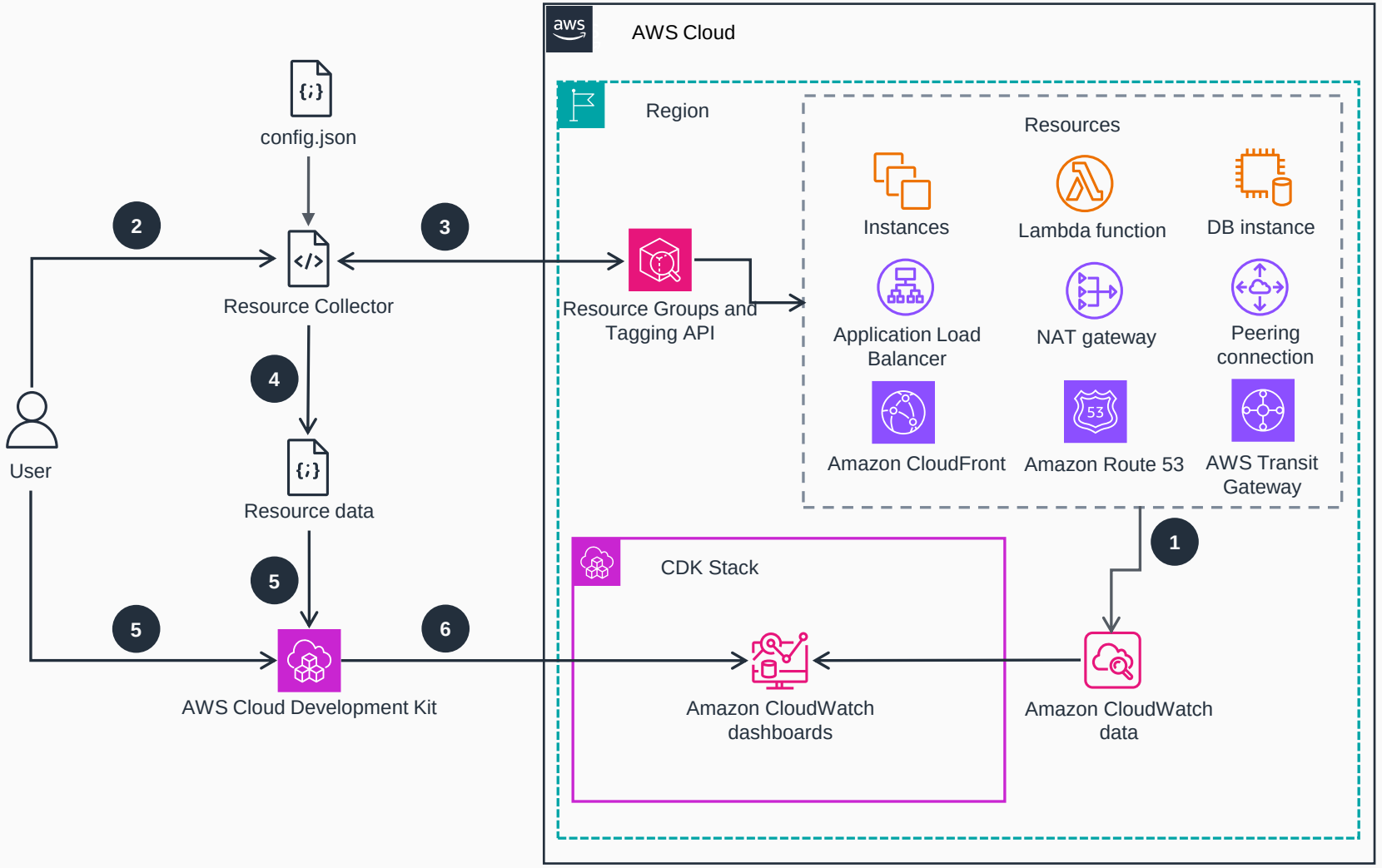
Guidance for Automating Networking Monitoring and Alerting on AWS

Overview

This architecture diagram illustrates the high-level automation process of deploying Amazon CloudWatch dashboards for network monitoring and alerting. The subsequent slides provide more detailed information on the configuration of monitoring (slide 2) and alerting (slide 3).



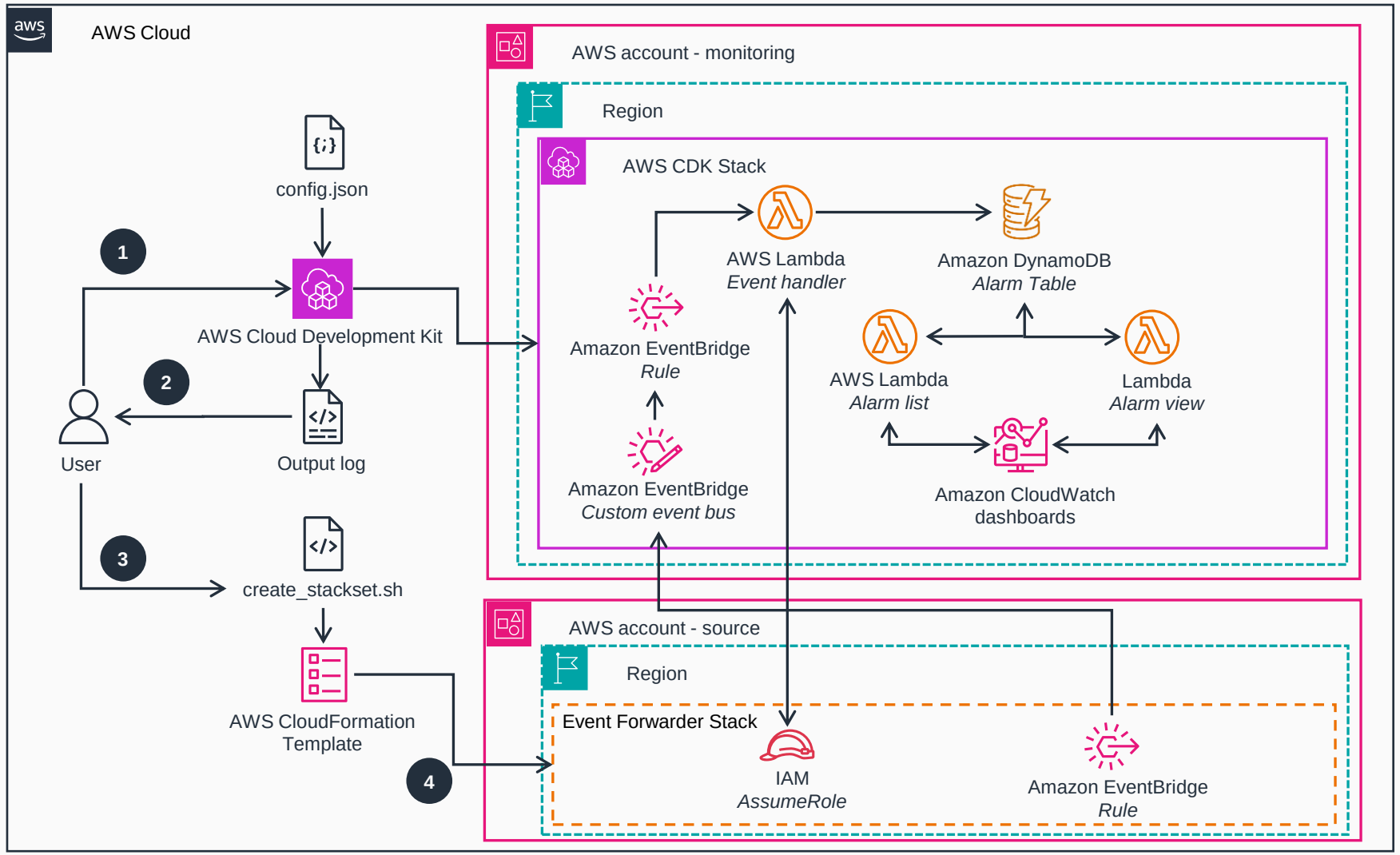
1. A group of **AWS Cloud** resources continuously store related metrics in the **Amazon CloudWatch** data store.
2. The user initiates the Guidance Resource Collector script that uses the config file.
3. The Guidance Resource Collector fetches resources matching the config file from the **AWS Resource Groups Tagging API Reference**.
4. The Guidance Resource Collector saves resource data in a JSON file.
5. The user initiates the **AWS Cloud Development Kit** (AWS CDK) to synthesize an **AWS CloudFormation** template. The **CloudFormation** template is using AWS monitoring best practices.
6. The user is asked for confirmation to deploy the template. Upon confirmation, the **AWS CDK** deploys the synthesized template containing **CloudWatch** dashboards.

AWS Reference Architecture

Guidance for Automating Networking Monitoring and Alerting on AWS

Monitoring

This architecture diagram shows how to generate and deploy the "Event Forwarder Stack," which is required for configuring the AWS accounts where the resources being monitored reside. These are the accounts that need to be configured to forward the CloudWatch alarm events to the central "monitoring" account.



1. The user runs the `cdk deploy` command to generate the **CloudFormation** template and deploy the infrastructure within the designated “monitoring” account.
2. The user records the output of the deployment, which contains the **Amazon Resource Names** (ARNs) of the central custom **Amazon EventBridge** event bus and the **AWS Lambda** function execution role.
3. The user provides the ARNs obtained from the previous step to generate the **CloudFormation** template for the "Event Forwarder Stack," which is required for configuring the source accounts.
4. The user deploys the **CloudFormation** template for the "Event Forwarder Stack" to the intended source accounts, either individually or across multiple accounts and Regions, using **CloudFormation** StackSets.

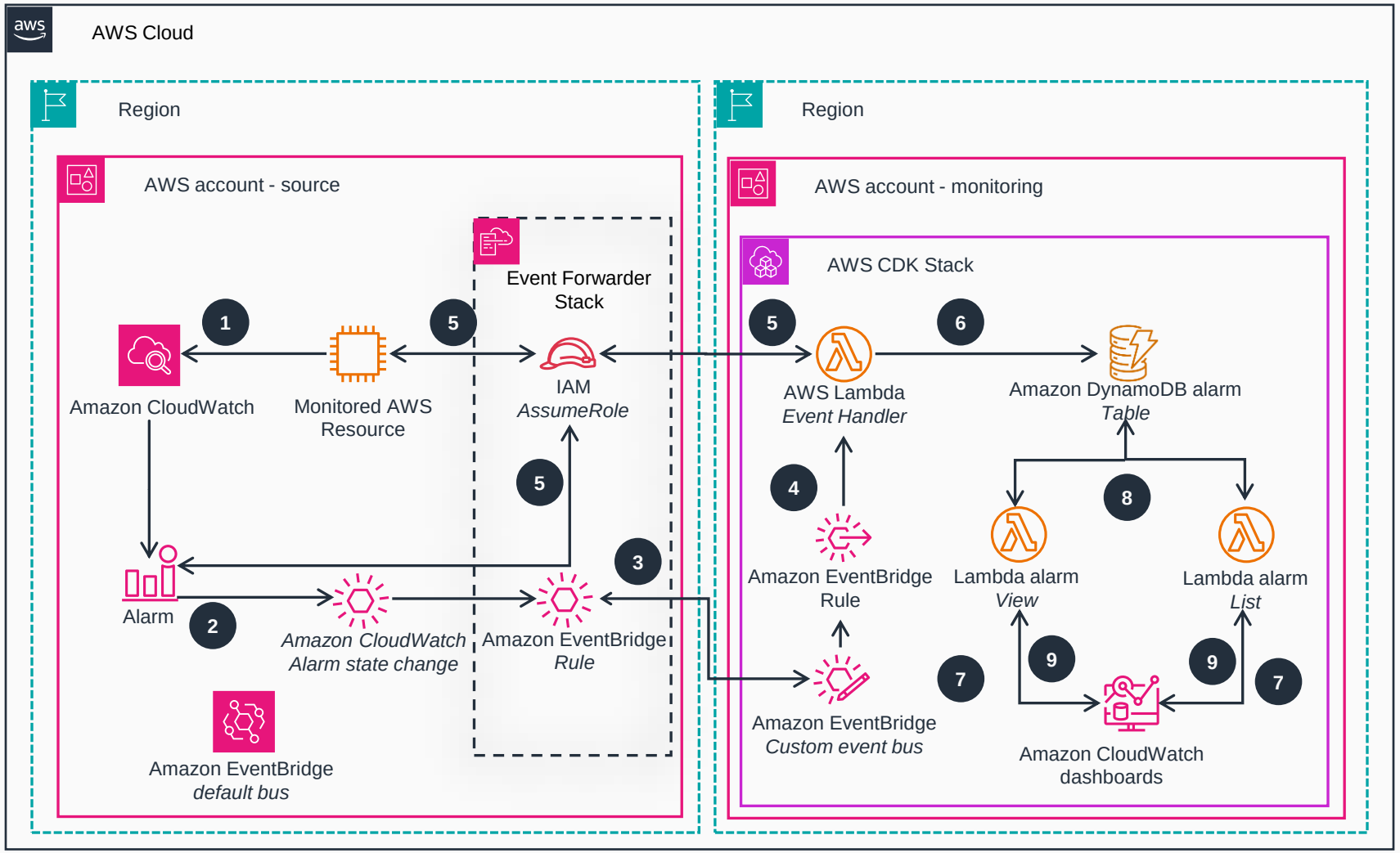
Reviewed for technical accuracy September 3, 2024

AWS Reference Architecture

Guidance for Automating Networking Monitoring and Alerting on AWS

Alerting

This architecture diagram shows the flow of events when a CloudWatch alarm is triggered. The alarm event is forwarded to an Amazon EventBridge event bus and processed by an AWS Lambda function. The ‘view’ and ‘list’ Lambda functions retrieve and render the alarm data in the CloudWatch dashboard.



1. An **AWS Cloud** resource sends a metric that breaches a threshold defined in a **CloudWatch** alarm.
2. When the alarm is triggered, **CloudWatch** emits a “*CloudWatch Alarm State Change*” event on the **EventBridge** default bus within the respective account.
3. An **EventBridge** Rule on the default bus forwards the event to the central custom **EventBridge** event bus.
4. An **EventBridge** Rule defined within the central event bus dispatches the event to the ”Event Handler” **Lambda** function that analyzes the event.
5. The ”Event Handler” **Lambda** function assumes an **AWS Identity and Access Management** (IAM) role that has been deployed by the “Event Forwarder” **CloudFormation** stack set in the source account. It then queries the monitored resource and the **CloudWatch** alarm for additional details.
6. The “Event Handler” **Lambda** function consolidates the additional details with the event and stores the combined information in an **Amazon DynamoDB** alarms table.
7. The **CloudWatch** dashboard, which includes custom **CloudWatch** widgets, triggers the execution of two **Lambda** functions—"View" and "List" —upon each dashboard refresh.
8. The “View” and “List” **Lambda** functions retrieve and filter the alarm data, then generate HTML code for rendering within the respective **CloudWatch** custom widgets.
9. The “View” and “List” **Lambda** functions return the HTML code to the **CloudWatch** widgets, which then render the code, including the relevant metrics, on the **CloudWatch** user interface.

Reviewed for technical accuracy September 3, 2024

AWS Reference Architecture