

# Pipeline laravel

- create project in gitlab : name is laravel-project
- composer create-project laravel/laravel laravel-project
- Push an existing folder

```
git init --initial-branch=main
```

```
git remote add origin
```

```
https://gitlab.com/wissemsghaier2000/laravel-project.git
```

```
git add .
```

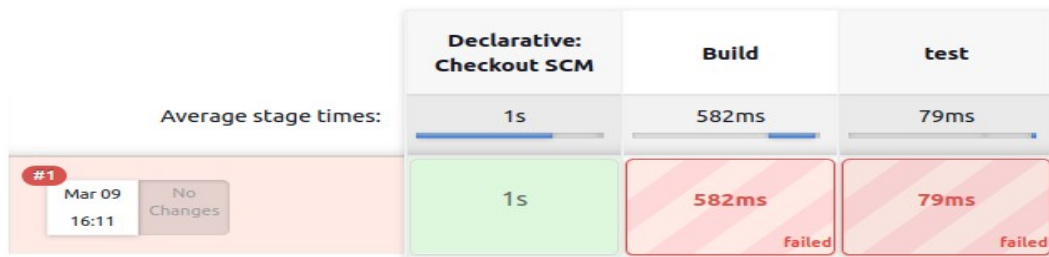
```
git commit -m "Initial commit"
```

```
git push -u origin main
```

- create jenkinsfile in directory laravel-project :

```
pipeline {  
  agent any  
  
  stages {  
    stage('Build') {  
      steps {  
        //sh "composer update"  
        sh "composer install"  
        sh 'cp .env.example .env'  
        sh 'php artisan key:generate'  
      }  
    }  
    stage('test') {  
      steps {  
        sh "php artisan test "  
        sh './vendor/bin/phpunit'  
      }  
    }  
  }  
}
```

## Stage View



## Permalinks

- Edit dockerfile in jenkins

```
FROM jenkins/jenkins:2.393
```

```
LABEL maintainer="mstewart@riotgames.com"
```

```
USER root
```

```
RUN mkdir /var/log/jenkins
```

```
RUN chown -R jenkins:jenkins /var/log/jenkins
```

```
RUN apt-get update && \
```

```
    apt-get install -y apt-utils && \
```

```
    apt-get update && \
```

```
    apt-get install -y software-properties-common && \
```

```
    apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
```

```
E5267A6C && \
```

```
    add-apt-repository ppa:ondrej/php && \
```

```
    sed -i 's/lunar/focal/g' /etc/apt/sources.list.d/ondrej-ubuntu-  
php*.list && \
```

```
    apt-get update && \
```

```
    apt-get install -y php8.1-cli php8.1-xml php8.1-curl
```

```
RUN curl -sS https://getcomposer.org/installer | php -- --install-  
dir=/usr/local/bin --filename=composer
```

```
USER jenkins
```

"apt-get install -y apt-utils"

- to install the "apt-utils" package.
- used with the Advanced Packaging Tool (APT)

**apt-get install -y software-properties-common**

- "software-properties-common" package contains the common infrastructure for adding and managing software repositories

**apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E5267A6C**

- This is typically used to add a trusted key for a software repository.

**add-apt-repository ppa:ondrej/php**

- adds the PPA (Personal Package Archive) "ppa:ondrej/php" to the list of software repositories
- This PPA contains packages of the PHP programming language that are not included in the default repositories.

**apt-get install -y php8.1-cli php8.1-xml php8.1-curl**

- The "php8.1-cli" package contains the PHP command line
- "php8.1-xml" package contains the PHP XML extension
- "php8.1-curl" package contains the PHP cURL extension

**RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer**

- install Composer, a popular PHP dependency manager
- "curl" utility to download the Composer installer script from the official website
- "sS" options suppress any error messages.
- "--install-dir" option specifies the directory where Composer will be installed
- "--filename" option specifies the name of the Composer

You can install the "mbstring" extension inside jenkins CLI by running tests in pipeline

- apt-get install php8.1-mbstring
- apt-get install php8.1-xml

inside jenkinsfile:

- To generate an application encryption key, you can run the following command

```
sh 'cp .env.example .env'
```

```
sh 'php artisan key:generate'
```

will generate a new encryption key and add it to your application's .env file

Once the key is generated, you should be able to run the php artisan test command

```
sh './vendor/bin/phpunit'
```

- The command ./vendor/bin/phpunit is used to run PHPUnit tests in a Laravel project.

