

TECH Startup

Career Simulation 3

Eddie Leach

IAM Users and Permissions, Create EC2, Create S3, Knowledge Check

As the Cloud Solution Architect for TECH-Start's infrastructure migration initiative, my primary goal was to design and implement a scalable, secure, and cost-effective cloud environment using AWS Free Tier services. This project involved provisioning core resources to support the company's diverse team, including EC2 instances for application hosting and S3 storage for data access and collaboration. I established IAM users with role-based access controls aligned to each team's responsibilities, ensuring that developers, data scientists, and marketing personnel have appropriate permissions. Through the creation of these services and configurations, I've laid the groundwork for a cloud-native environment that supports TECH-Start's continued growth while adhering to AWS best practices in security and cost management.

Task 1: Create Multiple User Accounts to define IAM concepts.

Users (4) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

<input type="checkbox"/>	User name	Path	Group:	Last activity	MFA	Password age
<input type="checkbox"/>	admin-user	/	0	✓ 4 hours ago	Passke...	✓ 4 hours
<input type="checkbox"/>	datascientist-user	/	1	-	-	✓ 22 hours
<input type="checkbox"/>	developer-user	/	1	-	-	✓ 22 hours
<input type="checkbox"/>	marketing-user	/	1	-	-	✓ 22 hours

Task 2: Set permissions on User Accounts.

Data Scientist Permissions

datascientist-user [Info](#)

Summary

ARN
[arn:aws:iam::727646475213:user/datascientist-u](#)
Created
April 24, 2025, 00:27 (UTC-05:00)

[Permissions](#) [Groups \(1\)](#) [Tags \(1\)](#) [Seci](#)

Permissions policies (3)

Permissions are defined by policies attached to the u

- ☐ [Policy name](#)
- ☐ [AmazonEC2FullAccess](#)
- ☐ [AmazonS3FullAccess](#)
- ☐ [IAMUserChangePassword](#)

Developer Permissions

developer-user [Info](#)

Summary

ARN
[arn:aws:iam::727646475213:user/developer-](#)
Created
April 24, 2025, 00:11 (UTC-05:00)

[Permissions](#) [Groups \(1\)](#) [Tags \(1\)](#) [s](#)

Permissions policies (2)

Permissions are defined by policies attached to t

- ☐ [Policy name](#)
- ☐ [AmazonEC2FullAccess](#)
- ☐ [IAMUserChangePassword](#)

Marketing Permissions

marketing-user [Info](#)

Summary

ARN
[arn:aws:iam::727646475213:user/marketing](#)
Created
April 24, 2025, 00:31 (UTC-05:00)

[Permissions](#) [Groups \(1\)](#) [Tags \(1\)](#) [:](#)

Permissions policies (2)

Permissions are defined by policies attached to t

- ☐ [Policy name](#)
- ☐ [AmazonS3ReadOnlyAccess](#)
- ☐ [IAMUserChangePassword](#)

Additional Task: The permissions were set by adding users to groups. This enables any new users added to the same group to inherit the same permissions. See the visual below.

Groups																			
<div>User groups (3) Info</div> <div>A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.</div> <div><div>Q Search</div></div> <table><tr><td><input type="checkbox"/></td><td>Group name</td><td>▲ Users</td><td>▼ Permissions</td></tr><tr><td><input type="checkbox"/></td><td>Data-Scientists</td><td>1</td><td>✔ Defined</td></tr><tr><td><input type="checkbox"/></td><td>Developers</td><td>1</td><td>✔ Defined</td></tr><tr><td><input type="checkbox"/></td><td>Marketing</td><td>1</td><td>✔ Defined</td></tr></table>				<input type="checkbox"/>	Group name	▲ Users	▼ Permissions	<input type="checkbox"/>	Data-Scientists	1	✔ Defined	<input type="checkbox"/>	Developers	1	✔ Defined	<input type="checkbox"/>	Marketing	1	✔ Defined
<input type="checkbox"/>	Group name	▲ Users	▼ Permissions																
<input type="checkbox"/>	Data-Scientists	1	✔ Defined																
<input type="checkbox"/>	Developers	1	✔ Defined																
<input type="checkbox"/>	Marketing	1	✔ Defined																

Data-ScientistsInfo

Summary

User group name
Data-Scientists

Users(1)

Permissions

Users in this group (1)

An IAM user is an entity that you can use to manage permissions.

Q Search

<input type="checkbox"/>	User name↗
<input type="checkbox"/>	datascientist-user

Data-ScientistsInfo

Summary

User group name
Data-Scientists

Users(1)



Permissions

Access Advisor

Permissions policies (2)Info

You can attach up to 10 managed policies.

Q Search

<input type="checkbox"/>	Policy name↗
<input type="checkbox"/>	 AmazonEC2FullAccess
<input type="checkbox"/>	 AmazonS3FullAccess

DevelopersInfo

Summary

User group name
Developers

Users(1)

Permissions

Users in this group

An IAM user is an entity that you can use to manage permissions.

Q Search

<input type="checkbox"/>	User name↗
<input type="checkbox"/>	developer-user

DevelopersInfo

Summary

User group name
Developers

Users(1)


Permissions




Access Advisor

Permissions policies (1)Info

You can attach up to 10 managed policies.

Q Search

<input type="checkbox"/>	Policy name↗
<input type="checkbox"/>	 AmazonEC2FullAccess

Marketing Groups: Users/Permissions									
<div><div><div><div><div>Marketing Info</div><div><div>Summary</div><div>User group name Marketing</div></div></div><div><div>Users (1)</div><div>Permissions</div></div><div><div>Users in this group</div><div>An IAM user is an entity that you can use to manage permissions.</div><div><div>Q Search</div><table><tr><td><input type="checkbox"/></td><td>User name ↗</td></tr><tr><td><input type="checkbox"/></td><td>marketing-user</td></tr></table></div></div></div><div><div><div><div>Marketing Info</div><div><div>Summary</div><div>User group name Marketing</div></div></div><div><div>Users (1)</div><div><div>Permissions</div><div>Access Advisor</div></div></div><div><div>Permissions policies (1) Info</div><div>You can attach up to 10 managed policies.</div><div><div>Q Search</div><table><tr><td><input type="checkbox"/></td><td>Policy name ↗</td></tr><tr><td><input type="checkbox"/></td><td> AmazonS3ReadOnlyAccess</td></tr></table></div></div></div></div></div></div>	<input type="checkbox"/>	User name ↗	<input type="checkbox"/>	marketing-user	<input type="checkbox"/>	Policy name ↗	<input type="checkbox"/>	 AmazonS3ReadOnlyAccess	
<input type="checkbox"/>	User name ↗								
<input type="checkbox"/>	marketing-user								
<input type="checkbox"/>	Policy name ↗								
<input type="checkbox"/>	 AmazonS3ReadOnlyAccess								

Task 4: Types of storage options and build / create S3 Storage.

Part 1: AWS Storage Types

1. Amazon S3 (Simple Storage Service)

- **Object storage** for files, backups, media, logs, etc.
- Durable, scalable, and great for static website hosting or storing data shared across teams.
- **Best for:**
 - Hosting static content (e.g., `index.html`)
 - Marketing team data (read-only access)
 - Data scientists uploading/downloading files

2. Amazon EBS (Elastic Block Store)

- **Block storage** attached to EC2 instances (like a hard drive).
- Persistent — survives even if the instance is stopped/rebooted.
- **Best for:**
 - Storing your EC2 web server's OS, installed software, or logs
 - Hosting databases on EC2

3. Amazon EFS (Elastic File System)

- **File-level storage** shared across multiple EC2 instances.
- Automatically scales, accessible via NFS.
- **Best for:**
 - Applications needing shared file systems (e.g., microservices, machine learning jobs)

Summary:

Team or Feature	Recommended Storage	Why
Web app (on EC2)	EBS	Used by the EC2 instance to store OS and web files
File storage (images, backups, reports)	S3	Scalable, easy to manage, and accessible across users
Shared storage across EC2 (optional)	EFS	Not needed for this project unless you're sharing files between multiple EC2s

Part 2: Build / Create S3 Storage

The screenshot shows the Amazon S3 console interface. On the left, there's a navigation menu with options like 'General purpose buckets', 'Directory buckets', 'Table buckets', 'Access Grants', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', and 'IAM Access Analyzer for S3'. The main content area is titled 'General purpose buckets (1)' and shows a list of buckets. The first bucket listed is 'tech-start-bucket-leach-engineering' in the 'US East (Ohio) us-east-2' region. Above the list, there are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A 'View Storage Lens dashboard' link is also present. The console includes a search bar and filters for 'All AWS Regions' and 'IAM Access Analyzer'.

Knowledge Check:

1. Explain the importance of Availability Zones

- **Definition:**

An **Availability Zone (AZ)** is a distinct, isolated data center within an AWS Region. Each region contains multiple AZs, which are designed to be physically separate but interconnected through low-latency, high-throughput networking.

- **Contribution to Reliability and Scalability:**

Availability Zones allow AWS to provide fault tolerance and high availability by enabling workloads to be distributed across multiple physical locations. If one AZ experiences an outage, services in other AZs can continue operating. This setup also enables scalable architectures, where applications can elastically grow across zones to handle more demand.

- **Real-World Example:**

A web application hosted in AWS can run EC2 instances in two or more Availability Zones with a load balancer distributing traffic. If one zone fails, the application remains accessible via the other zone(s), ensuring business continuity.

2. Differences and use cases of AWS CloudWatch vs. AWS CloudTrail

- **AWS CloudWatch**

- Monitors performance metrics, logs, and alerts in near real-time.
- **Use case:** You use CloudWatch to monitor CPU usage on an EC2 instance and trigger an alarm when it exceeds 80%, automatically launching a new instance to handle increased load.

- **AWS CloudTrail**

- Logs API activity and user actions across your AWS account for governance, compliance, and auditing.
- **Use case:** You use CloudTrail to track who deleted an S3 bucket or modified IAM permissions, allowing for audit and rollback investigation.

- **Using Both Together:**

In a real-world application, you might use CloudWatch to monitor system health and auto-scale infrastructure, while using CloudTrail to audit all user activity and investigate incidents — giving both operational insight and security visibility.

3. Importance of a Well-Architected Framework

The AWS Well-Architected Framework is essential during a cloud migration because it ensures that critical elements like cost management, security, performance efficiency, and operational excellence are embedded into the design from the beginning, not added as afterthoughts.

- **Cost Optimization:**

Migrating systems without proper cost planning can lead to overspending. The framework encourages evaluating resource usage early, ensuring that only necessary services are provisioned and that cost-saving features like reserved instances, storage lifecycle rules, and scaling policies are used effectively.

Example: Migrating to AWS with S3 Intelligent-Tiering storage for infrequently accessed marketing files helps TECH-Start minimize storage costs automatically.

- **Security:**

A rushed migration can leave vulnerabilities open. The framework emphasizes setting up secure access control, encryption, monitoring, and auditing from the start.

Example: Implementing IAM roles instead of root access ensures that each TECH-Start team member has only the permissions they need post-migration.

- **Performance Efficiency:**

Migration isn't just about copying servers over—it's about improving performance. The framework guides teams to choose modern architectures (like serverless or scalable services) rather than simply replicating old designs.

Example: Instead of lifting and shifting a legacy app, TECH-Start could move to EC2 with Auto Scaling for dynamic traffic handling.

- **Operational Excellence:**

The framework stresses building operational visibility early by automating deployments, monitoring system health, and preparing for failure recovery.

Example: Setting up CloudWatch alarms during migration allows TECH-Start to immediately detect and respond to issues without manual checking.

- **Reliability:**

Migrations must be planned for resilience. The framework ensures that workloads are spread across Availability Zones and that backup and disaster recovery are addressed early.

Example: Hosting the TECH-Start website across two AZs with an Elastic Load Balancer ensures users won't experience downtime during an AZ outage.

Summary

Following the AWS Well-Architected Framework during migration reduces risk, controls costs, improves security, and sets a foundation for future scalability. It transforms a simple cloud lift into a strategic, resilient modernization for long-term success.