

# Architecting for Security on AWS

---

## PROTECTING AWS CREDENTIALS



**Kien Bui**

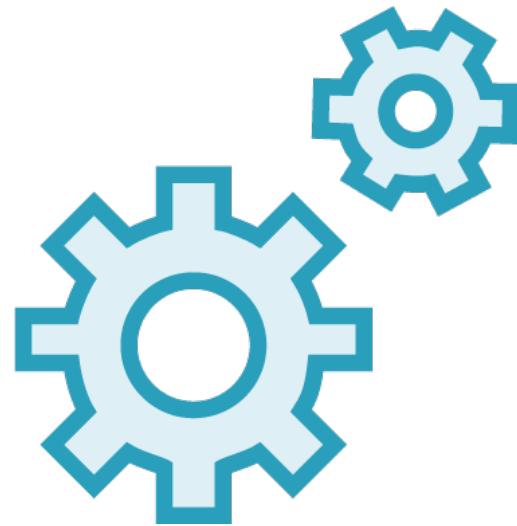
DevOps & Platform Engineer



# Views of Security



A good feeling



Restrictive  
configuration  
settings



Security  
patching



Monitoring



Security is about  
protecting data



# The CIA Triad

**Confidentiality**

**Integrity**

**Availability**



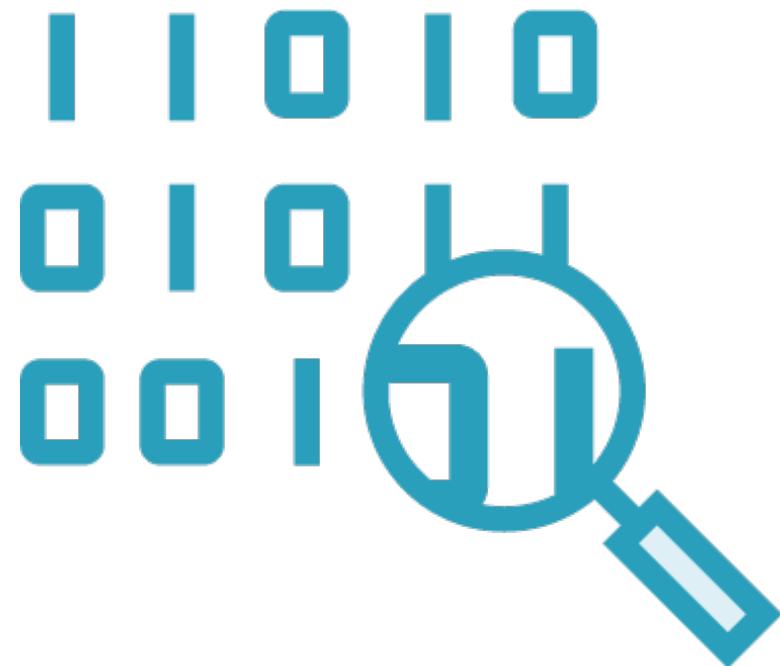
# Confidentiality



Only authorized parties can access data  
Examples: ACLs and encryption



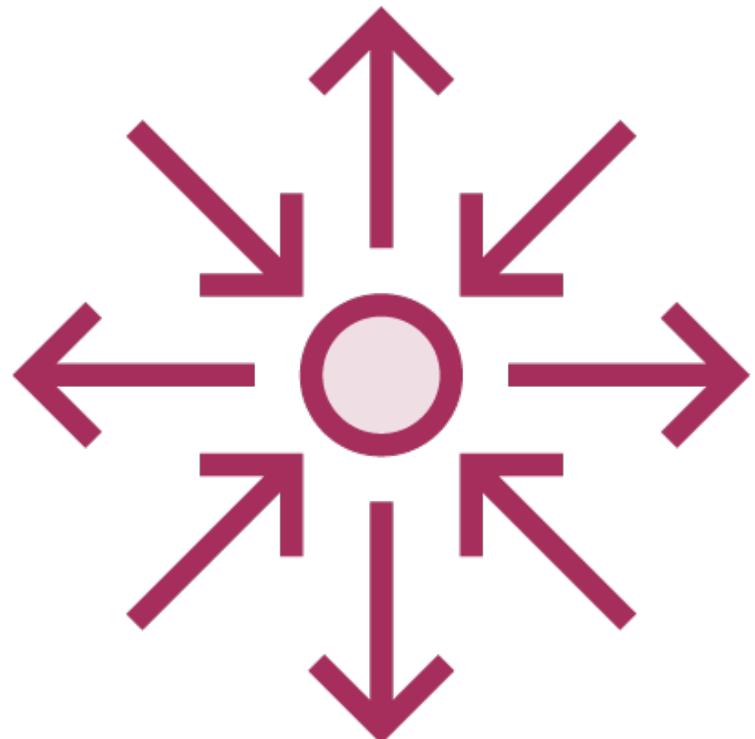
# Integrity



**Data has not been improperly modified**  
**Includes knowing if data has been modified**



# Availability



**Authorized parties have access to data when they need it**

**Includes protecting systems that store, process, and deliver data**



# Defense in depth

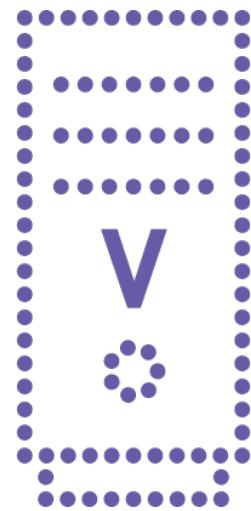
Protecting the confidentiality, integrity, and availability of data by securing everything that touches the data, including storage, compute, and networking



# Levels of Architecture



AWS services



Operating systems



Applications



# Course Overview

---



# Prerequisites



**6 months of technical experience  
with AWS**



**AWS root user access**



# Course Overview



**Protecting AWS credentials**

**Capturing and analyzing logs**

**Protecting network and host-level boundaries**

**Protecting data at rest**

**Protecting data in transit**

**Configuring data backup, replication, and recovery**



# Protecting AWS Credentials



**Identity and Access Management (IAM)**



# Capturing and Analyzing Logs



**What to log and how to organize logs**

**Storing logs**

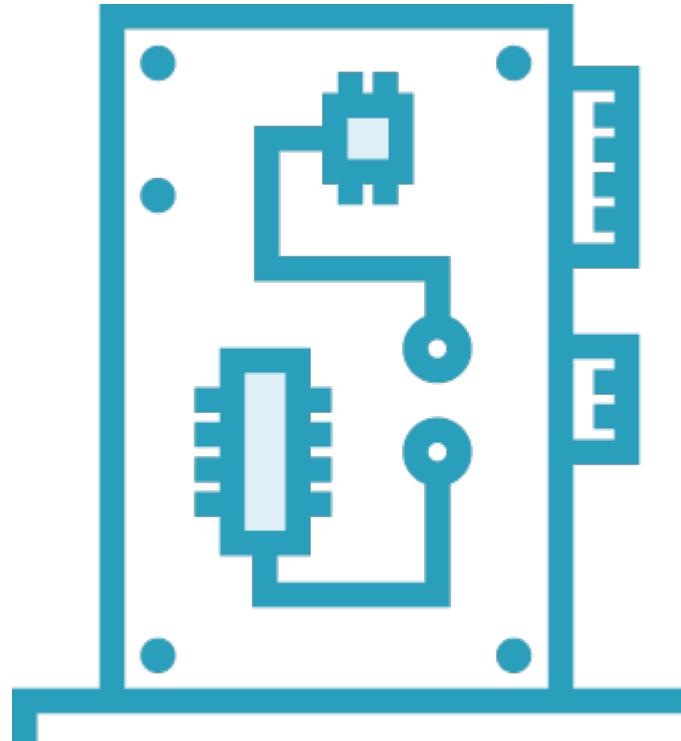
**Alerting**

**Searching**

**CloudTrail and CloudWatch**



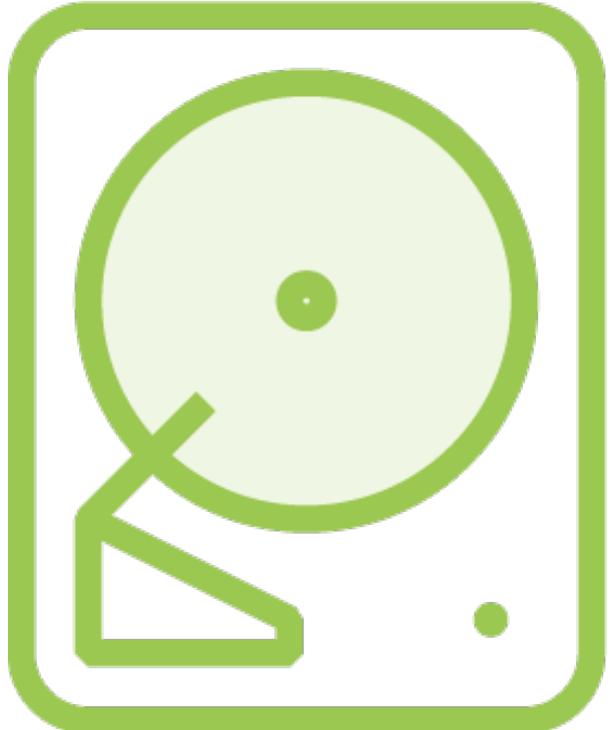
# Protecting Network and Host-level Boundaries



**Network and OS-based controls**



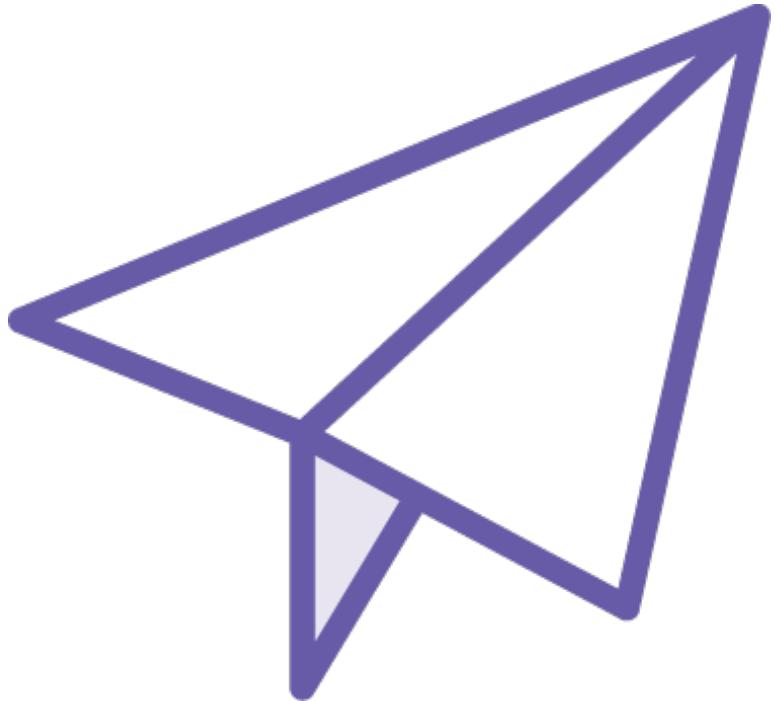
# Protecting Data at Rest



**Encryption**  
**Permissions**



# Protecting Data in Transit



**Encryption ensures confidentiality  
and integrity**



# Data Backup, Replication, and Recovery



**Ensures the availability of data**

# Understanding AWS Credentials

---



# AWS Credential Types

## Root user

Full access to all AWS resources

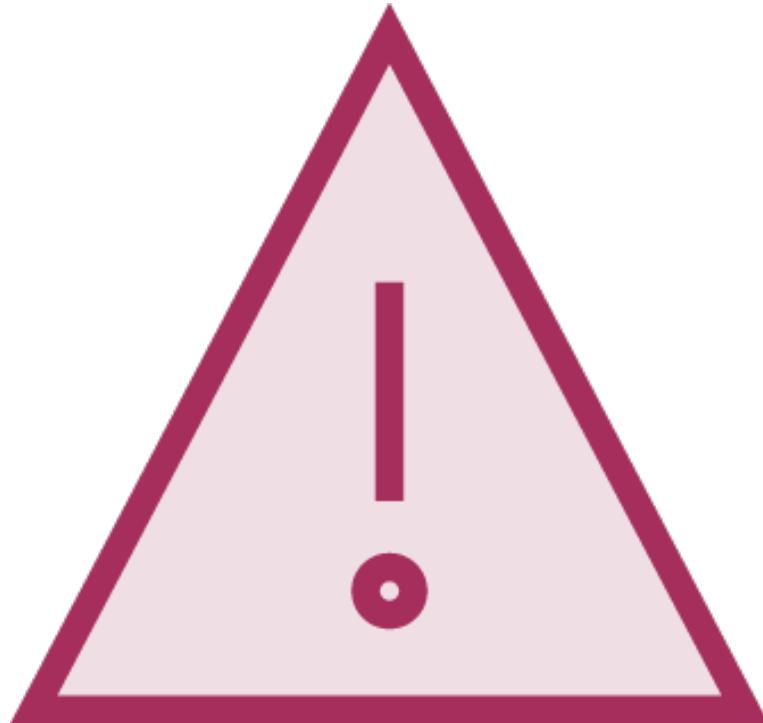
Only one root user per account

## IAM (non-root) principal

Any entity that can perform actions on AWS services and resources

Policies determine what permissions a principal has





**Be careful about using the word “account”**

**An AWS account is the container that houses resources and billing information**

**You can log into an AWS account using the root user**



# Locking Down the Root User

---



# Locking Down the Root User

## Enable multi-factor authentication (MFA)

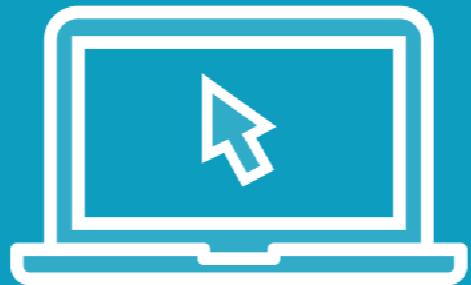
Requires an email address, password, and one-time passcode

## *Don't use the root user for administrative tasks*

Use a non-root IAM user with administrative permissions



Demo



**Enable multi-factor authentication for the root user**



# Introduction to Principals and Policies

---



# IAM Principal

The foundation of IAM

An entity that can take an action on an AWS resource

Often used as a synonym for identity

Principles include users and roles

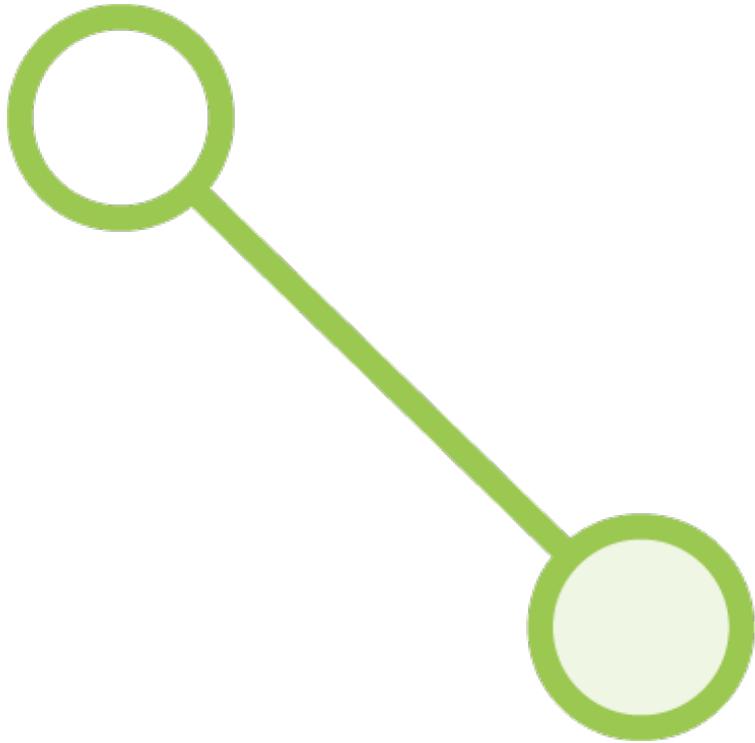




A non-root *principal* has no permissions by default

*Policies* determine what permissions a principal has





You must grant permissions to a principal  
by associating it with a policy

# Understanding Policies

---



# Permission Statement

Element	Example
Effect (allow or deny)	Allow
Service	EC2
Action/operation	RunInstances
Resource (depends on service)	image/ami-e5d9439a
Request condition (MFA, IP range, time)	198.51.100.0/24



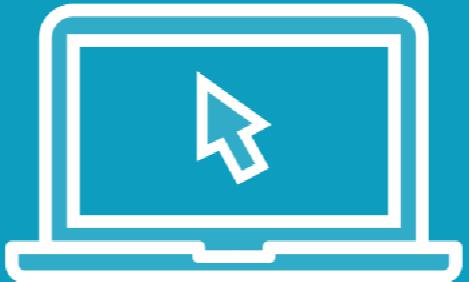
# AWS Managed Policies

Cover a variety of  
common scenarios

Updated regularly to include  
new services and actions



# Demo



**Examine the AdministratorAccess policy**

**Create a new IAM user**

**Attach the AdministratorAccess policy to the new user**



# Denying Access with User Policies

---

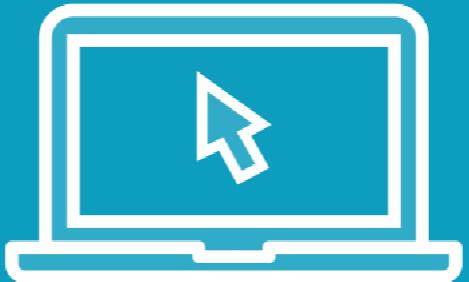


# Using the Deny Effect

<b>AdministratorAccess</b>	<b>RestrictAdmin</b>
Effect: allow	deny
Service: *	EC2
Action: *	TerminateInstances
Resource: *	*



Demo



**Create another user and associate with  
the AdministratorAccess policy**

**Implement another policy to deny access  
to the TerminateInstances action**



## Summary



**Implement MFA for the root user**

**Use an administrative user instead of root**

**AWS Managed Policies are updated as new services and actions are added**

**A policy permission consists of an effect, service, action/operation, and resource**

**A user policy is an inline policy embedded in a user**

**A group policy is embedded in a group**

**Customer Managed Policies work like AWS Managed Policies, but are created by you**





Coming up Next  
**Capturing and Analyzing Logs**

