Troubleshoot EC2 instances

AWS troubleshooting skills refer to the ability to identify and resolve problems with AWS services and infrastructure. This includes the ability to diagnose and resolve issues with AWS components such as EC2 instances, S3 buckets, and databases, as well as network and security-related issues.

Troubleshooting EC2 instances can involve a variety of issues depending on what's going wrong. Steps to follow:

1.<u>**Check Instance Status**</u>: Start by verifying the status of your EC2 instance in the AWS Management Console or using the AWS CLI. If it's stopped or terminated unexpectedly, you might need to restart it or launch a new instance.

2.<u>**Review System Logs**</u>: AWS provides system logs through CloudWatch or the EC2 Console. Check the system logs for any error messages or warnings that might indicate what's causing the issue.

3. <u>**Security**</u> : Ensure that the security group associated with your EC2 instance allows inbound traffic on the necessary ports. If your instance is unreachable, it might be due to incorrect security group settings.

4.<u>**Check network ACS**</u>: Network Access Control Lists (NACLs) can also restrict traffic to and from your instance. Make sure that the NACL associated with your subnet allows the necessary traffic.

5.<u>**Verify key pair**</u>: If you're unable to SSH into your instance (for Linux instances), double-check that you're using the correct key pair. If you've lost access to the key pair, you might need to create a new key pair and associate it with the instance.

6.<u>**Instance Type**</u>: Verify that the instance type you're using is appropriate for your workload. Insufficient resources can cause performance issues or even instance failures.

7.<u>**Instance Health Checks**</u>: Some instance types come with built-in health checks. Check if your instance passes these checks. If not, it might indicate underlying hardware issues.

8.<u>**Storage**</u>: Ensure that your instance's root volume (and any additional volumes) has enough free space. If the volume is full, it can cause services to fail or the instance to become unresponsive.

CPU and Memory Usage: Monitor CPU and memory usage using CloudWatch metrics. High CPU or memory usage could indicate that your workload requires more resources than the instance type provides.

9.**Software Configuration**: Review your application and system configuration for any misconfigurations or errors that could be causing issues.

10.Restart Services: If your application or services are unresponsive, try restarting them to see if it resolves the issue.

Check for AWS Incidents: Sometimes, issues can be caused by broader AWS service disruptions. Check the AWS Service Health Dashboard for any ongoing incidents in your region.

Instance Metadata: Utilize instance metadata to gather additional information about your instance, such as its IP address, instance type, and IAM role.

Cloud watch

CloudWatch is a powerful monitoring and logging service provided by AWS that can be instrumental in troubleshooting issues with your EC2 instances. Here's how you can leverage CloudWatch for troubleshooting:

Monitoring Metrics: Start by monitoring relevant metrics for your EC2 instances, such as CPU utilization, memory usage, disk I/O, and network traffic. Use CloudWatch dashboards to visualize these metrics over time. Spikes or unusual patterns in these metrics can indicate performance issues.

1.**Set Alarms**: Create CloudWatch alarms to notify you when certain metrics cross predefined thresholds. For example, you can set an alarm to notify you when CPU utilization exceeds 80% for a sustained period. This proactive monitoring can help you catch issues before they impact your application.

2.Log Monitoring: CloudWatch Logs allows you to centralize and monitor logs from your EC2 instances. Configure your instances to send logs to CloudWatch Logs, and then set up metric filters and alarms based on log events. This can help you identify errors, warnings, or other patterns in your application logs.

3**.Custom Metrics**: In addition to standard EC2 metrics, you can also publish custom metrics to CloudWatch. This allows you to track application-specific metrics that are relevant to your use case. For example, you could track the number of requests processed by your application or the latency of API calls.

4.**Insights**: CloudWatch Logs Insights provides an interactive query language that you can use to analyze log data in real-time. Use Insights to search and filter logs, identify patterns, and troubleshoot issues more efficiently.

5.**Cross-Account Monitoring**: If you have multiple AWS accounts, you can use CloudWatch Cross-Account Monitoring to aggregate metrics and logs from all your accounts into a single dashboard for centralized monitoring.

6.**Anomaly Detection**: CloudWatch Anomaly Detection analyzes historical data to automatically detect anomalous behavior in your metrics. This can help you identify issues that might not be apparent from simple threshold-based alarms.

By leveraging these features of CloudWatch, you can gain deeper insights into the behavior of your EC2 instances and quickly identify and troubleshoot issues as they arise.

Troubleshooting using ELB

Using the Elastic Load Balancer (ELB) service in AWS can help troubleshoot issues with your EC2 instances by distributing incoming traffic across multiple instances and providing features for health checks and monitoring. Here's how you can leverage ELB for troubleshooting:

1.**Health Checks**: ELB regularly performs health checks on registered instances to ensure they are healthy and able to handle traffic. If an instance fails a health check, ELB stops sending traffic to that instance and redirects traffic to healthy instances. Monitor the health status of your instances in the ELB console or through the AWS CLI.

2.**Access Logs**: Enable access logs for your ELB to capture detailed information about incoming requests, including the client IP address, request path, response codes, and more. Analyze these logs to identify patterns or errors that might indicate issues with your application or configuration.

3.**Monitoring Metrics**: ELB provides various CloudWatch metrics, such as request count, latency, and error rates, which you can use to monitor the performance and health of your load balancer and instances. Set up CloudWatch alarms to notify you when certain metrics exceed predefined thresholds, indicating potential issues.

4.**Cross-Zone Load Balancing**: By default, ELB distributes traffic evenly across all registered instances in all availability zones enabled for the load balancer. If you're experiencing uneven traffic distribution or performance issues, consider enabling or disabling cross-zone load balancing to see if it improves the situation.

Troubleshooting using cloud trail

CloudTrail is a service provided by AWS that records API calls and events for your AWS account. It can be a valuable tool for troubleshooting issues related to changes in your AWS environment, including changes to EC2 instances and other resources. Here's how you can use CloudTrail for troubleshooting:

1.**Event Logging**: Enable CloudTrail in your AWS account to start logging API calls and events. CloudTrail captures information such as the identity of the caller, the time of the API call, the request parameters, and the response elements returned by the AWS service.

2.**Event History**: Use the CloudTrail console or API to view a history of API calls and events in your AWS account. You can search and filter the event history based on various criteria, such as time range, event type, resource type, and more.

3.**Resource Changes**: Monitor CloudTrail logs for changes to your EC2 instances, including instance launches, terminations, modifications, and changes to security groups or IAM roles. If an instance is unexpectedly terminated or modified, CloudTrail can help you identify who made the change and when it occurred.

4.**Access Control Changes**: Track changes to IAM policies, roles, and permissions using CloudTrail. If access permissions are modified in a way that affects your EC2 instances, CloudTrail can help you trace the changes back to the responsible user or entity.

5**.Integration with CloudWatch Events**: Set up CloudWatch Events to trigger automated responses based on events captured by CloudTrail. For example, you can create CloudWatch Event rules to notify you when specific API calls are made or when certain resource changes occur.

6.Compliance and Auditing: Use CloudTrail logs for compliance auditing and regulatory purposes. CloudTrail provides a record of all API activity in your AWS account, which can be useful for demonstrating compliance with security standards and regulations

7.Log File Integrity: CloudTrail logs are encrypted and stored in an S3 bucket. Enable log file integrity validation to ensure that log files have not been tampered with or modified after they were delivered to the S3 bucket.

8.Multi-Region Logging: Enable multi-region logging to capture API activity from all AWS regions where your resources are deployed. This ensures that you have a comprehensive view of API calls and events across your entire AWS infrastructure

Troubleshooting using cloud front

CloudFront is a content delivery network (CDN) service provided by AWS that delivers content, including web pages, videos, images, and other assets, with low latency and high transfer speeds. While CloudFront primarily focuses on content delivery, it offers features and capabilities that can aid in troubleshooting various issues. Here's how you can use CloudFront for troubleshooting:

1.Monitoring Metrics: CloudFront provides several metrics related to request and data transfer behavior. Monitor metrics such as request count, data transfer, cache hit ratio, and HTTP status codes to identify any anomalies or issues with content delivery.

2.CloudWatch Alarms: Set up CloudWatch alarms based on CloudFront metrics to receive notifications when specific thresholds are exceeded. For example, you can create alarms for high error rates or a sudden drop in cache hit ratio.

3.Access Logs: Enable CloudFront access logging to capture detailed information about requests and responses served by CloudFront. Access logs include data such as client IP addresses, request paths, response status codes, and more. Analyze access logs to identify patterns, troubleshoot errors, and optimize content delivery.

4.Real-time Logs: CloudFront provides real-time logs for streaming distributions, allowing you to monitor viewer activity and troubleshoot streaming issues in real-time. Real-time logs include data such as viewer IP addresses, playback status, and buffer time.

5.Invalidations: If you suspect that stale or outdated content is being served by CloudFront, use invalidations to purge content from the cache. Create invalidation requests to remove specific files or directories from the CloudFront cache, ensuring that fresh content is fetched from the origin server.

6.Origin Server Monitoring: Monitor the health and performance of your origin server(s) using CloudFront metrics and CloudWatch alarms. High latency or error rates from the origin server can impact content delivery performance and reliability.

7.Origin Response Headers: Configure CloudFront to include additional response headers from the origin server in access logs. This allows you to capture custom headers that may contain important diagnostic information or debugging details.1.

SSL/TLS Configuration: If you're using HTTPS with CloudFront, ensure that SSL/TLS certificates are properly configured and up-to-date. Monitor SSL/TLS handshake failures and certificate expiration using CloudFront metrics and CloudWatch alarms.