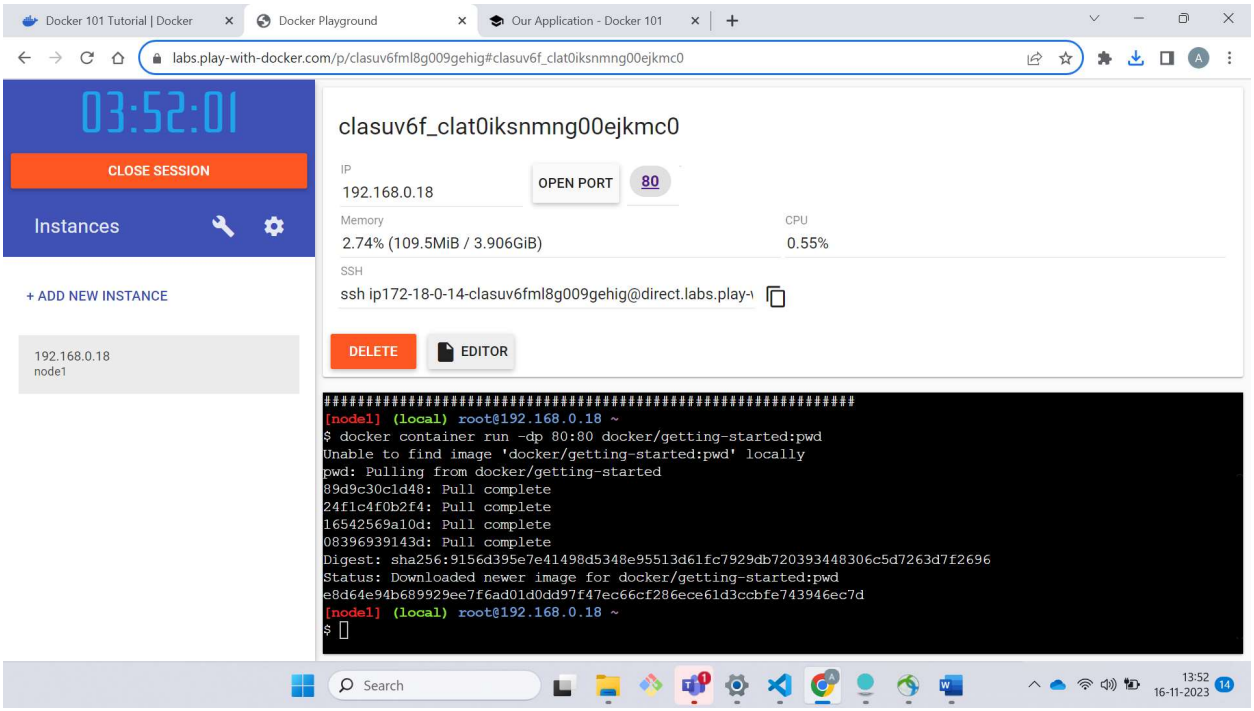
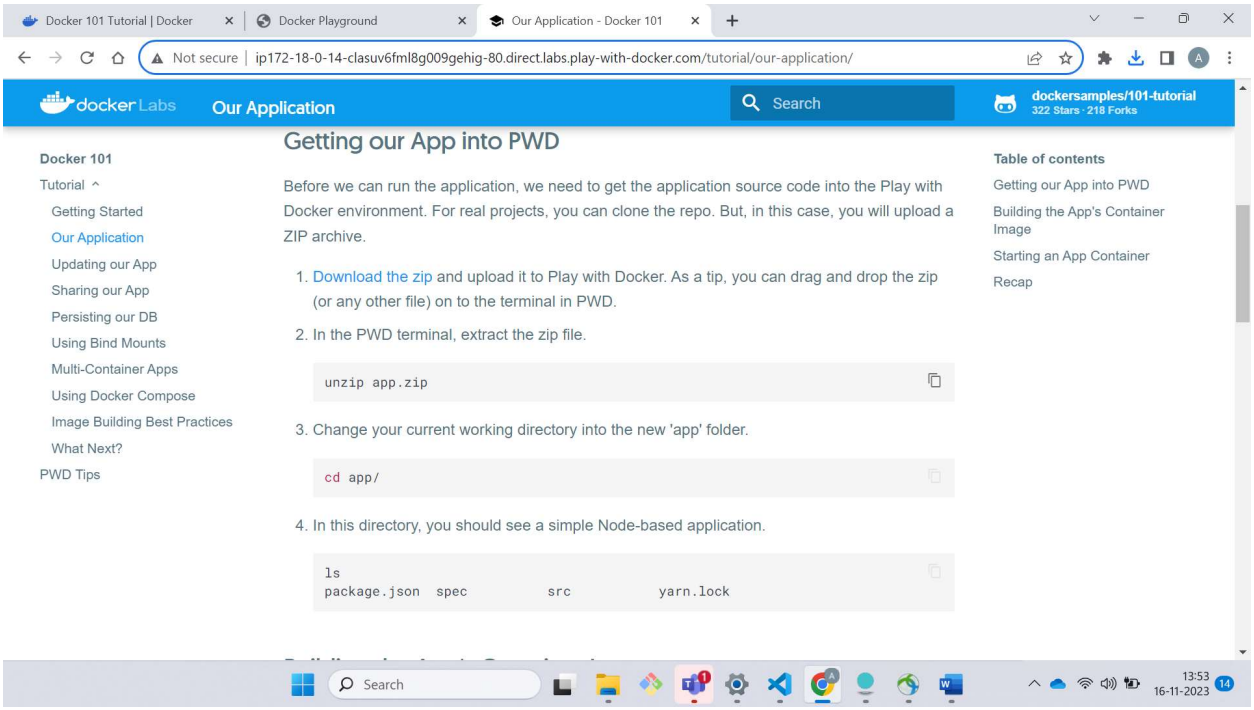
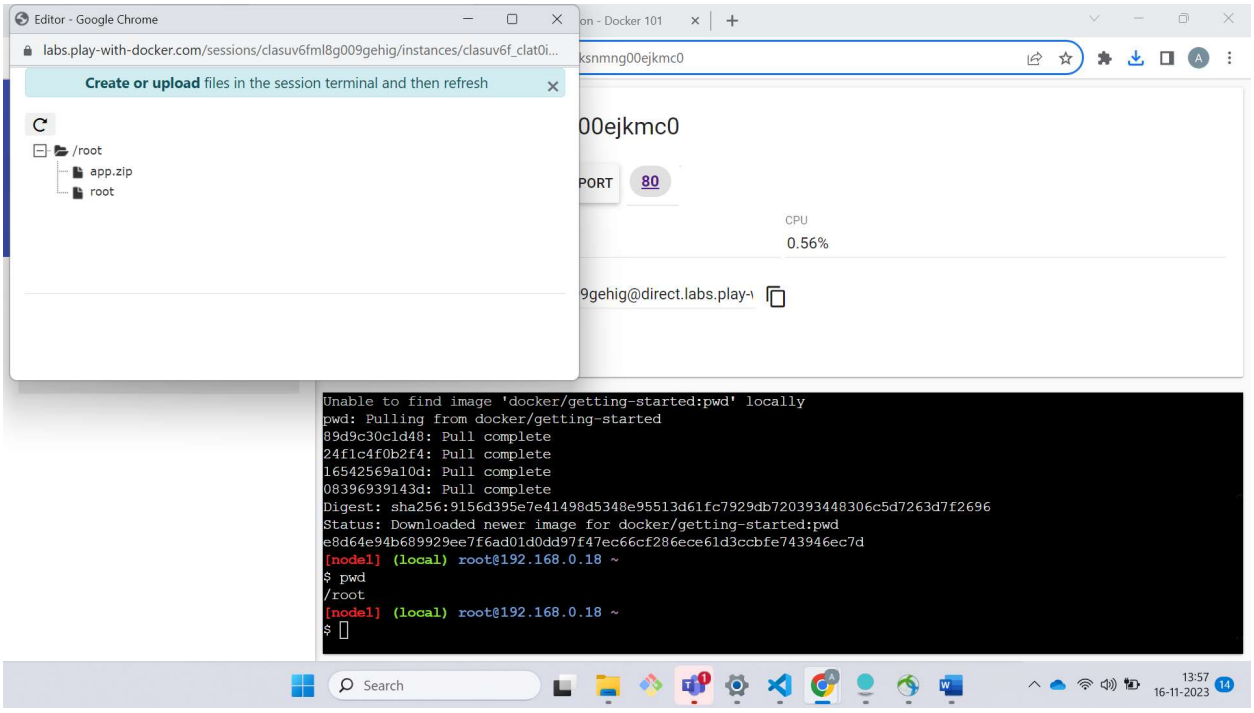
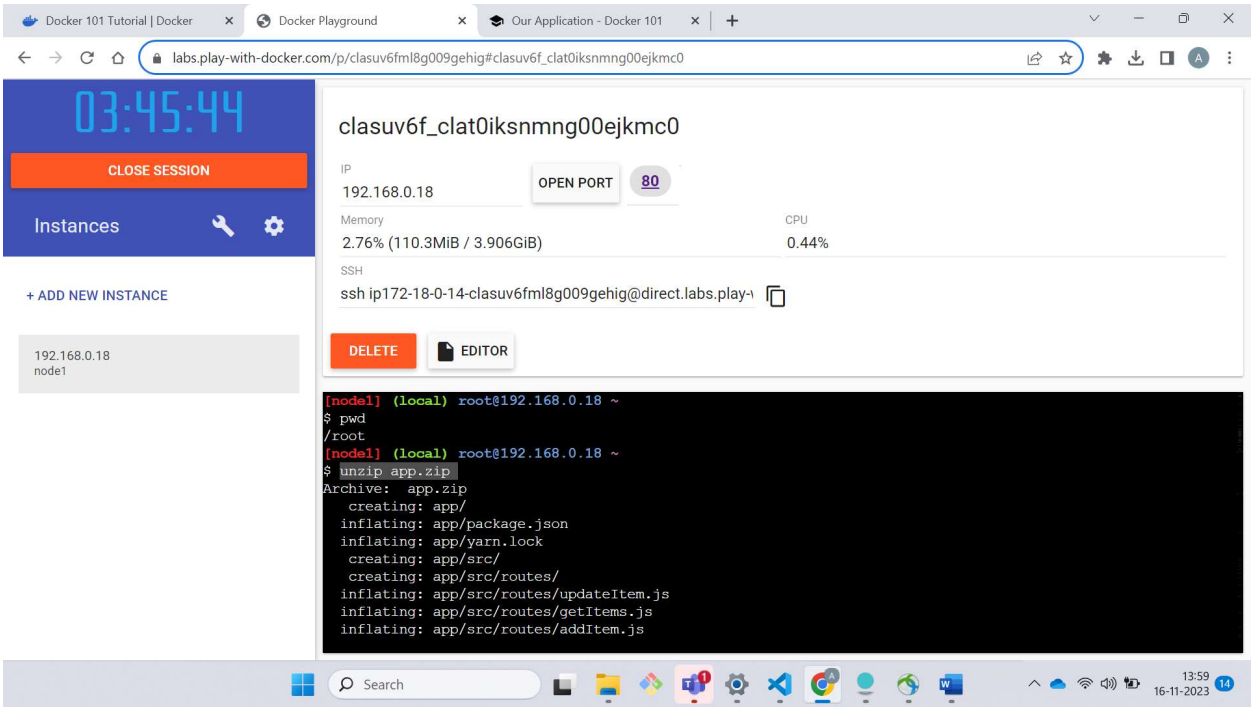
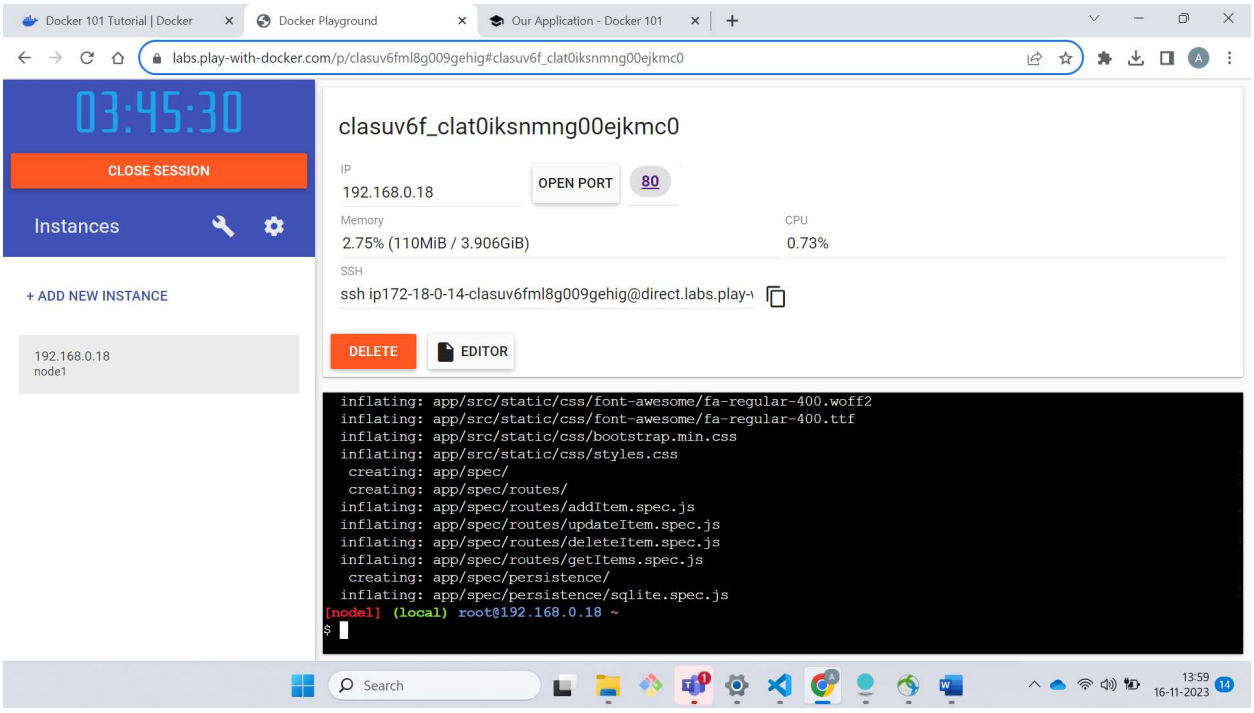
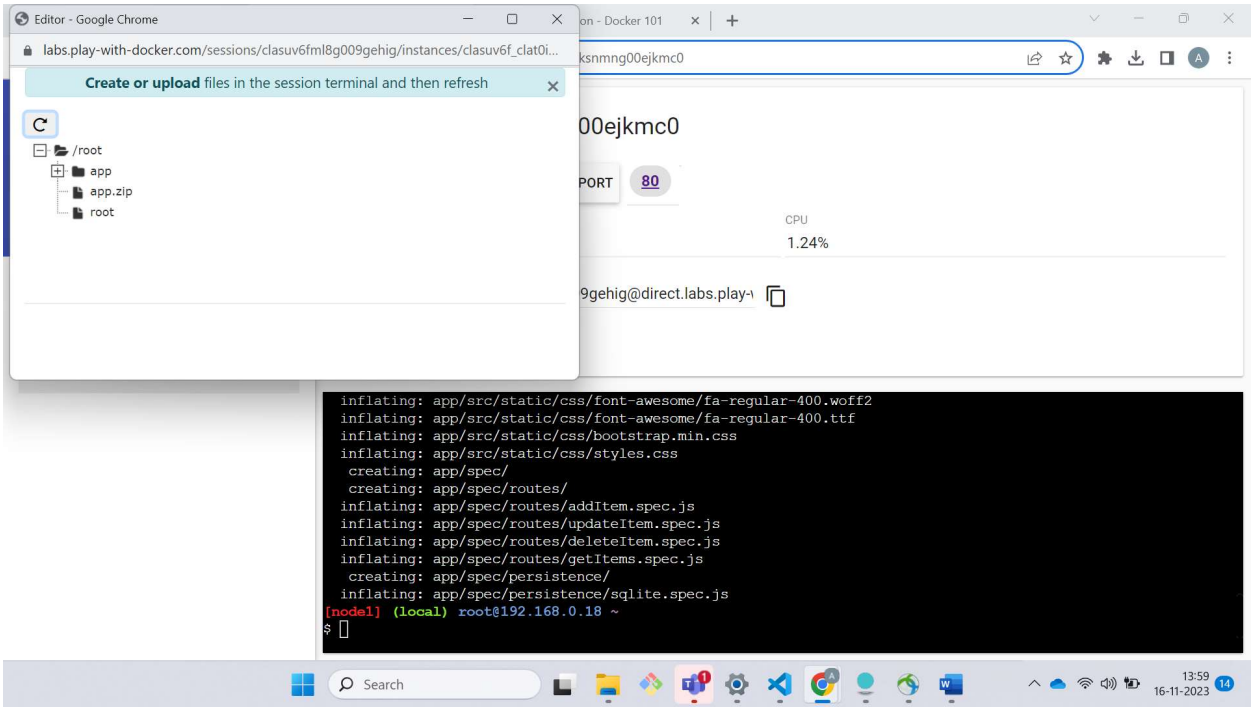

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clat0iksnmng00ejkmc0
03:52:01
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clat0iksnmng00ejkmc0
IP
192.168.0.18
OPEN PORT
80
Memory
2.74% (109.5MiB / 3.906GiB)
CPU
0.55%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node1] (local) root@192.168.0.18 ~
$ docker container run -dp 80:80 docker/getting-started:pwd
Unable to find image 'docker/getting-started:pwd' locally
pwd: Pulling from docker/getting-started
89d9c30c1d48: Pull complete
24f1c4f0b2f4: Pull complete
16542569a10d: Pull complete
08396939143d: Pull complete
Digest: sha256:9156d395e7e41498d5348e95513d61fc7929db720393448306c5d7263d7f2696
Status: Downloaded newer image for docker/getting-started:pwd
e8d64e94b689929ee7f6ad01d0dd97f47ec66cf286ece61d3ccbfe743946ec7d
[node1] (local) root@192.168.0.18 ~
$
Search
13:52
16-11-2023

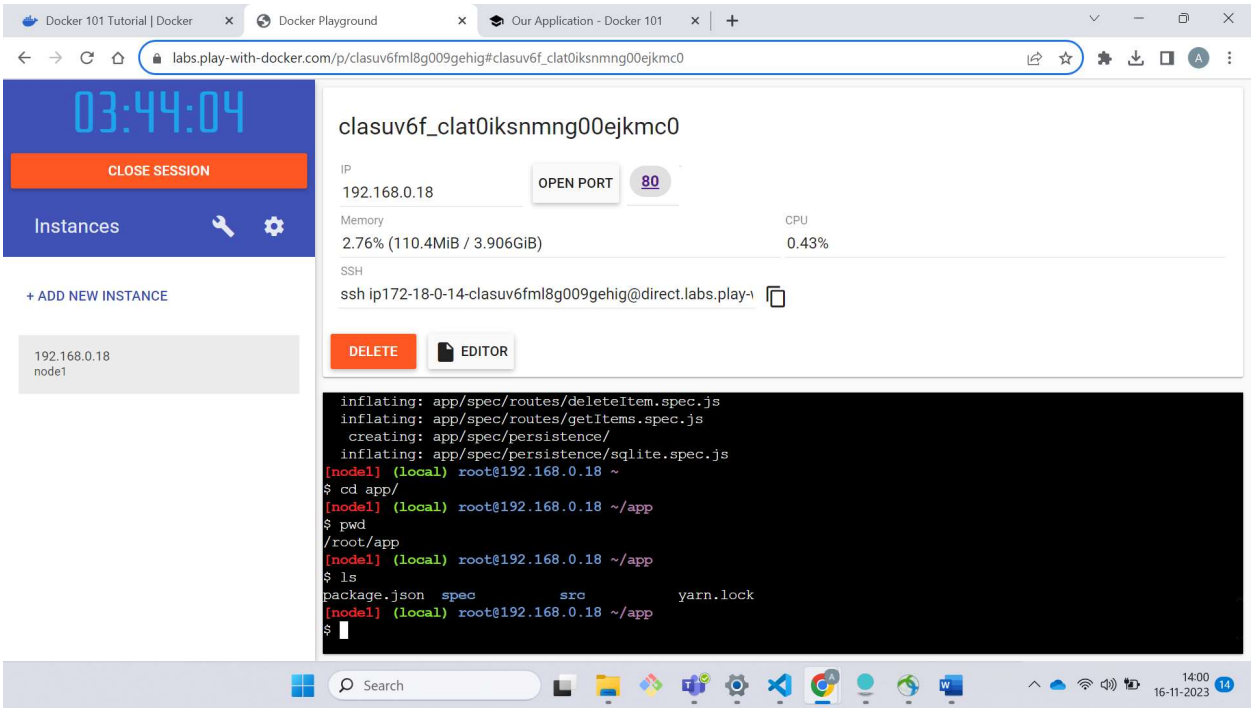

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
Not secure | ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/our-application/
docker Labs
Our Application
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
Getting our App into PWD
Before we can run the application, we need to get the application source code into the Play with Docker environment. For real projects, you can clone the repo. But, in this case, you will upload a ZIP archive.
1. Download the zip and upload it to Play with Docker. As a tip, you can drag and drop the zip (or any other file) on to the terminal in PWD.
2. In the PWD terminal, extract the zip file.
unzip app.zip
3. Change your current working directory into the new 'app' folder.
cd app/
4. In this directory, you should see a simple Node-based application.
ls
package.json spec src yarn.lock
Table of contents
Getting our App into PWD
Building the App's Container Image
Starting an App Container
Recap
Search
13:53
16-11-2023

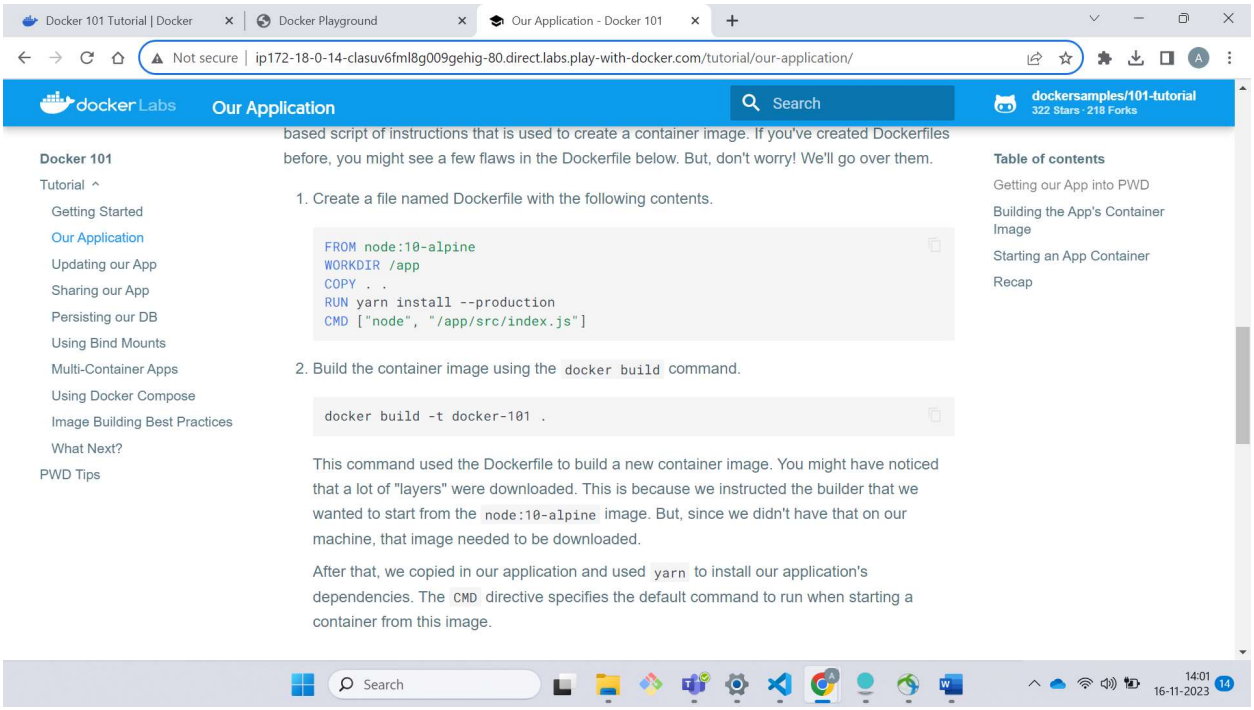

Editor - Google Chrome
labs.play-with-docker.com/sessions/clasuv6fml8g009gehig/instances/clasuv6f_clat0i...
Create or upload files in the session terminal and then refresh
/root
app.zip
root
00ejkmc0
PORT
80
CPU
0.56%
9gehig@direct.labs.play-
Unable to find image 'docker/getting-started:pwd' locally
pwd: Pulling from docker/getting-started
89d9c30c1d48: Pull complete
24f1c4f0b2f4: Pull complete
16542569a10d: Pull complete
08396939143d: Pull complete
Digest: sha256:9156d395e7e41498d5348e95513d61fc7929db720393448306c5d7263d7f2696
Status: Downloaded newer image for docker/getting-started:pwd
e8d64e94b689929ee7f6ad01d0dd97f47ec66cf286ece61d3ccbfe743946ec7d
[node1] (local) root@192.168.0.18 ~
$ pwd
/root
[node1] (local) root@192.168.0.18 ~
$
Search
13:57
16-11-2023

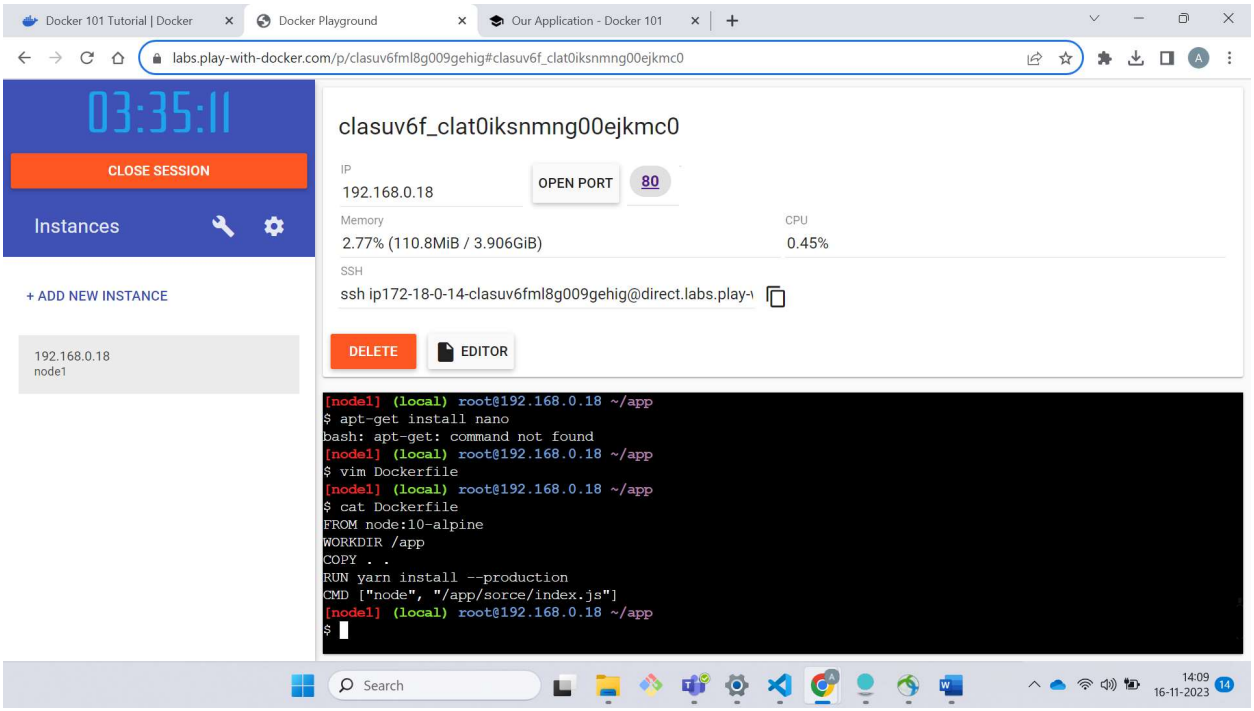

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clat0iksnmng00ejkmc0
03:45:44
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clat0iksnmng00ejkmc0
IP
192.168.0.18
OPEN PORT
80
Memory
2.76% (110.3MiB / 3.906GiB)
CPU
0.44%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node1] (local) root@192.168.0.18 ~
$ pwd
/root
[node1] (local) root@192.168.0.18 ~
$ unzip app.zip
Archive: app.zip
creating: app/
inflating: app/package.json
inflating: app/yarn.lock
creating: app/src/
creating: app/src/routes/
inflating: app/src/routes/updateItem.js
inflating: app/src/routes/getItems.js
inflating: app/src/routes/addItem.js
Search
13:59
16-11-2023

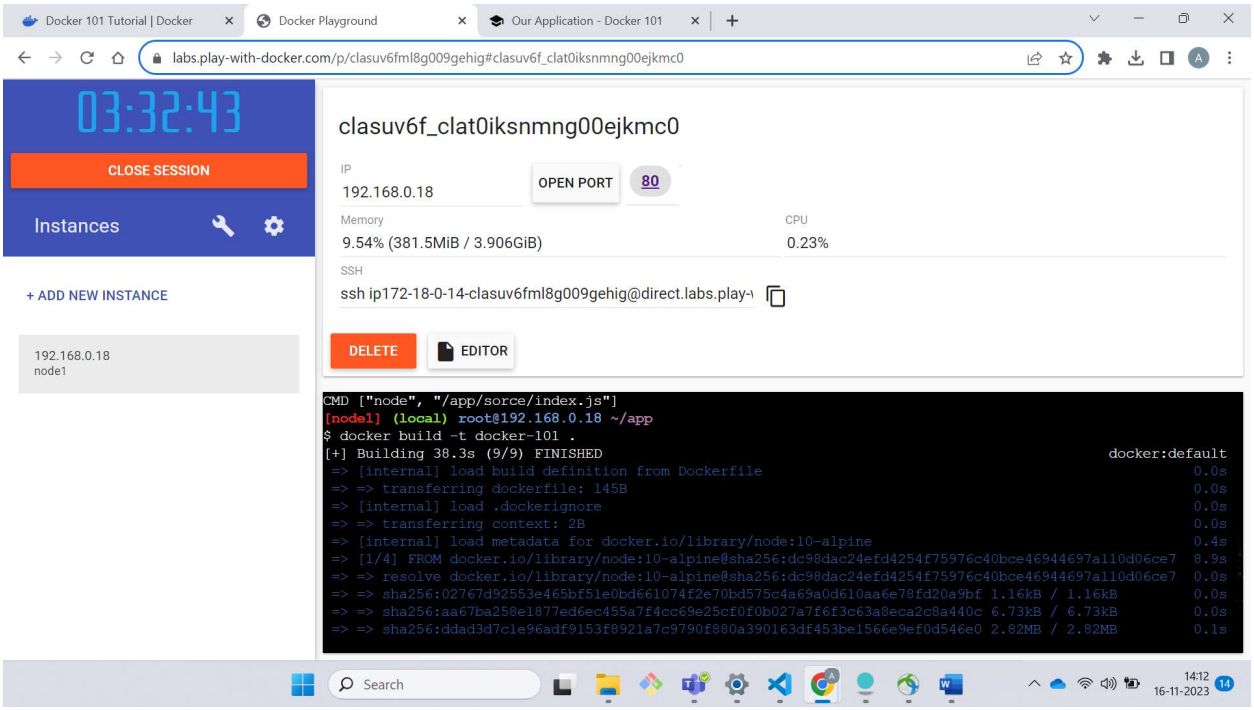

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clat0iksnmng00ejkmc0
03:45:30
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clat0iksnmng00ejkmc0
IP
192.168.0.18
OPEN PORT
80
Memory
2.75% (110MiB / 3.906GiB)
CPU
0.73%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
inflating: app/src/static/css/font-awesome/fa-regular-400.woff2
inflating: app/src/static/css/font-awesome/fa-regular-400.ttf
inflating: app/src/static/css/bootstrap.min.css
inflating: app/src/static/css/styles.css
creating: app/spec/
creating: app/spec/routes/
inflating: app/spec/routes/addItem.spec.js
inflating: app/spec/routes/updateItem.spec.js
inflating: app/spec/routes/deleteItem.spec.js
inflating: app/spec/routes/getItems.spec.js
creating: app/spec/persistence/
inflating: app/spec/persistence/sqlite.spec.js
[node1] (local) root@192.168.0.18 ~
$
Search
13:59
16-11-2023

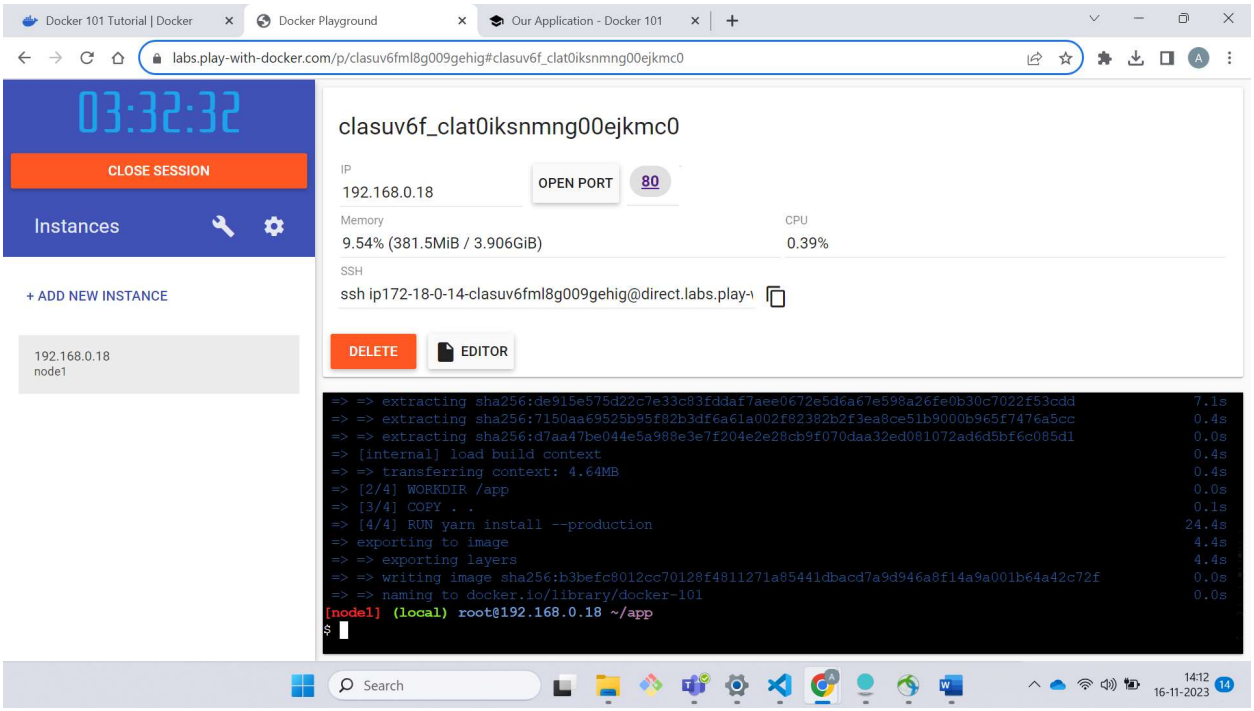

Editor - Google Chrome
labs.play-with-docker.com/sessions/clasuv6fml8g009gehig/instances/clasuv6f_clat0i...
Create or upload files in the session terminal and then refresh
/root
app
app.zip
root
00ejkmc0
PORT
80
CPU
1.24%
9gehig@direct.labs.play-
inflating: app/src/static/css/font-awesome/fa-regular-400.woff2
inflating: app/src/static/css/font-awesome/fa-regular-400.ttf
inflating: app/src/static/css/bootstrap.min.css
inflating: app/src/static/css/styles.css
creating: app/spec/
creating: app/spec/routes/
inflating: app/spec/routes/addItem.spec.js
inflating: app/spec/routes/updateItem.spec.js
inflating: app/spec/routes/deleteItem.spec.js
inflating: app/spec/routes/getItems.spec.js
creating: app/spec/persistence/
inflating: app/spec/persistence/sqlite.spec.js
[node1] (local) root@192.168.0.18 ~
$
Search
13:59
16-11-2023

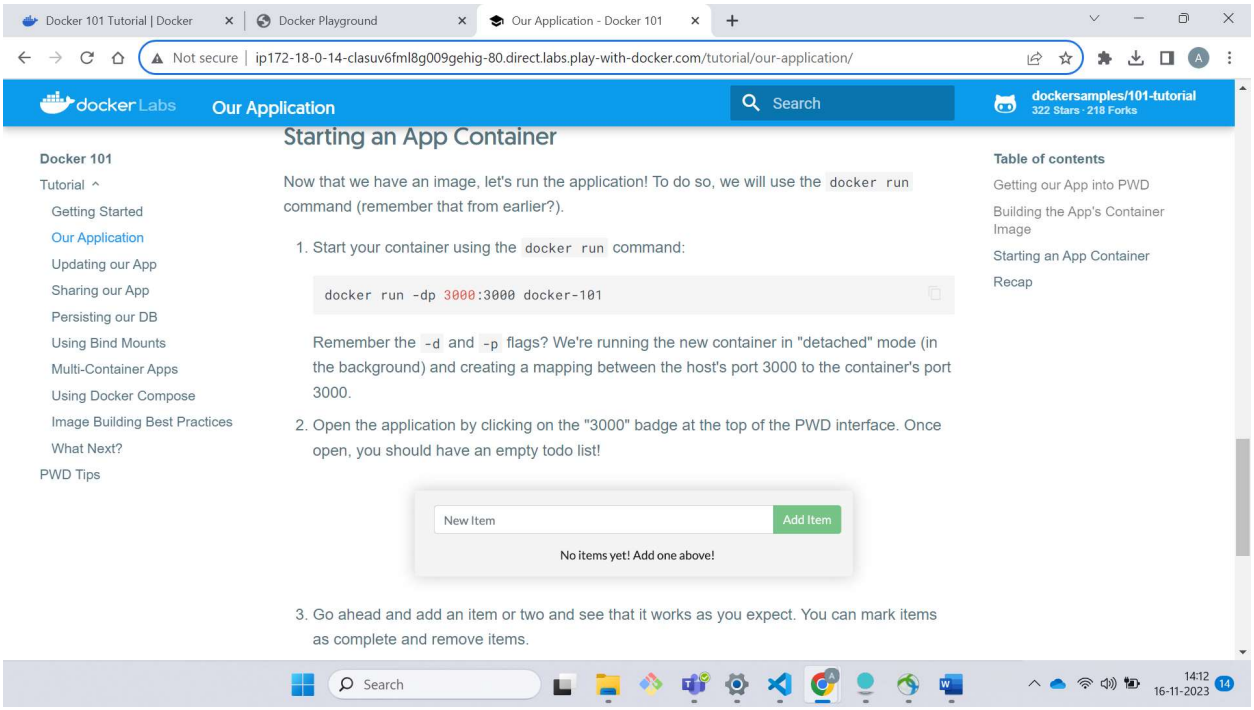

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clat0iksnmng00ejkmc0
03:44:04
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clat0iksnmng00ejkmc0
IP
192.168.0.18
OPEN PORT
80
Memory
2.76% (110.4MiB / 3.906GiB)
CPU
0.43%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
inflating: app/spec/routes/deleteItem.spec.js
inflating: app/spec/routes/getItems.spec.js
creating: app/spec/persistence/
inflating: app/spec/persistence/sqlite.spec.js
[node1] (local) root@192.168.0.18 ~
$ cd app/
[node1] (local) root@192.168.0.18 ~/app
$ pwd
/root/app
[node1] (local) root@192.168.0.18 ~/app
$ ls
package.json spec src yarn.lock
[node1] (local) root@192.168.0.18 ~/app
$
Search
14:00
16-11-2023

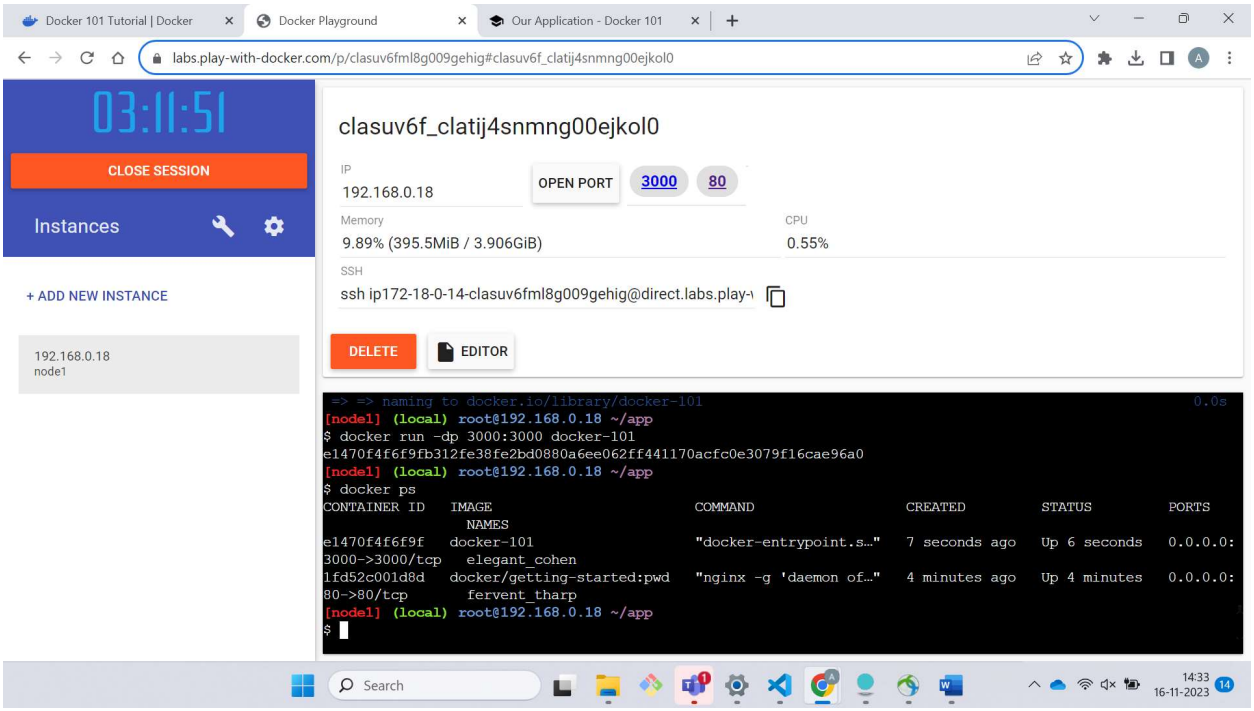

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
Not secure | ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/our-application/
docker Labs
Our Application
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
based script of instructions that is used to create a container image. If you've created Dockerfiles before, you might see a few flaws in the Dockerfile below. But, don't worry! We'll go over them.
1. Create a file named Dockerfile with the following contents.
FROM node:10-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "/app/src/index.js"]
2. Build the container image using the docker build command.
docker build -t docker-101 .
This command used the Dockerfile to build a new container image. You might have noticed that a lot of "layers" were downloaded. This is because we instructed the builder that we wanted to start from the node:10-alpine image. But, since we didn't have that on our machine, that image needed to be downloaded.
After that, we copied in our application and used yarn to install our application's dependencies. The CMD directive specifies the default command to run when starting a container from this image.
Table of contents
Getting our App into PWD
Building the App's Container Image
Starting an App Container
Recap
Search
14:01
16-11-2023

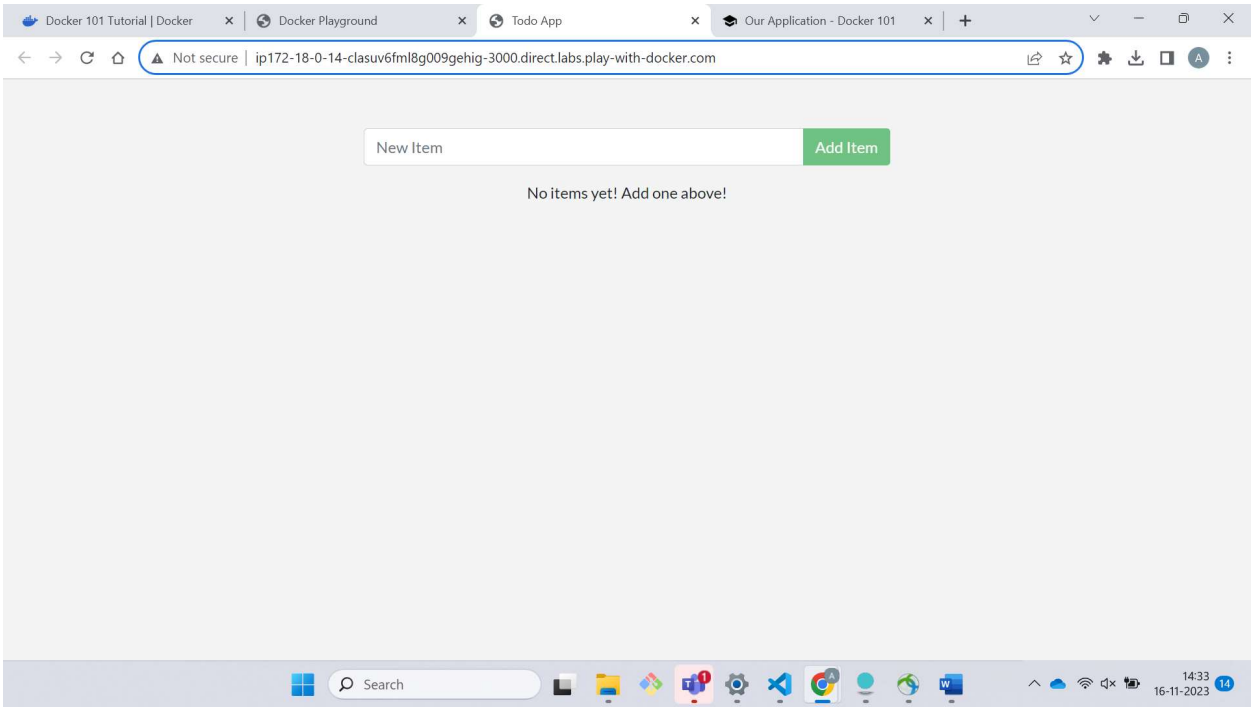

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clat0iksnmng00ejkmc0
03:35:11
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clat0iksnmng00ejkmc0
IP
192.168.0.18
OPEN PORT
80
Memory
2.77% (110.8MiB / 3.906GiB)
CPU
0.45%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node1] (local) root@192.168.0.18 ~/app
$ apt-get install nano
bash: apt-get: command not found
[node1] (local) root@192.168.0.18 ~/app
$ vim Dockerfile
[node1] (local) root@192.168.0.18 ~/app
$ cat Dockerfile
FROM node:10-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "/app/sorce/index.js"]
[node1] (local) root@192.168.0.18 ~/app
$
Search
14:09
16-11-2023

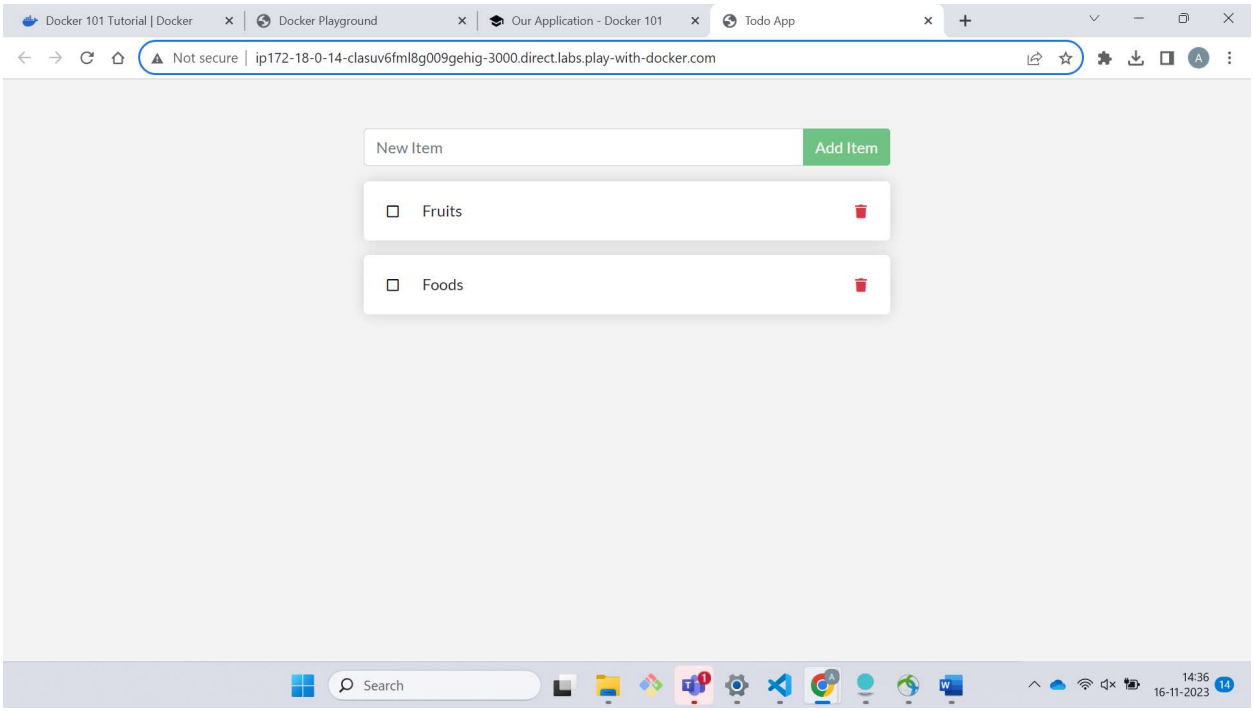

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clat0iksnmng00ejkmc0
03:32:43
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clat0iksnmng00ejkmc0
IP
192.168.0.18
OPEN PORT
80
Memory
9.54% (381.5MiB / 3.906GiB)
CPU
0.23%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
CMD ["node", "/app/sorce/index.js"]
[node1] (local) root@192.168.0.18 ~/app
$ docker build -t docker-101 .
[+] Building 38.3s (9/9) FINISHED docker:default
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 145B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/node:10-alpine 0.4s
=> [1/4] FROM docker.io/library/node:10-alpine@sha256:dc98dac24efd4254f75976c40bce46944697a110d06ce7 8.9s
=> => resolve docker.io/library/node:10-alpine@sha256:dc98dac24efd4254f75976c40bce46944697a110d06ce7 0.0s
=> => sha256:02767d92553e465bf51e0bd661074f2e70bd575c4a69a0d610aa6e78fd20a9bf 1.16kB / 1.16kB 0.0s
=> => sha256:aa67ba258e1877ed6ec455a7f4cc69e25cf0f0b027a7f6f3c63a8eca2c8a440c 6.73kB / 6.73kB 0.0s
=> => sha256:ddad3d7c1e96adf9153f8921a7c9790f880a390163df453be1566e9ef0d546e0 2.82MB / 2.82MB 0.1s
Search
14:12
16-11-2023

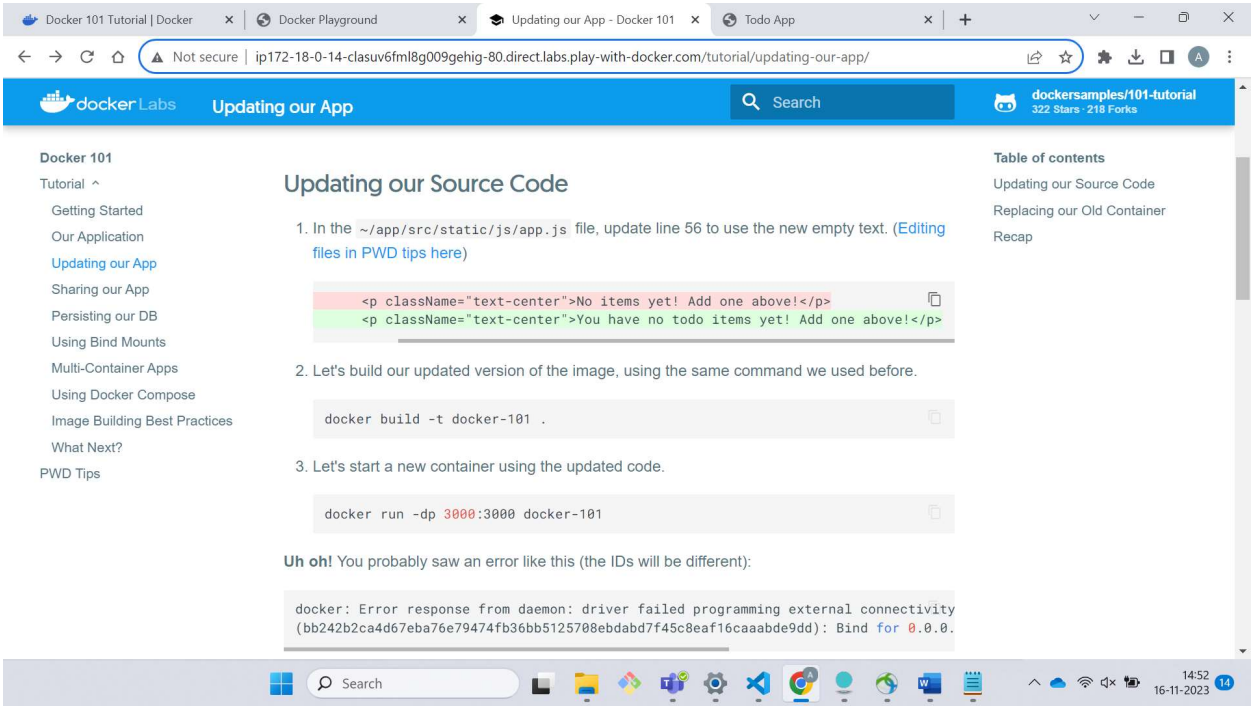

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clat0iksnmng00ejkmc0
03:32:32
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clat0iksnmng00ejkmc0
IP
192.168.0.18
OPEN PORT
80
Memory
9.54% (381.5MiB / 3.906GiB)
CPU
0.39%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
=> => extracting sha256:de915e575d22c7e33c83fddaf7aee0672e5d6a67e598a26fe0b30c7022f53cdd 7.1s
=> => extracting sha256:7150aa69525b95f82b3df6a61a002f82382b2f3ea8ce51b9000b965f7476a5cc 0.4s
=> => extracting sha256:d7aa47be044e5a988e3e7f204e2e28cb9f070daa32ed081072ad6d5bf6c085d1 0.0s
=> [internal] load build context 0.4s
=> => transferring context: 4.64MB 0.4s
=> [2/4] WORKDIR /app 0.0s
=> [3/4] COPY . . 0.1s
=> [4/4] RUN yarn install --production 24.4s
=> exporting to image 4.4s
=> => exporting layers 4.4s
=> => writing image sha256:b3befc8012cc70128f4811271a85441dbacd7a9d946a8f14a9a001b64a42c72f 0.0s
=> => naming to docker.io/library/docker-101 0.0s
[node1] (local) root@192.168.0.18 ~/app
$
Search
14:12
16-11-2023

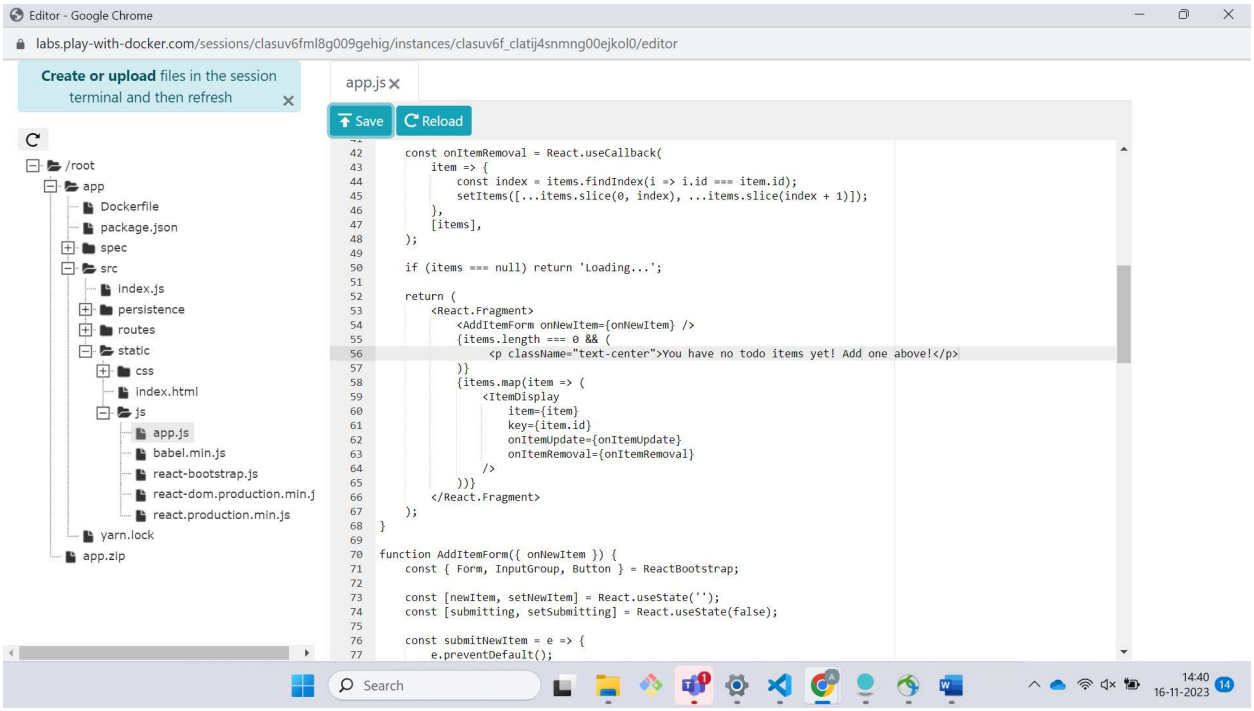

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
Not secure | ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/our-application/
docker Labs
Our Application
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
Starting an App Container
Now that we have an image, let's run the application! To do so, we will use the docker run command (remember that from earlier?).
1. Start your container using the docker run command:
docker run -dp 3000:3000 docker-101
Remember the -d and -p flags? We're running the new container in "detached" mode (in the background) and creating a mapping between the host's port 3000 to the container's port 3000.
2. Open the application by clicking on the "3000" badge at the top of the PWD interface. Once open, you should have an empty todo list!
New Item
Add Item
No items yet! Add one above!
3. Go ahead and add an item or two and see that it works as you expect. You can mark items as complete and remove items.
Table of contents
Getting our App into PWD
Building the App's Container Image
Starting an App Container
Recap
Search
14:12
16-11-2023

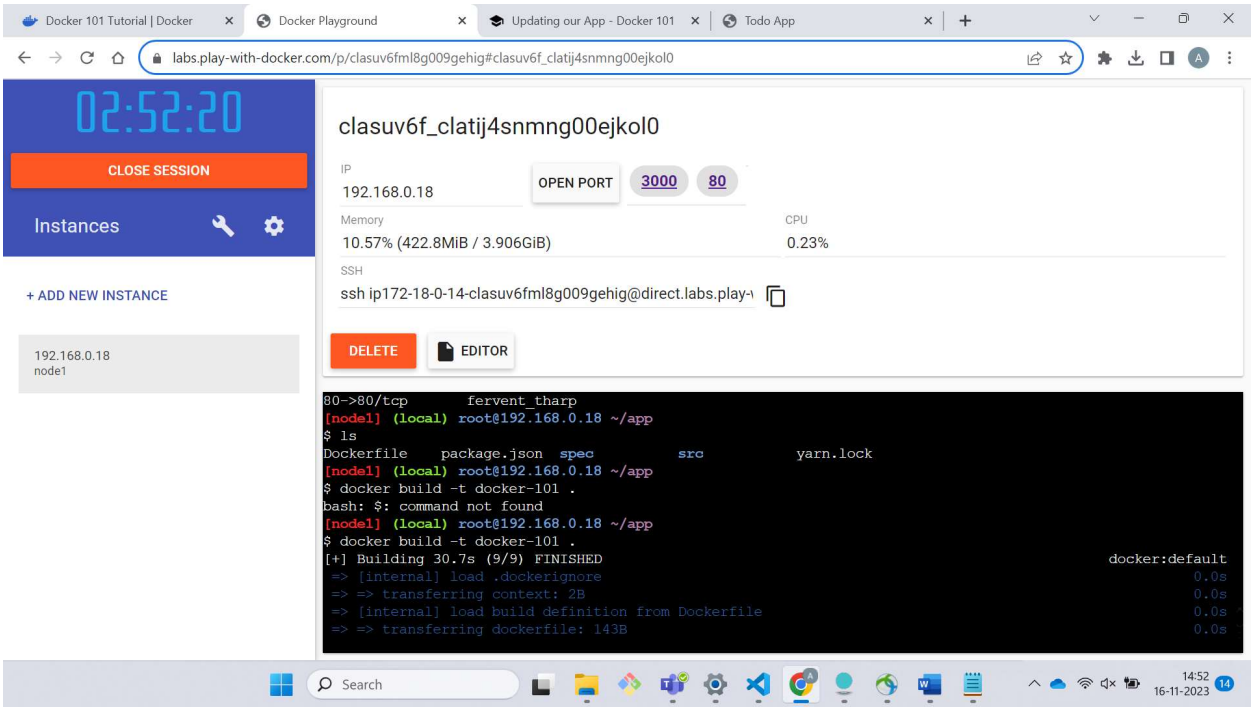

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clatij4snmng00ejkol0
03:11:51
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clatij4snmng00ejkol0
IP
192.168.0.18
OPEN PORT
3000
80
Memory
9.89% (395.5MiB / 3.906GiB)
CPU
0.55%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
=> => naming to docker.io/library/docker-101 0.0s
[node1] (local) root@192.168.0.18 ~/app
$ docker run -dp 3000:3000 docker-101
e1470f4f6f9fb312fe38fe2bd0880a6ee062ff441170acfc0e3079f16cae96a0
[node1] (local) root@192.168.0.18 ~/app
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
e1470f4f6f9f docker-101 "docker-entrypoint.s…" 7 seconds ago Up 6 seconds 0.0.0.0:
3000->3000/tcp elegant_cohen
1fd52c001d8d docker/getting-started:pwd "nginx -g 'daemon of…" 4 minutes ago Up 4 minutes 0.0.0.0:
80->80/tcp fervent_tharp
[node1] (local) root@192.168.0.18 ~/app
$
Search
14:33
16-11-2023

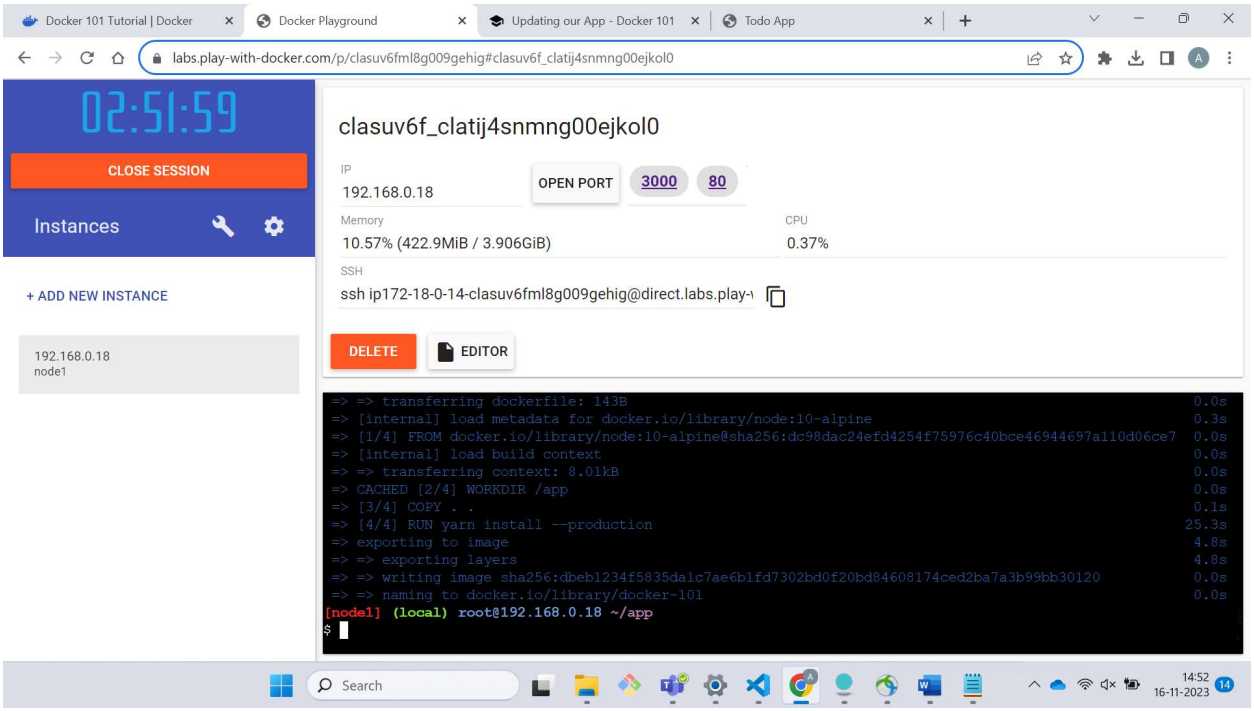

Docker 101 Tutorial | Docker
Docker Playground
Todo App
Our Application - Docker 101
Not secure | ip172-18-0-14-clasuv6fml8g009gehig-3000.direct.labs.play-with-docker.com
New Item
Add Item
No items yet! Add one above!
Search
14:33
16-11-2023

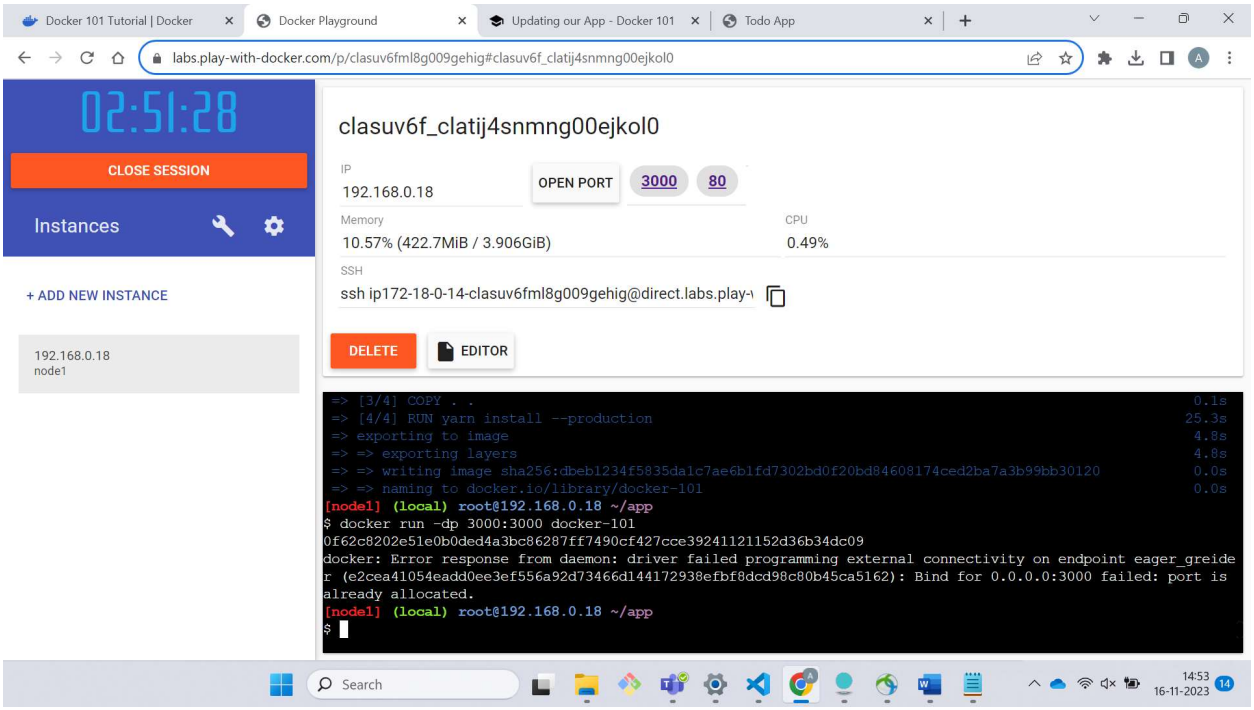

Docker 101 Tutorial | Docker
Docker Playground
Our Application - Docker 101
Todo App
Not secure
ip172-18-0-14-clasuv6fml8g009gehig-3000.direct.labs.play-with-docker.com
New Item
Add Item
Fruits
Foods
Search
14:36
16-11-2023

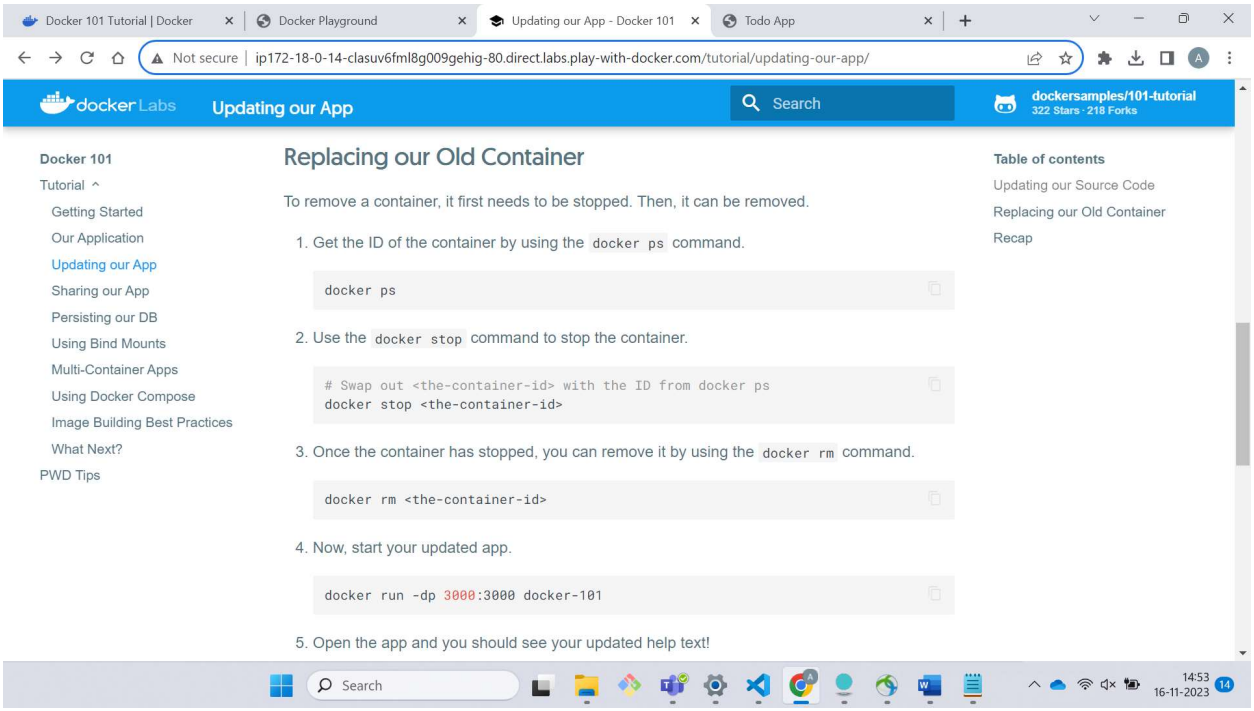

Docker 101 Tutorial | Docker
Docker Playground
Updating our App - Docker 101
Todo App
Not secure
ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/updating-our-app/
docker Labs
Updating our App
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
Updating our Source Code
1. In the ~/app/src/static/js/app.js file, update line 56 to use the new empty text. (Editing files in PWD tips here)
No items yet! Add one above!
You have no todo items yet! Add one above!
2. Let's build our updated version of the image, using the same command we used before.
docker build -t docker-101 .
3. Let's start a new container using the updated code.
docker run -dp 3000:3000 docker-101
Uh oh! You probably saw an error like this (the IDs will be different):
docker: Error response from daemon: driver failed programming external connectivity
(bb242b2ca4d67eba76e79474fb36bb5125708ebdabd7f45c8eaf16caaabde9dd): Bind for 0.0.0.
Table of contents
Updating our Source Code
Replacing our Old Container
Recap
Search
14:52
16-11-2023

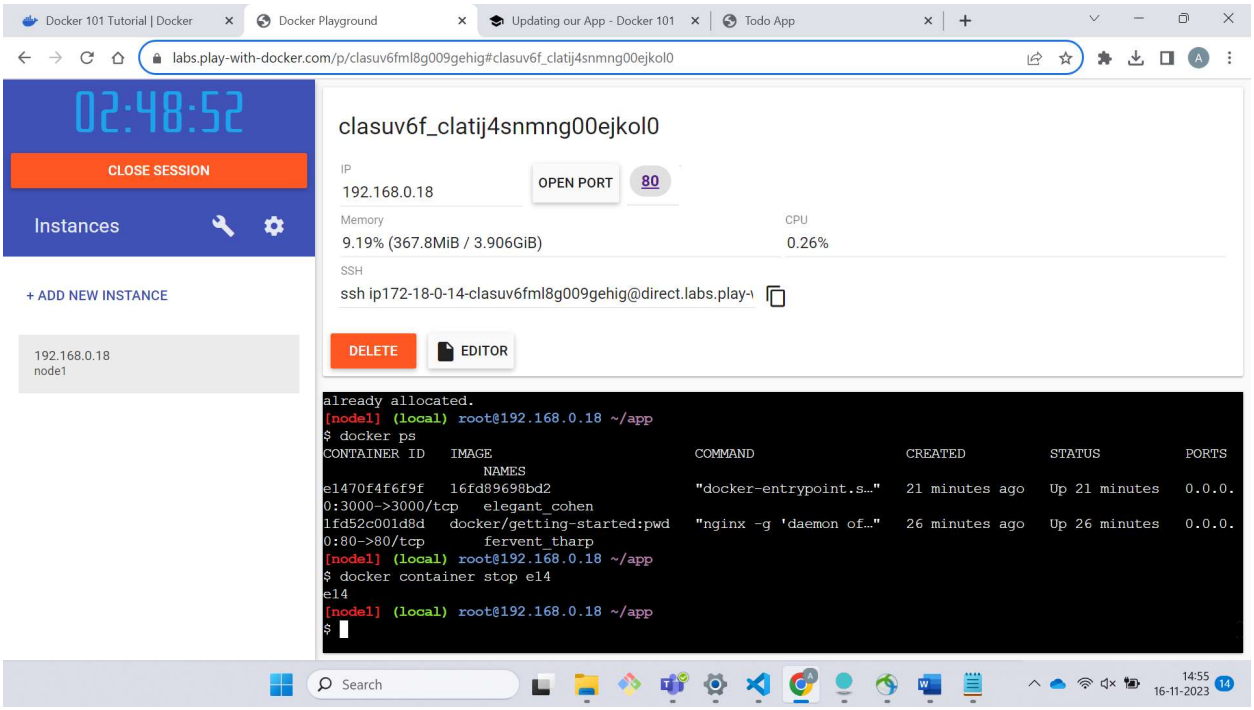

Editor - Google Chrome
labs.play-with-docker.com/sessions/clasuv6fml8g009gehig/instances/clasuv6f_clatij4snmng00ejkol0/editor
Create or upload files in the session terminal and then refresh
app.js
Save
Reload
/root
app
Dockerfile
package.json
spec
src
index.js
persistence
routes
static
css
index.html
js
app.js
babel.min.js
react-bootstrap.js
react-dom.production.min.j
react.production.min.js
yarn.lock
app.zip
const onItemRemoval = React.useCallback(
item => {
const index = items.findIndex(i => i.id === item.id);
setItems([...items.slice(0, index), ...items.slice(index + 1)]);
},
[items],
);
if (items === null) return 'Loading...';
return (
<React.Fragment>
<AddItemForm onNewItem={onNewItem} />
{items.length === 0 && (
You have no todo items yet! Add one above!
)}
{items.map(item => (
<ItemDisplay
item={item}
key={item.id}
onItemUpdate={onItemUpdate}
onItemRemoval={onItemRemoval}
/>
))}
</React.Fragment>
);
}
function AddItemForm({ onNewItem }) {
const { Form, InputGroup, Button } = ReactBootstrap;
const [newItem, setNewItem] = React.useState('');
const [submitting, setSubmitting] = React.useState(false);
const submitNewItem = e => {
e.preventDefault();
Search
14:40
16-11-2023

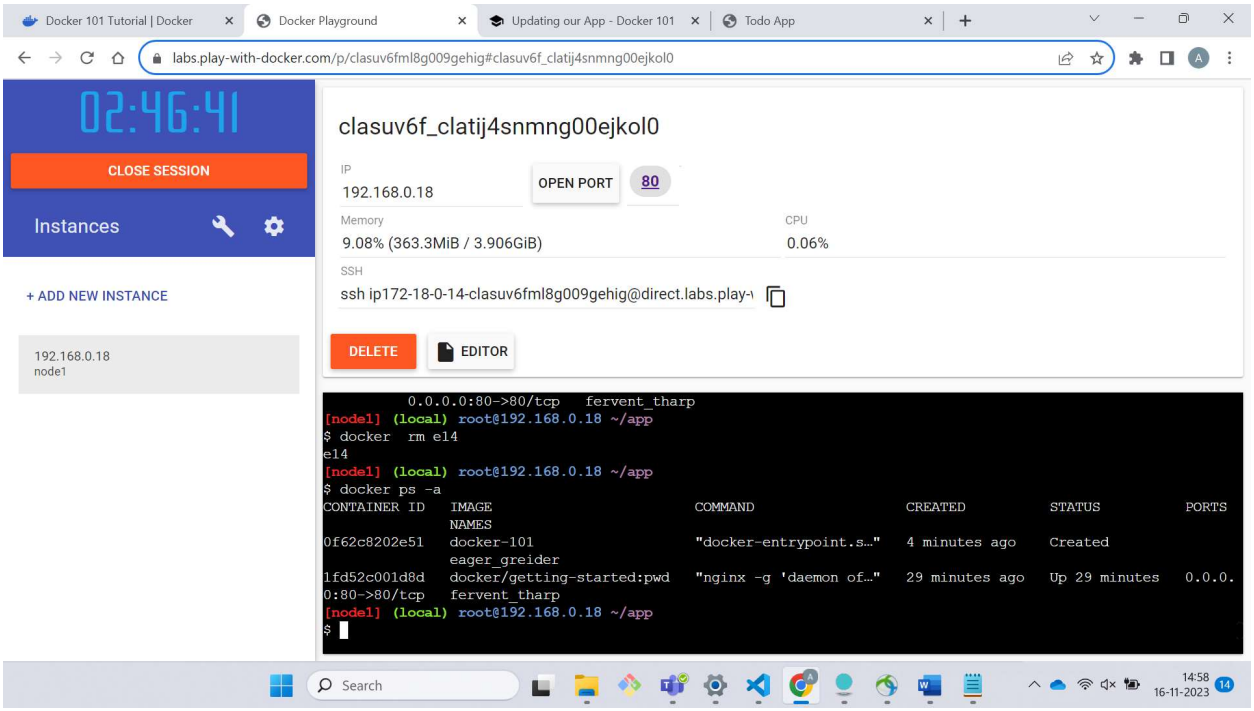

Docker 101 Tutorial | Docker
Docker Playground
Updating our App - Docker 101
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clatij4snmng00ejkol0
02:52:20
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clatij4snmng00ejkol0
IP
192.168.0.18
OPEN PORT
3000
80
Memory
10.57% (422.8MiB / 3.906GiB)
CPU
0.23%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
80->80/tcp fervent_tharp
[node1] (local) root@192.168.0.18 ~/app
$ ls
Dockerfile package.json spec src yarn.lock
[node1] (local) root@192.168.0.18 ~/app
$ docker build -t docker-101 .
bash: $: command not found
[node1] (local) root@192.168.0.18 ~/app
$ docker build -t docker-101 .
[+] Building 30.7s (9/9) FINISHED docker:default
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 143B 0.0s
Search
14:52
16-11-2023

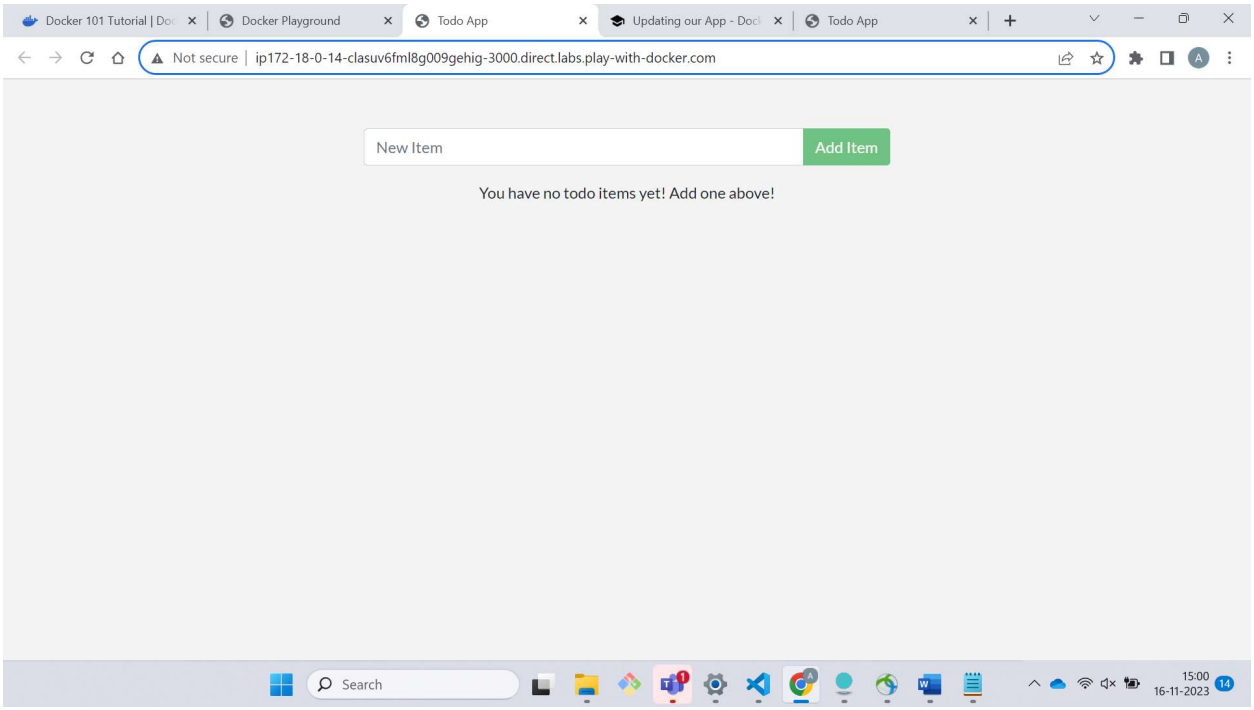

Docker 101 Tutorial | Docker
Docker Playground
Updating our App - Docker 101
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clatij4snmng00ejkol0
02:51:59
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clatij4snmng00ejkol0
IP
192.168.0.18
OPEN PORT
3000
80
Memory
10.57% (422.9MiB / 3.906GiB)
CPU
0.37%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
=> => transferring dockerfile: 143B 0.0s
=> [internal] load metadata for docker.io/library/node:10-alpine 0.3s
=> [1/4] FROM docker.io/library/node:10-alpine@sha256:dc98dac24efd4254f75976c40bce46944697a110d06ce7 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 8.01kB 0.0s
=> CACHED [2/4] WORKDIR /app 0.0s
=> [3/4] COPY . . 0.1s
=> [4/4] RUN yarn install --production 25.3s
=> exporting to image 4.8s
=> => exporting layers 4.8s
=> => writing image sha256:dbeb1234f5835da1c7ae6b1fd7302bd0f20bd84608174ced2ba7a3b99bb30120 0.0s
=> => naming to docker.io/library/docker-101 0.0s
[node1] (local) root@192.168.0.18 ~/app
$
Search
14:52
16-11-2023

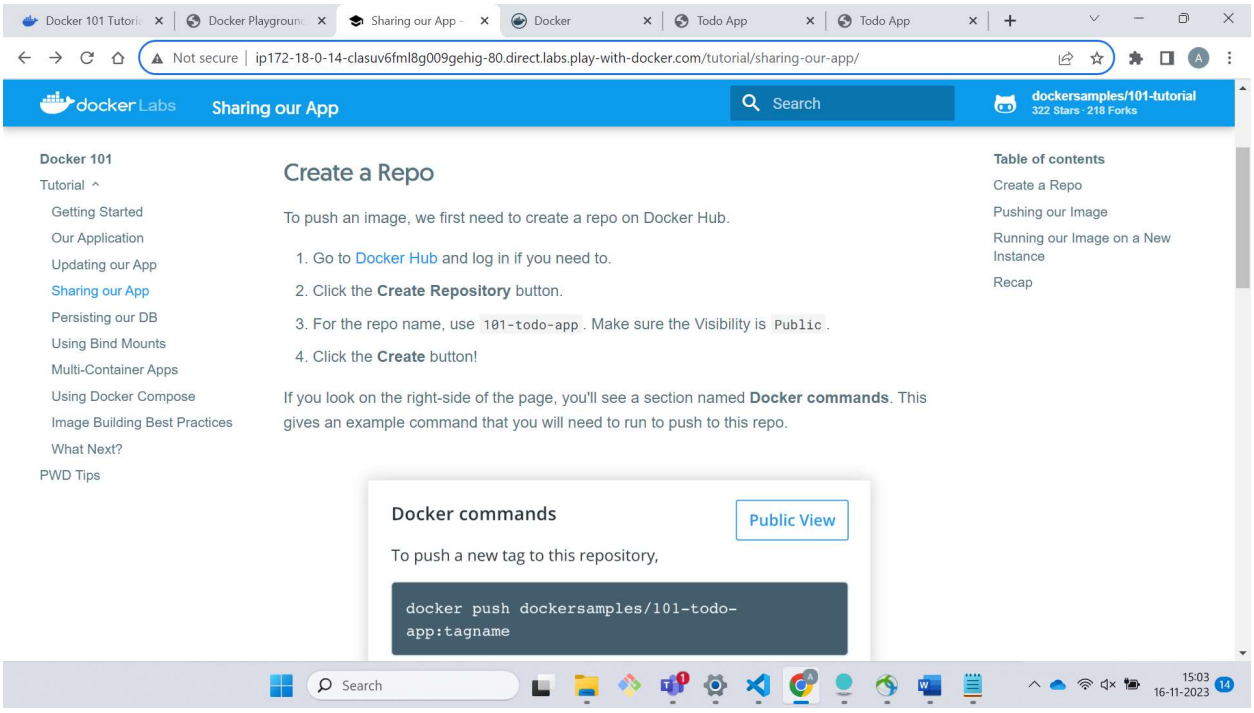

Docker 101 Tutorial | Docker
Docker Playground
Updating our App - Docker 101
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clatij4snmng00ejkol0
02:51:28
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clatij4snmng00ejkol0
IP
192.168.0.18
OPEN PORT
3000
80
Memory
10.57% (422.7MiB / 3.906GiB)
CPU
0.49%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
=> [3/4] COPY . . 0.1s
=> [4/4] RUN yarn install --production 25.3s
=> exporting to image 4.8s
=> => exporting layers 4.8s
=> => writing image sha256:dbeb1234f5835da1c7ae6b1fd7302bd0f20bd84608174ced2ba7a3b99bb30120 0.0s
=> => naming to docker.io/library/docker-101 0.0s
[node1] (local) root@192.168.0.18 ~/app
$ docker run -dp 3000:3000 docker-101
0f62c8202e51e0b0ded4a3bc86287ff7490cf427cce39241121152d36b34dc09
docker: Error response from daemon: driver failed programming external connectivity on endpoint eager_greide
r (e2cea41054eadd0ee3ef556a92d73466d144172938efbf8dcd98c80b45ca5162): Bind for 0.0.0.0:3000 failed: port is
already allocated.
[node1] (local) root@192.168.0.18 ~/app
$
Search
14:53
16-11-2023

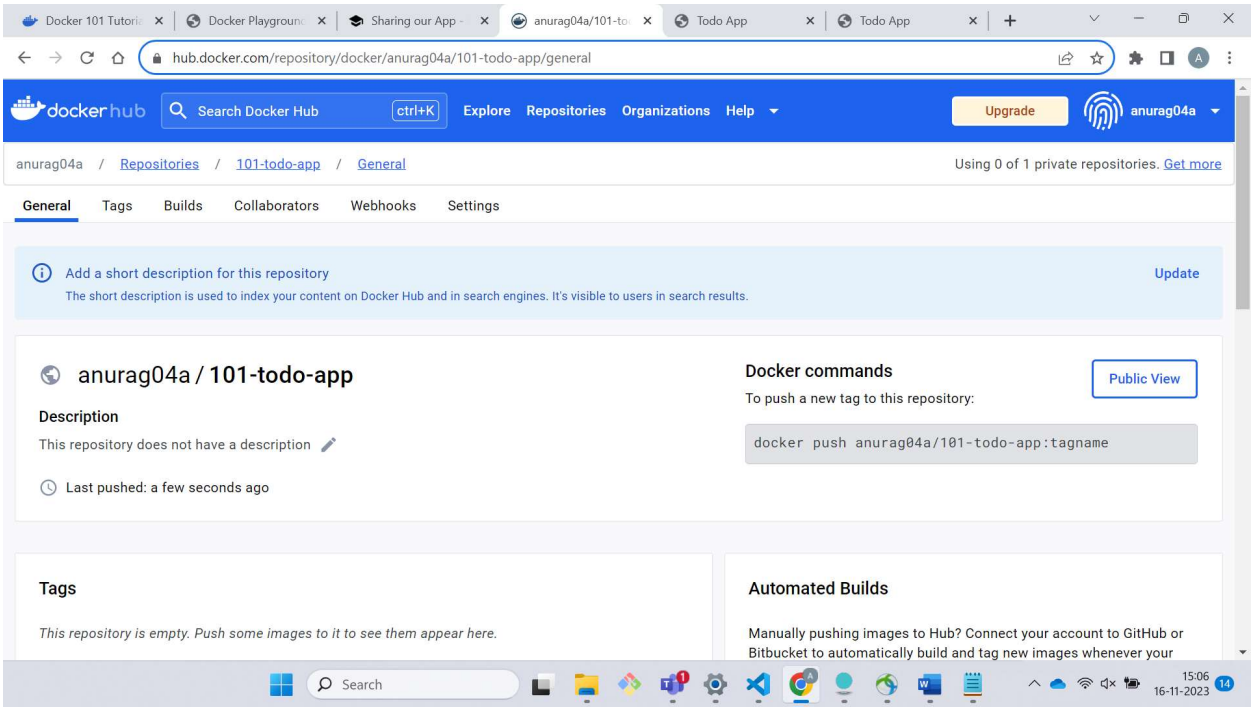

Docker 101 Tutorial | Docker
Docker Playground
Updating our App - Docker 101
Todo App
Not secure
ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/updating-our-app/
docker Labs
Updating our App
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
Replacing our Old Container
To remove a container, it first needs to be stopped. Then, it can be removed.
1. Get the ID of the container by using the docker ps command.
docker ps
2. Use the docker stop command to stop the container.
Swap out <the-container-id> with the ID from docker ps
docker stop <the-container-id>
3. Once the container has stopped, you can remove it by using the docker rm command.
docker rm <the-container-id>
4. Now, start your updated app.
docker run -dp 3000:3000 docker-101
5. Open the app and you should see your updated help text!
Table of contents
Updating our Source Code
Replacing our Old Container
Recap
Search
14:53
16-11-2023

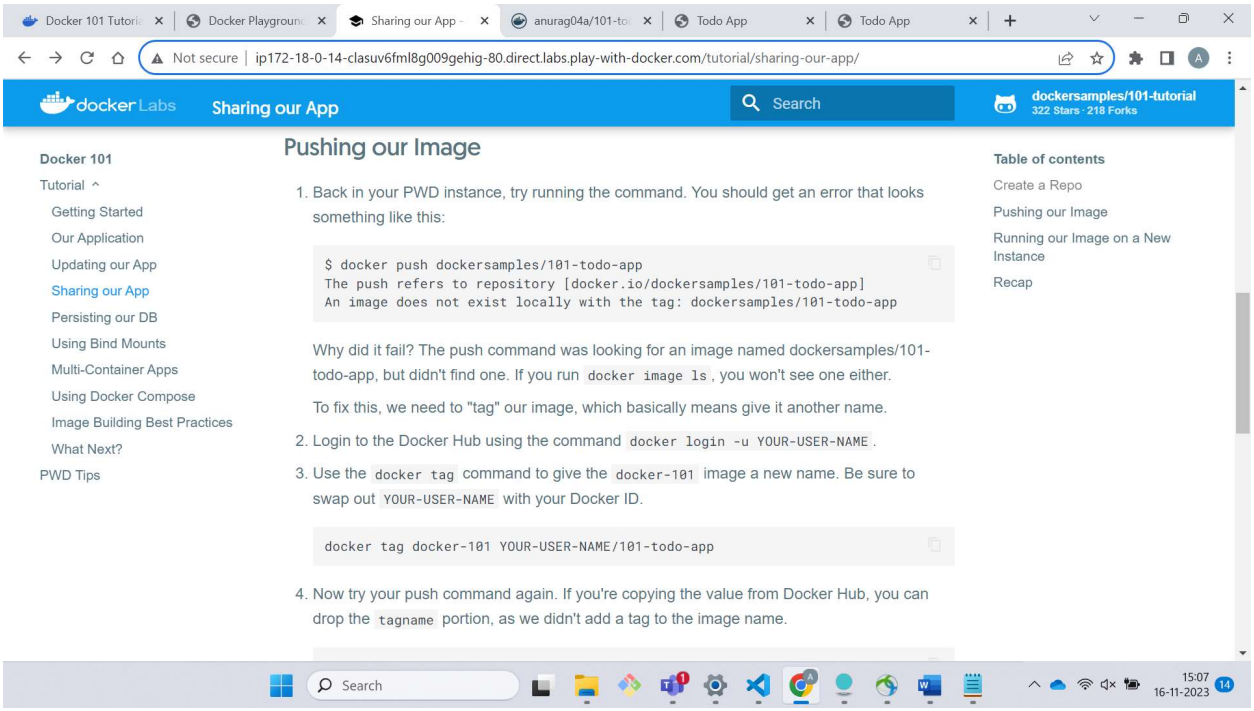

Docker 101 Tutorial | Docker
Docker Playground
Updating our App - Docker 101
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clatij4snmng00ejkol0
02:48:52
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clatij4snmng00ejkol0
IP
192.168.0.18
OPEN PORT
80
Memory
9.19% (367.8MiB / 3.906GiB)
CPU
0.26%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
already allocated.
[node1] (local) root@192.168.0.18 ~/app
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
e1470f4f6f9f 16fd89698bd2 "docker-entrypoint.s…" 21 minutes ago Up 21 minutes 0.0.0.
0:3000->3000/tcp elegant_cohen
1fd52c001d8d docker/getting-started:pwd "nginx -g 'daemon of…" 26 minutes ago Up 26 minutes 0.0.0.
0:80->80/tcp fervent_tharp
[node1] (local) root@192.168.0.18 ~/app
$ docker container stop e14
e14
[node1] (local) root@192.168.0.18 ~/app
$
Search
14:55
16-11-2023

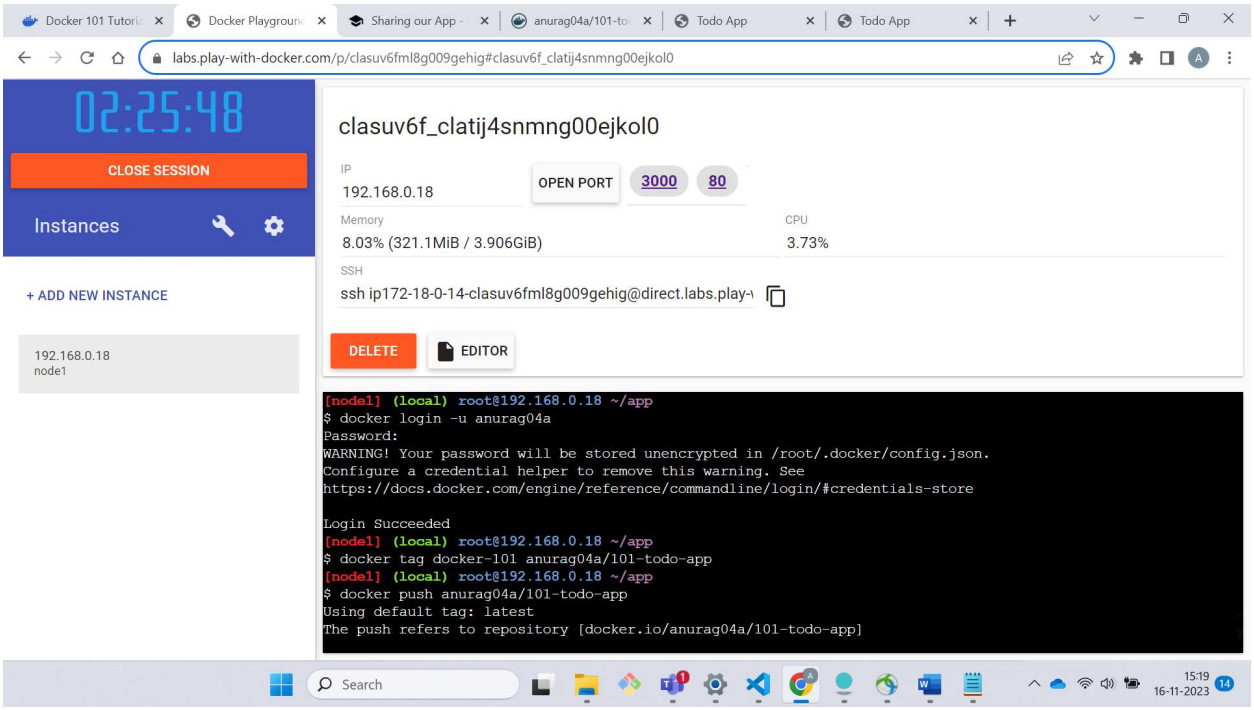

Docker 101 Tutorial | Docker
Docker Playground
Updating our App - Docker 101
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clatij4snmng00ejkol0
02:46:41
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clatij4snmng00ejkol0
IP
192.168.0.18
OPEN PORT
80
Memory
9.08% (363.3MiB / 3.906GiB)
CPU
0.06%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
0.0.0.0:80->80/tcp fervent_tharp
[node1] (local) root@192.168.0.18 ~/app
$ docker rm e14
e14
[node1] (local) root@192.168.0.18 ~/app
$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
0f62c8202e51 docker-101 "docker-entrypoint.s…" 4 minutes ago Created
eager_greider
1fd52c001d8d docker/getting-started:pwd "nginx -g 'daemon of…" 29 minutes ago Up 29 minutes 0.0.0.
0:80->80/tcp fervent_tharp
[node1] (local) root@192.168.0.18 ~/app
$
Search
14:58
16-11-2023

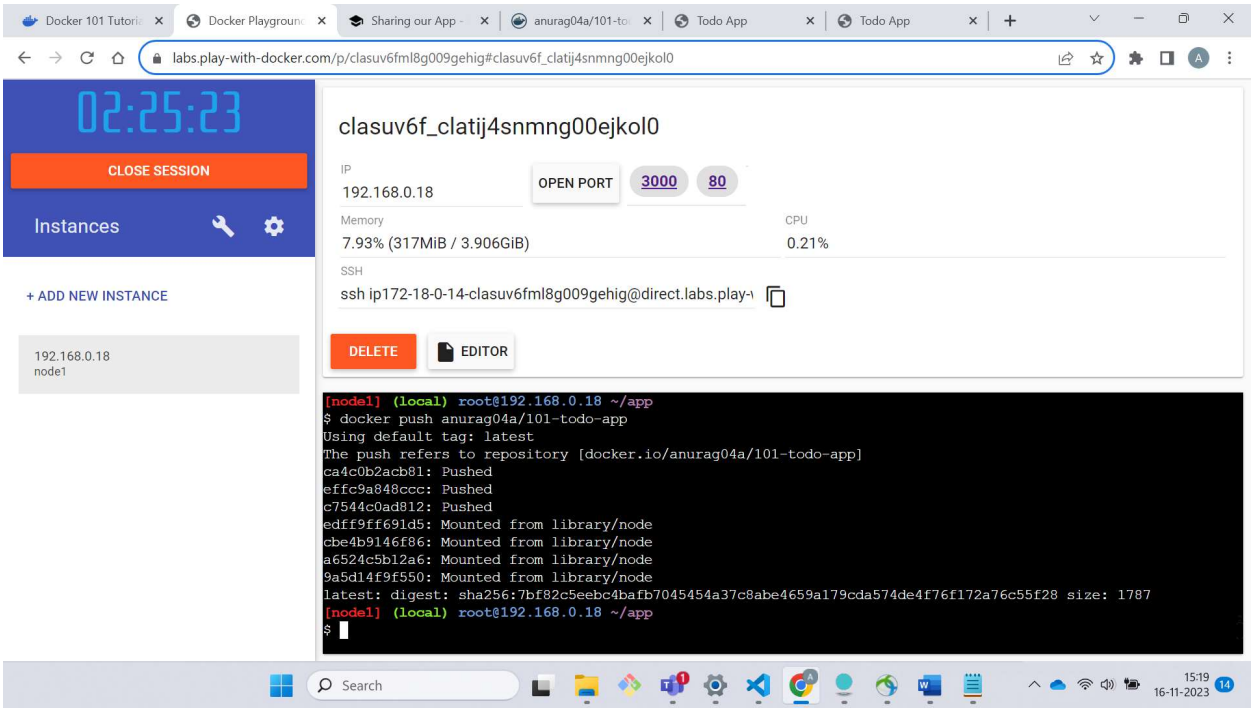

Docker 101 Tutorial | Doc
Docker Playground
Todo App
Updating our App - Doc
Todo App
Not secure | ip172-18-0-14-clasuv6fml8g009gehig-3000.direct.labs.play-with-docker.com
New Item
Add Item
You have no todo items yet! Add one above!
Search
15:00
16-11-2023

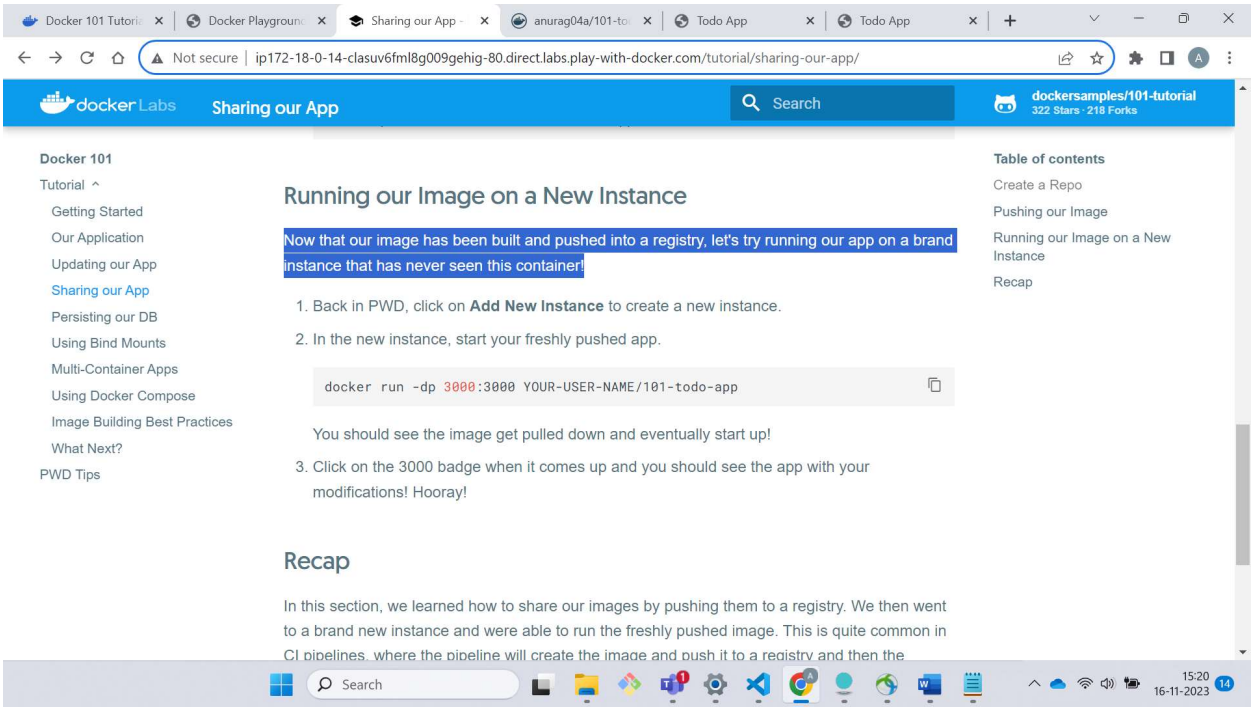

Not secure ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/sharing-our-app/
docker Labs
Sharing our App
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
Create a Repo
To push an image, we first need to create a repo on Docker Hub.
1. Go to Docker Hub and log in if you need to.
2. Click the Create Repository button.
3. For the repo name, use 101-todo-app. Make sure the Visibility is Public.
4. Click the Create button!
If you look on the right-side of the page, you'll see a section named Docker commands. This gives an example command that you will need to run to push to this repo.
Table of contents
Create a Repo
Pushing our Image
Running our Image on a New Instance
Recap
Docker commands
Public View
To push a new tag to this repository,
docker push dockersamples/101-todo-app:tagname
Search
15:03
16-11-2023

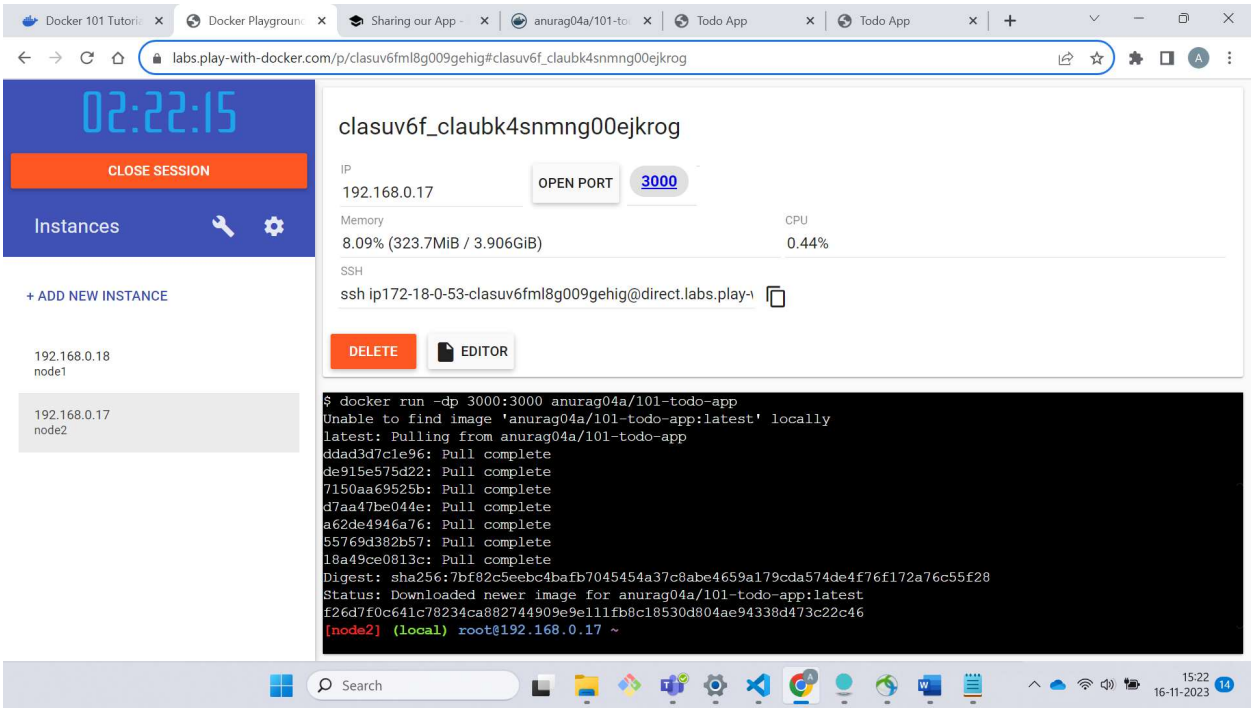

hub.docker.com/repository/docker/anurag04a/101-todo-app/general
docker hub
Search Docker Hub
ctrl+K
Explore
Repositories
Organizations
Help
Upgrade
anurag04a
anurag04a / Repositories / 101-todo-app / General
Using 0 of 1 private repositories. Get more
General
Tags
Builds
Collaborators
Webhooks
Settings
Add a short description for this repository
The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.
Update
anurag04a / 101-todo-app
Description
This repository does not have a description
Last pushed: a few seconds ago
Docker commands
To push a new tag to this repository:
Public View
docker push anurag04a/101-todo-app:tagname
Tags
This repository is empty. Push some images to it to see them appear here.
Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your
Search
15:06
16-11-2023

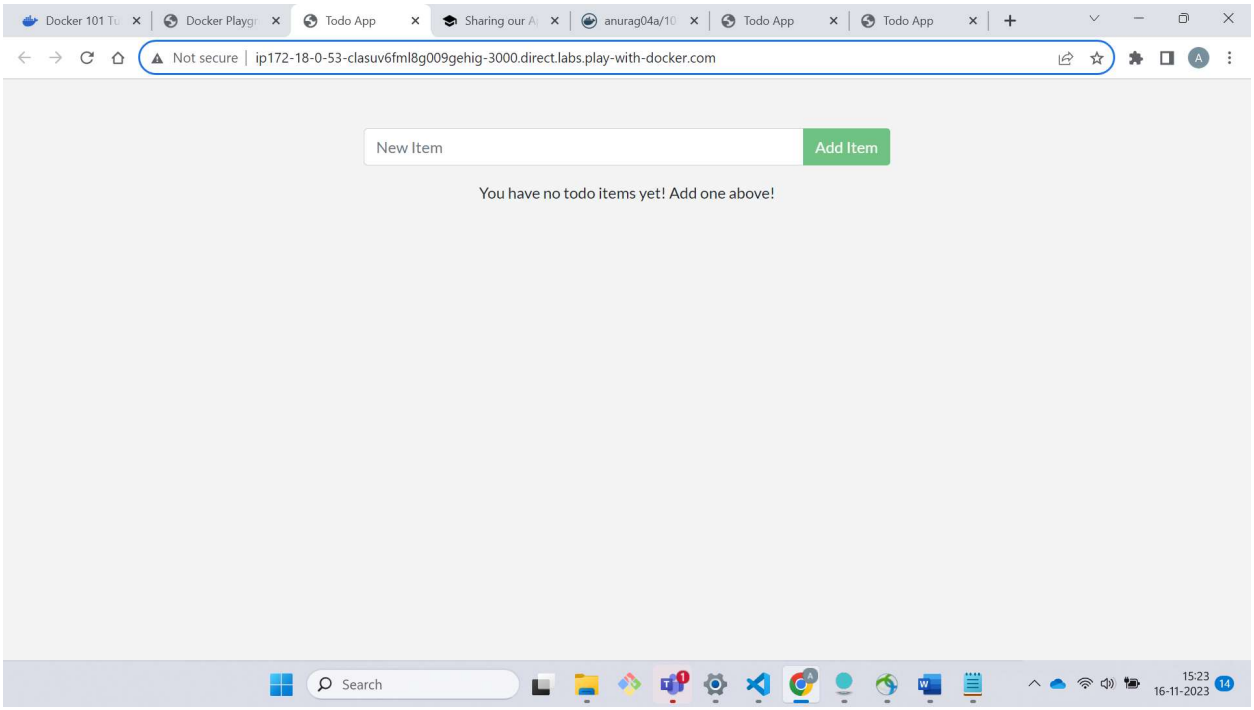

Docker 101 Tutori
Docker Playgroun
Sharing our App -
anurag04a/101-to
Todo App
Todo App
Not secure
ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/sharing-our-app/
docker Labs
Sharing our App
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
Pushing our Image
1. Back in your PWD instance, try running the command. You should get an error that looks something like this:
$ docker push dockersamples/101-todo-app
The push refers to repository [docker.io/dockersamples/101-todo-app]
An image does not exist locally with the tag: dockersamples/101-todo-app
Why did it fail? The push command was looking for an image named dockersamples/101-todo-app, but didn't find one. If you run docker image ls, you won't see one either.
To fix this, we need to "tag" our image, which basically means give it another name.
2. Login to the Docker Hub using the command docker login -u YOUR-USER-NAME.
3. Use the docker tag command to give the docker-101 image a new name. Be sure to swap out YOUR-USER-NAME with your Docker ID.
docker tag docker-101 YOUR-USER-NAME/101-todo-app
4. Now try your push command again. If you're copying the value from Docker Hub, you can drop the tagname portion, as we didn't add a tag to the image name.
Table of contents
Create a Repo
Pushing our Image
Running our Image on a New Instance
Recap
Search
15:07
16-11-2023

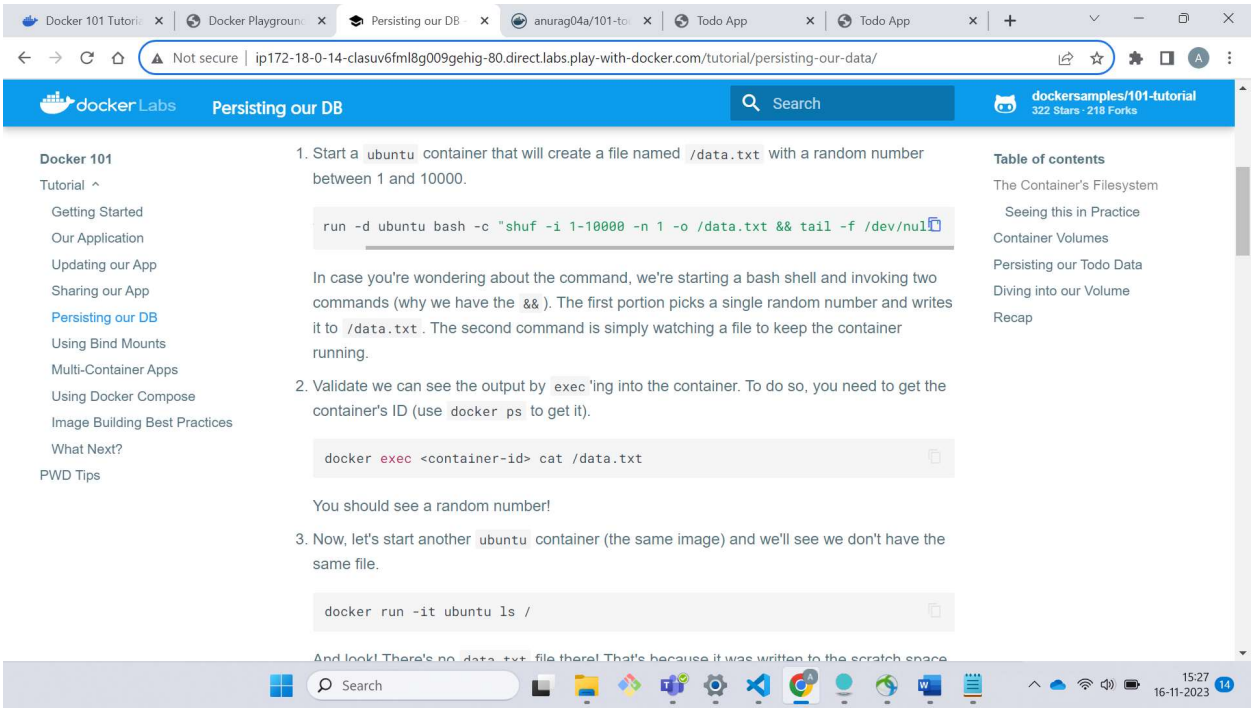

Docker 101 Tutori
Docker Playgroun
Sharing our App -
anurag04a/101-to
Todo App
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clatij4snmng00ejkol0
02:25:48
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clatij4snmng00ejkol0
IP
192.168.0.18
OPEN PORT
3000
80
Memory
8.03% (321.1MiB / 3.906GiB)
CPU
3.73%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node1] (local) root@192.168.0.18 ~/app
$ docker login -u anurag04a
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
[node1] (local) root@192.168.0.18 ~/app
$ docker tag docker-101 anurag04a/101-todo-app
[node1] (local) root@192.168.0.18 ~/app
$ docker push anurag04a/101-todo-app
Using default tag: latest
The push refers to repository [docker.io/anurag04a/101-todo-app]
Search
15:19
16-11-2023

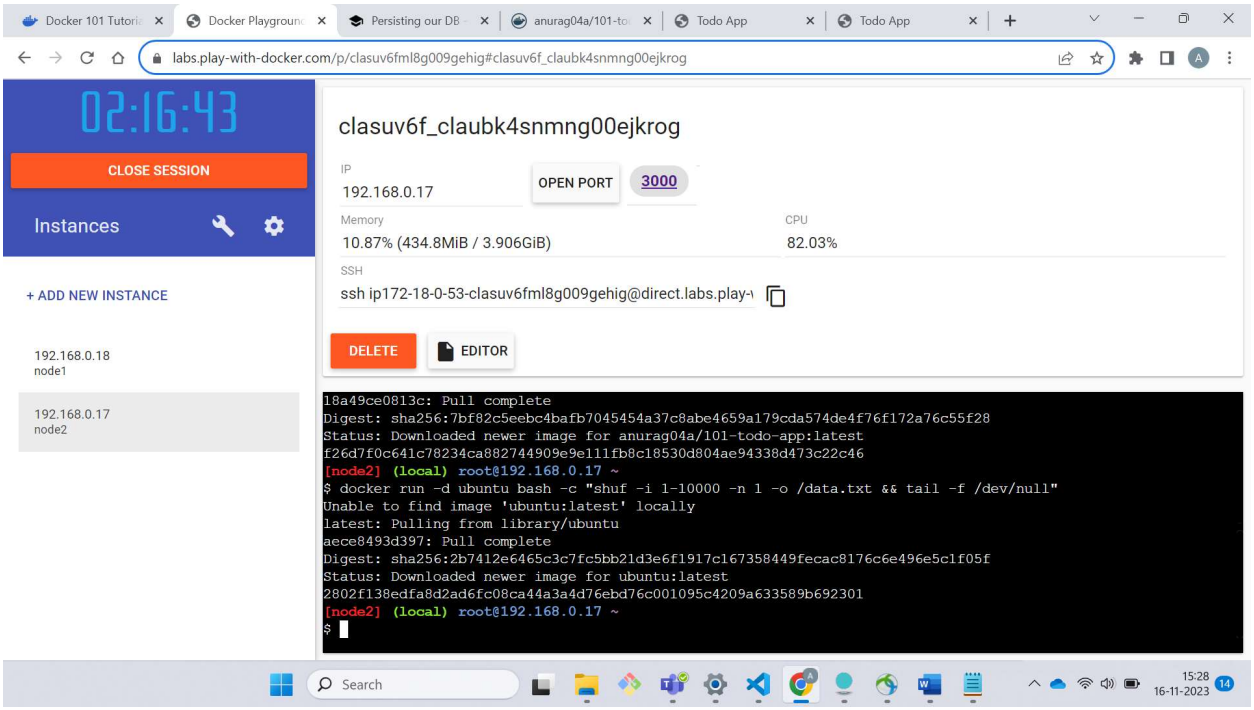

labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_clatij4snmng00ejkol0
02:25:23
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
clasuv6f_clatij4snmng00ejkol0
IP
192.168.0.18
OPEN PORT
3000
80
Memory
7.93% (317MiB / 3.906GiB)
CPU
0.21%
SSH
ssh ip172-18-0-14-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node1] (local) root@192.168.0.18 ~/app
$ docker push anurag04a/101-todo-app
Using default tag: latest
The push refers to repository [docker.io/anurag04a/101-todo-app]
ca4c0b2acb81: Pushed
effc9a848ccc: Pushed
c7544c0ad812: Pushed
edff9ff691d5: Mounted from library/node
cbe4b9146f86: Mounted from library/node
a6524c5b12a6: Mounted from library/node
9a5d14f9f550: Mounted from library/node
latest: digest: sha256:7bf82c5eebc4bafb7045454a37c8abe4659a179cda574de4f76f172a76c55f28 size: 1787
[node1] (local) root@192.168.0.18 ~/app
$
Search
15:19
16-11-2023

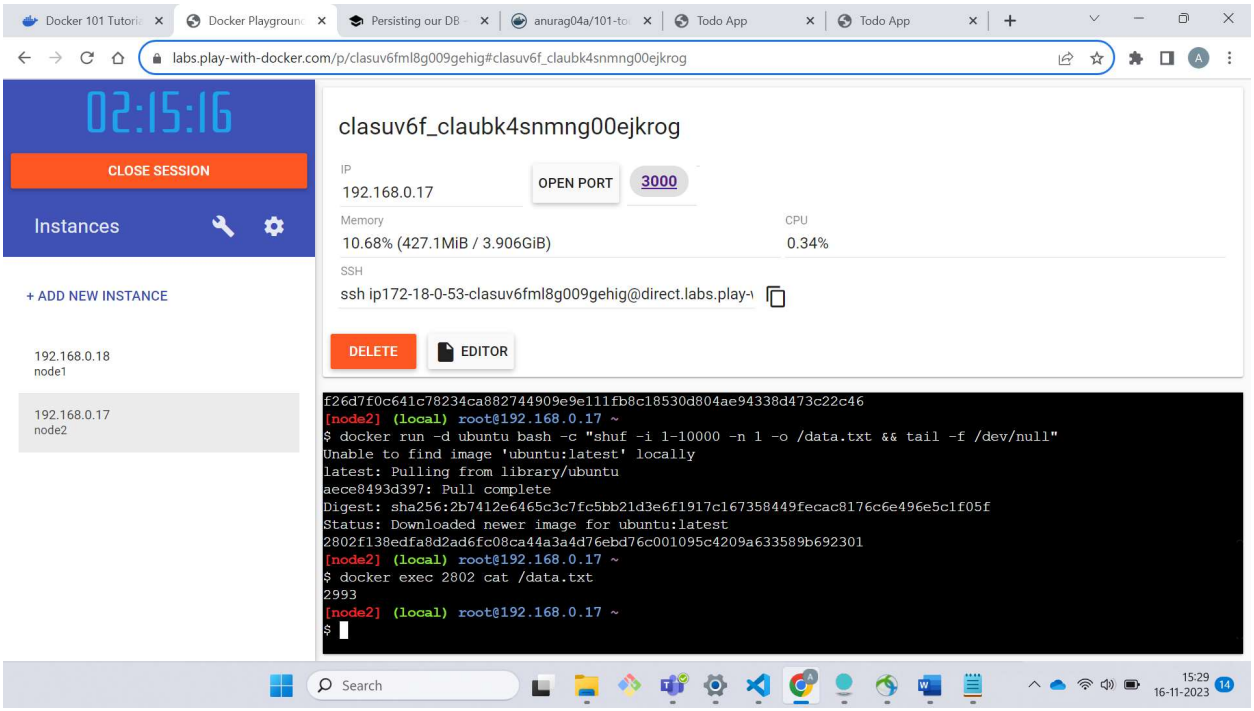

Not secure | ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/sharing-our-app/
docker Labs
Sharing our App
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
Table of contents
Create a Repo
Pushing our Image
Running our Image on a New Instance
Recap
Running our Image on a New Instance
Now that our image has been built and pushed into a registry, let's try running our app on a brand instance that has never seen this container!
1. Back in PWD, click on Add New Instance to create a new instance.
2. In the new instance, start your freshly pushed app.
docker run -dp 3000:3000 YOUR-USER-NAME/101-todo-app
You should see the image get pulled down and eventually start up!
3. Click on the 3000 badge when it comes up and you should see the app with your modifications! Hooray!
Recap
In this section, we learned how to share our images by pushing them to a registry. We then went to a brand new instance and were able to run the freshly pushed image. This is quite common in
Search
15:20
16-11-2023

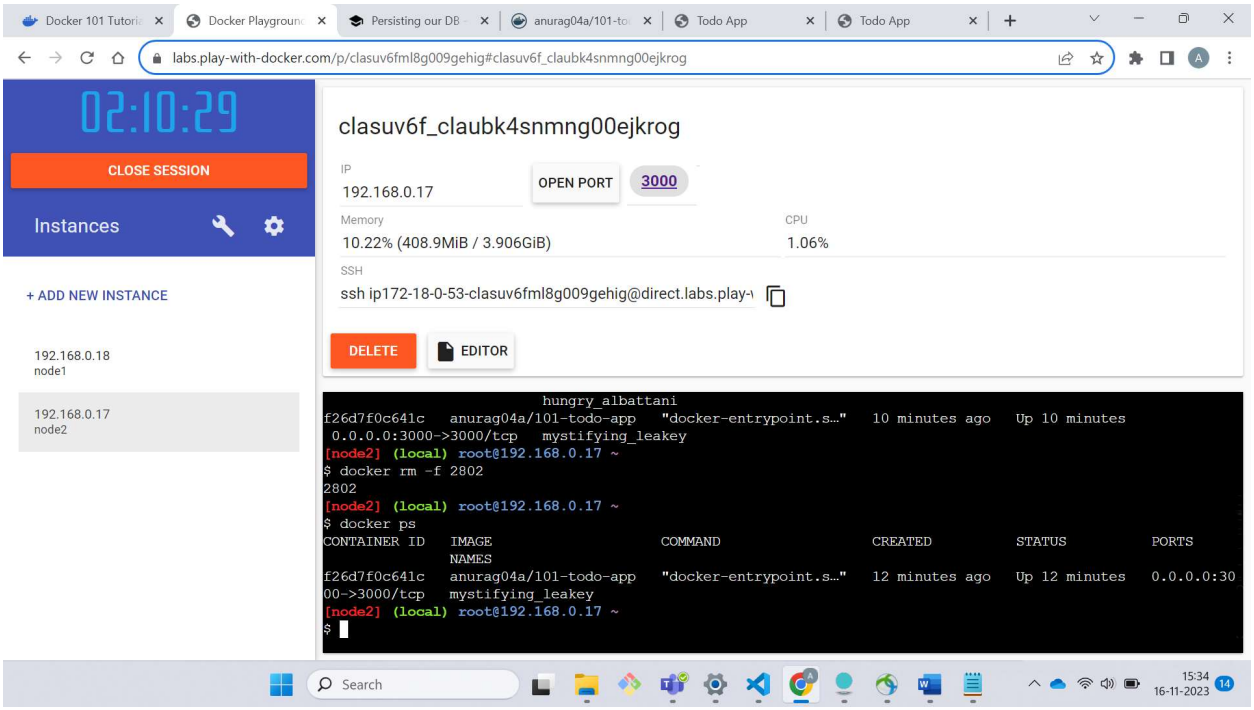

Docker 101 Tutori
Docker Playgroun
Sharing our App -
anurag04a/101-to
Todo App
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
02:22:15
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
3000
Memory
8.09% (323.7MiB / 3.906GiB)
CPU
0.44%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
$ docker run -dp 3000:3000 anurag04a/101-todo-app
Unable to find image 'anurag04a/101-todo-app:latest' locally
latest: Pulling from anurag04a/101-todo-app
ddad3d7c1e96: Pull complete
de915e575d22: Pull complete
7150aa69525b: Pull complete
d7aa47be044e: Pull complete
a62de4946a76: Pull complete
55769d382b57: Pull complete
18a49ce0813c: Pull complete
Digest: sha256:7bf82c5eebc4bafb7045454a37c8abe4659a179cda574de4f76f172a76c55f28
Status: Downloaded newer image for anurag04a/101-todo-app:latest
f26d7f0c641c78234ca882744909e9e111fb8c18530d804ae94338d473c22c46
[node2] (local) root@192.168.0.17 ~
Search
15:22
16-11-2023


Docker 101 Tu
Docker Playgr
Todo App
Sharing our A
anurag04a/10
Todo App
Todo App
Not secure | ip172-18-0-53-clasuv6fml8g009gehig-3000.direct.labs.play-with-docker.com
New Item
Add Item
You have no todo items yet! Add one above!
Search
15:23
16-11-2023


Docker 101 Tutori
Docker Playgroun
Persisting our DB
anurag04a/101-to
Todo App
Todo App
Not secure
ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/persisting-our-data/
docker Labs
Persisting our DB
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
1. Start a ubuntu container that will create a file named /data.txt with a random number between 1 and 10000.
run -d ubuntu bash -c "shuf -i 1-10000 -n 1 -o /data.txt && tail -f /dev/null
In case you're wondering about the command, we're starting a bash shell and invoking two commands (why we have the &&). The first portion picks a single random number and writes it to /data.txt. The second command is simply watching a file to keep the container running.
2. Validate we can see the output by exec'ing into the container. To do so, you need to get the container's ID (use docker ps to get it).
docker exec <container-id> cat /data.txt
You should see a random number!
3. Now, let's start another ubuntu container (the same image) and we'll see we don't have the same file.
docker run -it ubuntu ls /
Table of contents
The Container's Filesystem
Seeing this in Practice
Container Volumes
Persisting our Todo Data
Diving into our Volume
Recap
Search
15:27
16-11-2023


Docker 101 Tutori
Docker Playgroun
Persisting our DB
anurag04a/101-to
Todo App
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
02:16:43
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
3000
Memory
10.87% (434.8MiB / 3.906GiB)
CPU
82.03%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
18a49ce0813c: Pull complete
Digest: sha256:7bf82c5eebc4bafb7045454a37c8abe4659a179cda574de4f76f172a76c55f28
Status: Downloaded newer image for anurag04a/101-todo-app:latest
f26d7f0c641c78234ca882744909e9e111fb8c18530d804ae94338d473c22c46
[node2] (local) root@192.168.0.17 ~
$ docker run -d ubuntu bash -c "shuf -i 1-10000 -n 1 -o /data.txt && tail -f /dev/null"
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
aece8493d397: Pull complete
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Downloaded newer image for ubuntu:latest
2802f138edfa8d2ad6fc08ca44a3a4d76ebd76c001095c4209a633589b692301
[node2] (local) root@192.168.0.17 ~
$
Search
15:28
16-11-2023


Docker 101 Tutori
Docker Playgroun
Persisting our DB
anurag04a/101-to
Todo App
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
02:15:16
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
3000
Memory
10.68% (427.1MiB / 3.906GiB)
CPU
0.34%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
f26d7f0c641c78234ca882744909e9e111fb8c18530d804ae94338d473c22c46
[node2] (local) root@192.168.0.17 ~
$ docker run -d ubuntu bash -c "shuf -i 1-10000 -n 1 -o /data.txt && tail -f /dev/null"
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
aece8493d397: Pull complete
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Downloaded newer image for ubuntu:latest
2802f138edfa8d2ad6fc08ca44a3a4d76ebd76c001095c4209a633589b692301
[node2] (local) root@192.168.0.17 ~
$ docker exec 2802 cat /data.txt
2993
[node2] (local) root@192.168.0.17 ~
$
Search
15:29
16-11-2023


Docker 101 Tutori
Docker Playgroun
Persisting our DB
anurag04a/101-to
Todo App
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
02:10:29
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
3000
Memory
10.22% (408.9MiB / 3.906GiB)
CPU
1.06%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
hungry_albattani
f26d7f0c641c anurag04a/101-todo-app "docker-entrypoint.s…" 10 minutes ago Up 10 minutes
0.0.0.0:3000->3000/tcp mystifying_leakey
[node2] (local) root@192.168.0.17 ~
$ docker rm -f 2802
2802
[node2] (local) root@192.168.0.17 ~
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
f26d7f0c641c anurag04a/101-todo-app "docker-entrypoint.s…" 12 minutes ago Up 12 minutes 0.0.0.0:30
00->3000/tcp mystifying_leakey
[node2] (local) root@192.168.0.17 ~
$
Search
15:34
16-11-2023

correct data is provided.

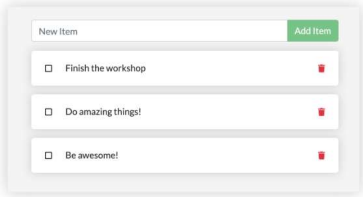
1. Create a volume by using the `docker volume create` command.

```
docker volume create todo-db
```

2. Start the todo container, but add the `-v` flag to specify a volume mount. We will use the named volume and mount it to `/etc/todos`, which will capture all files created at the path.

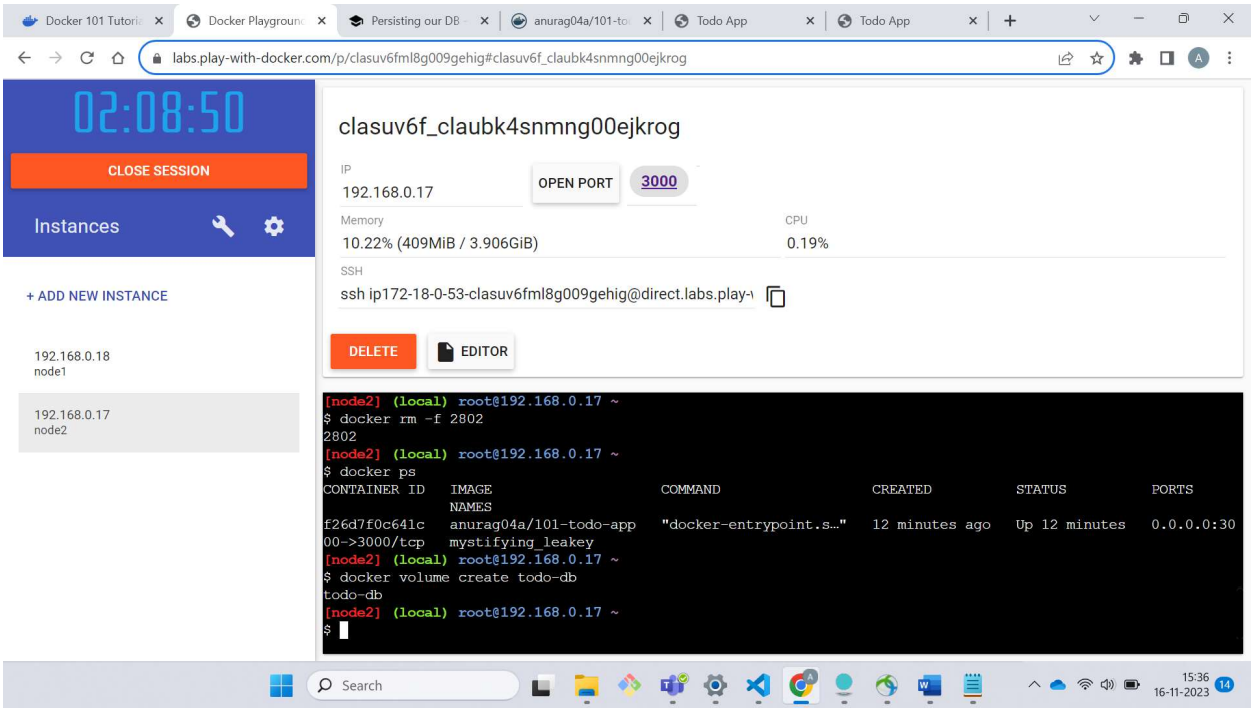
```
docker run -dp 3000:3000 -v todo-db:/etc/todos docker-101
```

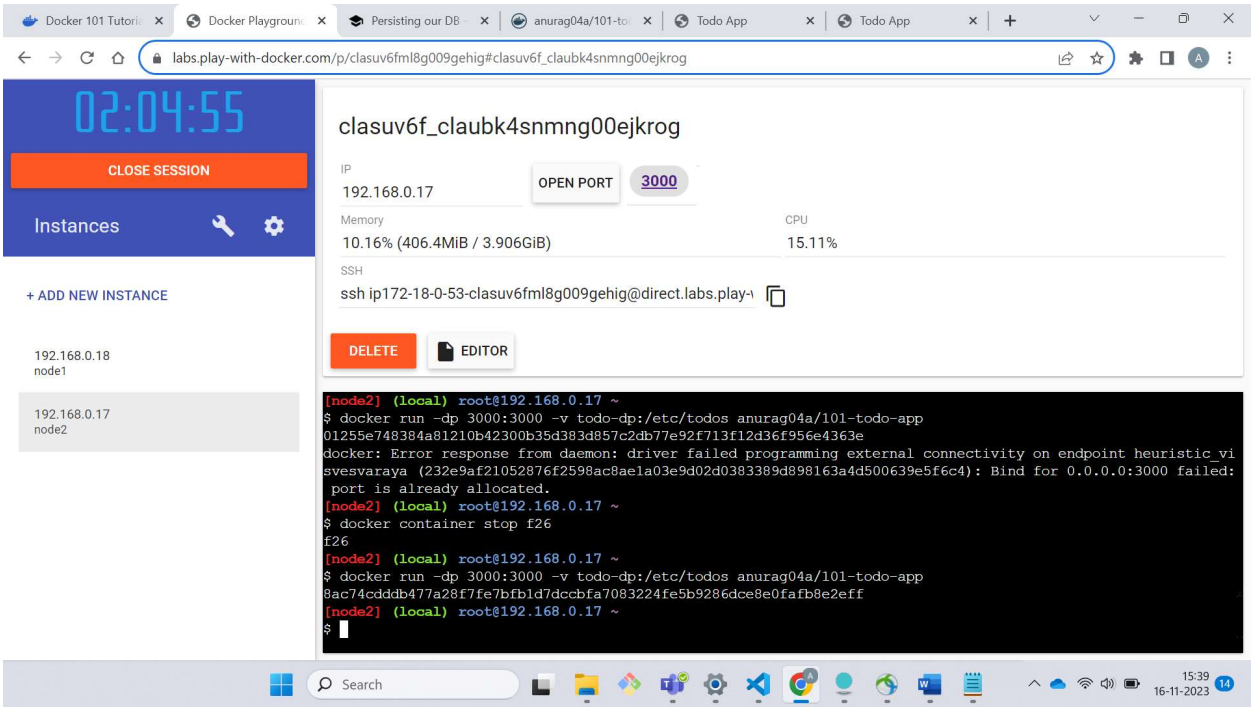
3. Once the container starts up, open the app and add a few items to your todo list.

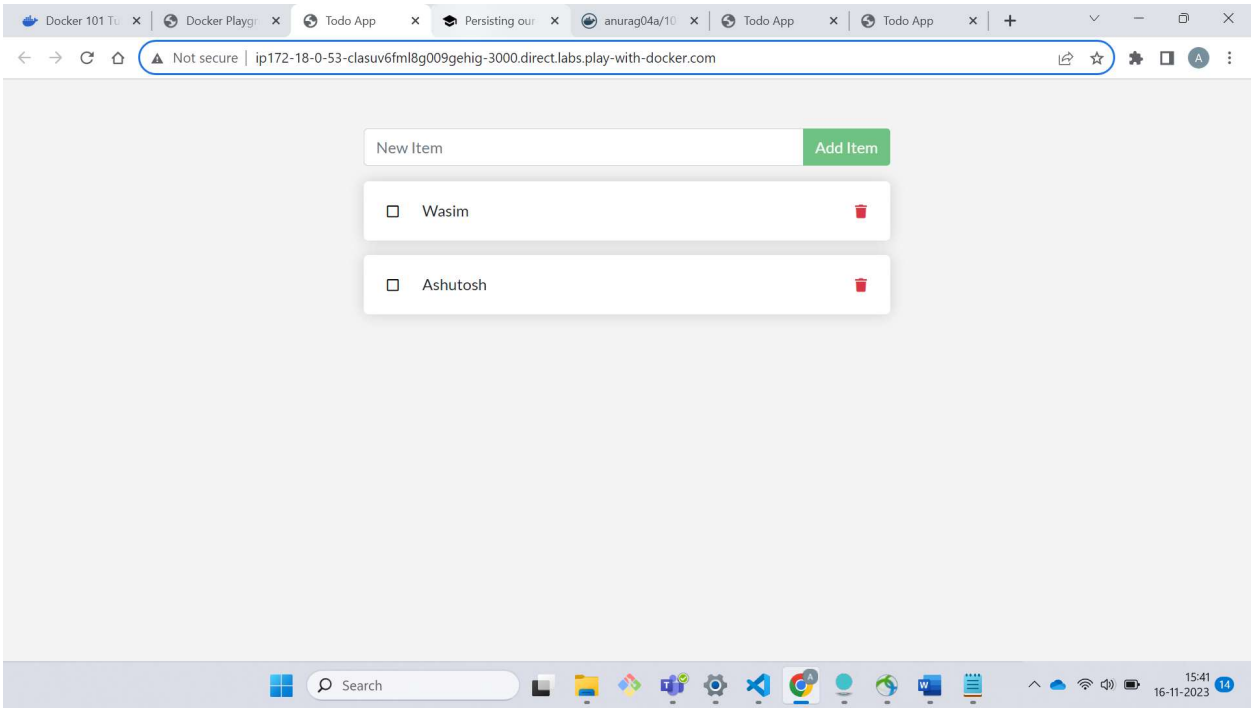


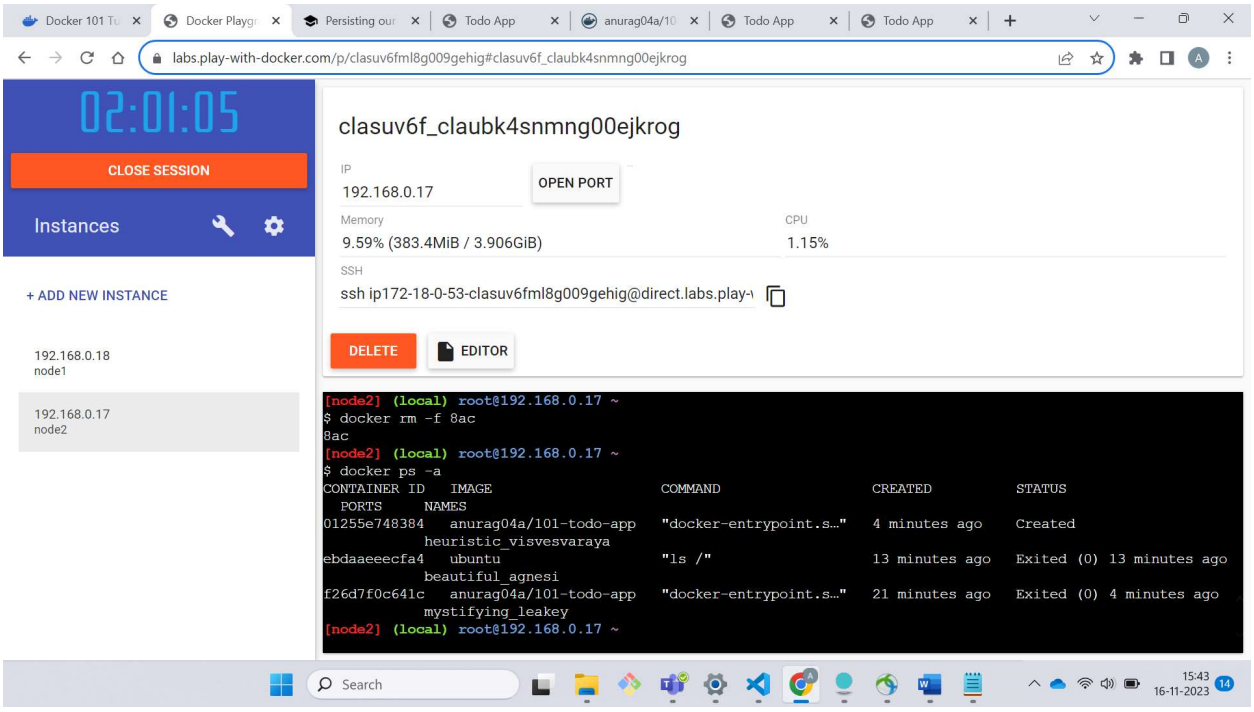
4. Remove the container for the todo app. Use `docker ps` to get the ID and then `docker rm -f <id>` to remove it.
5. Start a new container using the same command from above.
6. Open the app. You should see your items still in your list!
7. Go ahead and remove the container when you're done checking out your list.

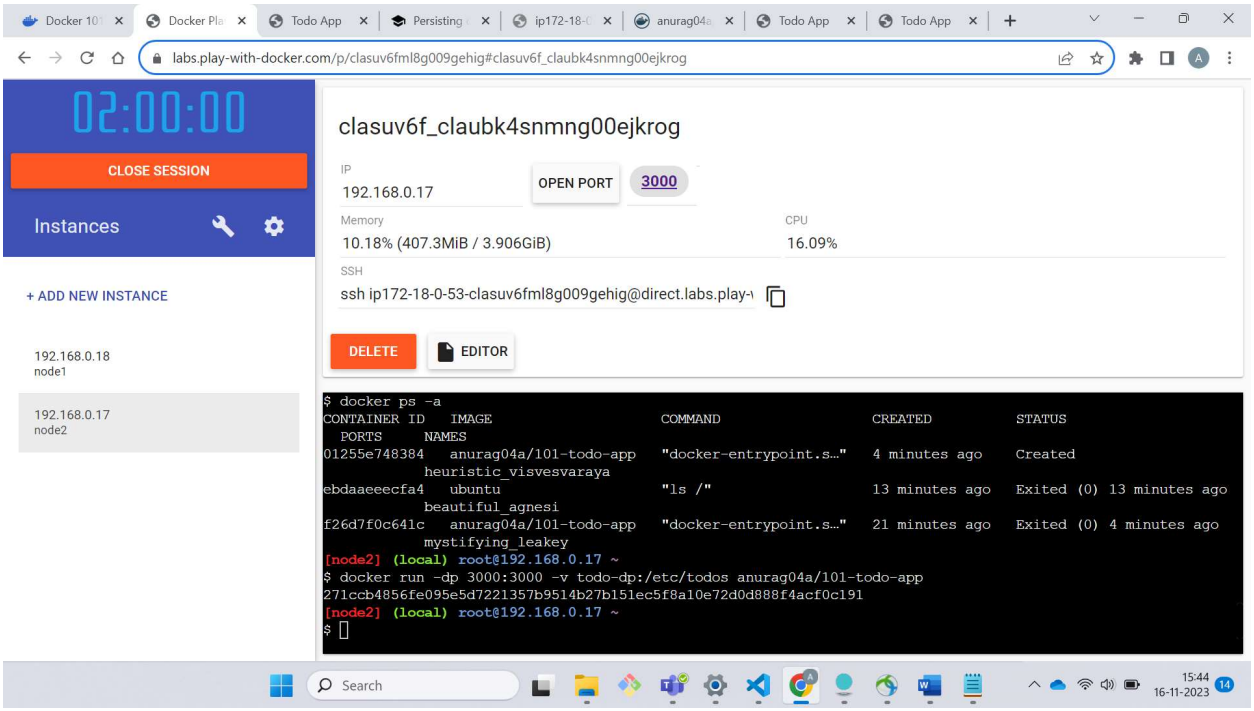
Hooray! You've now learned how to persist data!

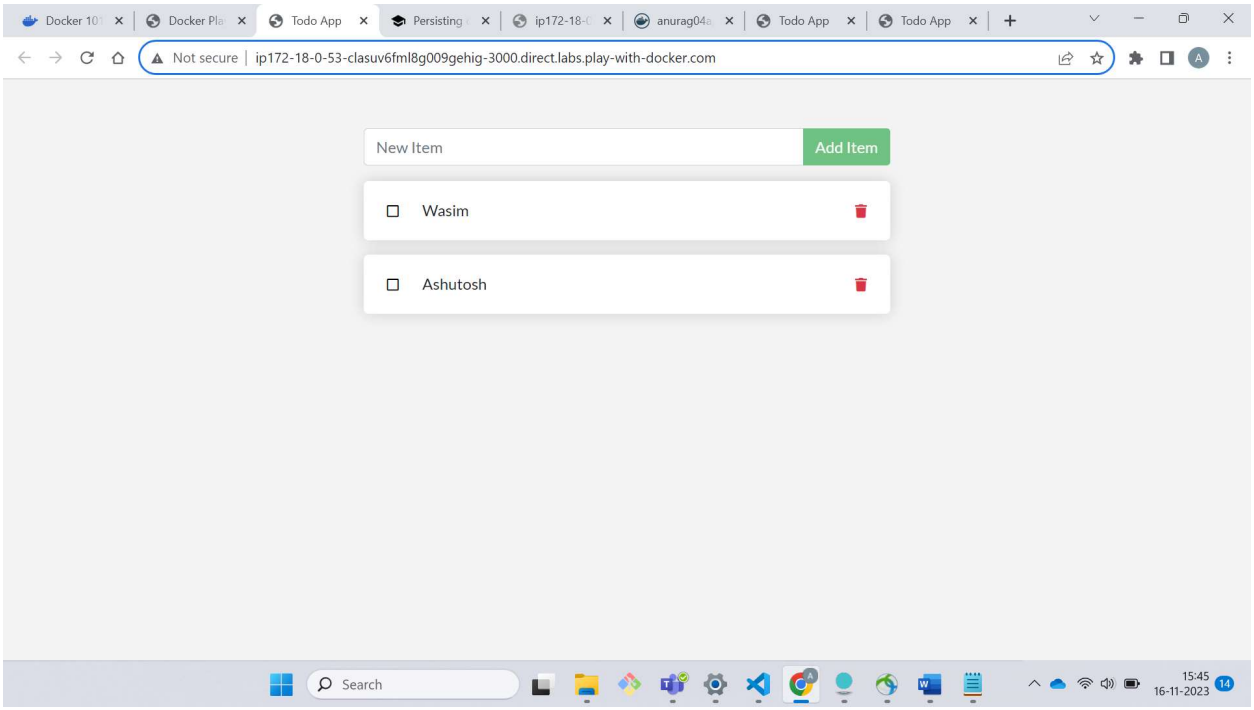

Docker 101 Tutori
Docker Playgroun
Persisting our DB
anurag04a/101-to
Todo App
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
02:08:50
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
3000
Memory
10.22% (409MiB / 3.906GiB)
CPU
0.19%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node2] (local) root@192.168.0.17 ~
$ docker rm -f 2802
2802
[node2] (local) root@192.168.0.17 ~
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
f26d7f0c641c anurag04a/101-todo-app "docker-entrypoint.s…" 12 minutes ago Up 12 minutes 0.0.0.0:30
00->3000/tcp mystifying_leakey
[node2] (local) root@192.168.0.17 ~
$ docker volume create todo-db
todo-db
[node2] (local) root@192.168.0.17 ~
$
Search
15:36
16-11-2023

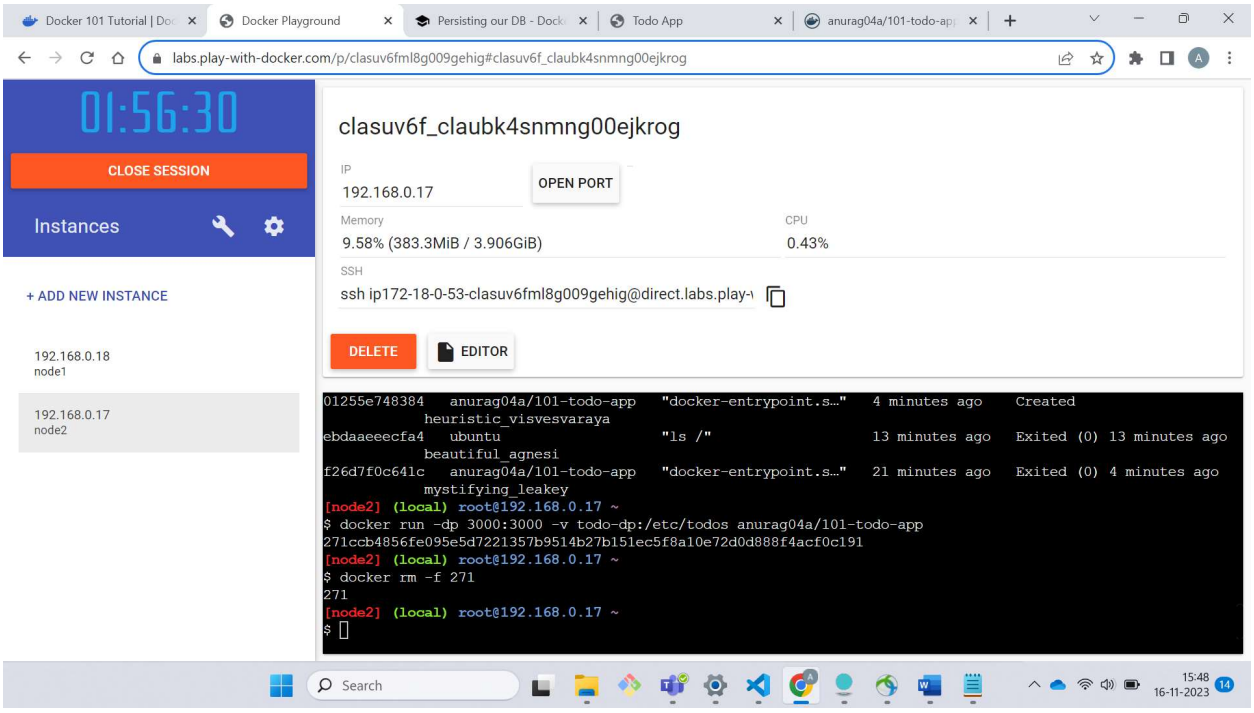

Docker 101 Tutori
Docker Playgroun
Persisting our DB
anurag04a/101-to
Todo App
Todo App
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
02:04:55
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
3000
Memory
10.16% (406.4MiB / 3.906GiB)
CPU
15.11%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node2] (local) root@192.168.0.17 ~
$ docker run -dp 3000:3000 -v todo-dp:/etc/todos anurag04a/101-todo-app
01255e748384a81210b42300b35d383d857c2db77e92f713f12d36f956e4363e
docker: Error response from daemon: driver failed programming external connectivity on endpoint heuristic_vi
svesvaraya (232e9af21052876f2598ac8ae1a03e9d02d0383389d898163a4d500639e5f6c4): Bind for 0.0.0.0:3000 failed:
port is already allocated.
[node2] (local) root@192.168.0.17 ~
$ docker container stop f26
f26
[node2] (local) root@192.168.0.17 ~
$ docker run -dp 3000:3000 -v todo-dp:/etc/todos anurag04a/101-todo-app
8ac74cdddb477a28f7fe7bfb1d7dccbfa7083224fe5b9286dce8e0fafb8e2eff
[node2] (local) root@192.168.0.17 ~
$
Search
15:39
16-11-2023

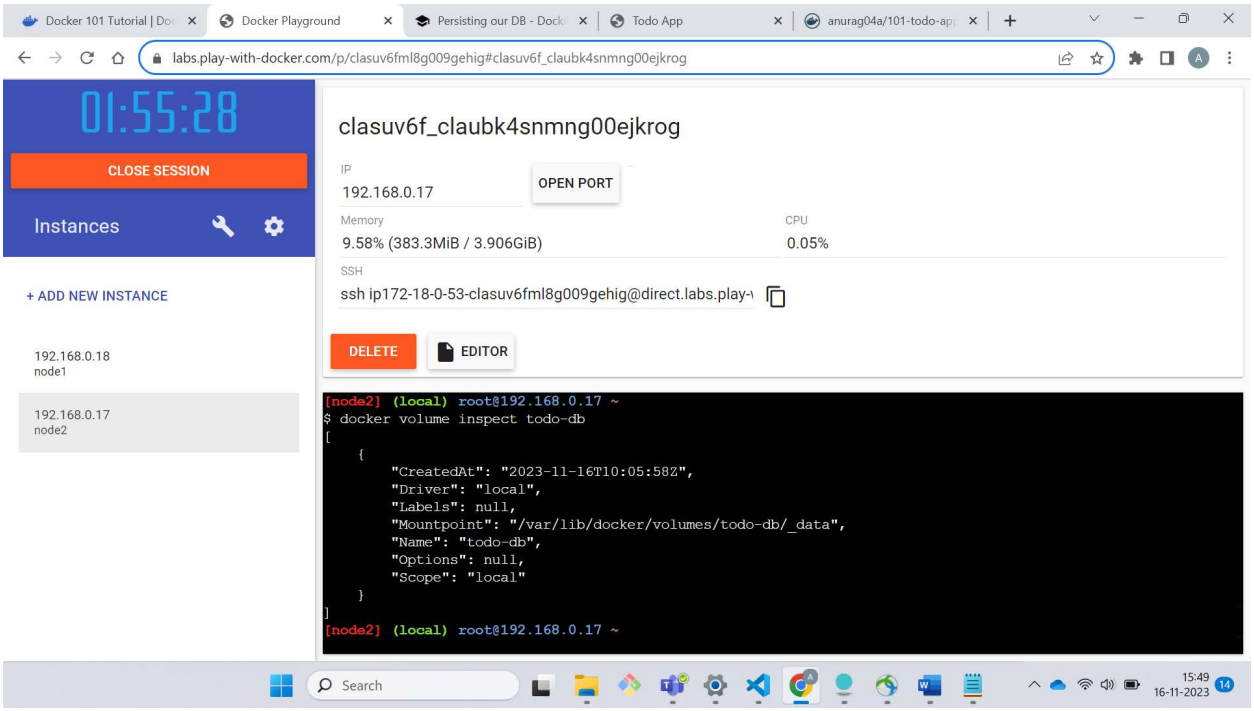

Not secure | ip172-18-0-53-clasuv6fml8g009gehig-3000.direct.labs.play-with-docker.com
New Item
Add Item
Wasim
Ashutosh
Search
15:41
16-11-2023

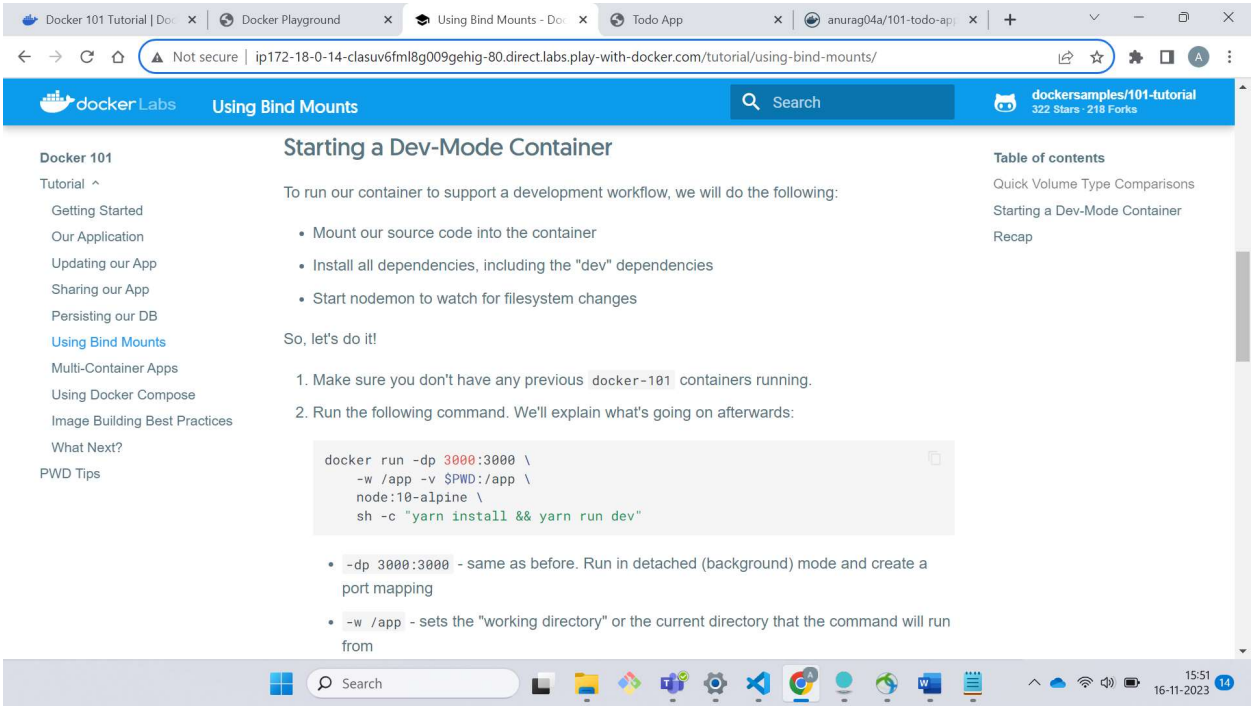

labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
02:01:05
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
Memory
9.59% (383.4MiB / 3.906GiB)
CPU
1.15%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node2] (local) root@192.168.0.17 ~
$ docker rm -f 8ac
8ac
[node2] (local) root@192.168.0.17 ~
$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
01255e748384 anurag04a/101-todo-app "docker-entrypoint.s…" 4 minutes ago Created heuristic_visvesvaraya
ebdaaeeecfa4 ubuntu "ls /" 13 minutes ago Exited (0) 13 minutes ago beautiful_agnesi
f26d7f0c641c anurag04a/101-todo-app "docker-entrypoint.s…" 21 minutes ago Exited (0) 4 minutes ago mystifying_leakey
[node2] (local) root@192.168.0.17 ~
Search
15:43
16-11-2023


labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
02:00:00
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
3000
Memory
10.18% (407.3MiB / 3.906GiB)
CPU
16.09%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
01255e748384 anurag04a/101-todo-app "docker-entrypoint.s…" 4 minutes ago Created
heuristic_visvesvaraya
ebdaaeeecfa4 ubuntu "ls /" 13 minutes ago Exited (0) 13 minutes ago
beautiful_agnesi
f26d7f0c641c anurag04a/101-todo-app "docker-entrypoint.s…" 21 minutes ago Exited (0) 4 minutes ago
mystifying_leakey
[node2] (local) root@192.168.0.17 ~
$ docker run -dp 3000:3000 -v todo-dp:/etc/todos anurag04a/101-todo-app
271ccb4856fe095e5d7221357b9514b27b151ec5f8a10e72d0d888f4acf0c191
[node2] (local) root@192.168.0.17 ~
$
Search
15:44
16-11-2023


Not secure | ip172-18-0-53-clasuv6fml8g009gehig-3000.direct.labs.play-with-docker.com
New Item
Add Item
Wasim
Ashutosh
Search
15:45
16-11-2023


Docker 101 Tutorial | Do
Docker Playground
Persisting our DB - Dock
Todo App
anurag04a/101-todo-ap
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
01:56:30
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
Memory
9.58% (383.3MiB / 3.906GiB)
CPU
0.43%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
01255e748384 anurag04a/101-todo-app "docker-entrypoint.s…" 4 minutes ago Created
heuristic_visvesvaraya
ebdaaeeecfa4 ubuntu "ls /" 13 minutes ago Exited (0) 13 minutes ago
beautiful_agnesi
f26d7f0c641c anurag04a/101-todo-app "docker-entrypoint.s…" 21 minutes ago Exited (0) 4 minutes ago
mystifying_leakey
[node2] (local) root@192.168.0.17 ~
$ docker run -dp 3000:3000 -v todo-dp:/etc/todos anurag04a/101-todo-app
271ccb4856fe095e5d7221357b9514b27b151ec5f8a10e72d0d888f4acf0c191
[node2] (local) root@192.168.0.17 ~
$ docker rm -f 271
271
[node2] (local) root@192.168.0.17 ~
$
Search
15:48
16-11-2023
14


Docker 101 Tutorial | Doc
Docker Playground
Persisting our DB - Dock
Todo App
anurag04a/101-todo-ap
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
01:55:28
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
Memory
9.58% (383.3MiB / 3.906GiB)
CPU
0.05%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node2] (local) root@192.168.0.17 ~
$ docker volume inspect todo-db
[
{
"CreatedAt": "2023-11-16T10:05:58Z",
"Driver": "local",
"Labels": null,
"Mountpoint": "/var/lib/docker/volumes/todo-db/_data",
"Name": "todo-db",
"Options": null,
"Scope": "local"
}
]
[node2] (local) root@192.168.0.17 ~
Search
15:49
16-11-2023


Docker 101 Tutorial | Doc
Docker Playground
Using Bind Mounts - Doc
Todo App
anurag04a/101-todo-ap
Not secure | ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/using-bind-mounts/
docker Labs
Using Bind Mounts
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
Starting a Dev-Mode Container
To run our container to support a development workflow, we will do the following:
Mount our source code into the container
Install all dependencies, including the "dev" dependencies
Start nodemon to watch for filesystem changes
So, let's do it!
1. Make sure you don't have any previous docker-101 containers running.
2. Run the following command. We'll explain what's going on afterwards:
docker run -dp 3000:3000 \
-w /app -v $PWD:/app \
node:10-alpine \
sh -c "yarn install && yarn run dev"
-dp 3000:3000 - same as before. Run in detached (background) mode and create a port mapping
-w /app - sets the "working directory" or the current directory that the command will run from
Table of contents
Quick Volume Type Comparisons
Starting a Dev-Mode Container
Recap
Search
15:51
16-11-2023

- `sh -c "yarn install && yarn run dev"` - the command. We're starting a shell using `sh` (alpine doesn't have `bash`) and running `yarn install` to install *all* dependencies and then running `yarn run dev`. If we look in the `package.json`, we'll see that the `dev` script is starting `nodemon`.

3. You can watch the logs using `docker logs -f <container-id>`. You'll know you're ready to go when you see this...

```
docker logs -f <container-id>
$ nodemon src/index.js
[nodemon] 1.19.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] starting `node src/index.js`
Using sqlite database at /etc/todos/todo.db
Listening on port 3000
```

When you're done watching the logs, exit out by hitting `Ctrl + C`.

4. Now, let's make a change to the app. In the `src/static/js/app.js` file, let's change the "Add Item" button to simply say "Add". This change will be on line 109.

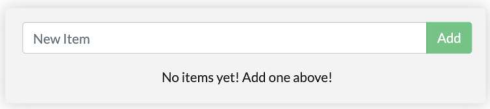
```
-                    {submitting ? 'Adding...' : 'Add Item'}
+                    {submitting ? 'Adding...' : 'Add'}
```


Docker 101 Tutorial | Do | Docker Playground | Using Bind Mounts - Do | Todo App | anurag04a/101-todo-ap
Not secure | ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/using-bind-mounts/
docker Labs | Using Bind Mounts | Search | dockersamples/101-tutorial 322 Stars · 218 Forks

4. Now, let's make a change to the app. In the `src/static/js/app.js` file, let's change the "Add Item" button to simply say "Add". This change will be on line 109.

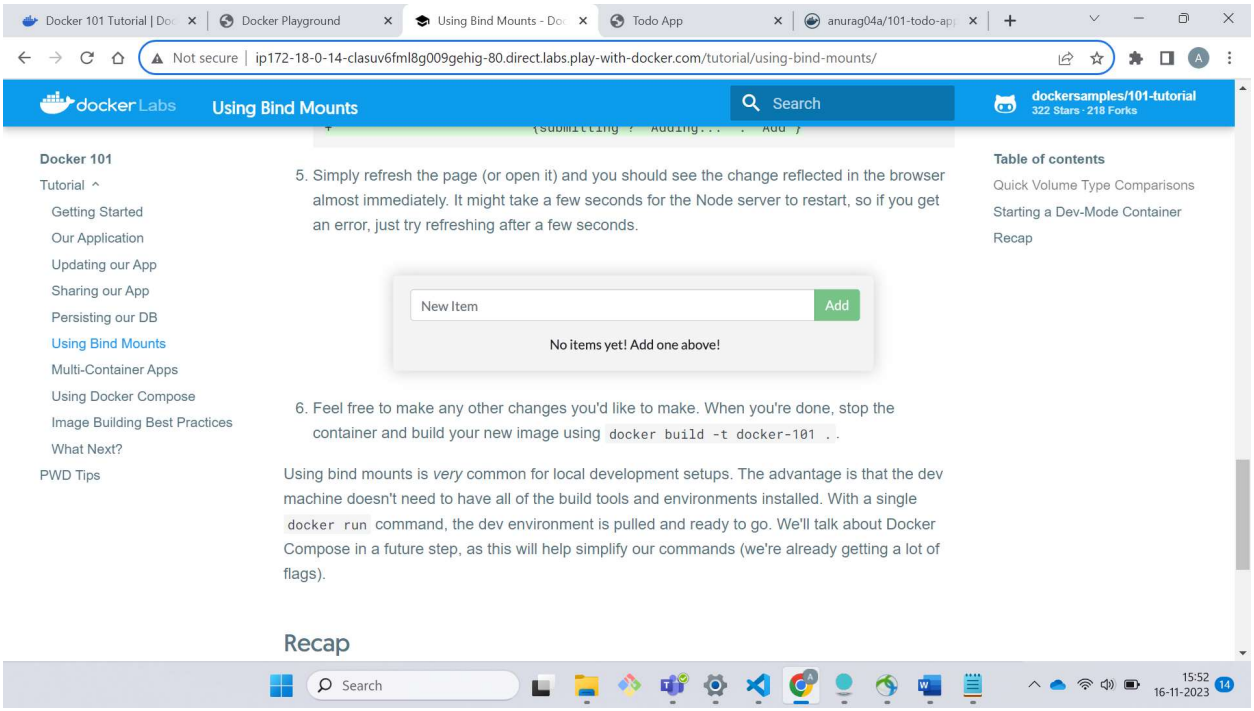
```
-                    {submitting ? 'Adding...' : 'Add Item'}
+                    {submitting ? 'Adding...' : 'Add'}
```

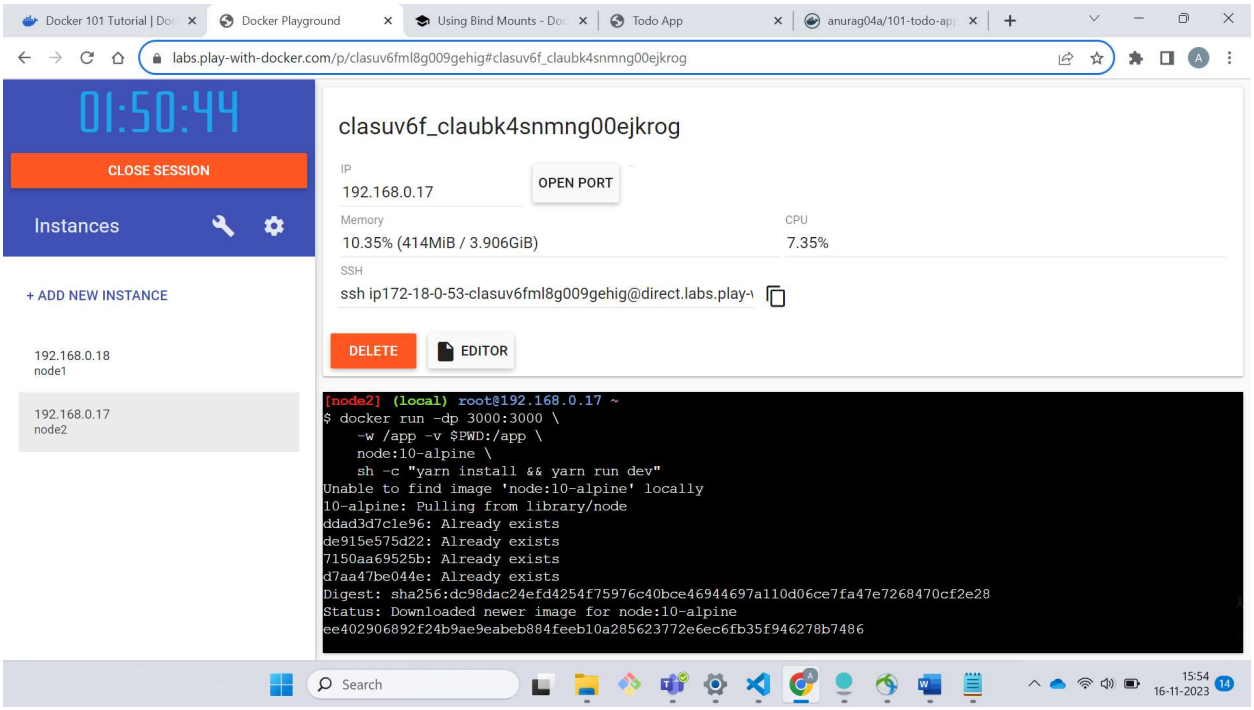
5. Simply refresh the page (or open it) and you should see the change reflected in the browser almost immediately. It might take a few seconds for the Node server to restart, so if you get an error, just try refreshing after a few seconds.



6. Feel free to make any other changes you'd like to make. When you're done, stop the container and build your new image using `docker build -t docker-101 .`.

Using bind mounts is *very* common for local development setups. The advantage is that the dev machine doesn't need to have all of the build tools and environments installed. With a single `docker run` command, the dev environment is pulled and ready to go. We'll talk about Docker


Docker 101 Tutorial | Do
Docker Playground
Using Bind Mounts - Do
Todo App
anurag04a/101-todo-ap
Not secure
ip172-18-0-14-clasuv6fml8g009gehig-80.direct.labs.play-with-docker.com/tutorial/using-bind-mounts/
docker Labs
Using Bind Mounts
Search
dockersamples/101-tutorial
322 Stars · 218 Forks
Docker 101
Tutorial
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?
PWD Tips
5. Simply refresh the page (or open it) and you should see the change reflected in the browser almost immediately. It might take a few seconds for the Node server to restart, so if you get an error, just try refreshing after a few seconds.
New Item
Add
No items yet! Add one above!
6. Feel free to make any other changes you'd like to make. When you're done, stop the container and build your new image using docker build -t docker-101 .
Using bind mounts is very common for local development setups. The advantage is that the dev machine doesn't need to have all of the build tools and environments installed. With a single docker run command, the dev environment is pulled and ready to go. We'll talk about Docker Compose in a future step, as this will help simplify our commands (we're already getting a lot of flags).
Table of contents
Quick Volume Type Comparisons
Starting a Dev-Mode Container
Recap
Recap
Search
15:52
16-11-2023


Docker 101 Tutorial | Do
Docker Playground
Using Bind Mounts - Do
Todo App
anurag04a/101-todo-ap
labs.play-with-docker.com/p/clasuv6fml8g009gehig#clasuv6f_claubk4snmng00ejkrog
01:50:44
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
192.168.0.17
node2
clasuv6f_claubk4snmng00ejkrog
IP
192.168.0.17
OPEN PORT
Memory
10.35% (414MiB / 3.906GiB)
CPU
7.35%
SSH
ssh ip172-18-0-53-clasuv6fml8g009gehig@direct.labs.play-
DELETE
EDITOR
[node2] (local) root@192.168.0.17 ~
$ docker run -dp 3000:3000 \
-w /app -v $PWD:/app \
node:10-alpine \
sh -c "yarn install && yarn run dev"
Unable to find image 'node:10-alpine' locally
10-alpine: Pulling from library/node
ddad3d7c1e96: Already exists
de915e575d22: Already exists
7150aa69525b: Already exists
d7aa47be044e: Already exists
Digest: sha256:dc98dac24efd4254f75976c40bce46944697a110d06ce7fa47e7268470cf2e28
Status: Downloaded newer image for node:10-alpine
ee402906892f24b9ae9eabeb884feeb10a285623772e6ec6fb35f946278b7486
Search
15:54
16-11-2023