**Basic Monitoring with CloudWatch:**

**I found creating a custom CloudWatch Dashboard to be fairly straight forward while following this guide made by AWS:**

**https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create_dashboard.html**

1. **Amazon EC2:**
    - CloudWatch automatically collects and stores basic metrics such as CPU utilization, disk and network usage every 5 minutes.
    - Detailed monitoring (data collected every minute) is available but must be enabled and incurs additional charges.
2. **Amazon RDS:**
    - Basic metrics like CPU utilization, database connections, and read/write throughput are automatically monitored and available in CloudWatch.
3. **Elastic Load Balancing (ELB):**
    - CloudWatch collects metrics like request counts, request latency, and error codes for your load balancers automatically.
4. **Amazon EFS:**
    - Basic metrics including read/write data and throughput are automatically collected.
5. **AWS S3:**
    - CloudWatch doesn't automatically monitor S3 bucket sizes or object counts. However, you can activate these metrics in S3 for CloudWatch to collect them.
6. **AWS Route 53:**
    - CloudWatch automatically collects some metrics, like health check status and DNS query volumes, if you have configured health checks.

Creating an effective CloudWatch dashboard involves not only selecting the right metrics but also presenting them in a format that makes the data easy to understand and act upon. Here's a guide on how to best display each of the suggested metrics for your EC2 Instances, RDS, ELB, and EFS:

**1. EC2 Instances:**

- **CPU Utilization:**
    - **Display Type:** Line chart.
    - **Rationale:** Shows trends over time, making it easy to spot spikes or drops in CPU usage.

- **Disk Read/Write Ops:**
    - **Display Type:** Line chart.
    - **Rationale:** Useful for tracking I/O trends and identifying periods of high disk activity.
- **Network In/Out:**
    - **Display Type:** Line chart.
    - **Rationale:** Line charts are effective for visualizing network traffic patterns over time.
- **Status Checks:**
    - **Display Type:** Number widget or text widget.
    - **Rationale:** Quickly shows the current status (e.g., "0" for no issues, "1" for issues detected).

**2. Amazon RDS:**

- **CPU Utilization:**
    - **Display Type:** Line chart.
    - **Rationale:** To observe CPU load trends and potential performance bottlenecks.
- **Database Connections:**
    - **Display Type:** Line chart.
    - **Rationale:** To track connection patterns and detect sudden increases or decreases.
- **ReadIOPS/WriteIOPS:**
    - **Display Type:** Two line charts (one for Read and one for Write).
    - **Rationale:** Helps in separately analyzing read and write operations over time.
- **Free Storage Space:**
    - **Display Type:** Stacked area chart or line chart.
    - **Rationale:** To monitor storage capacity trends, which is critical for database health.

**3. Elastic Load Balancer (ELB):**

- **Request Count Per Target:**
    - **Display Type:** Line chart.
    - **Rationale:** To see the distribution of requests over time across different targets.
- **HTTP 4XX/5XX Errors:**
    - **Display Type:** Two line charts or bar charts (one for 4XX and one for 5XX).

    - **Rationale:** To distinguish between client-side and server-side errors.
- **Latency:**
    - **Display Type:** Line chart.
    - **Rationale:** To monitor how quickly the ELB is processing requests.
- **Healthy Host Count:**
    - **Display Type:** Number widget.
    - **Rationale:** Provides a quick view of the number of healthy instances.

**4. Elastic File System (EFS):**

- **Read/Write IOPS:**
    - **Display Type:** Two line charts (one for Read IOPS and one for Write IOPS).
- **Rationale:** Line charts effectively show the input/output operations over time, helping in identifying periods of high activity or potential bottlenecks.
- **Total IO Bytes:**
    - **Display Type:** Line chart.
    - **Rationale:** A line chart can display the trend of data transfer which is crucial for assessing throughput and performance.
- **Burst Credit Balance:**
    - **Display Type:** Line chart.
    - **Rationale:** A line chart will show the fluctuations in burst credits over time, which is important for understanding and managing performance during high loads.

AWS CloudTrail is a crucial service for governance, compliance, operational auditing, and risk auditing within the AWS ecosystem. Understanding its key aspects and its relationship with AWS APIs will help you better utilize its capabilities. Here are the important things to know:

**Key Aspects of CloudTrail**

1. **API Call Tracking:**
    - CloudTrail primarily tracks API calls made to your AWS account. This includes calls made via the AWS Management Console, AWS CLI, AWS SDKs, and other AWS services.
2. **Audit Trails:**
    - It provides an audit trail of all user activities, including who made the API call, from what source IP, at what time, and what actions were performed.

3. **Security and Compliance:**
    - CloudTrail is essential for security analysis, resource change tracking, and ensuring compliance with internal policies and regulatory standards.
4. **Log File Storage and Integrity:**
    - Logs are delivered to an Amazon S3 bucket specified by you, and log file integrity validation can be enabled to ensure the logs have not been tampered with.
5. **Integration with Other Services:**
    - CloudTrail can be integrated with Amazon CloudWatch Logs for real-time monitoring and with AWS CloudWatch Alarms to set up alerts based on specific API activity.

**Relationship with AWS APIs**

- **API as a Core of AWS:**
    - In AWS, almost every action taken – whether through the Console, CLI, or SDK – is an API call under the hood. AWS operates on an API-driven model, meaning each service exposes a set of APIs that perform specific actions.
- **Tracking API Calls:**
    - CloudTrail tracks these API calls, providing visibility into user and resource activity in your AWS account. For example, launching an EC2 instance, modifying a security group, or changing IAM policies are all API calls that CloudTrail can record.

**What CloudTrail Tracks**

1. **Management Events:**
    - These are control plane operations that manage AWS resources. Examples include creating an S3 bucket, launching an EC2 instance, or adding a user to an IAM group.
2. **Data Events:**
    - These are data plane operations that perform management of data within a resource. For example, putting an object into an S3 bucket or sending a message to an SNS topic.
3. **Insights Events:**
    - These are events based on unusual patterns of API activity, which can be automatically analyzed by enabling CloudTrail Insights.

**Use Cases**

- **Security Analysis:** Detecting suspicious activity, like unauthorized API calls or unusual spikes in activity.
- **Operational Problem Solving:** Troubleshooting operational issues by understanding the sequence of API calls leading up to the problem.

- **Compliance Auditing:** Providing historical data for internal audits or regulatory compliance.

**Best Practices**

- **Enable CloudTrail in All Regions:** This ensures that API activity is captured across all geographic regions.
- **Secure Your S3 Buckets:** Ensure the S3 buckets storing your CloudTrail logs are properly secured and access is controlled.
- **Regularly Review Logs:** Regular analysis of logs can help in early detection of security threats and operational issues.