

# Setup

Thursday, December 23, 2021 4:08 PM

## Planloader Backend Services Setup (NodeJS)

Create following 5 environment variables to your host OS.

- 1) **PlanloaderConfigDbHost**=uat-database.c8htwvcfrdzd.eu-west-1.rds.amazonaws.com
- 2) **PlanloaderConfigDbUsername**=<db\_user\_with\_read\_access>
- 3) **PlanloaderConfigDbPassword**=<db\_user\_password>
- 4) **PlanloaderConfigDbDatabase**=PlanloaderConfigDb
- 5) **PlanloaderConfigEnv**=development (for test set it to 'test', for server set it to 'production')

### Local/Development Setup:

Git clone your repositories from azure [https://dev.azure.com/GagaMuller/PlanLoader/\\_git](https://dev.azure.com/GagaMuller/PlanLoader/_git)

- Each service require some configurations to run, configurations keys are saved in [config/custom-environment-variable.yml](#)
- These configuration come from **PlanloaderConfigDb**
- Each service save its required configurations in **.env** file and start itself
- In order to get latest configurations run this command **\$ node ./setup-self.js**
- This Depend on what you set for **PlanloaderConfigEnv** variable, If you set it development then it get development configurations from db and create **.env** file.
- Then you can run your service by **nodemon** or **pm2**
- If you want to run with you can **pm2** eliminate above step by running this **./restart-pm2.sh** shell file. This execute above file **./setup-self.js** and stop/restart service by **pm2**
- Give execute permission to **restart-pm2.sh** file as **sudo chmod 755 ./restart-pm2.sh**

### Config-Listener Service:

- This service is use to monitor configuration changes in **PlanloaderConfigDb**
- It recheck after a specific period of time (1 min)
- If there is any change in configurations, 'update\_delivered' field in the database will be 0 its mean there is a change in configuration, so rollback all the services
- If configuration change is found then it execute the **restart-all-services.sh** shell file which go to each directory of services and call **./restart-pm2.sh** shell file. In short it restart all the services and service get the latest configurations.
- Give execute permission to **restart-all-services.sh** file as **sudo chmod 755 ./restart-all-services.sh**
- Make sure this service is in the same directory where you clone the planloader service

**Config-Listener Service Additional:**

This service once first start, create vhost configuration file "planloader-vhost.conf"

-> See section "setting Apache virtual host"

which is used in Apache http server this file is create in one directory back from this service

**e.g:** If config listener service is clone at -> /Users/basittraza/Repos/planloader/Config-Listener/

Then the generated file will be at -> /Users/basittraza/Repos/planloader/planloader-vhost.conf

And this file path is exactly used in Apache vhost configuration

After creating planloader-vhost.conf it tries to restart httpd service with linux OS command,

In-case your OS is different or httpd service failed to restart, still planloader-vhost.conf file has been generated you can restart http service manually

**Server Setup:**

1. Set **PlanloaderConfigEnv**=production (see above for detail)

Everything else is same.

## Setting Apache Virtual Host

1. Install XAMPP on your system
2. Go to your Xampp httpd.conf and add below line, replace path by your path
 

```
# ----- INCLUDING PLANLOADER VHOST -----
IncludeOptional <path_of_your_repositories_root_folder>/planloader-vhost.conf
#e.g /Users/basittraza/Repos/planloader/planloader-vhost.conf
# ----- INCLUDING PLANLOADER VHOST -----
```
3. All done now restart Apache web server (from XAMPP)

## Clone all services, Start Config Listener and Setup Vhost Configuration and restart httpd service Automatically

1. Go to your main directory where you want to clone repositories
2. Create a file "**clone\_planloader\_repos.sh**" and give execute permission to this bash file as "**sudo chmod 755 clone\_planloader\_repos.sh**"
3. Open this file and add below content and follow step#4
4. Run this bash file from your terminal as "**./clone\_planloader\_repos.sh**"

```
#!/bin/sh
```

```
#Exit if any command failed
```

```
set -e
```

```
AuthService="master"  
RouterService="master"  
UserService="master"  
LocationService="master-v2"  
TaskActivityService="master"  
AccountService="master"  
ProjectService="master"  
MessagingService="master"  
PubnubService="master"  
NotificationService="master"  
BoardService="master"  
SwaggerService="master"  
ConfigListenerService="master"
```

#### # Cloning Auth Service

```
git clone -b $AuthService  
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-auth-  
service  
cd planloader-auth-service  
npm install  
cd ..
```

#### # Cloning Router Service

```
git clone -b $RouterService  
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/Planloader-API-  
gateway  
cd Planloader-API-gateway  
npm install  
cd ..
```

#### # Cloning User Service

```
git clone -b $UserService  
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-user-  
service  
cd planloader-user-service  
npm install  
cd ..
```

#### # Cloning Location Service

```
git clone -b $LocationService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-web-
location-service
cd planloader-web-location-service
npm install
cd ..
```

#### # Cloning Task Activity Service

```
git clone -b $TaskActivityService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-task-
activity-service
cd Planloader-task-activity-service
npm install
cd ..
```

#### # Cloning Account Service

```
git clone -b $AccountService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-
accounts-service
cd planloader-accounts-service
npm install
cd ..
```

#### # Cloning Project Service

```
git clone -b $ProjectService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-
projects-service
cd planloader-projects-service
npm install
cd ..
```

#### # Cloning Board Messaging Service

```
git clone -b $MessagingService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-board-
messaging-service
cd planloader-board-messaging-service
npm install
cd ..
```

#### # Cloning Pubnub Service

```
git clone -b $PubnubService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-pubnub-
subscription-service
cd planloader-pubnub-subscription-service
npm install
cd ..
```

#### # Cloning Board Service

```
git clone -b $BoardService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-boards-
service
cd planloader-boards-service
npm install
cd ..
```

#### # Cloning Notification Service

```
git clone -b $NotificationService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-
notification-service
cd planloader-notification-service
npm install
cd ..
```

#### # Cloning Swagger Service

```
git clone -b $SwaggerService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/planloader-swagger
cd planloader-swagger
npm install
cd ..
```

#### # Cloning Config Listener Service

```
git clone -b $ConfigListenerService
git@ssh.dev.azure.com:v3/GagaMuller/PlanLoader/PlaLoader-
ConfigListener
cd PlaLoader-ConfigListener
npm install
echo 'All planlaoder services are cloned'
echo 'Starting Config Listener services with PM2'
pm2 restart ecosystem.config.js
```

```
echo 'All Done!'
```

```
# keep track of the last executed command
trap 'last_command=$current_command;
current_command=$BASH_COMMAND' DEBUG
# echo an error message before exiting
trap 'echo "\"${last_command}\" command filed with exit code $?."'
EXIT
```

## Verify Results

After cloning all repositories and setting Apache XAMPP, make sure atleast swagger-service is running  
Then go to browser and hit <http://localhost.planloader.com/api-docs>