EC2 Slowness Troubleshooting Runbook

This runbook will help you troubleshoot and identify potential causes of EC2 instance slowness using AWS CLI commands and the AWS Management Console. We'll cover steps to check performance metrics, resource utilization, and other configurations that might be affecting the instance's performance.

Step 1: Identify the Affected EC2 Instance

- If the instance ID is already known, proceed to Step 2. Otherwise, use the AWS Management Console to identify the instance exhibiting slowness.

Step 2: AWS CLI Command to Get EC2 Instance Information

- Run the following command to get detailed information about the EC2 instance:

bash

Copy code

```
aws ec2 describe-instances --instance-ids YOUR_INSTANCE_ID
```

Step 3: Check CPU Utilization

- Review the CPU utilization of the instance over time using CloudWatch metrics. AWS Management Console:
  - Go to the CloudWatch service.
  - Under "Metrics," select "EC2" and then "Per-Instance Metrics."
  - Choose "CPUUtilization" and select the affected instance to view the graph.
- AWS CLI Command:
  - To get the average CPU utilization for the instance, run:

bash

Copy code

```
aws cloudwatch get-metric-statistics --namespace "AWS/EC2" --metric-name
"CPUUtilization" --dimensions "Name=InstanceId,Value=YOUR_INSTANCE_ID"
```

Step 4: Check Memory and Disk Usage

- Analyze the memory and disk usage of the instance, as they can impact performance.
  AWS Management Console:
  - In the EC2 dashboard, select the instance.
  - Go to the "Monitoring" tab and check "Status Checks" and "Instance status" for any issues.
- AWS CLI Command:
  - To get memory and disk usage metrics, use the following commands:

bash

Copy code

```
# Memory usage
aws cloudwatch get-metric-statistics --namespace "System/Linux"
--metric-name "MemoryUtilization" --dimensions
"Name=InstanceId,Value=YOUR_INSTANCE_ID" --start-time YOUR_START_TIME
--end-time YOUR_END_TIME --period 300 --statistics Average

# Disk usage
aws cloudwatch get-metric-statistics --namespace "System/Linux"
--metric-name "DiskSpaceUtilization" --dimensions
"Name=InstanceId,Value=YOUR_INSTANCE_ID" --start-time YOUR_START_TIME
--end-time YOUR_END_TIME --period 300 --statistics Average
```

Step 5: Review Network Metrics

- Check the network metrics to identify any potential networking issues affecting the instance.
  AWS Management Console:

- In the EC2 dashboard, select the instance.
- Go to the "Monitoring" tab and check the "Network In" and "Network Out" metrics.

- AWS CLI Command:
  - To get network metrics, run:

bash

Copy code

```
# Network In
aws cloudwatch get-metric-statistics --namespace "AWS/EC2" --metric-name
"NetworkPacketsIn" --dimensions "Name=InstanceId,Value=YOUR_INSTANCE_ID"
--start-time YOUR_START_TIME --end-time YOUR_END_TIME --period 300
--statistics Sum

# Network Out
aws cloudwatch get-metric-statistics --namespace "AWS/EC2" --metric-name
"NetworkPacketsOut" --dimensions "Name=InstanceId,Value=YOUR_INSTANCE_ID"
--start-time YOUR_START_TIME --end-time YOUR_END_TIME --period 300
--statistics Sum
```

Step 6: Check Storage I/O Performance

- Review storage I/O performance to see if any disk-related issues are impacting performance.
  AWS Management Console:
  - In the EC2 dashboard, select the instance.
  - Go to the "Monitoring" tab and check "EBS Read Ops" and "EBS Write Ops."
- AWS CLI Command:
  - To get storage I/O metrics, run:

bash

Copy code

```
# EBS Read Ops
aws cloudwatch get-metric-statistics --namespace "AWS/EC2" --metric-name
"EBSReadOps" --dimensions "Name=InstanceId,Value=YOUR_INSTANCE_ID"
--start-time YOUR_START_TIME --end-time YOUR_END_TIME --period 300
--statistics Sum

# EBS Write Ops
aws cloudwatch get-metric-statistics --namespace "AWS/EC2" --metric-name
"EBSWriteOps" --dimensions "Name=InstanceId,Value=YOUR_INSTANCE_ID"
--start-time YOUR_START_TIME --end-time YOUR_END_TIME --period 300
--statistics Sum
```

Step 7: Review Application and System Logs

- Access the EC2 instance via SSH or RDP.
- Check application and system logs for any errors or warnings that could be causing slowness.

Step 8: Check Instance Type

- Verify that the instance type is appropriate for the workload and resource requirements.
  AWS Management Console:
  - In the EC2 dashboard, select the instance, and the instance type will be displayed.
- AWS CLI Command:
  - To get the instance type, run:

bash

Copy code

```
aws ec2 describe-instances --instance-ids YOUR_INSTANCE_ID --query
"Reservations[*].Instances[*].InstanceType" --output text
```

Step 9: Check CPU Credits (T2/T3 Instances)

- If using T2/T3 instances, check CPU credit balance to see if it's sufficient for the instance's workload.
  AWS Management Console:
    - In the EC2 dashboard, select the instance.
    - Go to the "Monitoring" tab and check "CPUCreditBalance" and "CPUCreditUsage."
- AWS CLI Command:
    - To get CPU credit metrics, run:

bash

Copy code

```
# CPUCreditBalance
aws cloudwatch get-metric-statistics --namespace "AWS/EC2" --metric-name
"CPUCreditBalance" --dimensions "Name=InstanceId,Value=YOUR_INSTANCE_ID"
--start-time YOUR_START_TIME --end-time YOUR_END_TIME --period 300
--statistics Average

# CPUCreditUsage
aws cloudwatch get-metric-statistics --namespace "AWS/EC2" --metric-name
"CPUCreditUsage" --dimensions "Name=InstanceId,Value=YOUR_INSTANCE_ID"
--start-time YOUR_START_TIME --end-time YOUR_END_TIME --period 300
--statistics Average
```

Step 10: Review Security Group Rules

- Verify that security group rules are configured correctly to allow necessary traffic.
  AWS Management Console:
    - In the EC2 dashboard, select the instance.
    - Go to the "Description" tab and check the security groups associated with the instance.
- AWS CLI Command:
    - To get security group information, run:

bash

Copy code

```
aws ec2 describe-instances --instance-ids YOUR_INSTANCE_ID --query
"Reservations[*].Instances[*].SecurityGroups[*].GroupName" --output text
```

Step 11: Analyze Load Balancer (If Applicable)

- If the instance is part of a load-balanced setup, check the load balancer's configuration for any issues.

Step 12: Verify External Factors

- Consider checking for any external factors (e.g., network, application dependencies) that may be affecting the instance's performance.

Step 13: Mitigation and Resolution

- Based on the findings from the above steps, implement any necessary changes to address the slowness issues.
- Monitor the instance's performance after making changes to ensure improvement.