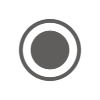
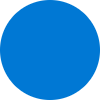
**Transcript**

31 July 2025, 05:45am

 **Vaishnavi Reddy, Attapuram** started transcription

 **Daniel, Ebbie** 0:03  
Open then it will connect to Telius because it's browser based in Telia network. So here this is VDA nothing but how we are having desktop right short button everything the same thing you are able to see here if you go into this icon VDA virtual desktop. So if you can see this see this is.  
Nothing but Telias virtual desktop. So here now how normal desktop we have the same way we have everything here. So it's like completely based on Telias Telias desktop. So here.  
So you will be going through. So I'm just trailing about the repository, some of the repository. So how we should go. So we have this link. So here this is Gitco IN that we are having the group called Telia company. So under this Telia company we have all.  
We we have under this we have ALL the repos so we need to find out. We need to search IT here. So which repo we should do. So basically the say if you are working on Scala or PI or common repo whichever repo you want to work you need to search IT here. So like sample.  
We can there are, there are. I will share something here.  
So repository repo is nothing but repository where the code will be there. So example you will we have common repo.  
Common repo is there. So it's this repo will be used by ALL other repository. So that is what this is called common repo. So here like Nair flow configuration repo.  
Then another repos many repos are there which is specific to which can be used for other repos. So other this like dag using pyspark.  
Then Scala, Scala, Maven.  
OK, so I will. So if for example, supposedly if you are not many squads, right? SANA.  
DFA.  
P2X.  
B2X then RA. So there are multiple squads are there. OK, so you will be working on any of the repos which is present IN SANATFARNT. So like common OK common repo I will tell later but here right?  
So mostly we are working on the migration part. Migration is nothing just we are moving, we are migrating the data from CDL. We are referring the code of what CDL we are migrating the CDL to.  
Argus.  
So we we are refer we have the base code. Base code is present CDL so that one we are my using IN Arcus code using pyspark.  
So.  
And Scala.  
OK, mostly for Scala transformation and the data movement we are using for DAG level we are using. So here we need to refer because since we are migrating CDL code and we need to refer CDL code first. So based upon CDL whatever they have defined so that we need to convert.  
So this way. So we have common templates are there, but IT depends actually. So IT will be changed. We need to refer this one accordingly. We should do the changes here. So here the data moment is like last yesterday or some Vinayak mentioned like we have.  
DDL there are data movement is there. What is that edge?  
Egitura then bass.  
Then access. So this is what CDL they're following.  
So now the there is A, change IN Arcus.  
So here the data will be arriving IN raw then curator.  
And access. So this is the flow actually. So some changes we are doing here but we are referring base code of this one and we are doing these changes here. So we with this code we need to do the changes for our.  
Standard. So this is what we are we need to follow. So we need to look this file, we need to look this repo CDL repo and we need to work on this way orchus code. So the DAG will be there and Scala will be there for transformation soon.  
So So what you should know for regarding repository? First we need to know the repository of CDL.  
So we need to find out the CDL repo. OK then using that repo we need to do the changes using there are that will be like DAG pyspark DAG code as well as Scala code.  
So we need for some of the streams whichever who are worked already. So they have written the code for this one. OK example.  
So we need to. I don't know the exact report, so I will make it. I will put IT there later, but for some time what we can do.  
You can available.  
Yeah, we OK, we I don't know the exact repo for CDL, but for time being what I can do, I will share these two for this is for stream colleagues. Am I audible right now?  
Able to hear.

 **Paidi, Hemasundara Rao** 8:43  
Yes, yes.

8:43  
Yes, yes.

 **Vaishnavi Reddy, Attapuram** 8:43  
Yes.

 **M A, Umme Hani** 8:45  
Yes.

 **Daniel, Ebbie** 8:45  
OK so example I'm just referring this. So these two code they have refer based upon the CDL code they have to created 2 repos, one for DAG and another for for transformation using Scala. OK see here this is Scala, this is Python.  
Code. So here an example I.  
So based upon the CDL code, they have created 2 reports. That is what I am I was mentioning here this code. So one is like the tag OK.  
And another one is for Scala. It is for transformations, data movement mostly happening through this Scala code. OK, so here mostly the DAG we can able to see, yeah.  
So here if you just I open one of the repo the the dag repo. So if you check here this is the main branch. So if you click here you can able to see main branch. But here list of feature branch branch are the feature branches I think but the core is working on the development they.  
So they will be creating brands and they will start updating the code here. So here this is the code. See this is 7 months old. So the the time when they have, I mean they have created this repository seven months back. OK so while they are creating repo this is these are all.  
All the default this these are all the default Python code. So we we have one. I think Lekhana might have covered how to create a post. Lekhana have you are this done or?  
Not it. I think you got shared the document, right? I think you got shared that. Oh yeah, so according to.

 **M, Lekhana** 10:39  
Uh, no, no. Uh, yeah, yeah, it's there in the document.

 **Paidi, Hemasundara Rao** 10:45  
No, but we need to go through that as well. Lekhana, add into your KT topics.  
How to create? Did you create Madam? You you are supposed to create right? You also not created.

 **M, Lekhana** 10:53  
OK.  
No, like uh, I was facing some issue so I I couldn't create it that day.

 **Paidi, Hemasundara Rao** 11:05  
Now it is solved or now it is.

 **M, Lekhana** 11:08  
Oh no, I'm have to look into it.

 **Paidi, Hemasundara Rao** 11:12  
Okay. Okay, fine. I'll go ahead.

 **Daniel, Ebbie** 11:15  
Oh yeah, so this is the default files. So since this is created for that, I really stucked.  
OK so this is the default DAG when we creating the repos using the repo create repo methods. So we have one link where we we need to mention the squad stream source stream and what we are creating. This is for DAG so these these.  
This is the format for creating repo. So if you check here so here this is like and it is it will be ending with Maven Spark Maven like that. So this is the initial repo. See here this is this is what that's why we are able to see messages like cookie cutter. So using cookie cutter.  
Here we are creating this repo. So this one someone has created. This is Mayank has created this repo and this one again this one Mayank has created. OK so this is how the initial repo looks like. OK so after this one we need to create a branch.  
OK we need to create a branch then we need to work on that one. So how to create a branch we need to create click here OK click here and we need to click new branch. Now if you click new branch then we need to we need to clone it from main branch or another future branch is these are all some.  
Someone has already done OK development. Somebody has already created and they have moved their code into this repository that's that code will be present in move development and this is another another branch. So now either we are taking it from main branch or someone who has already suppose mostly some cases you.  
You might have you need someone code. OK on top of their code you also should update. So in that case we need to know the exact branch. You need to select the branch and we need to select it and from that branch OK that means you are taking data. We are you are taking code of that branch to your branch. So your branch.  
You should mention here then you need to click create branch so that if you just leave if you leave as it is then main branch code will be used in your branch. So if you want to there if you want to get there there code they can then you need to mention it here.  
Like this. OK so like that you can able to see branch here. OK next you need to select the branch. OK you need to select the branch then you need to work on your branch. So example this is one of the branch. OK if you go to action so you.  
Click action there you can able to see who are all worked on this branch and who has recently pushed. OK so example this this is the development branch. OK then development branch Vinayak has worked and I have moved the code so everyone we can able to see so we can able to get to know like OK this is the branch.  
Recently someone has worked and we need to work on. We need to coordinate with them and we need to work OK. If you're freshly working on this this source for that list then you should create it from the main branch only. But now already the someone has worked so that's why you're able to see this branch so you want.  
On top of that one you need to you need to work and you need to push the code so that we can able to see whether it is deployed or not. We can able to if we can able to see the green tick mark here then it means like it is deployed into the respective enrollment meaning here when whenever we we push the code directly.  
It will move deploy it into into development development environment. That's why you are able to see here development. OK, so this develop product only it won't be deployed, but this will be deployed into development only. So whenever you push the code directly.  
It's not like it's not only moving into into git, it's deploying it into as well. Then we can if we do run the DAG right then you can able to see the result also. So this is what the.  
This what? OK, you got it right. The initial repose and on top of that one we need to create a branch and we need to move the code and and like here right? If you check here, suppose I'm doing some changes. OK, so this see this is development branch selected.  
Now here if I want to change something you need to go into the respective folder and change something. So what you should OK here I think you won't be doing here editing like this also we can do like.  
If you go here and do, we can do, but it's not recommended. You need to go through through Visual Code, Visual Studio, right VS Code. Through that you will be modifying the code.  
So here also we can do but it's not recommended to do. So if it's a simple changes we can do like using insteadof.com just mention that do so you you can able to do the changes. It's like we shall code get editor kind of thing.  
So if here if you check here then here we can go to the respective script and we can modify it and we can commit it like this.  
Yeah, once we modified it, we can click here and we can submit it. This is how normally if there is a little changes if you want to do, but if you are firstly if you are working on the DAG part then you should go with the visual code only.  
Yes code OK then commit after comments it will ask one message to update. OK so then whatever messages you are giving so that will be coming here and it will update like this is the update. So OK here we need to.  
Follow one principles here while we are updating into git and the messages should be starting with feed column. OK feed column. Since this is we are working on this part. I mean then a dag part right dagger Maven. Normally we should use this.  
Feet only OK, but for some connectivity issue or something, if you're working on the common repo there we need instead of using feet, we should use fix FX.  
We need to use fix. OK, you got it right. Two message, whatever message you are updating here. Before that you should use feet normally. Generally we are using feet, but if you are updating something some you are fixing something, I mean the connectivity is you are changing something for government repo.  
There you need to mention fix FAX column. So this is how the Telia will understands what we are doing on that particular report. So while while reviewing right while we are raising PR, PR is nothing but after code changing, after code judging and once we test the DAG and everything once we tested so and take.  
We should take evidence, then we should raise a PR. While raising a PR, we should attach that evidence as well snapshot so they will verify. OK, so then that time while they are verifying, they will come to know OK what we are doing, what are the commits, what messages we have added.  
While come, while committing. So these are all mandatory, otherwise they won't approve the PR. That's why we should follow this principles. I mean we need to follow the steps.  
I got it right. Any questions?

 **Vaishnavi Reddy, Attapuram** 19:24  
No.

 **Daniel, Ebbie** 19:27  
No.  
OK, once you raise pull request, right then we can pull request is nothing but once this development we are putting everything into one branch, right? Development branch. So this branch we need to.  
I need to be merged with the main branch. That is what we are doing. PR means a peer review. So we are pull request actually. So we are pulling. I mean we are moving our code into main branch. So far we don't have direct access to main branch directly we can't.  
Right. We need to create branch and we need to work on the lower local branch, our own branch, then the admin needs to be merged with the main branch. So that's why the PR is coming here. So PR is nothing but pull request. We are example here if you click someone already raised PR. OK, so this is what they have mentioned here.  
So if you click this one, so here it is what it is doing. Vinayak wants to merge this branch in OK this this has two 210 comments are there. OK this has 210 commits here. So that means 210 times they have updated something and they have.  
More they have shared something. OK, so that is more those many comments. I mean those entire branch informations from development. OK, from development into main branch. OK, it's another way into main branch from development. So that means development code will be merged into.  
Merged into main branch. OK once the so but this need approval. OK this need for approval. We need to add the reviewer here. So some case here we get to approve. Is it visible? No. OK to approval is needed at least to approval. So we need to.  
To add 2 members here squad team members here OK then once they one they should they will be reviewing and they will be giving comments so so they will if they give comments right then we can able to see the comments here and after that after we've changed.  
The code after we modify, after we worked on that comments, right, we need to add the comments. OK, this is whatever comments they have given against that comment, you need to mention the updates. OK, I have changed it. Can you please review it like that? We need to mention here so that they will be notified.  
OK, OK. So this is what I happened. So once it is approved right they were they will once they approve we can able to see the message like OK they have approved. So after that immediately when we once we we need to confirm OK we need to confirm like whether it is moved to the.  
Main branch.  
OK then that time if you check here main branch then we can able to see the our information. I mean main branch we can able to see the our code, recent code, updated code we can able to see here and we can able to see we can't able to it will be I mean we can able to see OK recently it is have committed that information we can.  
We can get and we can check the code also to verify whether actually it is merged with the main branch or not. OK main. Once the code is merged with the main branch then this code will be deployed into UAT. OK UAT.  
So for dev we need we need we need different things, different process to deploy into product. Sorry prod prod. OK OK for this one right? If we immediately after committing this will be deployed into test enrollment as well development.  
Development enrollment is like when we are working on develop any other branch or local branch if you commit that that will be deployed into development branch. But here once the data is merged after rising PR right there development branch code will be moved to main branch then immediately this code will be available in.  
Test environment so that the testing team I have who are working on test environment they can check whether the code is working properly or not. OK this is what this one's OK. So this code normally we can see in this DAG right we can build to.  
See.  
OK, here in this call like um.  
You said I'll.  
I think this is the latest one. So here in this dag they we here we can able to see it.  
Maybe this is not that one.  
OK, OK, so here we can able to see some of the DAG code here. So mostly.  
OK this is example I will cover just rightly like this is the main code. This is this is what we can able to see this DAG ID right this we can able to see in where.  
Yeah, so I think they can have you have it covered this airflow UA.

 **M, Lekhana** 25:35  
Yeah, I did show them the UA, like what tasks are happening and all.

 **Daniel, Ebbie** 25:39  
Oh.  
This is the DAG. OK, so whatever you're working here, so OK, it's this is the DAG. OK, normally we can able to see the file name and so we can see this the DAG. So this is the initial program. So if you check here and if you.  
Go to the code.  
Yeah, this code we can able to see here. OK, they by seeing this one we will get to know what they have used here. OK, this is how.  
We have this code. See this is how this is it will be reflecting here. We can able to see here also this code we can. So this is how this DAG is defined here. So whatever you're doing this is the initial thing. So this tag ID only.  
We can able to see here this is the functions to use the version and vector ID and this is how this is defined.  
So not not much here like yeah so this is the DAG part and so this one will call this create task which is present in which is present in task. So this is how we we can able to see multiple task ID there.  
Example, if you come here, this is what task here. So here multiple task IDs like for these are all different different processes you know right? As you are aware the data is moving from different different layers, right?  
So like example here OK downstream task monthly secured. This one you might be no edge to RA task and RA to curator. So these are all different tasks where we can check here task ID every code.  
Will be having task ID. So this task ID OK see even and then GX also. So this task ID is what we can able to see here. OK so this task ID will be nothing. These are all the task IDs.  
So whatever task ID we are defining there that we can able to see here everywhere we can see task IDs. This is task IDs. So validation and this one. These are all task IDs so that we can able to see the mostly whatever we are working on this dag right that.  
Task IDs will be coming there and from the here this will be calling the spark code. OK indirectly it will be calling. We are calling configuration and we are passing this value into artifactory and that that will be connecting to the Scala code. So that Scala code will be.  
From here this it will call the Scala code and there the data transformation will be happening through Scala code. So that's why this Scala code is using here. So here if you check this is a Scala Scala code so here like action.  
Here the data transformation we are the from DAG right? The code will be calling the Scala code to do the transformation. So the data will be moving into the respective layer like either it's a to curator or curator to access.  
So that information we can able to see here. So this is that's why one some people will be working on Scala part. So whoever is working on Scala part, they will be working on this part, whoever working on DAG part, they will be working on this DAG part, OK.  
OK, that is main mostly this DAG part is mostly by Python by Spark and this is mostly Scala code here. So this is how the data transformation I will be receiving at this data defining.  
Ice book table configuration while they're they're storing into access layer right? So that configuration they have written here and so to curated what what they are doing here. So it's like DAG will be calling here and the Scala will do the transformation and then IT will put the file into respective.  
So that's why these are ALL interrelate. Uh, related. OK then then.  
So that I'd see A, data transformation. So here they're creating more columns. OK, so more ingestions columns was country injection date and how the ingestion date should be there, whether IT should be like.  
YYYYMMDD so that information though these are ALL nothing but the transformation. So using Scala they are doing the transformations.  
Yeah, so mostly like example if you I told right common repo. So common repo is like what do I mean like you're working on this Arcus OK whatever code, whatever library you are using, you will be importing right using this one.  
So this like A, these are ALL the common repo. So here this common repo like OK normally we won't touch this one the airflow variable configuration right that that and ALL we will normally we will work.  
So these are all. If you want we can able to get that repo. Actually this repo will be plus an example.  
OK see here the stream configuration A, Python artifact. These are ALL we have used here. So example Python artifact this is Arcos dot whatever wherever you are using Arcos dot so we can able to see that IT didn't here. So like this is.  
What example Python artifact we are referring. So this code you are using there. So normally we won't touch this one but airflow configuration. So here they tell you already defined this one. So example here the Python.  
Arcos dot OK this is Python at somewhere else. The Python artifact we have used here. That is Python artifact. Yeah Python artifact operator Java spice park.  
Operator Java.  
Calculator Java.  
Spark.  
This is how we need to like here the Python is there. So if you OK not this one so like this we need to go OK.  
It's it's like kind of overloaded. So this is like overall information. So this is how we need to know. OK, that's why I need to cover this DAG part as well as Maven and how whatever we are creating DAG right that we can able to see IN UI.  
This is UA. This is what I want to cover. That's it. Any doubt. So other than that you would be working on some other airflow variable. So example.  
Suppose if you want to change some values so our variables so that time you will be working on this common repo. OK like A, for example some if you want to add the variables like for great expectation.  
Like myself was working on A, great expectation. So there like someone I need to add 1 variable. This variable need to be used IN our code so we need to define like this. OK so example after variable under score whatever you are writing here that will be used IN our code. So wherever you.  
If we are using this, if we are using this variable right that value will be taking from here. So that value is nothing but it's mail OK mail ID. If you use this one that indirectly that IT will take the data value.  
OK, the value for this variable is this one. Just like that we need to define the values here. Apart from that the connectivity details example for.  
Here IN so we have A, connectivity details. So this is YAML file. Mostly the connectivity details we have variable declaration, everything we should mention IN this common report airflow airflow configurations.  
Oh, like I think I can wind up today. It will be overloaded. Tomorrow we'll cover this part. So just I will cover it. So like A, variable. So this is 1 variable. So this variable contain the database information, connect credentials actually.  
So like which what is the post, what is the connection type, login, password and so this we won't define this variable directly because it's security is purpose. We have vault is there when to go connect to the vault and get the information.  
So same way whatever you're adding here IT should be reflected IN prod also prod and test also so that that will be while will be while the code is deploying. So during the prod this code will be used. So this is what.  
We are doing, yeah.  
So I will try to download one repo these two so far your information so you can go through IT and yeah you'll get some idea on these reports because mostly.  
Not that much. I mean like mostly we are following some common code but some cases squad to squad IT will be changing actually and stream wise IT will be changing. So if for now if you go through any one repo then that will be useful.  
For you to understand what IT is doing.  
Yeah, any doubt?  
Oh, IT is recorded.  
Yeah, I need doubt. I think you if since IT is A, record that right, IT will be useful for you whenever you are going through.  
Any doubt? Any?  
Your name is Atapor Vaishnavi, right?

 **Vaishnavi Reddy, Attapuram** 39:18  
Yeah, yeah.

 **Daniel, Ebbie** 39:19  
Yeah, and.  
Kantali, can you pronounce your name?

 **Shriya, Gantali** 39:26  
I.

 **Daniel, Ebbie** 39:37  
OK.  
OK, then Srinidi. OK.

 **Sai Sreenidhi, Dongre** 39:42  
Sreenidhi.

 **Daniel, Ebbie** 39:54  
Yeah, any doubt, anything.  
It's overloaded or this is just an introduction, right? It's just how we can use, but this is what we are doing here, so.  
Yeah. Anyone worked before by Spark? Yeah.  
Are you just fresh?

 **Shriya, Gantali** 40:24  
We have idea about it, but didn't work.  
OK.

 **Daniel, Ebbie** 40:28  
Oh, OK. OK.  
Is it useful this this information?

 **Balam, Srivani** 40:36  
Yes, yes.

 **Shriya, Gantali** 40:36  
Yeah.

 **Sai Sreenidhi, Dongre** 40:36  
Yes, yes, yes, yes.

 **Vaishnavi Reddy, Attapuram** 40:37  
Yes.

 **Shriya, Gantali** 40:39  
Oh, OK.

 **Daniel, Ebbie** 40:42  
Yeah, like, I think we can wind up, right?

 **M, Lekhana** 40:46  
Yeah, yeah, we can write it for today.

 **Daniel, Ebbie** 40:48  
Anything you want to add?

 **M, Lekhana** 40:49  
Yeah, guys, one more thing. Today I'll be sending a mail regarding your planned leaves from August to December. So if you have any planned leaves, please reply to that mail.

 **Shriya, Gantali** 40:51  
Yes.

 **Vaishnavi Reddy, Attapuram** 41:02  
Like we don't know like how many days can we like.

 **Shriya, Gantali** 41:03  
OK.

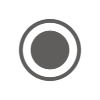
 **Balam, Srivani** 41:03  
OK.

 **M, Lekhana** 41:08  
How many days like?  
Uh, that depends on your like what plans you have.  
OK, uh, decide like when you want and let me.  
And we can drop, no doubt.

 **Daniel, Ebbie** 41:33  
Yeah, yeah. Thank. Thank you, Arun.

 **M, Lekhana** 41:36  
Thank you. Thank you.

 **Sai Sreenidhi, Dongre** 41:38  
Thank you.

 **Vaishnavi Reddy, Attapuram** stopped transcription