**CD INTEGRATION:**

Search for the Code deploy:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Code deploy -> Create application.

A screenshot of a computer

AI-generated content may be incorrect.

*One application can be considered as one microservice. Like Amazon has payment as one microservice, login/logout would be another microservice.*

Provide the name and select instance as EC2.

A screenshot of a computer

AI-generated content may be incorrect.

Click on Create application.

Now if there are 100 or 500 applications, we need to use the CLI for the same or we need to use the UI and create the application one by one, which would take a very long time.

We need to deploy the application on the EC2 instance with the same steps that has been mentioned above while creating the instance.

TAGS in EC2 instances:

Manage tags option is there in the EC2 instance or during creating the instance, that can also be done.

A screenshot of a computer

AI-generated content may be incorrect.

We can filter out on these tags:

A screenshot of a computer

AI-generated content may be incorrect.

Like we can filter out, which resources are created for the “payments” team. At that point, tags can be helpful.

Multiple tags can be created like for the project or for specific to the team as well:

A screenshot of a computer

AI-generated content may be incorrect.

Tags can also be used by the Code deploy service to identify the EC2 instance, it can be either one or more.

We need to install an agent inside this EC2 instance.

A screenshot of a computer

AI-generated content may be incorrect.

We need to login to this instance using the location of .pem file and using the ssh command.

A computer screen with green text

AI-generated content may be incorrect.

After login, following commands needs to be executed in the terminal for the agent creation.

A screenshot of a computer

AI-generated content may be incorrect.

In the following command, we need to replace the bucket name with the bucket that is available to us and the region identifier with the same.

A screenshot of a computer

AI-generated content may be incorrect.

And bucket name specific to location can be find from this reference.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen with green text

AI-generated content may be incorrect.

We need to start the service using the following commands:

A screenshot of a computer

AI-generated content may be incorrect.

If while starting the agent, if there is any issue, then we again need to start the service using the agent start and stop using the agent stop command.

Next step to giving the permissions to this EC2 instance. As AWS Code Deploy is going to talk to the EC2 instance.

We need to create a role and need to assign that role to the EC2 instance to communicate with the AWS Code Deploy.

Important: Role needs to be created to start the communication between the services. Like if one service wants to interact with or want to talk to the other service, then we need to assign the roles.

Create a role:

A screenshot of a computer

AI-generated content may be incorrect.

Select Code deploy:

A screenshot of a computer

AI-generated content may be incorrect.

Click Next.

A screenshot of a computer

AI-generated content may be incorrect.

This role with name needs to be created.

A screenshot of a computer

AI-generated content may be incorrect.

Now that role needs to be assigned to the EC2 instance.

Follow the same path:

A screenshot of a computer

AI-generated content may be incorrect.

Select code-deploy role and then click update.

A screenshot of a computer

AI-generated content may be incorrect.

After that agent service needs to be restarted.

A screen shot of a computer

AI-generated content may be incorrect.

Need to provide all the information related to Code deploy needs to be provided for the code deploy, and other configurations as well.

And, also for the Code deploy, we need to provide the target folder or the EC2 instance where the deployment would takes place.

A screenshot of a computer

AI-generated content may be incorrect.

Here we also need to provide the access to that role for the EC2 access.

*So, we can either create two roles here, one with EC2 access and one with code deploy access and use the, or use both permissions in one role only, as we are using here.*

Add permissions in EC2 code deploy role and add permissions.

A screenshot of a computer

AI-generated content may be incorrect.

Add permissions -> Attach policies.

A screenshot of a computer

AI-generated content may be incorrect.

EC2 full access.

A screenshot of a computer

AI-generated content may be incorrect.

Return to the code deploy screen. Blue/Green model is where you would be needing the load balancer and other things as well and will be discussed later.

A screenshot of a computer

AI-generated content may be incorrect.

Select Amazon EC2 instance checkbox and search with the key value pair.

If you search with the key value pair, one unique matched instance would be there as shown below and in the similar way, if there are 100 EC2 instances, then we need to configure them the same way, through the tags.

A screenshot of a computer

AI-generated content may be incorrect.

If we match anything, that is not there, then no match would be found.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

No need to load balancer:

A screenshot of a computer

AI-generated content may be incorrect.

Click create and deployment group would be created.

Code deploy application and EC2 instances are created and integrated, and now how to deploy is left, which we need to configure and need to create the appspec.yaml file. For the CI, the file is buildspec.yaml

After creating the deployment group, we need to create the deployment in the AWS. Click on the “Create deployment” button.

A screenshot of a computer

AI-generated content may be incorrect.

Provide the name of the deployment group in the search bar as:

A screenshot of a computer

AI-generated content may be incorrect.

Need to select GitHub, as appspec.yaml is on the GitHub application.

Repository name is “<username>/<repository name>”

Check for the latest commits and paste the latest commit id.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Copy and add the commit id.

A screenshot of a computer

AI-generated content may be incorrect.

After that click on the Create deployment.

A screenshot of a computer

AI-generated content may be incorrect.

Under “View events”, we need to check the status.

A screenshot of a computer

AI-generated content may be incorrect.

Connection is successful but it is not able to fetch the appspec.yaml file, as file is not there.

A screenshot of a computer

AI-generated content may be incorrect.

When you go to the GitHub repository and click on the “.” Dot button on the keyborad, then you would be able to open the VS code.

After opening the VS code, create the appspec.yaml file in the same folder as following:

A screenshot of a computer program

AI-generated content may be incorrect.

We need to create two files, one is start\_container.sh and other is stop\_container.sh under the scripts folder.

A screenshot of a computer program

AI-generated content may be incorrect.

Stop\_container.sh contains:

A screenshot of a computer

AI-generated content may be incorrect.

In the start container, we can add the script for docker pull and then script for docker run. Docker pull command would be available in the Docker hub.

A screen shot of a computer

AI-generated content may be incorrect.

We can rerun this one, or we can delete this deployment created above.

A screenshot of a computer

AI-generated content may be incorrect.

But again, under the events, we will be able to see the error on the page.

A screenshot of a computer

AI-generated content may be incorrect.

Issue is docker command not found.

A screenshot of a computer

AI-generated content may be incorrect.

As Docker was not installed on the Code deploy. Use the command to install the docker.

A screenshot of a computer

AI-generated content may be incorrect.

Then rerun the code deploy.

A screenshot of a computer

AI-generated content may be incorrect.

We can use some commands like date or something else in your script. Like while pushing the image, you can use the “latest” tag, and while pulling that also, you need to use the “latest” tag as well.

Add the CD stage to the pipeline, which we have skipped during creating the CI.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Click on the Edit and add a new stage.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Under Code deploy, we need to add the action group.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Click on the Done and then click on Save. And code deploy pipeline would be created for the same.

A screenshot of a computer

AI-generated content may be incorrect.

Push a simple change to test the CD pipeline along with the CI.

And there is one issue while deploying.

A screenshot of a computer

AI-generated content may be incorrect.

We need to check what happens with the docker image:

A computer screen with green text

AI-generated content may be incorrect.

Python app is also running, checked using the docker ps tag.

A computer screen with green and blue text

AI-generated content may be incorrect.

While triggering manually, the image was deployed at some port.

And, after integrated it with the pipeline, again, we are deploying on the same port, which is causing the issue, without stopping the container as mentioned in the error.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Click on the Link to execution details.

Click on view events.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Issue:

A screenshot of a computer

AI-generated content may be incorrect.

So, we need to update the stop\_container.sh to stop the container and then deploy again, so we need to update the stop\_container.sh file.

Other way is that we can change the port in the start\_container.sh, which is not the recommended way.

So, we can get the container id and just delete that, getting the container id in a variable using the following command.

So, we need to update the stop\_container.sh to stop the container and then deploy again, so we need to update the stop\_container.sh file.

A screenshot of a computer program

AI-generated content may be incorrect.

After running the pipeline again, we can deploy the docker image. And also check that container image is up and running.

A screen shot of a computer

AI-generated content may be incorrect.

*ASSIGNMENT: Now, we need to go to the EC2 instance and allow the inbound rules and access this application.*

A screenshot of a computer

AI-generated content may be incorrect.

**AWS CLOUD WATCH**

Cloud watch is like a gatekeeper and do the following things and other things as well and it basically keep watch on the activities that has been happening on the AWS cloud, creating the EC2 instances, or creating any kind of services on the AWS cloud.

So, like the gatekeeper keeps track of what activities are happening in the private space, in the similar way, cloud watch keeps track of what’s happening in the AWS cloud.

* Monitoring
* Alerting
* Reporting
* Logging

Features of the Cloud watch:

* Plays a critical role in the infrastructure monitoring.
* Real life metrics. Example: metrics like how many requests EC2 instance receive inside the AWS, like how much CPU utilization is there for the EC2 instance for the last 30 minutes. Or you can simply say, give me metrics regarding the Memory consumption, or metrics regarding the CPU utilization.
* ALARMS: like when we get the metrics for the CPU utilization and checks that if the CPU utilization has reached the 80%, then send out a notification on the respective mail and take some specific action, or when reaches 60%, then raise an alarm, that it started getting over-utilized.
* Log insights: any service or person tries to access to EC2 instance, that we can log. Like in the code deploy, the EC2 instance tries to access the code deploy services, so that we can check in the cloud watch with the exact timestamp.
* Custom metrics: CPU utilization is the by-default functionality provided in the cloud watch, but if you want to check for memory utilization, you need to use the custom metrics.
* Cost optimization: Major purpose of moving to the AWS is to reduce the maintenance and improve the security. Second importance is cost optimization, which can be done by cloud watch. Cloud watch can integrate with other services like lambda functions which are serverless to optimize the cost. As the cloud watch is gate keeper, so it has the idea of what is happening inside the cloud, so it can provide that idea.
* SCALING: cloud watch can send notification that scaling group is at 80%, so it can trigger that notification and ask to scale the EC2 instances to the next level, so increase one more EC2 instance.

Search for the CLOUD WATCH:

A screenshot of a computer

AI-generated content may be incorrect.

List of features cloud watch supports are being displayed at the left side:

A screenshot of a computer

AI-generated content may be incorrect.

While creating the project, cloud watch automatically creates the log groups for the specific service and you can check all your logs inside that.

A screenshot of a computer

AI-generated content may be incorrect.

Under log streams, we can see all the available logs for everything we have configured and can check in the logs group.

A screenshot of a computer

AI-generated content may be incorrect.

Entire log of the build process is available in this one and even if you deleted the project as well, still logs are available there.

**A screenshot of a computer

AI-generated content may be incorrect.**

And, we can check the log insights from that log group using some queries which can be written further in the video:

A screenshot of a computer

AI-generated content may be incorrect.

Under Metrics, you can see these following parts mentioned which we can use:

A screenshot of a computer

AI-generated content may be incorrect.

Let’s open the EC2 metrics and following are the metrics created for the EC2 only:

A screenshot of a computer

AI-generated content may be incorrect.

We can open the Per-instance Metrics, we can check multiple metrics, which are shown below:

A screenshot of a computer

AI-generated content may be incorrect.

Like if we click this one, we can check the CPU Utilization of last one hour for this instance as shown below:

A screenshot of a computer

AI-generated content may be incorrect.

Create a new EC2 instance named as cloudwatch-demo as shown below:

A screenshot of a computer

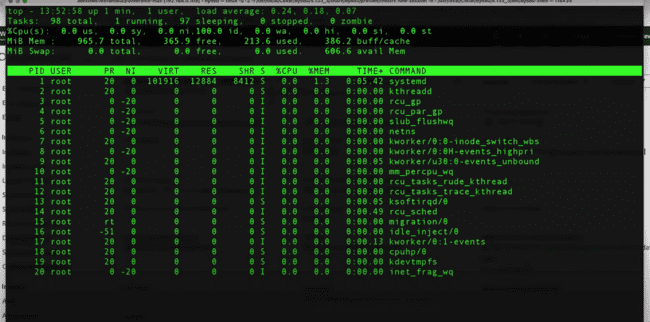
AI-generated content may be incorrect.

Login to the EC2 instance:

A computer screen with green text

AI-generated content may be incorrect.

Use the command “top” to do this one:



Under Cloud watch -> Metrics -> Browse. There would be two links under this:

Per-Instance Metrics: This is for the checking the metrics of the EC2 instances using their name and we can check the CPU utilization and other metrics in this.

Across All Instances: This is for checking the metrics of all the EC2 instances collectively, for all the instances created on the AWS.

A screenshot of a computer

AI-generated content may be incorrect.

Checking the CPU utilization of all the EC2 instances created on the AWS:

A screenshot of a computer

AI-generated content may be incorrect.

Under Monitoring tab, we can check all the information related to EC2 instance:

A screenshot of a computer

AI-generated content may be incorrect.

EC2 instance sends metrics every 5 minutes. So, you can go to the “Manage detailed monitoring”, enables this feature and metrics is sent every 1 minute.

A screenshot of a computer

AI-generated content may be incorrect.

Create the file named as cpu\_spike.py and paste the contents of the file in the GitHub to that file:

A screenshot of a computer

AI-generated content may be incorrect.

Under the EC2 instance, CPU utilization can be shown as follows:

A screenshot of a computer

AI-generated content may be incorrect.

You can get the information in multiple formats as shown in the following formats:

A screenshot of a computer

AI-generated content may be incorrect.

We can check the Average, Minimum, Maximum and sum also in the CPU utilization of the metrics as shown below, and mostly we check the average of the metrics in the AWS.

A screenshot of a computer

AI-generated content may be incorrect.

**IMPORTANT: If the CPU utilization goes to 100% and immediately comes back to the 10 or 20 percent, then it won’t be an issue, but if the CPU utilization goes to the 70 percent for one minute or more in the average case, then it has been an issue.**

Now, when the metrics is there, then we need to initiate the alarms for the same to send out to the developers in the team to act upon that. Like when the CPU utilization reaches 50 percent, send me an alarm.

Create an alarm:

A screenshot of a computer

AI-generated content may be incorrect.

After that, select the metric:

A screenshot of a computer

AI-generated content may be incorrect.

Select the metric, you want for the EC2 instance.

A screenshot of a computer

AI-generated content may be incorrect.

After EC2 instance, click on the Per Instance Metrics.

A screenshot of a computer

AI-generated content may be incorrect.

Select this instance:

A screenshot of a computer

AI-generated content may be incorrect.

After selecting the metric, you need to configure to check that maximum in the 1 minute, if the CPU utilization is exceeding the 50 percent limit, then we need to create the alarm.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

After clicking next, you need to create a new topic as cloud watch-topic and also need to configure the mail address on which we need to send out the notification.

A screenshot of a computer

AI-generated content may be incorrect.

After providing the email address, click on the Next button and provide the alarm name and message which needs to be send.

A screenshot of a computer

AI-generated content may be incorrect.

Alarm is created but not activated. Click on this button, to check whether alarm is enabled or not. This needs to be activated via email.

A screenshot of a computer

AI-generated content may be incorrect.

We need to confirm the same on the mail which has been received.

A screenshot of a computer

AI-generated content may be incorrect.

After confirmation, alarm is enabled, and we can trigger the same, so that we can receive the notification using the command prompt.

A screenshot of a computer

AI-generated content may be incorrect.

Run the same program which would increase the CPU usage, then we’ll receive the notification.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

If we are not receiving the mail, then we can check the same on the SNS in the AWS.

A screenshot of a computer

AI-generated content may be incorrect.

Click on the topic and search for the cloud watch topic and check:

A screenshot of a computer

AI-generated content may be incorrect.

We can also check in the promotions folder as well in the INBOX to check whether email is received or not.

DASHBOARD:

A screenshot of a computer

AI-generated content may be incorrect.

We can create the dashboard for the following widgets on the AWS:

A screenshot of a computer

AI-generated content may be incorrect.

LAMBDA FUNCTIONS:

Comes from the compute family, which is same as that of the EC2 instance, but follows the serverless architecture.

For the EC2, we need to provide the configuration of all the things that are required for the application, like the RAM, CPU, scaling up and down, but LAMBDA function creates all these things based on your application and automatically scales up and down the application as per the requirements of the application.

For example: when you are making an application for the food delivery application, and there is food order service, order placed service, payments service. So, once the user places the order and make the payment, the AWS lambda will tear down the infrastructure, it has created for the payment’s architecture. And that would be done automatically.

But with the EC2 instance, you need to pay as you use, so you need to scale up and scale down the instance and you need to manually tear down the EC2 instance.

So, with LAMBDA, everything happens automatically, so the architecture is called as serverless.

IMPORTANT: In the LAMBDA, there are no servers, so once requirement is there, AWS will create the server for you and once requirements are not there, AWS will tear down the server for you.

A diagram of food delivery

AI-generated content may be incorrect.

The LAMBDA function does not have the IP address, and on which place they are hosted, that information is also not there, and related to the auto scaling group, how they operate also that information is not there.

But with the EC2 instance, user is the complete owner of the server. And IP addresses, subnet information, all the things are there in the EC2 instance.

A diagram of a diagram

AI-generated content may be incorrect.

COST OPTIMIZATION:

Example: if there is an EBS volume which is not used for the last 30 days, so you need to delete that volume and send out the notification to the develop for its deletion.

And if there more than 20 or 100 AWS services, so we can write the LAMBDA FUNCTIONS to govern these resources like monitor and report.

In the same way, we can write the script for the LAMBDA functions for the usage.

IMPORTANT: if you have 10 lambda functions for monitoring and want to run these in the 10am in the morning, so you can use the SERVERLESS architecture for that instead of manually creating the EC2 instances and deleting the same.

We can ask the cloud watch to trigger the lambda functions, as they are only event driven. And after executing the scripts, it tears down the server after the function.

Features:

* Cost Optimization: As discussed above.
* Security

In the security point of view, if someone has created the scope of lambda functions as public or created the GP2 volume instead of GP3 volume, so we can send out the notification to the team related to this thing.