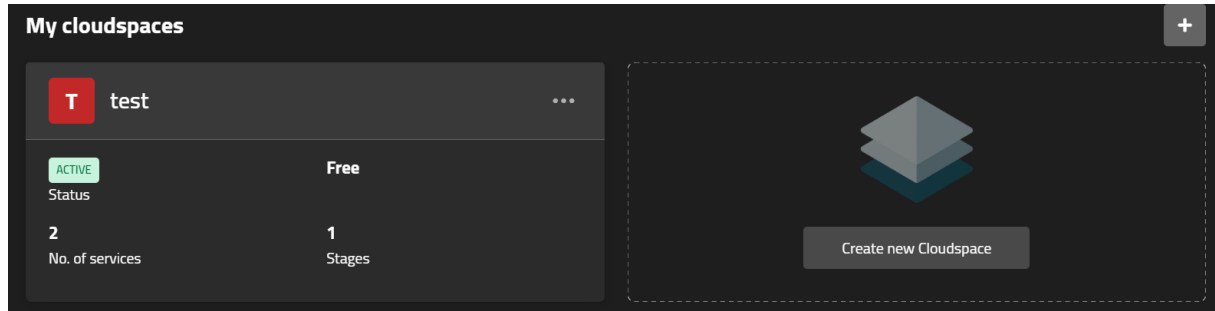


# TD Mise en production

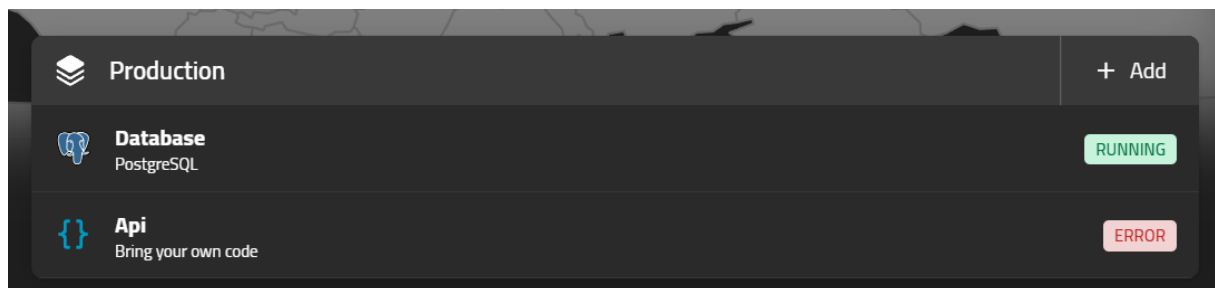
## Mogenius

Il faut tout d'abord créer un clouspace.



On lui donne le nom que l'on souhaite.

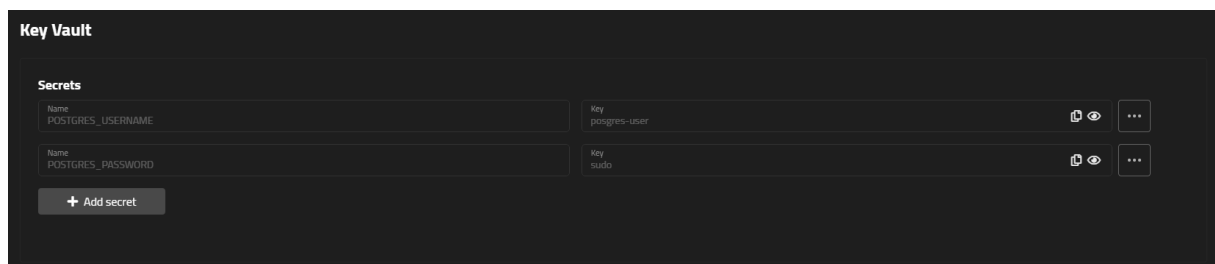
Une fois créé, ajouter des productions :



On peut choisir entre des templates déjà fait ou alors importer notre propre code depuis un dépôt git.

Pour la base de données, nous allons utiliser le modèle PostgreSQL. A l'initialisation, nous allons donner les valeurs du username, password et du nom de la base de données.

Pour faciliter la tâche, nous allons utiliser le Key Vault permettant d'enregistrer des variables d'environnement que nous pourrons réutiliser.



**Environment variables**

Define environment variables that your service can access at runtime. They are automatically injected so you can call them from your application. Read more

Type: Key Vault	Name: POSTGRES_PASSWORD	Secret: POSTGRES_PASSWORD	+ Create new secret
Type: Plaintext	Name: POSTGRES_DB	Value: covid-db	
Type: Volume Mount	Name: VOLUME-MOUNT	Source: pgdata-h10zy Destination: /var/lib/postgresql/data	
Type: Key Vault	Name: POSTGRES_USER	Secret: POSTGRES_USERNAME	+ Create new secret
Type: Change Owner	Name: CHOWN	User: 999 Group: 999 Folder: /pgdata-h10zy	

Maintenant que la base de données est initialisée, passons à la mise en place de l'API.

**Instance Logs** | **Deployment Logs**

● Instance 1

```

PostgreSQL Database directory appears to contain a database; Skipping initialization
2022-12-02 20:27:49.351 UTC [1] LOG:  starting PostgreSQL 14.2 (Debian 14.2-1.pgdg110+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit
2022-12-02 20:27:49.352 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
2022-12-02 20:27:49.353 UTC [1] LOG:  listening on IPv6 address "::", port 5432
2022-12-02 20:27:49.363 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2022-12-02 20:27:49.456 UTC [27] LOG:  database system was shut down at 2022-12-02 20:26:09 UTC
2022-12-02 20:27:49.527 UTC [1] LOG:  database system is ready to accept connections
  
```

Nous allons cette fois-ci importer du code depuis Git. A chaque push fait sur le repo (<https://github.com/Peepers/MogeniusTD/tree/main>), Mogenius va automatiquement clone le repo dans le service créé.

On initialise les variables d'environnements dans la page ENV VARS.

**Environment variables**

Define environment variables that your service can access at runtime. They are automatically injected so you can call them from your application. Read more

Type: Plaintext	Name: SPRING_PROFILES_INCLUDE	Value: no-liquibase	
Type: Key Vault	Name: SPRING_DATASOURCE_USERNAME	Secret: POSTGRES_USERNAME	+ Create new secret
Type: Key Vault	Name: SPRING_DATASOURCE_PASSWORD	Secret: POSTGRES_PASSWORD	+ Create new secret

C'est variable évité d'écrire directement l'username et el mot de passe dans le fichier application-prod.yaml et donc de les voir en clair sur Git.

! application-prod.yml ✕ ! application.yml

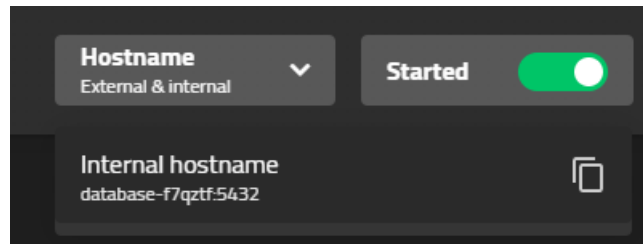
src > main > resources > ! application-prod.yml

You, il y a 1 seconde | 1 author (You)

```

1  spring:
2    datasource:
3      url: jdbc:postgresql://database-4s5skc:5432/covid-db
  
```

L'url de la base de données se récupère dans le service de celle-ci.



Le service démarre ensuite et se connecte à la base de données :

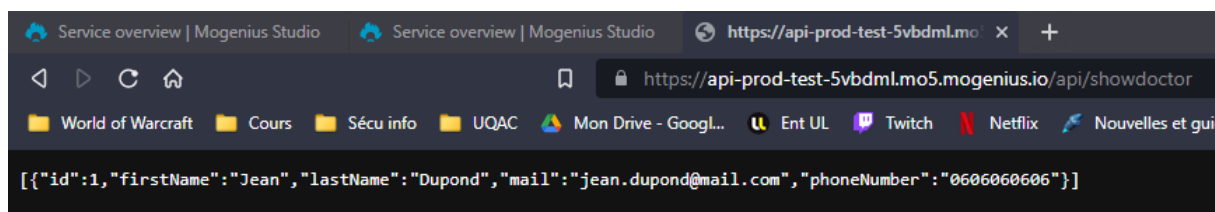
```
Instance 1
:: Spring Boot ::
2022-12-02 20:51:28.081 INFO 1 --- [main] o.polytech.covidapi.CovidapiApplication : Starting CovidapiApplication using Java 17.0.2 on api-3dck0-6656f698b-ghbd with PID 1 (/api/covid-api-0.0.1-SNAPSHOT.jar started by root in /api)
2022-12-02 20:51:28.084 INFO 1 --- [main] o.polytech.covidapi.CovidapiApplication : The following 2 profiles are active: "no-liquibase", "prod"
2022-12-02 20:51:35.500 INFO 1 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-12-02 20:51:36.204 INFO 1 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 697 ms. Found 8 JPA repository interfaces.
2022-12-02 20:51:42.488 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatEmbeddedServer : Tomcat initialized with port(s): 9797 (http)
2022-12-02 20:51:42.681 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-12-02 20:51:42.681 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.65]
2022-12-02 20:51:43.386 INFO 1 --- [main] o.a.c.c.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-12-02 20:51:43.488 INFO 1 --- [main] w.a.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 14796 ms
2022-12-02 20:51:47.508 INFO 1 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-12-02 20:51:49.583 INFO 1 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-12-02 20:51:51.384 INFO 1 --- [main] liquibase.database : Set default schema name to public
2022-12-02 20:51:52.686 INFO 1 --- [main] liquibase.lockservice : Successfully acquired change log lock
2022-12-02 20:51:55.218 INFO 1 --- [main] liquibase.lockservice : Reading from public database changelog
2022-12-02 20:51:55.885 INFO 1 --- [main] liquibase.lockservice : Successfully released change log lock
2022-12-02 20:51:56.788 INFO 1 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2022-12-02 20:51:57.186 INFO 1 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.11.Final
2022-12-02 20:51:58.310 INFO 1 --- [main] o.hibernate.annotations.common.Version : HC40000001: Hibernate Commons Annotations {5.12.1.Final}
2022-12-02 20:51:59.181 INFO 1 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.PostgreSQLDialect
```

Et créer ensuite les tables et les remplies avec les données :

```
Instance 1
Hibernate: alter table admin add constraint FKfk0v0z9r1k1mmv0p0u9p foreign key (id_center) references center
Hibernate: alter table center add constraint FKfk0v0z9r1k1mmv0p0u9p foreign key (id_address) references address
Hibernate: alter table doctor add constraint user_id_address_fk foreign key (id_address) references address
Hibernate: alter table doctor add constraint doctor_id_center_fk foreign key (id_center) references center
Hibernate: alter table patient add constraint user_id_address_fk foreign key (id_address) references address
Hibernate: alter table super_admin add constraint user_id_address_fk foreign key (id_address) references address
2022-12-02 20:52:06.348 INFO 1 --- [main] o.a.t.s.schema.internal.SchemaCreatorImpl : HHH000476: Executing import script 'jar:file:/api/covid-api-0.0.1-SNAPSHOT.jar!/ROOT-INF/classes/import.sql'
Hibernate: INSERT INTO address (city, street, zipcode) VALUES ('Nancy', 'Place Stanislas', '54000')
Hibernate: INSERT INTO address (city, street, zipcode) VALUES ('Lyon', 'rue Jean-Luc', '69000')
Hibernate: INSERT INTO address (city, street, zipcode) VALUES ('Paris', 'rue de la Paix', '75000')
Hibernate: INSERT INTO address (city, street, zipcode) VALUES ('Nancy', '1d de Strasbourg', '54000')
Hibernate: INSERT INTO center (name, capacity, timetable, id_address) VALUES ('Centre de Nancy', '40', 'timetable', '5')
Hibernate: INSERT INTO center (name, capacity, timetable, id_address) VALUES ('Saint-Jac', '40', 'timetable', '1')
Hibernate: INSERT INTO center (name, capacity, timetable, id_address) VALUES ('Saint-Seb', '40', 'timetable', '5')
Hibernate: INSERT INTO center (name, capacity, timetable, id_address) VALUES ('Stan', '40', 'timetable', '2')
Hibernate: INSERT INTO patient (firstname, lastname, mail, phone_number, id_address, vaccinated) VALUES ('Jean', 'Dupond', 'jean.dupond@mail.com', '0606060606', '1', '1')
Hibernate: INSERT INTO doctor (firstname, lastname, mail, phone_number, id_address, id_center) VALUES ('Pierre', 'Moulin', 'pierre.moulin@mail.com', '0606060606', '2', '1')
Hibernate: INSERT INTO admin (firstname, lastname, mail, phone_number, id_address, id_center) VALUES ('admin', 'admin', 'admin@mail.com', '1234567', '2', '1')
Hibernate: INSERT INTO admin (firstname, lastname, mail, phone_number, id_address, id_center) VALUES ('admin2', 'admin2', 'admin2@mail.com', '1234567', '2', '1')
Hibernate: INSERT INTO admin (firstname, lastname, mail, phone_number, id_address, id_center) VALUES ('admin3', 'admin3', 'admin3@mail.com', '1234567', '1', '1')
Hibernate: INSERT INTO super_admin (firstname, lastname, mail, phone_number, id_address) VALUES ('supadmin', 'supadmin', 'supadmin@mail.com', '1234567', '4')
Hibernate: INSERT INTO appointment (jour) VALUES ('25-11-2022')
```

On peut maintenant accéder au l'adresse :

<https://api-prod-test-5vbdml.mo5.mogenius.io/api/showdoctor>



Arguments possibles disponibles dans le readme du dépôt.