SISTEMA DE APOYO A LA OPERACIÓN - SIAPO

Manual de Instalación y Actualización para Ambiente en Producción

TRANSMILENIO S.A.

Versión 16

Mayo 2025

Information Trade Management Consulting de Colombia Ltda.
Sede Bogotá - Colombia:
Autopista Bogotá - Medellín Km 3,5 Vía Siberia
Municipio de Cota Centro Empresarial Metropolitano
Centro de Negocios Ofc. B - 19 - Costado Norte. Colombia.
PBX - FAX: (57 1) 841 55 51 Teléfono: (57 1) 821 91 72 / 76
Sede Miami - USA:
7015 N.W. 173 Dr Of 209 33015 Hialeah Florida Estados Unidos
PBX: (305) 600 - 0930
www.itmcolombia.com info@itmcolombia.com

Contenido

I. 3

Base de datos 3

Creación base de datos y usuario 9

Cluster Kubernetes 11

Cloud Shell 16

Base de datos y cluster kubernetes desde Cloud Shell 16

Crear bucket 17

Reservar dirección IP 18

Configurar datos de conexión de la base de datos y vpn 18

Instalación elasticsearch 19

Jenkins 20

Instalar plugin Gitlab 23

Conexión con los repositorios en Gitlab 27

Crear Jobs 30

Despliegue automático 33

Crear token 34

Ingreso a la aplicación 38

II. 44

III. 46

Devolver a una versión anterior 41

RollBack de una Instalación 42

Base de datos 42

Almacenamiento 43

Google Cloud Console 44

I.Despliegue Interventoría producción

Ingresar a la plataforma de google cloud (https://console.cloud.google.com) y seleccionar o crear el proyecto en el que se desean crear los recursos desde el menú superior:

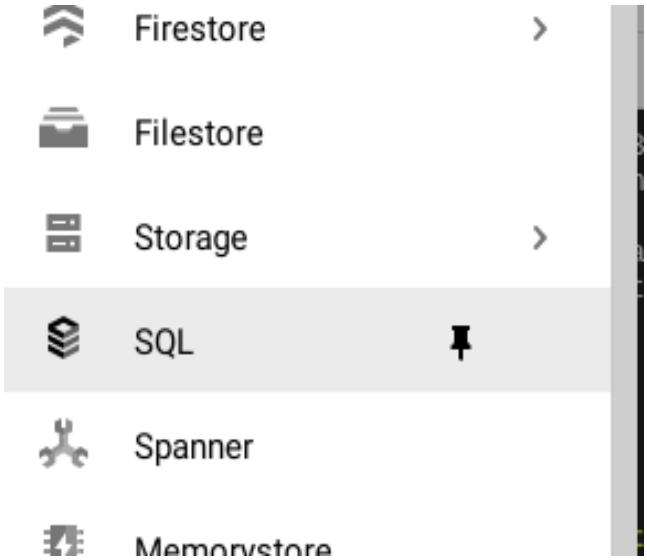


Base de datos

Para el proyecto se requiere una base de datos PostgresSQL 9.6 con un usuario y dirección IP privada. A continuación, se describen los pasos para crear una:

Crear la instancia de base de datos PostgreSQL

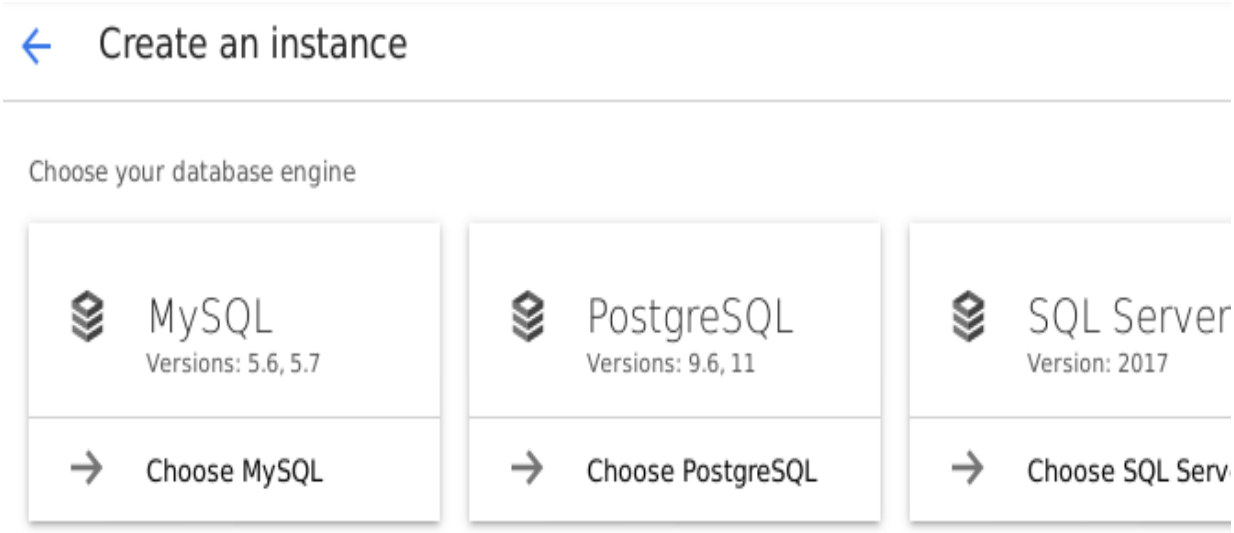
Ir a la página de Cloud SQL desde el menú que se encuentra en la parte izquierda de la plataforma de google cloud.



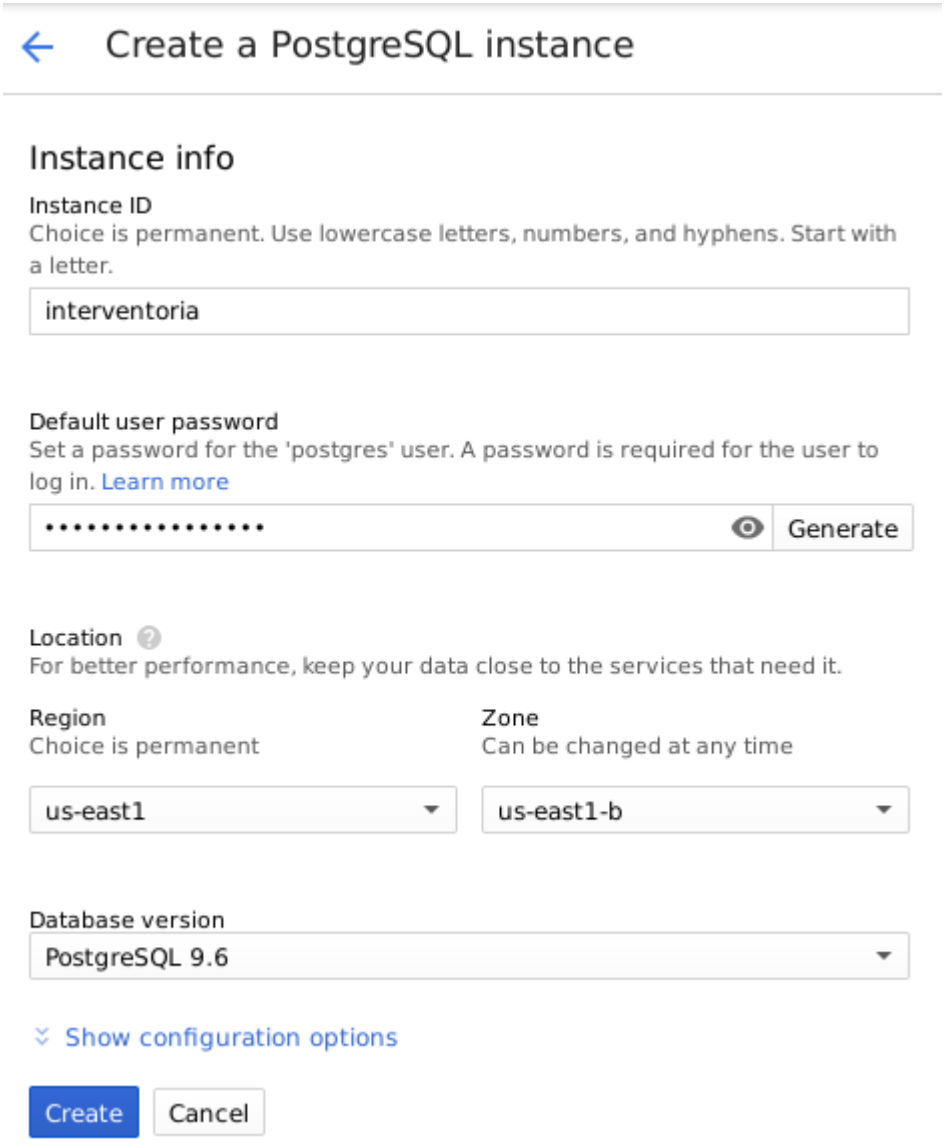
Seleccionar “*Create instance*”



y seleccionar crear instancia PostgreSQL



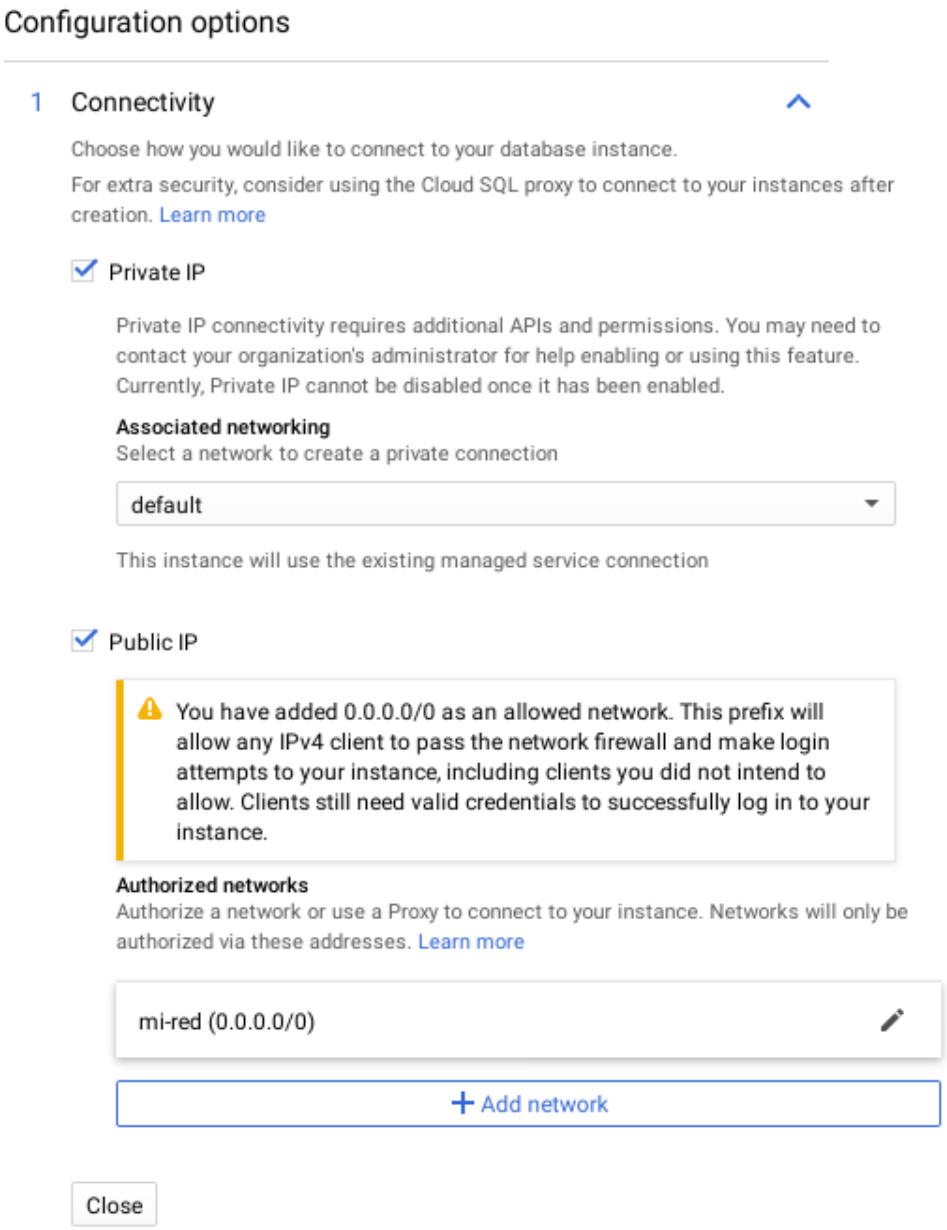
Se debe escribir el ID de la instancia, dar una contraseña al usuario Postgres, seleccionar región y zona deseada y la base de datos versión 9.6



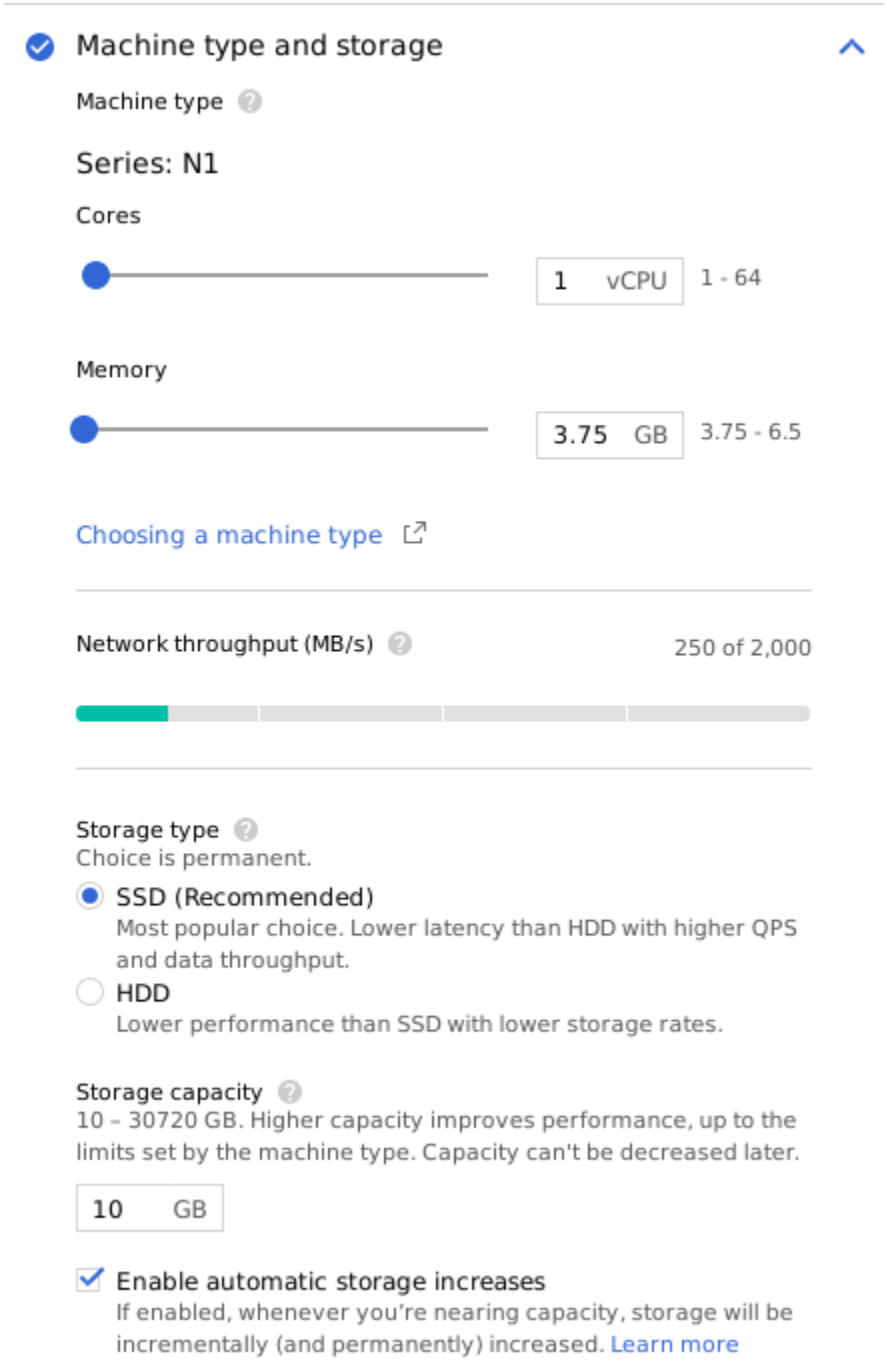
Expandir opciones e indicar conectividad, para este caso se requiere IP privada.

Si se desea se puede habilitar la IP pública para administración. En authorized networks se debe ingresar la dirección de red desde donde nos conectamos. Para este caso se usó 0.0.0.0/0 que permite conectar desde cualquier dirección IP lo que no es recomendable por seguridad.

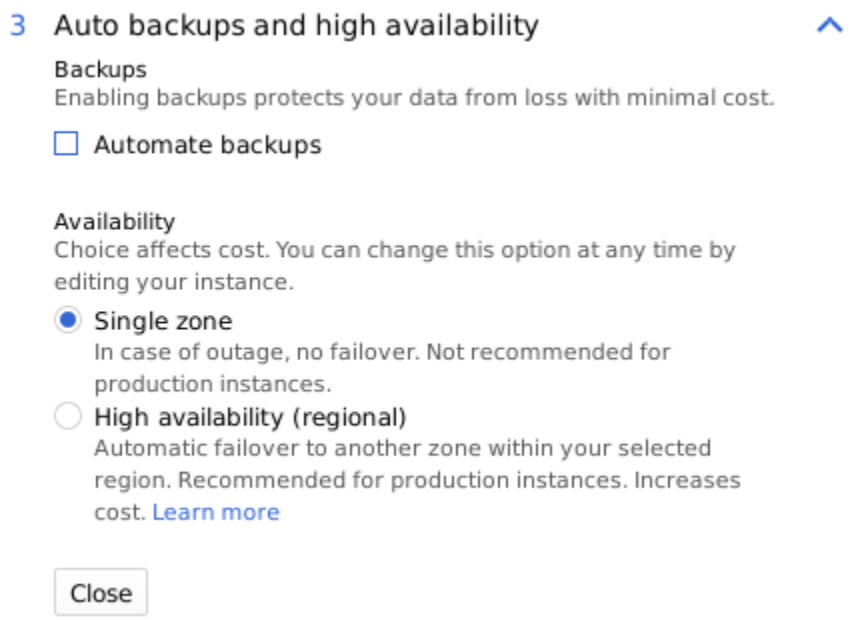
Seleccionar red asociada si se utiliza una diferente a la por defecto.



Seleccionar el tipo de máquina y almacenamiento, marcar la opción de incremento automático

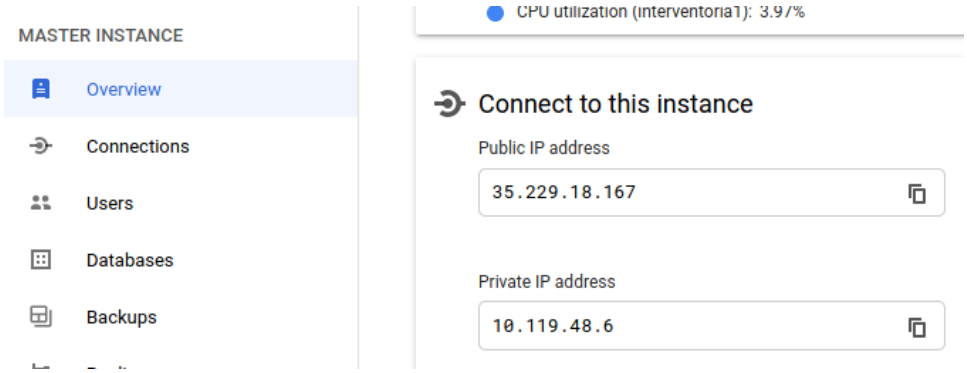


Si se desean hacer Backups automáticos o alta disponibilidad habilitar las opciones.



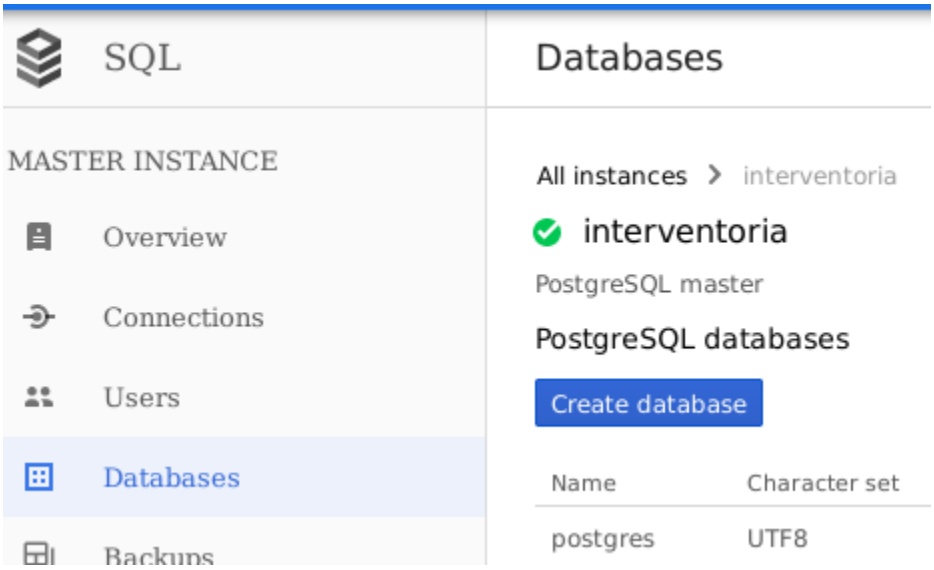
Seleccionar Crear

La dirección IP privada asignada se puede observar en el menú overview, se utilizará más adelante haciendo referencia como <IP_PRIVADA_BD>:

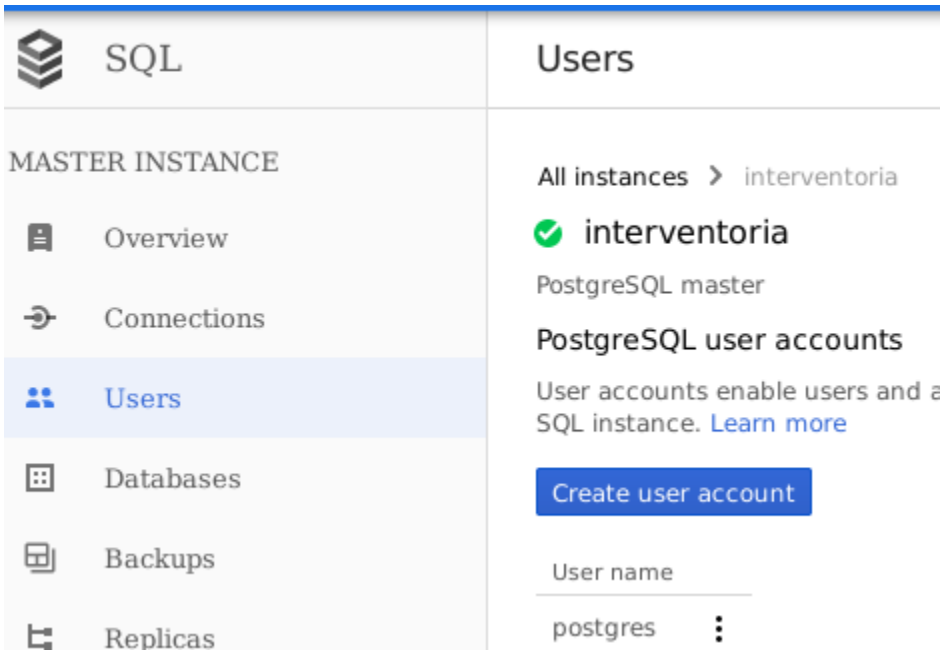


Creación base de datos y usuario

Sobre la instancia creada dar clic en crear base de datos e indicar el nombre, para este caso interventoria.

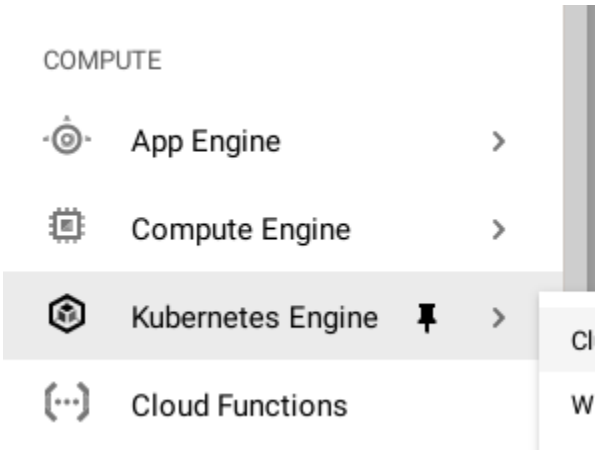


Si se desea un usuario diferente a postgres dar clic en crear cuenta de usuario, para el despliegue se utilizó tmuser

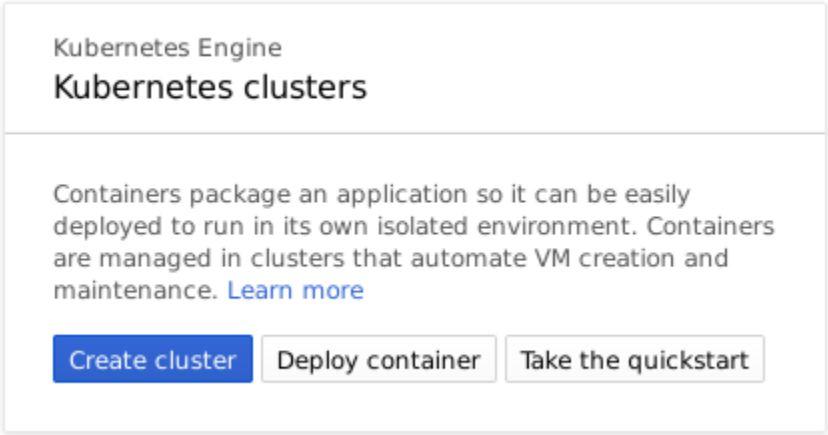


Cluster Kubernetes

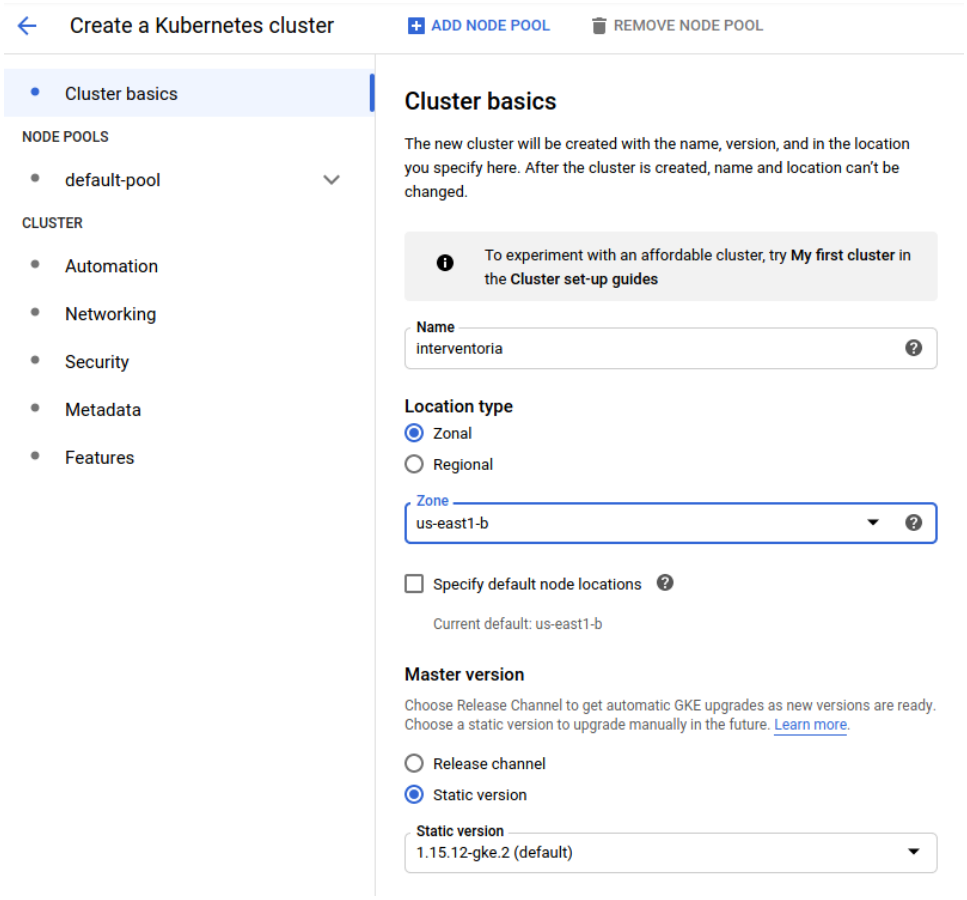
Desde el menú de Google Cloud dar clic en Kubernetes Engine



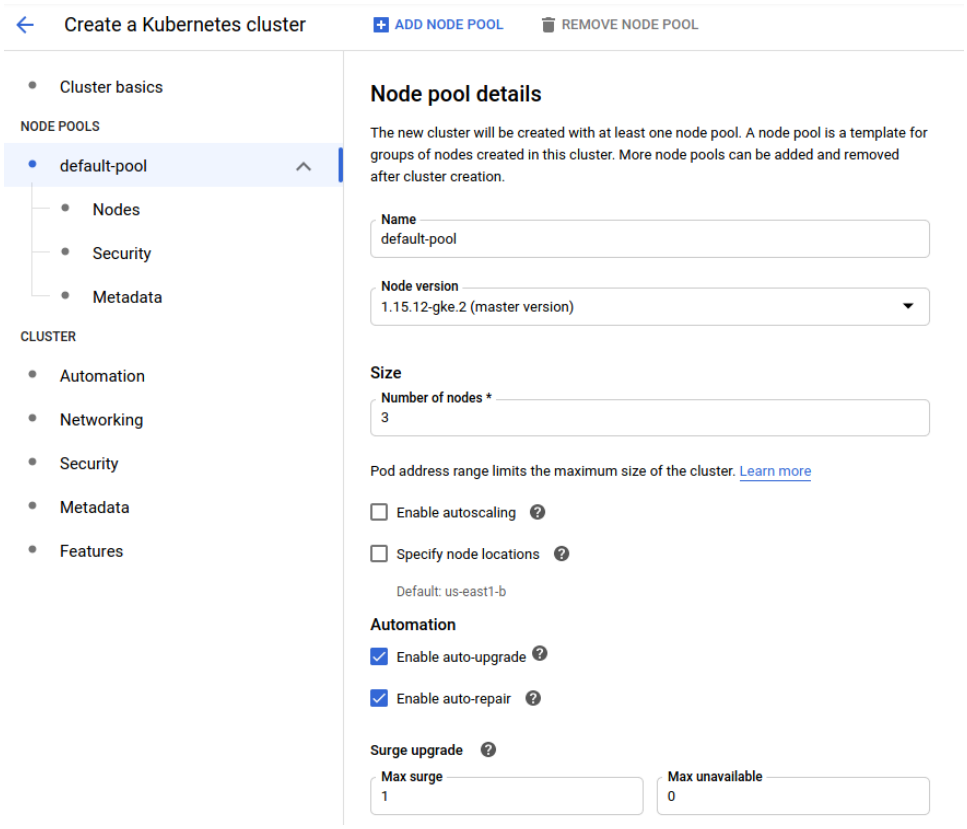
Seleccionar crear cluster



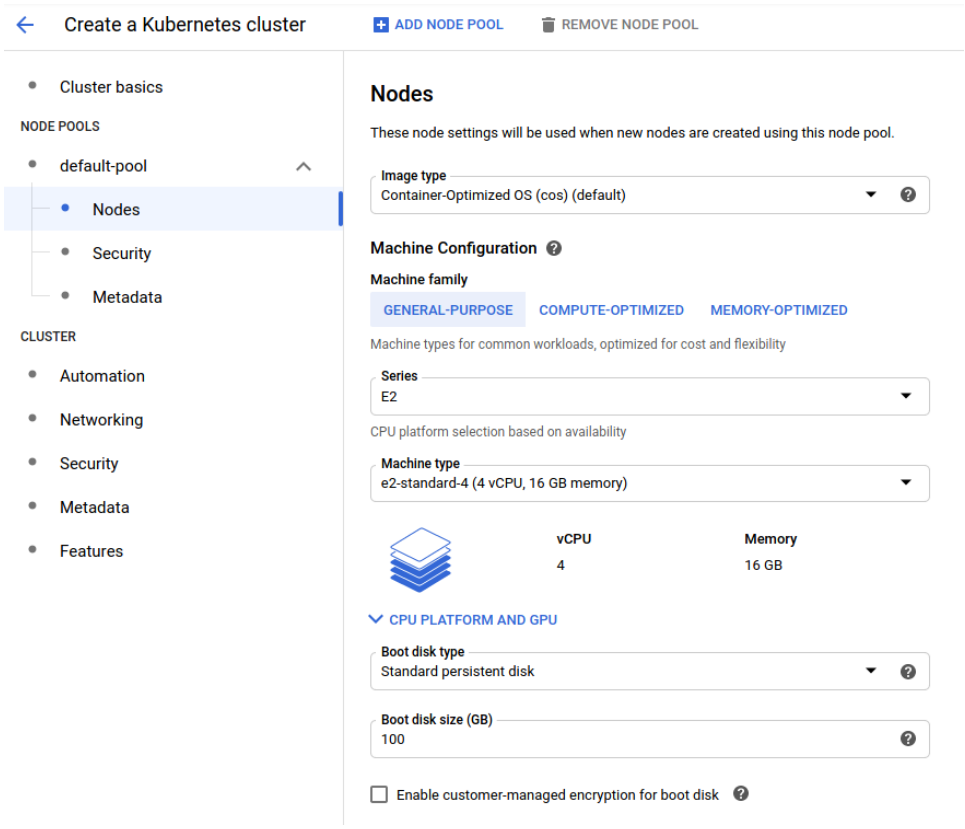
Escribir el nombre del cluster y seleccionar la zona (debe coincidir con la zona de la base de datos para poder realizar la conexión con la dirección IP privada)



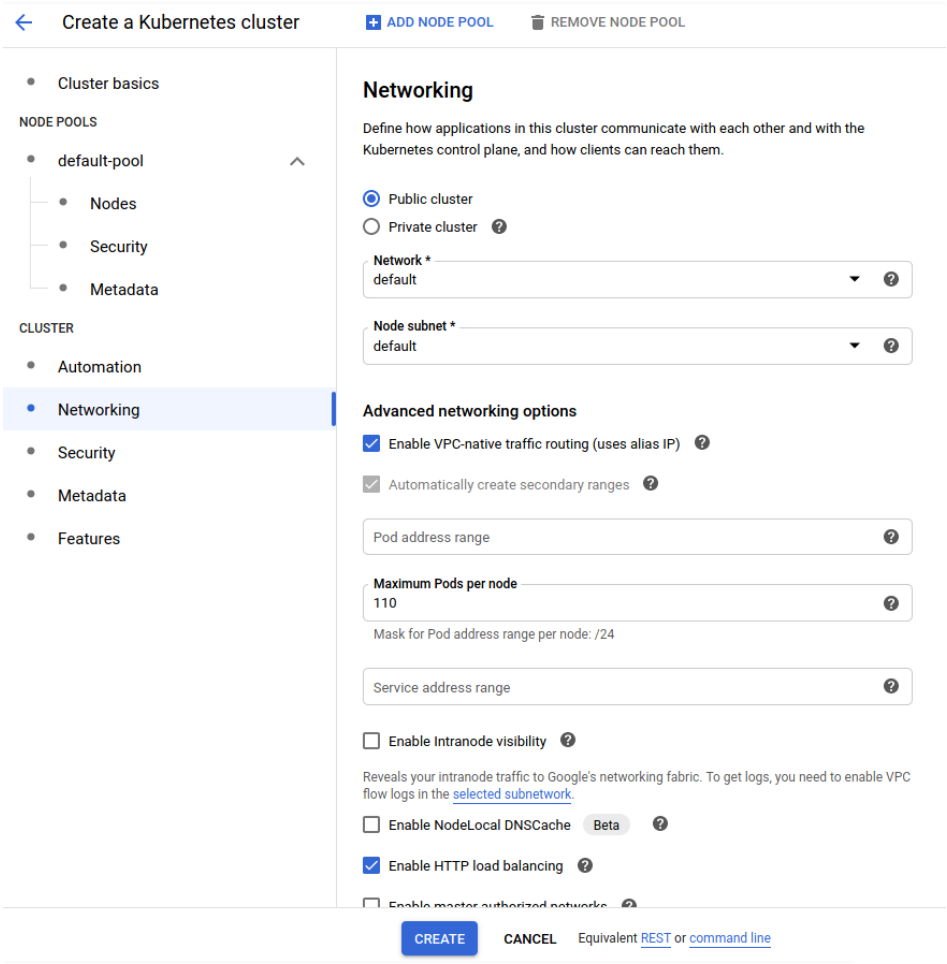
En default-pool, seleccionar el número de nodos (3)



En el menú Nodes, seleccionar el tipo de máquina, e2-standard-4



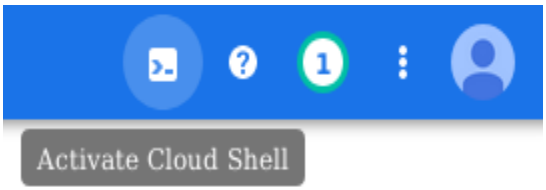
En Networking, seleccionar Enable VPC-native para permitir la comunicación con la base de datos por medio de IP privada. Seleccionar red de ser una diferente, en este caso es la misma de la base de datos por defecto.



Dar clic en crear.

Cloud Shell

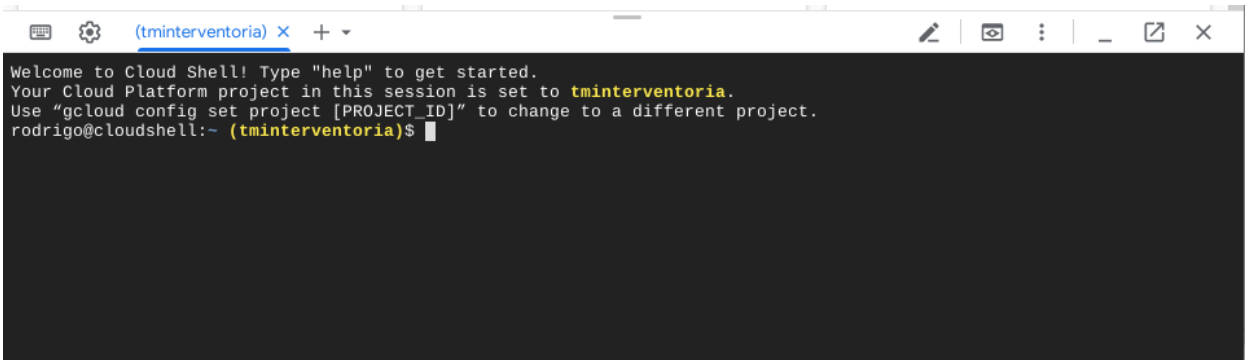
Activar Cloud Shell dando clic en el botón en la parte superior derecha



Una vez activada aparece en la parte inferior, donde pueden ser ingresados los comandos.

Se debe verificar se encuentre activo el proyecto (tminterventoria). Para activar el proyecto tminterventoria ejecutar el comando indicando el id del proyecto:

```
gcloud config set project tminterventoria
```



Base de datos y cluster kubernetes desde Cloud Shell

A continuación, se describe los pasos para crear la base de datos y el cluster por medio de comandos desde Cloud Shell, omitir si ya se encuentran creados.

Para crear la instancia de base de datos desde Cloud Shell, ingresar el siguiente comando:

```
gcloud beta sql instances create interventoria --network default --database-version=POSTGRES_9_6 --cpu 1 --memory 3840MiB --no-assign-ip --storage-auto-increase --zone us-east1-b
```

Crear base de datos

```
gcloud sql databases create --instance interventoria interventoria
```

Crear usuario

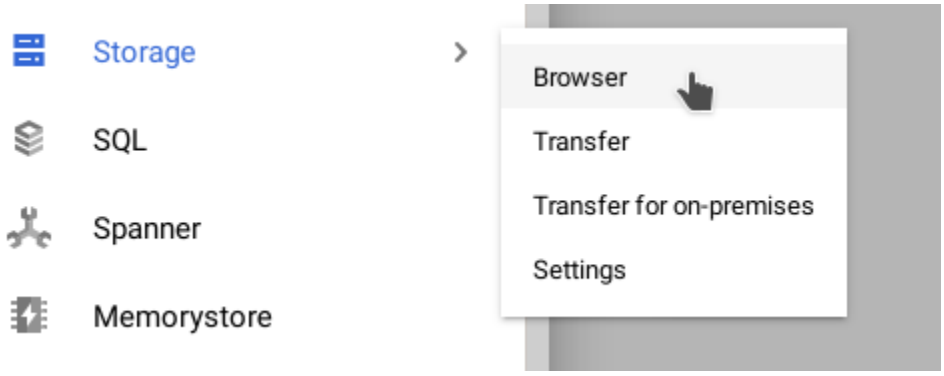
```
gcloud sql users create tmuser --host=% --instance interventoria --password oophim7WuNai
```

Cluster kubernetes

```
gcloud container clusters create interventoria --num-nodes 3 --zone us-east1-b --machine-type=e2-standard-4 --enable-ip-alias
```

Crear bucket

Ingresar en la plataforma de google cloud en el munu Storage - Browser:



Dar clic en crear bucket:



Escribir el nombre de bucket, tener presente el nombre ya que será usado más adelante en Jenkins para configurar el módulo.

Dar clic en crear.

Reservar dirección IP

Desde Cloud Shell se reserva una dirección IP que es la que permitirá el ingreso desde internet a la aplicación, ejecutar:

```
gcloud compute addresses create interventoria-ip --global
```

Para ver la dirección IP reservada ejecutar:

```
gcloud compute addresses describe interventoria-ip --global
```

en el campo address, o desde la plataforma de google cloud en el menú VPN network > External IP address con el nombre de interventoria-ip.

Configurar datos de conexión de la base de datos y vpn

Desde Cloud Shell ejecutar el siguiente comando para conectar con el cluster creado:

```
gcloud container clusters get-credentials interventoria --zone us-east1-b
```

Ejecutar los siguientes comandos, reemplazando los campos <PASS_BD> por la contraseña de la base de datos y <IP_PRIVADA_BD> por la dirección privada asignada a la base de datos.

```
kubectl create secret generic db-secret --from-literal=username=tmuser
--from-literal=password=<PASS_BD>

kubectl create configmap db-config --from-literal=host=<IP_PRIVADA_BD>
--from-literal=port=5432 --from-literal=name=interventoria
```

Para la conexión a la vpn:

```
kubectl create secret generic vpn-secret --from-
literal=user=<USER> --from-literal=pass=<PASS>

kubectl create configmap vpn-config --from-literal=host=<HOST> --
from-literal=port=<PORT> --from-literal=digest=
275dc15c38599c5a5a571eac12abb01f3795411a14c7e28d4ebe1caaaf173f50
```

Instalación elasticsearch

Para la instalación de elasticsearch se realiza con helm, desde Cloud Shell ejecutar:

Añadir repositorio de elasticsearch

```
helm repo add elastic https://helm.elastic.co
helm repo update
```

Instalar elasticsearch

Se requiere el archivo elasticsearch-values.yaml que se encuentra en el repositorio, al ejecutar git clone se solicitará un usuario y contraseña con acceso al repositorio:

```
git clone https://gitlab.transmilenio.gov.co/rhernandez/k8s-
interventoria-prod.git

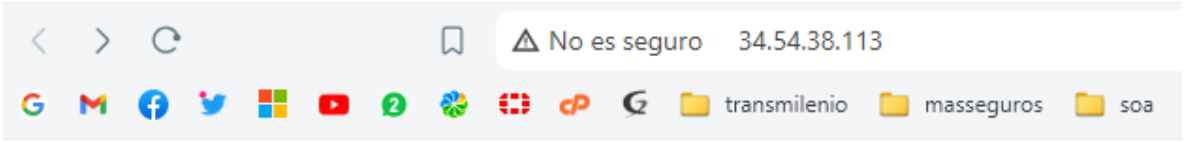
helm install elasticsearch elastic/elasticsearch -f k8s-
interventoria-prod/elasticsearch-values.yaml --version 6.8.9
```

Esperar que termine la instalación de elasticsearch y restaurar la base de datos:

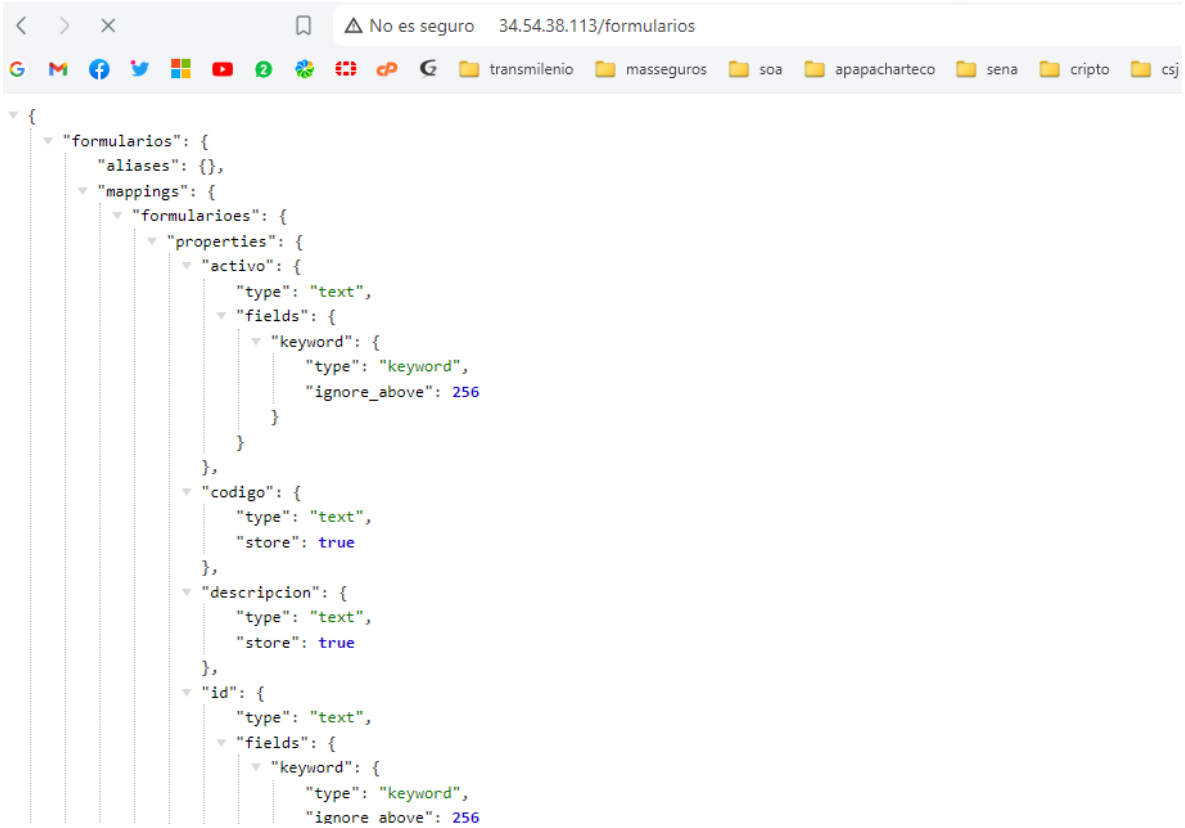
```
cat k8s-interventoria-prod/elastic-200824.tar.gz | kubectl exec -
i transmilenio-cluster-master-0 -- tar xzf -

kubectl exec -i transmilenio-cluster-master-0 -- curl -s -X POST
http://localhost:9200/_snapshot/backup/200824/_restore?pretty
```

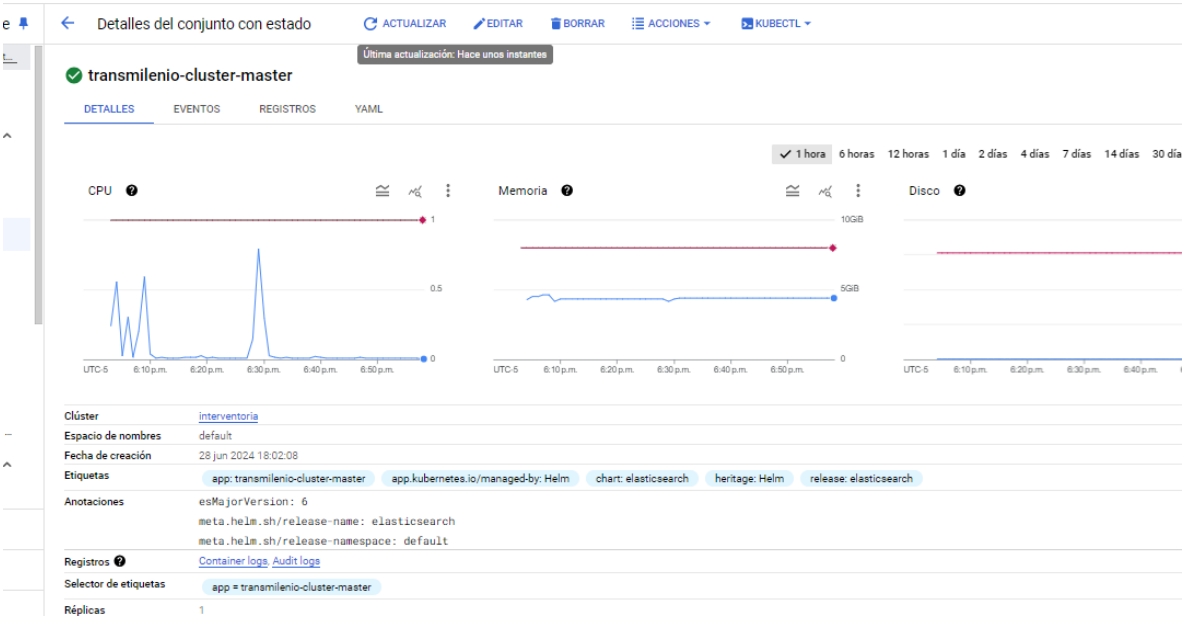
Migración elasticsearch




```
{
  "name": "transmilenio-cluster-master-0",
  "cluster_name": "transmilenio-cluster",
  "cluster_uuid": "qZF5WV1cSMeYMRnFMkJifA",
  "version": {
    "number": "6.8.9",
    "build_flavor": "default",
    "build_type": "docker",
    "build_hash": "be2c7bf",
    "build_date": "2020-05-04T17:00:34.323820Z",
    "build_snapshot": false,
    "lucene_version": "7.7.3",
    "minimum_wire_compatibility_version": "5.6.0",
    "minimum_index_compatibility_version": "5.0.0"
  },
  "tagline": "You Know, for Search"
}
```




```
{
  "formularios": {
    "aliases": {},
    "mappings": {
      "formularioes": {
        "properties": {
          "activo": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "codigo": {
            "type": "text",
            "store": true
          },
          "descripcion": {
            "type": "text",
            "store": true
          },
          "id": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
```

Ubicar `image: docker.elastic.co/elasticsearch/elasticsearch:6.8.0`

```
279         - name: node.master
280           value: "true"
281         image: docker.elastic.co/elasticsearch/elasticsearch:6.8.0
282         imagePullPolicy: IfNotPresent
283         lifecycle:
284           postStart:
285             exec:
```

Y cambiar por `image: docker.elastic.co/elasticsearch/elasticsearch:7.17.22`

Y agregar

```
      - env:
        - name: discovery.type
          value: single-node
```

```
232                   values:
233                   - transmilenio-cluster-master
234               topologyKey: kubernetes.io/hostname
235       containers:
236       - env:
237         - name: discovery.type
238           value: single-node
239         image: docker.elastic.co/elasticsearch/elasticsearch:7.17.22
240         imagePullPolicy: IfNotPresent
241         lifecycle:
```

Guardar y Validar

Jenkins

La instalación de jenkins se realiza por medio del gestor de aplicaciones Helm. Desde Cloud Shell ejecutar:

```
kubectl create clusterrolebinding jenkins-deploy --clusterrole=cluster-admin --serviceaccount=default:jenkins

helm repo add jenkins https://charts.jenkins.io
helm repo update
```

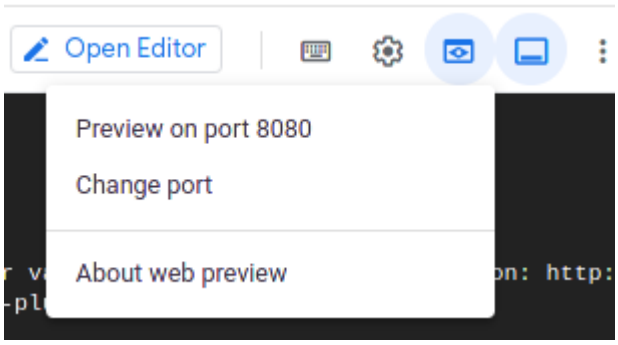
Para reinstalar se debe desinstalar la versión previa con: helm uninstall jenkins

Instalación sin publicar en internet

Para acceder de forma local con un reenvió de puerto con kubectl o desde cloud shell, ejecutar:

```
helm install jenkins jenkins/jenkins
kubectl --namespace default port-forward svc/jenkins 8080:8080
```

Desde cloud shell en el icono web preview dar clic en preview on port 8080 para ingresar a jenkins.



Instalación publicando en internet

Para instalar jenkins para acceder desde internet ejecutar:

```
helm install jenkins jenkins/jenkins --set
controller.serviceType=LoadBalancer,controller.servicePort=8090
```

Se debe esperar unos 5 minutos que es el tiempo que toma en instalar jenkins y el load balancer, Ingresar a jenkins con la IP obtenida con el siguiente comando en el puerto 8090

```
kubectl get svc --namespace default jenkins --template "{{ range (index
.status.loadBalancer.ingress 0) }}{{ . }}{{ end }}"
```

http://<IP_JENKINS>:8090/

Acceso

Obtener la contraseña del usuario admin, y guardarla en un lugar seguro

```
kubectl exec --namespace default -it svc/jenkins -c jenkins --
/bin/cat /run/secrets/chart-admin-password && echo
```



Welcome to Jenkins!

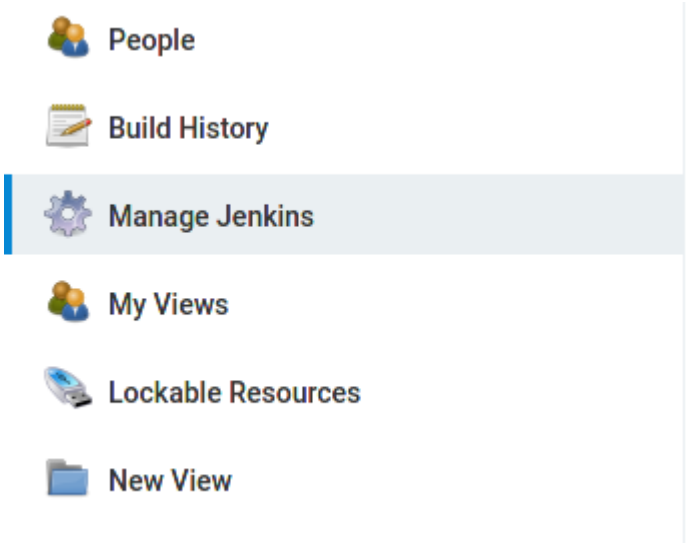
Username

Password

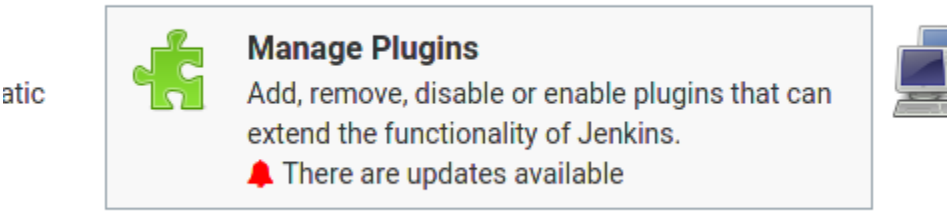
Sign in

Instalar plugin Gitlab

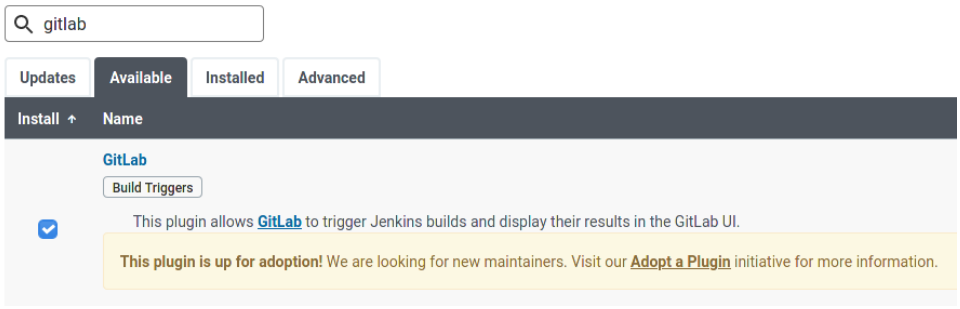
Ingresar a la administración de jenkins en el menú de la izquierda:



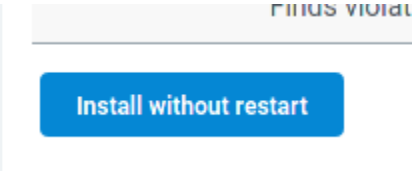
Dar clic en administración de plugins



Y en disponibles buscar gitlab, se debe seleccionar con la casilla de la izquierda:



Y clic en instalar



Para tener acceso a subir las imágenes a gcr desde cloud shell ejecutar los siguientes comandos:

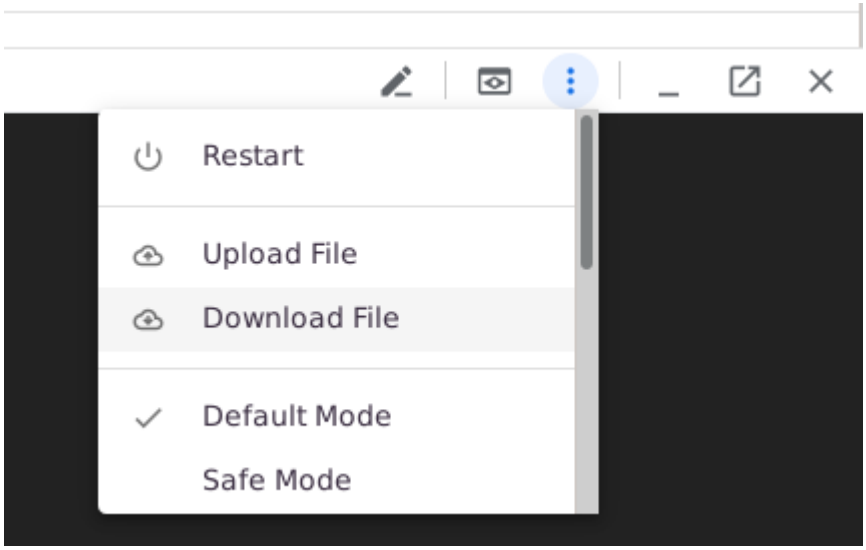
```
export PROJECT_ID=$(gcloud config get-value project)

gcloud iam service-accounts create jenkins-gcs \
--display-name "Jenkins Cloud Storage"

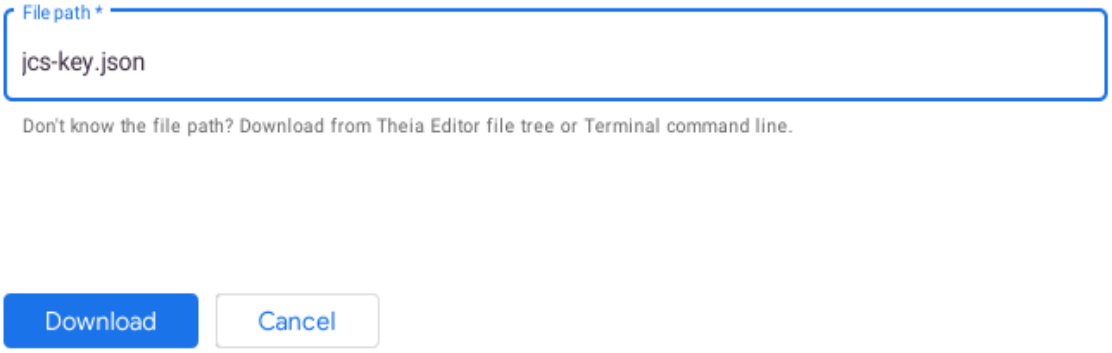
gcloud iam service-accounts keys create --iam-account \
jenkins-gcs@${PROJECT_ID}.iam.gserviceaccount.com jcs-key.json

gcloud projects add-iam-policy-binding --role roles/storage.admin
${PROJECT_ID} \
--member=serviceAccount:jenkins-
gcs@${PROJECT_ID}.iam.gserviceaccount.com
```

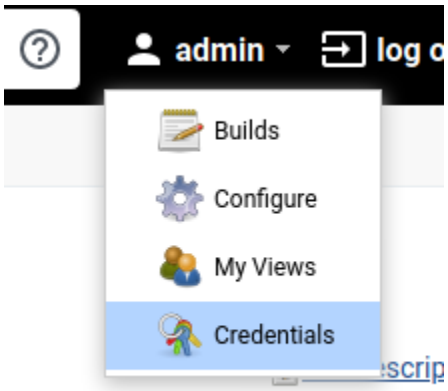
Descargar el archivo json generado, dando clic en la parte superior de cloud shell, Download File



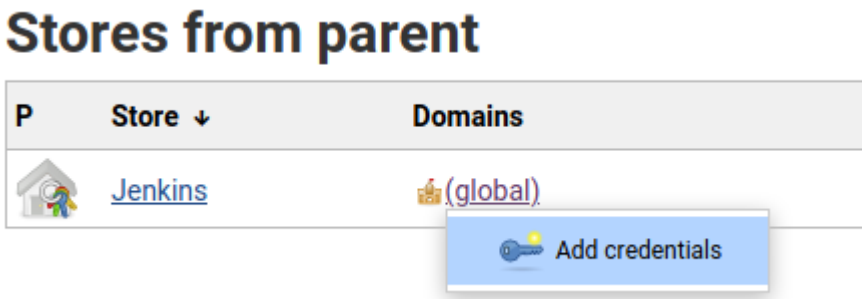
Ingresar el nombre del archivo **jcs-key.json** y clic en Download



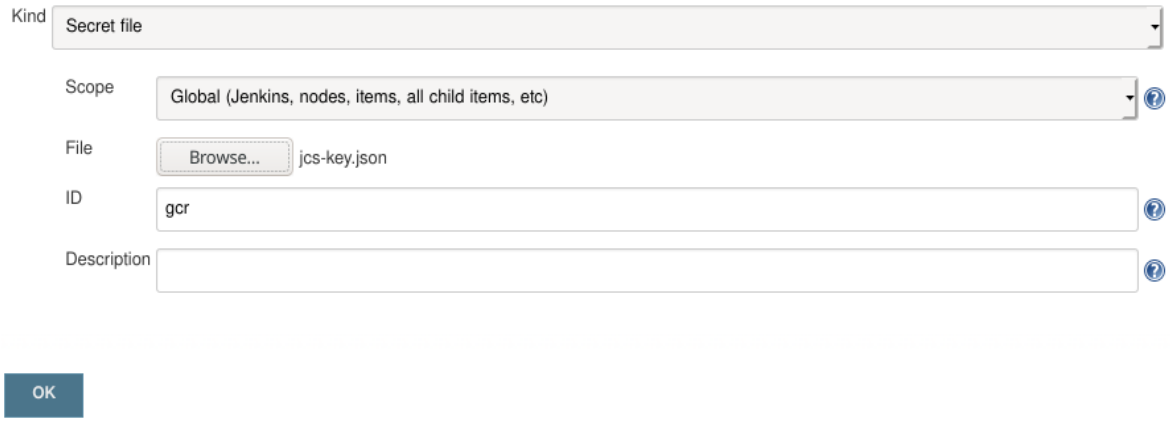
Ingresar a jenkins y en el menú admin en la parte superior derecha y seleccionar credentials



Seleccionar el enlace global add credentials



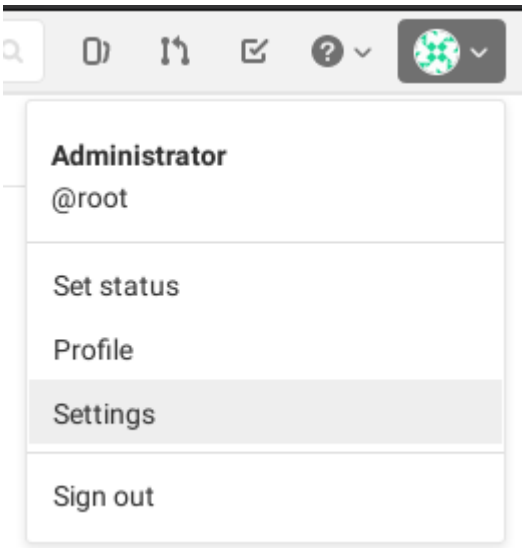
Seleccionar tipo Secret file, escribir ID: gcr y en File seleccionar el archivo json descargado



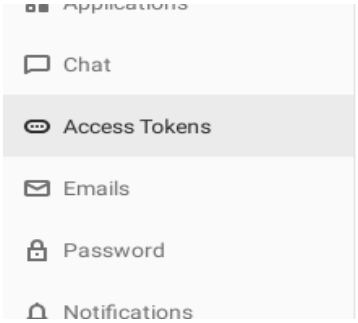
Dar clic en ok.

Conexión con los repositorios en Gitlab

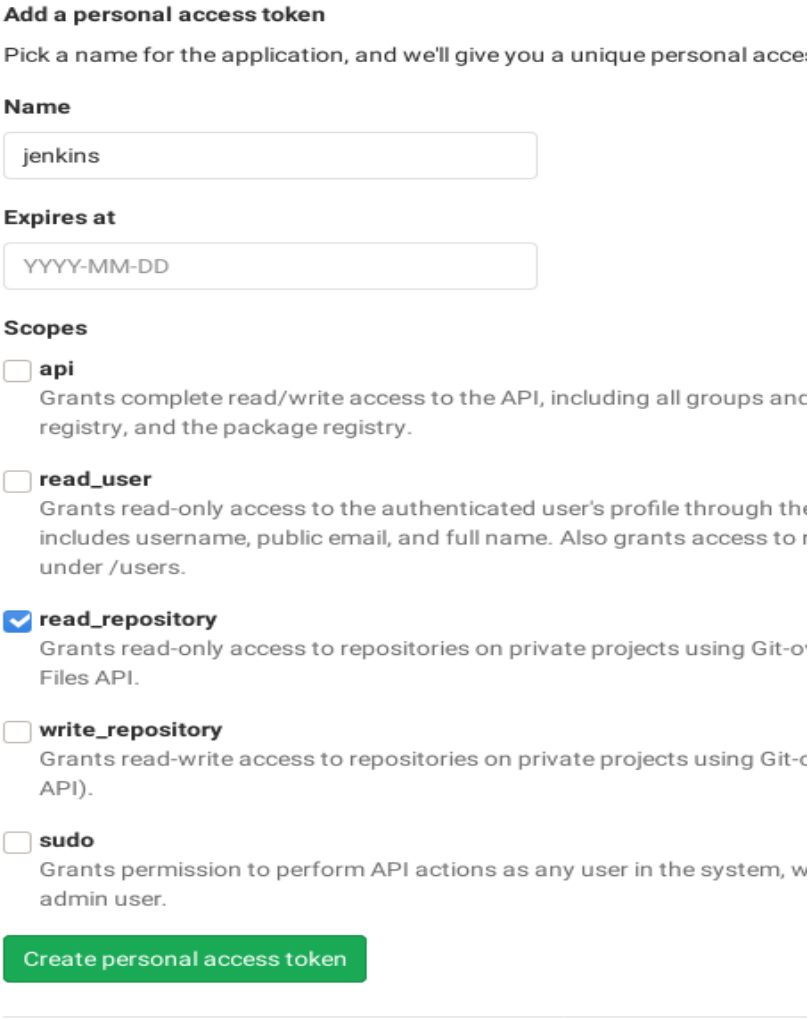
Para conectar jenkins con los repositorios de gitlab se debe crear un token, ingresar a gitlab con un usuario que tenga permisos en los repositorios del proyecto y desde el menú superior derecho seleccionar settings:



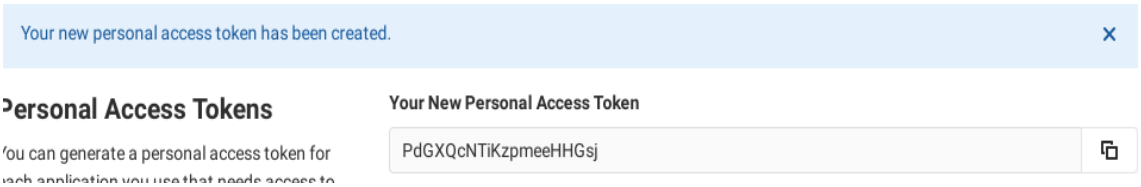
Dar clic en el menú izquierdo en Access Tokens:



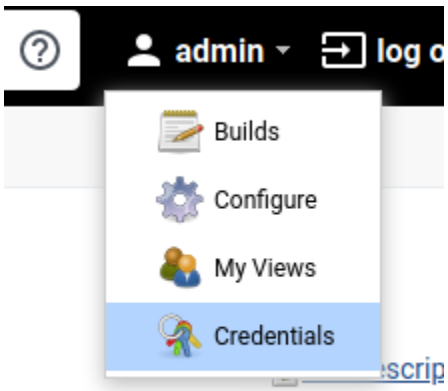
Completar los datos del formulario: nombre: jenkins, y marcar read_repository:



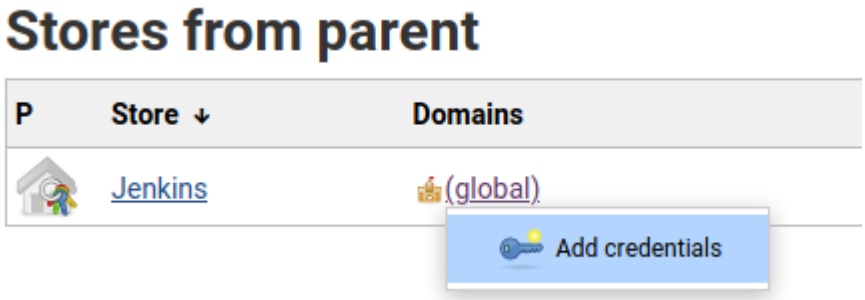
Copiar el nuevo token generado



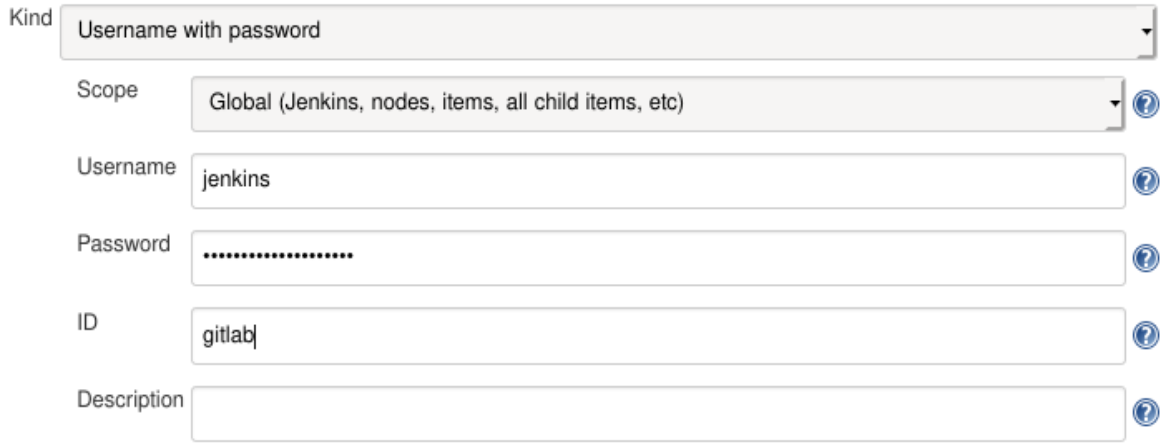
Ingresar a jenkins y en el menú admin en la parte superior derecha y seleccionar credentials



Dar clic el enlace global, add credentials:



Y completar el formulario con el token generado anteriormente, usuario jenkins y ID gitlab



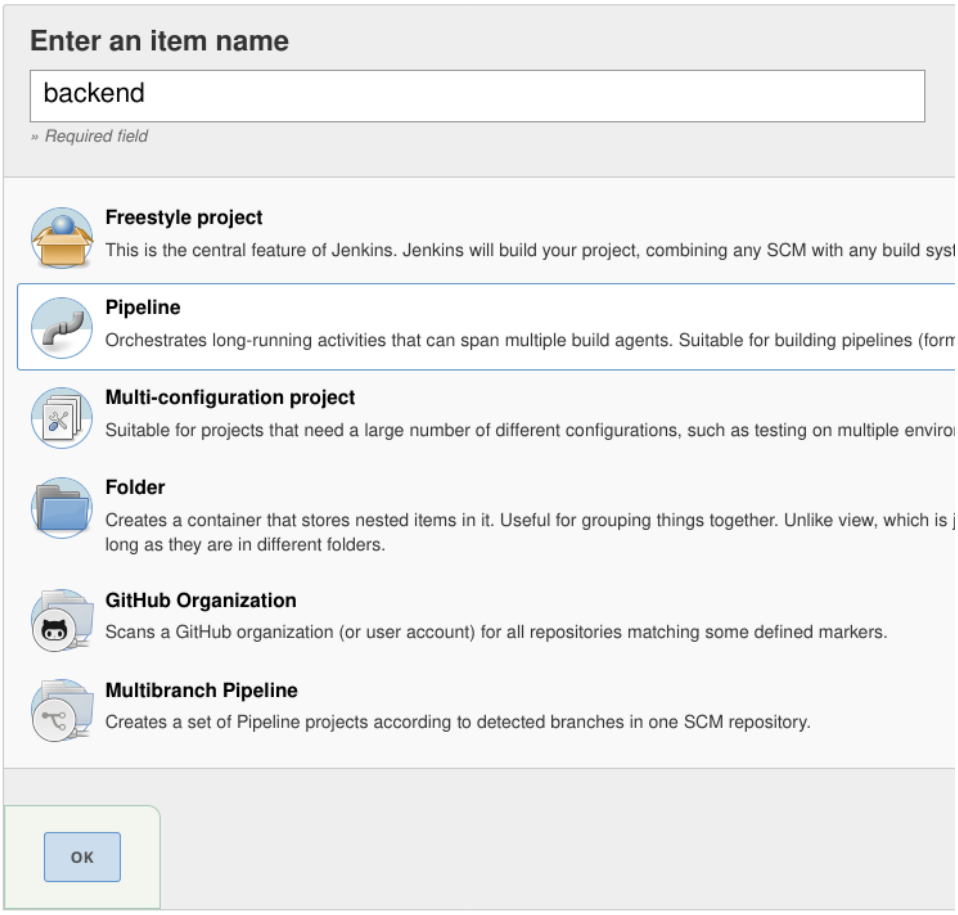
Crear Jobs

Los jobs se encargan de compilar el código, crear la imagen, subirla a gcr de google y desplegar en kubernetes.

Desde Jenkins dar clic en New Item



Se debe ingresar el nombre del job en este caso **backend**, seleccionar Pipeline y aceptar



En la parte inferior en el cuadro de texto de Pipeline pegar el contenido de backend.groovy y editar las primeras líneas y escribir el dominio que será utilizado por la aplicación y el nombre de la dirección IP reservada:

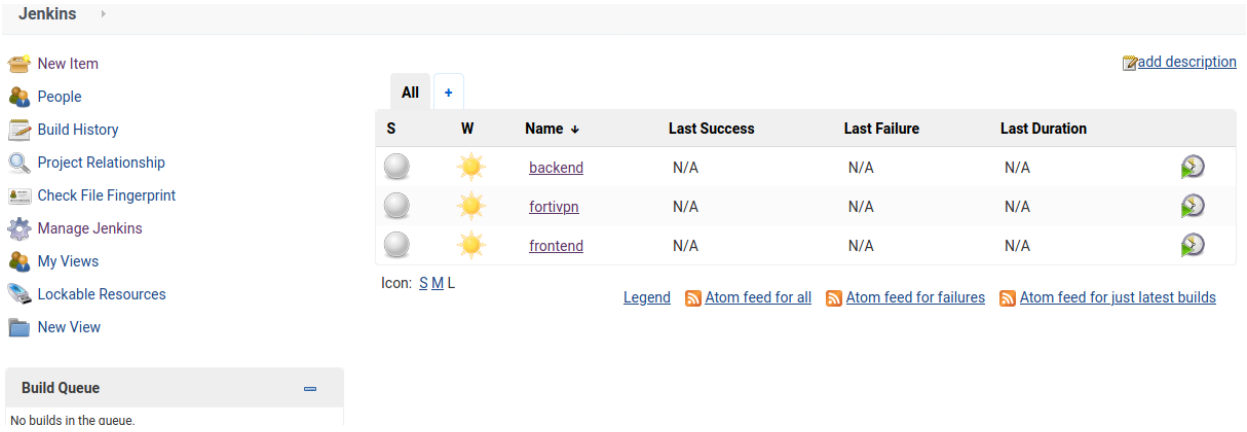
```
def HOST = "<DOMINIO>"
def IP_NAME = "<NOMBRE_IP_RESERVADA>"
```

dar clic en guardar. Los archivos se encuentran en el repositorio de gitlab https://gitlab.transmilenio.gov.co/rhernandez/k8s-interventoria-prod



El procedimiento de debe repetir creando **frontend** con frontend.groovy.

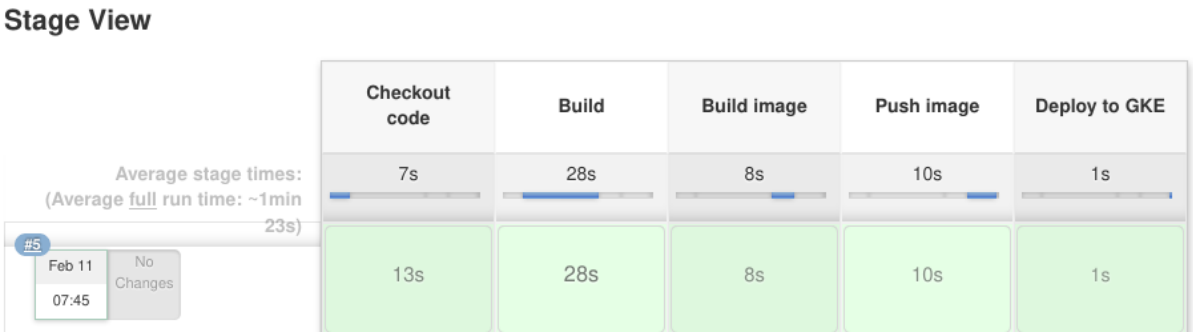
Al finalizar se ven los jobs de la siguiente forma:



Para iniciar el despliegue de un módulo se debe dar clic en el icono de Build, ejecutar fortivpn, backend y frontend:



Al ingresar por el enlace nombre del módulo se observa el último estado de ejecución, si fue exitoso salen en verde los pasos, si ocurrió algún error en color rojo.

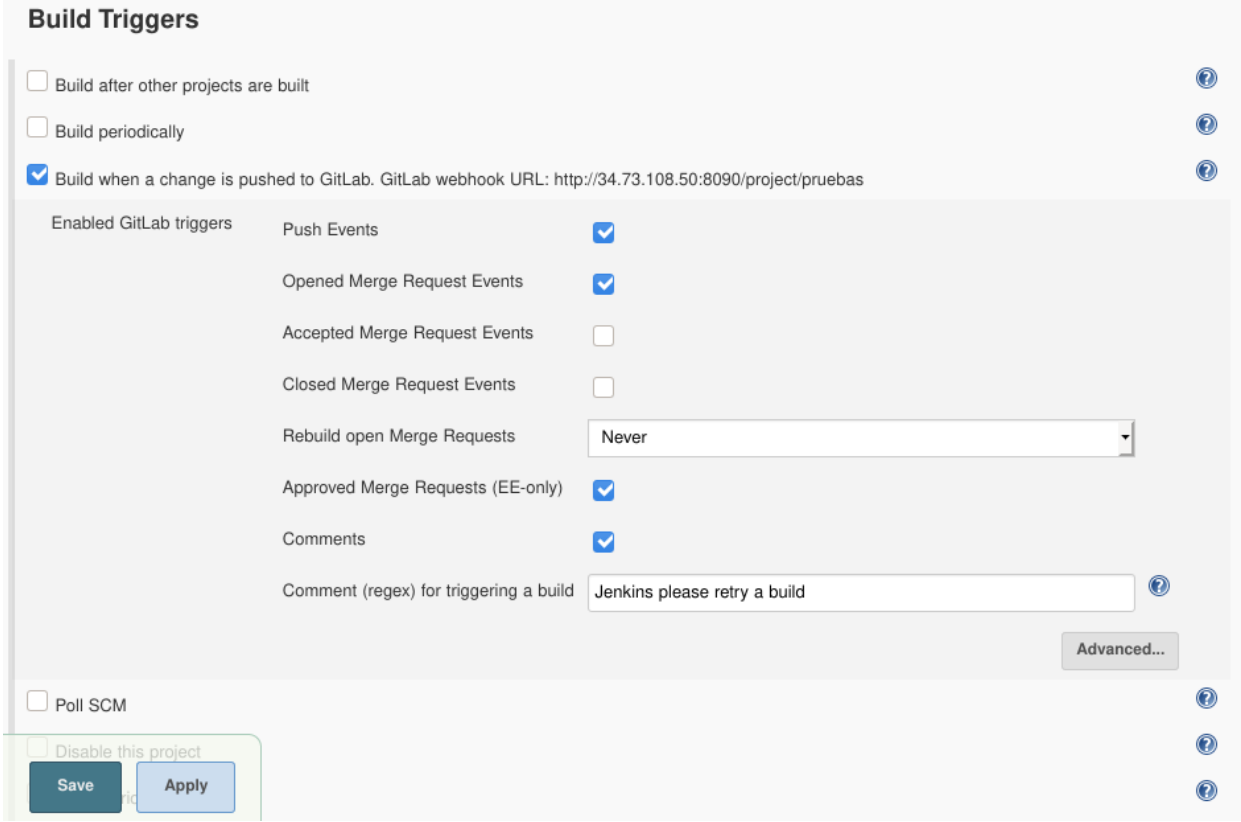


Despliegue automático

Cuando se activa el despliegue automático es ejecutado después que un commit es realizado sobre el repositorio del módulo. Si se desea activar el despliegue automático ingresar desde jenkins al módulo deseado y en dar clic en el menú a la izquierda en Configure:

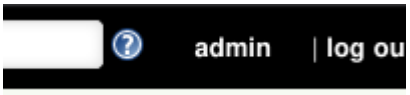


Ir a la sección Build Triggers y activar la casilla de **Build when a change is pushed to GitLab**, y dar clic en guardar

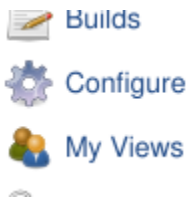


Crear token

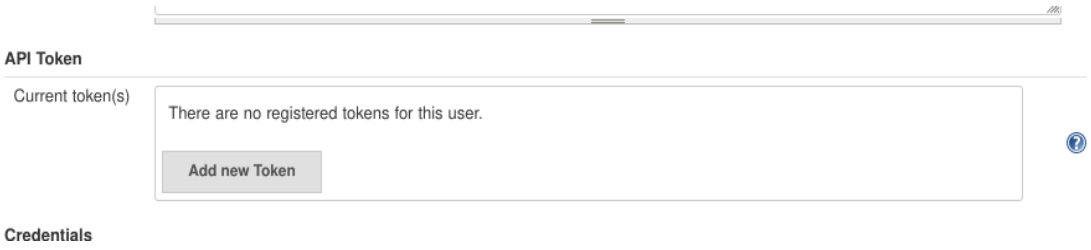
En el menú superior derecho, dar clic en el nombre de usuario (admin):



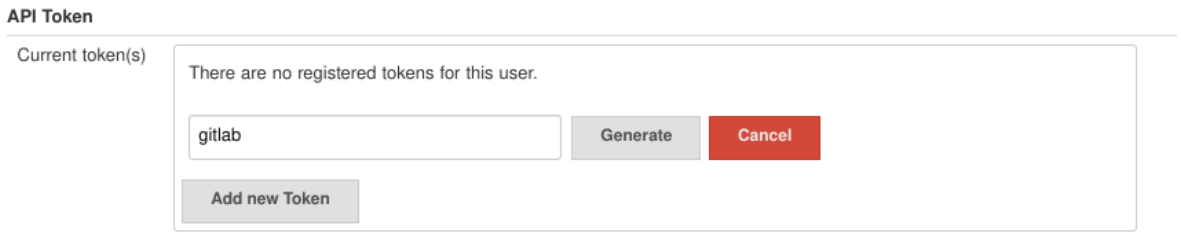
En el menú izquierdo seleccionar configure:



En la sección API Token dar clic en el boton Add new Token



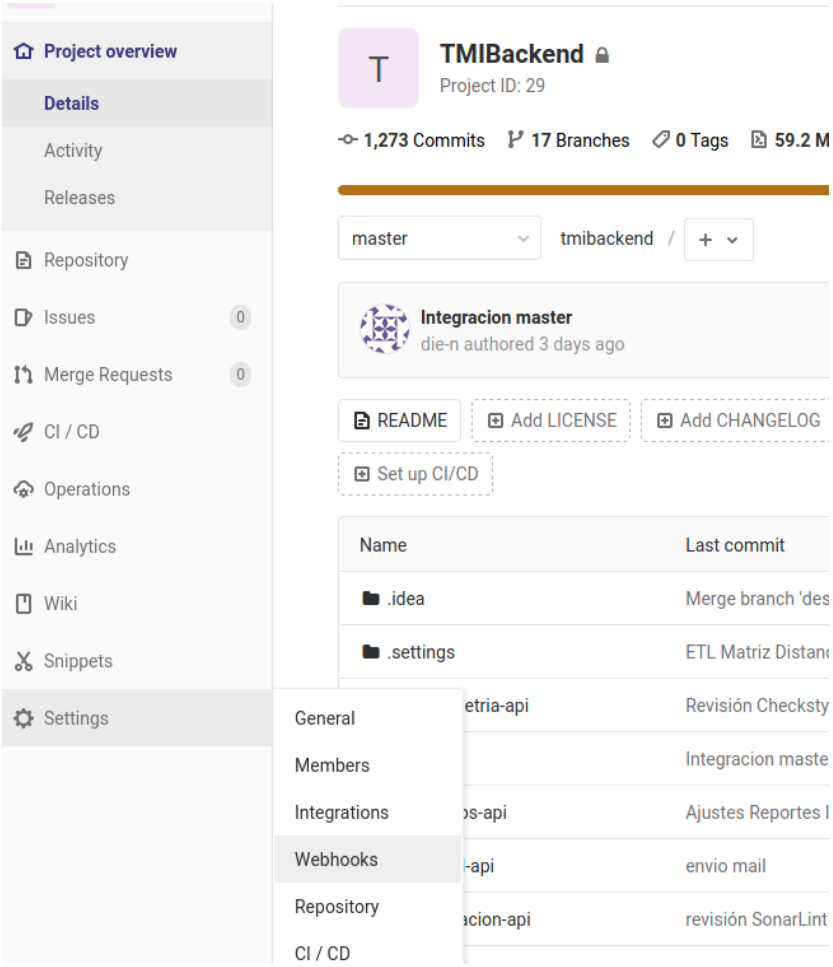
Escribir el nombre, ej: gitlab y dar clic en generar



Guardar el token que será usado en el siguiente paso



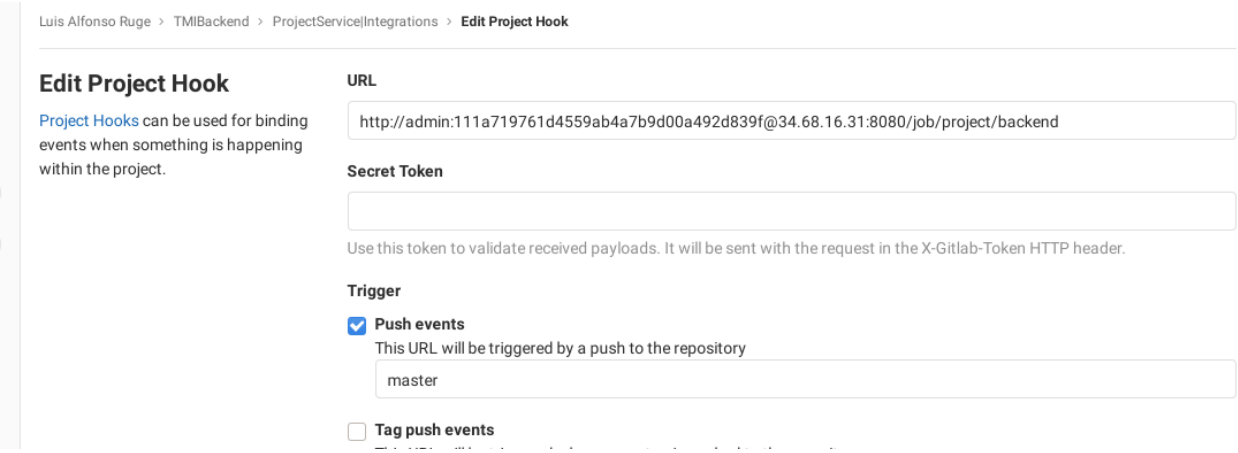
Ahora se debe activar desde gitlab el trigger, ingresar en gitlab al repositorio del módulo y en el menú Settings seleccionar Webhooks



En el campo URL colocar la ruta del job en jenkins con el usuario y token generados en jenkins ej:

http://admin:<TOKEN>@<IP_JENKINS>:8090/project/backend

Dónde admin es el usuario y token e ip_reservada son reemplazados por los valores generados.

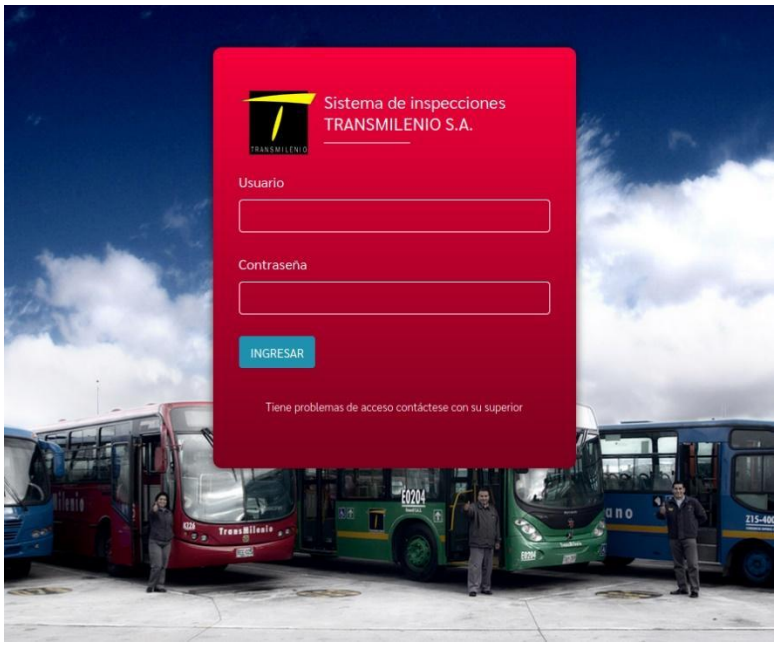


Activar Push events con el nombre de la rama del repositorio y dar clic en guardar.

Se deben repetir los pasos con cada repositorio de cada módulo.

Ingreso a la aplicación

Una vez desplegado todos los módulos se ingresa con el nombre de dominio



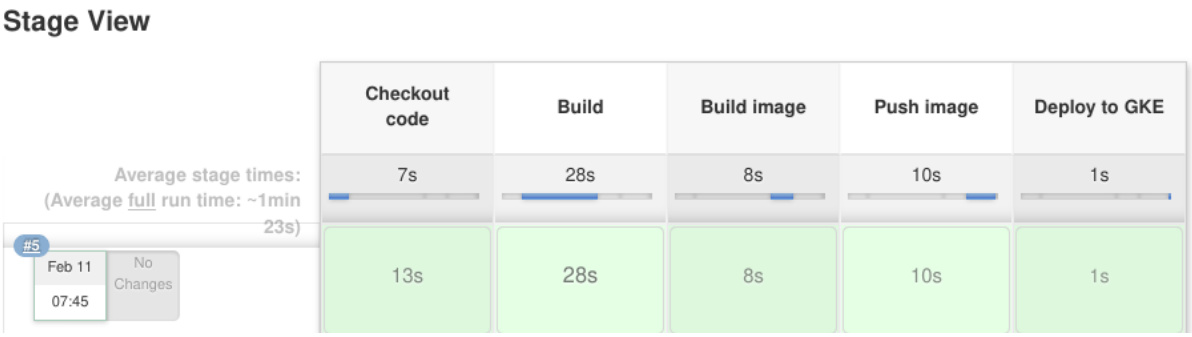
II. Actualización del Sistema en Ambiente de Producción

Prerrequisito: Garantizar que la última versión del código que contiene los paquetes de actualización, este en la rama master de los proyectos de GitLab.

Para iniciar la actualización del sistema se debe ingresar a Jenkins y dar clic en el icono de Build para ejecutar los Jobs backend y frontend:

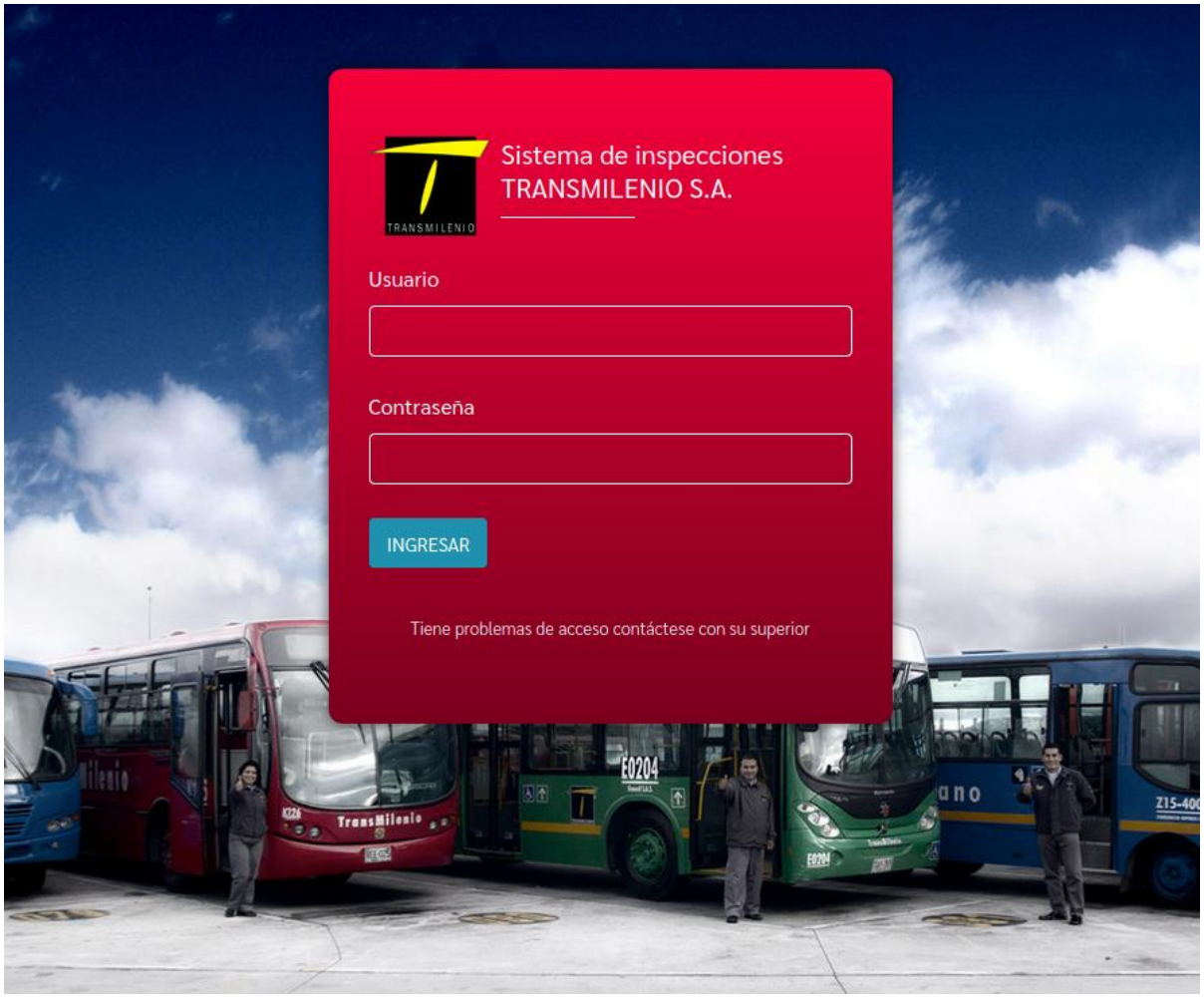


Al ingresar por el enlace nombre del job se observa el último estado de ejecución, si fue exitoso salen en verde los pasos, si ocurrió algún error en color rojo.



- **Ingreso a la aplicación**

Una vez desplegado todos los módulos se ingresa con la dirección IP reservada o si es asignado un nombre de dominio



Information Trade Management Consulting de Colombia Ltda.

Sede Bogotá - Colombia:
Autopista Bogotá - Medellín Km 3,5 Vía Siberia
Municipio de Cota Centro Empresarial Metropolitano
Centro de Negocios Ofc. B - 19 - Costado Norte. Colombia.
PBX - FAX: (57 1) 841 55 51 Teléfono: (57 1) 821 91 72 / 76

Sede Miami - USA:
7015 N.W. 173 Dr Of 209 33015 Hialeah Florida Estados Unidos
PBX: (305) 600 - 0930

www.itmcolombia.com info@itmcolombia.com

III. Procedimiento de Reversión de Cambios

Devolver a una versión anterior

Para deshacer una actualización de una aplicación ser realiza con *rollout undo*. El comando se debe ejecutar desde cloud shell indicando el nombre de la aplicación, ejemplo con seguridad-api:

```
kubectl rollout undo deployment/seguridad-api
```

Si el comando es ejecutado de forma correcta devuelve:

```
deployment.apps/seguridad-api rolled back
```

La aplicación seguridad-api es devuelta a la versión anterior.

Para ver el histórico de las actualizaciones utilizar *rollout history* y el nombre de la aplicación:

```
kubectl rollout history deployment/seguridad-api
```

El resultado debe ser la lista de revisiones:

```
deployment.apps/seguridad-api
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
3         <none>
4         <none>
5         <none>
6         <none>
7         <none>
8         <none>
```

Para devolver a una revisión especifica utilizar el comando con la opción *--to-revision* seguida del número:

```
kubectl rollout undo deployment/seguridad-api --to-revision 5
```

RollBack de una Instalación

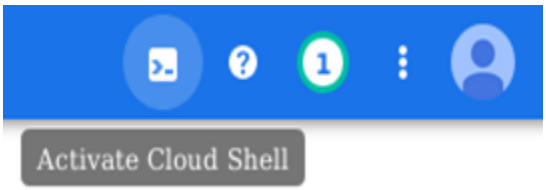
A continuación el procedimiento para Rollback, para usar en procesos de actualización del sistema de información que presenten algún tipo de evento no esperado.

Base de datos

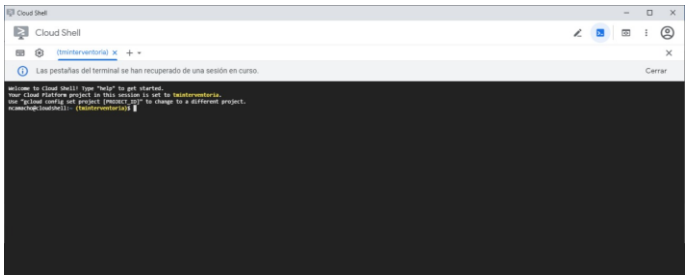
Para realizar la actualización del sistema de información, previamente al procedimiento se debe asegurar la base de datos de proyecto, a través de la ejecución de una copia de seguridad con la herramienta pg_dump. Desde la plataforma de Google Cloud se puede realizar en Cloud Shell.

Ingresar a la plataforma de Google Cloud (https://console.cloud.google.com/)

Activar Cloud Shell dando clic en el botón en la parte superior derecha



Una vez activada aparece en la parte inferior, donde pueden ser ingresados los comandos.



Ingresar el siguiente comando reemplazando *DIRECCION_IP* por la dirección IP de la base de datos, *NOMBRE_BASE_DE_DATOS* por el nombre de la base de datos del proyecto interventoría, una vez ingresado se solicita la contraseña del usuario PostgreSQL.

```
pg_dump -h <DIRECCION_IP> -U postgres -Fc -f db-interventoria.dump <NOMBRE_BASE_DE_DATOS>
```

Cuando finaliza la copia es generado el archivo *db-interventoria.dump*, para descargar el archivo ingresar el siguiente comando en cloud shell:

```
dl db-interventoria.dump
```

Una vez descargado se puede borrar la copia generada en cloud shell:

```
rm db-interventoria.dump
```

Almacenamiento

Para realizar la actualización del sistema de información, previamente al procedimiento se debe asegurar el contenedor de almacenamiento, con gsutil se puede descargar una copia completa del bucket en storage. Desde cloud shell ejecutar:

```
gsutil cp -r gs://tminterventoria .
zip -r tminterventoria.zip tminterventoria
dl tminterventoria.zip
```

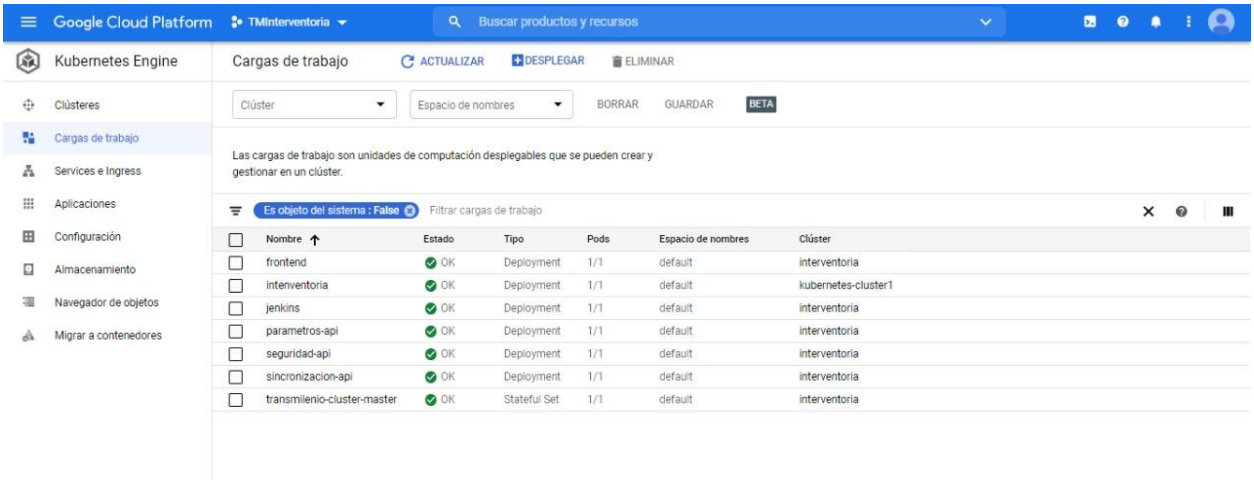
gsutil se encarga de copiar todo el contenido de bucket tminterventoria, zip comprime la carpeta en el archivo tminterventoria.zip y dl inicia la descarga del archivo.

Una vez descargado el archivo se pueden borrar en cloud shell:

```
rm –rf tminterventoria
rm tminterventoria.zip
```

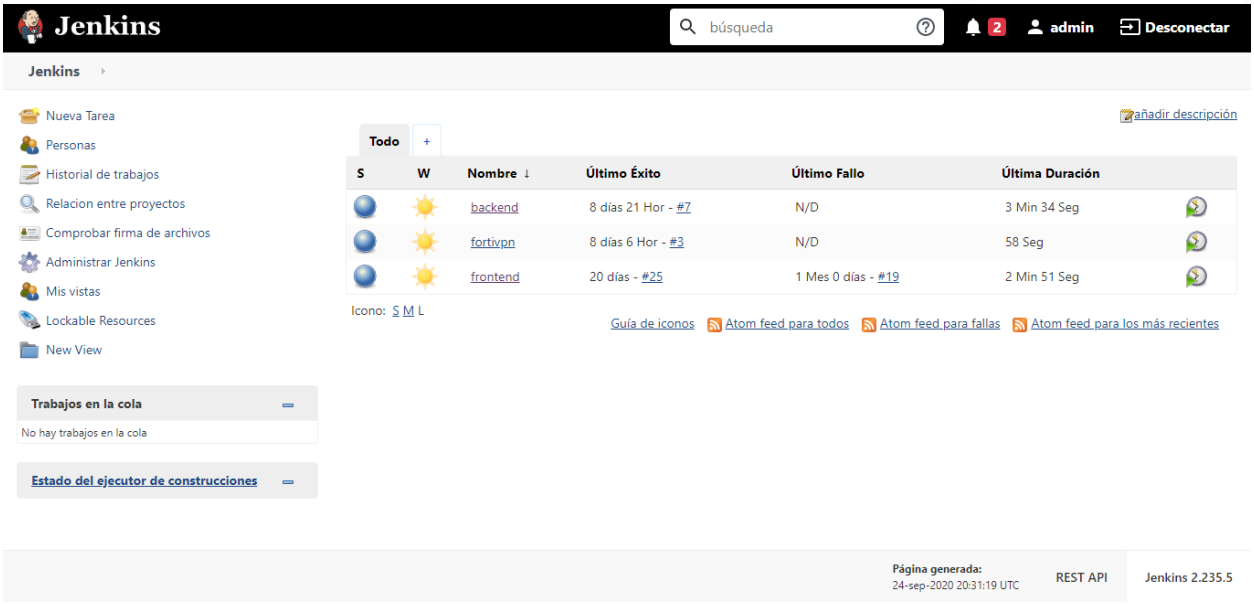
Google Cloud Console

Con el fin de asegurar que el sistema no presente fallas o advertencias de error, y estos problemas impidan la ejecución correcta de la actualización, se debe verificar en “GOOGLE CLOUD CONSOLE” en su componente de “KUBERNETES ENGINE”, verificando que los módulos desplegados se encuentren en estado correcto, de encontrar algún error verificar que los recursos de las cargas de trabajo este bien asignados.



- **JENKINS**

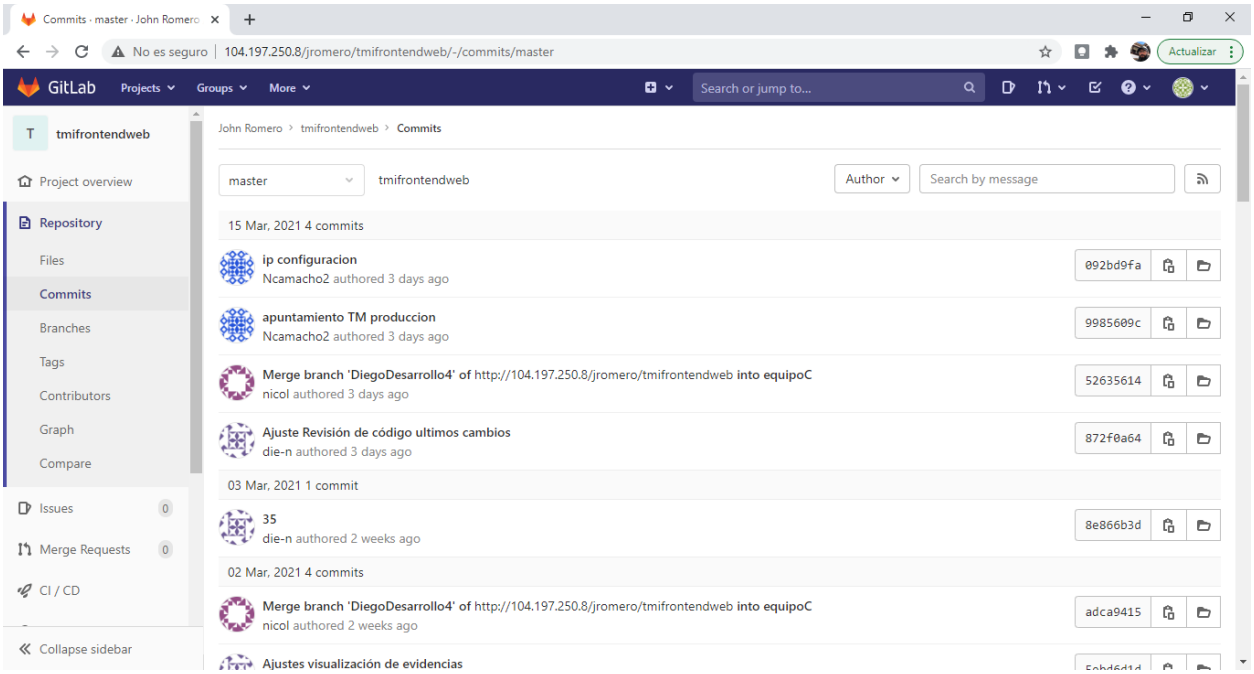
Con el fin de asegurar que el sistema no presente fallas o advertencias de error, y estos problemas impidan la ejecución correcta de la actualización, se debe verificar en Jenkins que las tareas se encuentren con el icono ☀️ el cual informa que la ejecución no presenta inconvenientes, si se encuentran iconos como los siguientes 🌤️ o ⛈️ deberá ingresar a la tarea y verificar el error que se está presentando y ejecutar “Construir Ahora”.

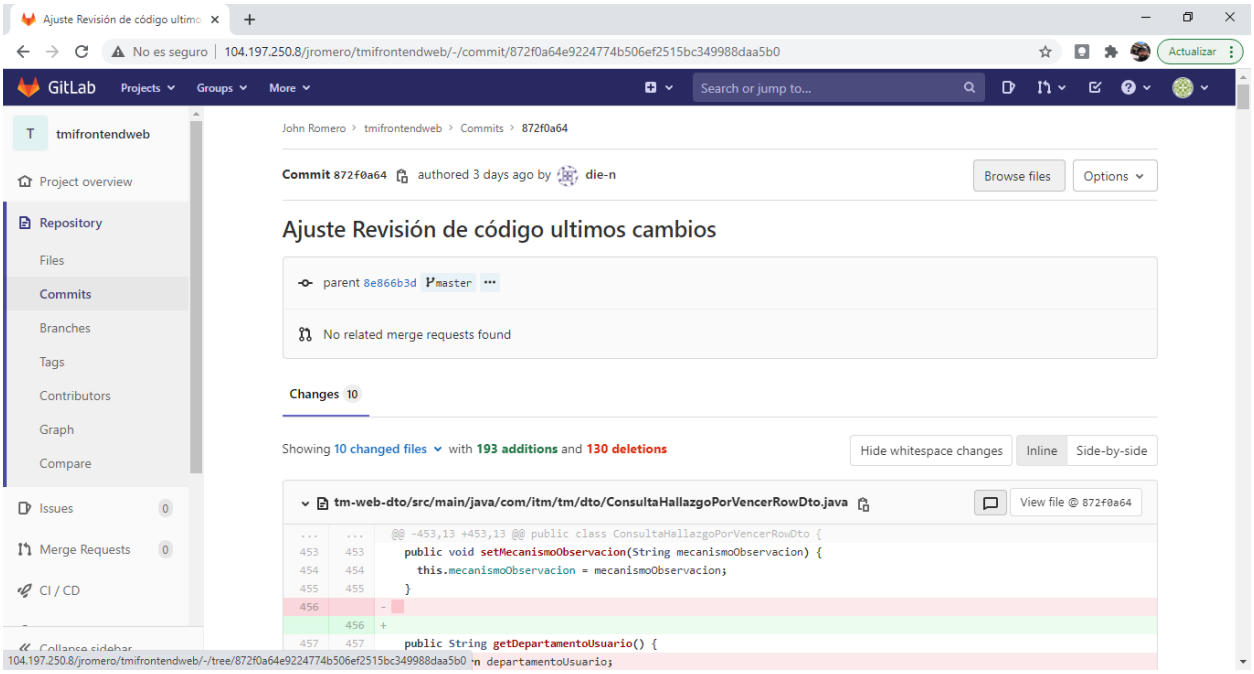


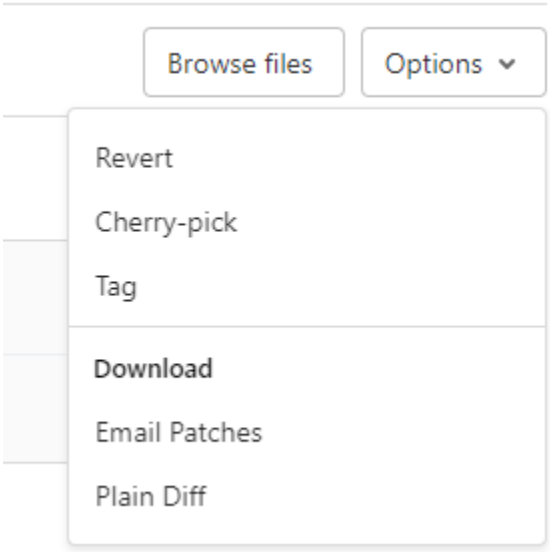
- **GITLAB**

  Revertir un commit

  Ingresar a la pagina de detalles del commit dar clic en optios y luego en revert







Information Trade Management Consulting de Colombia Ltda.
Sede Bogotá - Colombia:
Autopista Bogotá - Medellín Km 3,5 Vía Siberia
Municipio de Cota Centro Empresarial Metropolitano
Centro de Negocios Ofc. B - 19 - Costado Norte. Colombia.
PBX - FAX: (57 1) 841 55 51 Teléfono: (57 1) 821 91 72 / 76
Sede Miami - USA:
7015 N.W. 173 Dr Of 209 33015 Hialeah Florida Estados Unidos
PBX: (305) 600 - 0930
www.itmcolombia.com
info@itmcolombia.com

PROCEDIMIENTO DE INSTALACIÓN VERSIÓN ANTERIOR

Una vez se realice la actualización del sistema y se evidencie la ocurrencia de un evento no esperado, y se tome la decisión de reversar los cambios ejecutados por la actualización, se deben cargar los respaldos de la información de los componentes de:

- Base de datos
- Almacenamiento

Además se debe ejecutar el despliegue de la última versión estable de cogido del sistema de información en GITLab, que estaba en operación antes de la actualización.