



Slack Integration with AWS Chatbot

01

Overview of ChatOps and ChatSecOps concepts

ChatOps, short for Chat Operations, involves using chatbots, tools, and communication platforms to manage and execute operational tasks. By utilizing existing Slack channels and Amazon Chime chatrooms, you can receive real-time alerts and notifications about operational issues, and respond to them directly within the same chat environment. This approach streamlines communication and accelerates response times.

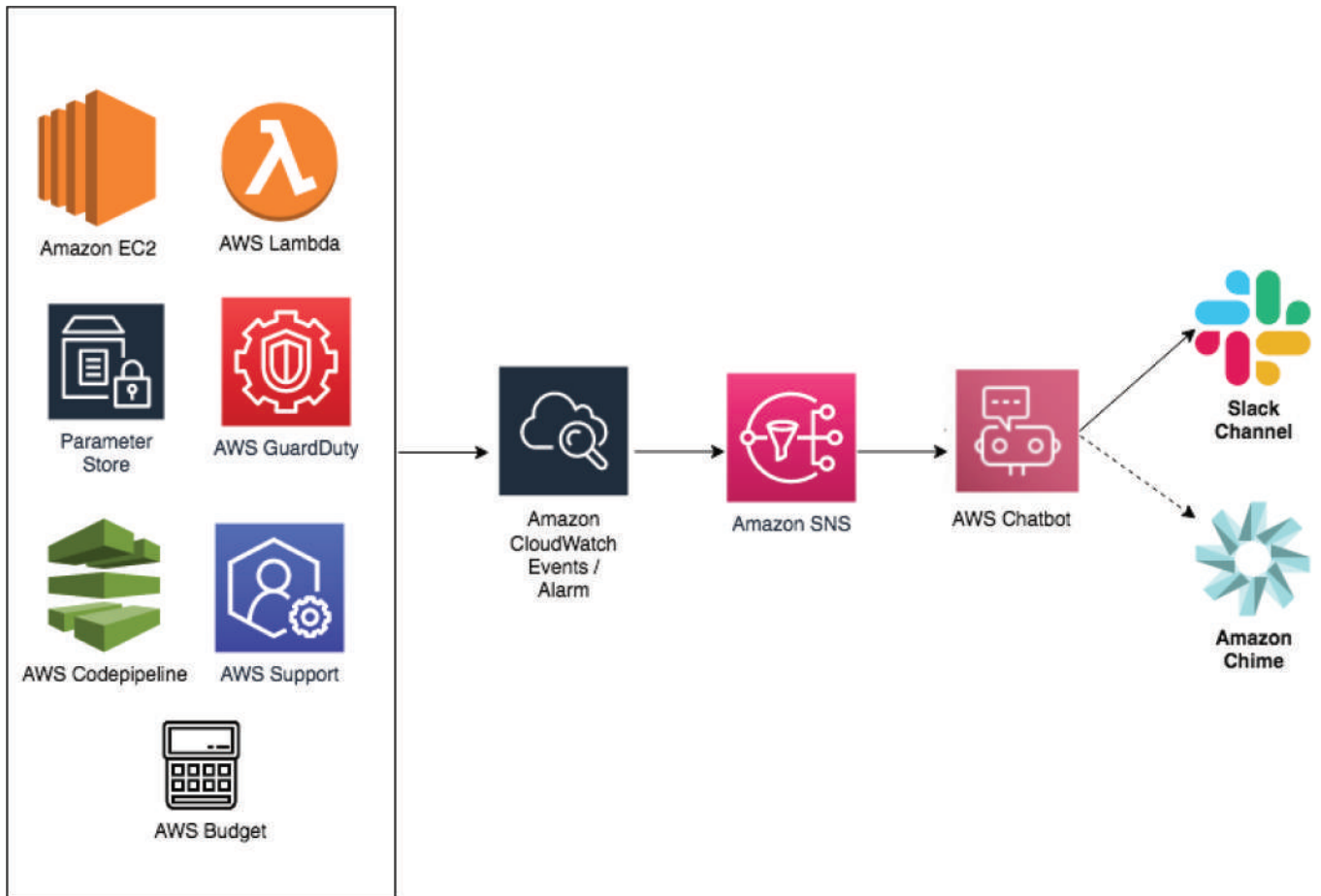
SecOps is a philosophy that fosters collaboration between IT Operations (ITOps) and Security teams to enhance an organization's security posture. ChatSecOps, or Chat Security Operations, extends the ChatOps model to support SecOps by integrating security operations into the collaborative chat environment.

ChatSecOps enhances this collaboration by delivering security-related notifications to shared chat rooms used by SecOps teams, ensuring that all team members have visibility into ongoing issues and the actions being taken to address them. In these channels, SecOps teams can share threat analysis reports, compliance findings, and details on security vulnerabilities, working closely with DevOps teams to conduct further analysis, investigation, and remediation. This integrated approach promotes the DevSecOps philosophy by ensuring that security considerations are embedded into the development and operations processes, fostering seamless collaboration between SecOps and DevOps teams.

AWS Chatbot

AWS Chatbot is an interactive tool that simplifies the monitoring and management of your AWS resources directly within Slack channels and Amazon Chime chat rooms. With AWS Chatbot, you can receive real-time alerts, execute commands to retrieve diagnostic information, trigger AWS Lambda functions, and open AWS Support cases. There are multiple ways to integrate AWS Chatbot with other AWS services. In this blog, I've highlighted seven common use cases that are relevant across all customer domains. These use cases demonstrate how to consolidate notifications from various areas such as security, performance monitoring, CI/CD workflows, and compliance to proactively detect and prevent potential issues.

AWS Chatbot is an interactive tool that simplifies the monitoring and management of your AWS resources directly within Slack channels and Amazon Chime chat rooms. With AWS Chatbot, you can receive real-time alerts, execute commands to retrieve diagnostic information, trigger AWS Lambda functions, and open AWS Support cases. There are multiple ways to integrate AWS Chatbot with other AWS services. In this blog, I've highlighted seven common use cases that are relevant across all customer domains. These use cases demonstrate how to consolidate notifications from various areas such as security, performance monitoring, CI/CD workflows, and compliance to proactively detect and prevent potential issues.



Prerequisites

To get started, you'll need the following prerequisites:

- An active AWS account
- A Slack account
- Slack workspace ID and channel ID

Note: You must have administrative permissions for your Slack workspace or have the ability to work with workspace owners to get approval for installing AWS Chatbot.

Set up Slack permissions

To manage user permissions in Slack channels integrated with AWS Chatbot, you can choose one of the following approaches:

1. Associate a Channel IAM Role with AWS Chatbot:

This method grants the same permissions to all members of the Slack channel by linking an IAM role to the channel. It's ideal when all channel members need uniform access rights. Additionally, the channel IAM role can be used to limit the permissions granted by individual user IAM roles, ensuring that everyone operates within the defined boundaries.

2. Define User Roles:

User roles allow each channel member to select their own IAM role, enabling different users to have varying levels of permissions. This approach is particularly useful when you want to customize permissions for individual users or when you prefer that new channel members do not automatically have the ability to perform certain actions upon joining.

Once you've set up the Slack channel with the necessary permissions, you can integrate the ChatOps for AWS app with your channel by following these steps:

How to Integrate Slack Channel with AWS Chatbot

1. Log in to Slack:

Access your Slack account using either the Slack app or a web browser.

2. Select Your Channel:

In the Slack sidebar, under the Channels section, select the channel where you want to integrate AWS Chatbot.

3. Open Channel Configuration:

In the right pane, click on the channel name to open the channel's configuration window.

4. Add the AWS Chatbot App:

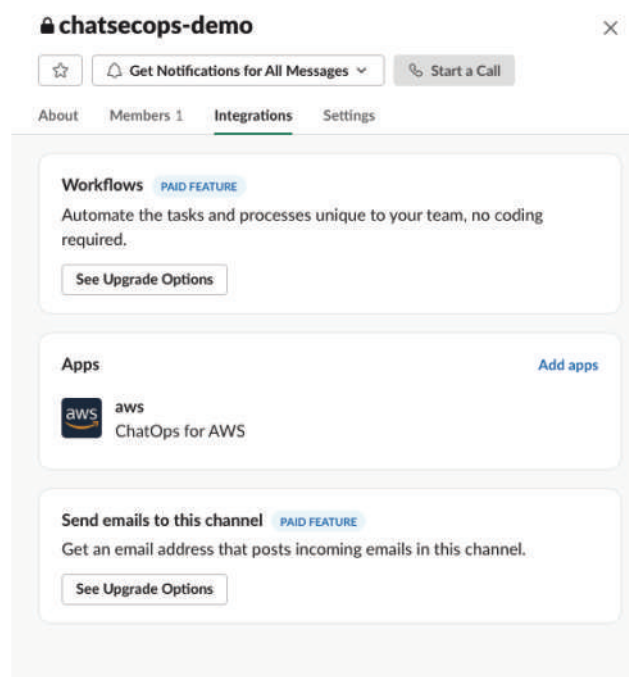
Navigate to the Integrations tab and select "Add an App."

5. Search for AWS Chatbot:

In the search bar, type "AWS Chatbot" and then click the "Add" button next to AWS Chatbot in the search results.

6. Verify Integration:

After adding, go back to the Integrations tab. Under the Apps section, you should see "ChatOps for AWS" listed, confirming that the integration is complete.



Tutorial

In this article, let's discuss the following steps:

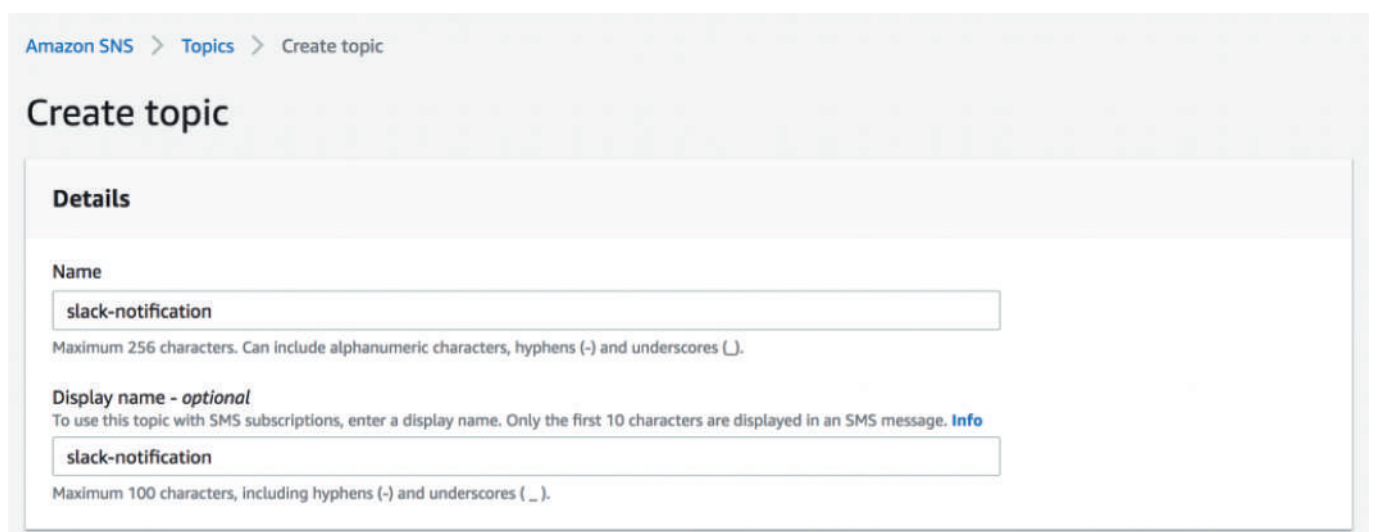
Initial configurations:

1. Create Amazon SNS topic
2. Configure AWS Chatbot on Slack
3. Create Amazon SNS topic

To use AWS Chatbot, you must have an Amazon SNS topic setup. Follow the steps to create an Amazon SNS topic.

Navigate to Amazon SNS console

In the Create topic section, enter a topic name, for example slack notification.



Amazon SNS > Topics > Create topic

Create topic

Details

Name

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - optional

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. [Info](#)

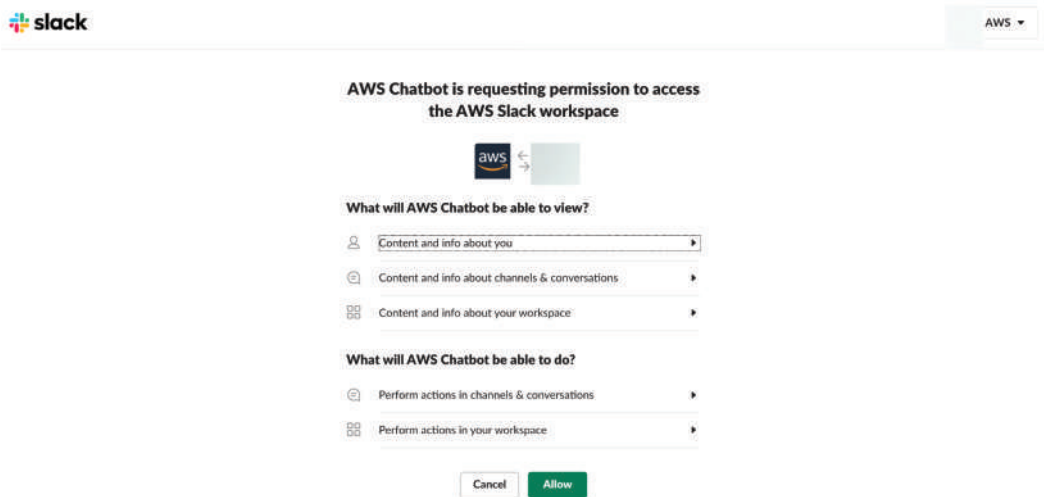
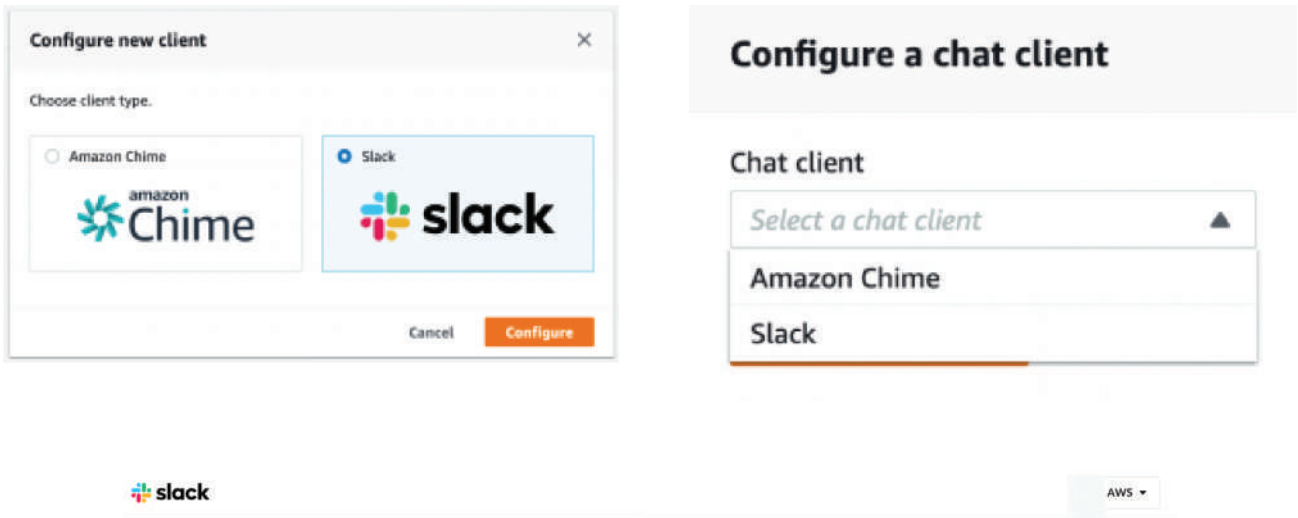
Maximum 100 characters, including hyphens (-) and underscores (_).

Configure AWS Chatbot on Slack

A Slack channel is a single place for a team to share messages, tools, and files. In Slack, teamwork and communication happen in channels. Let's discuss how to configure AWS Chatbot on Slack.

On the AWS web console, search for the service AWS, and select Slack as chat client from the dropdown list.

Select “allow” on the next screen.



Under Configuration details, enter a name for your configuration. The name must be unique across your account and can't be edited later. For the Slack channel, choose the channel that you want to use. To use private Slack channel with AWS Chatbot, choose Private channel.

In Slack, copy the Channel ID of the private channel by right-clicking on the channel name and selecting Copy Link.

On the AWS Management Console, in the AWS Chatbot window, paste the ID into the Channel URL.

Configure Slack channel

Configuration details

Configuration name
Name your configuration to identify it easily later. This name can't be changed after you create the configuration.

slack-channel-configuration

The name can have up to 128 characters. Valid characters: a-z, A-Z, 0-9, and - _

Logging - optional
You can turn on logging for your configuration. There is an additional charge for using Amazon CloudWatch Logs. [Learn more](#)

Publish logs to Amazon CloudWatch Logs

Slack channel

Channel type
Choose public channels from the list. To choose a private channel, enter the channel ID.

Public
Anyone in your workspace can view and join public channels.

Private
You can join or view private channels only by invitation.

Private channel ID
Find the channel ID in Slack by right clicking on the channel in the channel list and copying the link. The channel ID is the string at the end of the URL.

C01

The channel ID can be an alphanumeric string (C1111122233) or the entire channel URL (https://workspace.slack.com/messages/C1111122233).

Define the IAM permissions that the AWS Chatbot uses for messaging your Slack chat room

For Policy templates, choose Notification permissions. This is the IAM policy template for AWS Chatbot. It provides the necessary read and list permissions for CloudWatch alarms, events and logs, and for Amazon SNS topics.

Permissions

AWS Chatbot requires an IAM role to access CloudWatch metrics, run commands, and respond to notification actions. All users in the Slack channel will have the permissions defined by the role.

IAM role
Defines the permissions for AWS Chatbot.

Create an IAM role using a template

Role name

AWSChatBot-role

The name can have up to 64 characters. Valid characters: a-z, A-Z, 0-9, and + = , . @ - _

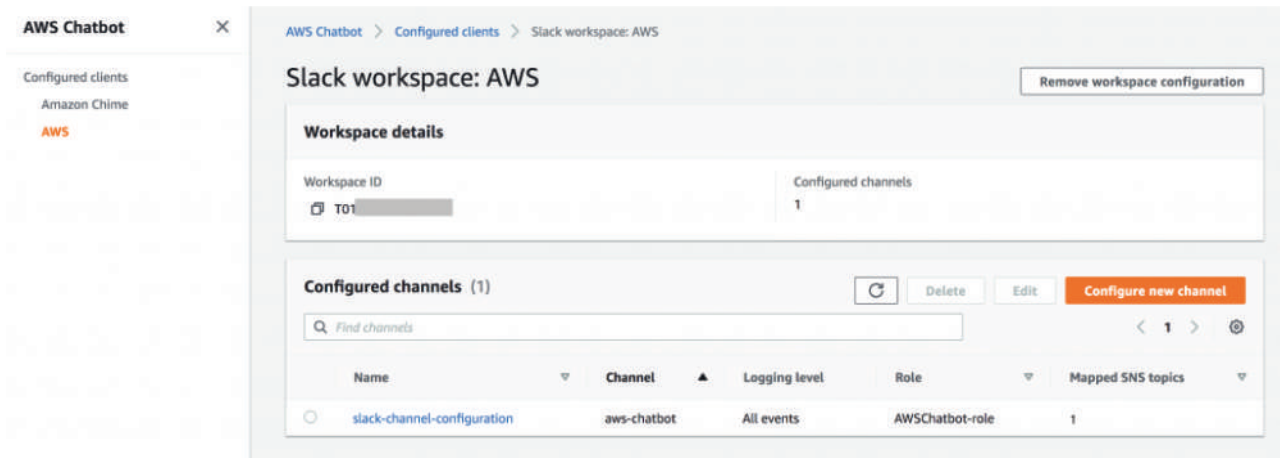
Policy templates
Choose one or more policy templates. AWS Chatbot will generate a role for you. For information about the permissions that each policy template adds to your role, see the AWS Chatbot User Guide. [Learn more](#)

Select template(s)

- Notification permissions
Allows AWS Chatbot to retrieve metric graphs from Amazon CloudWatch.
- Read-only command permissions
Allows read-only commands in supported clients.
- Lambda-invoke command permissions
Allows Lambda-invoke commands in supported clients.
- AWS Support command permissions
Allows calling AWS Support APIs in supported clients.

Choose the Amazon SNS topic you previously created that sends notifications to the Slack channel.

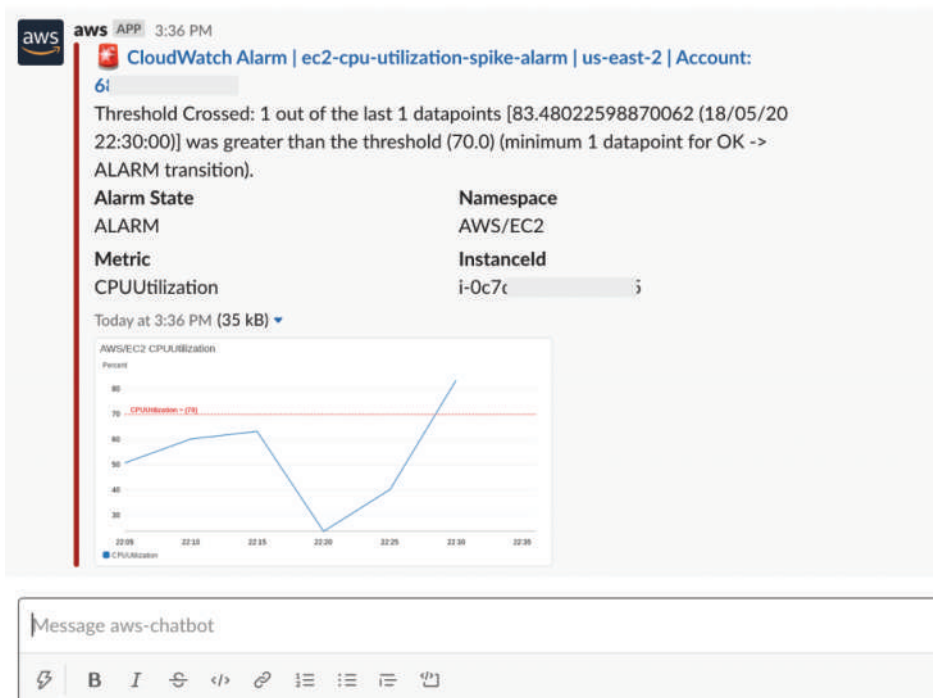
You should see following screen once you configure the channel



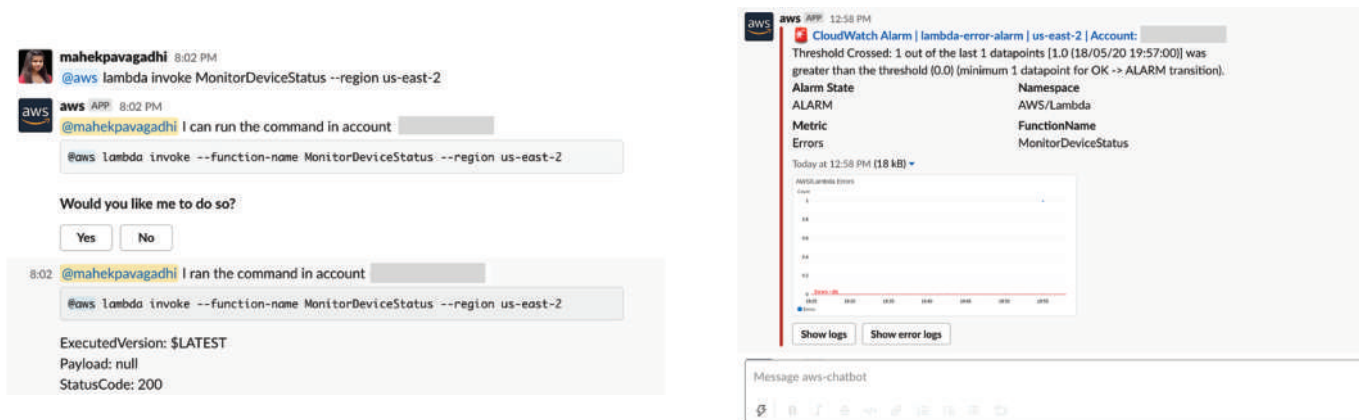
Now that we have initial set up ready, let's discuss a few use cases where you can use the bot with other AWS services.

Use cases

Notify Slack on Amazon EC2 CPU usage spike

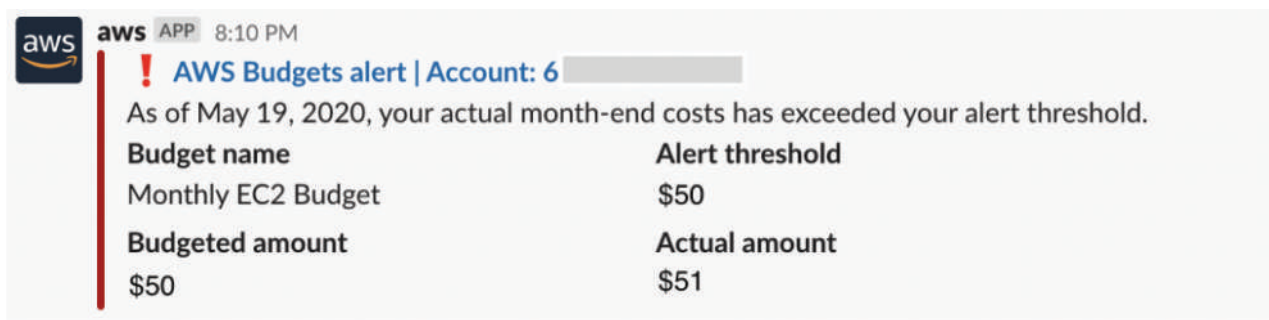


Run AWS Lambda Notify Slack on execution errors



The screenshot shows a Slack conversation on the left and a CloudWatch Alarm notification on the right. In the Slack chat, a user named mahekpavagadhi asks for help running a command. The AWS bot responds with the command `@aws lambda invoke --function-name MonitorDeviceStatus --region us-east-2` and offers to execute it. The bot then reports the execution status: `ExecutedVersion: $LATEST`, `Payload: null`, and `StatusCode: 200`. On the right, a CloudWatch Alarm notification is shown for 'lambda-error-alarm'. It indicates that the threshold was crossed (1 out of 1 datapoints) and provides details such as Alarm State (ALARM), Namespace (AWS/Lambda), Metric (Errors), and FunctionName (MonitorDeviceStatus). A graph shows the error count over time, and there are buttons to 'Show logs' and 'Show error logs'.

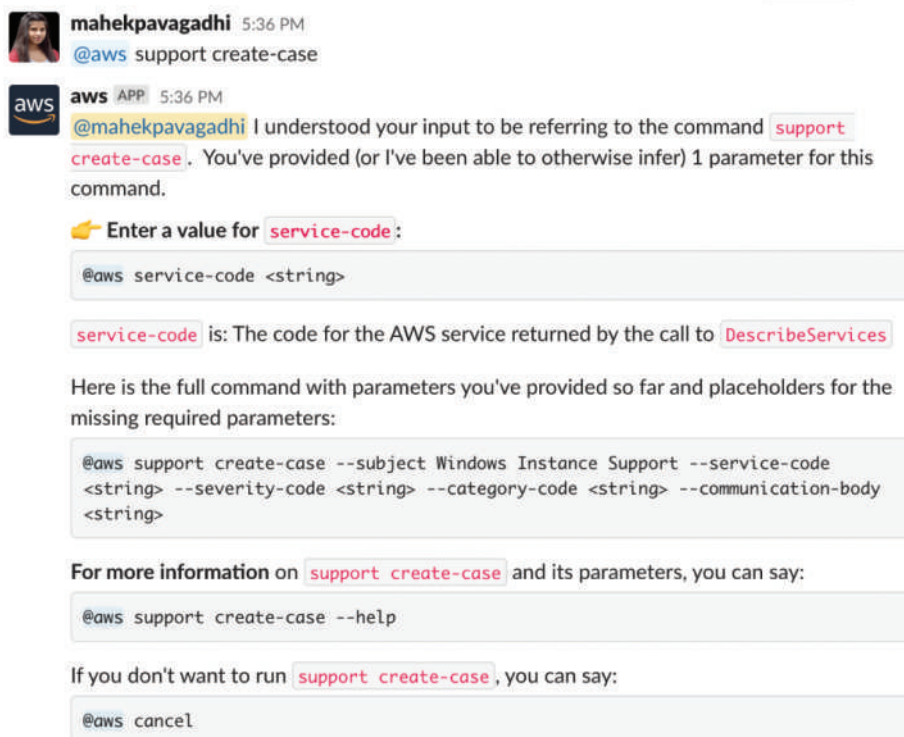
AWS Budget notifications on Slack when it exceeds threshold



The screenshot shows an AWS Budgets alert notification in Slack. The notification is titled 'AWS Budgets alert | Account: 6' and states: 'As of May 19, 2020, your actual month-end costs has exceeded your alert threshold.' Below this, a table provides details:


| Budget name | Alert threshold |
|--------------------|-----------------|
| Monthly EC2 Budget | \$50 |
| Budgeted amount | Actual amount |
| \$50 | \$51 |


Create AWS Support case using Slack



The screenshot shows a Slack conversation where a user asks for help creating an AWS support case. The AWS bot responds with instructions and a command. The bot asks for a 'service-code' and provides a definition: 'service-code is: The code for the AWS service returned by the call to DescribeServices'. It then shows the full command: `@aws support create-case --subject Windows Instance Support --service-code <string> --severity-code <string> --category-code <string> --communication-body <string>`. For more information, the bot provides the command `@aws support create-case --help`. Finally, the bot offers a way to cancel the command: `@aws cancel`.

GuardDuty security threat alerts on Slack

 **aws** APP 1:26 PM


 **GuardDuty Finding | us-west-2 | Account:** [REDACTED]


Finding type: Stealth:IAMUser/CloudTrailLoggingDisabled

AWS CloudTrail trail arn:aws:cloudtrail:us-west-2:[REDACTED]:trail/GuardDuty-CloudTrail-Monitor was disabled by Admin calling StopLogging under unusual circumstances. This can be attackers attempt to cover their tracks by eliminating any trace of activity performed while they accessed your account.

| | |
|------------------------------|------------------------------|
| First Seen | Last Seen |
| Tue, 2 Jun 2020 20:18:11 GMT | Tue, 2 Jun 2020 20:18:11 GMT |
| Severity | Threat Count |
| LOW | 1 |

Notify Slack on AWS CodePipeline errors

 **aws** APP 7:01 PM


 **AWS CodePipeline Notification | us-east-2 | Account:** 6E [REDACTED]


CodePipeline S3 Deploy action **FAILED**.

Additional Information: The input artifact, codepipeline-us-east-2-324174976817/nginx-build/BuildArtifact/3Bv6scr, does not exist or you do not have permissions to access it: The specified key does not exist. (Service: Amazon S3; Status Code: 404; Error Code: NoSuchKey; Request ID: 9A018E:[REDACTED]; S3 Extended Request ID: yQRbkZT+mWo+EjCatwLMv1a[REDACTED]3onHLAW6/s4hIssNwRlckQNHqqGC6EzA=)

| | |
|-----------------|--------------|
| Pipeline | Stage |
| nginx-build | Deploy |
| Action | |
| Deploy | |


Monitor operations on AWS Systems Manager parameter


 **aws** APP 5:17 PM

 **Systems Manager Event | us-east-2 | Account:** 6E [REDACTED]

Event Type: Parameter Store Change

Name github-token
Type SecureString
Operation Delete

 **aws** APP 4:52 PM

 **Systems Manager Event | us-east-2 | Account:** 6E [REDACTED]

Event Type: Parameter Store Change

Name github-token
Type SecureString
Operation Update

02

Conclusion

In summary, ChatOps and ChatSecOps revolutionize how teams collaborate by integrating operational and security workflows directly within chat environments like Slack and Amazon Chime. By leveraging tools like AWS Chatbot, organizations can streamline communication, receive real-time alerts, and quickly address operational and security issues. This approach not only accelerates response times but also fosters a culture of collaboration between DevOps, SecOps, and IT teams, aligning them toward a unified goal of efficiency and security. Embracing these concepts helps teams work smarter, enhance visibility, and maintain a proactive stance in managing both operational and security challenges.



Contact Us

Elevate your Team's collaboration and efficiency.
Schedule a personalized consultation today to explore how integrating
Slack with AWS Chatbot can transform your organization!



marketing@enreap.com



India | Singapore | Malaysia | UAE | USA



(1800) 2020-122



www.enreap.com