😊 Streaming Deployment logs to CloudWatch Logs

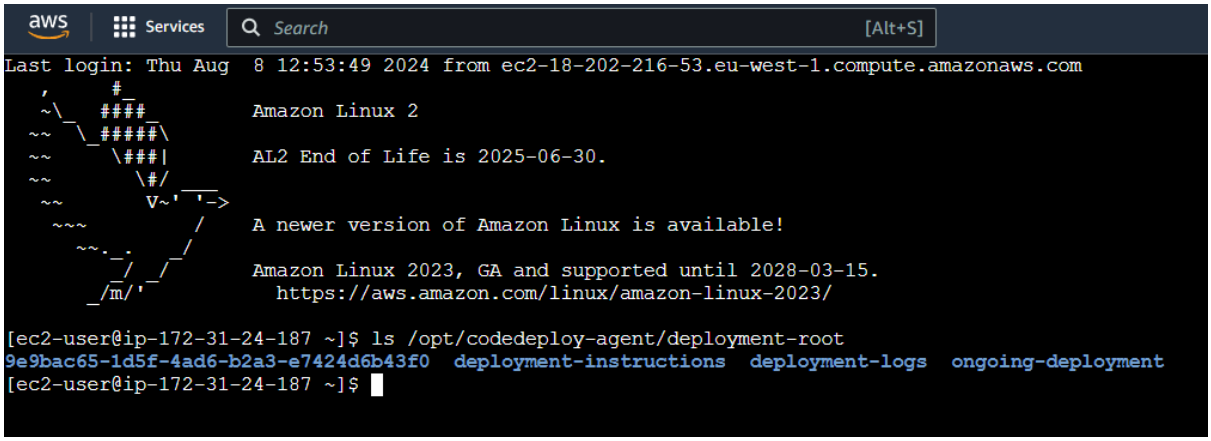
The task involves managing deployment logs on an EC2 instance. Initially, you learn to access these logs directly on the instance. However, manually checking logs on multiple instances is inefficient. To solve this, the logs are centralized by streaming them to AWS CloudWatch, a logging service. This process includes assigning permissions to the EC2 instance's IAM role, installing the CloudWatch agent, and configuring it to send the logs to CloudWatch. The end goal is to automate and centralize log management, allowing you to monitor deployment logs from multiple instances in one place, improving efficiency and oversight.

😄 To begin with the Lab:

😁 Step 1: Viewing Logs

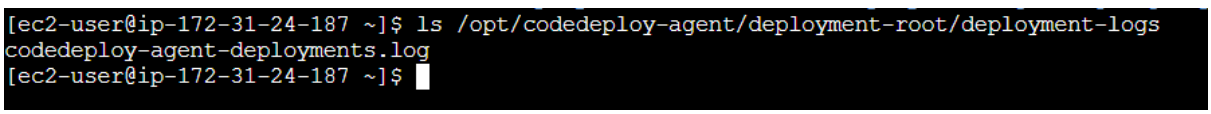
1. In this lab first we will learn how to view deployment logs on our EC2 instance. So, you need to connect with your instance first using EC2 instance connect and then we need to run some commands.
2. To start with, let's view the contents of the folder where the Code Deploy agent stores all data related to your deployments. It is /opt/codedeploy-agent/deployment-root folder. In this folder, there is a deployment logs sub-folder which contains all logs produced by the Code Deploy agent.

   ls /opt/codedeploy-agent/deployment-root



3. Now let's go inside the deployment-logs folder. So, this is the file containing our logs.

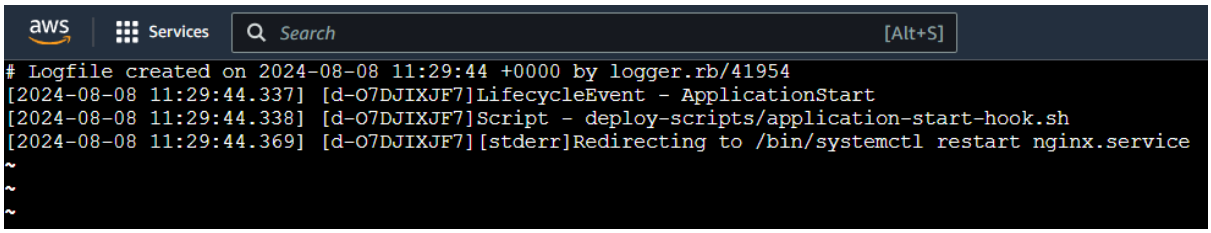
   ls /opt/codedeploy-agent/deployment-root/deployment-logs



4. Now using the below command, you can view the contents of this file. So, this line says that the lifecycle event is Application Start and all logs until another event will belong to this event. Then, in the next log, we see the location of the script file executed in the

revision. And finally, the outputs of the script. By the way, although this log is written to stderr it is not an error. Otherwise, our deployment would fail. For a deployment failure, it should return a non-zero value regardless of where the logs are written.

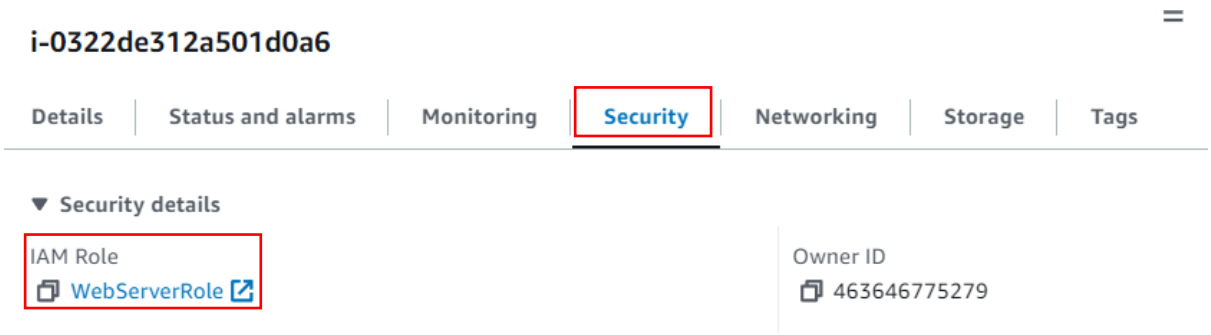
ls /opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log



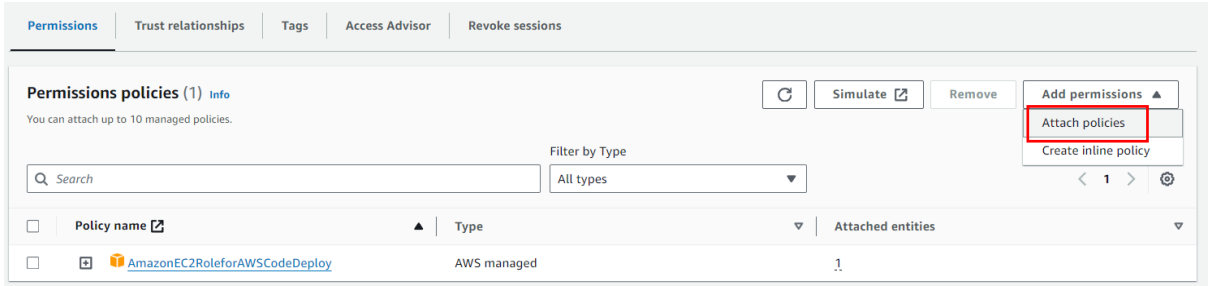
5. This is how we can view the deployment logs on our instance.

😁 Step 2: Centralizing the Logs

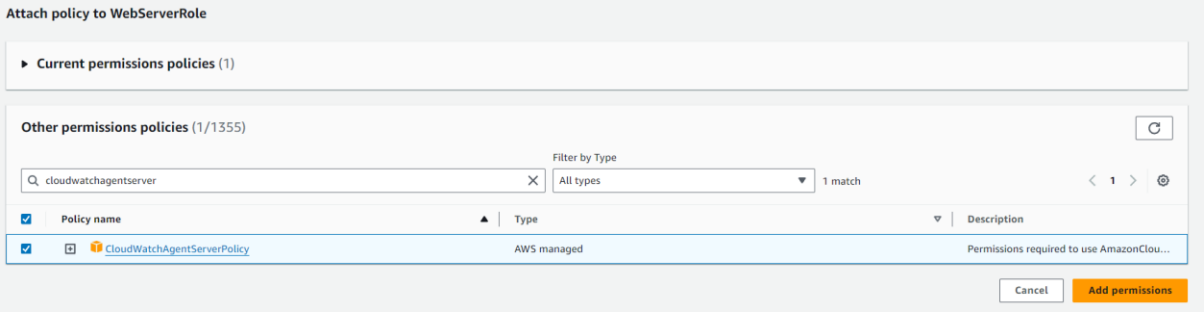
1. But what if we have tens or hundreds of instances? It will not be efficient to connect to the instances each time we need to view the deployment logs, right? So, is there any way to stream them to a centralized logging service? In this lecture, we will talk about how to stream the deployment logs from our EC2 instances to CloudWatch Logs in the simplest way possible.
2. So, CloudWatch is a centralized logging service of AWS. But to get our logs to CloudWatch we need to assign the necessary permissions to our IAM role which is attached to our EC2 instance.
3. So, come back to EC2 select your instance, and then go to security here you will see your IAM role, click on it.



4. Now here you need to click on add permission and choose attach policies.



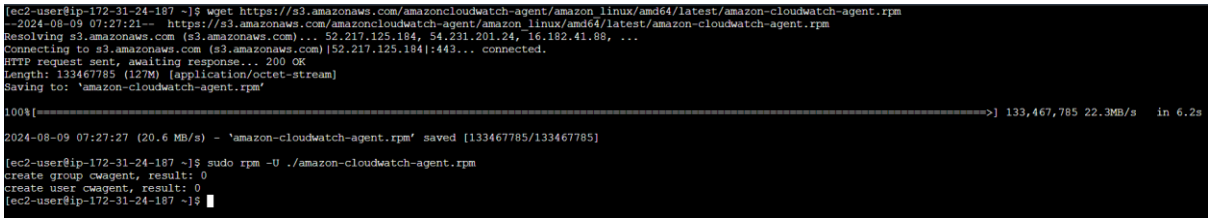
5. Now you have to search for CloudWatch Agent Server Policy, add this permission then click on save.



6. Come back to your EC2 instance connect or open a new connection if you have closed the previous one.
7. Now you are going to install the CloudWatch agent on your instance for that you can run this command given below.

   wget https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
8. Next, we'll install this package using RPM. This command installed the CloudWatch agent and created a CW agent user and group for it.

   sudo rpm -U ./amazon-cloudwatch-agent.rpm



9. Now we need to create a configuration file using the CloudWatch agent wizard, for that issue this command. Then it will ask some questions so you need to choose the same option from the snapshot.

   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard

```
[ec2-user@ip-172-31-24-187 ~]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
============================================================
= Welcome to the Amazon CloudWatch Agent Configuration Manager =
=                                                              =
= CloudWatch Agent allows you to collect metrics and logs from =
= your host and send them to CloudWatch. Additional CloudWatch =
= charges may apply.                                           =
============================================================
On which OS are you planning to use the agent?
1. linux
2. windows
3. darwin
default choice: [1]:

Trying to fetch the default region based on ec2 metadata...
I! imds retry client will retry 1 timesAre you using EC2 or On-Premises hosts?
1. EC2
2. On-Premises
default choice: [1]:

Which user are you planning to run the agent?
1. cwagent
2. root
3. others
default choice: [1]:

Do you want to turn on StatsD daemon?
1. yes
2. no
default choice: [1]:
2
```

Use the below path for the log file path.

/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log

```
Do you want to monitor metrics from CollectD? WARNING: CollectD must be installed or the Agent will fail to start
1. yes
2. no
default choice: [1]:
2
Do you want to monitor any host metrics? e.g. CPU, memory, etc.
1. yes
2. no
default choice: [1]:
2
Do you have any existing CloudWatch Log Agent (http://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AgentReference.html)
1. yes
2. no
default choice: [2]:

Do you want to monitor any log files?
1. yes
2. no
default choice: [1]:

Log file path:
/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log
Log group name:
default choice: [codedeploy-agent-deployments.log]

Log group class:
1. STANDARD
2. INFREQUENT_ACCESS
default choice: [1]:

Log stream name:
default choice: [{instance_id}]
```

```
Do you want to specify any additional log files to monitor?
1. yes
2. no
default choice: [1]:
2
Do you want the CloudWatch agent to also retrieve X-ray traces?
1. yes
2. no
default choice: [1]:
2
```

```
The config file is also located at /opt/aws/amazon-cloudwatch-agent/bin/config.json.
Edit it manually if needed.
Do you want to store the config in the SSM parameter store?
1. yes
2. no
default choice: [1]:
2
Program exits now.
```

10. So, you will choose the configuration by answering the questions, so it will create a JSON file as you can show below, and give you the path where it is saved. So, now you need to copy that path, open the file, and make some changes.

```
Existing config JSON identified and copied to:  /opt/aws/amazon-cloudwatch-agent/etc/backup-configs
Saved config file to /opt/aws/amazon-cloudwatch-agent/bin/config.json successfully.
Current config as follows:
{
        "agent": {
                "run_as_user": "cwagent"
        },
        "logs": {
                "logs_collected": {
                        "files": {
                                "collect_list": [
                                        {
                                                "file_path": "/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log",
                                                "log_group_class": "STANDARD",
                                                "log_group_name": "codedeploy-agent-deployments.log",
                                                "log_stream_name": "{instance_id}",
                                                "retention_in_days": -1
                                        },
                                        {
                                                "file_path": "/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log",
                                                "log_group_class": "STANDARD",
                                                "log_group_name": "codedeploy-agent-deployments.log",
                                                "log_stream_name": "{instance_id}",
                                                "retention_in_days": -1
                                        }
                                ]
                        }
                }
        }
}
Please check the above content of the config.
The config file is also located at /opt/aws/amazon-cloudwatch-agent/bin/config.json.
Edit it manually if needed.
```

11. In that file you just need to add the time stamp as shown below.

    "timestamp_format": "[%Y-%m-%d %H:%M:%S.%f]"

```
"retention_in_days": -1
"timestamp_format": "[%Y-%m-%d %H:%M:%S.%f]"
```

```
{
    "agent": {
        "run_as_user": "cwagent"
    },
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log",
                        "log_group_class": "STANDARD",
                        "log_group_name": "codedeploy-agent-deployments.log",
                        "log_stream_name": "{instance_id}",
                        "retention_in_days": -1,
                        "timestamp_format": "[%Y-%m-%d %H:%M:%S.%f]"
                    }
                ]
            }
        }
    }
}
```

12. Now we will use a command to start our CloudWatch agent.

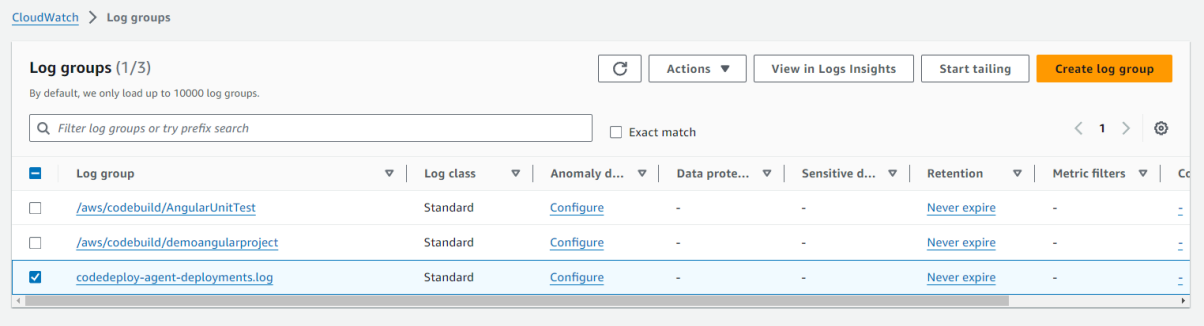
    sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s

13. Then use this command to check the status whether you CloudWatch agent is running or not.

    sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status

```
[ec2-user@ip-172-31-24-187 ~]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status
{
  "status": "running",
  "starttime": "2024-08-09T07:53:14+0000",
  "configstatus": "configured",
  "version": "1.300042.0b733"
}
```

14. Now go to CloudWatch then to log groups here you will see your log group.



15. Then open your log events and here you will see that same logs which you had seen in your instance deployment logs.