

PROCESS DE DEPLOIEMENT D'UNE APPLICATION

Pour déployer votre application, vous avez plusieurs techniques et outils possible selon votre type d'application et votre catégorie de serveur (serveur mutualisé, serveur dédié ...) et vos besoins.

NB :

- Si vous utilisez un serveur mutualisé(achat d'une partie de espace d'hébergement avec un fournisseur de service), généralement aucune config n'est nécessaire car votre fournisseur se charge d'installer et de partager les parts de chaque utilisateur;
- Si vous utilisez un serveur dédié, le serveur vous appartient, vous devez donc faire toutes les config dont vous avez besoin

Cas pratique de déploiement d'une application Java/Spring et Angular sur un serveur dédié (Oracle Linux)

Par où commencer ?

Vous pouvez commencer par installer les outils dont vous avez besoin sur votre serveur.

Dans notre cas particulier nous aurons besoin de :

- Vérifier la version du serveur Oracle Linux;
- Installer un JDK (prenez la même version que celle de votre application);
- Une version de PHP;
- Installer et configurer un serveur Nginx (Ce que nous allons utiliser) | ou apache;

NB : Pour un serveur Linux nous avons 2 commandes principales selon la version :

- yum (pour oracle 7)
- dnf (pour Oracle 8+)

Vérifiez la version de votre serveur

```
cat /etc/oracle-release
```



Un aperçu en image

```
[root@localhost Apis]# cat /etc/oracle-release
Oracle Linux Server release 8.8
```

Notre utilisons un serveur

Oracle Linux 8 ce qui veut dire que les commande vont commencer par `dnf`

I- Installation des outils

1- Installez un JDK (Java 11 dans notre cas)

```
sudo dnf install java-11-openjdk-devel
```



```
[root@localhost ~]# sudo dnf install java-11-openjdk-devel
Last metadata expiration check: 3:37:04 ago on Mon 31 Jul 2023 11:22:41 AM GMT.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
java-11-openjdk-devel                  x86_64            1:11.0.20.0.8-2.0.1.el8    ol8_appstream      3.4 M
Installing dependencies:
copy-jdk-configs                       noarch            4.0-2.el8               ol8_appstream       30 k
java-11-openjdk                        x86_64            1:11.0.20.0.8-2.0.1.el8    ol8_appstream       473 k
java-11-openjdk-headless               x86_64            1:11.0.20.0.8-2.0.1.el8    ol8_appstream       42 M
javapackages-filessystem               noarch            5.3.0-1.module+el8+5136+7ff78f74 ol8_appstream       30 k
lksctp-tools                          x86_64            1.0.18-3.el8            ol8_baseos_latest   100 k
ttmkfdir                             x86_64            3.0.9-54.el8            ol8_appstream        62 k
xorg-x11-fonts-Type1                  noarch            7.5-19.el8              ol8_appstream       522 k
Enabling module streams:
javapackages-runtime                  201801
=====
Transaction Summary
=====
Install 8 Packages

Total download size: 46 M
Installed size: 185 M
Is this ok [y/N]: Y
Downloading Packages:
(1/8): copy-jdk-configs-4.0-2.el8.noarch.rpm          50 kB/s | 30 kB    00:00
(2/8): lksctp-tools-1.0.18-3.el8.x86_64.rpm          134 kB/s | 100 kB   00:00
(3/8): java-11-openjdk-11.0.20.0.8-2.0.1.el8.x86_64.rpm 502 kB/s | 473 kB   00:00
(4/8): javapackages-filessystem-5.3.0-1.module+el8+5136+7ff78f74.noarch.rpm 183 kB/s | 30 kB    00:00
(5/8): ttmkfdir-3.0.9-54.el8.x86_64.rpm              545 kB/s | 62 kB    00:00
(6/8): java-11-openjdk-devel-11.0.20.0.8-2.0.1.el8.x86_64.rpm 4.0 MB/s | 3.4 MB   00:00
(7/8): xorg-x11-fonts-Type1-7.5-19.el8.noarch.rpm     1.9 MB/s | 522 kB   00:00
(8/8): java-11-openjdk-headless-11.0.20.0.8-2.0.1.el8.x86_64.rpm 17 MB/s | 42 MB     00:02
-----
Total                                          14 MB/s | 46 MB     00:03
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

2- Vérifiez si le JDK est parfaitement installé

```
java -version
```



Un aperçu 🖱️🖱️

```
[root@localhost ~]# java -version
openjdk version "11.0.20" 2023-07-18 LTS
OpenJDK Runtime Environment (Red_Hat-11.0.20.0.8-2.0.1) (build 11.0.20+8-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-11.0.20.0.8-2.0.1) (build 11.0.20+8-LTS, mixed mode, sharing)
[root@localhost ~]#
```

3- Installation de PHP 7 (version 7.2 dans notre cas)

```
sudo dnf install php
```



- Taper y pour accepter. A la fin de l'installation 🖱️
- Vérifiez si php est bien installé :

```
php -v
```



```
Installed:
apr-1.6.3-12.el8.x86_64
apr-util-bdb-1.6.1-6.el8_8.1.x86_64
httpd-2.4.37-56.0.1.module+el8.8.0+21057+13668aee.6.x86_64
httpd-tools-2.4.37-56.0.1.module+el8.8.0+21057+13668aee.6.x86_64
oracle-logos-httpd-84.5-1.0.1.el8.noarch
php-cli-7.2.24-1.module+el8.2.0+5510+6771133c.x86_64
php-fpm-7.2.24-1.module+el8.2.0+5510+6771133c.x86_64
apr-util-1.6.1-6.el8_8.1.x86_64
apr-util-openssl-1.6.1-6.el8_8.1.x86_64
httpd-filesystem-2.4.37-56.0.1.module+el8.8.0+21057+13668aee.6.noarch
mod_http2-1.15.7-8.module+el8.8.0+21057+13668aee.3.x86_64
php-7.2.24-1.module+el8.2.0+5510+6771133c.x86_64
php-common-7.2.24-1.module+el8.2.0+5510+6771133c.x86_64

Complete!
[root@localhost ~]# php -v
PHP 7.2.24 (cli) (built: Oct 22 2019 08:28:36) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
```

4- Installation et configuration du serveur Nginx

Étape 4-1: Installer Nginx

```
sudo dnf install nginx
```



Étape 4-2: Démarrer le service Nginx Une fois installé, démarrez le serveur

```
sudo systemctl start nginx
```



Étape 4-3: Activer le démarrage automatique de Nginx (pas obligatoire mais recommandé )

Si vous souhaitez que Nginx démarre automatiquement à chaque démarrage du système(après coupure d'électricité par exemple), exécutez la commande suivante pour activer le service au démarrage :

```
sudo systemctl enable nginx
```



Étape 4-4: Vérifier l'état de Nginx

```
sudo systemctl status nginx
```



Vous devez voir 📌📌

```
[root@localhost Apis]# sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/nginx.service.d
            └─php-fpm.conf
   Active: active (running) since Mon 2023-07-31 14:44:58 GMT; 4h 25min ago
     Main PID: 1154 (nginx)
       Tasks: 3 (limit: 5208)
      Memory: 2.5M
     CGroup: /system.slice/nginx.service
            └─1154 nginx: master process /usr/sbin/nginx
               1159 nginx: worker process
               1160 nginx: worker process

Jul 31 14:44:58 localhost.localdomain systemd[1]: Starting The nginx HTTP and reverse proxy server...
Jul 31 14:44:58 localhost.localdomain nginx[1020]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jul 31 14:44:58 localhost.localdomain nginx[1020]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jul 31 14:44:58 localhost.localdomain systemd[1]: Started The nginx HTTP and reverse proxy server.
```

Bien ! A ce stade vous devez pouvoir voir sur un navigateur web la page d'accueil de votre serveur nginx à l'adresse de votre serveur exemple 192.168.25.25 .

🚧 Cependant si votre serveur contient un pare-feu 🚦 , il peut vous empêcher d'avoir accès ❌

Alors

- Vérifiez d'abord l'existence d'un pare-feu.

```
sudo systemctl status firewalld
```



■ S'il n'existe aucun pare-feu 🚦❌📌

```
[root@localhost Apis]# sudo firewall-cmd --add-service=http --permanent
Firewalld is not running
[root@localhost Apis]#
```

■ Si le pare-feu existe 🚦✅📌


```

[root@localhost Apis]# sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-08-01 11:49:03 GMT; 46s ago
     Docs: man:firewalld(1)
  Main PID: 17324 (firewalld)
    Tasks: 2 (limit: 5208)
   Memory: 29.0M
    CGroup: /system.slice/firewalld.service
            └─17324 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

Aug 01 11:49:02 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Aug 01 11:49:03 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Aug 01 11:49:03 localhost.localdomain firewalld[17324]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will b
lines 1-13/13 (END)

```

- NB : Si vous n'avez pas de pare-feu en exécution vous pouvez passer directement à l'étape II - DEPLOIEMENT DES PPLICATIONS

Au cas où il existe un pare-feu sur votre serveur : Étape 4-5: Configurer le pare-feu (firewalld) pour Nginx (En cas de besoin) Si vous avez activé le pare-feu firewalld sur votre système, vous devrez peut-être configurer les règles pour permettre le trafic HTTP (port 80) et le trafic HTTPS (port 443) pour Nginx. Pour autoriser le trafic HTTP, utilisez la commande suivante :

```
sudo firewall-cmd --add-service=http --permanent
```



Si tout s'est bien passé, vous devriez voir un message indiquant que le service est actif et en cours d'exécution. Sinon si aucun pare-feu ne bloque l'accès, vous devriez voir un message du genre 🙌🙌 :

```

[root@localhost Apis]# sudo firewall-cmd --add-service=http --permanent
Firewalld is not running
[root@localhost Apis]#

```

Vous pouvez avoir besoin d'autoriser le trafic HTTPS également;

- Pour autoriser le trafic HTTPS, utilisez la commande suivante :

```
sudo firewall-cmd --add-service=https --permanent
```



Après avoir ajouté les règles, rechargez votre pare-feu (firewalld) pour qu'elles prennent effet :

```
sudo firewall-cmd --reload
```



Bien ! A ce stade vous devez pouvoir voir sur un navigateur web la page d'accueil de votre serveur nginx à l'adresse de votre serveur exemple 192.168.25.25 . 🖱️🖱️

Welcome to **nginx** on Oracle Linux!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Oracle Linux. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

For information on Oracle Linux, please visit the [Oracle website](#). The documentation for Oracle Linux is [available on the Oracle website](#).



Super, votre environnement est prêt à recevoir vos applications (java et Angular)

II- Déploiement des applications

Cette partie considère que vous avez déjà vos applications prêts pour la production (build déjà ok);

Vous devez envoyer vos applications(java et angular) sur votre serveur, pour ce faire vous pouvez utiliser des outils tels :

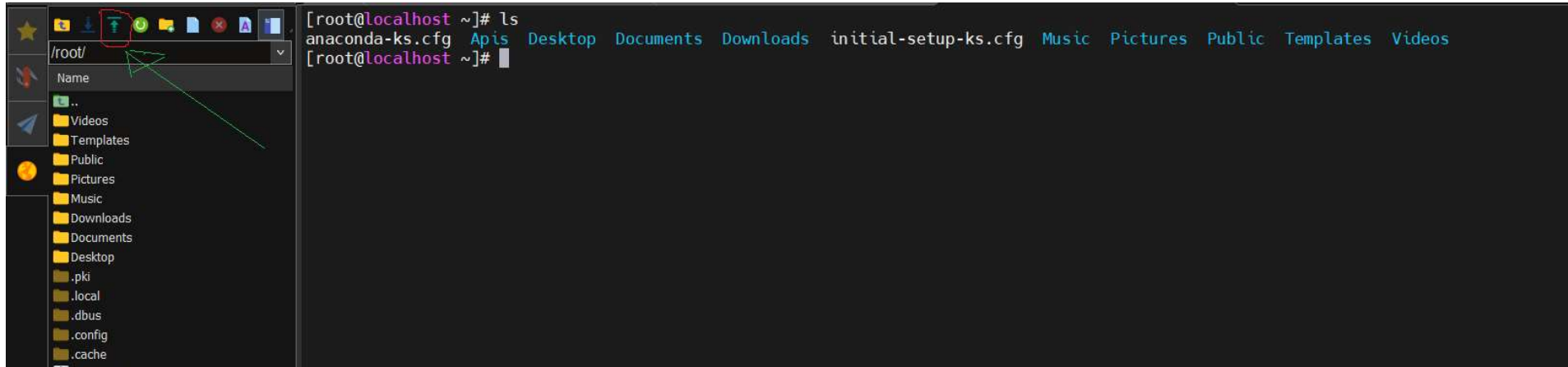
■ [MobaXterm](#)

■ [FileZilla](#)

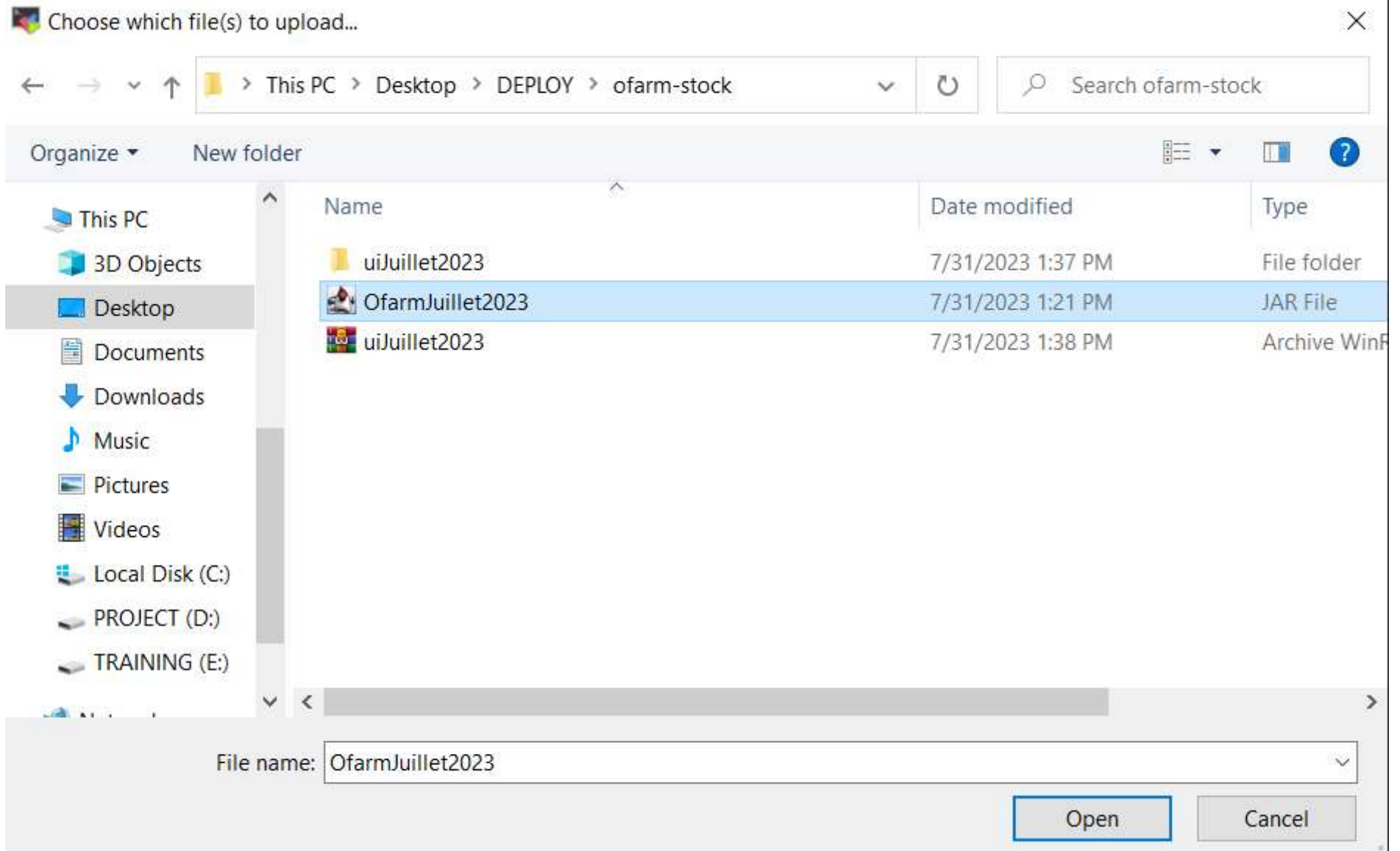
Dans cet exemple nous utiliserons MobaXterm

1- Uploader le(s) fichier(s) de déploiement back (java)

Une fois connecté avec MobaXterm, vous pouvez cliquer sur cette flèche comme le montre l'image :

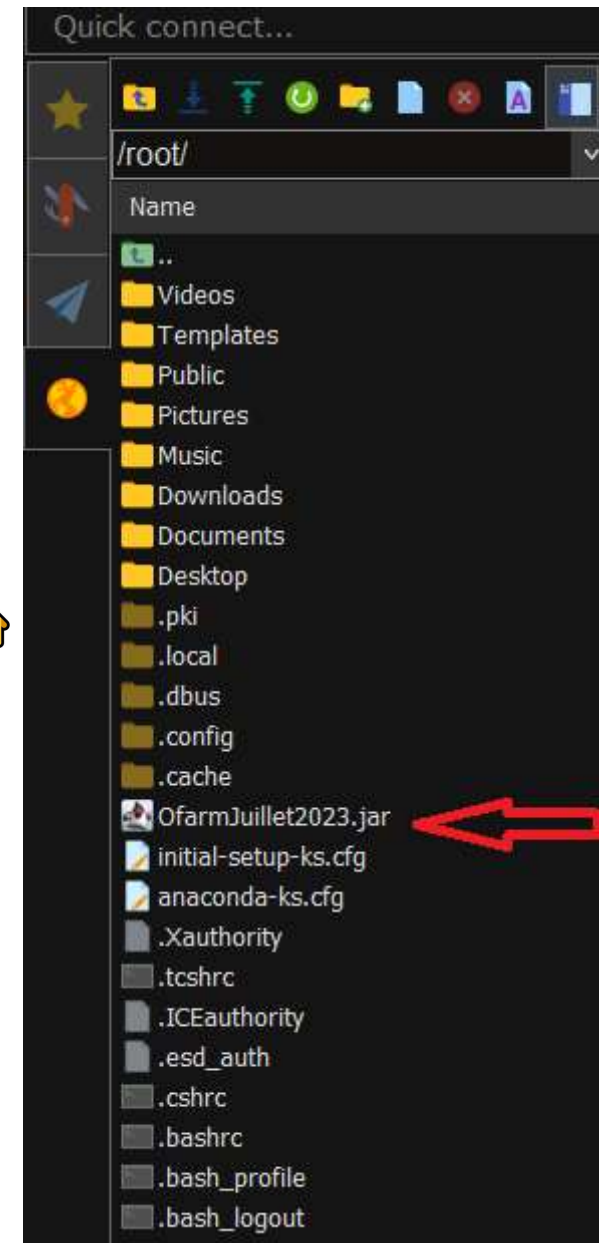


Choisissez le(s) fichier(s) à uploader.



Après chargement..

Vous devez avoir le(s) fichier uploadé à la racine de votre serveur comme sur l'image 📌



Vous pouvez le déplacer dans un dossier de votre choix, ensuite exécutez la commande suivante pour démarrer l'application java :



```
[root@localhost api]# ls
0farmJuillet2023.jar web-app web-app.zip
[root@localhost api]# java -jar 0farmJuillet2023.jar
```

```

      .__
     / __ \____ _ __| | | |
    / /_/ /_ __/ __/ |_| | |
   / ____/___/_/|\__/\___|_|_|
  :: Spring Boot ::                (v2.5.5)

2023-07-31 16:44:54.242 INFO 222097 --- [main] c.t.e.g.GestionStockApplication : Starting GestionStockApplication v0.0.1-SNAPSHOT using J
ava 11.0.19 on localhost.localdomain with PID 222097 (/root/api/0farmJuillet2023.jar started by root in /root/api)
2023-07-31 16:44:54.249 INFO 222097 --- [main] c.t.e.g.GestionStockApplication : The following profiles are active: development
2023-07-31 16:44:59.428 INFO 222097 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mo
de.
2023-07-31 16:45:04.123 INFO 222097 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 4672 ms. Fou
nd 42 JPA repository interfaces.
2023-07-31 16:45:05.726 INFO 222097 --- [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.security.access.expression.met
hod.DefaultMethodSecurityExpressionHandler@59d2103b' of type [org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler] is
not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2023-07-31 16:45:05.737 INFO 222097 --- [main] trationDelegate$BeanPostProcessorChecker : Bean 'methodSecurityMetadataSource' of type [org.springf
ramework.security.access.method.DelegatingMethodSecurityMetadataSource] is not eligible for getting processed by all BeanPostProcessors (for example: not elig
ible for auto-proxying)
2023-07-31 16:45:06.802 INFO 222097 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8083 (http)
2023-07-31 16:45:06.868 INFO 222097 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-07-31 16:45:06.869 INFO 222097 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.53]
2023-07-31 16:45:07.065 INFO 222097 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-07-31 16:45:07.066 INFO 222097 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in
12571 ms
```

Super ! votre application java est lancée, maintenant, faite pareil pour votre application angular

Dans notre nous avons importé le fichier angular(le build compressé) sous le nom de uiJuillet2023.zip

Nous devons le déplacer dans un dossier de notre choix sur le serveur, mais il est recommandé de mettre les interfaces dans le dossier `var/www/html`.

- Déplacez donc votre fichier dans le repertoire en question :

```
sudo mv <NOM_DU_BUILD_COMPRESSE_DE_VOTRE_APPLI> /var/www/html
```



Dans notre cas(nous considérons que votre fichier est à la racine du serveur :

```
mv /root/uiJuillet2023.zip /var/www/html
```



- Ensuite, décompressez le fichier (unzip pour .zip et unrar pour .rar)

```
unzip <NOM_DU_BUILD_COMPRESSE_DE_VOTRE_APPLI>
```



Ce qui donne dans notre cas :

```
unzip uiJuillet2023.zip
```



- Entrez dans le dossier décompressé pour lancer l'application :

```
cd /var/www/html/uiJuillet2023
```



- Lancez l'application Angular

```
php -S 0.0.0.0:PORT_LIBRE_DE_VOTRE_CHOIX
```



Dans notre cas lançons l'appli sur le port 87;


```
php -S 0.0.0.0:87
```



Sur votre navigateur aller à l'adresse 📍 : `adresse_de_votre_serveur:port_choisi` exemple : `192.168.25.25:87` votre application est bien lancée 🚀

Bravo ! vous venez de déployer votre application sur un serveur dédié.

Have Happy Works <\ 🎉 >
Alban 🙌