Day 1 Task:

- *Created Docker images*
- *Pushed images to Docker Hub*
- *Deployed Nginx using Docker*

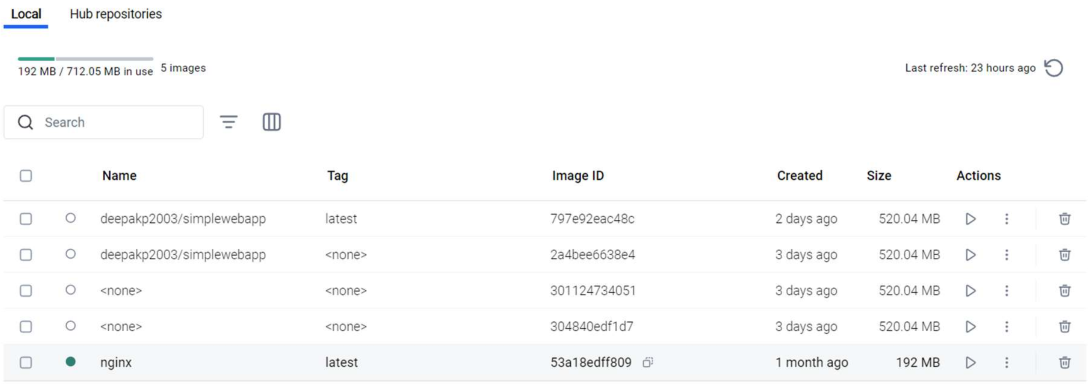
Software Development Life Cycle

- *SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step.*
- *The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user's requirements.*
- *SDLC in software engineering models outlines the plan for each stage.*
- *Each stage of the software development model - perform its task efficiently to deliver the software at a low cost.*
- *Within a given time frame - meets users requirements.*

Phases of SDLC

- ***Planning:** Defines project goals, requirements, constraints, and timelines.*
- ***Analysis:** Analyzes the requirements for the software Design: Creates the software's architecture and design.*
- ***Development:** Transforms the design into a functional application.*
- *Testing: Tests the software for performance, usability, security, and more.*
- ***Deployment:** Releases the software to end users.*
- ***Maintenance:** Assesses the software and makes updates as needed.*

Docker Images

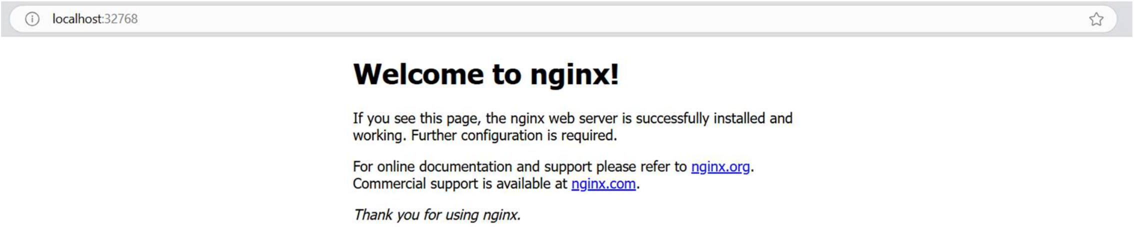


192 MB / 712.05 MB in use 5 images

Last refresh: 23 hours ago

□		Name	Tag	Image ID	Created	Size	Actions
□	○	deepakp2003/simplewebapp	latest	797e92eac48c	2 days ago	520.04 MB	▷ ⋮
□	○	deepakp2003/simplewebapp	<none>	2a4bee6638e4	3 days ago	520.04 MB	▷ ⋮
□	○	<none>	<none>	301124734051	3 days ago	520.04 MB	▷ ⋮
□	○	<none>	<none>	304840edf1d7	3 days ago	520.04 MB	▷ ⋮
□	●	nginx	latest	53a18edff809	1 month ago	192 MB	▷ ⋮

Nginx output



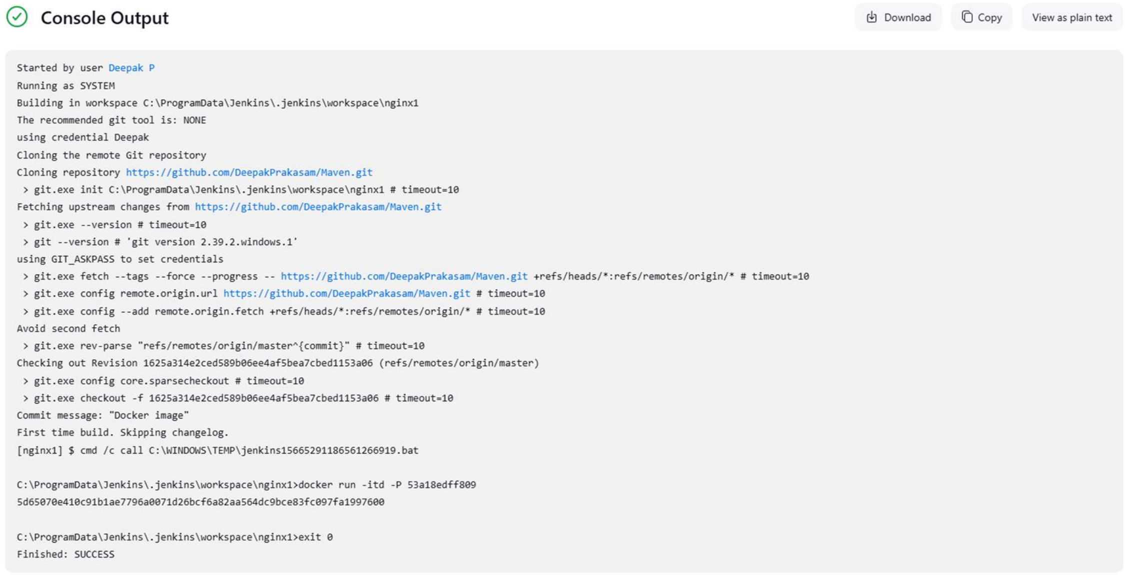
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Console output nginx



```
Started by user Deepak P
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\nginx1
The recommended git tool is: NONE
using credential Deepak
Cloning the remote Git repository
Cloning repository https://github.com/DeepakPrakasam/Maven.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\nginx1 # timeout=10
Fetching upstream changes from https://github.com/DeepakPrakasam/Maven.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.39.2.windows.1'
using GIT_ASKPASS to set credentials
 > git.exe fetch --tags --force --progress -- https://github.com/DeepakPrakasam/Maven.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/DeepakPrakasam/Maven.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 1625a314e2ced589b06ee4af5bea7cbed1153a06 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 1625a314e2ced589b06ee4af5bea7cbed1153a06 # timeout=10
Commit message: "Docker image"
First time build. Skipping changelog.
[nginx1] $ cmd /c call C:\WINDOWS\TEMP\jenkins15665291186561266919.bat

C:\ProgramData\Jenkins\.jenkins\workspace\nginx1>docker run -itd -P 53a18edff809
5d65070e410c91b1ae7796a0071d26bcf6a82aa564dc9bce83fc097fa1997600

C:\ProgramData\Jenkins\.jenkins\workspace\nginx1>exit 0
Finished: SUCCESS
```