**ECS Alerts with Slack Integration for success**

**1. Service Setup**

**Event Bridge Configuration**

1. **Navigate to Amazon EventBridge**:
   * Open the [AWS Management Console](https://aws.amazon.com/console/) and go to **EventBridge**.
2. **Create a Rule**:
   * Name the rule appropriately (e.g., ECS-success-Alert).
   * Select **Event Source** as **AWS Services** and choose **ECS**.
3. **Add Custom Event Pattern**: Copy and paste the following JSON into the **Event Pattern** section, updating the service name (e.g., service:react-docker):
4. **Set Up a New Rule**:
   * Create another rule in EventBridge for **successful deployments**.
   * Use this event pattern:

**Code for event bridge**

{

"source": ["aws.ecs"],

"detail-type": ["ECS Task State Change"],

"detail": {

"lastStatus": ["RUNNING"],

"desiredStatus": ["RUNNING"],

"clusterArn": ["arn:aws:ecs:us-east-1:221082201369:cluster/express-cluster"],

"group": ["service:react-docker"]

}

}

1. **Configure a Lambda Function**:
   * Use the following code for the Lambda function:

**Code for lambda function**

import json

import http.client

from urllib.parse import urlparse

SLACK\_WEBHOOK\_URL = "YOUR\_SLACK\_WEBHOOK\_URL"

def lambda\_handler(event, context):

print("Event received:", json.dumps(event, indent=2))

detail = event.get('detail', {})

last\_status = detail.get('lastStatus')

desired\_status = detail.get('desiredStatus')

if last\_status == 'RUNNING' and last\_status == desired\_status:

message = "ECS task is successfully deployed."

slack\_data = json.dumps({"text": message})

url = urlparse(SLACK\_WEBHOOK\_URL)

conn = http.client.HTTPSConnection(url.netloc)

conn.request("POST", url.path, body=slack\_data, headers={'Content-Type': 'application/json'})

response = conn.getresponse()

if response.status != 200:

raise ValueError(f"Error {response.status}: {response.read().decode()}")

conn.close()

return {'statusCode': 200, 'body': 'Success alert sent to Slack!'}

* + Replace "YOUR\_SLACK\_WEBHOOK\_URL" with your Slack Webhook URL.

**Attach EventBridge Trigger**:

* + In the Lambda console, go to **Configuration** > **Triggers**.
  + Add EventBridge as a trigger and link it to the rule created earlier.

1. **Attach EventBridge Trigger**:
   * Link this rule to the Lambda function for successful deployment.