DevOps - Module 7: Build And Release Management For DevOps

DevOps

Module 7: Build And Release Management For DevOps

Video: Web App Types

Web applications in that static applications
and dynamic applications, right?
I said specifically we are working with web applications
and in simple terms, what we can access them
with web browsers over the internet, isn't it?
Now in that we got something called like static dynamic.
So when I'm talking about static
applications, you see what they're saying?
Applications with fixed content, meaning
the data is not gonna change
or the content is not gonna change based upon the user.
See blogs, portfolio sites, these are all what?
Static applications.
Whereas dynamic web applications,
basically their content changes based upon the user.
Okay, so you see every e-commerce site is a dynamic site.
My orders will be different from your orders. Correct.
My addresses will be different from your addresses.
My uh, last six months, uh,
order transactions will be different from yours.
And same thing like social media platforms.
My photos are my uh, what you say, uh,
images might be different from yours, agree or not.
Uh, now we are working with dynamic web applications.

Static web application is nothing
but what you did in the last session,
login app is static app, is that having any functionality?
As of now, we just like went and you hosted one app,
but there was no one logging,
no one performing any activity.
Now we want to make it dynamic.
Take your LMS app.
Based upon the logging, the courses are being changed.
I want something similar. So if I want to go ahead
and work with dynamic applications
specifically like web applications behind the scenes,
what are all the technologies we,
how they interconnect with each other.
We want to understand what I'm trying to say.
So we'll go ahead and start working with
that particular approach because now once the code is done,
the LMS app you are having now in the last session
what you did, the analysis,
it's actually a dynamic web application,
so it has multiple components.
What are those components in the future when you work
with other companies or other projects?
Something similar might be there,
but conceptually I would say
syntactically, they might be different.
Are you clear what I'm trying to go up with?
So maybe once we start discussing about these things,
it might give you a sense, right?
Okay. So here if I want to go ahead
and start doing the next part, which is building
and trying to understand the applications, you need
to understand the architecture of applications
and especially we are talking about web applications,

which are dynamic, right?
So better I'll add this points here.
Also for your understanding,
We are talking about web applications
and in that specific label, dynamic applications.

Video: Application Stacks

Stacks application stack,
you can replace like this, it can be LMS
or it can be some e-commerce system, okay?
Or it can be some uh, HRMS system.
Clear what I'm trying to say. It can be any system,
but in order to build that system, we use some stacks.
You see what it is saying?
A stack nothing but what set of, like you said it stack
of books, stack of plates, same thing
in terms of technology.
An application stack is nothing
but what set of software programs that will include software
for your front end, the application layer,
and as well as database layer.
So what technologies we use, uh, if in case you're aware
of it, well and good, but if you don't
know, let's see what there.
I just collected one website
maybe can be helpful for you too.
Most popular stats that we use for billing applications,
I just, you say it's from official Wikipedia.
Simply speaking, the technologies used in the most popular
websites and if you see right what it is saying,
agree or not, they're all dynamic websites
and their development typically involves what?

Server side coding, client side coding database, nothing
but what client side means.
Fronting server side means application database
and the programming languages that you apply
to deliver such websites will vary
from one site to another site.
Just have a look at these things is isn't it like most
popular sites, Google, Facebook, YouTube, Yahoo, Amazon,
Wikipedia, Bing, eBay, LinkedIn, red, WordPress, Netflix.
These are one of the most widely used sites along
with their subscribers or users.
You can see it quite high
and all of them have what common.
It's not like everyone is doing the same,
but something you might see most often used
for front end is JavaScript.
JavaScript technologies I would say.
Okay, clear everyone.
So here Google went with some stack,
which might be different when you compare it
with something like uh, what you say, Facebook maybe,
but Yahoo it is a little bit different.
Who will select it? DevOps engineer has
nothing to do with this.
Software architects are the ones who design this.
People who know system design, engineering,
software architecture part,
those people will go and work on it.
They'll give the high level architecture.
Then for this developers will be hired based upon the
technologies we are selecting
and those developers will give us the code.
So obviously now for LMS application also there should be a
stack agree or not according to this thing.

What we are discussing

Obviously because we also have one frontend backend

and database system to work with.

So this is what we used actually

the database we used to build a system is post this.

The backend system we went with is a no js,

JavaScript based framework

and then we got frontend JavaScript,

which is basically React J.

Okay. Does it mean

that every company should use the same in No,

it can be completely re replaceable instead of Postgres.

I can go with my scale instead of node JI can go with Java

instead of uh, what you say react.

I can go with Angular

or it can be as simple as simple.

HTML and CSS should also work.

It depends upon your criteria

of the application you're going with.

So this is like very popular.

It's also basically like we have something called like Mon

Stack in that this is what we use along

with Momo DB actually,

but this is what we have selected for our application.

Like when we are going and working with this app.

Uh, now whatever the code I have given you right in the last

session, it has all

that code, okay?

That front end backend and database code is available.

I'll show you also Nouse,

We have the code link.

Now

This web app you're seeing now,

so name itself is saying what should be this code?

Frontend coach. That's the frontend coach.
This a p, you're seeing that? That's my backend coach.
Okay. Generally when we say backend,
the term we use now it is combination
of both application plus database.
When you say a backend system means this combination,
you'll have your application layer
plus database layer together.
That's called as a backend system.
I'll go to my backend system here. You see this?
This is all the code written by my development here.
Uh, I'm just gonna show you,
there is something called like Prisma.
It's basically a database model tool.
I said that these are all technologies we work,
I mean like developers work with technical,
the database client actually it'll help you model the tables
that are required to store the data.
Developers learn all of these things. Generally
you see what file I'm having, you know
what SQL is used for.
It's generally a what you say.
Uh, I can say it's like basically instructions
that you want the database to follow.
Basically queries spirit, like how we are giving come
in terms of Linux to talk to your database.
We give queries that queries are domain space.
So basically we use a language structured query language.
You see what it's saying? It's a domain specific.
Uh, if you take Python, right,
it's general purpose programming language,
meaning you can use it for multiple things.
SQ is specific to only databases.
You cannot use SKL to go ahead

and design some web applications.
Not possible. But if I use Python,
I can design web applications.
That's called general programming
or general purpose programming language.
So SQ is used to manage what data,
especially in something called like
relational database management systems.
R-D-B-M-S. Okay, so here
if you, if I open this particular file,
we are creating tables basically data.
When you organize it, it'll be organized in the form of rows
and columns, like how you have Excel sheets now
where you have rows
and columns in the same way we are going to
organize the data, uh, to organize it.
We use some instructions. This is a Q.
We are creating a table
saying the table name should be course
in which you are having what Id title, some description,
uh, then a module, then topic.
If you observe this structure,
I hope you can see the relevance.
The same thing. Where did you see
when you log into the LMS?
I think this is what the structure isana course is.
When you click on a course, it'll show you modules.
Under the modules. Each day is basically like a topic.
This is a structure which was making the thing work.
Are you clear what I'm trying to say?
Or maybe one second.
I'm just showing you the backend system. Okay,
I have got admin access so I can do all this sort of things.
Same LMS only, but this is admin portal.

You see I'm seeing what courses.

When I go to some specific course,

I got modules.

I can add multiple modules under

modules topics.

This is what we do actually, so

that you'll see in your sessions when you're watching

someone behind the scenes.

Are you clear is what I'm saying?

Uh, this structure you see is this code.

Now I hope you are connecting the things.

So like this, every application will be modeled properly.

I said no. At the time of architecture,

all these s will happen.

What table, how the structure should be there.

They'll define the schemas and everything accordingly.

Developers will write the code.

None of that is mattered for DevOps engineers.

But you need to understand how this stacks work.

When things go wrong, you should able to understand

that okay, this might be the place where something was break

or something was not working functionally as expected.

So we are not getting the proper responses clear.

Uh, now this entire piece of code is for database

now to do this manipulations and everything.

Now I said no, there is lot of source code over here.

You see courses,

related courses you see there wrote all the routes, how

to get the courses, get all courses like this,

not the entire application.

This is only like one module.

They have kept it open so that you'll understand the thing,

not the entire app clear.

I hope you understood the point. Yes.

So this is basically the backend code. Uh, same way I said.

Now we also have the front end code.

You see guys here I have all the frontend reports.

See some index pages, something related to uh,

what do you say, tailwind, CSS configurations.

This is all frontend code technically.

So you have frontend backend database,

all the codes implemented

and everything is coming from there.

Obviously from it GitHub

repository, clear everyone.

So like this word, it can be any particular application.

That's what I'm trying to say specifically.

It's not always this way. The application can be designed.

It depends upon the architecture, what kind

of components they're going and working with.

They'll add clear everyone,

at least from this application point of view.

I hope you were able to correct like this.

You go for other technologies, it's almost same, right?

Okay.

Video: Build Management

Now how all this work together is a thing, right?

Mm-hmm. That's where build come into the picture.

See, they have wrote the code, right?

Writing the code is nothing.

It's not gonna work actually

unless you actually use the system.

So to use the system, we need to build, okay,

clear what I'm trying to say.

Now we understood the technology stacks.

Now we are going with the software
build and you see what they're saying.
The term build refers to a process
in which the source code is converted into binary code,
which basically in terms of software, we call it
as an artifact, an artifact is a usable product.
Good? Okay. Now here one thing is pretty common.
Any sort of build system, you go with it.
The process always remains.
See the itself is saying what it is.
Process always will be almost same,
like you do the same work.
Outcomes might be different.
Any sort of build implementation, like any application you J
with Java, Python, t net, what is this?
No Js, any kind of system you go ahead and work with.
There will be some things which are always common.
These are the common things.
Source code, any application source code is required.
Agree or not
and just read it once.
Is it what we did in the previous session?
I use visual studio code, wrote the program done
as a developer, updated to Git.
That's what uh, source code files are along
with source code files.
You'll also have something called metadata.
Generally metadata is data about data.
That's what metadata means.
If I talk about this, uh, television, right?
You see voice brand, it has size like maybe 43 inches
or something, okay?
That's metadata data booth.
Something here in terms of project metadata is nothing

but project information.

That is what metadata is.

Configuration details like what project name version,

I said no dependencies, right?

Who is the developer maintainer?

All that information will be there in metadata

for different projects, different things.

I'll show you maybe a couple

of examples in our project.

I'm talking about web application.

This file is called

metadata package.

Do json.

This is information that might be there in terms of metadata

is what I'm saying.

This is for JavaScript technology.

Is

It almost same information,

name version dependencies,

but this is for Java applications

format will be a little bit different

version name artifact

dependencies, but what's the

difference format?

That was in JSON format. This is in XML format.

Same thing you will have for uh, Python applications. Also

some dependencies.

Their versions like the Python applications.

So I hope you understood the point is, regardless of

what technologies we use, it'll always be there.

Source code is must.

Metadata value will also be there along with your projects,

which contains information about your project.

Simple.

Next, so far,

the source code and metadata that will be there in GitHub.

This will not be there in

GitHub, okay?

GitHub is going to contain only the source code metadata

files, but not the binary code.

Binary code. You see when it is going to come

after the build are done.

Artifacts technically,

and it says build artifacts are the files produced by build?

Uh, same thing. This is also handled by,

so technically speaking,

who will write the source code means developer

who will write the metadata, code codes.

Developer who will do the build winds Developer.

Yes. Generally when I say automatically generated, it comes

with what you say, empty content.

Then what you need for your project, you'll update, right?

So if you see, maybe I'll show you one example.

I assume that I am learning reactive.

I'm trying to create a react application.

Like I said, I'm a developer.

I want to write the react

because our stack is reactive, front end part.

This developer,

see they created a project

By default package, JSON key, A template

is how it is there by default

then what developers will change

according to their requirement.

So what they do, my package JSO looks

quite different, isn't it?

So like this developers will update you as a DevOps senior.

You need not need to worry about that.

Writing the code in terms of uh, source
and as well as metadata.
It is completely their work.
But you need to know why metadata is important
because the versions
and everything, you'll pick it up from here.
Technically in upcoming sessions you'll see that, okay?
Which now as I was discussing in terms of uh,
buy and record build artifacts will be produced
after the build is done.
So how to do the bins,
they'll give the instructions, we'll see what those are.
I and who developers will only keep it.
So this is also done by developers in their laptops.
Um, then you might ask, what is the need
for learning all of these things.
It's been simple and straightforward.
End of the day application will be there in the laptop
and is it going to be used by customer?
You're accessing LMS application.
Do you think that is there in my server, uh, sorry,
laptop where it is hosted.
Does developers know how to work with servers?
That's where you commit the picture.
This part
I understood the point is how things are going to connect.
Developers will do the build.
Agree, but the artifact is specific. What's his scores?
What about developer five,
developer 10 and other developers?
All of that. We need to do the build as well, right?
That is going to be done on the server side.
Understood what I'm saying?
That is what you need to be responsible for.

And same thing for 1.1. You repeat.
Ravi, can you come again ma?
Uh, so what I'm trying to say here is whatever the work
is being done so far it was done in the developer laptop,
but I'm saying would the application be used
by you People like students from a laptop,
you're connecting it from a server and you're using that.
So where is that build present on a server
and who is responsible for that means a DevOps.
So taking care
of the things on the server side, we are going
to look into that patch.
I hope it's clear.
Yes, yes. Okay, perfect.
Now simple thing guys.
So obviously now you need to be aware of build agree or not.
That is what we are going to.
And same thing obviously out of the box,
you'll never know what's the build and all this
unless you already have some experience with development.
If you don't have, I'm assuming you don't have
as a developer, I'm gonna give you the kit
that happens in the real time.
Assume that I'm working with some new application.
I don't know how to do the bill.
Obviously that's when development, uh, basically as a part
of scrum team, all these things will happen when you go
ahead and start working with,
but generally you'll get that
information from the development team.
But technically speaking, for faster immunity learn,
all this information is going to be available
inside your repository zone.
In the README files you'll find it.

You have GitHub repository. Read MENA in that.
You'll find it technically right? I'll show you
Like they tell you, but you might not know
what those commands and all this
because you never had an experience with them.
At least one app,
definitely every DevOps person should know full fledged
not coding from taking the source code
to building it, to managing the releases
and deploying this procedure.
You need to understand not with simple applications,
a three data architectures must at least now in the current
scenarios, what,
what's the three data architecture application?
What are the components, how they work, how they connect,
and at least one system,
full end end, you need to have an A.
Then next time when you work with other applications,
now these commands will change.
That's it
are clear and where they will be found wins
in your repository only.
You don't need to ask actually,
but first time I'm pretty much sure if you're learning
someone needs to teach
and is clear everyone.
So that's exactly what we will be going
and working next, right?
So you'll get Katie, that's what I'm giving you right now.
You guys are all DevOps engineers.
You don't know how a project works. I'm explaining you it.
But once you start getting into the real time will not get
this much precise information.
They'll just let you know that my code is here in this

repository, in this file bill instructions are there.
You can go ahead and build. That's it.
Simple, clear everyone. Okay?
Now I'll give you some developer guides
also how things work.
I said it earlier,
just one generic site which I showed you, steps
to build a react project from the scratch.
The earlier link I showed, you know, the same thing.
Tell you all the things how to write the code.
You are not bothered about it though.
Where you are going to look into, it's the final part.
The once entire thing is done,
then your work starts technically there.
What building and deploying it.
So they use one command called NPM Run Build.
You might not know initially
simple if I don't know something.
Did you get the point is, and you can see here also,
that's what we are looking for, isn't it?
And simple. It's like not something everyone knows out
of the box, but you'll do the research
and you'll eventually figure out
how things work about it.
Same way for Java applications, it might be different.
Like you see, uh, generally React is actually
created by Facebook.
So these are official repository.
And you see same thing there. It's repository.
So there are 27,000 folks, meaning what?
People will get the basic code from here,
then they'll start building on top of it.
And how simple, I'll scroll down. What is it?
How to create a project, how to start,

where the code will be there, how to do the build.
You see how they mentioned this is
what developers will also use to learn.
Clear everyone. Now, I hope you understood the point is
how the things are being implemented, right?

Video: LMS Server Setup

This was the element technology.
I said we are going to use a database for Postgres
and the backend system will be
and then followed by the front end system will be built
on of React.
Okay? Um, now
This is how the dots will be connected with this.
From the front end we'll hit the React, then react will talk
to node and then Node will talk to database.
And I said all these connections will have something called
like a database, URA code number, use the password
and of now whatever we discuss in the previous
session, we'll see it in match.
So we'll go ahead and perform LMS bill.
We have the code already.
So how this is going to be built,
we'll get an denied at least one
application you need to know.
Then later on other applications you can figure out based
upon the project you're working
with are maybe I would say the company I work, either
of these particular things will give you that plan.
Okay? So now in terms of LMS application,
let's see what's the requirement to go ahead
and build all of it

here.
The first thing is
we're saying database called Postgres works on 5, 4, 3 2.
The backend system will go on 80 80.
The frontend system is going to run on eight.
And I need a server call T two medium instance,
meaning this one also does not come out of free trial.
The same hardware we used, it's required.
So when you're going with Azure,
also same in the last session, how I showed you
to launch an instance with a specific security group.
Now you need to add this if you want to do it same way
how we did in the last session.
You can part, so now what?
I'll create one server for this element, right? Okay.
Now let's see guys, how am I going
to start working with the things?
I'll say launch an instance.
As I said, T two micro is not sufficient.
So I'm gonna go with T two medium as required.
Then I'm gonna add the SSSH in terms
of port number 22.
Another one is for
database called Postgres,
then followed by 8 0 8 0
for the backend connection
and then for frontend eight
And Then followed
by uh, storage.
This said A GB as usual.
I'm just going with the same thing done
and I'm gonna say launch and Azure also same thing.
You just, and once you log in,
there is no difference as usual.

How we go? We'll just go ahead with the part.
I think this is my network issue. Mostly
that was just network issue you just tried out.
Now once
that server comes in, um, don't worry.
These issues I'm seeing
because of network problem, that's it.
But you'll see it as usual how you go out and work.
Okay? That's the system
ip
and if you see the things that you'll not find any things
that you want in the system as of now, solo
except 22, there is nothing related
to 5, 4, 3, 2 for nor 80.
80, not 80.
Now I'm gonna start installing all
the required things over here.
I said it's already there in the document.
Also in the step-by-step manner.
I'm just gonna update it for you
all the billing instructions, like I said, not generally.
These instructions will be coming in the
what you say GitHub repository
itself in the Read Me section.
But I said you are first time doing it.
So I'm assuming that you don't know any of it.
So I'm going in a step by step.
But generally all these instructions will be there
inside your what you It build notes are
basically the read me notes.
With there. I'll update all those particular commands step
by step.

Video: Database Setup

So we are done.

So first thing is we are going to set up the data

and you see we have an official site over here

from the Postgres goes, go to their part,

simple and straightforward.

Scroll down, you have a shell script.

You wrote shell script in the last part, right?

They wrote a shell script to set up Postgres.

I just copy that script, paste it, right,

click, paste, execute.

That's it. It'll take care of the database, correct.

So Postgre is also a open source? Yes. Open source. Yeah.

All in our course, everything is going

to be open code, whatever we are going to learn.

Okay,

completed.

That's now we can see it. Code number 5, 4, 3, 2 used here.

And it is being done with posts now.

Now, as I said, now once the database is done in the last

part I said anything that wants to communicate

with the database, we need

to have credentials right in order to work with it.

So to set up the credentials of database follow this way

I said no database connections are property details will be

there in the last session, which includes the

following details, isn't it?

Address, code, user name, passwords.

To set the password, we are switching

to a user called us postcards.

You already know that's SU command.

And there I'm gonna run one per command called SPS Scale,

which means basically it'll give you psq, shell meaning database related commands.

You can go and run over there.

And then you can use this command called slash

password to set the password.

Which password? Database. Password. Let's see.

Not in the open user. I'm with the Postgres

User.

It was my regular Shena B.

Now I'm inside Postgres.

To set the password,

the tasking set up the password

for this user called us Postgres.

I'll go with some password over here.

Confirm the same password.

Okay, that's my database.

I went set up and password is there.

Now you have all the details.

Database address means nothing but the same machine.

Then you have a username called Postgres.

I have set the password.

It has a port number called 5 4 3, all that,

that was database details.

Video: Backend Setup LMS

Is not the front end directly.

Our application layer is going to talk

and our application layer is built using node JI was

saying in the last part.

So now in the same server, I need

to install something called like Node J.

So you need to open a new session with the wound use

because you're already connected

with which user post this user.

So I cannot use it, let it be like that.

What the clarity is what I'm doing.

I opened one more session, you see there.

And from here I'll go ahead and connect

and I'll perform the next operations.

Right? Okay, let's try to do that now.

Uh, you can try again of anything. End of the day.

Just need to work out right?

Either way you want, you can try it out.

Now you see, let me log into the system once again here.

I'm also opening one more session.

So I have two different sessions.

Like I said in one session,

I will go with the post this user.

Sir.

Sir?

Yeah. Without,

without Pim file is open in is possible, sir.

Not possible unless you go

with password based authentication.

Sir, I, I actually, I using uh,

session VM works.

Same thing. That's why I say it.

Uh, if you're going with VM Workstation,

then you can directly use the username and password.

Yeah. If you're doing inside a Ws, this is

how you're supposed to go with a WS or Azure.

Oh, then this. Now I have my database.

In one session I have my system connectivity in one session.

Now, as per the instructions, I said no, we need node

to run nodes because the code is going to be off node.

So when I run this particular incident,

then we will work Node is

Neither NPM Node package manager.

This is the command that we use

for building the applications.

There is a command called NPM Run Build.

That is what developers use to build the applications.

But currently, if I check

neither note is there nor NPM is there.

Both are not available first, right? Okay.

Now what I'm going to do is I'm gonna start installing it

then followed by

now.

I'll check both the versions.

So technically your system is capable of running backend.

Simple how you install the web server

to run your front end to run the backend.

I have installed this comp.

If Java application is required, you'll install Java.

If Python application is required, you'll install Python.

If Golang is required, you'll install Golang.

Am I clear what I'm trying to say?

So you'll configure your server according to the parts.

Video: LMS Backend Build

See what I'm doing.

I'm cloning. Don't clone mine.

Clone yours because in the previous session you

already forked the project.

Now whatever the UR you got for your account, update

that clear everyone.

I'm gonna copy that.

I'm gonna say clone,

I got the same LMS project that you got in the last session.

Our backend source code is this particular part.

Inside LMS, there is a folder called SAPI.

I'm going into that particular folder.

Good. You got all this set of things over here.

Now I'm creating on file called

ELB Environment file.

In this file only, we are going

to fill the database details.

That's how your backend system knows.

How am I going to connect? Yesterday I showed you it.

They were writing some files in which address port number,

username, password is there, same thing.

First connection data files.

I'll create this file right now

in that I'm gonna copy paste this entire details.

Now I'm saying this mode is currently under the

development dev environment.

RQA pro. Whatever you want, you can update.

The backend should run on port number 80.

This is where they're defining where the backend should run.

Database details, res username,

your pass before at the rate,

whatever is there, delete it

and whatever the password you updated

after slash password updated.

That was a password I have

set at the rate.

Local host meaning what?

Where is the database In the same system now

self-identifying and on which port we are

running 5, 4, 3, 2.

And the database name itself is post press.

What we did so far,

I'm gonna save these

details once I

Now I'll tell the system

to use these database details to do the next work.

And how we are going to basically get that done is,

let's see a step by step

Under the new session O to user, we are going

to run one particular command called As NPM Install.

So this command will install dependencies required

for your project in the package JS I showed you list

of dependencies, right?

Those are all required to continue with the work.

So we need to say NP install file, why we are running it.

I said no. Developers will write the dependencies

or modules we call them to get those modules.

We need to run this particular command are simple.

These are build instructions. Pretty straightforward.

How would I say you need to add stage command if I want

to build, this is a process as a DevOps engineer.

You'll initially not know unless you don't go ahead

and read it from the repository.

Read me file or someone explaining you the

details from development.

Good, clear every, if you're a developer, you might not.

So now I'm gonna say, sorry,

I'll also show you see list

how many folders you're seeing right now.

Two folders on grid. One is SRC, other one is

now I'm gonna run the same command, what I said NPM install.

See what happens,

you said is downloading something by connecting to internet.

Install it. Now let's see what happened.

You got what? Node modules.

So those are your development dependencies

to make this project work

clear everyone what I'm trying to say.
Once you get the dependencies, then
I'm gonna show you one particular command.
I'm opening one SKL file.
I think I showed you in the GitHub
also in the previous session.
But anyway, once again
I said no, there will be a table called user course module
and all these things, but right now it's just the code.
It was not yet created where this should be created
tables and all where they'll be presented
database, correct?
I said, I said right Maria,
last session you'll have a database in.
The database will be having tables and all this part.
Now where is my database here?
No, I'm going back I think one second. I got disconnected
in the database.
I got certain commands.
See old session user.
I got a command called s dt. Describe tables,
not tables, nothing.
Now I'll go back to this user as mentioned,
we got this command called prisma generate and DB push.
What this commands basically does, it'll read the ENV file,
take the database details, execute it a scale file
so that the required structure
for your application is right.
Am I clear everyone?
So it is technically executing the database base to prepare
that structure for your application.
I'll show you also, you see what it's saying?
Values loaded from dot P
and B file C,

data source, database name, local host, 5, 4, 3, 2.
All the details were picked
and it says your database is in sync.
I'll go back, I'll run the same command.
Did I get those tables?
Codes, module, topic, user,
whatever was there in terms of code.
Another structure is right
now you see how they're communicating.
I hope you understand. So when we do any buttons
and all of these clicks, now behind the scenes,
that code is written in that form.
We are executing the command,
but from the front end they're all buttons and clicks clear.
Okay, now I got the structure,
but obviously we will not do it.
The final end user is going to do.
Then it'll reflect into these things
in this particular tables.
Okay, now good everyone.
So how DB and as well as my ap, I are both connected,
but we did not do the build yet.
We just only connected. Now I need to build a system.
What is the meaning of build a
usable product should be there.
We did not have it yet. So now we are going to the build.
Let's see
again, I'm with the user
three folders.
So once the build is done,
you'll actually get a folder called AS build.
Where do you find it now?
When it will come, when you run a command call
NPM Run build.

That's the command to actually do the build.

Guys, I'm gonna take that.

I'm gonna say NPM run

done.

Now let's see,

did you get the folder?

So that is the actual thing

that you need to run your software.

Which software? Only the backend part. How to run.

I'll show you right now.

I also use the same command

to verify whether anything is running on the backend.

Do you say eight zero? Eight zero?

No meaning backend is not it started to start.

That is a command in the build folder.

You have index file that you need

to execute with which command?

No.

So it'll hold what you say terminal

or basically your GI badge will be hold.

Now you can't do anything. So

that's why I said open one more new session again

with TI user and then verify.

Now the backend system is functioned, right? Yeah.

Uh, during when we, when we joined the organization,

we joined the organization

and does all the commands, will the organization provide us?

They'll, but you need to know why they are provided

and what is the work, what is the need of it.

Just like how I was explaining you step by step. Okay?

Okay. Right? Yes.

Okay. Now we can see our backend system is running on this

particular port called 80 80.

Now the problem is this will only work

till I connected over.

You see what I'm doing?

If I cancel using control, C

88 is done, meaning it is not

continuously running in the background.

Like how Postgres is running continuously.

Uh, I is not to run it continuously.

Third party, that's called as PM two.

By default, it'll not be there if I go ahead and check.

Now PM two, we'll install

and using this, we are going to run the backend continuously

a process product, sorry, production process manager

for no J applications.

Pretty simple. All you need to do is PM

to start on your application.

Our application is index rate.

Now you see it's not holding the process,

it is running background.

You can check here, zero status online. How I can verify

now, no issues, right?

Everything is working perfectly fine.

I can see 88 is there 5, 4, 3, 3.

Is there, I can still do my work.

Now I don't need this additional thing.

Here it is. No backend is running.

My database is also continuously running.

Uh, now I just same.

So you can also verify like this

from your local server itself.

Using the curl command, I'm gonna hit API.

This is a message you're supposed to get.

If you get this message, it means backend is running.

You can check here or you can also check with your ip,

verify your browser

and you should get some message like this.

Let's see,

8 0 8 0 slash my API

clearance.

Right now, the backend is working,

but this is what you can't do much anything about backend

unless there is a front end to invoke the things.

What is backend? I said right.

When you click something, what should happen?

This is the goal. Now what

I need to host the front end to use this back.

Video: LMS Frontend Build

They said change the directly to web app

because a PA was backend code, web app is frontend

is all my frontend code.

Same repetition. What I'm saying

in this environment file.

I'll update the backend details in the first ENV file,

updated database details in this EN, we backend it.

It's already there with this part,

but I need to update from my public end

because front end will load from the internet.

Got it. So now what I'll do

again, same process.

Get the dependencies. See only

two folders as of now

Only see one question only sequence

of commands, which we have.

Uh, if we forget this command, will we be able

to get it in somewhere or

You can. But for

that, like I said, no, you need to understand
how the things work.
What we discussed in the last,
so front frontend connects with backend.
Backend connects with database.
On the basis you need to go ahead and check this
and this is completely developer part.
We like literally speaking, I don't know all this stuff
unless someone from my development team told it.
This is how it works.
So this right configurations as well.
We don't do or we'll do, right? As a DevOps engineer,
Which one? Which
confirmations
The all these installation, which you, oh, you
Need because this is something
that we will write it as ACR next.
Okay, Now I'm doing manually, right?
But in real time we'll not do it right.
We use scripts, docker scripts
and all of it, which are, that's why I'm showing you
before development, how the things are working.
Uh, I would say before DevOps.
So from DevOps, what are all the things
that you take from development operations and you continue
Currently, this is
before the automation platform manual task,
which we are doing right now.
Uh, you're basically learning
how developers actually build an application so
that in the future you will do it by yourself.
Okay? Alright.
And if the build instructions are incorrect,
then never the things will work.

And then when that's when you basically revert back
to the development team, you'll ask why it is failing,
what dependencies are required, what ports you need
to enable, which environments you need to add.
That is the thing that you
as a DevOps simulator is responsible for.
Making sure that entire flow is properly set
up Right?
Okay, now I did the install
again, same like earlier node model schemes.
Now again, same thing. I need to do the build.
So you'll get a folder called us dis, which we don't see as
of now because build was not there.
Now what it does, it'll take the backend URL you have given
up, it'll embedded it into the front end patch.
That's how frontend know when I click something,
how to go to backend.
When I click backend, it knows how to go to the database.
It's all because of the values that you kept in e
and b Cliff everywhere.
This is the important point.
Next time when you get the issues
that DevOps this the issue it says it cannot connect to
and cannot connect to database then
where you will check in your environment files whether you
have given the correct address, correct username,
correct password.
These are the things you're going to check out.
Install license version. Mm-hmm Does it ask any per geo?
Uh, same thing. Same thing. Okay. Process.
It just like prompts you and you need to update the case.
You did the build. Now you see
I got this dis folder.
Now what I need to do, I need to take this dis folder.

If you see the content inside the dis

folder, it does make sense.

Index images, JavaScript files, all front end.

Now this is what the application,

but I cannot see it in the CLA.

So this needs to be hosted with on a web server.

You already know what's the purpose of a web server.

That's what exactly we need.

Once you're done with the build, now you need what?

A web server. This part, you're already clear

only frontend you hosted earlier with the login app.

Now I'm going to literally do the same thing

but not login app,

S app Already saw it.

We don't have anything running on 80 is it?

There's a thing that is missing,

but this will run what This default welcome page, right?

We don't want this. We want what? Our application.

Do you remember document route concept where

www HTML, that's where you clone your project.

Now I cannot clone because GitHub does not contain builds.

The server builds.

Now these builds I need to load with in the document, which,

how did you load it now?

I built it and upload it.

I hope you remember this part to contain some sample page.

I don't want this right?

I'm removing everything that is there in that location.

Recreate the folder and you see what I'm copying Once.

Web app. This all the content in the dis too wire.

Now if you see the

whatever I built, now it is available in the next server.

Now let's see what it

ah, this thing you'll not see

because it's a backup from here we update the courses
and all of it, the admin portal.
But concept is same.
So you see, I'm going to say admin portal over here.
You guys can verify the same way.
One second. Let me complete.
I'm saying add a course.
See is it the same information you had
earlier in the tables?
Yes. This is where the data will be fit.
Now from front end it is loading 180.
Uh, let's say something like
that's what we did.
Technically it can be an application,
but end of the day, whether you have a front end backend
database, connectivity, all work,
whether it is the main thing.
Now when I click on this create button, what it is,
it'll hit the backend.
The backend is gonna take these details
and load it in details.
Code added it says you need to update that live class URL.
This is what you'll see later on in your portal
from the student search.
When they add some new, they're going
to add in this particular banner like this.
So this demonstrates that.
What, how all the things are connecting together.
How do I troubleshoot
and understand this is what I'm going to watch step by step.
See all this work? Reason because
do you see four three to running 80, 80 running, 80 running.
Correct? That's the reason why when I went
and refreshed also everything is working absolutely fine.

Any problem? Now

let's try to mess up the things.

I'm stopping post response.

What happen happened to five four threes?

I stopped Postgres. So can I request database ma?

Because there is no one to respond.

Earlier it was working fine, right?

Let me refresh now

what to show.

There is no database. Where can I fed the data from?

You understood the concept based now, no, fine.

Started it back.

Is it working again?

It is functional.

Clear everyone.

Uh, same way. This is

responsible for backend PM two

because we started with PM two start no js,

it has an ID called zero.

No

stopping the id.

It was only now it is

clearly is what I'm trying to say.

So now what I'll do,

I'm clicking is I click multiple times.

Is there anything being done?

Database is there, but there is no one to

take the request from the front end.

We will take the request from front end

your application layer API and currently that is down.

Agree or not where? 80 80.

I understood the logic of a PA.

Now same way. I know why it was done

because I wanted to read that.

Is it back online is 80 80 functional?

Now I also create

because backend

function, if I go back, right?

Same way we have the front end part

which is running on engineers.

No port 80. Obviously if the front end is there,

the application will itself not

now.

You're clear what I'm trying to say.

If you're taking any three architecture application, this is

how it works,

right?

And fine. First, let's fix it.

Now this app is working now.

Now from time to time what code will update? Agree or not?

So let's say I didn't really like this green background.

Uh, so that is all frontend backgrounds

and all these things come under the front end.

So I'm currently the front end only

Landing page.

I opened the index. And what is the background color Now,

like I did in the HTML, it'll not work here.

If I do,

I say

it'll never work.

You need to rebuild once again.

Anything front end backend, any changes happen.

It'll not reflect directly.

Now we need to rebuild, meaning simple.

Go back

how you did sonar analysis for every new change in the code,

for every new change in the code you rebuild.

Once again, rebuild, redeploy.

I do rebuild once

I did the rebuild, right?

Uh, let's say if, if things are really working or not,

because we need to redeploy.

Are you clear? Is what I'm trying to say.

So meaning again, I need

to repeat the same command,

copy all the dis content which was newly generated.

Update it in my thanks. Done. I recopied it.

This for front end. Same thing for backend.

Again, I need to go back, do the backend build again.

PM two, start again. PM two start.

I need to repeat the same process like this.

Ah, now you need to understand

per version 1.1 2.1 3.1 x 0.1,

how I was going and re-analyzing the code.

I need to always rebuild to

this is what you need.

Your DevOps engineer work is not writing the code,

checking this working or not.

That is not your work. Whenever new updates come,

whenever new issues come re

or I would say get the new code.

Reanalyze the new code. Rebuild the new code,

redeploy the new code.

Video: LMS Environments

Supporting that project.

You will always do this

for 1.1, 2.1, 3.1.

That's why I said it's

a repeated work.

Agree or not. Tell me till as a developer,
you always write the update, do the analysis.
Any issues, come rework, do the build, any issues,
work rework.
This is what continuously
happens, okay?
As DevOps, this is the point you need to catch.
Is it it repetitive or not?
But this how we are going to write automation scripts.
That is your work. If you don't know this process,
what you'll automate tell me once.
So that's why as a DevOps engineer, you need to have
complete connectivity of how applications connect and work.
Actually one technology is fine.
I don't know in the future which company you might work
or which projects you'll work on that basis, this
internal things will change.
Frontend is always frontend.
It'll always talk to a backend system.
Backend is always done.
It'll always talk to your database commands change.
Whatever the commands I have given,
we have given it specific to this application what we went.
When you go to your companies
and your projects, you'll get the relevant things
and those are the actions that you are supposed
to write scripts and automate later.
Now you understood why we are learning all of it
and how it is going to connect to the DevOps
clear, right?
So it's repetitive work technically,
and that's the reason I actually, I kept this uh, thing
for you maybe in the,
what I did.

I need to do it across multiple places.

Does it make sense Doing the same thing again manually again

and again and again and again.

No. That's where you are going to make the things better.

You'll implement new practices.

Nothing but new tools you are going to simply is

how traditionally used to work

and you're going to see how we are going

to all of it in the future.

Good, everyone. So next word.

When I start using the tools,

now this project is going to be same.

Output is same. The way you work is

going to be completely different.

I, we introduced tools like Docker,

Jenkins, Kubernetes.

To make this process more

and more efficient, you'll only let me,

so today we did the bill for what, almost one hour.

Now when I, when you guys are done with the docker next,

you'll do it in less than five weeks.

The same thing, everything.

That's what you're going to do.

That's what your work is actually not learning the code,

what it does and all of it.

Why we are running that commands. It is none of your work.

It's all about how I take these things

and make it more efficient, more faster, more reliable.

That's what your work is. Read as a DevOps engineer.

Now what you learned is not DevOps technically.

We just got to know what are the

problems that I need to fix.

Next we'll start using the tools to go ahead.

Now, before going to that, there is one more concept

that we did not discuss release management today.
What I did, I did the build and I deployed it,
but actually there is a process in between release.
So what is it? How the artifacts will
be managed, all of that.
We did not discuss tomorrow showcase, I discussed it.
Then there will be disconnectivity between to this class.
That's why I did not introduce it directly with built
and in between environments will be there now.
So that part will try
to figure out next how to go ahead with it.
Good, clear everyone.
I hope we understood the logic is
now this is very important class,
literally in every tool you're going to work with,
this project will come in Docker, in Kubernetes,
in Terraform, in Jenkins, whatever it is.
So that's why this particular part,
you need to a hundred percent.
See that's why I kept each
and every step what I did throughout these things.
How to build, how to deploy, how
to verify, how to troubleshoot.
Also, you see all the troubleshooting commands.
So make sure you complete this lab to understand
next step you class are nothing but your DevOps classes.
If you want to work with the tools, right?
You need to know what is this project,
how it works on this project.
Only all our tools will be available.
Understood, yeah. That's is once you've done the
lab, terminate this.
There is of no use.

Video: Release Management

In terms of release management, how we manage the things.

Okay, let's have a look at this Arrangements.

I see, this is what I'm saying.

Okay, so just like in the

earlier sessions when we were discussing about uh, our uh,

uh, versioning, I said, uh, in terms of

source code, we go ahead

and keep all our application uh,

code in the GitHub repository, isn't it?

We'll just go ahead and manage all of

that code in the GitHub repository.

Uh, now here, whenever we are going

and working with, uh, what you say, binary code,

we were not storing binary code anywhere, isn't it?

Directly we are taking the binary code

and we are going with the deployment,

but actual implementation when

you work in the real world, right?

We take this particular code, binary code

and store it in a place called this Nexus repository.

You see the flow one, one by one.

First you write the code,

There's a first step, you write the code,

then update it in GitHub, maybe analysis build.

And once the build is done, we are going to use one

repository called Nexus repository.

This will hold what?

Binary code, not the source

code artifacts.

Technically, that's the reason why.

It's also called as artifact management tool. Okay?

And in here, whatever the artifacts you're getting,

I said there can be different versions of the artifact.
Now you see here version 1, 2, 3, 4,
and those versions will get promoted from dev to QA
to UAT to pro.
So next time, whenever I want to pick up the version,
what I want and I keep wherever I want
or whatever the customer who needs it further part,
I don't need to repeat all this process.
Once second, it becomes all option.
So this way Nexus can help you hold the binary code
and you can pick up the versions, whatever you want,
and you can deploy it.
Actually, that's how the actual procedure goes.
Now, this part we are going to implement yesterday, okay?
So the idea is basically to go ahead
and manage different environments
using Nexus is what the lab.
And now to manage this different environments,
obviously I need couple of,
or at least two to three missions to go with now.
So what is the plan?
You know, this is
something that I use for you to demonstrate.
So all together I will take four different server
space, okay?
For you to understand. These are four servers
and same, I'll take two different versions.
one.one version, two.one version.
And it can basically increment in the future three dottx,
four dottx like this.
We can go ahead and increment. So we know this part already.
The code will be there in GitHub.
The source code. That code we are going
to get into build server.

So these are first server.

We will build the artifact, which artifact.

Let's assume first we are starting

with version 1.1 artifact.

So I build the artifact, then that artifact,

I will load it in the second server.

Uh, this is nothing but technically excess software.

So first time I'll build one.one.

I'll update Next time I'll build

two.one and then I'll update.

Okay? Alright. Clear this.

So both the versions, I'm gonna take it

and these versions I'm gonna update inside the nexus.

And once these artifacts come up,

I'm gonna take two different servers.

The third server can be something like Devon, QA server,

and fourth servers can be something like production.

So one version of my product is accessed

by developers are maybe QA engineers,

and the other product is managed by my customers.

You see how the thing is finally working up.

So in order to do this demonstration,

I need four servers, okay?

And as well as for this production environment, you need

to buy the following things.

You need to buy a domain name.

And if required, you also need to buy SSL certificates.

Video: Nexus Setup

Process you need to do a lot of uh, installations

to go with all of it.

So unnecessary waste of time.

So what I'm going to do is I'm
gonna apply the same approach.
I'll use Docker
and with the help of Docker, I'm gonna start, uh,
what do you say is setting up this particular uh
uh, what do you say, tool called Nexus first.
So here for Nexus also this is the requirement.
Same like in the last session.
I need to go with a T two medium instance, uh, which goes
with around four gigabytes of ramen, two CPUs,
and this will take around 15 gigabytes of uh, storage.
Same with Azure.
Also like how we used to create V two S machine,
we are going to create V two S machine
and this ne Nexus works on port number 8 0 8 1.
Okay? And on top
of it we are installing Docker, you see?
And same just like sonar cube,
how we ran one commander.
I just want you to run that particular command nexus.
We'll be reading and then by default the username is admin
and this is the link for password.
Okay? I'll show you the step-by-step process how I went
with this approach and then later I'll create the other
servers also in this sequential manner
that for you see.
So I'm just showing the same thing
what I was explaining earlier,
I just practically did it over here, right?
Okay. This is the same document,
it is the same lab document,
which I did in my previous session.
I just uploaded it in your document also.
And same process you see practically, how am I going it?

I'm going to launch that particular instance, which is going
to be my next server right now.
Let me get back to my AWS account and I'll start it.
I said night. We'll work on port number 8 0 8 1.
See as of now, I don't have any server created.
That was my bill server, which you did in the last session.
Like exactly that is the last last session guys.
You installed NPM and all that now same.
That's the exact same machine. This is my Nexus server.
Now you see I'm saying release,
which is basically like Nexus server
and see same process O 1 2, 22 0.04.
There is no difference in the way you used to work.
That is exactly same T two medium instance
and I'll add the port number 8 0 8 1 additionally.
And when you're working with Azure also
you just update the same thing.
Yeah, 8 0 8 1
and storage increase to 15 GB
because of the requirement that
I set. Now you can
just say same.
Right now I'll go ahead
and complete this setup process by launching the instance.
Now in this I'm gonna install the Nexus software
by following the same documentation.
What I just updated from here. So now I have two servers.
One is build server, what you created in the last session,
and this is an access server bill server is nothing
but what you did in the last, there is no uh,
additional thing to be done there.
Now I'll connect with those machines
and I'll start with you see I'm
connecting with the access server.

You see? Yes, I just connecting to Next Server.
There is no difference in that part.
It's exactly same thing, just giving the P key username
and connecting with the ipa.
You just connect the way I'm doing right now.
Once connected I will show you that
by default on the system there is nothing running on
port number 9,000.
I'm installing Docker first.
That's the same thing I showed you in the previous session.
Maybe before that you see I'm going
with SS TPI install Docker.
You see there is nothing running on port nine.
Uh, 8 0 8 1
just copied that particular command in terms of Docker
and it is setting up the NEX software first.
You see now there is port number 8 0 8 1 running.
I'll just go ahead and browse it.
Just like how we used to do with every server.
It'll take around two to three minutes to come up with, just
be some, give it some time till it's completely running.
Then you will verify the webpage.
Now you can see that Sonotype Nexus re risk coming up.
So this is where our artifacts will be technically stored.
Once comes up, it'll ask you for sign in.
You can see on the right side corner
there is a sign in option.
Username is by default admin
and the password you need to use the below link
username is admin, that's the password.
So if you copy that command
and paste, you'll get the password I'll show you, right?
It's asking you to update a new password.
You can just update whatever the password you want.

Here. There is an option, right?

Enable and disable anonymous access.

We are saying disable anonymous access, meaning simple guys.

Disable means what?

People cannot directly uh, log in without credentials.

If I say enable anonymous access,

now everyone can access the things.

So that's why we are saying disable so that only

with passwords you can connect with this repository.

If not anyone can publicly go ahead and do it.

So it says not disabled will require the credentials.

That's it. Good practice generally.

So we went with the same thing.

Now we are inside this system right here.

You see there is a browse section.

These are default repositories already there,

but we want to create our own.

We'll see how to create them.

Also on the top, you want

that setting cycle, right?

That is where you can create the repositories.

Actually that button, once you click,

it's an administration page.

I'm just duplicating it so that in one page I can view

and in another page I can create.

Here on the left side you've got repositories.

You can say create repository

and give the project name like we are creating

the LMS repository.

No. And one thing here it is type is nothing about

what type of artifacts.

I am going with raw artifacts, meaning any kind

of artifact can be stored.

That's what raw means, zip files, DLL files, anything.

So I'm going with raw

name of the repository.

I'm calling it as LLS. Let's say create.

Now a new repository is created for you

and when I click on it, that is A-U-R-L-S.

This is where I uh, this is the URL.

I'm going to use order to upload artifacts.

Okay, I'll just copy it and I'll keep

it here.

Earlier it was not.

Then once I refresh, I'll say that repository called LMS,

but, but for timing it is empty.

No components are formed.

Now, once a build is done, we need

to upload the components into this particular place.

Let's see, how am I going to do that as well?

Like once the build is done,

we'll get the artifacts no from there.

How am I going to pick up and update it into Nexus?

Also Will.

Now once this part is done, I'm going

with a concept called as GitHub tax.

I said no, things will keep on coming up

as we go and progress.

So this is basically called as tags.

So this is used generally to go out

and work with uh, what you say versions when we are going

and relying, uh, in terms of uh, what you say, uh, versions.

I'll show you like how we can create tags basically.

So these are like basically code versions.

Uh, in terms of source, the same thing in terms of Nexus.

We use it. Look,

it says right GI tags are helpful when you want to point

to a specific release in the chronological order, you can see all the release names are versions, okay?

This is specific to source code.

That's why I'm comparing it.

You see how I created TAG for Notepad plus Plus

that company created 8.6 0.9 8.6 0.8 in the GitHub.

You'll find the same versions are there in

the downloads page also.

You see that's how they're trying to help you understand.

Okay, so this is binary code that is source code.

So exactly same thing in terms

of GitHub will try to perform it.

Actually you see I'm going with Notepad plus plus software.

I mean for reference, when I say download,

I'm seeing so many versions.

Now these are all what binary codes

and these binary codes are coming basically from Nexus like

software, all these versions,

the source code is going to be presenting GitHub obviously.

So when I go into the GitHub repository right now here

you see there is a section

called here.

You see there are multiple tags. 6, 8, 6, 7, 6, 6.

These are all

tags, right?

They both match. No, that's what I was trying to say.

So I want to do the same thing for LMS application, right?

For my LMS application.

Also 1.1 2.1 is there no I want to create something.

Then in that scenario we need to go ahead and use tax.

You see how am I going to create the tag? No.

So whatever the code that you forked,

see zero tags out there, I'm gonna click on the tags.

You see there are no releases.

So we are going to create a new release right now.

So where these version numbers will be picked up
with from the uh, metadata file, uh, I said no,
there is package do json from there,
metadata files will picked up.
You see that versions 1.1 2.1 version.
I said now earlier those versions I'm trying to refer right here.
I'm gonna say go to my LMS project,
I'll move the dev branch because
I'm making the changes in the dev.
You see I'm going into web app
because I will take the front end code.
There's again going and modifying back
and is quite tedious job.
So directly I'm taking the front
end code to help you understand.
So I'm going into web app directory.
You see which file did I opened this index file in the last
session I showed in a index html page I opened
and there you're seeing the background like green.
So I'm going to take these two different versions
with two different colors.
You see the red server, green server, uh,
that's the thing like what change you do,
it does not matter end
of the day it can be any specific thing,
but that versions will be there in the package.
Or Jason one. You see I clicked on package dot Jason
and I'm editing that file.
Currently I'm making it as one.one.
So that's what the version right now I'm taking
see one.one I said commit.
So this is right now the place
where we take the versioning.

Now I said green color, right?

Background color equal to green.

Now you see I'm going back to my repository

and checking the tags.

Now I'll tag to that version. See how am I going to tag?

Pretty simple. It's asking choose a tag or create a tag.

I'm gonna call it as one.one with the

naming convention that you're seeing here.

one.one create new tag, like how you create a branch

and point this thing to specific branch.

Currently, where is the latest code in the dev branch?

That's where I'm pointing it right now.

So what it does, you know, whatever the code,

which is there till that, uh, what you say version, uh,

till the last change called the green color something I

updated 1.1, the code will be picked up.

Actually you see I'm calling it as production release,

1.1 publish.

Uh, now you see like notepad. We are getting the versioning.

Okay, here what I'm just taking some small piece of code.

But in your perspective it can be a big change.

Also depends upon the code change technically,

but concept remains exactly

same, same thing.

I'm gonna make one more new change

and I'm gonna call it S 2.1.

Again, you see guys repeating the same story.

I'm going into index page,

but this time I'm gonna update from green to writer

the version number.

Also, I'm gonna upgrading to what? 1.1.

Hey, I'm clicking on releases.

I'm gonna say 2.1,

it's development release publish.

You see now slowly it is building up how we solve it.

Slowly it's coming up. Same 1.1, 2.1.

All those particular changes we can see is just

how we software notepad plus plus.

This is how we do the tagging for source code.

This is source code, not the binary code

for this particular source code.

We do the bills and then we basically go ahead

and upload it into the nexus, which we'll see.

Video: Upload Artifacts

You see, I'm gonna get back

to my bill server.

Actually, you see guys I'm logging

into which server Bill, server.

Bill server is nothing but the last class server

that you went and work installing Mode and n bm.

Technically I'm logging into the bill

server and I do build steps.

What we did in the last session, simple by what?

Getting this new versions 1.1 and 2.1 versions.

Are there that code I'll get right now,

you see when I said list, nothing LMS is there

but it's of no use.

I'm going to take the new versions.

You see I'm deleting that entire code only.

I want two versions, 1.1 and 2.1.

That code I'm gonna get right now,

see how am I going to do it?

You see as of no, no code, nothing.

Now whatever the tags are there not from there.

I'm going to retrie the specific version.

You see notice there in this machine

and NPM is there so that you can do the builds.

Now I just said right click, duplicate

the link what you past.

So that what it'll download that specific version for you.

It's the source code downloaded. You see one.one zip.

Same way. I'll also go ahead

and get the second version,

right click copy link address,

then same double gate, that link.

So you got one.one and two.one over here,

one zip.

Now I'm gonna extract the content from that particular code,

1.1 code and as well as 2.1.

Now if you see I got both my LMS 1.1 version

and as well as 2.1 version.

Okay? One second.

You see yes, no see I got both the versions, 1.1 version,

I got 2.1 version.

But this does not contain any builds, only the source code.

You see I'm going into that one

Point, 0.1 application

and you see there is no, uh, what you say,

this folder in the web app, you see

only the source code is there.

No node modules, uh, nothing related to build

and all the departments.

So I'll do the same thing. NPM run, build

and install whatever the commands I use

for the build in the last session.

You see I'm doing literally the same thing.

Install and as well as go ahead and do the build.

This will generate me the dis folder.

For version

one.you

can say, now it is building the version one.one, right?

See you got the dis folder

I'm using.

I'm installing this so that I can create a zip file

and I can send it

Now, whatever the thing I was having in the dis,

you see I'm using the same zip command.

What you learned, and I'm going to give the same number

because I need to match

what is there in the versioning part.

Soen dis star, meaning everything in

that particular dis folder will be added.

You see? Now what I have is 1.1, this is binary code.

Now this binary code, I'm gonna upload it where?

Inside Nexus,

yes, that is a command.

Same. It's there in your document also. I'll show it.

We use the following command to go ahead

and upload the artifacts to Nexus.

I'm using the same command on same call, username, password,

then password of what Nexus.

Then upload file

and the file name that you want

to upload is LMS version 1.1, point zip, then URL, the URL

that you had in the nexus.

See as of no,

No, no confidence.

Now I'm giving upload into this particular link. Simple.

I hope you understand the same text.

We are giving the username

and password of Nexus survey, what file to upload

and the URL where the nexus is running.

Once I execute this thing, now you see it said upload.

This is a message I got.

Now if I go back to my nexus,

you can see the artifact portion,

one.one from the build server.

We uploaded it into the nexus.

Same way how I did for 1.1,

I literally do the same process for 2.1.

You see, I'm just going into 2.1. Again, same.

There was no build folder for this.

I'm repeating the build process.

I said in the last session also.

Now with this what you can understand,

you'll always get the new code.

You'll always go through re-analyzing the code.

You do the rebuild and also you go ahead

and re-release the product.

Then deployment. See again, repetitive action.

Every version 1.1, 2.1 x 0.1.

See, this time we are building version 2.1.

As you can see,

same again.

Instead of saying 1.1, I'm saying 2.1. So meaning what?

Isn't it? Again, repetitive action.

The same thing I'm doing for multiple versions.

Again, same command, a replace with 2.1.

This is what the thing

that you need to automate in the future.

Upload it. Now, if I go back to next, I'll see two versions,

1.1 version is there and as well as 2.1.

Video: Download Artifacts

Listen, I can use that link later on.

This link is exactly equal into what the link
that you were seeing in your uh, downloads.
You see this is the command to download the artifacts.
I'll also show you, now technically I'm explaining,
see two versions are right now in the nexus.
I'll take the appropriate version
and then deploy it into appropriate servers.
So for that reason only, I'm going
to create two more servers.
You see again, I'm launching a new server like a dev server,
elements dev server and same process base open to T two.
I think T two Micro is sufficient
because we are only hosting the front end T two Micro
and same key pairs and all that is literally same.
And along with uh, 22, we are adding port number 80
http traffic so that I can access a web application.
Yes, I'm gonna launch this server.
I'm taking the dev dev server IP
because I'm opening one more new GI Pass session.
This is a new session and in this I'm going
to connect with this dev server.
You see I'm connecting with the dev server this time.
So yes here
as usual, I need to set up nGenx software
to access the front.
You see I'm installing nGenx.
You see I'm just going and browsing the IP address
and I'll not get any response
because, oh sorry, I got this engine next response
because that's a default site.
Now what I need to download the artifact, no
downloading the artifacts from Nexus.
That's the command. The same command.
Second, you just need to replace the links. That's it.

Username and password.

Where am I getting that link from? Nexus,

you say copy the link that is there in the nexus

and I'm saying download that link

output file name is LMS 1.1 is what I'm just going with.

Download what you can see. It's saying everything. Zero.

Zero. No because I did a mistake.

You'll see right now I'll try to install

Unzip

going with ZI

and removing because I said no.

I have given the incorrect username and password.

See the username? I just kept like that only base.

See here still I'm getting the same problem actually

install unzip.

So sorry.

And when I try to go ahead now, you'll see the problem.

You see it did not work right Reason

because I have given the incorrect username and password.

I did not download the file correctly.

So I'm removing that file. That's the reason why.

Now I'll go ahead and update the command

with the proper user corrections.

I'm selecting again on 0.1 or whatever the version you want.

Pick up the version because then we want 2.1 version.

So I'm picking 2.1 version for them

saying I'm repeating the command.

That's where the mistake

person, username and password.

You see, yes it is username and password,

but the actual username is what I change.

The username is admin and are the Nexus password.

I updated the password, I'm passing right?

That was my actual password.

Now you see it started showing me numbers.

Now actually the file is properly downloaded.

Unzip that and now you see guys, I got the dis content now.

Yes, this dis content is what I need

to keep in the dev server and the same concept.

I'm gonna remove all the unwanted files first

from our WW HTML and copy like in the last session.

What I copied for now, let's go ahead

and see the dev server will contain the service called red.

You see, backend is not,

that's why we are not seeing anything.

But we can still figure out that.

Yes, uh, we got it in the problem.

This is my dev server and you see guys

what we're saying, not secure.

So the development environment, what we went with it,

we did not think anything about the security aspect of it.

So I just went and launched the server.

And you see I'm also accessing with the domain

in the real world production scenario, what you use,

domain names and as well as you'll be having some, uh,

what you say, uh, secure connections.

We generally call them as H-T-T-P-S connections.

But right now in the dev we don't have,

but this is how I want the dev system.

No issues.

Video: Production Environment

You see you need to go and buy a domain name, GoDaddy.

So this is where you can search

for the domain names that you want to buy.

You say I search for LMS.

It says there are so many different domains with
that relevant name.
If you want, you need to buy them. I see I already have.
So I'm just showing you from the one which I bought at you.
I log in and then I'll show you.
You see I just logged in there
and I already have one domain in this particular account.
That was my domain name just
but with my name, I just got it.
And you see I'm clicking on DNS.
So here in this DNS case, you need to update your server id.
If you want any server to get a need.
That is basically all Ds, like
how you have your mobile number map
with some needs same day.
I'm mapping a domain, uh, with a IP address of the server.
It can be anything, AWS, Azure, whatever the server is.
So it says no, your connection is not secure.
If you enter any sensitive information like card numbers
or passwords, it can be hack.
Hack or attackers can steal.
So I don't want this kind of environment
for my customers now.
So I'm going to set up the production environment right now
and you see I'm gonna launch a new server
for the production environment.
That's my dev server.
Now I'm launching a production server
fraud server and small changes that you will see right now.
So hardware and everything is same one only.
There is no difference in that particular part
with G two micro security groups.
I'll change you
see guys, I'm allowing HJTP.

So this is what we use for secure traffic.
S-T-D-P-S goes with port number 4 43. Okay?
If you want connections to be secure, we need to end about
that, but that needs to be mapped
with a domain name also, right?
You'll see how am I going to do that, right?
Once this server called production server is launched,
you say production server is created.
The last one I'm selecting it,
I'm taking the IP address of that server.
Now I will map it to my domain name
and simply where you purchased there,
you need to go and map it.
I purchased in GoDaddy,
so I'm going in mapping it over there.
I'm adding in your record.
So records are nothing but uh, interest basically.
Which domain name should map to what.
So I'm saying LMS is my domain
and the IP address of the EC2 server
and the type you need to select basically, see
that is my actual main domain.
I need to select an option called a record.
A record means IPV for address and say same.
Now technically you have given a name
to your server rather than going with an IP address.
So how I can manage using that full domain name,
meaning the name over here and the actual domain name.
Now you see how am I going to connect also now,
now I'll open one word, new session.
This is going to be my production set,
same city downloads.
And now I'm not gonna use the IP address any longer
because I have a name lms.my domain name.com

because that was the need we have given to the server.
So I'm using that name now.
So when uh, here the issues basically
this name already I have
for some other parties, that's why it's killing.
But you'll not get this error. I'm just cleaning right now.
So that names are already there.
So I'm just gonna clean it up for you.
Don't, this just came into my system
because I used it multiple times.
Now you see again, I'm doing the same
as dis connecting back.
You see now with the help of name I connected against, not
with the IP address, same way I can also use my application
with the domain I showing
and same process again, I'm installing engine X.
You see how redundant the work is.
I'm doing the same actions again and again
and again multiple times.
Same thing. I'm copying
that artifact called 1.1 version here
because it is production,
environmental production we said will go with one dot
same.
I'm going with one.one version.
And now if you see
I,
Yeah, let's see this.
Now same thing.
I'm un zipping the version one.one this time instead
of two dot
I just copied the same.
Now I need to access the application.
This time I'm not going to access the application

with uh IP address rather domain name.
You see I'm gonna access it with a domain name,
but I need to tell the system now
where the domain name should be there.
So this is the configuration file you need to update.
That's a configuration file.
Once you open the configuration file,
there will be something called s server name.
You see that server name is under.
You need to replace that blank
with the actual name of the server.
That's the name, domain name I'm mapping.
So this is how EngineX knows that when someone connects I
with this ip, this is where I need to send the traffic.
Now we will save this configuration
and we need to restart the system
because we updated the configuration file.
Once I restart,
yeah, I'm going with the same domain
name you see?
Yes. Now I got the green version
because the artifact we went with is green,
but it also saying what?
Same like IP address here, domain name.
But here also it is still saying not secure. No.
Now to secure we need
to again get something called like uh, SSL certificates.
This SSL certificates will be generated on the
basis of domain names.
And this also real time wise, we need to buy them.
Actually same in GoDaddy. How did you bought a domain name?
You can also buy SSL certificates. So.

Video: SSL Encryption Certbot

This will give you temporary https, not like permanent.

This is for free trial only.

Okay, if not, again, I need to invest in, uh,

buying the SL certificates.

So that's the reason why I relied on this site.

Paul Third board, this will help you generate

temporary https for your apps.

If not, you want permanent, you need to buy, you see, again,

I'm going through the process, selecting nGenx,

open Door Linux.

Then it'll give you the guide.

This process you need to go.

I just documented the same thing for,

I'm just installing that guide given in order to enable

that htt PS.

So

as you go through this series of steps, now, uh,

it is asking you email address and all of it.

You need to update this.

Updating my personal will go

with service agreement and all of it.

Just say yes. Uh, now it detected that

that was a domain name.

Now it is asking would you like

to activate H-T-T-P-S for this domain?

And it got number, number one. So I'm saying one.

That one. And then I'm gonna say enter.

Now you see it is saying, requesting a certificate

for this name and you see it has given

you an expiry date.

But it is going to be, like I said, no, uh, valid

for short durations.

Okay? It says for temporary purpose.

Right now you've gott TPS enabled.

You want permanent, you need to buy them.

You see it has got some expiry date.

I think like you'll get around, uh, three months of uh,

free SL certificate with this model.

Now, earlier, my site was not secure.

No, the green version, see, not secure. Now I'll refresh.

You see SI will no longer find it insecure part,

no connection is secure.

So now any data that I pause over here, right,

it can be kind of what you say, uh, uh, hacked

or any issues might be coming back.

I.