

DevOps - Module 9: Advanced Docker

DevOps

Module 9: Advanced Docker

Video: Docker Data Volumes

Volumes.

So volumes they are saying what?

It's a mechanism for persisting data generated
by the contents and never generated, but it was lost.

So now we want to have persistence.

The concept we use is call us volumes.

And they're saying volumes are essential
for managing storage.

Very important. This line independent
of containers lifecycle meaning
what if container is destroyed?

Remote data still will exist.

Meaning you're separating data, you're separating container
are is what I'm saying.

You're separating both entities, allowing the data
to be persisted even when the containers are deleted.

So in order to do this, what volumes do internally,
volumes are stored on what?

Post file system in the following location.

So any data you do,
it'll not be part of container, rather it is part of
host file system.

So this way two issues are fixed. One is data persistency.
Second issue is container will not be big
because data is not virtually present inside the container.

It is actually present on the host.

Now if you think, isn't it almost comparable to networking?

You routed what you say,

host network, two container network.

Agree with that? Then why shouldn't I do the same thing

for data route host file system

two, container file system issues, result.

That is what Natalie we are doing here.

Are you getting the point is what's the idea here?

Clear. So here, technically

that's why you see they're saying what?

That's the advantage you'll get with docker containers.

Data persistency, meaning the data will process beyond the life of the container

and also sharing the data,

meaning volumes can be shared across multiple containers,

allowing further data to be viewed

and modified by different containers.

Meaning not only data, persistency,

shareability is also part of volumes.

Video: Docker Managing Volumes

That's the combined view.

As of now, no volumes.

I want to create one

create followed by the volume name you need to mention.

No, the volume name, it's your choice.

Maybe something like important

volume mark like that I can give.

That's the path I said no, the host, it will be created.

That is a path. So I

did not give any docker command.

Is it normal pseudo list? Why?

Because pseudo, it's in a different location.

Do you see any data? No, because it's an empty volume.

Now this volume needs to be mapped with our contents.

Simple.

This was the one we created earlier.

Name of the content. I'm adding one more thing.

Additionally, volume. Now the things start like this.

Dash V stands for volume. What volume.

This is the volume I want to attach to this container.

In which location The data is there.

What you want to preserve.

If I say slash, it'll preserve the whole root wall,
which is not required for what is required for me.

You need to anticipate where you are keeping the data.

Location is what important. Underscore data.

So that is where I'm gonna start using
the what you say, storage.

Meaning. Now I don't need to create a directory.

Whenever I say create a volume implicitly, it'll create
that path inside the container
and it'll map the path to post.

How did you say hyphen P in the last session?

Post port colon, container port.

Agree or not Last class. Uh, syntax. Now what?

Volume. Colon path inside the container.

Okay, everyone. This is where Inside container.

And this is on the post. Just like last class.

Are you good? What I'm trying to say?

So technically where is this location?

This will be ma to this.

Let's see. Once I create, you might get an idea.

Now I have two containers.

Now earlier with this I needed

to create, but did I create here?

No. Right? Did I log in

and say important data, MKDR, nothing like that, right?

Did I create it still came because of volume

and this thing, it is map to where?

On the host. How can I verify means?

I'll show you both. Inspect this one.

Did I create any volume nor it the old one?

Scroll up. You'll find something called as Mals md.

Old container.

Inspect new container.

Is it empty

source and destination

clear is what I'm trying to say.

So that's why

earlier it was empty.

Now also it's empty. I,

I did the same operation.

Where did I do this operation? Inside the container

and where it is reflecting on the host,

that's volume frame.

Now advantage.

Both containers I'm getting rid of.

No contains no, technically data should go on, right?

According to the last scenario, we saw

volume still persists.

That's the nature of volume. See the definition? One second.

Is it doing the same thing?

I deleted the container, but data still persists.

Still exists.

Did you lose the data? That's volume. Fine.

Simple. That's it. Now,

in the last class we did the application, right?

Uh, login app. So we had issue.

Whenever we are going and keeping the data,
installing the app, it's going off, isn't it?
But can I replace that with volume?
Why not how I used and text file? I'll use my code.
Now technically it is possible.
Let's give it a try once if it works or not.
You understood the concept now everyone what I'm saying?
Volume concept. You got it. Uh, sharing.
Also, I'll show you because with this example
it does not make sense.
The next part what I'm going
with the application makes sense.
So now
if I do like this, will I get volume?
No. So I need to create,
now I did the volume creation explicitly enough
by saying volume create it is optional.
If you create the name, sorry,
if you give the name here directly will create
also hyphen V.
Let's say
do we have volume?
No rate with this name, no issues.
It'll create colon.
Now where should be the
path inside the container?
Where can I give like this?
Let tell, create and see what happens.
It's working. But now
if I take that ip,
it's not loading, right?
Oh, see now what is coming up?
But is it loading my log enough?
Because what I need to do the work where,

okay, tell me like what you know, if I want my app
to be running up, what did I do in the last session?
Docker container.

Execute actively on this container.

Bash, isn't it? Now where I need to do the work.

First thing I need get
which location.

If I want the data to be
persisted, where should I keep the data?

According to your logic. That's it.

Now important data is where you have set the
volume I went and kept.

So I have created wrong volume
where the data should be mapped.

Technically

This is the place where ingenix loads the application,
but where did we update it?

Important data. So that is very important.

You need to figure out what is the appropriate path
inside the container to load the data from.

Now I kept important data.

It does not make sense because Ingenix
does not read important data.

Nix read what data user share.

Gen X, HT M makes sense, right?

Everyone. Uh, now in that particular regard, the correct way
to do is what

I clearly is what I'm saying.

I'm saying map the volume with this particular data
would clear everyone what I'm saying.

Yeah. Okay.

So, uh, we are creating container
with login app one using,
Oh, sorry. Oh, go ahead. Yes.

Uh, and also we are creating a volume login volume
and we're share sharing using that uh, path.

That's it, right?

That's it. Yes. Technically we are saying

NX document route is this one.

So I'm mopping the NX document route with the volume name.

But how do we know where the, where, I mean
how do we know the path there?

Yesterday we discussed, how do I know?

How did I get to know that is user sharing.

X-H-T-M-L Answer is in your registry.

Okay.

Okay. Got it. Right.

So everything in the emails they're already giving you
from there, we need to pick it up.

Okay? Right.

So now we already created a,
we're creating a container, right?

If we delete the container, it'll be there in the volume.

Exactly. Again, can we get uh, uh, back
to the container using the volume?

That's what we are going to do. Yes.

That's what we are demonstrating right now.

So once I do this, elaborate, you'll be getting it Same out.

Okay, so now I did that. Right Now what I'll do,

So

Delete it, cl everything
we go that app.

Now I want to test
remote, but what will exist?

Is it there? Next time I'm not going to do the same thing.

See now if I go ahead and check, it'll not be there
because gone,

I did not do anything.

Just relaunch the container. But what was the volume name?

I have given all the data is there.

Now the data is getting loaded. Where?

Into user sharing the next system.

Now, same way. I want one more copy
of this application on 8 0 8 0.

Do I have it?

I'm sharing the data across multiple s.

My network is bit slow.

Seely,

oh that word?

Eight. Eight

network problems.

I think a lot of people connected to this,
but technically it is working.

I need to verify also,

Nancy data is also being shared across multiple contexts.

That is what volumes are used for.

And actually this, I'm not saying
your official Docker says let's go to official page
V.

Go with user sharing in x html.

This arrow O means read only so
that you should not modify the conduct fixed content.

And same concept. Like right now what we did
and now with this, I no longer need
to perform the same activity.

Once again, that's what you wanted. Now.

So if I delete the container, will I lose the data now? No.

That is the concept of data persistence.

This is one approach. There is also one more approach called
as bind mounts.

We also call them as uh, host volumes.

Technically these are called as named volumes.

Now why we use bind bins for live reloading?
You have some development work going on.
Continuously changes are being done.
So I don't want to perform the operations in the container
on the host level itself.
I want to do, it should reflect
it's called as bins.
So what we did right now from container two host vice versa,
host two container.
It's called bins. The reason for
that is right now I created the name of volumes.
Now what I went with this syntax
for them will not get control.
It is in which location? Where?
Lip docker something source that is not under your control.
It is only under the docker control.
I want my control for the containers I'm creating.
Then in that scenario, you will go with bin modes.

Video: Docker Bind Mounts

I'll tell you the problem statement.
Currently the site is saying what? Edify. LMS login, right?
Some other client came.
So I'm going to make modify some changes,
permit to the change.
No, I want this change to be reflected.
No I cannot. Are you getting the point is
what I'm trying to say Because this change, if it needs
to reflected where it needs to be done inside the contain.
If I want update here, what should I do?
Agree or not?
You got the data not GI again, I need to install,

which is a problem statement.

And I don't know when this thing will be there.

When this thing will be gone, it's fine. Okay?

If it does not work here, we know this volume is mapped
to some location on the host.

There is a path

it so is not working normal way.

It is not, it don't have the permissions
to manage docker volumes.

Now in this scenario, the solution is buy mobs.

Are you clear what I'm saying? So I cannot do anything.

Now it is fixed.

If I want to make change again, install it again,
do the pull and do all the outpatients.

Clear what I'm trying to say.

So here what now I want to use bind modes only.

Small difference is syntax.

Uh, you see now I'll do this app this way.

Do I have GI on the system?

I thrown it. Where?

On the system, on the host.

Do I have control over here?

Can I do all these operations yesterday? No problem.

Now you see how am I going to change the things?

So now obviously there will not be any app
because we took down both the containers.

Take away the name of the volume, give the path
where your data exists.

That is called bind.

Where is a path?

This is called bind mode.

The only differences path
earlier path has been given.

What name? Some name that you have specified.

Here it is Spark. That's the reference.
And when I create you see what happens.
Create it. But interesting thing is
you'll not see because it is not docker controlled volume.
It is user control volume.
So, so user need to docker no longer keeps track of it.
Then how do I know?
Earlier it was volume, now it is bin.
Earlier source was docker controlled.
Now sources user control,
say they're called also host volumes
or user control volumes or bind.
Bind monster official term.
Now if you go back,
sorry, 80, 80, right?
I got same new LMS flow and it is also shareable.
If I want one more copy to run on 80.
Same like how you got the typical characters
and you want to do the live updating and all this thing.
Right now you don't need any hard work.
I'm updating bad
the change.
Can I do it here then update it.
Welcome it's network issue.
Now you got the difference. So simple.
If data is fixed, go with named volumes.
Data changes, updates will be done, go with bin
only differences sentence.
But outcome is same.
And in another way you can say
named volumes are from container to host.
Bind volumes are from host to container.
That's it, right? I'm doing on the host,
it is reflecting in the containers.

Earlier what I was doing on the container,
it was reflecting on the host.
I hope you understood the logic between both the things.
So what use it depends upon the scenario you are working.
If something is under the development, which I need to
frequently modify, rely on by amounts,
no data is fixed, it's not gonna change.
One time it's there. Go with names.
Volumes, okay, clear.
Alright. Now the important thing is yes I did this,
which is good, but the problem is
this volumes whatever we are using buy mons
or name volumes, anything you go with, they are specific
to one system model, isn't it?
Now this volumes data,
everything is there where in the system.
So in the future, what happens?
You cannot predict at what time in the infrastructure
what component might go down.
So currently what happened, like according
to the things we are learning right now,
that's the server we created.
Now all the data is here, agree
or not, which is there in this zone.
Let's say unfortunately this zone went down.
Now our system will not work, right?
So what I'll do, I want to make them portable.
Now do you think these applications will be portable?
No, of course not. Portable means what?
Anywhere It should gimme the same object.
Now what I'll do, like
how you have Docker installed in your system?
No, like in your server. I have Docker installed in my lap.
I said Docker desktop is there.

You can install it in your Windows Macin anywhere.

I have already installed it. I'm starting the docker here.

You see on the top my Docker desktop
is starting.

Just give it a minute.

You see Docker is starting now open one more

GI Bash

and this thing is what?

It's not connected to the system. Agree or not
what I do I'm doing in my system.

You can do it. Also open your GI Bash.

Once you install Docker desktop, you can do the same thing.

Now literally you see guys, what I'm doing means
unable to find engineer.

So it is downloading.

How come there will be this location here?

It'll not be right

home to login.

Where is it inside your server?

How can you expect that path in my system?

So there is no way it's never gonna work. Meaning what?

Volumes cannot be portable.

So always giving you an issue in terms
of going and scaling across different machines.

In that particular scenario, we want everything
to be coming up from mic container only without
any external dependence.

Then you might ask what is the use of volume?

Then I said no.

In certain scenarios, when you want
to quickly load the data into the containers,
then volume is the best choice.

But in this scenario it's not.

Uh, then how you prepare minutes.

Next we will be learning it.

See, for every particular problem there will
be some solution moving forward.

Airline data was not persistent. Now you persistent.

But after persisting one more issue further against some
other solution, earlier you were not able
to access the things networking home
like that step by step here.

So now you understood the concept base.

So now I created, but it is of no use actually.

So now we need to understand how I can make them portable
and run anywhere in the world I want.

With volume, you'll not be able to a hundred percent get it.

So we need to understand new concepts. Got it?

No problem statement everyone. So what was the problem?

Right now we have, I hope you understood the point here.

Clear everyone which.

Video: Dockerizing Applications

In the last session, right, we had the uh, discussion
regarding the buy mon and volumes
and we were clear that how to manage the data using, uh,
what you say, uh, like migrating the data
to the host so that you have access
to the data in case if there is any sort
of, uh, what you say.

Uh, destruction happened with the data,
but eventually, uh, we had one problem.

Though we are creating the data
and managing inside the containers,
we cannot be a hundred percent sure the data
can be reliable on this.

Meaning if the system, what we created that system
by chance it goes down.

Meaning what I need
to have the data created across multiple measures,
which can be little bit complex
are you'll get better solutions in the future also,
but for time being what, instead of hard coding, the things
with the server, we want to separate it out
earlier what you did, the data which is there
inside the container, you're separating out
and you're keeping on the host,
but there can be chances of host going down.

Now I'm saying let's not keep the data in the host somewhere
else, which might not go down
or is what I'm trying to say in that particular regard.

You can go ahead and rely on a concept called as
custom images.

Okay, so when I go with custom images, simple
and straightforward, what is the idea?

You know, so if you see here,
actually the application that I'm looking for is what
some logging app according to the criteria
that you're trying to go ahead
and logging.

2, 5 0 1.

I don't have technical, do I have any such image like
login 2, 5, 0, 1.

Now the idea is I want to create my application as an image.

I already said what is the definition of image?

Image contains everything you want in order to go
and uh, what you say run an application.

It might be OS related components, binaries
and libraries application code itself.

Uh, now what? Let's remove the dependency

with the host system and then create one custom image by yourself and then later on put it in dock.

How did you have GitHub to keep your application code?

Now I'm saying we'll use Docker hub to keep your docker custom applications

or you can also call them as custom images.

I'm not saying Docker itself is saying, let's see the thing.

What did you use in gen next? Now

this is saying what I'll only give in Gen X.

Does it say that? I'll give you some login app.

E-commerce app or some uh, what do you say? LMS app?

No it's not.

Did you use this approach earlier?

Clearness syntax, what type of volume is it

according to their given syntax

bin bond.

How can I say bind mode

Because I have given the slash some content.

Basically it's not the volume name, it's some path.

If you are giving the path according to the definit, sorry,
according to the syntax, it was what Bind mode.

Agree or not. Now you see they're only saying

alternatively

a simple docker file.

You remember Docker file I have was discussing when you're
talking about the networking,

when you go into a specific image post this image

sonar cube image, I said for all these images,

Docker file is a source code.

Where did you see that expose

instruction in the docker file.

You remember networking class?

I just went and I clicked on some specific docker file.

It was showing you all the instructions.

That is a source code of image, right?

So they're saying, why don't you alternatively prepare a docker file, which can be used for what you generate your new image, our own image that includes all the necessary content, which is what?

A much cleaner solution than what say it is recommended by Docker itself.

So rather than going and relying on this volumes and bind modes, you just prepare your own Docker file and then start placing the instructions.

What you want to do with Dock, you remember Shells script?

What was Shells script containing?

You wrote on Shells script, right?

What was it Shells script doing? Technically I

Command Linux commands.

Commands for what? If not,

I'll open that thing for you.

If you're getting or if you've got forgotten that login 2 5 0 1, right?

We wrote that some.

This was a script you wrote, right? Just have a look.

What it is trying to do

Is, uh, installing zip and then nix,

Then followed by Uh, removing the HTML files and it is cloning the login page.

Technically your application is ready or not.

If I do all of it, what is the outcome of the script?

Simply deploying your login app.

I'm saying instead of going into one server and doing all of this work, I'll write script in terms of Docker, it's a shell script.

Now, if I use this shell script, I need one machine.

In that machine only I need to execute. Agree or not.

Now I'm saying do I, can I do something like this

without installing N engine X going
and doing all this activity, can I directly get an package
of log app
are clear what I'm trying to say?
Take LMS application also what you did for LMS application.
You went installed mode, installed NPM, you installed nGenx,
you did the build, you got the artifacts.
Artifacts you loaded into next access.
From there you download into some system.
In that system again you installed engine X,
you remove the old code, you place a new code,
you did all this work, right?
Which was taking two, three hours like earlier
what we discussed and that was the plan to go with Docker.
I don't want to do all of it, start with a small code.
Instead of doing all of it, I want
to prepare an image which will gimme this result.
So how do I tell that means you'll not write a shell script,
you'll write Docker file.
Shell script is what a repetitive action
that you perform on the system.
The same thing I want to do with container.
In that scenario, you need your own niche
and how do you create an arrangements?
The source code is Docker file.
Are you clear? Yes. Everyone, if any one of you wants
to go ahead and take your own custom applications
inside your organization
and you want them to be compatible with Docker,
your goal are your work starts from writing Docker files.
Let's see. They have given you engine estimation,
so obviously there should be a docker file for that agree
or not
and tax you remember tax.

When did we discuss the tax?

What was the concept
question exactly?

The versions in GitHub,
the source code versions were called tags.

Now in Docker it's the same thing. Why?

Because next part I'll show you what is the meaning
of tags in this part There are multiple tags,
but I'm selecting whatever is there in the top here.

You see now you see
here what it's showing me and it is available where
GitHub, it's a repository
and if you're seeing that code means what?

It is source code.

I recall what I'm saying. Now I want you to compare
what we did in that uh, lab where I created four machines.

You created one build server,
you created one release server, you created couple
of deployment servers.

Agree or not amia, what I'm trying
to say, same idea.

This is a docker file which is a source code, which is
they also went and build in some machine.

How did you say NPM run build which will give you artifact.

Here there is a command called docker build.

It'll give you docker image.

Are you good everyone? You did NPM run build there?

No here the equal and command is what Docker build.

The docker build source code is what? This file.

It'll read instructions from this
once the build is done in LMS project.

What you got this folder? You remember NPM brand build it.

What was the outcome of it? This folder here,
the outcome is docker image.

Where did you store the artifact nexus?

Here we'll store it. Where?

Not GitHub Dock registry.

Are it clear? See this is a source code.

It is not there in docker hub.

It is there where GitHub,

but this image what was generated where it is locker rails,

that is exactly what we are going through

and if you see the instructions right this file,

isn't it the same thing as user ad group ad app

to get update, install all those command,

we said app install engine X.

Now this is the very raw version of engine X.

They're going in installing to minimize that size

and everyth this docker file is nothing

but how to install engineers.

Are you clear everyone what I'm trying to say?

Now using this as a use case, we want

to build our own LMS application moving forward.

Goodnight everyone. I hope we are able

to connect all the things.

What we did, we are going to completely scrap them off.

We are completely reintroducing

and we are completely REIT initializing this entire project

with Docker, GitHub,

docker hub build, meh.

What is build? Meh, the Docker host you have here

and I everyone you're able to connect

or not, right?

So now let's see about the process.

We are going to start working with

the same thing I'm saying, right?

The concept we are going is we call this process

as dock rising but also called

as building custom docker images.

Same thing. I just took the snap of that so that
for your reference you can check it, but what is it?
Is actual process doing
earlier?

An image is giving you a general software
in Ubuntu.

These are all general softwares.

Now I want a specific software according to my choice
and can you, I would say, can Docker predict
that someone is going to develop an LMS project,
A CRM project, HRMS project?

They cannot generalize it, right? I hope you understand.

So what they have given the platforms which are required
for building any applications they have given, we use them
and you build your own things.

The idea clear everyone.

So now we are going to do the same thing.

So now what I want to do it, right?

So how, let's see that process.

Just read this couple of statements to once
here.

So why we are creating means because this is the main thing.

I know the process.

I want to optimize it like how I wrote shell script.

Same way. Now I want to create my own instructions
to prepare the things I want.

Video: Docker Push Images

I have Docker in my system.

Now I'm starting the docker here.

I'm gonna open one more new session.

So if everything is portable
and universal means spot, one copy paste should work simple.
Also did the same thing like last class with the volumes.
It did work. It said it failed
because that path was not there
already saying
unable to find because that image is there where uh, in your
server it's not there.
Universal what now you need to do same GitHub.
It's like combination of GitHub and Nexus.
Go back to your registry, to your account.
Account settings,
personal access tokens generated token.
I got this token now. Now you see what I do?
Call Copy that command.
Go to docker.
I think this got disconnected.
One second. I'll reconnect with the server
because I switched the network
password.
It is signal. Don't give the password.
You need to give the token just like GitHub.
Login success
GI push.
Docker Bush.
Last update was nine days ago,
just less than a minute ago.
I have this image updated.
Now this image universal I can use anywhere
clearly is what I'm saying.
Now I'll just switch back to my thing.
Now this image I pushed to the docker register.
This is my local system.
See we just downloading that image from the docker registry,

it's all slow because of network issues.

If not, it would be like done in instant. Actually,

I'm not pushing the repository to GitHub.

I have updated it in the docker registry.

Later on I will push this Docker file to GitHub

because for that application the source code will be in the

GitHub, whereas image will be in the

Docker network problem.

So it's like super duper slow

but generally it'll be done faster.

Now, done right? I created

on my system, it is nothing on my system.

There was no code, no engine X, nothing, right?

Because it's my system. I'm using local host Colin.

Now it's portable. I can literally

run anywhere in the world.

No need for you to install Ubuntu operating system.

No need for you to install nGenx, no need for you

to install gate, don't need to clone, go

to the document road, execute that command.

Anywhere in the world you get login app wherever you want.

That is why we go with customization.

Now it is available in the registry, right? As of no public.

If you want to go with private, you need to pay.

That's where their cash, I think \$6 per month or something.

But now what? This app,

I can literally run anywhere in the world I

So now do you think next time onwards, if I want to go ahead

and run any dependencies there with any system,

just give this image your application box

here is what I'm trying to say.

Actually in the real world with Docker, this is your mail.

They'll give you GitHub repositories.

You need to understand how that project works.

Then you need to start building docker
images for the projects.

Meaning what do you do?

Just for understand

That's how it'll be there
and any change comes up.

You rebuild that part.

How do you manage effectively tomorrow show.

But now you've got an essence
and see literally I was explaining you,
but all these concepts are already there in Docker website.

I'm not out of the box thinking or something like that.

This your next page. Now official page. Scroll down.

First they said what? This is how you need to do. We did it.

Next what they said,
tell me, did I do anything different?

They copied their public static directly.

Something we said not
because everything is there in the current path
and next commands you sees Some
random name they have taken, I have taken my account ID
with that name we have tagged.

Then there are a container. Did we do the same thing?

That's what we did.

Now if you go back
success, you wrote the docker
file, you build the image, you push the docker file
to GitHub, you put the docker image too Docker,
and then you created a container
verified in your system server
working verified in my laptop.

It is working. You verify in your laptop of windows
or anywhere else also it will work.

And what I'm trying to say is that is exactly

what you work in terms of dockers.

It's a very simple scenario to
get the concept in a clear way.

Next, what I'm going to do, I'll prepare docker files
for my LMS project.

If you work with CRM project docker files, if you work
with e-commerce project, docker files,
developers will only give a code.

You'll prepare the environment in the form of doc.

This is one part of the work.

Later on, other tools will come next.

Clear everyone, I hope.

Now you're good with the concept,
what we discussed now the same thing.

What we did literally here,
I'm gonna keep all these instructions for you
along with all the instructions I am giving you here.

Just follow them step by step.

Alright? Just change your id.

I have given relevant to some other repository.

Go with your repository.

You already have your own repository screen, right?

Just go with it. Why default it fails.

How to write the docker file, what instructions, everything,
what each command does, step by step, everything is there.

Then later on, finally go ahead
and start pushing those images.

You'll be good, clear as everyone.

I hope you got the point here.

And finally run your container
and verify This is a very simple app
because I said right, I want you to understand the prompt.
Of course our docker file will be complex once we
go into the L LMS apps.

Complex in the sense simple.

You need to understand the flow,
you know the login app flow, you created it.

You need to know your CRM app flow.

Then you can create, now we know LMS app. Flow it.

Next we'll start creating for LMS app.

Good Everyone, yeah, clear it from online.

Any questionnaires. I hope you all understood the concept,
what we just went with right now.

Any confusion regarding this?

Only one, one question I would like to ask is

Ingenix, uh, what is the role?

Is it play here? It'll as a web server only
or any other, uh,

That's it. India

next purpose is to basically host websites

Only hosting website.

Yes. That's the reason why we call it as a web server.

And uh, it'll, it can act as a load balancer as well.

Can we configure it in that way? Or you,

You can make server, sorry, engine X has a load balance
as system Yes possible.

But here I don't have that use case.

Okay? First as what we are going
to use the load balances from the cloud itself
in the Kubernetes, you'll see all of that stuff
because in productions we need load balances,

Okay?

Right? Production part.

When we go there, you'll understand why
load balances are required.

What is the need for all of them?

That is purely Kubernetes part there.

We will discuss all of it.

Okay? Just go ahead, try to do that for your application and verify it in the next class.

Tomorrow I'm gonna start working towards few more concepts.

Are there, uh, like uh, tags and all of it and then compose multiple things.

Once that part is done, then finally we will end up with the dock rising elements.

I said, right? That was what the goal was.

I think like though we were discussing all these internal components, everything,

what was the main concept of documents?

If I go back here,

I'm gonna prepare majors simple so that I can reproduce and run anywhere in the world.

Like that's the idea. Next, I'll show it.

Once I am done with the part,

I'll quickly create the containers and just within minutes I'll be able to run that application anywhere in the world.

Earlier what it was like a lot of time for you to go and do all of that, right?

You are no longer going and now you are like almost close to the things.

Now let's go back to the docker commands.

I have executed for it earlier and tell me now if you're facing any challenge.

I think static code analysis where I did the docker, right, check the command and let me know pseudo means anyway, you know why we need pseudo, but after psdo, you see anything that is not clear for you in that Pentax, except that restart always is nothing.

But whenever the system is stopped, containers are stopping.

No, when the system is on, containers should start.

You need to add that additional tag. Restart always.

If I remove that, restart always everything is same.

Detach mode is always running.

But what happens when your system hardware is shut down?

Obviously it'll shut down.

So when the system is shut down, you want container to automatically start, then you will add restart.

All. If you don't add it, container will stop when you shut down the system.

So explicitly you need to start.

If you don't want add that restart.

Now you see you're able to clearly understand the flu.

That's what I said, like in the initial part.

Also you're able to understand it once you understand dock, right, which that's, this is what I wanted to go with Next.

I'll start working with, uh, what you say tags compose, custom dock images.

Three, four concepts are still there.

We'll discuss and then later on few concepts will also come in Kubernetes at that point of time.

Again, we'll go ahead and recollect them back, right?

Uh, we'll catch up tomorrow with our remaining Docker concepts, right.

Video: Docker Tags

Docker tags are basically labels which are assigned toward docker images.

They're used to provide human readable and user friendly names to the images, making it easier to identify and refer.

Are reference. Different versions are variations of it.

Particular are same image, LMS 1.1,
LMS 2.1, LMS 3.1 here,
image colon, tag one, tag two, tag three.

A docker image can have multiple tags.
Each representing a different version are
configuration of the image.

Okay, clear what I'm trying to say.
Each version, that's green color,
red color version was what we discussed.

But functionality point of view it might be different.
But here what each tag represents a different configuration
are a different codes.

End of the day, and I'll show you one second,
format goes like this, I'm like, it's your convenient
how you want to do the things anyway.

And I'll show you, how am I going to use the same thing
for our demonstration purpose?

That's a format, image, name,
colon tag.

That's how the name will be there
and that's how tag will be present.

And in here,

This is one way you can represent image, name, colon tag
are some user friendly.

Strings are like text also you can just go ahead
and give it your choice
how you want to go ahead and start working.

So in our particular part earlier to uh,
and this a comment, build image
tags, clearness.

I hope you understood the logic like that.
You can go ahead and create the multiple tags for your image
comparison perspective.
GitHub tags you already know which refer to source code.

Next is also tags were there, which refer to binary code in docker tags refer to directly applications with all the required things to run the software.

O is binaries, libraries, code, everything is included.

So now with this word nexus can become an optional layer moving forward.

Nexus fine. Good, but it is only giving me code.

What is the use doctor is saying forget about code.

I'll give you everything you need.

Making it much can be clear.

Uh, now I'm gonna take the same demonstration and I'll try to show you.

I'm gonna take some tags like this.

We have login app. Now in that I'm gonna create three tags and the relevant tags.

I'm gonna start working with docker roads. Good.

What I'm trying to say, I'll show you.

Log 2, 5 0 1. Right?

That was the thing.

This was the docker branch I created.

Now, now you see what was the default uh, thing which was there in our page with the welcome screen.

LMS login. This is like the default.

Default tag name is called latest.

If you don't specify anything, always there tag is latest.

How you can verify that.

Let me connect with the server for you.

Latest. Uh, this is like absolute image because once you delete that container, it's no longer there.

See latest this is the version you selected. 22.04.

If not, it'll come. Latest. You remember.

Now how did I say Ubuntu? 22.04. That was a tag.

Now we are creating. Now
let me go ahead and showcase here.

1.1 is admin 2.1 is trainer 3.1 is
student edit.

What is 1.1? Admin.

Admin. LMS. Login.

Admin login. Commit to this docker branch.

Remember the lab tags create a new release.

What is the version?

Which branch?

Just something for your terms.

Got the idea.

Same way. Talk on
branch going with index file,
how it is coming up like for your applications,
you software lms, same.

Now earlier for this you took that code,
went into some other server, did some other operations,
then stored into some other server,
then deployed into some other.

Now you see this is not our work.

Technically it's all developer's work from there.

What we take and we build the images.

Now I'm going to do the same simple.

You see that's my docker system.

Uh, so let's assume there is nothing on
this system as of now.

Brand new docker system

and I'll

go like this, double it.

Copy link address.

All the three versions of the code base came up.

I think Docker file already is there.

I guess Docker file is already there

because from the docker branch only created the things.

Now no, no issues

because already Docker file has all the instructions ready.

Now earlier, how did I do the build

log?

This is latest but I don't want it.

How did you sync there? I'm syncing here if you want.

Also I can log in into docker registry

for your confirmation.

Uh, dot doc.

I think it's logged out because I logged out

of the system in the last session.

But anyway, I can show you

slash login.

2 5 0 1

za, only one, which is latest stack.

Okay, let it be. Now I'll go back.

Dot represents the current director.

This is separate code.

They have repetitive actions.

All the versions I want as applications.

They are there right now.

Simply if the advantage is what,

even if I don't have the system here,

everything is cleaned up also no issues

because now along with code artifact everything is present.

If not, I can do like this. Once again, everything is there.

Here, no

got rid of everything.

And for confirmation also, I'll do one more thing like this.

Removing all the containers

idea or name anything will work out

if a brand new system agree or not.

No images, no containers

and no code.

Now I want then these three copies of application for
non 80 80
admin rate.

Full command is for downloading the image if you want,
but optional you can directly say
and also Google, see which tag I have
given 1.1.

How convenient is compared to how we did the things
in the previous sessions.

Everything is version controlled tagged
artifact available in docker registry
apps running in small sized advancements.

Same thing. You literally took almost like one hour on the
tip and with so many servers created all
of now everything I got with one machine.

Easier not lot of effort required from your side.

When you want to repeat the same thing next time,
now everything is there right in the image
only wherever you want.

This pull done,
pull in the sensor running the container clear.

I hope. Now you understood it's the same thing
what I did here, right?

You guys remember maybe, I think I kept in a document also.

Anyway,
there's the same document.

I said GitHub tax. These are coming from here.

Same thing. Almost took one hour for you
to prepare all the things and get ready.

But now just like hardly two, three commands.

Good. Everyone clear That is the purpose
of working with tax.

Maintain, keep them whenever you want.

Pull whatever the version you want.

Deploy right from online.

Clear is everyone. So I hope you understood the concept of how things are working in, uh, what you say.

Terms with respect to docker tax.

This is one of the advantages of using Docker.

Again, oh, another advantage.

Now is it faster than this model or not earlier.

It was one hour now I would say hardly five minutes.

Next time.

Video: Docker Compose

There is a concept called as Docker Compose.

I'll tell you where it can be helpful again, how it can help you go with the things.

See what it is saying
says it's a tool like Docker.

Compose is a different application. Docker is different.

Compose is different. Compose uses
Docker to get the work done.

Compost is a tool for defining and running work.

Multi container docker applications with compost.

We can use YAML files
to configure your application services,
meaning I want to create
multiple containers using multiple
different, different services.

You remember microservices example I took for Uber,
we had multiple services, isn't it
or not like passenger service, driver management, service,
payment services, notification services, individually.

All these are applications. Agree or not.

Now I want to create them via containers.

So don't you think I need to go with multiple commands
and docker commands are this small like ls, mv, rm,
nor it, they're quite lengthy.

If you saw like from time to time.

Now, is there any way instead
of creating this multiple containers, each
and every time I'm going and working with, can I go ahead
and rely on some scripts
how you wrote shell script.

I'm saying is there any way I can write scripts
for multiple containers?

Answer is compos,
but what they use something called Yael format.

Don't worry as of now I'll not
discuss syntax and everything.

Just try to understand how it goes.

Yael, you will learn in the Kubernetes.

Next YAML also is used for Kubernetes.

That's where we'll discuss what is it, how it works
and everything as of now what it's like a template.

We'll use the templates
and we'll go with clear problem statement.

I hope you understood and same thing what I'll do,
like earlier I showed you without volumes,
without network, right?

Without buying amounts, without tags issue.

Same way without com. What is the issue with com?

What problem it solves? Got it. Okay.

Let's see. This kind of thing is, so far this word,
I'm taking multiple applications to demonstrate for you.
Uh, one more thing. I will write some points also how
to work with compost.

This is what you do as a workflow.

You define services.

That's the term we use in com for containers.

Five containers means five services

that make up your application in a file called as, uh, like
how you said script do as such.

Composed yml, y ml or Y ml. Anything is acceptable.

It's your choice. You can go with anything so
that they can run.

Meaning all these containers can together run in an
isolated environment.

Define all your application environments.

In Docker file, which we are already familiar
with this concept and we use this command called compose
up to start your applications.

If the app is already image is built.

If the image is not built now, you'll go ahead
and try, try to rebuild the app.

So in that scenario, either this way it will work

Or

this is how you can do

build means images will be built also dynamically.

No need of images. Images are
already built, everything is ready.

The top command is sufficient.

No, you want to build again dash dashboard.

I think like it'll make sense once I go ahead
and start doing the things in the practical.

These are the two different syntaxes that we can go ahead
and rely on to use it.

Uh, now as I said, now I'll take one application
and I'll try to demonstrate without compost
how the things are working and then
go ahead and start doing the work.

Let it be here. So I have like already

a few services running here.

Now let it be no problem. Now you see
how I'll start doing the work.

My application services,
I have multiple projects.

One is just the one you just used.

This is one service, one application.

Microservices means what?

Collection of multiple things, right?

Each and every individual service,
all the small services will make up your main application.

That was what we discussed the microservices concept in
that this is one service.

Same way I have
multiple projects with me like that one e-commerce project,
work it and use it if you want
like that.

I have multiple projects with you.

One restaurant related application
like this.

I can go ahead 5, 10, 15, whatever you have to, three things
for you to understand.

Now I want to go ahead and start running these things.

So assume that login app image is already built, isn't it?

It's already built. It's already there, right?

Like this. I do

see the command size, so

I don't give give, I didn't give the any tags.

Latest stack. See the size of the command. Quite big
created it.

Now I want to go with this. Uh, what is it?

Uh, food application where there is no image,
nothing, no, I want to build
and I want to create the category

or is there but no docker file.

I'm building the image by myself now,
but this is my second application.

I'm going back to my parent directory.

So now we have two additional
services which are running for us.

Uh, I also want to work with a third one
like the e-commerce application.

Same here also you see any docker file is there? No.

So again, I need to build and use it, right?

So I'm going with a different approach.

8 0 8, 7, right? Six.

We already went with
can I do this?

Yes, right? Volumes concept what you learned, buy modes
in the next image I'm taking, loading the data.

Yeah,
all the six services,
8, 0, 8, 5, 8, 0, 8, 6.

There's a restaurant application. I said
now all these things are making up my application Azure,
but isn't it multiple commands
and easy to remember?

Do you think maybe you might make so right? No.

Three are there. Let's say assume that you've got 30.

Do you think remembering all the commands options,
both numbers, image names, tags,
volume names, locations.

Is it easy if you have 30 complex rate?

So we don't wanna take the chance.

We want to write all these things in a form of script
because if you write code ones correctly,
whenever you reproduce, same outcome will come right
already observe multiple times.

I want to go ahead and use that approach.

Then the solution is compost.

Very easy. I'll show you.

They're saying the same thing.

You see they took some Python application,
they did the the same thing.

Created some folder, went with their code,
they wrote their own docker file,
they read the bills and everything.

Uh, this is important. What was the file we need?

Same to same item.

I'll copy this thing down.

Composed doc,
copy this.

All code insert
services under that.

Each one is one service.

Now you see same indentation is important.

This indentation only need to follow. I'll show
Service one.

Same colon enter, it'll take two. Two spaces.

Yeah, well I said no later on you'll understand how it works
and all of it Here they said build right meaning they're
doing the build off image, but already we have image.

Which image service one. How did you create

On which port it is running?

That's the port.

Port, sorry, ports same.

I'm following the same structure you see below how it is
that I'm copying dash

8 0 8 5 it

because I'll delete all those ones again,
pull and where it should go.

That is my service. Those three lines,

what you wrote, sorry, one
to three, four lines what you wrote.
The responsible for creating the
container called service one with the options.
Is it hard? But you don't need to buy hard.
It'll be there in GitHub it's called right?
Very push end of the day.
See this is what you do
as a DevOps engineer writing the
scripts and updating on GitHub.
A lady wrote shell script after that Euro docker file.
Now I'm writing compost files next like that.
I'll keep going. This is service one.
Enter again, go with indentation.
Second one. What I did, I did the build
build from where?
Which directory under the current directory
code is the directory where application code is present
there only Docker file is also there.
Now pick it up from there. What should be the image
name you want to give for that?
So now what I don't need to do docker build command.
I don't need to do docker. Run command
service
three image is already there.
What is image name
on which port should I run?
I'm getting rid of what I don't.
That's the compost file for my application
earlier what I need to remember,
big commands, all that stuff.
Right
now those services are not there
and maybe like what you say,

that image is also what I build.

I'll remove it off so that we can confirm
none of them are there, which are required
for this app services.

Now again, if I ask you to do everything,
wouldn't it be taking time now again, getting them,
cloning them, doing the bills, running the containers, going
with the volumes and all of it.

No knowledge. So now technically if you see all
of this will be down, right down,
down, down.

Now I want to set it up from the scratch.

All you need to do is one simple command.

Docker, compose up. Friendly.

Now you're doing the build also now.

So I'll say dash, dash, what it does,
it'll read the compose file constructs
that earlier.

10, 15 minutes now hardly two, three seconds.

Now anytime you want it, what I need to do is simply what
next.

Next time when I do clone, automatically files comes up.

Simply say Docker, compose up, build. Things are done.

And here, one more good thing about this
generally, how did you check earlier?

Docker container list,
hyphen it, showing you all these things.

Now it is confusing which one
belongs to what and everything.

It'll show you the things only related to your project.

Not confusing. If other project things are also there in
your system, in this project,
only three things are there, right?

Showing me three. Now even taking them also not so hard.

One command. Everything is swiped up.
You understand the point is select that if I use compost,
I can get these benefits
and see end of the day everything is becoming what,
as I said, the main purpose of going
with this particular tool in the first place, right?
I was showing you how slow things were
and how am I getting it maximized with efficiency
by making it more faster.
Talk about everything. What we did, everything so far we did
with Docker is all making them faster, isn't it?
Tags, compose, building the images,
running the applications.
That's what dock is. Clear.
I hope now you can connect the things
and understand how the dock is actually working.
That's the main criteria of going in using this software.
And now you know how the compose is working. Tax is working.
Now the next part we are going
to work is our main application.
What was the purpose of learning all of this?
Techniques means for this one only
clear, no, I want to just use my docker
and then completely docker as application so
that I don't need to spend two, three hours next time
I'll build all these things, keep it in
registry whenever I want.
Run it initially.
It'll take time because to learn and set up
and all of it next time always you don't do this kind
of repetitive work.
Again, rebuilding and all of these things, right?
You use one software called Jenkins, all these commands.
What you learned, you'll give to that particular tool.

It'll do this work for you.

Moving next how it works.

We'll figure, figure it out now concept-wise, clear everyone how Docker is making things much better moving forward as we progress next class on Monday, I'll go with this LMS app, dock it the next time onwards.

Again, I'm not going to take lot of time, like two, three hours to get this app.

Matter of minutes, I'll be getting the applications right?

Clear everyone from online.

Also, I hope you understood the concept here. No.

Each application we will run as a separate container or uh, in one container we can have multiple applications.

No, I think we already discussed this point earlier also.

Okay, one application per container.

So if I have 50 applications, I need to have 50 containers.

That's correct. Okay. Right.

So technically that's we currently did was the same thing.

What I discussed earlier, these are all contents.

Okay? Same if I had code, right?

This is what I would have done, right?

Like I said MKDR, app services, MKDR, Uber, login service, e-commerce service, food service.

These are all services.

Okay? Got the point is no, that's how we go ahead and start working with the things, right?

So next part I'll go ahead and show you how to work with our full fledged a.

So far it is also still minute only.

But the actual thing is if you're able to take one three tier architecture application, like

what we did earlier,
if you dockerize a product like this,
you can dockerize anything
because all the applications we work with
with three architecture applications, right?
All the popular apps. So next hours
is a three tire application.
We are going to dockerize this product.

Video: Project Dockerize LMS-1

Our LMS application, whatever the LMS app you went
and we did manually step by step.
Now I want to take the exact steps
and try to create it in terms of docker image
and see how the things behave.
Same thing, I'm not going to do anything new.
I'll just take our existing LMS application steps, try
to do it in terms of Docker and see what happens.
Then we'll go ahead and realize the things,
and finally we'll end up with having our own LMS application
as an image.
So one second,
another LMS code.
And in terms of dev code, whatever we have, I'll just use it
in the meanwhile.
Let me start the server first
with the dock post.
These cans we don't need,
so I'm just gonna quickly get rid of them.
No ERs in my system right now
and I want to start working with my LMS app step by step.
Now let's keep our steps, what we did in terms

of LMS first, and then we will proceed with the part.

That's how we started. And you already know
how the app works on the platform.

Post list works with 5, 4, 3 2. Our backend works on 80 80.

Front end is going to work with code number 80. Okay?

So here, I'll keep this document sideways side
and then we'll do the work, right?

So here.

Video: Project Dockerize LMS-2

Official images present, I'm gonna start using this
Postgres image to get the work started
and pretty much same I'm not going to do.

And now you see what I do.

Simply Docker container run,
uh, dash, dash me.

It says something like T one image name is
Postgres, just like how I used to do earlier.

And uh, I don't want it
to be started in the interactive mode.

So I'll say date,
start the container in the detached mode background
and use the image quarters Postgres too.

Run this criteria. Let's say what happens
as usual, how we do it.

I'm trying to do the same thing.

Okay, clear
what happened to the container.

So something went already said something went wrong.

So in simple way I can use what
logs of which container.

See what it is saying and error came.

Databases are initialized with a user password.

It is saying, for example, you need
to add something called like a password on which option
Docker run, whatever the password you say
by running the command call slash password
in the manual way here it is the same option.

So they're telling while creating the Postgres container,
it is mandatory for you
to pass this value in C you got.

But anyway, generally we use most of the time we
to figure out the things first time.

Now I came, what is it?

How to use this is what we did lot, lot
of times when we are starting the things
and what they are doing while
running the Postgres container, they're passing what?

And hyphen. Hyphen is nothing
but environment variable here also.

So we are saying while creating the container itself,
we are passing some environment theory.

Am I clear against what I'm trying to say here?

It's mandatory. If not container will not work.

Both in the CLA we saw as well
as in the graphical interface from the website also we saw.

So now what I'll go with the proper way container run.

Uh, let's say something like uh, LMS tv,
The password.

So while creating the container itself,
we are setting the database password.

What, what we did in our session.

LMS 1, 2, 3, 4, final, if you guys remember,
that's the same password I'm setting.

And let's see if I'm going to create this time.

It works or not. Check the

I see a command.

It is working and now things are good.

So a database is running

and also you see port number 5432 is also there.

So I can say docker container remove forcefully.

The other one, even one is not required

because this is working absolutely fine.

Now this is database. This database needs to be connected

with what, according to the knowledge, we know

for this we need node and NPM.

Again, I'm not going to do all this step and install

because I'm using something called like Docker,

which is very good in this point of view.

Video: Project Dockerize LMS-3

Database and you already know that point also, right?

How it is going to utilize it.

If you guys remember our previous sessions,

we wrote something called as an ENV file

and it contains all the details but further node

and all the things are required for us.

Now we know already in the system there is nothing there

but again what I have no intention

of installing all the things

so I don't need to install node and all that stuff.

Maybe I'll create a container

and I'll test analysis just like how I tried

to create a T one container.

I'm saying node

and which version of node we want 16

because of the requirement.

Now versions are called as what here tax.

Let's see, I'll try, if it fails I'll check the logs
and again, I'll go ahead and try to figure out the things.
So to build a backend, we need two tools.

One is node, another one is NPM.
Any problem working fine with node 16?

I want to verify we can
use execute command on which container one
no as a less both are working absolutely fine, no issues.

That's what we wanted done.

Next we need to get the code. Same as usual.
I'm going tell me if there is any difference later
after getting the port I will go to them back
and folder did the same thing here also.

Yeah, sure. Yes. Like we are using NPM for the build tool.

Yes, correct. Right then I'm creating
ENB file

then followed by
now here everything is same but here can I keep local host
because there is no database running in the local host
where this database is running inside a container
and arrive,
change the password also
because what is the container password and
arrive now connect
with this particular container when going and doing the work.

Got an idea, say next steps,
install Prisma baby push NPM, run,
build node, build index dos.

These are all the things I'll go ahead and try.

So now I have given the input.

Now I need to start building the image now step by step
because now technically this particular container,
do you think it'll have a code and all of it not right?
Obviously just check once

on T one.

Do you have your LMS application code to do the build?

No, of course not. So it'll obviously fail

if I say something like n pm install.

That was the first step you executed in the backend buildup.

Obviously it is failed saying what cannot continue

because you need your application code

and all this stuff right now I'm going

and setting up the things step by step, right? Yeah,

But there we already uh use the command GI clone.

You can tell them,

yeah, go ahead.

Sorry. You're uh telling a question there.

Uh, there already we have uh, used the command uh,

GI clone we have uh used.

Then why didn't we get the code?

Let's see. Git clone.

I did not ensue inside the container. I did it on the host.

Okay, okay, okay, Got it. Yeah.

Now containing with the part here.

So now what I want code

and you know how to copy the code, agree

or not, how do I copy the code?

Instruction called cvo PY.

In the last class for your login application,

you wanted the code inside the engine X.

How did you copy? You used copy.

I will go with the same approach. Why not?

So now what Here I'm gonna start repairing a docker file.

I think I already wrote my docker file from the previous session but I'm just going to get rid of it.

I'll write from the scratch. One second.

Bl docker file.

Now this is for what backup

because database will not have any data
unless application connects.

How does Postgres know that?

I'll set up an LMS application which has a table call
and all of it of course I don't know via backend only
that information will pass.

So that's why we just got a database there.

Now here I'm going to prepare a docker file
for my backend system, uh, to work with backend.

What is must tell me what should I say?

Should I say Ubuntu here
or should I say Postgres here?

Should I say nGenx here?

Think what might be the correct option here
or anything else you want to suggest?

Give it the thought process once.

Okay, Sri Moki said ingen X. Okay, anyone else?

Ingen X? No, not Ingen X Ingenix is required for what?

Frontend. But this is not frontend, right?

Unless backend is not ready, how can we connect
and work with frontend?

Not possible. So here the correct thing to go with this
because in order to work with backend we need node
and as well as NPM in Nix do you get node and NPM?

That's the reason why we went
and set the base image you already know
now from is used for what?

Because every backend system, next time you want to work
with this application,

MS especially no must without node
you can't do anything literally.

Now on top of this node, what we wanted to do, we wanted to
copy the code because do I have the code there?

No it copy code.

So further that what I use the instruction
dot means current directly.

Where should I copy container?

Where in the container where not dot
remember like login application.

Where did you copy? No which part?

Oh, user share x html.

Some you said but here it is backend, right?

Will you have user share XHTM? You need to decide that.

So if I say slash where it will copy under

Am I clear what I'm trying to say?

No. Okay, I'll try to do it and I'll show you.

copy.space/this

was the first container you already created.

I'm saying list slash Do you see

your code?

This was built earlier using only engine X.

Sorry, only node. Now you see what I
read the current dock file.

Now what it is doing it read node J and it copied the code.

This is how you created T one container.

Which image use

both T one, T two working in T one.

Did we have any application code? No.

Right, I'm going with T two.

T two is created using which image? LMS backend image.

Like my name slash backend image which you said copy the
code of current directly to route.

Did you get the code packaged js o Prisma, SRC.

Now you understood what I'm trying to plan how you did
for your frontend application in the last session.

I'm planning for my backend application
as per my convenience.

This point is very important.

See you need to understand what I'm trying to do.

Goal is what? To containerize your LMS app. Meaning what?

Every step you did in the VM manually.

Now I need to orchestrate it in the docker.

Am I doing that or not? Uh, no.

It might be looking clumsy, right?

See everything was in one place where I cannot figure out
what is my code and what is the system file system.

Let's make it more clear.

Okay, let's see.

It was clumsy.

What additionally you're seeing Now

it's more clear and in this

you said copyright copy it.

Now does it have everything to do the work?

I hope you understood the logic.

What I'm trying to say, everyone just preparing the things
according to what I did manually.

Now code is there. Now I can run NPM and all that stuff.

Now give it a

try, copy the code.

Next thing what I need to

do, execute what?

What's the first command that I need to go? So

how do I do

execute build related commands?

We use a thing called run

Making sense.

You see it's saying what failed

because NPM install will work inside the project folder.

I have but where is the code in backend?

What is the actual path?

It is not correct. So I need to set the path

that's a command working

directly meaning moving forward when I do any commands it should happen.

Where?

Here I'll show you the difference.

Rebuilding. See how writing the code, rebuilding writing the code rebuild Just like a develop.

Now you see NP install is working and it failed right now it is working earlier.

Which container? T three which was road.

Now I'll create a new container once again by using the same approach.

T four, T three road T four.

That's why end install.

If NBM installed work means what node models should come.

Now

did you see node models here in the T three check in T four because NPM install was successful.

Are you able to catch up?

I'm pretty much sure you can catch it up if you had really done this lab, whatever we did on 10th February something, right?

Almost like 15 days ago.

If you do, if you did this lab, this should automatically make sense, right?

No models came after this, then only the database tables and all of that will get uh, updated.

Isn't it or not? I'll do the same thing.

Why not exactly what I'm trying to do there.

I'm literally performing the same activities over here also end of the day, not much drastic difference.

I hope you understood. Same what I do.

Yeah, file.

What is the next command

I report?

You can write it in separate command also no issues

what this command does you remember, right?

It'll read the environment file, connect to the database,
execute this particular database script.

Then only I'll be able to do the build.

I'll go and set

again building this time along with install

this Uh, Prisma DB post should do it.

So basically it'll take the ENV file

and it'll see now we'll see

you see primar DB push

just exactly the same thing what we did there.

Images. Now, now

we have the DB related things also what is say synchronized.

If not what I can do one more for that stuff

for clarification?

No, if I go ahead

and create a container like let's say T four earlier I

created and I'll create one more container called

TFI for verification.

T four tfi, right?

Did we get any build folder?

Because we did not do the build.

We only updated the database.

Stable maybe, but I don't think it'll capture it.

Log Steve. Yeah, it'll not capture

because it is running in the backlog environment.

Once the build is done,

maybe I can fix that particular thing.

Now we went, we updated the DB tables

into the database part.

Now I want to go ahead
and do the build clear is what I'm trying
to say because I already showed you
in tfi did I have any build folder,
which means build is not done yet.

What was the command?

The only way I can test creating one more catheter.

Earlier we saw there was no build folder
then to run it.

What I did, that's when you confirmed it.

Backend dysfunctional. Agree or not?

Am I doing anything new here?

Old, same story but doing it via doc
again rebuild.

You already saying there was the same message you
got now but see

it is still saying on the top what building

You remember I said this node process, if we hold the things
and you can't go ahead and out it see now you're not
building technically is this command comes under build.

Think once. Go back. I said this is for what?

Building this command is for what Or start.

It's not building. Run is used for building the things
run you cannot use for starting.

So it things that you're still building.

Image is never built. It's still building only
good.

What I'm trying to say this knowledge you'll get
with the context of application running is different
or I would say building is different.

Running is different. Now I'm not doing the build here,
I want to run in this regard.

This is not the correct way. So it'll just track like this.

It is no use image.

If I want to come out, what I'll say control signal.

If I do control C images cards.

So what is the proper way?

Tell here what you did the bit here. What I need to do

Run the A PA, right?

In that particular regard we don't use run.

Actually it's a little bit contradicting

by using a command called CMD.

The command CMD is used to say

when you're able to run an image,

which command you want

to execute on the container is the question.

When I say docker container run, what should happen

is this one and here it'll take the instructions in the

format of list north.

And this quotes we need

to give like this followed by spaces.

It's a rule base. Now I'm saying what

build and when.

I want to go ahead and run the container.

Use this particular commander.

You can say it's a startup command for container.

When I fire a container, what should happen?

That's the thing. Let's just,

if this time it is going to fail or not.

Again, I'm doing the bit.

Let's see if it'll still struck like the earlier part.

Now did I got any failure or did it got struck? So now what?

I have an image. Let's see now when I try

to run the container, what happens?

T seven, right?

So far these mini test containers I created from the

last half hour onwards.

I was just testing and trying testing and trying testing

and trying till I get the thing.

This is what you do in the real world.

Preparing the docker file

for your project is your responsibility

and you can do so if you only know

how your application works

and you understood why I said

that development process is important for this things.

Now, uh,

how can I verify the things are working or not means?

Docker container in the T six container.

We have some address now or docker container.

Inspect T six, correct ip.

That's IP address girl.

This IP address 80 slash api.

That's how you check the backend.

Remember this command,

local host because same mission,

but here it is running in the container.

No. Any response?

Okay, which container?

I'm checking because here you did everything correct?

Yes, here also I got an IP

towards the ip.

Is that the same message you got when you

tried the thing over here?

Now it's not coming from local host

where it is coming from a container.

So now your backend is entirely working in the container.

Agree or not? Isn't it how you can verify?

Simple On my system there is no

node on my system.

There is no NPM on my system.

There is nothing running with port number

88 then isn't it?

Everything coming from container? Yes, yes.

Now officially my backend is ready
and everything is coming from container
but small confusion till now.

All good.

When I was running a backend container,
it is showing me 5, 4, 3.

When I was running engine X container also it was showing
80, but why it is not showing.

Did you get the point? What I'm trying to say?

Did I say here, run it on 5, 4, 3, 2 or something like that?

No, right, but still it came up.

If I give one more context for you,

P eight in Gen X

80 came then for my application Y it is not coming.

Uh, because we did not write one instruction expose
C one by one where the things come in that spot.

So before running it only we write it.

Which port? Because you said your backend is gonna work on
port number 88.

T eight we created. Now I'll go with T nine.

Now it's more convenient for person who goes
and runs it for the first time.

No, it's working exactly like a database or a web server.

Our backend is also showing yes, your app is running on 88,
but generally the backend, how did we test?

Finally, if everything is done
via public IP like this, if I go
take my public ip, if I give it try
with 80 80 slash VPI.

No response because it's not there in your system. Why not?

Then I'll give it a try link and this containers.

Now I tested all the things now T one, T two, T three,

T 40, 60, 79,

all the test containers I'm removing
because now end of the things are working
as now you see the idea.

Can anyone tell me how should I give
the port forwarding now?

Hyen P what and what?

Okay, someone said eight zero.

Eight zero, okay, next colon
again 80, 88, 80, only 80.

Think once is this correct?

R this correct.

R this correct.

Past second, third, what is the correct combination?

Second and third are correct.

Technically first is incorrect.

The right side port is on the container.

Is your application running on port 80?

So obviously this is
incorrect, this is not correct, correct?

This two are correct. If I do this way from
outside also I can check on 80 80.

But if I do this way from outside, I need to check on 80.

If I do this for front end, problem comes
because you cannot take one port for multiple apps.

We already aware of it. So in technical terms both are
correct, but in our scenario this is correct.

Outside port 80, 80 is sent to 80 80 inside
everything is coming from contain.

Video: Project Dockerize LMS-4

You already know it.

It's not new for you. What we did
go back to, okay, fine,
I'll do the same thing then.
What you did ENV file,
you did the same thing there.
Also in your previous session
and what you did updated with you
save it,
then you went with this command
where I need to write the commands.
No, exactly.
Already have a docker file for front end,
but I'll remove it again for your reference.
What, what are all the things I need to think once
to do the build all old
document again.
I need NPM install. NPM run Build meaning again node
and all of it is required.
Right? Why to write from the scratch
Copy in my LMS app
in the API folder already wrote docker file
copy here documents current direct.
I have the docker file, which is of what As is backend,
but yes, small changes.
Do I need to do all of it?
Because backend does not expect all.
Sorry, frontend does not expect all of it.
Frontend is expecting only two things. Install, run, build.
After this what you'll get,
yes, I want to see the same thing.
Then once you get the dis folder, then you install engine X,
then you copy the dis folder into Linx.
Then you got the app. Agree or no?
First let me see if build works or not.

Then I'll go ahead next
Back and build rent and build.
I did the fend build right?
I just wanted to do again a small experiment
to see if all these things are working as expected or not.
So
no issues.
Steven is still running. I
front end code
resolved.
Did you get what you wanted?
Several list and then this was the content you had now, uh,
like engine, so did not take the capture or something,
but you remember some image file, some HTML files, assets.
I got the same thing, but now is it running?
No, you only did the build. Agree or not.
You only did the build. Yes.
Then why it'll run, of course it'll not run
after the build is done.
What you did, uh, you set up web server,
then you hosted these files in the web server.
Agree or not? That was the construct.
Sorry, that was the instructions you went with.
Uh, do I need to install Nix and all of it now?
Because all nix is there idea.
See the idea now
already you have node in node.
Do you have ingenix? No. Node is different. Nix is different
from Ingenix.
Now, interesting copy source
and destination destination.
You already know what's the destination.
This you have the clarity. What is what about source?
Where should I copy source?

What is the meaning of source here? Disc content.

That's it. No, this is your source.

I need to keep it inside the internet. Now where is this?

This

thing once

is it there in my home directory?

If you want, I can show you. Do I have any dis
got the clarity on what I'm trying to say
at you copied code.

Now I'm trying, trying to copy, but this is not corrected
because I cannot copy the dis from here.

Where is dis under front end image?

Agree or not? Unless

I said from T one container.

This T one container created from where?

From the front end image. Then meaning what?

From one more image, I need to take the output
and copy it into another
image earlier.

You copy it from host system. Now you're saying what?

Copy from an image where already things are now in
that scenario slide change in the syntax.

Copy from

the thing is from where?

From the frontend image, right?

So you need a reference act to add a reference
to say as whatever the name you want.

This is what

copy from

in here where the uh, output folder is there.

Now you've got the clarity. Guess
any confusion with this comment?

Clear everyone. What I'm trying to do,
because I cannot use normal copy

because it's not there in the system,
it's there in the image that we try to build
clear good everyone what I'm trying to say here.
Give it a try unless we don't build, we don't know.

Again, I'm rebuilding the front end image.

That clear

two, just testing
now it's coming as four eight
and how I can verify
execute on D two.

List me user

share ingen X.

Perfect. We got all that assets, everything.

Uh, now one final way I can confirm is
now we are clear, right?

These things are working so I can get rid of T one, T two
and I'll simply go ahead and do the things in a proper way.

Yeah,

and take my thing and I'll save from port 80.

Same app literally.

Perfect. That's what we were anticipating
and now you see for your confirmation
on my system I'm checking.

None of those ports are there except the routing
ports your routing.

They're not running in the system technically
is what I'm trying to say.

So everything is coming from containers. Now what?

Pushing the things so that anywhere, anytime I want
to run the app can just do it.

I agree. First time it took a lot of effort
to write the docker file, preparing and all that thing
because we never did it first time.

Obviously it'll be difficult for you to figure out anything

then pushed it.

Brenda, also, I'm pushing, uh, now simple.

Let it get pushed. Done.

Now in my local system, I'll show you
nothing, right?

No LMS app, nothing.

This is my local host, meaning my laptop in my system.

Dock is there already.

Doctor is starting it.

I'm opening a new shell

and literally

I'm not doing anything

Now, same procedure.

Let's get back to what I was saying

when I started with the docker.

What was our idea

in this system?

I don't have anything literal.

No node, no NPM, no n engine X, no post risk, nothing.

Literally, I copied one command.

It says the name is already, sorry,

I'm running in the same system, right?

But my bad, I need to supposed to run in this one.

Unable to find no issues. We already have it.

Now did it take two, three parts, just executed the command.

See, it's learning in my system and yes, still I'm able.

Tasks

are clear is what I'm trying to say.

If not, give it a try. No issues.

I hope you understood the point, is what I'm trying to say.

Is it what we actually wanted?

Now I no longer need hours of time to go ahead

and do the things once I prepare and keep it with Docker.

It's just like matter of minutes to run the things.

Is it what we wanted?

Isn't it Now building is faster, the releasing is faster.

Deploying is also faster.