provisoner:

we just did not create the infrastructure but we use this concept of provisioner to execute Plus implement some actions during the creation right so what did I do at the time of creation of easy to instance I have copied the file Plus I have executed some commands right so provisioners are resources in terraform or provisioner is a concept in terraform

that will ex that will let you to copy some particular things or execute some particular actions at the time of creation not only at the time of creation but you can also use provisioner at the time of Destruction that means during the deletion of resources also you can use this concept called provisioners by default if you execute it will run like let's say you have created a provisional Concept in your terraform project by default if you are not providing anything they will run during the creation but if you want to run them during the destruction you have to tell inside the provider block that please run this particular activities during destruction now let's try to understand why will devops Engineers use this provisional concept during creation or destruction and what are the different types of provisioners so as a devops Engineers right when you are using terraform you will see some challenges with terraform that means even terraform is very very powerful but still it lacks some basic things example is what we have seen in the particular task right so using terraform you have created ec2 with all the required configuration VPC and all but you were not able to install the python if you don't use provisioners so what will happen let's say if terraform did not have a concept called provisioner the problem would be you need to use any other tools like ansible right or you need to use a shell script to connect to the instances right if there are thousand instances so you need to use ansible to connect to it for this very simple task to just run the python application you have to write a ansible Playbook or you have to write a shell with connecting to the instances and you have to execute all of these things so to solve this problem terraform said okay don't worry I will provide you two things one is called as remote exec provisioner and the other is called as local exec provisioner so what does remote exec provisioner do is if you are using remote exec provisioner at the time of creating ec2 instance itself you can connect to that particular instance or you can connect to that particular resource and what you will do is you will execute any kind of commands it can be installing python it can be installing node.js it can be installing uh Java right anything so any shell commands that you want it can be one shell command it can be 100 cell commands it can be thousand shell commands it does not matter using remote exec you will connect to the instance at the time of creation of the instance and you can run all of these things then there is something called as local exec so local exec is a fairly simple thing so let's say you want to print anything on the console okay while running the terraform project you see that terraform prints a lot of things right so here terraform prints that okay you are this particular resource is created right uh your ec2 instance is created but during the creation itself if you want to say that uh just print a echo statement for your customer or your user saying that okay a terraform has done one out of eight tasks terraform has done two out of eight tasks or you can just say uh create a file okay you can create a file uh for your entire terraform execution Now by default what terraform does whenever you run a terraform apply sorry whenever you run a terraform apply terraform will print everything on the console okay all the resources that it is creating everything is in the console now let's say I want to go back and see the terraform output I can't see because see I'm scrolling down but I am not able to see because your terminal has a specific number of lines that it can show but what if your terraform project has some 10 000 lines of output so instead of printing everything on the console along with printing what you can do is you can use the local exec okay and you can ask terraform to copy all the output to a particular file okay it can be anything like you can show the progress of your execution in this particular file you can say ec2 instance is now created subnet is now created etc etc okay you can print all of those things so this is about legal exec I have explained what exactly is a remote underscore exec right now along with this there is one more very important thing called as file provisioner file provisioner is basically the name itself says that it is used to copy the files okay let's say you have created a RDS instance or you have created an ec2 instance and in that particular ec2 instance you want to copy some 10 files it can be source code it can be some configuration files it can be some Json files yaml files whatever it is okay so using the file provisioner you can copy the files that's what we did right so that's why I did today's demo with both the file provisioner as well as remote exec what I've done is I've used the file provisioner to copy the app.py onto the instance and then I've used the remote exec to copy the file sorry to execute the file that is copied using the file provisioner right so you can use local exec as well in this particular demo and just print that hello that terraform execution is going on right now the instance is created right or anything that you would like to you can maintain a single file and let's say you have 10 resources so here you have one resource here you have two resource three resource in all of these things what you can do okay of course you cannot do it here because these are security groups and other things but for some resources example here when you are using the ec2 instance here what you can do is you can write the local exec and you can print anything on this particular terminal or in a particular file right so this is about uh today's video I will cover local exec in some other demonstration uh because I don't want to complicate this particular thing probably when we do the eks cluster creation or when we do the complicated project using terraform there I will show along with the local exec I will also show the remote exit because there it is most needed as well when we are creating such complicated projects provisioners will come a lot of Handy the reason why we are learning these things like day one day two day three four and five why we are learning things with scenarios why we are learning things with Theory and practical because they will help you in creating big projects these are day-to-day tasks of devops internet but why you are scared when you want to create eks cluster because some people don't understand the concepts that we are learning in day one to day 7 and if you want to directly create eks cluster using terraform eks cluster will be some thousand lines of terraform script for example they will be scared okay it is very complicated terraform is very complex but if you don't have the basic knowledge yes terraform is complex but if you have the basic knowledge I just wrote this 100 lines in just 10 minutes of course I am using this IDE I'm using the plugins I have taken help of documentation but it is very very simple because I know the concept whenever I am getting some doubt let's say I don't know how to I don't remember the syntax of remote it what I'll do similar to you people I'll just go to terraform documentation and I'll search for remote exec okay so terraform will give you the entire thing see it gave me it gave me the syntax that I'm using what did it say resource AWS instance web it provided me the connection block it provided me the remote exit block and this particular thing I just removed this particular lines and provided my things see this configuration is exactly same in my case I just said connection I use the remote exec provisioner I use

the inline and in inline instead of puppet I use the python script right isn't it amazing perfect now there is one simple task to you people I want to see if how many people are following it right how many people are actually understanding what we are doing with terraform so try to execute this project and let me know in the comment section if you are able to do it or not and finally try to debug what exactly went wrong here right it's a very simple thing if you try to understand what exactly is happening you will understand where and what exactly did we miss but uh it's

okay when you are trying to do the initial demonstration follow the same video and if this particular thing is missing in your case as well just try to log into the instance and execute the command and in the comment section let me know what went wrong what did I miss in this particular action and why the python did not run or why the python application did not run directly using terraform so all these 101 lines got executed but what happened to this particular line okay anyways I hope you enjoyed today's video and I hope this video is very very helpful to you people let me know in the comment section what is the feedback and how did you find this day-to-day task of devops engineer and are you confident that you will do this particular task thank you so much for watching the video again take care everyone bye bye

remote backend:

intead of creating statefie in vcm it is stores in s3.

s3 will host the state file

we can restrict the access of statefile using policied and IAM

statefile is automatically updated in s3 once the devops engineer runs terraform apply command

while running terraform init, terraform will understand where is the statefile.

lockfile:

if multiple people are trying to update the terraform file so in that case only one will be allowed at a time to perform the changes

okay let me quickly explain so what you will tell in your interviews that let's say we are a devops engineering team of five to six people where we have a GitHub repository okay we have a GitHub repository that hosts the terraform code and that code will have AWS resource creation logic so let's say we are using the terraform project to create eks or we are using terraform project to create vpc3 Tire architecture and what we do is instead of storing the state file in the GitHub repository itself we will use S3 bucket as remote backend okay that's how if any of the five to six devops Engineers if any person wants to update the eks or VPC 3tier architecture logic they can simply clone the repository on their laptop they can make the changes they will verify the changes using terraform apply and what they will do is with these changes they will raise a pull request back PR back to the GitHub repository and because they executed the terraform apply command this apply will update S3 bucket with the logic what is that logic it will update that okay this particular change is made so I need to track it as part of the S3 bucket perfect and once they raise the pull request someone would approve another devops engineer would approve the pull request and now the new terraform project is ready with the source code that is the terraform project changes as well as the state file is updated in the S3 bucket so this is the workflow that you have to explain to your interviewer

====================================================================