|  |  |
| --- | --- |
| **Project Title** | **Prometheus Monitoring and Alerting for Multiple EC2 Instances in Multiple Accounts with Slack Integration** |
| **Technologies** | **EC2, Prometheus, Alert, Alert Manager, Slack** |

**Approach:**

To create a Prometheus monitoring system for multiple Amazon Elastic Compute Cloud (EC2) instances in multiple accounts and send alerts to a Slack community when any of the servers are down, you will need to perform the following steps:

1. Set up a Prometheus server on one of the EC2 instances. This can be done by following the instructions in the Prometheus documentation.
2. Set up one or more Prometheus exporters on each of the other EC2 instances that you want to monitor. An exporter is a piece of software that runs on a host and exposes metrics in a format that Prometheus can scrape. There are many exporters available for different types of systems, such as the Node Exporter for Linux servers and the MySQL Exporter for databases.
3. Configure the Prometheus server to scrape metrics from the exporters. This can be done by modifying the configuration file for the server and adding a job configuration for each exporter.
4. Set up an alerting rule in Prometheus to send a notification to a Slack channel when a server goes down. This can be done by modifying the configuration file for the server and adding an alerting rule that specifies the condition under which the alert should be triggered and the Slack webhook to which the alert should be sent.
5. Test the system to ensure that it is working as expected. This can be done by simulating a server failure and verifying that the alert is sent to the Slack channel.

|  |  |
| --- | --- |
| **Project Title** | **Designing an Automatic Data Collection and Storage System with AWS Lambda and Slack Integration for Server Availability Monitoring and Slack Notification** |
| **Technologies** | **AWS Lambda, Amazon RDS, CloudWatch, Slack API** |

**Approach:**

1. Create an AWS Lambda function and configure it to be triggered by an Amazon CloudWatch Event that occurs every 15 seconds.
2. In the function's code, use the requests library to make a GET request to the API to fetch the data.
3. Use a library such as psycopg2 to connect to the Amazon RDS instance and store the data in the database.
4. Use Amazon CloudWatch to set up a monitoring alarm that will trigger when the server is unavailable.
5. Use the Slack API to send a message to your Slack community when the alarm is triggered.
6. Test the function to ensure that it is able to fetch and store the data correctly, and that the monitoring and alerting functionality is working as expected.
7. Deploy the function to run indefinitely, continuing to fetch and store the data on a regular basis.