ORACLE

# OCI Events

## Level 100

KD Singh
Abhiram Annangi
Oracle Cloud Infrastructure
September, 2019

# Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

# OCI Events Service - Objectives

After completing this lesson, you should be able to understand:

- What are events and how can we use them?
- Events Service overview
- Key features of Events Service
- Core concepts of Events Service
- Common use-cases of Events Service

# OCI Events Service - Overview

cloudevents

- OCI Events service is a fully managed event-routing platform that simplifies the creation of event-driven cloud-native applications and serverless workflows.

- OCI Events service provides a platform where customers can subscribe to changes in their resources and automatically react to them in near real time using Fn, trigger notifications, or write to stream for later analysis.

- **Open Source** - the Events service implements the Cloud Native Computing Foundation's (CNCF) cloudevents open source standard. CloudEvents describes event data in a common, consistent, and accessible way across cloud native applications.

## Key Features

OCI Events service provides a robust Event-Routing Platform with some key features

- Integration with Oracle Functions, Oracle Streaming Service, and Oracle Notification Service, which provides you with a powerful reactive programming model for staying informed about your cloud environment.
- Support for diverse suite of out-the-box event types - supports all API call events from all your OCI resources (Create, Update, Delete via Audit), as well as scenario-specific events from Object Storage and Autonomous Database (such as Database Backup Complete).
- Integrated with Identity and Access Management and Monitoring services
- Accessibility through REST APIs, OCI console, SDKs, CLI, Terraform

# Common Use-cases

Polling resources continuously to track changes has problems

- More CPU Cycles
- More Network Traffic
- More Engineering Resources
- Delay in Responding

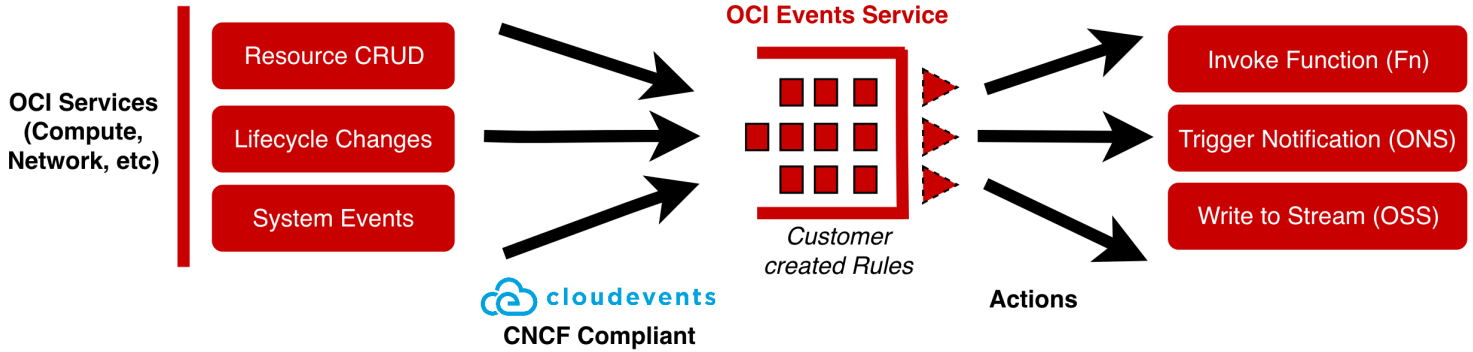
Following are some common actions that Events can perform:

- Trigger a function when new files are uploaded in an Object Storage bucket.
- Publish a notification when long-lived tasks complete, such as Autonomous Database backup completion.
- Archive all events in a specific compartment to a stream for later analysis.
- Publish a notification when a new resource is created or deleted in a compartment.

# Concepts

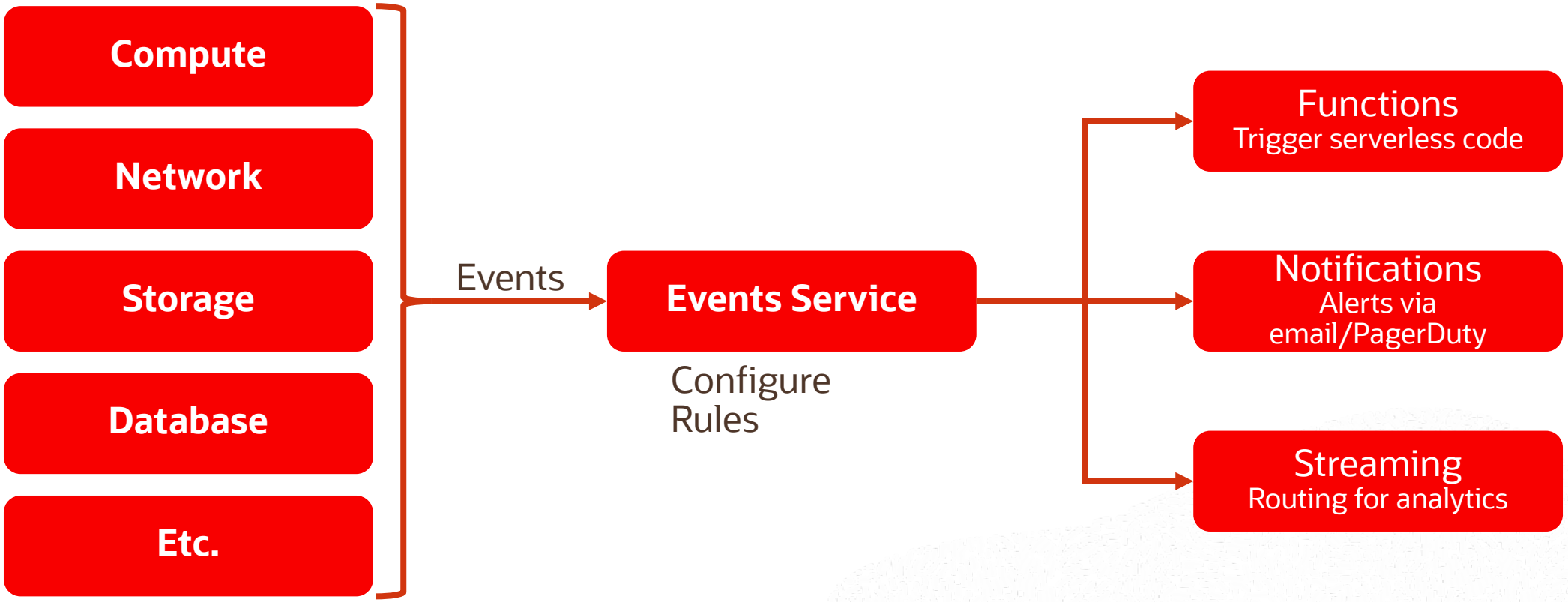
There are three core concepts in OCI Events service:

- **Events** - A structured and schematized message that denotes a change in a resource.
- **Rule** - The object where a user defines which events they care about, and trigger an action if it occurs. A first-class OCI object created per compartment.
- **Actions** - The user-defined response to when an event occurs. For example, triggering a function, or writing to a stream.



# OCI services that trigger Events and take Actions




Events Service integrates with OCI Services **at the platform-level** to deliver resource change events.

# Concepts - What are Events?

An event is a structured, lightweight, actionable message that denotes a change in a resource. Unlike raw generic log entries, events have derived context and structure, and are guaranteed to be actionable.

An event can be:

- **User initiated CRUD operation** - "Bucket Updated", "Bucket Deleted"
- **Resource life cycle state change** - "Instance Stopped", "Backup starting"
- **System event** - "Instance Rebooted - Hardware Failure" (Via PULSE)

Each event describes:

- The source (i.e service responsible)
- Time stamp
- eventType (i.e Backup Complete)
- a detailed service-specific inner payload to describe the change and resource in more details (including the tags associated with the resource).

# Concepts - How does an Event look like?

Events follow CNCF format and has the following schema:

```
{
  "eventType": "string", #Example: ADW.instance.backup.complete
  "eventTypeVersion": "string",
  "cloudEventsVersion": "string",
  "source": "URI/string", #Example /service/dbaas/resourceType/ADW/
  "eventID": "string",
  "eventTime": "timestamp", #Time the event occured
  "contentType": "string", #application/json
  "extensions": {
    "compartmentId": "string" #extension to show compartment ID
  },
  "data": "JSON object" #Inner payload with service specific details.
}
```

# Concepts – What are Rules?

- To interact with the OCI Events service, users create and manage "Rules".
- Rules are objects that allow customers to select which event types to monitor, and automatically trigger actions when those events occur. To be able to create rules in a given compartment, users will need the INSPECT compartment permission.
- All rules are validated for IAM permissions.

A rule is simple; to create one, the user specifies:

- **Name & Compartment** - A name, and the compartment where you want the rule to be created.
- **Trigger Condition** - The event types you care about, as well as any other property filters. A user can subscribe to ALL events in their compartment.

  Example: "Event Type = Delete bucket" and "Tags = Prod_Dont_Delete"
- **Response Action** - The automated action to be executed once the condition above is met. You can have multiple actions per rule.

  Example: "Notify DRI (ONS)" and "Trigger my backup scripts (Fn)"

# Design Considerations - Rules

- Rules are compartment based, and will support nested compartments. If a customer wants to set up a tenancy-wide rule, they can create one in the root compartment.
- To be able to create a rule in a given compartment, customers will need the compartment INSPECT permission. All rules, and actions are validated for proper IAM permissions. Customers will also need the 'manage cloudevents' permissions to be able to CRUD rules.
- Before using the service, customers will need to set a policy to allow events service to deliver events to action resources.
- When events are generated, they also include the tags of the resource that fired the event. Customers will be able to create rule filters that match their resource tags. Rules resource themselves will also support tags.
- The max amount of rules a user can create in their tenancy is 50. (This can be increased)

# Concepts – What are Actions?

- Actions are the user defined response to a rule being matched.
- A user can specify multiple actions per rule.
- The service guarantees at least one delivery attempt for all actions.
- Supported actions include:
    - Trigger Function (Fn)
    - Publish notification (ONS)
    - Output to Stream (OSS)

# IAM policies required to work with Events #1

**Policies for the Events service so that it can deliver event messages to action resources, which can be any combination of topics, streams, or functions**

- give the Events service the ability to deliver events messages to a topic

  allow service cloudEvents to use ons-topic in tenancy
- policy for Events to deliver event messages to functions

  allow service cloudEvents to use functions-family in tenancy
- policy for Events to deliver event messages to streams

  allow service cloudEvents to use stream-push in tenancy

  allow service cloudEvents to read streams in tenancy

# IAM policies required to work with Events #2

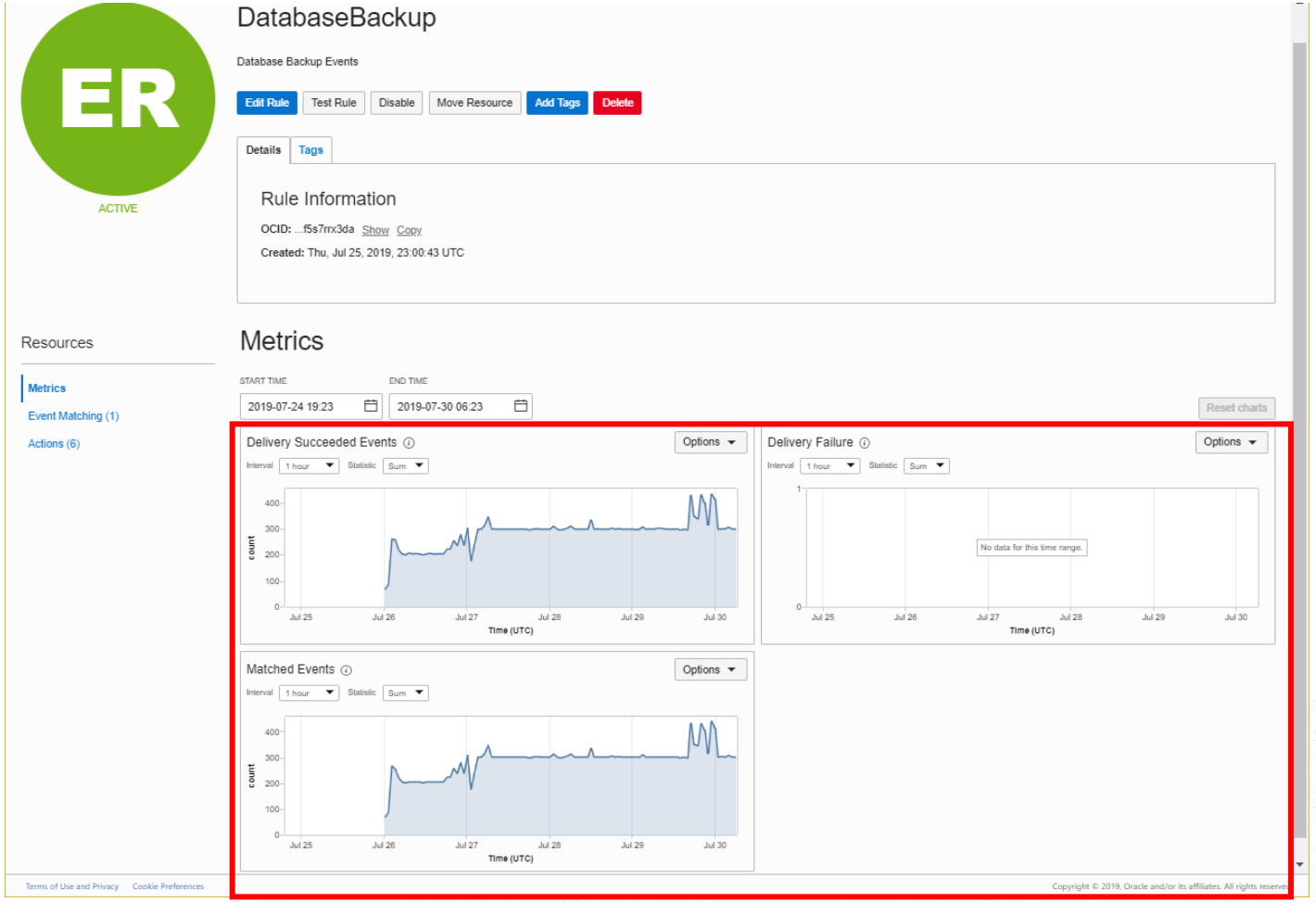
## Policies for users so that they can create and manage rules

- give users inspect access to resources in compartments to select actions
  allow group <RuleAdmins> to inspect compartments in tenancy
- give users manage access to rules for Events
  allow group <RuleAdmins> to manage cloudevents-rules in tenancy
- give users access to Streaming resources for actions (if required)
  allow group <RuleAdmins> to inspect streams in tenancy
  allow group <RuleAdmins> to use stream-push in tenancy
  allow group <RuleAdmins> to use stream-pull in tenancy
- give users access to Functions resources for actions (if required)
  allow group <RuleAdmins> to use virtual-network-family in tenancy
  allow group <RuleAdmins> to manage function-family in tenancy
- give users access to Notifications topics for actions (if required)
  allow group <RuleAdmins> to use ons-topic in tenancy

# Events Service Metrics

Metrics supported for:

- Events received from all resources
- Events matched by a rule
- Successful deliveries by a rule
- Failed deliveries by a rule



# Service Guarantees

OCI Events Service offers the following guarantees:

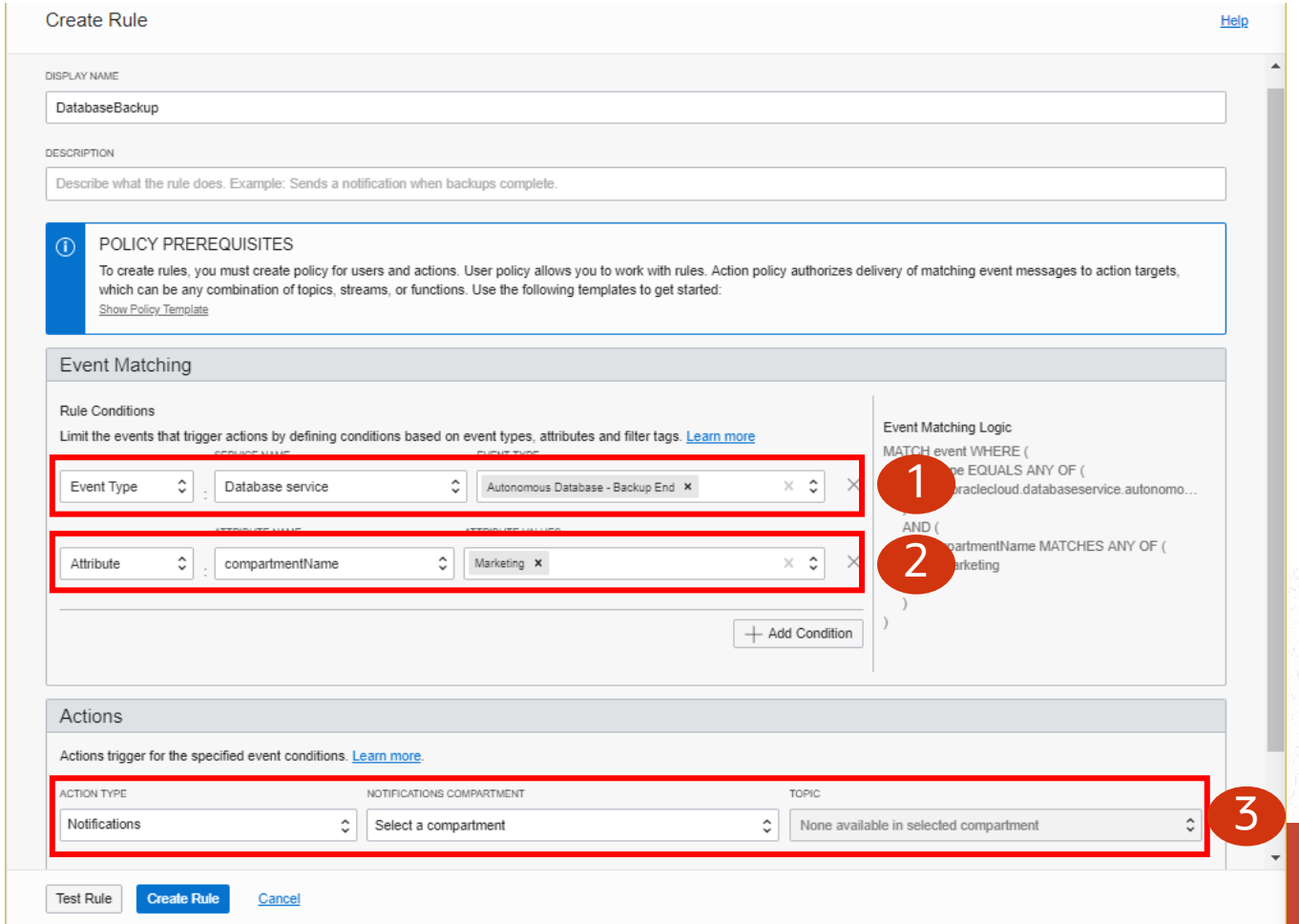
- If an event is ingested, it guarantees that it will be evaluated at least once against user rules.
- If a rule is matched, it guarantees at least one delivery attempt for all actions.
- Events are **NOT** guaranteed to be processed or received in order.
- If an action target is not responsive, the service will retry delivery for up to 5 hours or until a non-retryable error occurs. Otherwise a failure metric will be emitted, and no further retries will occur.

# Configuring Events Service Rules in OCI Console

Step 1: Select Service and Event Type

Step 2: Set a filter

Step 3: Select an action



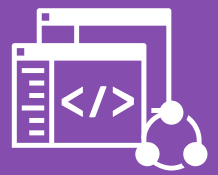

 

# Event-Driven Design Patterns - Common Use Cases

Event-driven applications

Web, Mobile, IoT Backends

Real-time File, Stream Processing



DevOps, Batch Processing

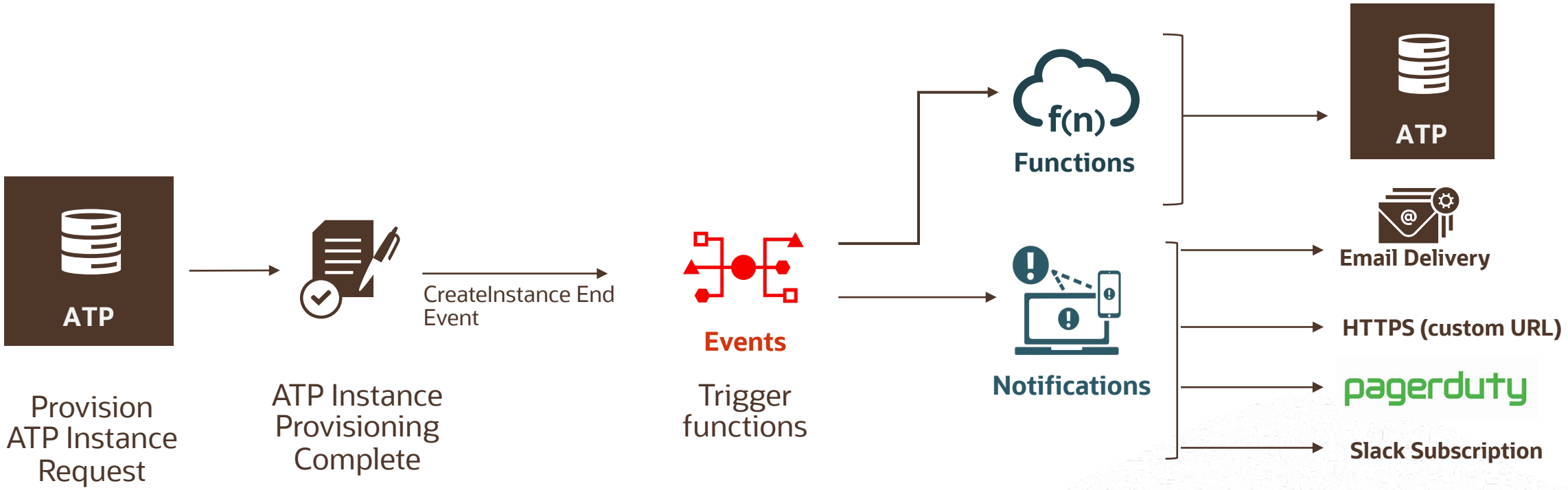
Glue Cloud Services, Event-driven

Security Operations

IT Operations

Multi-Cloud

# An Example of OCI Events Service in Action




- Functions runs scripts to create schemas, tables and import golden data.
- Notifications triggers email and PagerDuty alerts.

# More Examples – Automate Corporate Security Actions


Provision Instance Request

Instance Provisioning Complete

Trigger functions

Functions check if the instance complies with security policies. If not, kill the instance & send a new provision instance request

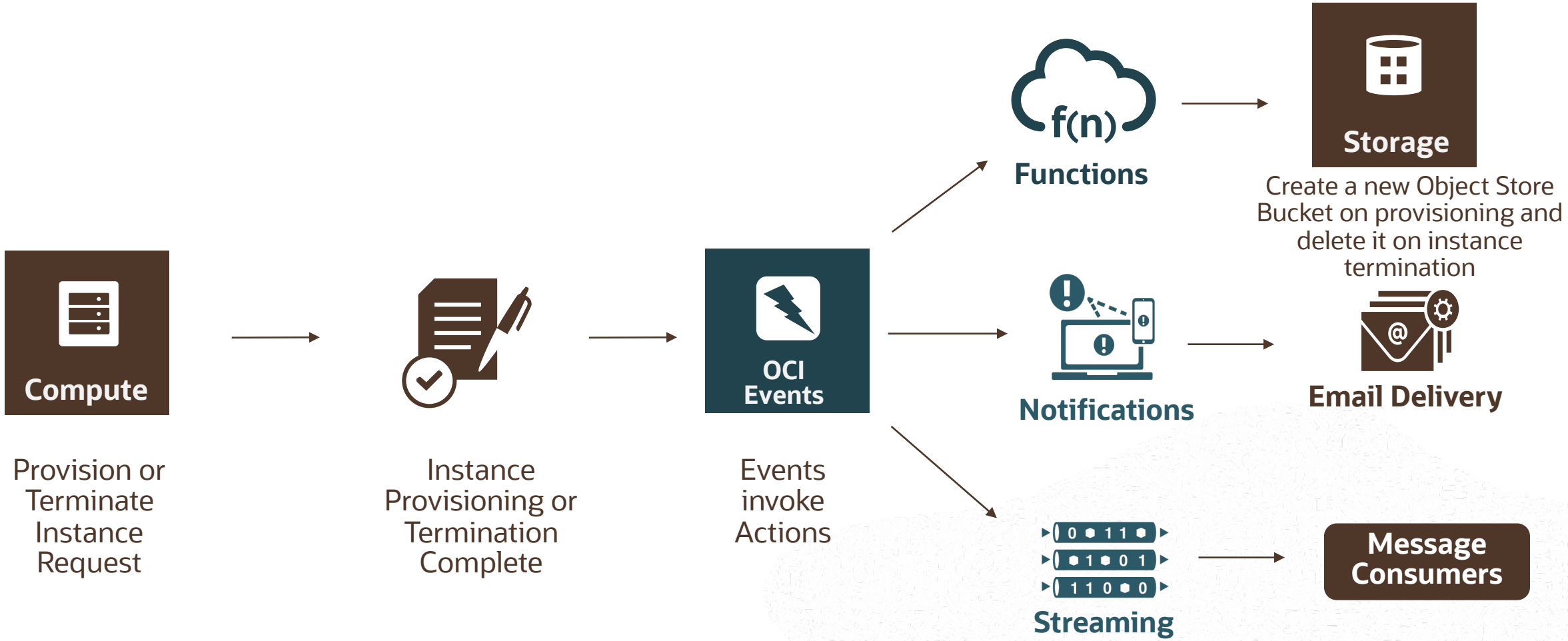

Network Access Logs

Log Files in Object Store

Trigger functions

Functions read the network access logs and send them to customer's Splunk for security analysis

# Demo: Integrating an OCI Service event with Events, Functions, Notifications and Streaming



# Summary

- Fully managed event ingestion and routing platform that enables users to automatically detect changes on their resources and act upon them.
- Customers simply pick the services they care about, the type of event they want to monitor, and the actions they want to take.
- **Free** service with a native **CNCF** cloudevents support.
- Integrated with IAM and Monitoring
- Accessible through REST APIs, OCI console, SDKs, CLI, Terraform
- Roadmap
  - Support for custom events via OSS
  - Support for “Advanced” flow, which allows users to input custom verbose json for more complex rules.

ORACLE

**Oracle Cloud always free tier**:
oracle.com/cloud/free/

**OCI training and certification**:
cloud.oracle.com/en_US/iaas/training
cloud.oracle.com/en_US/iaas/training/certification
education.oracle.com/oracle-certification-path/pFamily_647

**OCI hands-on labs**:
ocitraining.qloudable.com/provider/oracle

**Oracle learning library videos on YouTube**:
youtube.com/user/OracleLearning