

# 3share-cli

---

The 3Share CLI helps AEM developers save time in building local development environments for their projects. It utilizes docker to build isolated AEM environments and the configuration is save in your project directory. There is a built reverse proxy that gives you a local url access to the servers in your stack.

## Why use 3share-cli

Many time you may want to work on different AEM version, content and code bases. Managing the running AEM servers, JDK versions and Maven setup can get problematic. With the cli, you have an isolated environment with the right AEM, jdk and a build conatiner that has your Maven dependencies. Using it it's possible to run multiple stacks at the same time.

## Prerequisites

In order to take advantage of the cli, you need to have already installed Docker and Node.

### Installing Node

The easiest way to install node on a Mac is with Homebrew

```
brew install node
```

### Installing Docker

See this link: [<https://docs.docker.com/docker-for-mac/install/>]

or use this link if you are still on OSX Yosemite 10.10.5  
[<https://download.docker.com/mac/stable/18950/Docker.dmg>]

## Install the cli

The cli is written in Javascript and is packaged as a runnable npm module. To install or update run:

```
npm install @3share/3share-cli -g
```

This will install the cli globally and it will be accessible from the command line anywhere in your environment.

## How to get started

For each project you want to use the tool for you need to go to tyhe directory for your project (or a directory that will be the base for your project).

```
→ g6 3share-cli init
? What template do you want? author
? What AEM version do you want? 6.5
? What Java JDK version do you want? 8u231
? What environment name would you like to use? g6
? What domain do you want to use, like docker.localhost? g6.localhost
? Do you want sample data installed? No
? What port do you want to access your environment on? 8280
? What port do you want your control panel to be on 8281
? What is the username in your AEM license? If you do not have it go to
https://licensing.adobe.com/ client-name
? What is your AEM license serial number? If you do not have it go to
https://licensing.adobe.com/ xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
? Where is the project located? Use an absolute path otherwise it will be
located in the .3share-cli ./
? Where do you want the Mvnvm directory? The Mvnvm directory is where
Maven files and repos are saved. ../mvnvm
```

## Initialize Project

Create a new directory or go into an existing project directory. Then type this, don't forget the sudo:

```
sudo 3share-cli init
```

Answer a series of questions and now your project is ready for the cli.

---

## Setting Host file for local domains

```
→ sudo 3share-cli hosts
Password:
Initializing hosts file
set /etc/hosts successfully!``
Running as sudo allows the cli to make changes to your hosts file.
---
```

## Build your Containers

Build will create base containers and register them in your local docker repository. These base images can be used by multiple stacks if they have the same needed jdk. These base images will be used to build your stack. This can take some time and produce a lot of output. It's been abbreviated below.

```
→ 3share-cli build
Building 3share/base
Sending build context to Docker daemon 2.577MB
```

```
Step 1/4 : FROM centos:7
----> 9f38484d220f
Step 2/4 : LABEL Vendor="3Share"      Maintainer="Gordon Pike"
----> Using cache
----> ad37d01f7b2e
Step 3/4 : RUN yum -y --setopt=tsflags=nodocs update &&      yum clean all
&&      yum install ansible -y &&      yum install git -y
----> Using cache
----> 163199d1ea97

...

Successfully built 5572bf7d8631
Successfully tagged mvnvm:latest
child process exited with code 0
```

This only need to be run once per stack unless you upgrade.

---

## Bring up Your Environment

```
→ 3share-cli up
Environment Name:xxx
```

This will start your new environment spinning up all of your containers. By default all output from each server goes into the console. You can run it in detached mode by adding `-d` or `--detach`

```
→ 3share-cli up --detach
Environment Name:xxx
```

This will run the stack in the background.

---

## Access Traefik Dashboard

```
3share-cli dashboard
```

This will open your default browser and point it to the dashboard for Traefik. You can see all backed and frontends of your environment.

---

## Navigating to Services

Each of the services in your stack can be accessed from the domain. Services at the moment include author, publish, dispatcher.

To reach the author type the following:

```
3share-cli open author
```

Your default browser will open and be directed to the public url for the author.

---

## Bring Down Your Environment

There is no need to bring down the containers as long as you have enough room and memory in docker. If you need to bring them down then type:

```
3share-cli down
```

This will stop the containers.

---

## Build Container for compiling project files

Along with the other servers in your stack there is also a container preconfigured with mvnvm (<http://mvnvm.org/>) a Maven Version Manager. At the base of your project create a mvnvm.properties file that has the version of Maven you want to use for your project. The first time you build using the container it will download the appropriate version of Maven.

```
$ echo "mvn_version=3.0.5" > mvnvm.properties  
$ git add mvnvm.properties
```

To use your build container run the dev command in a terminal.

```
3share-cli dev
```

This will bring up a terminal inside the build docker container. The container has access to your project files under /project and can deploy to your author because it has an alias with a command like:

```
mvn -PautoInstallPackage -Padobe-public clean install -  
Dmaven.test.skip=true -Dcrx.host=author -Dcrx.port=4502 -e
```

The package would be installed on your Author server and the 4502 port is the internal port for author in your stack.

---