**Deployment of NPM-repositories with Artipie.**

In this article I would like to demonstrate how it is simple and quick to deploy own NMP-repository server of NPM-packages with help of Artipie.

**A few words about Artipie**

[**Artipie**](https://github.com/artipie) is a free open source binary artifact management tool project under a MIT license.

Artipie is quickly growing project that was born in 2020 and currently supports plenty repository types:

* NPM - for storing and sharing of Node JS packages.
* Docker - Docker registry for images.
* RPM - repository of .rpm-files for RHEL, CentOs, Fedora, PCLinuxOS, AlmaLinux, openSUSE, OpenMandriva, Oracle Linux, etc.
* Debian - repository packages for Debian-based Linux distros(Debian, Mint, Ubuntu, MX Linux, Raspberry Pi OS, Parrot OS, etc).
* Go - storage of Go packages.
* Maven - Java, Kotlin, Groovy, Scala, Clojure artifacts of such types as .jar, .war, .klib, etc.
* PyPI - Python packages index.
* Anaconda - packages for data science for Python, R, Lua, C, C++ and etc. languages.
* HexPM - for storing and sharing packages for Elixir and Erlang languages.
* Gem - hosting service of RubyGem for Ruby language.
* Helm - Helm charts repository.
* NuGet - .NET package hosting service.
* Composer - PHP source packages.
* Binary(files) storage - for hosting any types of files.

Artipie stores artifacts in storages and provides range of well-known storage types:

* File system storage
* Amazon S3 storage
* Redis storage

Additionally, it provides ability to write custom storage type.

Artipie is fully asynchronous internally implementing asynchronous file and network operations that allows to organize storage for high load system of artifacts.

Artipie is secure server that allows to grand permissions for users and also allows to organize groups of users with granted permissions.

Artipie allows to store artifact either all artifacts in one flat-repository or separate set of repositories.

For now, that’s all about Artipie and let’s continue with deployment of NPM-repository. If you are interested in more information about Artipie please visit its [github](https://github.com/artipie/artipie) and [wiki](https://github.com/artipie/artipie/wiki) pages.

**Starting of NPM-repository.**

So, I am going to use Docker-engine and Docker compose as deployment environment to deploy NPM-repository based on Artipie. Of course, there is ability to run Atripie from Jar-file but it is more simply to use Docker compose for it.

So you should be sure that [Docker-engine](https://docs.docker.com/get-docker/) and [Docker compose](https://docs.docker.com/compose/install/) are already installed on your PC before continue deployment.

First of all we are required Docker compose configuration file of Artipie, so we need to get ‘docker-compose.yaml’ file and it can be obtained from one of ways:

* download from [Artipie’s github page](https://github.com/artipie/artipie/blob/master/docker-compose.yaml)
* clone [Artipie git repository](https://github.com/artipie/artipie) by using [git](https://git-scm.com/) and [clone](https://git-scm.com/docs/git-clone) command

On my PC I am going to use one single folder “C:\npm-repository\” where be stored all necessary configuration files that will be necessary to launch NPM-repository and also this folder will keep all NPM-artifacts.

So now I put downloaded ‘docker-compose.yaml’ file into “C:\npm-repository\” folder:

|  |
| --- |
| c:\npm-repository>ls .  docker-compose.yaml |

The content of ‘docker-compose.yaml’ file is simple enough and defines backend and frontend docker applications of Artipie:

|  |
| --- |
| version: "3.3"  services:  artipie:  image: artipie/artipie:latest  container\_name: artipie  restart: unless-stopped  environment:  - ARTIPIE\_USER\_NAME=artipie  - ARTIPIE\_USER\_PASS=artipie  networks:  - artipie-net  ports:  - "8081:8080"  volumes:  # change /usr/local/artipie to any convenient location, this is the config folder  - /usr/local/artipie:/var/artipie/repo  front:  image: artipie/front:latest  container\_name: front  restart: unless-stopped  networks:  - artipie-net  environment:  - ARTIPIE\_USER\_NAME=artipie  - ARTIPIE\_USER\_PASS=artipie  - TKN\_KEY=abc123  volumes:  # change /usr/local/artipie to any convenient location, this is the config folder  - /usr/local/artipie:/var/artipie/repo  ports:  - "8080:8080"  networks:  artipie-net:  driver: bridge |

I am going to store all Artipie’s-files in one place as separate sub-folder “c:\npm-repository\artipie\”

and I suppose to use “c:\npm-repository\artipie\” folder as mounted Docker-volume so a little change of ‘docker-compose.yaml’ file is required to do mounting of volume:

(See highlighted text with added changes)

|  |
| --- |
| version: "3.3"  services:  artipie:  image: artipie/artipie:latest  container\_name: artipie  restart: unless-stopped  environment:  - ARTIPIE\_USER\_NAME=artipie  - ARTIPIE\_USER\_PASS=artipie  networks:  - artipie-net  ports:  - "8081:8080"  volumes:  # change /usr/local/artipie to any convenient location, this is the config folder  - C:\npm-repository\artipie:/var/artipie/repo  front:  image: artipie/front:latest  container\_name: front  restart: unless-stopped  networks:  - artipie-net  environment:  - ARTIPIE\_USER\_NAME=artipie  - ARTIPIE\_USER\_PASS=artipie  - TKN\_KEY=abc123  volumes:  # change /usr/local/artipie to any convenient location, this is the config folder  - C:\npm-repository\artipie:/var/artipie/repo  ports:  - "8080:8080"  networks:  artipie-net:  driver: bridge |

Now we are ready to launch Artipie in Docker compose.

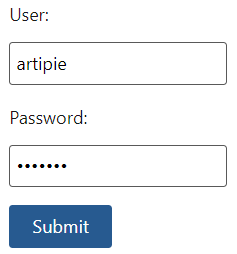
So change current working directory into “C:\npm-repository” and run command:

|  |
| --- |
| docker-compose up |

This command starts Artipie server and Artipie-dashboard.

Let’s open Artipie-dashboard in browser: <http://localhost:8080/dashboard/artipie>

In prompted login form please enter:

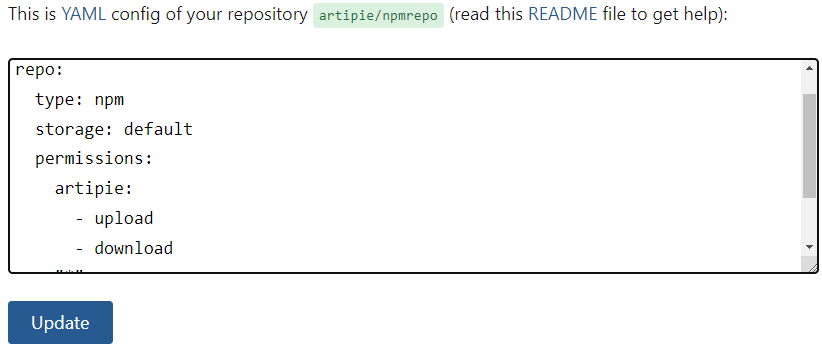


‘artipie’ as user and ‘artipie’ as password and click ‘Submit’-button.

On opened page please enter ‘npmrepo’ as repository name and choose ‘NPM’-type and click ‘Add’-button:



Now click ‘Update’-button to save configuration of new repository on Artipie-server:



My congratulations! You have just created NPM-repository!

**Publishing of package**

Now we have running NPM-repository and this is time to create npm-package for publishing.

Let’s create simple node js file in separate folder, I create folder ‘C:\package\’ and put there two files: index.js and package.json.

The content of index.js:

|  |
| --- |
| exports.greeting = function () {  console.log("Hello world!");  }; |

The content of package.json:

|  |
| --- |
| {  "name": "greeting",  "version": "1.0.0",  "description": "Greeting",  "main": "index.js",  "author": "",  "license": "ISC"  } |