

5.2. Student Handout

EC2 Monitoring and Management with Amazon CloudWatch: Student Handout

Introduction

This handout provides a concise overview of monitoring and managing EC2 instances using Amazon CloudWatch. It covers key concepts, including setting up custom metrics, alarms, scaling instances, securing them, and creating backups using AMIs.

1. Monitoring EC2 Instances with CloudWatch

CloudWatch allows you to monitor various metrics of your EC2 instances, such as:

- **CPU Utilization:** Measures the percentage of CPU resources being used.
 - Example: Monitor CPU utilization to ensure it stays below 70% for optimal performance.
 - Example: Set an alarm if CPU utilization exceeds 90% to prevent overloading.
 - Example: Track CPU usage trends to plan for future scaling needs.
- **Memory Utilization:** Measures the amount of RAM being used.
 - Example: Monitor memory usage to avoid application crashes due to insufficient RAM.
 - Example: Set an alarm if memory usage exceeds 80% to take corrective action.
 - Example: Analyze memory usage patterns to optimize application performance.
- **Network Traffic:** Measures the amount of data being sent and received.
 - Example: Monitor network traffic to identify potential bottlenecks.
 - Example: Set an alarm if network traffic exceeds expected levels, indicating a potential DDoS attack.
 - Example: Track data transfer rates to optimize bandwidth usage.
- **Disk I/O:** Measures the amount of data being read from or written to the disk.
 - Example: Monitor disk I/O to ensure efficient data processing.
 - Example: Set an alarm if disk I/O exceeds normal levels, indicating potential hardware issues.
 - Example: Analyze disk I/O patterns to optimize storage performance.

2. Setting Up Custom CloudWatch Metrics and Alarms

Custom metrics allow you to monitor specific aspects of your EC2 instances not covered by default metrics.

- **Example 1:** Monitor application-specific metrics, such as request latency or error rates.
 - **Example 2:** Set up a custom metric to track database query performance.
 - **Example 3:** Create an alarm for custom metrics, such as memory usage exceeding 75%.
-

3. Monitoring CPU, Memory, Network Traffic, and Storage Utilization

Key metrics to monitor include:

- **CPU Utilization:** Ensure it remains within acceptable limits to prevent performance degradation.
 - Example: Monitor CPU spikes during peak usage times.
 - Example: Set alarms for sustained high CPU usage.
 - Example: Analyze CPU trends for capacity planning.
 - **Memory Utilization:** Prevent application crashes by monitoring RAM usage.
 - Example: Monitor memory leaks in applications.
 - Example: Set alarms for high memory usage.
 - Example: Track memory usage patterns for optimization.
 - **Network Traffic:** Identify potential bottlenecks and optimize bandwidth usage.
 - Example: Monitor traffic spikes during promotional events.
 - Example: Set alarms for unexpected traffic surges.
 - Example: Analyze traffic patterns for load balancing.
 - **Storage Utilization:** Ensure sufficient disk space for application data.
 - Example: Monitor disk space usage to prevent data loss.
 - Example: Set alarms for low disk space.
 - Example: Track storage trends for capacity planning.
-

4. Scaling EC2 Instances

Scaling ensures your instances can handle varying loads.

- **Vertical Scaling:** Increase the size of your instance.
 - Example: Upgrade from a t2.micro to a t2.large instance for more resources.
 - Example: Increase instance size during peak traffic periods.
 - Example: Scale up to handle increased application demand.
 - **Horizontal Scaling:** Add more instances to distribute the load.
 - Example: Add additional instances during high traffic events.
 - Example: Use multiple instances to improve application availability.
 - Example: Scale out to handle increased user demand.
-

5. Auto Scaling Groups and Policies

Auto Scaling automatically adjusts the number of instances based on demand.

- **Example 1:** Add instances when CPU utilization exceeds 80%.
 - **Example 2:** Remove instances when CPU utilization drops below 30%.
 - **Example 3:** Use Auto Scaling to maintain a minimum number of instances for redundancy.
-

6. Setting Up Load Balancers for High Availability

Load balancers distribute incoming traffic across multiple instances.

- **Example 1:** Use a load balancer to distribute traffic during peak hours.
 - **Example 2:** Ensure high availability by routing traffic to healthy instances.
 - **Example 3:** Improve application performance by balancing the load across instances.
-

7. Securing EC2 Instances

Security is crucial for protecting your instances.

- **Patching:** Regularly update instances with security patches.
- Example: Apply security patches to prevent vulnerabilities.
- Example: Schedule regular patch updates for all instances.
- Example: Use automated patch management tools.

- **Firewalls (Security Groups):** Control inbound and outbound traffic.
 - Example: Restrict access to specific IP addresses.
 - Example: Allow only necessary ports for application access.
 - Example: Use security groups to isolate instances.
 - **IAM Roles:** Control access to instances.
 - Example: Assign IAM roles to limit access to sensitive data.
 - Example: Use IAM roles for secure API access.
 - Example: Implement least privilege access for users.
-

8. Creating and Managing AMIs for Instance Backups

AMIs provide a backup of your instance configuration.

- **Example 1:** Create an AMI before making significant changes to an instance.
 - **Example 2:** Use AMIs to quickly launch new instances in case of failure.
 - **Example 3:** Maintain a library of AMIs for different application versions.
-

Hands-On Steps

Step 1: Setting Up CloudWatch Alarms

1. Go to the CloudWatch dashboard.
2. Select the metric to monitor (e.g., CPU utilization).
3. Set a threshold (e.g., 80% CPU utilization).
4. Create an alarm to notify when the threshold is crossed.

Step 2: Setting Up Auto Scaling

1. Go to the EC2 dashboard.
2. Create an Auto Scaling group.
3. Define scaling policies (e.g., add an instance when CPU utilization is above 80%).

Step 3: Creating an AMI

1. Go to the EC2 dashboard.
2. Select the instance to create an AMI from.

3. Click on "Create Image" to create an AMI.
-

Conclusion

Monitoring and managing EC2 instances with CloudWatch is essential for ensuring optimal performance and availability. By setting up custom metrics, alarms, and scaling policies, you can automate much of the management process and focus on building your applications.

Key Takeaways

- **CloudWatch:** Monitor EC2 instances by tracking metrics like CPU, memory, and network traffic.
 - **Custom Metrics and Alarms:** Set up notifications for specific performance thresholds.
 - **Auto Scaling:** Automatically adjust the number of instances based on demand.
 - **Load Balancers:** Distribute traffic for high availability.
 - **Security:** Regularly patch instances and use firewalls.
 - **AMIs:** Create backups of your instances for quick recovery.
-

Thank you for reviewing this handout! If you have any questions, feel free to ask.