DevOps - Module 10: Jenkins for CI-CD

DevOps

Module 10: Jenkins for CI-CD

Video: Repetitive Tasks

Automation and integration, right?

So just give it a thought in terms

of infrastructure automation.

We did the work. Terraform was helping you there. Clear?

Uh, now when I say automation integration,

where the things will fit once,

give it a thought process about the entire course once,

where can I integrate and where can I automate

or just give a think about

what were the repetitive actions we did so far?

Okay? Mm-hmm.

Okay. So where did you do lot of repetitive actions?

In which cla which sessions of your course?

In which part of the course Project one.

Yes. Isn't it a lot of things

that you did manually here

and isn't it repeating for version 1.1, I need

to do all this work for 2.1.

I need to do all this work for 3.1 4.1 x dot.

It's a repeating part, isn't it?

Now our goal is I want to automate this.

Uh, when I say I want to automate,

I'll explain what's the context behind it.

But yes, technically this is where we want

to perform automation.

So technically before automation, this was the scenario.
I just copy the same image for you.
This is a story before automation and integration.
Also, you see all the tools talking
to each other, isn't it or not?
Without GitHub, can I do code
analysis without build?
Can I upload the artifacts into nexus without artifacts,
can I do the deployment?
No. Meaning they, there are some dots which needs
to be connected together to make the flow work.
So now in this flow
and the work we are trying to do is lot
of repetitive actions.
So this repetitive actions, is there any way I can
minimize the human effort and try to completely eliminate
or minimize whatever is possible?
So for that particular part, we go
with a new practice in terms of this thing.
So that is basically what us CIC, continuous integration,
continuous deployments.
Let's see what it is as we progress.
So end of the day, simple, as I already mentioned earlier,
80% of the tasks you already know.
The thing is where to place
what in Jenkins is something you need to learn
and that is nothing but plugins.
Next, you'll see in the next class what are plugins,
why we need, like how you added AWS plugin and connected AWS
and how you added Azure plugin connected Azure
with terraform in Jenkins.
Literally I do, but not accounts. Here we will add.
So tools, how you can integrate GitHub with Jenkins.
How to integrate Jenkins with, uh, n access, how

to integrate Jenkins with sonar queue.
Those are, that
got the point is everywhere.
So that is the whole idea to go ahead
and implement the things in terms of Jenkins,
that's what CIC means.
So by the end of this Jenkins part, what
will write f flow in such a way
that when developer updates the code,
the entire process should be done by itself
from picking up the code to deploying when things fail.
Automated notifications, all that part will say when,
when we are working with a concept call less pipelines.
You'll do all of it step by step in this week, it'll be able
to understand all of it, okay?
Clear from everyone, like online.
Also, I hope you understood the concept of working with,
uh, Jen Kins.
Now, any questions?
Right? That's it. This is what I wanted to teach.
In the next session, I'll slowly start taking our LMS
related to context and then I'll start
then why the plugins are required without plugins by,
I cannot do some activities.
All those things one by one will go ahead and implement.
Then finally we'll write the pipeline, pipeline,
the last class for this pipeline, how to write everything.
Will, will sit it on one particular day, like
where we need some time and then we'll complete, right?
That's it guys. I'll stop the session tomorrow.
I'll go ahead and start writing more jobs in terms
of Jenkins then.

Video: CI/CD Introduction

Frequently deliver the applications to the customers
by introducing automation where
stages of application development,
what were the stages in the application development, build,
release, test, deployment.
These are all application stages only, right?
Others are all that, but technically we can ignore them like
going with the design analysis and all that planning.
But technical work starts from the coding point
of once the code is completed, then we go with the build,
testing, release and deployment process.
So in that part, I want to get automation done.
It is possible, means yes and technically you see here, also
see the official biki definition two one.
You'll get an idea.
See this definition and tell me
what does it resemble like from here?
At least CICD bridges the gap between development
and operation teams
by enforcing automation in building testing and dev deploy.
If I literally replace CICD with DevOps, it's almost same.
You see, DevOps is also something similar.
You're saying DevOps is the process in which you basically
club development and operations
and try to enforce the automation in building, testing
and deploying with reliability and as well as fastness.
But here that automation term basically comes from CIC.
Now, Claire, so almost if I implement CICD,
technically you're implementing
DevOps on top of your project.
But from online I hope it's clear for you.
No, this is practice. Uh, I said, right, this is a process.

To do this, we need some sort of tool.

And the tool name is Jen Kids.

Video: Jenkins Introduction

Almost similar to DevOps and CIC.

Yes. Jenkins season open source automation server.

It'll help you to automate the parts

of software development related to building testing

and which facilitates ous integration

and continuous delivery.

Same definition. It almost now,

definition wise, always, if I keep it aside,

one simple thing, I want you to remember this point.

Simple and straightforward. You have some a, uh, sorry.

Task that activity, you're repeating

that again and again and again.

Simply go and give that process to Jenkins

or tease that process to Jenkins.

He will do it. Simple and straightforward.

Play with the point, what I'm trying

to say, simple and straightforward.

And I think I also kept some sort

of similar thing in the initial part of your post.

Maybe you remember or not

remember this statement,

Right?

And just go ahead and have a look at our thing.

All of these activities just recollect

everything you read in CLL,

all the activities.

So our analysis, writing the code is not our work.

Keep that essay after writing the code,

build sonar analysis, release deployments,
multi environment deployments, everything.
Use commands only, right?
So why can't we automate it
or why did in the CA for this reason, action
project four is nothing but project one.
Automation is project for simply
whatever project one you did, which is this thing.
If you automate, you're done with project.
Introducing automation into project one is end of
S clear everyone.
So technically what we do in this part,
this is what we,
You got developers, maybe a manager
or a lead, someone who assigns the tasks.
This part, we already saw it, right? We'd have issues.
They'll go ahead and pick up the issues.
They'll start writing them who,
and technically this is source code.
Uh, where do we check in all this code? Not sonar.
Something like GitHub. This is developer work.
Now we are not involved here.
And once the code is checked in, next what we need to do,
we'll test other release and deployment.
Also that entire picture, you assume here in this rollout,
and you see now someone is fetching the changes.
Someone is building them, someone is testing,
someone is releasing, someone is deploying.
And who is that?
A continuous integration.
Our CACD server, which is nothing
but right now here
it's a tool.
You can use anything you want.

Jenkins, bamboo Circle, ci.
There are so many tools in the market.
Jenkins is very popular. We are going with this.
And you see later on what it is also doing,
notify whether this particular activity that you're trying
to do this time got failed our path so
that no human intervention is required to go ahead
and check things going wrong.
They will automatically get notified to the development team
as well in this particular regards.
Okay, clear what I'm trying to say.
So now simple and straightforward.
Before automation, how the things look.
First is the thing right
after automation, see what happens.
I kept Jenkins in between all
of them will communicate with Jenkins.
Uh, when I say all of them will communicate, you're supposed
to make Jenkins communicate with all of them.
That is nothing but integration.
And it should be repetitive, right? Automated.
Technically this concept is call us designing
a pipeline for your project.
Like how a pipeline in your house connects different rooms.
Same way Jenkins pipeline will connect different,
different activities
or different stages inside your application development.
And wherever possibility of automation is there,
we can apply it in simple terms.
After the code is done, we did lot of manual activities.
You went and cloned or if new changes are coming, you pulled
or maybe you switch the branches,
then you did the build frontend,
build backend, build commands were there.

Then build might be failed again,
you might give the notification
to a developer saying, boss, this got failed.
Then again, they'll start writing the code again.
Repeat the process, do the build.
If things pass this time, then go ahead
with the next NAR queue.
Again, things got failed. Reword back again. Repeat.
Then going with the release, going with the deployments.
We were doing all these particular activities. No.
So if you are aware of how to work with uh, Jenkins,
you no longer need to do these activities
by yourself with human effort.
This will be, this will be replaced with scrappers.
Clear. No. And I already said why automation is possible
because of this image.
I collected everything so
that I can showcase easily for you.
Tell me if you're not aware of these commands, what we did,
right?
Anything that is confusing, uh, you did not know.
And technically if I ask you
to write a shell script, you can write.
But Shells script is not useful here.
It is not going to integrate and do all that work.
That's why we write something called Jenkins Pipeline
Script, which is
simple and straightforward.
An automated way of
getting the things from source code management tools,
building, testing, deploying in an environment
and giving it to the respective target customer.
Everything is automated.
As a developer, I'll write the code.

I'll just watch A DevOps engineer will create the pipeline.

It'll do all these activities automated.

Yeah, everyone. So I hope you understood the point in terms of Jenkins from online.

Any doubts? Nothing. That's it. Literally speaking.

What is this? Project number? Five. Four.

Project four. So this project five.

So technically project five,

you already know 80%.

You already 20% is just adding automation.

Now you understand the importance of project one Action.

If you don't know these things, what I taught you earlier,

project five is not possible.

If you don't know what is built, then

how do you automate the build?

If you don't know what is released, how do you automate?

If you don't know the deployment, how that is exactly

what we are going to start working.

If you want to do it simple, you need

to understand how Jenkins work.

That's the reason why Jenkin learning Jenkins is very easy

now because you already know what are the inputs

that you need to give to Jenkins.

It is something like you can compare it with your, uh,

generative tools like Chad, GPTR, deep Sick,

how you give an input, it'll process,

you're giving the input to Jenkins moving forward.

Video: Jenkins Setup

This is what I'm supposed to do.

First thing, we need Jenkins ready

to do all these activities

and pretty simple and straightforward.
You have directly a script installation script given.
We can go with. So Jenkins also something like T two large.
T two Medium is not so eight G Berra, it's cost is like
around eight rupe per hour.
The same thing in Azure is D two s, V three
B one s, B two s, no.
Now D two SV three.
This gives you a GB, RAM two, cbs,
because why am I going with this?
Because multiple tools will be there, right?
So it is not possible to go with one simple,
uh, T two medium instance.
That's the reason why we are, And these are
all the things that you can add.
Jenkins runs on port 80, but I'm adding all traffic
because a lot of things we need to additionally add like 80,
89,000 8 0 8 1 different ports.
So instead it can add all traffic
or if not, add relevant tools.
What you want according to the tools you work. I don't know.
Maybe instead of SonarCube in your company,
you might use something else on which port it runs at
that put your Nexus server might be running on some other
thing or maybe an equal end to Nexus.
You have some other tool that might be
running on some other port.
Add it, your choice what the idea.
So now I'll just go ahead and set it up.
And then this is the installation script for Jenkins.
This is also in the official page only.
I will go ahead and show you here
only from the official document.
You'll get the script. Same thing.

Just I copied and paste it over here. I can try it out.

Now let me launch an instance first.

Now, I'll no longer need this instance

that I'll just get rid of it,

which goes with two CPUs in a GV.

Keep uh, selecting this, no, I'll say edit.

And instead of SSH I'll say all traffic.

But you can add whatever the rules you generally.

80, 80, 90, 98, 0 8, 1 year choice of tools.

You're using 20 GB as the storage.

I'm going with, I say

launch in Azure.

I said no D two SV three.

Just check that particular thing in the sizes.

You'll find it. People are doing it in Azure

Now.

You'll see there is nothing running on. 80. 80.

Pretty simple and straightforward. Copy the script.

What I just given you the same script.

It's already there in this document.

You can just get from there also. That's it.

Yes, there's the entire, and install the Jenkins first.

That's the initial password.

Copy this over the server and cut it out.

That's the first password. Later on it'll ask

you to change the password.

You can simply update it.

It says customize if you see here, right?

What it saying, you see

this is almost similar to what we just learned.

Terraform providers, how you have Terraform providers

to communicate with different vendors.

Jenkins says we have plugins to work with different tools.

They're also, I say I use the same term, if you remember Terraform plugins are like kind of

what you say, tools, which can help you connect

with the vendors and then do the work.

No, here we technically call em as Jenkins plugins.

It says you can install the community plugins.

Sorry, you install what you want. I'll go with option.

This option, what I want install. I can check.

It's showing me list of lot of plugins.

As of now, I'm not interested. So I'm saying none.

I don't want to install any plugin.

Once I understand the concept,

then I'll install what I need.

Make sense? So I'll simply send none

and then I'll scroll and say install meaning installing

Jenkins without any plugins.

Then just update the details. Maybe I'll say I,

no Jenkins is ready.

We can start using the Jenkins.

Video: Jenkins Jobs

This is how Jenkins looks.

Initial version and like how

you created a repository in GitHub.

You create a project here in technical terms,

we call them jobs

In Jenkins,

and activity you want to perform is basically called as

what A job and simple.

I just want you to read this below statement.

Eventually understand

All these activities are

what you read in the command line are jobs.

Reading the points as simple as it. That's it.
I'll show you how we can create a job.
Very simple and straightforward.
I said no this you need just 10 to 15% of effort.
What you learned in Kubernetes, Terraform, and all of it.
Because everything you already,
it is just like connecting the dots where it took place.
What if you know you're done that is like, I kept it last so
that you can connect all the activities over
here at the final stage.
That's my Jenkins. I can say create a job here or here.
Anything is fine once read it
and tell me if you understood the concept is saying
something like freestyle job or a freestyle project.
Yeah. So simply I'm gonna say a job
and that's why I'm saying I'll take very,
very simple example so
that you can actually connect the dots simply.
You see what I'm going to say?
Selected the job type and say, okay, any job you create,
you have job sections, general source,
code management, trigger build steps post like if I click,
it'll take me to the specific part.
Uh, generally is where you give basic description,
that kind of information.
Uh, remove the old history, those sort of activities.
Code management where code is required, replacement of git
and gate related activities you'll do here.
Triggers when to do automation.
Uh, when to do task in your basic idea.
Build steps, how to do the task,
when to do the task, how to do the task
post build after task is done.
What you want,

how everything, what you want, how to do, what
to do, when to do after it's done.
Next word in
Jenkins, you can define all these code in CLA.
You cannot.
As of I'm doing graphically don't worry,
but it'll be replaced with code.
Next, just like VP C example, how I showed, I'm showing
so that you'll get an idea first.
If I directly write code again, you'll feel confused.
So I'm showing in a very easy way first
how the things are connected.
Then we'll start writing the code for all of it.
The same thing what I wrote there.
Everything will be in the code next.
Those are basically called as pipeline scripts.
Understood. Got the point clear. Now
this job I I said no.
You can say project or job, anything is finance up to you.
This job should do what
should display operating system.
Oh yes, to display operating system.
Should I need code?
Just ignore triggers. I said no.
I'll tell you like what is the meaning of trigger.
We'll keep it. This is when to do build step.
How to do
automate your build process with ordered tasks like code
compilation, testing, deployment, like
what we did in our that circle.
It is trying to say keep here post build.
Like I said, after like sending notifications,
archiving artifacts are triggering other jobs.
I'll tell you like these options once we start using,

but here now I want

to display meaning I want to do some task.

How I'm going to do it means by saying build step

windows commands, Linux commands.

Maven is basically a build tool for Java applications

and Jenkins related projects.

That's why they have already included

because Jenkins itself is built using Java language

but for the reason it's already preintegrated.

But we are in interested in what

execute shell When I say shell, right?

Simple and straightforward.

Every command you learned in the course can be kept here.

Now all the activities you did were commands only, right?

You can give it a try. So here,

but what I'm trying to say,

save the job here.

Bills means we never executed it. Zero bills.

If there are bills it'll show.

Now I want to run this act,

that play button you're seeing now.

That's the one build.

Now when I click on it, see what happens.

Build scheduled one got executed.

This time zone is with respect to the location.

You see what it's saying? Success, click on that button.

That's what we did, right? To show the operating system.

This was the command we use. I gave that command to Jenkins.

Now Jenkins is doing it on behalf of it.

That's the whole idea of working with Jenkins.

What you know, you need to feed it to Jenkins correctly.

Then it'll do it. If you feed

incorrectly, it'll do incorrectly.

Clear what I'm trying to say.

Good. Got the point is the whole idea is nothing.
But how are you going to use Jenkins
to tell your workflow and then make it do the work?
You don't do it by clicking
and doing all these
activities from
Is it clear everyone? What is a concept?
I wanted to go here, right?
So now, yes, activities being done. I did it one time.
Uh, next time if I do it again, it's manual.
I don't want to do it manual now. It should do it.
See first time it got executed. Now will it execute?
No, right? Unless I don't click this button,
it'll not execute again,
not execute it manually.
But now I'm saying I no longer want to do like this.
I want it to be automated.
I will go to configure
Triggers, read
it, what it saying
code change scenario C next,
but time I can schedule
that is called build periodically
schedule when uh, you cannot say randomly something like
today at 11:30 PM like this.
You cannot say it needs some symptoms
and Jenkins is very user friendly.
If you don't know something, click on that button.
It'll tell you what it's
no schedules will never run.
So the syntax goes like this.
It is called chron syntax.
Crown jobs we call it has this five fields, minute field,
H field, day of month, day of, sorry, month and day of week.

Self-explanatory. It is it confusing. Once you read

in this form, you need to write the syntax here like this.

You need to replace a minute.

However, day of month, month and day of week.

Uh, maybe if you don't know,

it's not very hard to understand.

You can simply take the help of this site,

chron, tab gu

and tell whatever it'll generate the schedule.

Maybe every three hours,

copy this

paste.

It'll show you. At this point of time it'll,

but three hours is very lengthy, right?

So maybe

all starts and it's not so hard to understand.

You see, I'll say day I move my cursor day, right?

I said already values are also one to 31.

In between the days, I'll say 15

on 15th every minute relates

are clear.

This value day of month, you got these values.

Now in between, you can go ahead.

Star wins every day, every month

are

every minute.

In April, maybe like only on 50 in April,

like that you can play around with this sentence.

So according to our thing, we saw it.

So here also I'll replace one star space is the delete.

Now I'll click here. You said identified again.

Do you mean that every minute you want to execute this?

Yes, and I'll say save it.

Uh, earlier, only two times a job done, right?

I'll keep my mouse aside
and see how it is basically going to do the activity
by itself and not anything.
Let's say will it do this thing by itself moving forward,
her job I did not touch.
It automatically happened
and let's see, for another couple of minutes it just goes
and does the tasks for us.
See now I'm not doing it manually.
And why don't you apply the same technique
with whatever you learned so far.
You know how to do it. Tell Jenkins and when to do it.
Now it took timer,
but timer might not be a feasible option
for every scenario Here.
Timer does not make sense, right?
Because I cannot predict a developer will a hundred percent
at this point of time.
Only he'll do the activity means I cannot.
So that's the reason why based upon the code updates,
also you can set the triggers which he'll learn.
When developer updates the code, then only it should start.
We call them as WebEx, like the different concepts will be
there, which we are going to learn next.
But now you see you got that idea in the mind, right?
What Jenkins can do is basically
whatever the activities you try to do, if you feed it
to Jenkins, then Jenkins can get that automation.
By avoiding that human work.
You might say, then if I do this,
don't we need DevOps engineers?
Then no, because your project will change.
Commands will change, tools will change.
Updates needs to be done. All that activities still in it.

Another one for one project, you might do it once,
but time to time you might update
or you might replace with some tools.
Don't know. Depends upon your company's part.
Now if I go back,
you see 3, 4, 5.
So if I see the first job, who started it?
Say the second job and see,
same out third job.
There's the thing. I did not
do it automatically.
Jenkins was basically going and doing all the tactics.
If you don't want that to happen, simple, go back
to configure.
Meaning you can reconfigure your jobs
and as simple as you added the timer, turn off the time,
but it'll not.
I get multiple options. You'll learn as we progress.
So end of the day, simple.
As I already mentioned earlier, 80%
of the tasks you already know.
The thing is where to place
what in Jenkins is something you need to learn
and that is nothing but plugins.
Next, you'll see in the next class what are plugins,
why we need like how you added AWS plugin and connected AWS
and how you added Azure plugin connected Azure
with terraform in Jenkins later,
but not accounts here will add.
So tools, how you can integrate GitHub with Jenkins.
How integrate Jenkins with, uh, nexus how
to integrate Jenkins with sonar queue.
Those are, that
got the point is everywhere.

So that is the whole idea to go ahead
and implement the things in terms of Jenkins,
that's what CIC means.
So by the end of this Jenkins part, what
will write a flow in such a way
that when developer updates the code,
the entire process should be done by itself
from picking up the code to deploy when things fail.
Automated notifications, all that part will say when,
when we are working with a concept called less pipelines.
You'll do all of it in this week.
You'll be able to understand all of it.

Video: Jenkins Plugins

It says plugins will enhance the functionality of Jenkins.
So the organization are user specific needs.
There are over 2000 plugins, which can be installed
with different things like tools related to build,
cloud related tools, analysis related tools,
report generation tools, much more.
All of this you can find in this bureau
here, you can search.
This plugin allows you to configure email notifications
for builder sets.
That's what we wanted.
This
Plugin integrates.
You'd have to,
this is like play store.
Simple. Get what you want, configure, uh,
this is their official website.
You can check here, but in your actual part you can see

here,

go to the dashboard.

Manage Jenkins is where all you do. Uh, settings.

You see here. Plugins, no plugins as of now.

Go to available plugins. Search. I said No.

There are over 2000 plugins.

If you select it, you can install it. Same way.

GitHub. See,

I selected these two plugins because I need them now.

I want code analysis to be performed without code.

I can't do anything going forward.

Build also code required, isn't it?

So I'm doing the same. And also I want to send the mail.

So something went wrong.

It should be able to, uh, what you say, notify the users

that something went wrong.

I'll say install.

And before installing, I'll show you

how the behavior changes.

I will go back to my dashboard. I'll open one of the jobs.

Configure, uh, you see in the source code.

Do you see anything related to get, uh, triggers

no trigger related to GitHub.

Meaning when the developer updates the code in

GitHub, something should start.

There is no trigger to GitHub, uh, post build action.

Also, there is no emails, nothing here.

This is how the settings were.

Now, now I'm going back

and I'm installing this plugins.

Do I set two plugins?

You see how many plugins are getting installed?

All the dependent plugins will be automatically installed.

It's the get client plugin credentials,

plugin tokens, plugin.

Because in GitHub, all of these things are needed.

Accordingly, it'll install all the intermediate things also

first

done.

And I'll go back. I'll get to the job configure.

Now you see how new options pop up

your L credentials, branches, everything.

We can see same way.

GitHub tokens in the trigger

post build email notifications,

which we didn't have earlier.

Now we got it because we have installed the plugs.

You need to install and then you need to configure

are clear is, I hope you got the point.

Now the same thing.

Let's say I want to do the analysis,

but here one thing you need to remember,

Jenkins is not replacement to any tool.

You need Git, GitHub separately.

So Sonar Cube, we needs separately.

Jenkins will not do that. Jenkins will only connect.

So if I want to do analysis, don't I need, uh, sonar Cube?

Obviously if I take

and if I verify with 9,000,

do we have anything on Sonar Cube here?

Of course, because you did not install it.

So we need to make sure all those activities are already

prepared first and then go ahead.

If there is no repository, how can I use that repository?

You need to prepare it all.

If there are no branches, then how can I use it?

Not possible. Same way every tool you use,

it should be already implemented.

Kept ready if you want to use it. Generates.

Now I wanna do analysis.

Not possible unless Sonar Cube is ready.

Good. What I'm trying to say, I'm pretty much same.

Now I'm just gonna go ahead

and start setting up that particular sonar cube

and then I'll start doing this analysis.

But I.

Video: Jenkins & SonarQube Integration

And everything, right?

It is not a big deal now because we already did house on

and all these things work out.

So I'll simply go ahead and log into the server first.

Nothing on 9,000

and yes, it's not a big deal for me right now.

We already have all the commands in terms of

so and all of it, right?

Why Jenkins? We are doing the sonar analysis.

So Docker and all those tools are required for us.

I'll quickly go ahead and do that.

These are all the things which we already did.

There's no difference in that part except going

and keeping the output in the appropriate Jenkins list.

Sonar is ready for us on 9,000.

Now I need to prepare the things, but I'll not do it there.

Why are Jenkins? I'll do it.

That's the reason why I went with T two large

because we need multiple softwares in this sonar queue.

Nexus, your LMS related build softwares in Gen X,

everything will be here on.

So that's the reason we went with the additional server.

Additional. We already know how to go with this.

All credentials studies name is admin.

Admin and we we'll change the password

Like how we did creating the project and all of that.

I do the same thing. Project name is LMS, branches dev

setting, create the project.

Then we need to generate the token.

Same thing. This was what we did there.

He also did the same thing.

Now these details, I'm gonna fill in jackets

that that's how you did uh, what you say, sonar analysis.

If you guys remember

you cloned it, then you went through this,

there's the URN based upon your user.

Based upon my user. I'll go ahead

now better I'll create a new project

and this is repetitive, right?

So that's the reason why I'm creating a job so

that it needs to automate it.

Which branch?

Simple. How did we do the analysis? Take those steps.

This is a project where I need to go on my particular part

where back after learning

automatically the project will come.

Come. So I'll simply say what web,

because not home directly.

It is not in the home directly, it comes in the Jenkins.

So that path context I'm giving.

That's it. I'll try to save

and I'll invoke the job

as of I'm doing manually, but we'll automate it.

I'm running. It

Failed as expected.

See cloning, everything was done cd.

When we went into the CD web, it tried
to run the docker command, do the analysis
and what it is saying failed reason
because this is the command I'm using here is pseudo.
It is not gonna work. I'll tell you why. One second.
So I'm running what Foer command with which user.
See, I'm doing this activated earlier. I used psdo commands.
It worked in the CLA.
I'm using psdo commands there in Jenkins.
It'll tell you who is using this thing.
It is not root user or it is not open to user.
This is Jenkins user.
Now obviously Jenkins cannot be a pseudo user.
No, he cannot do it.
So we need to grant permissions
for Jenkins user to do this additive.
And also technically I want
to get notifications also when things fail like this
notification should be sent so
that we will figure out like what is issue.
We'll see about that notifications later.
But as of now, what I want to fix this issue,
how does Jenkins build with the Jenkins user in
while it has no permissions?
Same like whatever the issue I
got is getting the same thing.
This is the fix
in the CLA unit.
Then add Jenkins user to your file.
Then restart Jenkins.
Multiple methods are there. This is one
method, this is one method.
This is another method.
I'm going this Psdo vs

O

Open.

Just go to the bottom of the page. This is called pseudo.

All your permissions are set here.

See why Psdo group has all permissions?

Because of this Why root user has all permissions

because of this same way.

Now I'm going to add what my Jenkins user permissions, how

I'll copy paste all of it

and I'm telling don't use any password

or any authentication moving forward.

Now save the changes to save, say control X, it'll exit at

that point of time, it'll ask you to write, hit enter

control XI said it's say, do you want to modify?

Hit enter. Yes, I think we need to type Y.

Sorry, Y and hit enter.

It's not VA editor, it's called Nanor.

So once again, if you want do it,

say control X, it'll last you.

Do you want to save? Save Y? Yes. Then it'll close.

Save control X. Unchanged.

So it just came out right? Uh.

Now once you do this, then you need to restart Jenkins

to check to think what you say to update this change.

It is started then.

Now I'll go back

then, oh see no analysis.

Nothing still like this. Only now

I'll go ahead and try to do the job.

Once I let it fail

because of permission issues, command was everything.

Almost same. There is no change.

Once again, I'm trying to rebuild.

Yes.

Yeah, just, just keep it in a slack group.
Once in the employee leave
or something is there that just came up.
Now you see we have the third job successfully and
because you have fixed issues, permissions were not there.
Earlier. We fixed the issues
and you see it went and updated.
I did not do any of it, right? Jenkins did it.
Or technical. I would say you taught Jenkins how to do it.
Next time I'll not do it one time, I'm doing it
next time it'll automatically do it how you'll see.
But I'm first teaching, this is how,
this is what you need to do.
And moving forward, it'll repeat this process.
Sorry, because don't I need to repeat for 1.1, version 2.1,
version three, 0.1 version.
Next time onwards, Zen kins will
automatically get it done first.
Clear. So it's not much different
what you did earlier in terms of
what you learned at that point of time.
The same thing. Literally we went in, we tried to
keep all those activities in terms of check kits.
Okay, clear, good with the part. Makes sense.
Read what I am trying to say here. So same way one by one.
Now I did the analysis.
Next word I want to do there build
and that two, what analysis, when it should be done.
Next time when we go ahead
and update the changes, then it should start working
with the things, isn't it?
Uh, now this is the idea. So like I did this.
So next what I want to do,
the build all and I want to connect.

Once this part is done, then go ahead
with the next one and then start with the things.
Now, in that particular part,
what I might prepare one more job to go
with build, agree or not.
You see right now.

Video: Jenkins & Build Integration

Is code analysis.
Same next to what?
LMS, maybe like front end algorithm
because I don't need to again, set up the database
and all that, which you already know you do it.
I'm going with the front end
because almost same operations, everything, code base,
everything is same
for the same branch is same.
Obviously commands would change.
Do you think it'll work? Let's see.
White got fit, see there?
Meaning what environment is not good, isn't fit
without this?
Can I do the bits possible? But it failed.
But what if I would have got this thing as a notification?
It would have been much better, isn't it or not?
So that someone can, the team can identify
and, and administrator, right?
Fix it. When developer is running this activity, uh,
now I want to receive this as a notification.
Can we do it? Yes. Email notifications.
We need to configure. Why not? I will do this.
I'll go to this job. Configure

Already.

Notification, plugin installed. Got it.

So it is asking whom you want to send.

Maybe this user space, multiple.

Your team. You cannot add developer, DevOps, engineer,

a system admin, project manager.

Everyone. We can include their mates. See?

I'll one second. Go ahead and do the build.

Let's see if it'll send a notification or not paid.

Obviously.

See what it is saying.

Hey, you're trying to send the mails who can send

and receive the mails in general.

So if you wanna send a mail, what software you need? Gmail.

Gmail, Yahoo. Outlook, correct?

Is Jenkins an email software? Jenkins cannot do it. Simple.

I already said you need to teach everything to Jenkins.

Jenkins is trying to use a default port called 25,

which is SMTP port for sending the mails.

Generally

simple mail transfer protocol.

This is also another protocol

you're not able to send because there is no one

or there is no software in your system.

Like S-S-H-S-S-H

22 htt P 80.

Do you have any mail software in the system?

Of course it'll not work. You can install if you know how

to set up the mail servers

and all of it, if not use, which is already defined.

Gmail is there, outlook is there, Yahoo is there.

You can use any mail, but you need to do the integration.

How you integrated GitHub, how integrated sonar queue.

We need to integrate email software.

So Gmail is proper, right?

So I have taken the example of Gmail.

I'll give you the documentation again,

follow up, step by step.

You'll be able to get it done.

I tried doing the bill failed.

Uh, now I'm saying I can integrate notifications.

I'll give you the process. Follow this process, document,

you'll be able to get it done.

Video: Jenkins & Email Notification Integration

It is most commonly used by what applications?

Like Gmail, outlook, apple Mail, Yahoo Mail, all of it.

Anyone who wants to send

and receive mail, SMTP should be there.

Simple and straightforward.

Anyone who wants to do S-S-H-S-S-H server

and client should be SATP.

Same client and server is required. Browser is required.

Engine X server is also required. Same way here.

Now here what I'm going to do, I'm using Jenkins

to talk to Gmail.

Then via Gmail, send notification.

So obviously now Jenkins needs to log into Gmail,

so it needs what Authentication

and SMTP details of Gmail.

You'll find it on internet. When you search.

These are the details, which quote number

from which account you're trying to send,

the mails, the password.

You need to follow this document.

If you do this step by step, you'll get it clearly is what I'm trying to say.

And where do you do it? There's a process.

Go to Manage Jenkins. Configure email notifications,

dashboard Manage

Jenkins, configure system.

Scroll down. You'll find notifications in the system page.

Fill these details

from which email you want to send the notification.

That's the email, not the receiver.

From which account you want to invoke this mail from.

From mail technical, no.

For that I have another account with me. I'm using that one.

This one

Promise This two is the other one.

And this, yes, we have a domain name with respect

to this one, we went, so make sure you

log into the tech one first.

Then I need to go with this approach.

Password. You need to update this password.

You need to update from your account.

I'll tell you how that is not Gmail password,

like access tokens.

Also for Gmail also, you can generate tokens.

Generally we log in with passwords, right?

But from the software perspective, it's all tokens.

IT hub token, sonar token, docker token here.

Also, Gmail token username is nothing but ID

Password, not the password that you're looking for.

Copy this link and one more thing in which

account you're trying to send the mail from, from account.

In that account, you need to enable

two factor authentication,

like OTP based login should be there.

Just check in the Google. You can do it.

Enable two factor identification for Gmail.

It'll go ahead with the process

of mobile linking and all of it.

Then you copy this link.

Go to the account that you're working

with in that account page.

This here

separated multiple tokens, deleting, which I'm not using,

not passwords, tokens, which account token.

This. Account token. Then followed by

use SSL port number.

Now you have given all your account details

to send the mail from.

Okay, and you want to get some personalized mail.

You can do it here. Also,

system admin, email address here.

You can give@therate.ai or whatever.

If not, you can give some text message like Jenkins

support, like you're getting it from the Jenkins support

like the main, I'll say test it

by giving some email id.

Maybe I'll give my personal email ID

and see if I'm really getting the notification

or not from this address sent to here.

That's my personal email id. Yes, you can see here,

but actually sources from where,

Right And where it got,

this was not working.

Emails can be sent.

Save the settings. Now Jenkins will use this SMTP

details to invoke the mails.

Now, same thing. I'll do the build.

Obviously we know it'll fail

and yes, I did got a notification

and all the details can also be seen

and we checked.

You gotta know what,

Now I can verify it, why it's failing.

Okay? There's no note, okay? Pretty much.

I already know how to deal with this stuff.

As an administrator, I check the mail.

I'm going and fixing that issue for the team of development,

installing northern NP.

Also, I got in mail out

back normal.

Yes, just folder everything came.

You can verify also here, go back to your job workspace,

web back dis all.

Look at

same thing.

What did no Jenkins did?

Next time, I'm not going to do all of it.

I'm configuring, I'll leave it next time. Once it'll repeat.

Now can I connect both the jobs

sonar cube will tomorrow I'll go with the deployment also

and I'll connect all the things in one.

Then developer will write the code.

He will push this entire thing works by itself.

Further, there is a concept called rev hooks.

Also tomorrow is, see all these things you already did,

but the things that you don't know is how to use gen.

That's what you're learning.

Video: Webhooks - GitHub & Jenkins Integration

Attendance.

So in the last class, right, we had the uh, what is it plugins installed.

Then we got to know

how we can extend Jenkins functionalities in case if you

don't have certain features.

Now today, uh, we'll go ahead

and try to see how automatically things can be started.

Earlier, what you manually went

and invoke the changes, right?

So now the idea is whenever a developer updates the code

automatically process should be

because technically that's what the concept is, right?

If you remember our previous, uh,

concept stone

Based upon the developer updating the changes,

then the process should begin.

But as of now, we are manually going and updating it.

So we no longer want to do that.

It should be completely automated.

So for particular part,

we have a concept called as webhooks.

We saw timer, we saw manually in working with the ur.

Third one is basically going with webhooks.

Webhooks basically give you you the functionality in such a

way that based on an event you can invoke something.

That event can be a push.

A developer pushing the code is an event.

Based on this event. Some action should occur.

That action is invoking the job.

So those, you can get it if you have a

concept called as webhooks.

And this will work with a plugin.

We already installed GitHub plugin.

Now at that point of time, this thing was already there,

but we did not discuss today.

We'll go ahead and see. Okay, so let me explain the concept.

Once

Web,

That's a concept called web hooks.

Web hooks inside Jenkins.

As I can see here, A web hook is a

mechanism to automatically trigger the build

of a Jenkins project.

Basically job in a response to a commit push

to the GitHub repository.

Click see the image.

Also

developers pushing the code to GitHub.

It'll invo it. Webhook Jenkins will start the job.

It'll do whatever you, if you want to run a bill command,

test, command shell command.

You want to send a email, you want

to write a database script,

database update, whatever you want.

It's your choice. Anything that you did in the CR live

or via scripting, you can implement it.

Clear so far right now here.

Uh, now the question is two things

they should know about each other.

GitHub should know where Jenkins is.

Jenkins should know which GitHub repository should work.

So that part, we need to go ahead

and do the connection, like integration.

Again, integrate GitHub repository with Jenkins

and Jenkins with respect to GitHub.

So how we do it, let's first

go ahead and try to figure it out.

You see in the last class we did this job. I agree.

This front end build is there.

How we did, we went and we manually
triggered button like this.
Then it went and it went work the job, isn't it?
If you see the job status,
I don't want it.
CACD basically starts from GitHub.
I said, if you guys remember,
so a developer updating the code should be the event.
Now let's go.
So first I'll do it from the Jenkins perspective.
This is a job that I want in work, configure
this option.
But if you take the sequence,
this is not the correct sequence.
What is the correct sequence from GitHub
to code analysis, then code build, then code release,
then code that agree or not.
So if I add webhook trigger here, it does not make sense.
I'll add it to the sonar, then followed by, I will do
that option of build.
After other projects are done, then I'll connect the dots.
Got the logic C I'll go back.
This was the thing that we are supposed to start first.
After the code is done, you do the analysis.
That is for sure. So I'll say configure
GitHub and you already know.
So I can do it manually. No problem.
If I said Bill, now uh, it might uh, fail.
Reason because IP address will change.
I'll show you right now.
Pretty much it'll fail right now
because sonar cube IP address is different.
When we configure today, the IP address will be different.
Hello?

See, it failed.

Obviously the reason for failure is

cannot be reached because IP address

is different from today's part.

That's why it failed. But technically they

should automatically start.

And obviously I got a mail also.

Pretty much if you see I go back to I

because I did not do the mail.

Sorry, that trigger I did not update, right?

So here I need to add post mail actions,

email notification sent to the now I'll get it next time

I let it in go because I did not update it as simple as it.

But now you got the proper problem, like why?

It is basically failing. So what I'll do right now,

so I already showed you I enable the webhook trigger now.

Now you might ask how does it know

to which GitHub it should connect

because you already updated what now, what the clarity.

So Jenkins knows which you are learning to connect.

But GitHub does not know which Jenkins to connect.

That's the case right now.

What I'll do, we'll jump back

to our project, our LMS project

settings,

web books.

There's some old web book which I created with my system.

I'll just delete this off.

By default, you'll not say anything.

It'll be like blank.

That's the concept. Web hooks will allow

external services, external services.

Here is nothing but what Jens,

it'll notify Jenkins based upon this particular change.

You're going in, going here.

Now, if I just go back

and say web hook, it says payload URL,

Jenkins Jenkins.

This post receive meaning basically is called hook name.

I'll tell you what is the hook name.

Let's jump back to Jenkins first.

GitHub web is the name that's fixed.

You need to always give this name.

It is called hook name for GitHub is the hope name.

Uh, and keep the things

and say item C on the push event.

Let's give a refresh. Now

it exists basically.

So now does your repository know which Jenkins to connect?

And Jenkins knows which repository to connect.

Good, connected. I'll save this.

Uh, before saving them,

I also changed IP address base 18.

1 3 3 2 3 2. Seven.

Four. Not 18 9,000

because sonar runs on that particular port.

I'll save it. And for your part also just for

verification, because in the last class when we did the

analysis, it'll show you last 23 hours ago,

24 hours ago, something.

So it's

last update was 24 hours.

Right now think as a developer,

a developer got a new, uh, issue assigned.

He picked up the issue day one class in GitHub.

Some issue was assigned.

He picked up, he start working with it.

This is developer machine.

So what he did,

as usual,

I just changed the color code, but it's the same thing.

Visual studio code. And if not, I will revert back to

as usual.

I think already my thing is already flow.

So if you want, you can flow like as usual,

how we use studio in the documents.

LMS is our repository. I already have it. You clone it.

If not, you have the UR and everything, right?

You already forward it, keep yours.

You see I got all this stuff here,

getting the latest changes over here later.

So currently, uh, what I'll do,

I'll update some change like

what package js is there now what is the current version?

1.2? No, I'll say 1.1 just for

checking the part if it is really working or not.

So this is purely developer going

and updating the things in this particular, uh,

what do you say, environment.

Now I'll say update 1.1,

commit sync.

Now, when I sync, it should automatically go ahead,

invoke the sonar analysis.

Let's see if it's going to work or not.

And if things fail, it should also basically give us the

notifications sync.

I'm pushing the change.

Push the change. Let's jump back to our Jenkins part.

You see, I did not do any job rate.

It automatically got in work by itself.

Then success,

most probably, I should also receive a mail also back to normal things are working.

I'll also go back towards sonar queue.

It was earlier 24 hours ago. Now the latest code is done.

One minute meaning things are working,

meaning developers write the code automatically.

Things are being Now, once this analysis is done, what next?

Wait, after bid, release, after release.

I want to do the same. Now let's go ahead and figure it out.

Meaning what I'm saying end of the day.

Video: Build & Release Artifacts - Nexus & Jenkins Integration

Developer write the code, push.

Run it now with the analysis. Next Bill.

We did it in the last class. Now I need to connect.

The release I want to do means I need Nexus software.

So what I'll do, I'll also go ahead and set up Nexus

and also what I want to do, deployment.

I want to do it in real world. What happens?

Not everything will be there in Jenkins server obviously

like how we did build server different.

Uh, you remember the lab which I showed you.

The video build was different, release was different.

Dev server was different. Broad server was different.

Same way in real time also it's the same case, right?

So sometimes Jenkins wants to connect

to some other mission and do the work.

As of now, everything being done in Jenkins,

but I'll show you a scenario

where Jenkins can jump into other missions

and get the work done for him.

That's called Jenkins.

Master and worker configurations similar
to Kubernetes master and worker mode.
Jenkins will be master, he will do
or he'll tell other missions will do.
Got the clarity. Let's figure it out.
So first thing in this machine, I want
to see Nexus is there or not?
It's not there. Let's set it up.
You already know how we can set up the nexus part.
Pretty simple for us.
All the documentation is already there.
Already. Docker is installed so I'm just going
and using this particular command
and I need to log into the system as well.
One second.
No, that's not a big
deal for us because we already have Docker in a minute.
We get Nexus
80.
Rate one with respect to things are ready.
It'll take one, two minutes for us to get the things.
Then same usual story. Create a repository called LMS.
Get the URL, give it to Jenkins.
It should basically take the artifact
and you already have the commands
for uploading artifacts also,
Same thing I gave in where Jenkins.
See you're nodding Anything new, whatever you did earlier,
we are just placing and we are using
Jenkins functionalities one by one.
Now we need the default password.
Now that's the command to get the default password.
All is already I taught.
So I'm just going in a quick way

because we already know all of it.

Admin signing.

Next, confirm password.

Next,

visible anonymous access.

We don't want people to

outside directly download the things.

Finish, go to server admin repost, create repository,

raw hosted project name.

Create

same.

Right? Till here everything was as usual what we did.

Now I'll get back

my part.

This is done. You see Now configure

or send a notification then sorry, I did the wrong

build other projects after this.

What is the next project to start

build Now in the build

A one small change email notification build

here

taking the URL to upload

which file to upload which

file do j

this folder.

But we need zip file because individually going

and loading one file does not make

sense for that.

We need software call just not there.

So I'm just preparing

Now it can run zip commands.

Zip.

Does it make sense? I'm saying whatever the folder you

build, what is the outcome of this?

This only. So this particular list, I'm saying use it here.

Understood. Just removing it. But you got an idea, right?

So now this zip file

upload where

This

The currently I show you the browse, nothing is there.

Build then take then update

the same thing you did in the CLI earlier in the class.

Now we are doing where generates any change

in the command process.

Nothing as it's

same.

Send the mail in case something fails,

Which Uh, now this will go ahead build

and then it'll store the artifact in the Nexus repository.

Correct. Now what again,

I'll do the change and I'll see if it is going to build

and also it'll store the artifact or not.

I want to verify again,

comes our developer guy for understanding

what I'll do One more small update, 2.1

Test process sync.

And automatically we connected both the jobs also now.

So now analysis should be done.

Then Bill should also be started,

right?

All that analysis started, let it go through.

Then next Bill is also started

whether done alert didn't had any artifact

and I did not do anything right?

I already gave everything to Jenkins

came, Jenkins took care of all of it.

Now I want to do the deployment where not in this server,

like in our class, some other server.

Did you get the point? What I'm saying?

In our class we did the same thing, right?

So dev server was different. Prod server was different.

QR server might be different.

So now I want Jenkins to connect to some other server.

Then do the deployment. So I'll create one server first.

Video: Deployment Server-Jenkins Integration

Is my deployment server

to micro, I'll go with only front end.

I want to deploy, keep there.

Along with SSH, I'm also adding STTP traffic

and as usual, I'm just playing say launch.

Obviously there will not be any web applications,

so you cannot access the things.

For sure.

Let's play around

and if for extracting the content

salary process.

So now what? Whatever the artifact that was built, I want it

to be deployed by a job.

Obviously

build, hold, release, already done, deploy.

Now do I need, again, GitHub code and all of it.

It was already built. Artifact is

there, so I don't need anything.

That's why I'm not copying it. I'll simply say,

do we need GI and all of it?

No. Here,

give the commands that you need.

So first of what I need to do, download the artifact.

And do we have the command for that? We uploaded it.

Same way. We need to download the artifact. Ma'am.

I think the command that we went with is this one.

That's the command.

The zip I will remove because I need to give

the file name is,

once you do it, we'll get which thing this folder

And where,

isn't it?

These were the same things that you did earlier?

No, I'm just feeding the same information to Jenkins.

I say, but where this job should run in,

not in Jenkins, in the deployment server.

If I do this build right now where it'll happen

in this mission, we don't want it.

We want it to be done in some other mission.

So that concept is called as what

Master and worker configuration.

I.

Video: Master-Worker Configuration

Master agent, anything you go with is used

to distribute the load

of building testing across multiple machines.

This will give you parallel processing

because multiple things can be done now

improves the build speed

because in different machines you're doing the work

allows better resource utilization instead of going

and keeping everything in one server distributed.

That was the thing that we learned now.

Now this is what you're doing.

I created one your choice

how many you want for a deploy.

I'm doing it right for Nexus.

You can take one server for sonar cubes.

You can take one server because all communicate via

ips only end of the here.

Also we do the same thing.

So here it says same like Kubernetes master is responsible

for manage, so all the administrative tasks will be done,

but jobs will not run here.

Jobs will run where On the workers,

the one which we created, right?

To make this work,

you need to install this plug.

It'll not work by default. You need one plugin.

What is that plugin name? S. SH Build agents plugin.

If you install this plugin,

then only you can use this feature.

I'll show you how. I'll go

back my dashboard.

Manage Jenkins

plugins.

SSH Build Agents Plug.

This plugin will launch the agents over SSH.

So master and worker will communicate over SSH.

Plugin is installed.

LMS deploy, configure.

See, we don't see any option right now here,

like run this job on this mesh in Kubernetes.

How we have labels, same concept. It's not yet enabled.

We need to enable how. Go back to Jenkins dashboard.

Once again, manage Jenkins.

No, this is built in note nothing but your mission.

Let's add new note.

Uh, now one thing you need to be also aware of

ENGOs.

Ensure Jenkins is installed

mastery accessible via web interface.

Already done. Master should be able to connect

to the worker node via SSH

and worker machine should also have Java installed.

If Java is not installed,

you cannot establish a communication.

It is must rules basically if you want to enable master

and worker configuration.

So this part, let's see

how I can going to connect right now.

Add new note. No name may be deploy,

create or deploy.

Web app, remote route.

Direct meaning the other mission directory

where the things you want to perform Here.

I want to

what's the label?

How you have given in Kubernetes. Yes.

Use this node as much as possible

and how do you want to launch it?

Give the address is my deploy server

and

how to connect with this mission.

No credentials add.

Shall I connect with username and password?

You can if you enable password based authentication, if not

as usual.

Username and passwords,

sorry, username and private key.

Uh, id, let's say deploy only prescription reply.

I'm just keeping like that. You can give any description you

want our uh, web app,

username, Private key,

no key store add here.

You need to copy the content of the key.

You already have the key in your laptop.

No copy the content.

I'll open one sir.

Is there in my local laptop only, right?

That downloads 2 5 0 2

dot that's my key.

You can see, you'll see it in the notepad

or GI bash copying the key

I Same.

I copy pasted the key. I'll say add

now it has username, key ips

and then host key verification.

This is like whenever you try to connect, it'll ask,

yes sir, but it's a machine or it's a tool.

It cannot type. Yes,

it does not have interface access, right?

I'll say non verify meaning if username

and password are correct, connect our username

and keys, correct Connect.

Just keep this agent online as much

as possible and I'll save it.

Now you see this is being launched. Let's see what happens.

You see what it's saying? Go ahead. See the logs.

See authentication is successful but what got failed

because in your work

node Java is not.

That's the reason, right?

Same thing like what I can install some Java,

like here J DK 17 or something is there.

If not in your document itself, the command is there

while setting up Jenkins, I have given it.

Now this will install Java

and then we'll verify if it is going to connect or not.

Then Java is installed. Then the Java come in and check.

Now go back. It'll fix by itself.

Why? Because it again verified.

Then next time what it form

Java it form somewhere here you see,

you see this time there is no cross mark.

Now this server is ready. Mm-hmm.

Now you see you connect like this.

You can add as many servers as you.

Now I'm gonna say go back

this deployment

job restrict where?

What was the label you assigned for your worker node.

That's it. It

is matched with OneNote.

So now these activities will be performed on that mission.

Clear. What I what I'm saying. I'll save it.

Uh, when this job should start technically

after the buildup,

after which project

LMS deploy

after build is done, then start deploy.

Now you connected all the dots like earlier.

See now technically what we discussed

Is ready developer will write the code

update automatically.

Code is pulled. Analysis is automatically being done,

code is built automatically updated to Nexus.

From Nexus being pulled in other machine.

We are also doing the deployment.

Same thing. What you read earlier manually.

Now everything is being automated.

You see, once I set up this process,

developers will just get the code updated automatically.

This entire process works good.

What I'm trying to say. Now

I'll test once again,

I'll say 1.1

and maybe for idea I'll also go with the core changes also.

Currently, what is the change greener.

Let's say I want it to be deployed with red version.

I'll save it. I'll say

check automation.

I'll it, I'll sync.

So automatically code analysis will get invoke

because of webhook trigger.

That will invoke build, release,

deploy, step by step.

All the instructions. Let's see,

just observe step by step.

It'll keep on going and doing the activities here.

Just give it one minute. So you just observe, I

done everything.

So build, yes,

took the code, got the zip file,

uploaded the zip file, everything was done.

Uh, deploy, downloaded the code,

deployed the dis and archived it.

Copied assets. So now technically

That's what we were actually looking forward to, isn't it?

So now if you compare the

thing that we started with,

Everything is Jenkins.

Jenkins is doing based on what you are telling.

So you need to know what actually needs to be done.

Then you need to feed it to the Jenkins.

Moving forward, Jenkins will do it. That was the story.
Now I did it manually. Right?
Next I'm going to write all of it as a script.
You no longer need to create the buttons,
like create job and all of it.
I'll write a script. This entire process, a script.
So our analysis, building, releasing, deploying,
everything is code.
That concept, we call it as pipelines code.
Now we did manually, this will help you understand the
concept, but in real time we'll not do it.
Maybe as a beginner you'll do it,
but once you get bit experienced,
everything is in the code format of like how we did
for Kubernetes, Docker, Terraform, same here.
The part is called as pipelines. And again, same thing.
It'll not come by default plugins,
we'll install again, some new plugins.
Then we'll start configuring all this work as code.
Then based on changes in the code, everything works out.
That's what CAC did.
Now you see, according to the definition,
we did got the thing that we are looking or not,
you know, technically Jenkins is where developers,
operations team, project management team,
everyone will be in once place being, building, testing,
deploying everything we can show.
I'll show you one project tomorrow. Like I said, I'll do it.
Couple of projects I'll give so that you will work and give.
Those are like your actual projects
that you need to implement and give it back.
I'll tell you like what to do
that once you have some practice on it.
But now technically we got what we wanted.

Developers writing the code, automated analysis being done,
automated bills being done, automated releases,
automated deployments.
You can go ahead and that's the reason why we did it.
All the things in CLA,
because tomorrow I'm gonna write all those scripts like
how you write shell scripts.
See, but it'll be in the format.
Paul as groovy Jenkins, like
how Kubernetes is taking the input in Yemen.
Jenkins takes in groovy.
Same goal is not different. Same thing. What I did today.
Tomorrow I'll do the same thing,
but instead of doing via graphical interface,
I'll do it via Coach Docker file.
You remember here we have Jenkins file, same to say,
but different format.
We'll say what is that format, how it works,
and then we'll write a complete pipeline
and we'll do multiple tests to verify that it is working
as expected or good.
Clear everyone. Did you understand the concept base?
Now what is the meaning of CACD?
This is literally what CACD means.
Continuously integrate the changes continuously,
build those changes continuously.
Test those changes continuously.
Release those changes continuously deploy those
applications.

Video: Project - Jenkins Pipeline

What are Jenkins pipelines?

How do you create the pipelines
and via core, how do you manage everything moving forward?
We can have everything. So let me start this.
Okay, now let's go ahead this.
So here, uh, in the last session, once the lab is done,
now I have deleted that node actually, which was not uh,
being used by me.
So this particular machine I just terminated
off because not needed it.
So I just terminated it. I will just delete this thing from
here because I don't need this machine any longer.
So I have only my built-in node as version.
We are back to that part. Now let's see what is the concept
of pipelines
in very simplistic way, this is what can
simple whatever the stages you followed in your application
lifecycle development, that series of steps you include,
it's basically a pipeline as simple as that.
Okay? So yeah,
and the same thing instead
of without pipeline also we would
basically did the same thing.
Ma, it's saying that same all it is possible
to execute manually all these steps without CICD pipeline.
But the true value of ci cd is should
always through automation.
Meaning what? Again, I don't want to spend a lot
of time clicking the buttons, configuring all of it.
So that's when we can rely on this particular concept called
as Jenkins pipelines.
Concept wise, you can say this is what it is.
I will just update once. I just want you to read it.
Pretty much. You'll understand the concept.
Just go through this

ones in the mean way.
Yeah, so technically what you did in your project one,
they're saying if you create A-C-I-C-D pipeline,
that will be completely automated process.
If I go to the all these things,
I just got it from their official
documentation, the same thing.
They explain what is pipeline and all that part.
And uh, when you're writing a pipeline,
you'll do something called like creating a Jenkins file,
like how you have docker file, step by step instructions.
Here we have Jenkins file.
Same step by step instructions
and I'll show you how it looks.
Also,
this format is called grow groove script.
Very simple. Pipeline
is a definition you would say agent you're seeing
this is like in the last class.
Deploy is the agent where you want, if you say any,
whatever the node it is there, it'll run it.
If you specify the name like deployment
or built in node, it will run there.
Last class, you executed one job
in another mission, deploy mission.
That label, you need to give it
if you want this activity to be run on.
That's it. If you say any,
whatever the server is, it will run it.
Uh, now yesterday you created three jobs, isn't it?
So that's the reason why I took this example.
In pipeline you have something called
like stages set of jobs.
Each stage is one job.

Job name LMS build
LMS test, LMS deploy whatever the names you want.
You gave it steps there they have these are comments
here in the place of
3, 5, 7, add your commands.
Are you getting the point? Yes.
As simple as we'll take this format.
In this format, we'll go ahead and write your commands.
That's it. How you took compose route,
the docker commands, how you took Kubernetes,
manifest files, wrote those things,
how you took docker file, wrote those things
in Jenkins file.
This is the format we follow
and update your regular commands.
Everything, yes,
when you say configured is nothing
but literally first time we need
to create next time onwards.
You don't need same like
once I create next time onwards, I'll not do this.
First time we don't have any other choice.
We need to do it next time onwards it'll be picked up
directly from GitHub on
go this, I'll show you like right now we'll go ahead
and start writing this particular script file
and then we'll see how it is basically going to behave.
Uh, now the thing is, I said now
by default you don't have the pipelines.
If I say create new, I'll only see project,
but no pipelines, nothing
because we don't have the plugins installed.
So first we need to make sure you install the plugin.
Then only you can go with

manage Jenkins, manage plugins.

This is what we are pipeline is nothing

but a suit of plugins.

We need a collection of plugins

that will orchestrate automation,

simple our complex activities.

We are working with pipeline as a code concept, not that is

where this plugin is used.

I'll select this and along with that I'm going

to install a few more plugins.

Also I'll tell why they're going to be used.

There is a plugin called as pipeline utility steps.

This one. So why this is used means I want

my Jenkins to read my code.

Also, I'll tell you a use case

according to our project

or any project, we have number of versions

1.1, 2.1 3.1.

Uh, where that versions are picked up from actually

In GitHub, where in GitHub

where it'll read from.

Okay, tags are, so how the tag is coming on what basis,

so how do you say this is the version do I follow?

So where it was updated as a DevOps engineer,

will you keep randomly something

or where did you pick up the value from

in the development from where a file called package

js o isn't it or not tags

and all of this you did not randomly go ahead and set it up.

You had some file coming from development.

It is called as a metadata file.

From that metadata file you had versions,

then you picked up the versions from there, isn't it or not?

Not based on that. Only for GitHub you kept for docker also you kept multiple places.

Now I'm saying in the future developers will keep on writing

the code, they'll update the versions

according to that versions.

You start doing the build, you release, you dev,

I don't want to do that manually In the last session

did I do any such thing?

I did not give any versioning,

nothing just I have directly given God zip file.

But do you think it is correct?

So I need to pick up the appropriate versions

by reading the files to make Jenkins read that files.

That is a plugin

pipeline utility step.

This plugin has the ability to read so simple.

What I'm saying is extract

version from which one?

Packaged json via Jen Kinner.

See how they're reading the package json,

they're storing the version based on

that version they'll look same again stage build.

We are doing it same while do it as well.

But just telling you why am I installing the plug

this plugin as usual, forgetting the new job call pipeline.

Uh, along with this there is also one more plugin

works place

this plugin.

So after the work is done, we are getting multiple builds

for each build data is being used.

Why to waste the storage after the build is done.

It is temporary anyway

because artifacts are being stored elsewhere.

So why to waste that storage once the work is done clean up

that is this blocking like

how am I terminating the servers once my work is done

for Jenkins Also it's the same in this way.

What when you do multiple times practice also your server

will not be out of storage.

If not you do a hundred builds.

Now out of storage here a hundred builds also

you do, it'll clean up the data.

Got it? This is something like RM hyphen RF

in technical terms.

Got the point what I'm saying

now I'll install all these plugins.

Uh, you might ask how do I know it means I said no,

this comes only from experience as you work,

but I'm explaining you so that you can understand, okay,

these are something that we do regularly

because daily bills will happen daily.

Intermediate compilation files,

build files will keep on generating for multiple builds.

If you just keep it like that, there will be lost lot

of wastage for resources like especially hard disk.

So what we are doing time to time, we are cleaning it up

and that I want it to be automated.

I don't want, again go ahead and clean up the disk

and all of it in the Jenkins itself.

I want to clean up that part.

So that's why I'm installing the plugin

and in the future when I give some projects for you,

Are you getting the idea like this?

You'll go ahead and start working. I'll give the

requirements later on anyway,

but for timing for pipeline, these are must according

to my part, so I'm gonna say install them.

I said only 1, 2, 3 plugins,

but you see a lot of dependent plugins are also being installed along with it.

Then all the plugins successfully installed.

Now I'll go back to my Jenkins dashboard

now I'll see new item Now you see guys, what you're seeing

earlier, only one job but now you see pipeline job kit.

It says orchestrate long running activities which will span

across multiple agents suitable for building what

pipelines are organizing complex activities

that does not fit in freestyle jobs

earlier, multiple freestyle jobs.

You see how many things we created Now all

of it I can keep in one place that's pipeline.

Are you good? What I'm trying to say

and how do you keep things based

on that particular sentence?

I'll say create new job pipeline job

LMS pipeline selected The pipeline activity here

I can go with webhook triggers if I want.

Same options, meaning this will automatically start when the

change is happening in the code level.

But I need to make the updates according to the ips

because IP addresses are changing, right?

So I need to go to the GitHub update accordingly. Like what?

In the last session,

I'll go to the web hooks.

This thing will not work because it was

my last class IP address.

I'm just gonna delete this thing off.

You'd have iPhone webhook push event, add the things

now and I'll go back.

Define your pipeline using groovy R.

Saying what? Pull it from source cut.

I can write the script here I want,

but this is not, again, same problem.

How many times will you keep on looking here

and update pipeline

from SEM tool?

Meaning what? Select your gi.

Uh, it's asking in which repository do you have the script?

Hmm, obviously my repository information first I'll you

Towards the branch obviously first I'm going

with the de, what's the file name?

Jenkins file. So it is looking for this file in which

I showed you earlier the instructions.

Those I need to give technical.

Let me give it a try now and see how it basically works. No,

I will say, I'll just do a manual thing

to help you understand what's going to happen.

I'll say build it fails obviously. Let's see why it fails.

Unable to find Jenkins reason

because you only said no while defining these Jenkins,

when I say configure,

but in your repository do you have Jenkins file?

If I go back to my branch, do you see any Jenkins file?

Obviously it's not one now

you need to start preparing

and I'll do it as we do in terms of the code.

That's my project. Same to same c dev branch.

I'm creating what Jenkins file.

And same to same whatever they have given earlier.

Some example.

So just copied. This is the place

where you make the changes here.

These are comments in growing double slashes.

I can give echo statements

or maybe I can directly give such command shell

and run whatever you want just for understanding.

I'm writing it. Okay, you need

maybe something like

I'm gonna sync these changes already.

Webhook trigger is also set up,

not set it properly.

I do the push once maybe.

Okay? Mm-hmm

You can see uh, it is going

and picking up these jobs, which we did the analysis.

So that's the reason why it got in

the holding position there.

So I'll remove the web from this because

after this then I think that will start.

So here one second in the branch LMS

name of this, but actually should not have

any See this

on figure figures.

Yeah, should barely do it.

One second.

Let me see if it is picking up the correct thing or not.

I, okay, it's picking up so I'll just try it once.

Yes, running Jenkins

and yes, it obtained the Jenkins file from our repository.

Now I see here,

but just doing the same thing,

what you wanted in terms of command.

Now here

Maybe something

like sonar analysis.

I'm just saying uh, let's say if not sonar analysis

or code analysis.

I just want to see if it is working or not

picked up.

Just now you can see automatically when I made the changes,

it started triggering the job just picked up, started

by our user and yes did to the test
and yes it is went with the for analysis stage.
Now literally all I need to do is take each
and every job that you did are each
and every step you have in your project,
prepare the pipeline.
Everything is simple commands
literally commands what we did.
Now, same thing. We are going to, are you good?
What I'm trying to say, what I'll do
already have it.
So step by step, I'll just go ahead
and update right now instead of that part
already hired.
So just copy based sonar analysis started
such command, cd, web dock run.
I need to replace the sonar cube address site. One second
sonar cube goes with 9,000 foot
and we can see that.
Uh, last project is almost like 24 hours ago something.
Now I'm gonna replace this ip.
Then token we already have in the last session.
This is the core token we had in the last class configure
Token and then I said just
analysis is completed same way.
Now next I want to do the build. Same to same.
I'll just copy paste the commands. What we learned earlier,
same code analysis,
cd, web app install
and BM Eric is,
that's why I said in the previous session also when I was
helping you go with this part,
just doing the command.
That's it. Literally, you need to know what your process is.

Then start telling Jenkins moving forward how it should be.
No bill will be done once the bill is done will get done.
Artifacts. Now same thing here.
Instead of going with, I'll add one more stage
before deploying, we want to do the release.
Already have that stage. One second.
Same. I wrote one more stage release.
Uh, you see here now Alex DF is used
for defining a variable in G groupy.
What variable I'm defining package js, ON.
Read the file Js O in the web app package J.
This is the same script I copied
from here for your reference.
That is the reason why I installed that plugin earlier
what I was discussing
then in that file,
whatever the thing called version is there
because if you see in our code also web app packaged,
do json
whatever the version this is there dynamically pick it up.
I'm just printing it on the screen for your reference.
Then I'm using word zip saying go
to web app lms,
do whatever the number is there.
Zip R, web Baptist.
Here in the last class I directly said LMS dots,
but now LMS dash that version.
Do dynamic version.
Then what you see, I'm going
and I'm uploading where same place now
because that's the repository we created in the last
class with Nexus.
There is no versioning, nothing.
Now moving forward, versioning should be

taken into consideration.
Clear what I'm trying to say makes sense.
This is that stage we are going in releasing it.
Now, same way I want to go
with the deployment for deployment.
Also what I need to do, I need
to download the artifact then go with deployment.
Same concept.
This stage I'll remember and I'll replace it.
See again, there is a new stage which is basically going
with it deploying same part, read the package JO
and you see I'm downloading from same wherever I upload,
download which version, whatever the version
that is coming from package, not JS one five,
meaning it's now not hardcoded based upon developers
what they're going and updating.
It'll pick up that appropriate version, it'll go through.
Then I'm removing the old code unzipping,
copying the dis folder too via WWS.
See this commands any confusion except this part.
This part I said this is
because of the plugins to read the version dynamically.
But these commands you already know.
Unzip, CP command, RM command, double gate command,
zip command, NPM, command, docker commands.
These are all the things we already have.
And you see step by step I'll prepare like this.
Uh, maybe like in the future, deploy LMS to dev,
qa broad UAD.
More steps will come.
Make sense? I'll give you, that's what I said for timing.
I'm showing you one. Later on I'll give you a few more.
You can work and submit. Hmm. Now what?
Once the deployment is done, I said no, I want to clean up

so that uh, we don't have any uh, unwanted data in here.

One more stage, I'll write simple

cleanup workspace.

I said no, this is what the part is.

So once the work is done, it'll just wipe out all the data

but no issues because we already have

nexus everything storing it up.

Uh, this is my pipeline

for development environment, let's say.

So these are all the commands that you went

and you used earlier.

Now they're all present in my

script, isn't it?

Like almost compost. They have given something.

Then I started rewriting my things

according to their report.

Let's say

one thing before this I, I'm trying

to copy and all these things.

Now here I'm pretty much sure

because said I deleted that machine, right?

So I'll just take this machine once.

I just want to do a few things quickly here

because these are all the things I need in order

to perform all these activities.

Now I have all the things ready

and if you see right now I would have some sample

application also pretty much loading from the same IEP here.

Now I'll just jump back and assessing this changes.

Now let's see if all the things that we wanted

will be picked up and deployed or not.

We'll go ahead and see right now

because pipeline is started.

There's a pipeline.

I see it fits the code going
through the code analysis one by one.
Now let's say I went with all these aspects.
Right now I see everything done, copied, downloaded, built
and see which version did it build.
I never said it but it went it read the file,
picked up the version, updated according version,
did the deployment stored it.
Now if I see five minutes ago
here, if I refresh,
I got the version one.one I want.
And if you see the deployment,
because currently the code was which version
I want to change and see if it is working
the same thing earlier.
Yes you did try, but everything was manual isn't it?
You went, you only did the analysis, you only did the build.
You only released it. You only deployed it.
But now you see I have orchestrated
everything in one script.
I stored that script in my repository.
I told Jenkins, use this script moving forward.
That's it, right? First time, like I said,
take some effort to prepare.
Now I will go back to the package or JSON
and I'm gonna say this is my version 2.1
and just like developer, I'm not touching anything.
Now as a developer, I have updated some new versions,
some new functionality was being given.
I'll say this is my green update.
I committed with the changes.
One second. I think I did some other changes.
So for reason it is apart. One second.
There is nothing but GI related all

because when I was doing, I updated some code change there.
So before that I'm trying to push now once I'll fix it,
Just pushed it uh, forcefully.
But error is nothing. But you'll not get it
because I was doing some changes.
I updated it. That's it in my repository.
Now the second version of this project started.
Now let's see if it does everything again.
Once again, by going with analysis,
you can see last like 10 minutes ago the
last analysis was done.
Now it should automatically pull up, sorry,
pick up the code, then go with analysis,
generate the new version, build it again, store it in Nexus,
then go with the deployment,
which is technically this entire circle.
That was our idea, right?
So before automation, how the thing was
you were the one who did everything.
Now, once I've prepared the script
and given it to Jenkins,
now Jenkins is following that script.
So you were work preparing those pipelines
done.
If I just go ahead, two minutes ago,
the latest update was done.
I will refresh. Bill was done.
And as well as version 2.1 is updated.
It's the same thing what you did earlier,
but now what I was able to orchestrate,
yes, this is the process.
Step by step video on the basis I went
and updated, then it started function clear.
Now you've got a nine and literally

in this I only showed you how to use gens.

Other activities we already done.

Uh, same way later on I would say I'll give you a couple

of projects that I want you to work in.

Figure it out. I'll tell you like next when we can

go ahead and start working with.

But the actual use case of Jenkins is this.

You go ahead and prepare pipelines in this particular form.

Good. Now it makes sense where you can use Jenkins actually.

Oh, I said no.

Later on I'll give you use cases with Docker

and Kubernetes as well.

Very, you use, this is my dev pipeline, QN prod pipeline.

I want you to prepare. I'll tell you like

what needs to be done next.

But in the meanwhile I want you

to get completely thorough with this.

All concepts, same thing.

I'll also keep it per your

reference in the document as well.

Once second.

Then Jenkins file everything what we just wrote.

Now, I'm also gonna keep it in your document too.

Same thing. Whatever that code, how to install that.

Pipelines, plugins, everything. I just updated accordingly.

Just go ahead and add it now. You'll be good.

So now what we understood, what is CICD

and why we need to prepare pipelines.