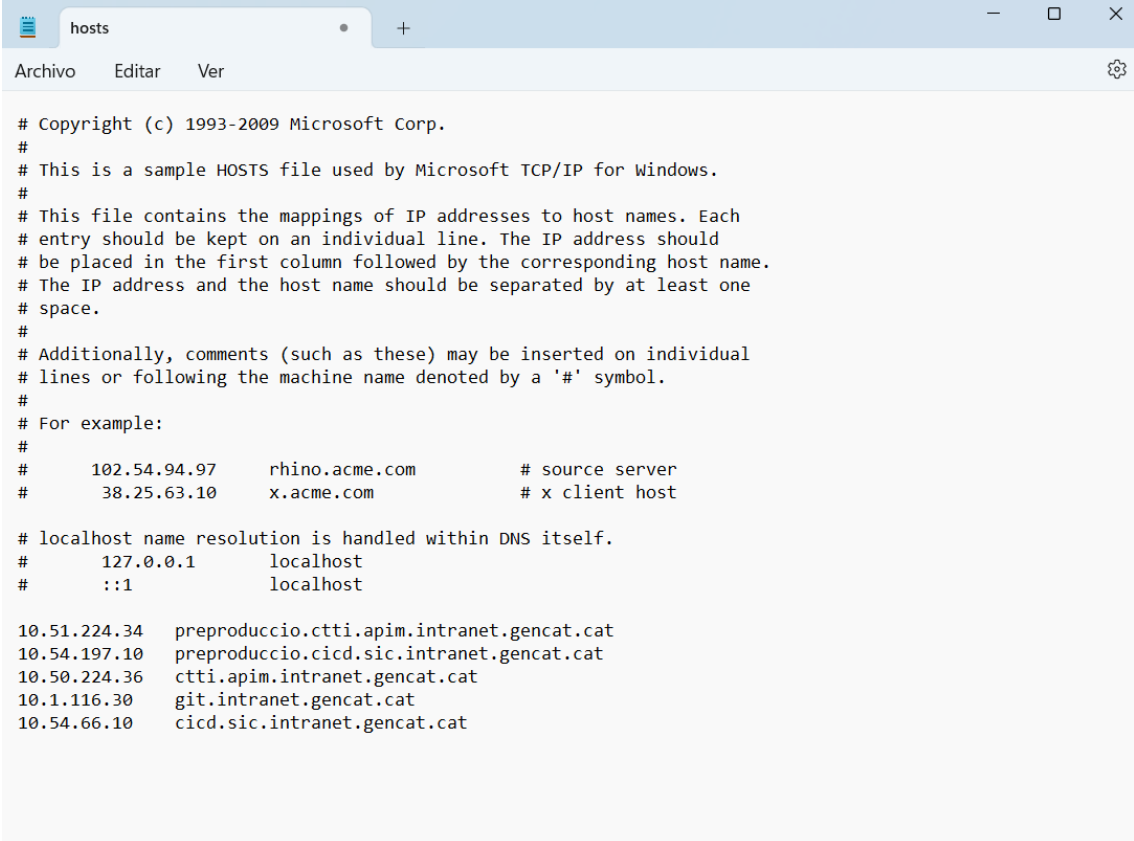
10.51.224.34 preproduccio.ctti.apim.intranet.gencat.cat (entorn PRIVAT-PRE de l'API-M)

10.54.197.10 preproduccio.cicd.sic.intranet.gencat.cat (entorn PRE SIC)

10.50.224.36 ctti.apim.intranet.gencat.cat (entorn PRIVAT de l'API-M)

10.1.116.30 git.intranet.gencat.cat (Gitlab de CTTI)

10.54.66.10 cicd.sic.intranet.gencat.cat (entorn PRO SIC)



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10      x.acme.com               # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1        localhost
#       ::1              localhost

10.51.224.34 preproduccio.ctti.apim.intranet.gencat.cat
10.54.197.10 preproduccio.cicd.sic.intranet.gencat.cat
10.50.224.36 ctti.apim.intranet.gencat.cat
10.1.116.30  git.intranet.gencat.cat
10.54.66.10  cicd.sic.intranet.gencat.cat
```

Després d'afegir aquestes rutes IP, li donem a guardar.

4. A continuació, copiarem l'arxiu **hosts** configurat a la ruta:
C:\Windows\System32\drivers\etc

2.3.2 API Connect Toolkit i Local Testing Environments (LTE)

Els passos per a la seva instal·lació i configuració de API Connect Toolkit i Local Testing Environments (LTE) es poden trobar en el següent document adjunt:



Guia_Instal·lacio_i_Configuracio_LTE_IBM_A

2.3.3 Gitlab

Gitlab és la plataforma de desenvolupament de programari que proporciona eines per a la gestió del cicle de vida d'aplicacions i la col·laboració en projectes de desenvolupament de programari al CTTI.

2.3.3.1 Requisits previs

Per procedir amb la configuració, cal tenir els permisos demanats següents:

1. Tenir accés a Gitlab (<https://git.intranet.gencat.cat/>), si no en té, vegeu [apartat 2.1](#).
2. Tenir accés a GICAR, si no en té, vegeu [apartat 2.1](#).
3. Tenir Git instal·lat, vegeu [apartat 2.2.1](#)

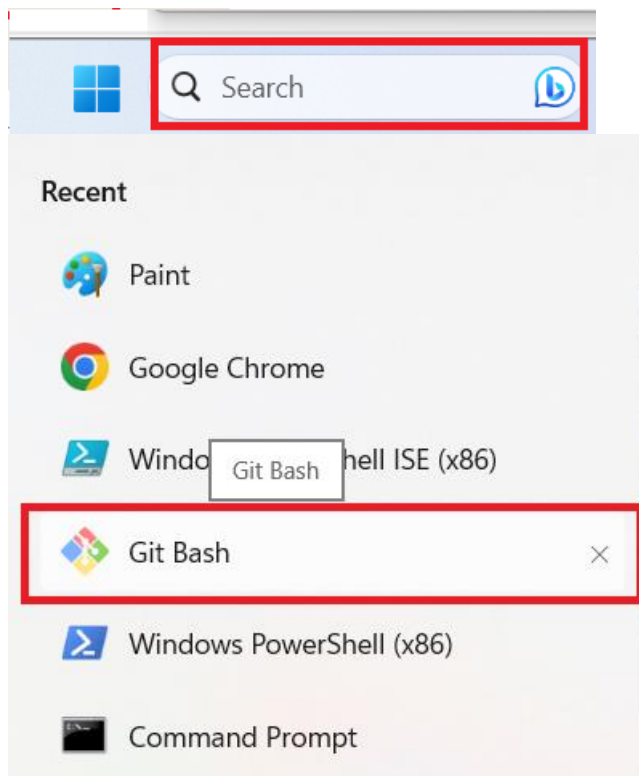
2.3.3.2 Configuració

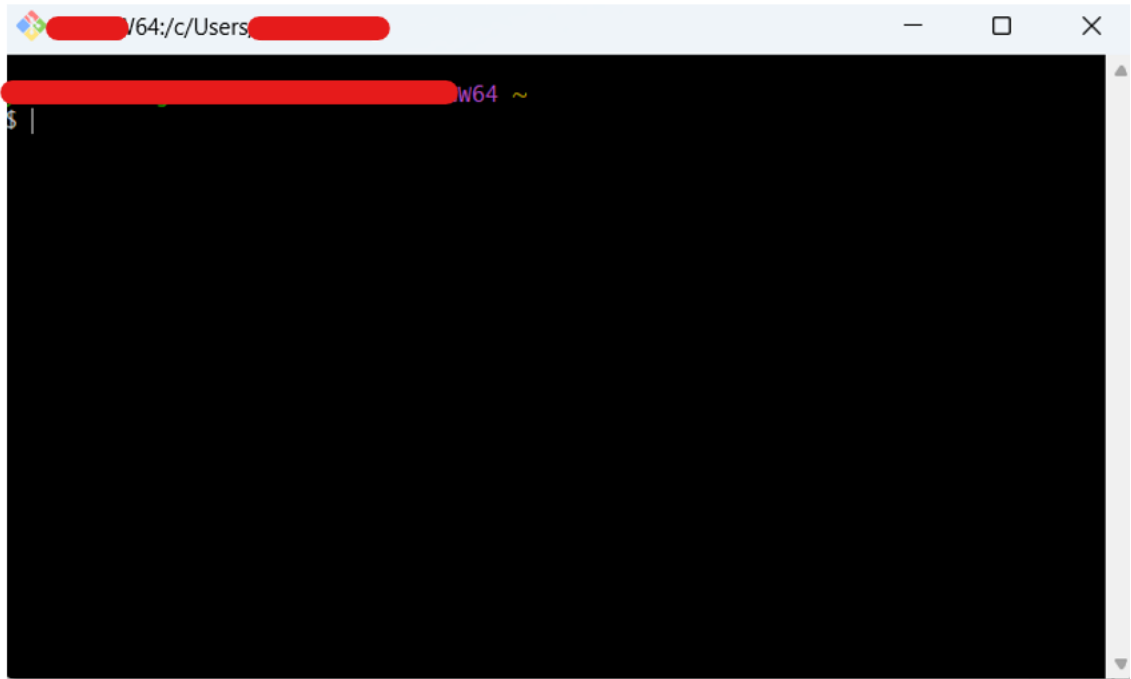
Git és un sistema de control de versions distribuït que farem servir en aquest guia per establir connexió i operar a Gitlab.

Nota: Cal esmentar que existeixen altres eines de Git disponibles, com Sourcetree, Tortoise Git, Fork..., però en aquest guia es farà servir Git com a exemple.

La configuració consta dels següents passos a seguir:

1. Obrir el Git Bash que hem instal·lat prèviament, a través de la barra de recerca de Windows.





2. Creem una carpeta que emmagatzemarà les claus que utilitzarem posteriorment per connectar-nos amb Gitlab.

```
mkdir ~/.ssh/gitlab_ssh
```

3. Generem aquestes claus mitjançant el següent comandament:

```
ssh-keygen -t rsa -b 4096 -C "<identificador>" -f  
~/.ssh/gitlab_ssh/myuser
```

Nota: "<identificador>" se substituirà pel **nom de l'usuari actual**, per exemple:

```
ssh-keygen -t rsa -b 4096 -C alberto.sanchez -f ~/.ssh/gitlab_ssh/myuser
```

4. Configurem el fitxer **config** de les claus públiques mitjançant un editor de text anomenat "nano", mitjançant el comandament:

```
nano ~/.ssh/config
```

5. Introduïm el següent text dins del fitxer (amb els espais inclosos)

```
Host git.intranet.gencat.cat  
  Hostname 10.54.67.30  
  PreferredAuthentications publickey  
  IdentityFile ~/.ssh/gitlab_ssh/myuser
```

```
4:/c/Users/.../.ssh
GNU nano 7.2 /c/Users/.../.ssh/config
Host git.intranet.gencat.cat
  Hostname 10.54.67.30
  PreferredAuthentications publickey
  IdentityFile ~/.ssh/gitlab_ssh/myuser
[ Read 4 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

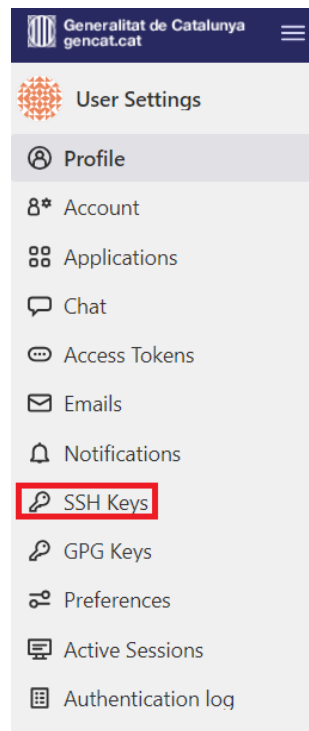
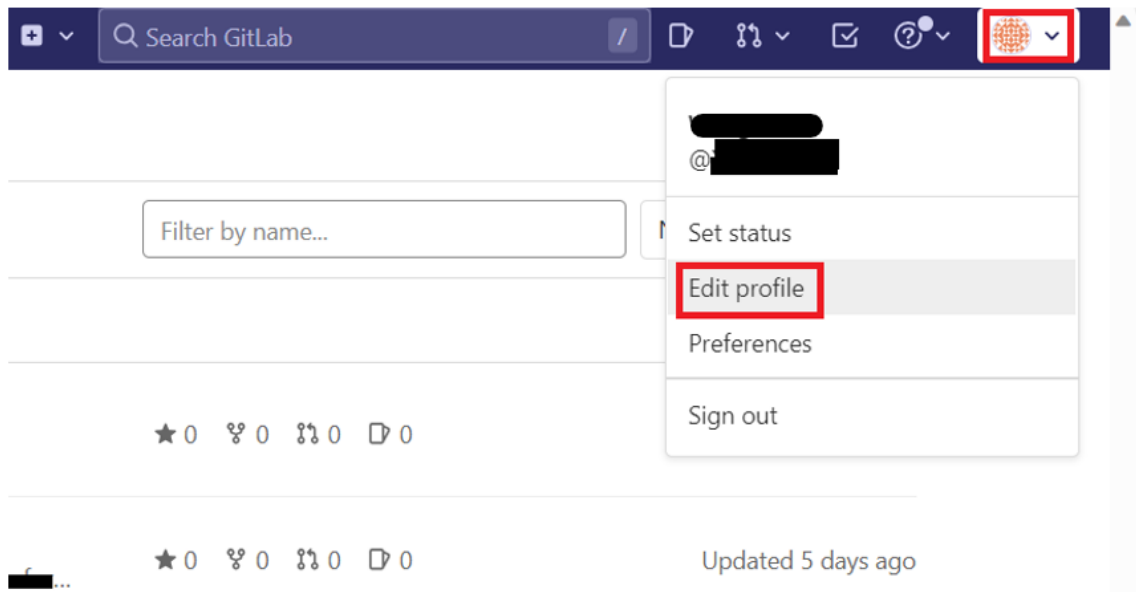
6. Posem les tecles CTRL+O i posem ENTER un cop per guardar els canvis i CTRL+X per sortir de l'editor de text.
7. A continuació, hem de recuperar la clau pública que hem generat en el pas 3. Per a això, introduïm el següent comandament:

```
cat ~/.ssh/gitlab_ssh/myuser.pub
```

Còpiem la clau que es mostra per la pantalla començant per ssh-rsa (l'inclouem en la còpia)

```
4 ~/.ssh
$ cat ~/.ssh/gitlab_ssh/myuser.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDdVFPmPgchfLwKFt3DwkWP/e0XfBtbPdLN38gUNNGN
3R1jKpD8LrvsgERSEarOMBURX3UFvmmwNSrGOI1bzC2RcV3gxpLqsmzztjvPNFsDQ/hyHxITzM iQA60Q
+H1LPNg0Nza61a7F2YqJp3ua1u4Yp0s0dDpf0pJpdQAnURzhAPw3tt+MmMNJPJEte0vVfBwpEPpMoysY
TSWcsySYEFERt+Tj+yFuq3YbGqWmTWZub667THA6NHmaFce7NbruuAX/a17mUChdYaEh+T/vWQl0tc0ayK
wQ==
```

8. Accedim a la pàgina de Git de CTTI mitjançant la següent URL: [Projects · Dashboard · GitLab \(gencat.cat\)](#) i ens identifiquem.
9. Importem la clau pública al nostre perfil de GitLab, accedint per "Icona – Edit Profile – SSH Keys".



10. Omplim els camps "Key", que correspon a la nostra clau pública, i "Title", que correspon a un identificador que li donem a aquesta clau (Pot ser qualsevol). Posteriorment li donem al botó "Add Key" per finalitzar la importació de la clau.

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

Add an SSH key for secure access to GitLab. [Learn more.](#)

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQAdVFPMpgchfLwKft3DWkWP/eOxftbPdLN38gUNN
GN3R1jkd8LrvsgERSEaroMBURX3UFvwmwNSrGOI1bzC2RcV3gxpLqsmzzjvPNFsdQ/hyHxITzMi
QA60Q+HILPNg0Nza61a7F2YqJp3ua1u4Yp0s0dDpf0jpdQAnURzhAPw3tt+MmMNPJEte0vVfB
wpFPpMoySYTSWsxSYFFERHji+yFug3YhGqWmIW7uh667THA6NHmaFce7NbruuAX/alZmUCHdY
```

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

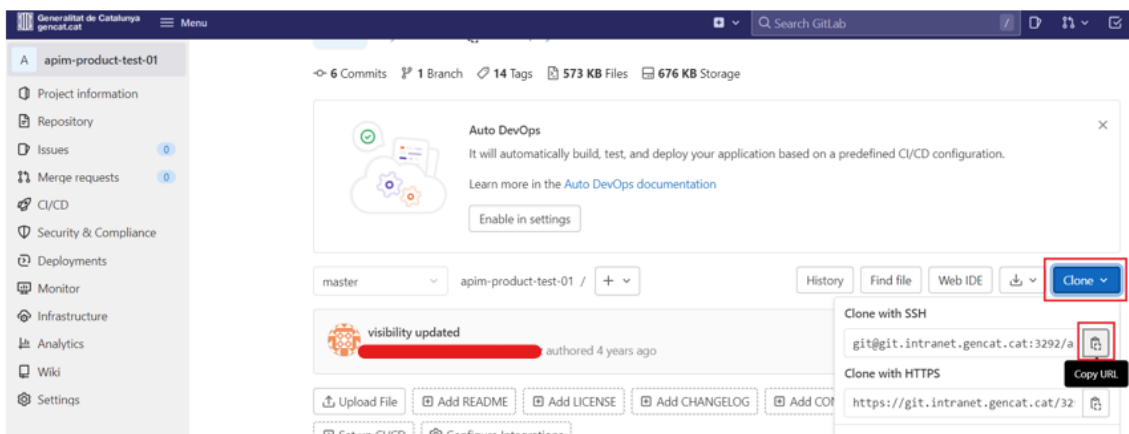
Title

Key titles are publicly visible.

Expiration date

Key can still be used after expiration.

- Una vegada finalitzada la configuració de la clau pública, accedeixen al projecte el repositori del qual vulguem clonar i li donem al botó "Clone" seguida pel botó "Copiar" en la secció "Clone with SSH".



- A continuació, tornem al nostre Git Bash i introduïm el següent comandament per clonar el repositori del nostre projecte.

```
git clone url-repositorio
```

Nota: (url-repositori) és la url que hem copiat al pas 11.

Qualsevol avís que ens pugui sortir per la consola, introduïm la lletra 'i' seguida d'un ENTER.

3 Desplegament de APIs

A continuació, es detallen els passos a realitzar per desplegar les APIs usant el pipeline del CTTI.

3.1 Consideracions prèvies dins del CTTI

L'organització de catàlegs, espais i productes és la següent:

- Actualment CTTI compta amb quatre catàlegs segons el tipus d' entorn i el tipus de visibilitat:
 - *privat_pre* i *privat* serien els entorns de preproducció i producció per desplegar en els catàlegs corresponents a la **Intranet**.
 - *public_pre* i *public* serien els entorns de preproducció i producció per desplegar en els catàlegs corresponents a l' **Internet**.
- Cada codi de diàleg disposarà d'un espai propi amb la nomenclatura "CD" + <codigo_diálogo>. Per exemple: "CD0192".
- Un producte és una agrupació d'APIs i plans que les acompanyen (unitat mínima a versionar, desplegar i subscriure).

Un cop fet el desenvolupament amb API Designer, caldrà exportar els YAMLS de definició del producte i els seus APIs per dipositar-los en el sistema de gestió de codi font (SCM - Source Code Management) del SIC d'acord amb les següents premisses:

- Cada producte es correspondrà amb un projecte dins del codi de diàleg adequat, de manera que tota la gestió posterior de pipelins i creació de peticions Remedy/JIRA s'eixirà a l'aplicació corresponent. Per aquest mateix motiu, no està contemplat la creació de subgrups de projectes, encara que l' eina ho permeti.
- Els projectes poden tenir tantes branques com siguin necessàries, però sempre s' haurà d' incloure la branca MASTER, ja que el contingut d' aquesta branca serà amb el que treballaran els pipelins de desplegament per defecte. No obstant això, el sistema permetrà opcionalment desenvolupar el codi font associat a la branca EVOLUTIUS.
- Aquest repositori no és un entorn de desenvolupament, per la qual cosa només les persones assignades com Release Managers seran les encarregades de consolidar el codi i lliurar-lo. Aquest codi font ja haurà d' estar validat en entorns de desenvolupament i es lliurarà quan es decideixi distribuir en els entorns dels serveis TIC centrals.
- Els pipelins seran les encarregades de generar els TAGS de Release de codi corresponents quan es desplegui amb èxit a Producció.

3.2 Creació de fitxers necessaris

De cara al funcionament correcte del pipeline cal que s' hagin creat prèviament uns arxius que s' allotjaran al Gitlab, bé al repositori del projecte o al projecte del SIC. Aquests fitxers són usats pel pipeline tant per a l' aprovisionament i creació del pipeline, com perquè el pipeline reculli el detall de la configuració de l' API a desplegar. Els fitxers són:

3.2.1 ACA (Arxiu generat pel desenvolupador)

Es tracta d' un **arxiu de configuració d' aplicacions imprescindible per a** qualsevol projecte **que s' ha d' allotjar en el repositori del projecte**, la funció del qual dins de l' autoservei de pipelins del CTTI, és la de configurar l' aplicació de tal manera que tingui activat el mode **Autoservei de Pipelins**, que és un servei mitjançant el qual es poden generar **automàticament** pipelins de construcció i desplegament d'aplicacions, amb el treball

col·laboratiu dels **proveïdors d'aplicacions i d'infraestructures** i sense la intervenció de l'equip del SIC. Per a més informació, consultar [Autoservei de pipelines \(gencat.cat\)](#).

Per tal de configurar la integració en el SIC, tots els projectes hauran de disposar d'una carpeta en el primer nivell i, dins d'aquesta carpeta, caldrà crear el fitxer **aca.yml**. Exemple de ruta :/sic/aca.yml

Un exemple del fitxer ACA seria el següent:

```
1 version: 2.0.0 # aca schema version
2 info:
3   version: 1.0.1
4   description: Api Connect Product
5 global-env:
6   - APIC_PRODUCT_FILE: 'product.yml'
7   - APP_NAME: 'technicalproduct'
8   - APIC_PRESERVE_ASSEMBLY: true
9 components:
10  - deployment:
11      environments:
12        - name: privat_pre
13          actions:
14            deploy:
15              steps:
16                - execution:
17                  env:
18                    - APIC_TARGET_URL: 'https://backend/pre'
19        - name: privat
20          actions:
21            deploy:
22              steps:
23                - execution:
24                  env:
25                    - APIC_TARGET_URL: 'https://backend/pro'
26 notifications:
27   email:
28     recipients:
29       - recipiente@correo.es
30
```

On els punts clau per tenir en compte serien els següents:

- **Version:** versió (independent de la versió de l'aplicació) que s'utilitza per fer seguiment de l'arxiu de configuració.
- **Environments:** Entorns als quals es desplega l'aplicació, així com el seu ordre i la modalitat de desenvolupament desitjada. Actualment CTTI compta amb quatre catàlegs segons el tipus d'entorn i el tipus de visibilitat:
 - **privat_pre** i **privat** serien els entorns de preproducció i producció per desplegar en els catàlegs corresponents a la **Intranet**.
 - **public_pre** i **public** serien els entorns de preproducció i producció per desplegar en els catàlegs corresponents a l' **Internet**.

En l'exemple exposat a dalt el desplegament de l'API es realitzaria en els entorns de privat_pre (preproducció) i privat (producció) pertanyents al catàleg d'intranet.

Nota: S'hauria d'indicar en aquest fitxer només els entorns on es vagi a desplegar el producte i l'API (podent ser un o diversos, en funció de la necessitat del projecte).

A més, cal destacar que en el cas que el pipeline no permet desplegaments mixtos; és a dir, perquè es pugui desplegar a l'entorn de Producció Externa "public", és

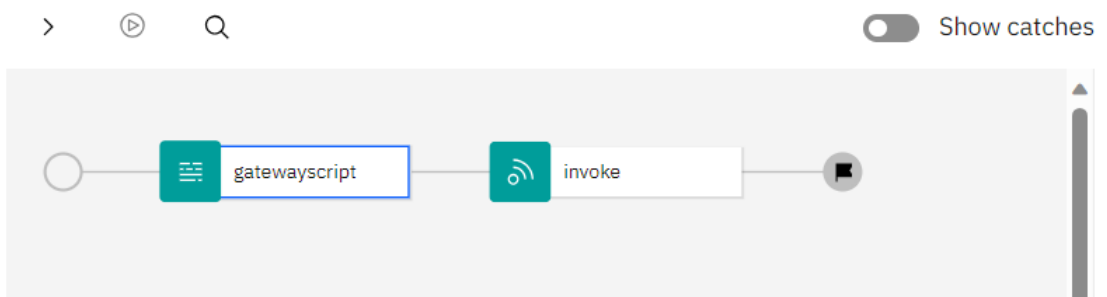
necessari haver desplegat explícitament a l'entorn de Preproducció Extern "public_pre".

- **Notificacions:** Definició d'adreces de correu electrònic on notificar accions manuals en espera i resultats de l'execució.
- **APP_NAME:** Nom corresponent al producte que es vagi a desplegar. Aquest nom no ha de contenir el codi de diàleg del projecte, ja que aquest serà inclòs de manera automàtica després de realitzar-se el desplegament amb el format <codi>-<APP_NAME>. Per exemple, si el nom del camp del YAML del producte té configurat el valor 3292-technicalproduct, el valor que s'indicarà en APP_NAME del fitxer ACA seria technicalproduct.

En cas que no es configuri aquesta variable en el ACA, se li assignarà per defecte el nom del projecte pel nom del repositori de GIT.

- **APIC_PRODUCT_FILE (opcional):** Ruta i nom del fitxer descriptor per al desplegament de l'aplicació a l'API Manager. La variable només serà requerida en cas que la ruta i/o nom del fitxer difereixi del suggerit.
 - *Valor per defecte: product.yml*
- **APIC_PRESERVE_ASSEMBLY (opcional):** Curricular que indica si s'ignora o no la regla que reemplaça l'acoblament, a través dels valors *true* o *false*, conservant l'acoblament original del YAML de l'API (secció assembly al YAML) creat pel desenvolupador, en cas que el camp s'informi amb el valor *true*. Això permetrà desplegar l'API amb lògica personalitzada.

En el cas que no s'inclouï aquesta variable o que s'informi a *false*, s'executaran les regles establertes en el pipeline, que substitueixen la lògica de l'API a desplegar l'acoblament, per una configuració estàndard en la qual només es realitza una trucada a l'endpoint definit en el fitxer aca.yaml (*APIC_TARGET_URL* o *APIC_TARGET_URL_{N}*). L'acoblament resultant seria el següent:



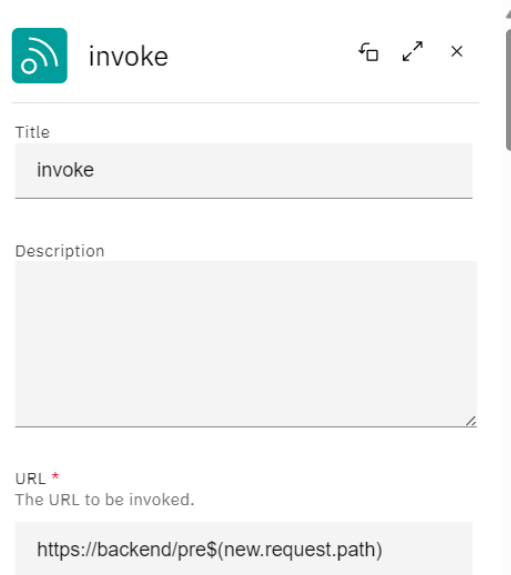
- **Gatewayscript.** *Contindria la següent informació:*

```
1 const requestPath = context.get('request
2   .path');
3 const newRequestPath = requestPath.replace
4   (/^\d*\.\d*/, "");
5 context.set('new.request.path',
6   newRequestPath);
```

Es recull la via d'accés (*request.path*) de l'API i es guarda en una constant. Se li dona format a aquesta constant i es genera la variable

de context '*new.request.path*', que posteriorment serà utilitzada en la política d'Invoke.

- **Invoke.** *Contindria la següent informació:*
<APIC_TARGET_URL>\$(*new.request.path*)



- **APIC_TARGET_URL:** URL de destinació de les APIs si és comú a totes, tot i que pot conviure amb *APIC_TARGET_URL_{N}* per especificitats.
NOTA: si el valor de *APIC_PRESERVE_ASSEMBLY* és true, s'ignorarà el valor que es posi en aquest camp.
 - *Exemple: APIC_TARGET_URL: 'https://backend/api'.*
- **APIC_TARGET_URL_{N} (opcional):** URL de destinació definida per a una API específica dins del producte, que no coincideixi amb la URL de destinació comuna per a totes. Permetent d'aquesta manera definir especificitats, encara que pot conviure amb *APIC_TARGET_URL global*.
NOTA: si el valor de *APIC_PRESERVE_ASSEMBLY* és true, s'ignorarà el valor que es posi en aquest camp.
 - Format de la clau: *APIC_TARGET_URL_{0-*9a-*zA-Z}*
 - Format del valor: *<api-file-name-with-extension>:<target-uri>* Per consultar més informació sobre el fitxer **Arxiu de Configuració d'Aplicacions**, consultar el [punt d'annex](#).
 - *Exemple: APIC_TARGET_URL_1: 'api_1.0.0.yml:https://backend/apio'*

Seguint aquest format, podríem construir el nostre fitxer ACA, per poder desplegar el nostre producte amb els seus corresponents APIs, en els entorns requerits dins de CTTI.

3.2.2 ACS (Arxiu generat per SIC)

L' ACS és l' arxiu de configuració generat i **proporcionat per part de l' equip de SIC**, el qual conté la configuració interna per al funcionament del pipeline i altres configuracions necessàries.

Un exemple d' un fitxer ACS és el que es mostra a continuació:

```
1 version: 2.0.0
2 global-env:
3   - CONTAINER_REGISTRY_PROJECT: project
4   - MAVEN_OPTS: '-Dmaven.deploy.skip=false'
5   - JOB_TEMPLATE: 'job-volume.yaml'
6
7   - APIC_ORGANIZATION: 'org'
8   - APIC_SPACE: 'space'
9   - APIC_PLANFILE: 'file.yaml'
10  - APIC_CATALOG_DES: 'des'
11  - APIC_CATALOG_PUB: 'null'
12
13 components:
14 - deployment:
15   actions:
16   deploy:
17   steps:
18   - container:
19     image:
20     remote:
21     name: registreimatges.sic.intranet.gencat.cat/gencat-sic/apic-deployer:3.0
22   resources:
23     limits:
24     cpu: 2000m
25     memory: 2Gi
26     requests:
27     cpu: 500m
28     memory: 512Mi
29   execution:
30   commands:
31   - run.sh $APIC_ORGANIZATION $APIC_SPACE $APIC_PLANFILE $APIC_CATALOG $APIC_PRODUCT_FILE $APIC_PLAN_MAP
32   environments:
33   - name: privat
34     confirmation:
35     required: true
36     staging: true
37     production: false
38     actions:
39     deploy:
40     steps:
41     - execution:
42       credentials:
43       - id: harbor-credentials-id
44         basic:
45         username: REGISTRY_USERNAME
46         password: REGISTRY_PASSWORD
47     env:
48     - NAMESPACE: null
49     - APIC_CATALOG: null
50     - JOB_ENVS: DB_TYPE=null|DB_HOST=null|DB_PORT=null|DB_DATABASE=null|DB_USERNAME=null|DB_PASSWORD=null
51
52   - name: privat_pre
53     confirmation:
54     required: false
55     staging: true
56     production: false
57     actions:
58     deploy:
59     steps:
60     - execution:
61       credentials:
62       - id: harbor-credentials-id
63         basic:
64         username: REGISTRY_USERNAME
65         password: REGISTRY_PASSWORD
66     env:
67     - NAMESPACE: null
68     - APIC_CATALOG: null
69     - JOB_ENVS: DB_TYPE=null|DB_HOST=null|DB_PORT=null|DB_DATABASE=
70     =null|DB_USERNAME=null|DB_PASSWORD=null
```

Alguns dels camps més importants de l' ACS són:

- **Remot:** Imatge i versió del contenidor constructor a utilitzar entre les disponibles en el registre d'imatges corporatiu (per a més informació consultar [Registre d'imatges \(gencat.cat\)](#)).

- **Resources:** recursos a assignar per a la correcta construcció del projecte (CPU i memòria).
- **Global-env:** Relació de variables globals necessàries per a l'execució del pipeline.
- **Credentials:** Credencial que s'aporta a la imatge perquè es pugui executar.
- **Commands:** Operacions (scripts) que s'executen per a la construcció del projecte.

3.2.3 ACD (Arxiu generat per SIC)

L'ACD és l'**arxiu de configuració de departament**, el contingut del qual detalla la informació sobre l'àrea tècnica del servei del CTTI, Datacenter a usar on es desplegarà l'API, l'identificador del proveïdor/projecte i el codi del departament o àrea tècnica del CTTI. L'**equip de SIC és el responsable de proporcionar i mantenir aquest arxiu.**

Un exemple d'un fitxer ACD és el següent:

```
version: 2.0.0
info:
  code: '3292'
  department: CTT
  short-name: APM
  app-provider: AM22_23
  provider: cpd2-public
  itsm-services:
    - "undefined"
```

Alguns dels camps més importants de l'ACD són:

- **Versió:** versió de l'arxiu de configuració.
- **Code:** Codi identificatiu del departament.
- **Department:** Nom del departament.
- **App-provider:** Identificador del proveïdor de l'aplicació.

3.2.4 ACI (Arxiu generat per SIC)

L'ACI és l'**arxiu de configuració d'infraestructures** (pot tenir-ne més d'un per aplicació-projecte), el nom del qual (sense incloure l'extensió) serà l'identificador facilitat al proveïdor d'aplicacions. Serà la responsabilitat del **proveïdor d'infraestructures** mantenir actualitzada aquesta informació i de notificar al proveïdor d'aplicacions quan s'hagi realitzat algun canvi sobre el fitxer.

Es tracta d'un arxiu de text en format YAML, que serà responsabilitat del proveïdor d'infraestructures mantenir-lo actualitzat, en el qual s'ha d'aportar la següent informació:

- **Versió:** versió de l'arxiu de configuració.
- **Components:** secció que recull tots els recursos de la part d'infraestructures. Actualment, només existeix el detall de cada infraestructura.
- **Infrastructure:** detall de les infraestructures incloses en aquest arxiu de configuració.

Un exemple d'un fitxer ACI és el següent:

```

1 version: 2.0.0
2 components:
3   - infrastructure:
4     type: apim
5     platform: cpd2-public-01
6     family: app
7     deployment:
8       environments:
9         - name: privat_pre
10          actions:
11            deploy:
12              steps:
13                - execution:
14                  credentials:
15                    - id: cpd2-public-api-token
16                      token:
17                        var: IBMCLLOUD_LOGIN_TOKEN
18          - name: privat
19            actions:
20              deploy:
21                steps:
22                  - execution:
23                    credentials:
24                      - id: cpd2-public-api-token
25                        token:
26                          var: IBMCLLOUD_LOGIN_TOKEN
27

```

3.3 Pujada de carpetes i fitxers al repositori

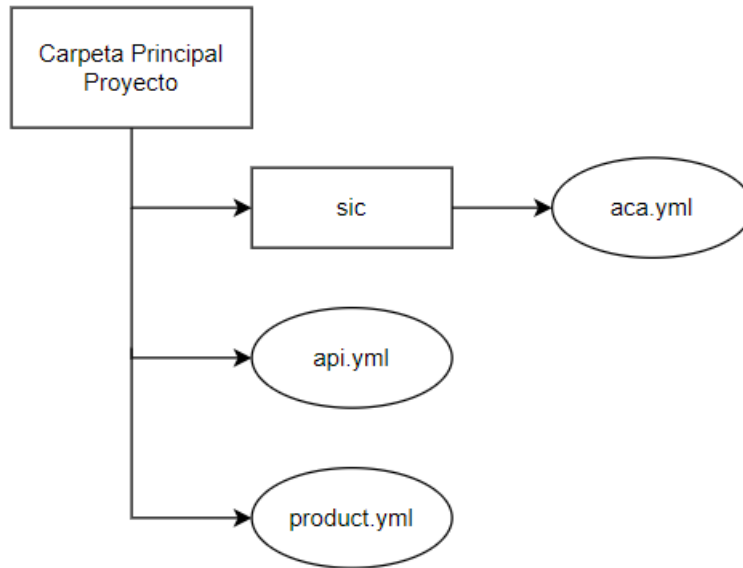
Una vegada finalitzades les fases de disseny tant de les APIs com dels productes, on s'han configurat els fitxers YAML corresponents a cadascun d'ells, s'hauran de pujar aquests fitxers al repositori assignat al projecte, per posteriorment procedir amb el desplegament entre entorns.

A continuació, es mostren els passos a seguir en el cas de l'ús de Gitlab des de la interfície web. En cas d'usar un client Git com Fork, TortoiseGit o Sourcetree, l'execució dels passos serà diferent, encara que el concepte darrere de cada pas és el mateix:

1. Accedim al [GitLab \(gencat.cat\)](https://gitlab.com) i ens lloguem amb la credencial de GICAR.
2. Accedim al repositori del projecte que vulguem desplegar i li donem al botó "Upload File".

The screenshot shows the GitLab repository interface. At the top, it displays repository statistics: 13 Commits, 2 Branches, 0 Tags, 184 KB Files, and 184 KB Storage. Below this is a notification for 'Auto DevOps' with an 'Enable in settings' button. The main area shows a merge branch dropdown set to 'master', a commit hash, and buttons for 'History', 'Find file', 'Web IDE', 'Download', and 'Clone'. A merge commit summary is visible, followed by a row of buttons: 'Upload File' (highlighted with a red box), 'README', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', and 'Set up CI/CD'. Below this is a 'Configure Integrations' button and a table header with columns for 'Name', 'Last commit', and 'Last update'.

3. Triem els fitxers que vulguem pujar (ho pugem un a un), tenint en compte que l'estructura del repositori ha de ser el següent:



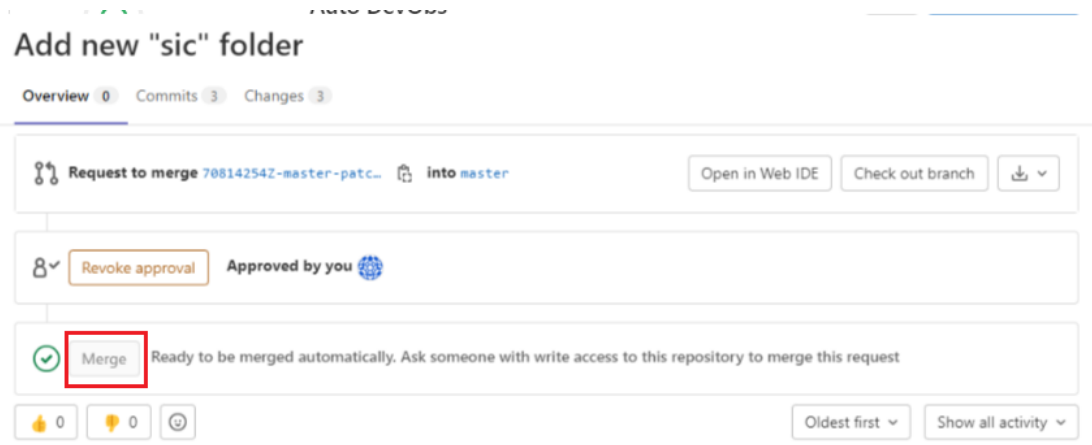
Nota:

- Un producte pot contenir diverses APIs, per la qual cosa un projecte de Gitlab pot contenir la quantitat d'APIs que es necessiti, sempre que estiguin referenciats en el corresponent producte (camp "apis" del YAML del producte).
- En el cas que es puguin APIs de tipus SOAP o SOAP-TO-REST, cal pujar l'arxiu ZIP de WSDL a la mateixa altura que el fitxer YAML de l'API. El YAML s'haurà d'haver generat abans amb l'ajuda de l'API Designer/Toolkit, que permet importar un fitxer WSDL i generar un YAML amb una API SOAP o SOAP-TO-REST definida en base a l'esmentat WSDL.
Un exemple d'un fragment del iml d'una api generada en base al WSDL seria el següent:

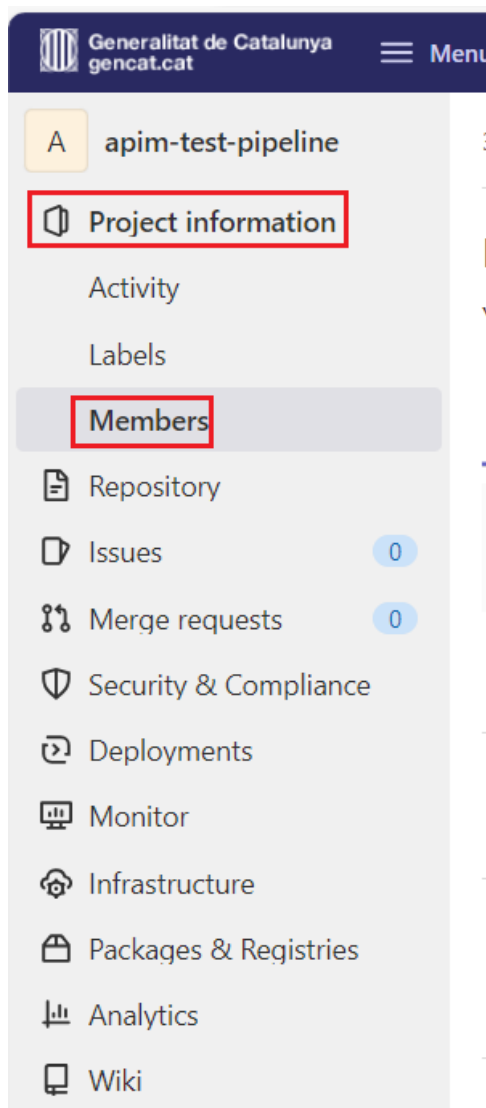
```

wsdl-definition:
  wsdl: wsdlFileName.zip
  service: wsdlNameService
  port: wsdlNameImplPort
  soap-version: '1.0'
  
```

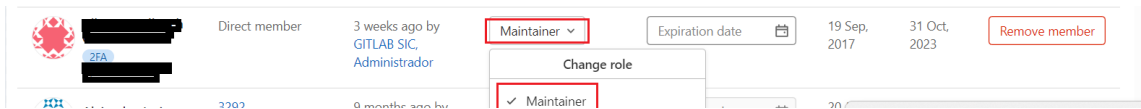
4. En el cas que s'hagi creat una branca nova diferent al "master" (branca principal), el següent pas és demanar el mergeig de la branca creada amb la branca principal. Per a això, es necessita l'aprovació d'un membre que tingui el permís de Maintainer (concedit pel Release Manager) o Release Manager (concedit pel SIC).



5. Per donar permís de *Maintainer* com *Release Manager*, naveguem per la secció "Project Information – Members".



6. Seleccionem el membre al qual vulguem assignar el permís de *Maintaner*.



3.4 Desplegament al Pipeline

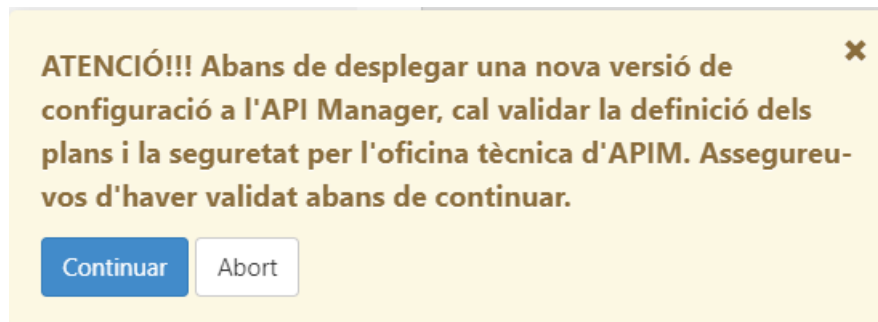
Un cop tinguem preparat el repositori amb els fitxers necessaris al Gitlab, procedirem a realitzar les proves de pipeline a l'entorn de preproducció per provar la funcionalitat de l'API desenvolupada (alternativament es pot provar a l'entorn local amb Toolkit, vegeu [apartat 2.3.2](#)).

3.4.1 Check de validació per als plans i seguretat

De cara a poder resoldre els gaps identificats en el pipeline respecte als plans de consum i la configuració de seguretat, s'ha adoptat com a solució temporal l'opció en la qual es requereix la intervenció per part de l'OFT d'API-M, que realitzarà les corresponents validacions de la definició dels plans i seguretat, de forma manual, sobre els productes i APIs que s'hagin de desplegar.

Perquè es pugui procedir amb aquesta metodologia, s'ha inclòs un nou pas en el desplegament de l'API, el procediment del qual és el següent:

1. En llançar el pipeline de l'API, l'agent del projecte que ha iniciat l'execució rebrà un missatge de confirmació per la consola (input), indicant-li que l'OFT ha d'haver fet prèviament la validació de la definició de seguretat sobre el YAML del seu producte abans de poder procedir amb el desplegament.



Missatge d'alerta (Check de plans i seguretat)

2. L'agent pot realitzar dues operacions davant d'aquesta alerta:
 - a. **Confirmar i procedir amb el desplegament**, de manera que es reprèn l'execució del pipeline i el producte quedarà desplegat en cas que l'execució finalitzi correctament. Si el responsable de l'execució decideix continuar amb l'execució sense haver acudit prèviament a l'OFT, aquest serà el responsable d'assumir els riscos i conseqüències que suposa la pujada d'un producte sense validar, i l'OFT podrà prendre accions en conseqüència.
 - b. **Cancel·lar i interrompre el desplegament**, de manera que es cancel·la l'execució del pipeline, i l'agent serà el responsable de notificar al seu equip la necessitat de consultar a l'OFT per fer les validacions corresponents.

3.4.2 Passos per executar el pipeline

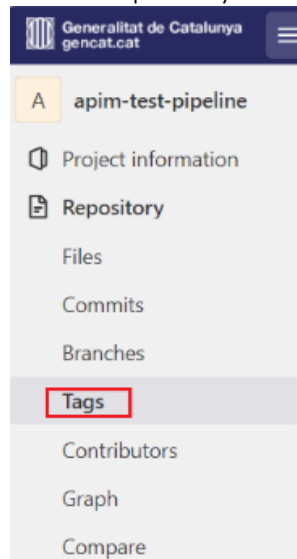
Un cop pujats els arxius necessaris, es procedirà amb el desplegament del producte i l' API a l' entorn de preproducció, el procediment del qual és el següent:

1. Comprovació i eliminació de l' existència del tag associat a la versió que volem desplegar.

Nota: Cada vegada que es desplega un producte, es genera un tag associat a la versió a desplegar, perquè des del SIC es pugui tenir una traçabilitat de les versions desplegades. Per tant, **a partir del primer desplegament, si es vol desplegar la mateixa versió del producte, per exemple, perquè s' hagi solucionat algun problema detectat en els desplegaments després de fer unes proves durant el cicle de desenvolupament, s' ha d' aplicar el workaround establert, esborrant de forma manual el tag creat associat a la versió del producte al GIT, per part del Release Manager o Maintainers assignats al projecte.**

(En el cas que no s'apliqui aquest workaround, es produirà l'error de "Ja hi ha un tag definitiu amb la versió: X.x.x" pel duplicat del tag al GIT).

- Accedim a la pestanya de tags a la part esquerra del Git



- Seleccionem la versió a esborrar i procedim amb la seva eliminació.



Delete tag



Deleting the 5.0.6 tag cannot be undone. Are you sure?

Cancel

Delete tag

- Després d'haver-nos cerciorat de l'eliminació del tag corresponent a la versió que volem desplegar, podem procedir amb el següent pas.
- 2. Accedim a Jenkins [Dashboard \[Jenkins\] \(gencat.cat\)](#) i ens identifiquem amb les credencials de GICAR.
- 3. Dins del Dashboard, accedeixen al projecte que correspon:

Bienvenidos al Servicio de Integración Continua de la Generalidad de Cataluña

Jenkins es la herramienta implantada en el SIC para la integración continua en el desarrollo de software. Se trata de un servicio en el que, a partir de la definición previa de tareas (jobs), se construyen las aplicaciones, se versionan, se realizan análisis de calidad, se ejecutan tests y se despliegan en los entornos preproductivos y productivos. Aspectos más relevantes de la versión actual:

- Acceso a la herramienta mediante las **credenciales GICAR**
- Tipo de Job utilizado actualmente: **Pipeline declarativa**

Podrá encontrar más información en el portal de Arquitectura, bajo el apartado [SIC](#)

[Ajo](#)

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		-APM	N/A	N/A	N/A	

Icon: S M L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

- 4. Seleccionem la carpeta del pipeline creat per part del SIC.

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		-APM-apim-test-pipeline	N/A	N/A	N/A	

- 5. ... i el pipeline que vulguem provar.

Nota: Es poden executar diversos tipus de pipeline, com ara PUBLISH, per publicar un producte i api, INFO per obtenir informacions dels productes desplegats en un catàleg concret, i altres pipelins com DEPRECATE, RETIRE, SUPERSEDE, DELETE i REPLACE que es troben a la carpeta "Advanced". Per informar-se del funcionament de cadascun dels pipelins, consultar el següent enllaç: [Autoservei de pipelines \(gencat.cat\)](#).

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
⊗	☁	██████-APM-apim-test-pipeline	2 hr 45 min #23	1 hr 57 min #25	3 min 44 seco	▶ ☆
⋮	☀	██████-APM-apim-test-pipeline-INFO	N/A	N/A	N/A	▶ ☆
📁	☀	Advanced	N/A	N/A	N/A	☆
📁	☁	Aux	N/A	N/A	N/A	☆

6. Al menú de l'esquerra, seleccionem "Build with Parameters".

- ☰ Status
- </> Changes
- ▶ Build with Parameters
- 🔍 Full Stage View
- 🛡 Embeddable Build Status

☁ Build History
trend ▼

🔍 Filter builds...
/

Pipeline

Full project name
Api Connect Pro

Stage View

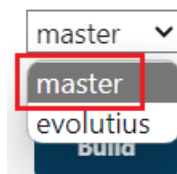
7. Seleccionem la branca "master" per desplegar la nostra branca principal del repositori i executem el pipeline donant-li a "Build".

Pipeline ████████-APM-apim-test-pipeline

This build requires parameters:

gitBranch

App git branch



Pipeline █████ -APM-apim-test-pipeline

This build requires parameters:

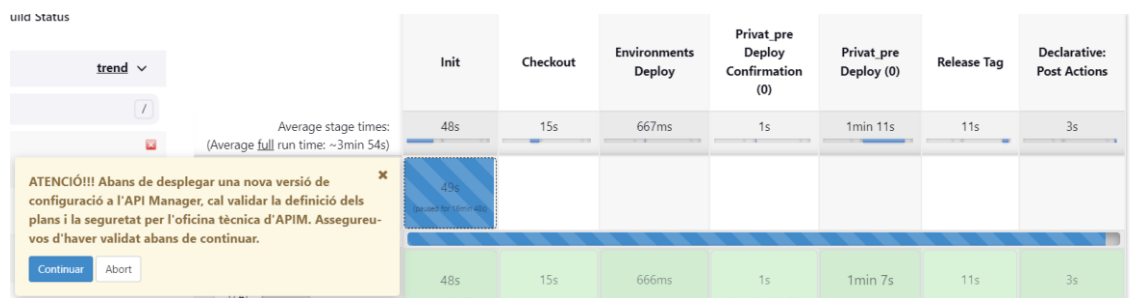
gitBranch

App git branch

master ▾



- Es pot veure que, en iniciar l'execució, es mostra per pantalla un missatge de confirmació per a l'agent, indicant-li que és necessària la validació de la definició dels plans i seguretat per part de l'OFT abans de continuar amb el desplegament. Posem sobre el botó "Continuar" per seguir amb l'execució.



- Al llarg de l'execució, per la consola del pipeline es demanarà a l'agent perquè introdueixi els paràmetres necessaris per procedir amb l'execució. Li donem click sobre l'enllaç proporcionat.

```
2023-11-13 16:01:05.016 [Pipeline] {  
2023-11-13 16:01:05.135 [Pipeline] input  
2023-11-13 16:01:05.184 Input requested
```

- Seleccionem l'entorn per on es desplegarà el nostre producte.
Nota: En aquest exemple, com que només tenim configurat en el fitxer aca l'entorn de reproducció privat, a la consola només es mostra l'opció de desplegar-se en PRE.

Introdueixi les dades i premi Continuar.

ENVIRONMENT

Catàleg de la API que desitja informació, a partir dels existents a la configuració del fitxer ACA.

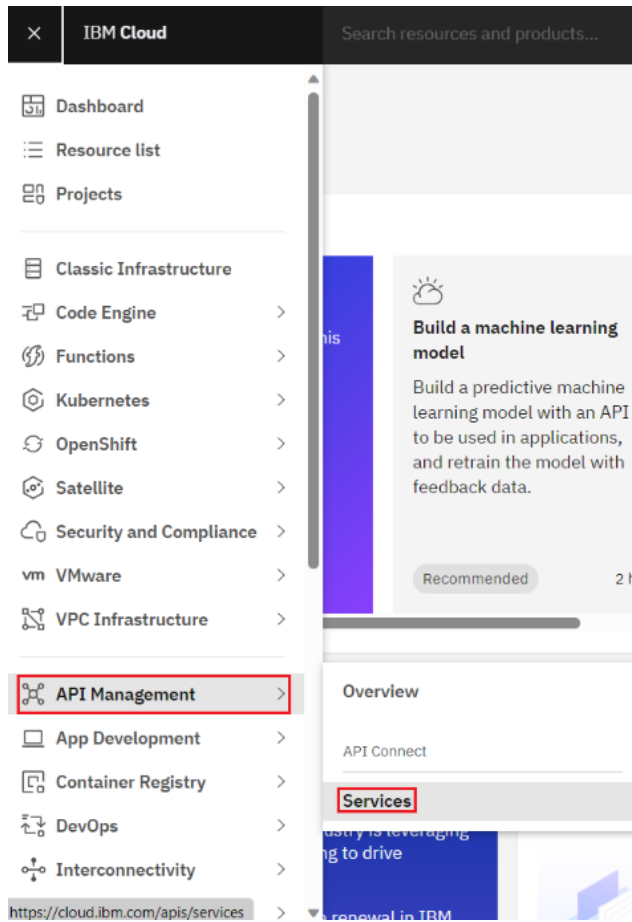
privat_pre-0



11. En el cas que el pipeline s'ha executat correctament, es mostrarà per la consola el missatge de SUCCESS.

```
2023-11-14 09:26:37.489 [Pipeline] // stage
2023-11-14 09:26:37.528 [Pipeline] End of Pipeline
2023-11-14 09:26:37.619 Finished: SUCCESS
```

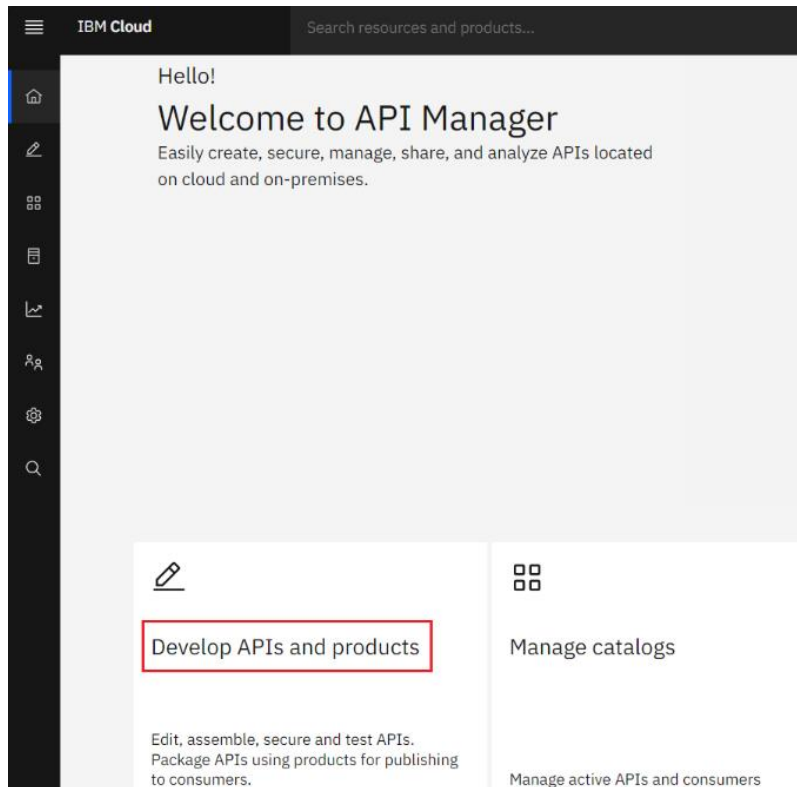
12. Després, per comprovar que el producte i l'api s'ha desplegat correctament, ens dirigim a <https://cloud.ibm.com/authorize/gicar> i ens identifiquem amb les credencials de GICAR.
13. Un cop dins, accedim al menú ubicat a l'esquerra i accedim a "API Management – Services".



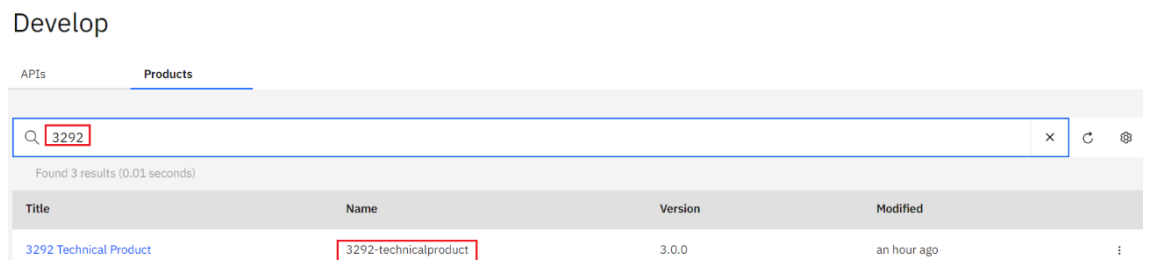
14. Ara accedeixen dins de CTTI.



15. Accedim a la secció "Develop APIs and products".



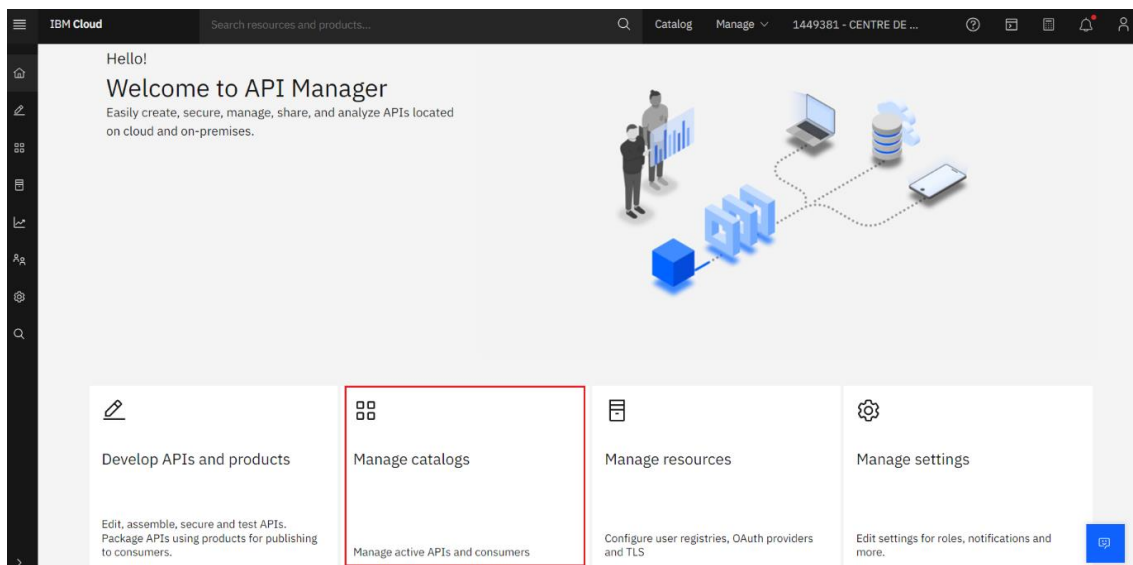
16. A la barra de recerca de productes, introduïm el nom del nostre producte desplegat i confirmem que s'ha desplegat correctament el draft del producte.



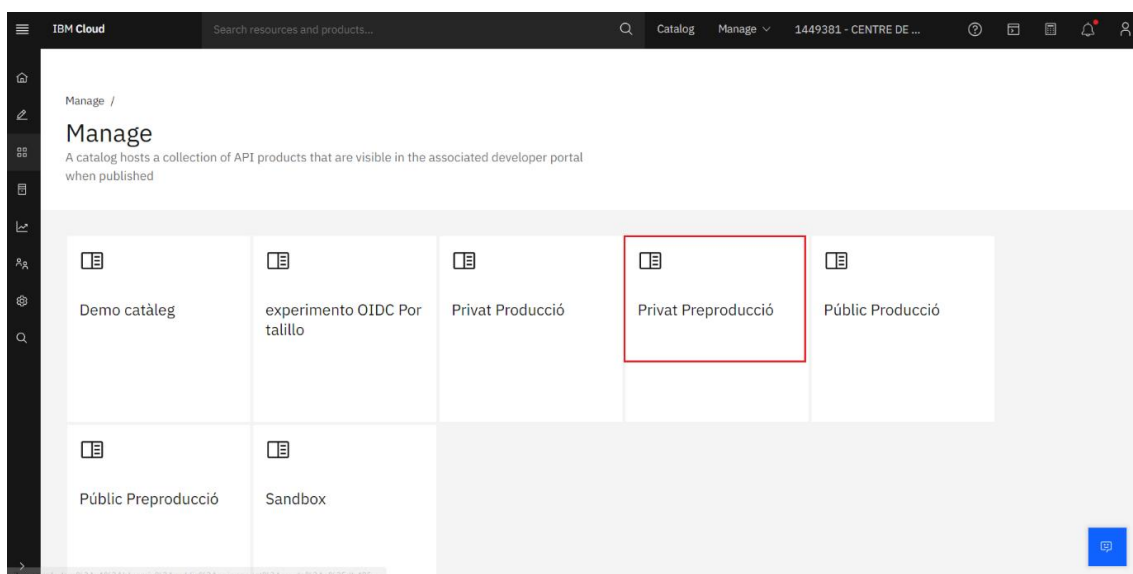
17. A la barra de recerca d'APIs, introduïm el nom de la nostra API desplegada i confirmem que s'ha desplegat correctament el draft de l'API.



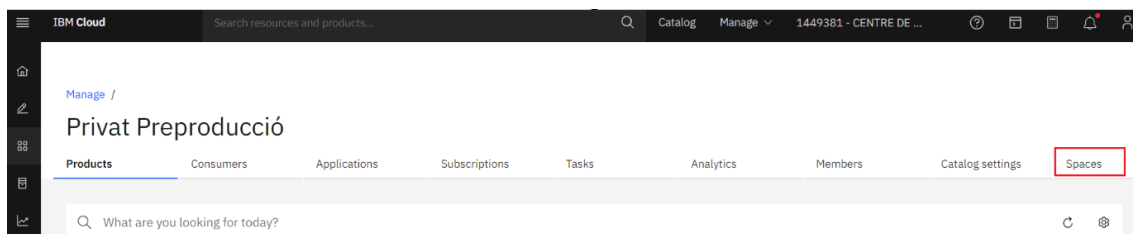
18. A continuació, comprovem que el producte s'ha desplegat al catàleg. Per a això, accedim a la secció "Manage Catalogs".



19. Accedim al catàleg per on s'ha desplegat el nostre producte, que en aquest cas és Privat Preproducció.



20. Ara entrem al nostre espai assignat dins del catàleg, que és el CD3292.



IBM Cloud Search resources and products... Catalog Manage

Manage / Privat Preproducció

Create and manage the spaces in your catalog; spaces allow you to partition your catalog to support different API provider development teams

Products Consumers Applications Subscriptions Tasks Analytics Members Catalog settings Spaces

CD0192 Date created 06/02/2022	CD0433 Date created 10/18/2023	CD0434 Date created 12/22/2022	CD0441 Date created 09/26/2023
CD0463 Date created 12/21/2022	CD1370 Date created 04/20/2022	CD1889 Date created 02/02/2023	CD3076 Date created 10/09/2023
CD3292	CD3801	DummySpace	HealthCheck

21. Dins de l' espai, podem comprovar els productes que s' han desplegat en el catàleg seleccionat.

IBM Cloud Search resources and products... Catalog Manage 1449381 - CENTRE DE ...

Manage / Privat Preproducció / CD3292


Products Consumers Applications Subscriptions Tasks Analytics Members Space settings


What are you looking for today?


Title	Name	Version	State	Plans	Subscriptions	Modified	
3292 Technical Product	3292-technicalproduct	6.0.1	published	4 Plans	0	a day ago	:
3292 Technical Product	3292-technicalproduct	5.0.2	published	4 Plans	0	a day ago	:
3292 Technical Product	3292-technicalproduct	6.0.0	published	4 Plans	0	a day ago	:
3292 Technical Product	3292-technicalproduct	5.0.0	retired	4 Plans	0	3 days ago	:
3292 Technical Product	3292-technicalproduct	5.0.3	published	4 Plans	1	3 days ago	:
3292 Technical Product	3292-technicalproduct	3.0.0	deprecated	4 Plans	0	3 days ago	:
3292 Technical Product	3292-technicalproduct	5.0.5	published	4 Plans	0	3 days ago	:


3.4.3 Consultar logs


Per consultar els logs de pipeline, accedim al "Build History" al menú de l'esquerra.


 Status |


 Changes |









 Build with Parameters |

 Full Stage View |

 Embeddable Build Status


 **Build History** trend ▾


 Filter builds... /

 #27	Nov 8, 2023, 5:10 PM	  
 #26	Nov 8, 2023, 3:55 PM	
 #25	Nov 8, 2023, 1:26 PM	
 #24	Nov 8, 2023, 12:56 PM	
 #23		


Seleccionem ara el número d'execució que vulguem investigar per consultar els detalls de l'execució.

Build #27 (Nov 8, 2023, 5:10:52 PM)


 Started by user [Yihui Wang](#)

 This run spent:

- 30 sec waiting;
- 6 min 16 sec build duration;
- 6 min 16 sec total from scheduled to completion.

 **git** Revision: ad461e85812e5e0c9782fbd427caa131b03ec4f4
 Repository: <https://git.intranet.gencat.cat/0192/sic-resources/shared-libs/pipeline.git>

- 2.2.0

 **git** Revision: 37392c5a39b8c25ee8e1bea5a76615b2e2eed844
 Repository: <https://git.intranet.gencat.cat/0192/sic-resources/shared-libs/common.git>

- 2.2.0

 **git** Revision: 50fa45e41535861db4cff8c3a0c5d31fc77fa1ba
 Repository: <https://git.intranet.gencat.cat/0192/sic-resources/shared-libs/deploy.git>

- 2.0.0

En aquesta secció es mostren els informes com el temps d'execució, l'iniciador de l'execució, i la imatge del pipeline que s'està usant. Després per consultar el log, accedim a la secció "Console Output".



Status



Changes



Console Output



Edit Build Information



Parameters



Timings

En aquesta secció es veurà amb detall tots els logs que s' han generat al llarg de l'execució, juntament amb la informació dels errors, si escau.



Console Output

```
2023-11-08 17:10:52.164 Started by user [REDACTED]
2023-11-08 17:10:52.164 Running as [REDACTED]
2023-11-08 17:10:52.863 Obtained apim-cpd2-public-01/publish-pipeline.groovy from git https://git.intranet.gencat.cat/0192/sic-resources/pipelines/dsl-pipeline.git
2023-11-08 17:10:52.884 Loading library pipeline@2.2.0
2023-11-08 17:10:52.884 Attempting to resolve 2.2.0 from remote references...
2023-11-08 17:10:52.884 > git --version # timeout=10
2023-11-08 17:10:52.895 > git --version # 'git version 2.38.3'
2023-11-08 17:10:52.895 using GIT_ASKPASS to set credentials gitlab credentials
2023-11-08 17:10:52.895 > git ls-remote -h -t -- https://git.intranet.gencat.cat/0192/sic-resources/shared-libs/pipeline.git # timeout=10
2023-11-08 17:10:53.416 Found match: refs/tags/2.2.0 revision ad461e85812e5e0c9782fbd427caa131b03ec4f4
2023-11-08 17:10:53.416 Resolving tag commit... (remote references may be a lightweight tag or an annotated tag)
2023-11-08 17:10:53.416 > git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-a29881f021cf85717fd4a87425ffafd9/.git # timeout=10
2023-11-08 17:10:53.428 Setting origin to https://git.intranet.gencat.cat/0192/sic-resources/shared-libs/pipeline.git
2023-11-08 17:10:53.428 > git config remote.origin.url https://git.intranet.gencat.cat/0192/sic-resources/shared-libs/pipeline.git # timeout=10
2023-11-08 17:10:53.446 Fetching origin...
2023-11-08 17:10:53.446 Fetching upstream changes from origin
2023-11-08 17:10:53.446 > git --version # timeout=10
2023-11-08 17:10:53.460 > git --version # 'git version 2.38.3'
2023-11-08 17:10:53.460 > git config --get remote.origin.url # timeout=10
2023-11-08 17:10:53.474 using GIT_ASKPASS to set credentials gitlab credentials
2023-11-08 17:10:53.475 > git fetch --tags --force --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
2023-11-08 17:10:53.475 > git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-a29881f021cf85717fd4a87425ffafd9/.git # timeout=10
```

4 Enllaços d'interès

A continuació, es proporciona una llista d'enllaços a documents d'interès que puguin servir com a suport:

- Document de preguntes freqüents amb relació al desplegament:
<https://canigo.ctti.gencat.cat/plataformes/sic/faq/>
- Documentació de suport per al desplegament a Gitlab:
[Custòdia de codi font \(gencat.cat\)](#)

- Canals de suport de CTTI per comunicar qualsevol incidència:
[Canals de suport \(gencat.cat\)](#)
- Funcionament dels pipelins:
[Autoservei de pipelines \(gencat.cat\)](#)
- Construcció del fitxer ACA:
[Com construir el fitxer ACA \(gencat.cat\)](#)