ABC is a fictional e-commerce business based in Dallas, TX, primarily serving US customers. Like many e-commerce platforms, ABC’s website stores sensitive customer information, including credit card data and billing addresses, and runs on Linux-based operating systems. As ABC seeks to improve its infrastructure, the company’s CEO and CTO are exploring cloud solutions with well-established providers like Google, Microsoft, and Amazon.

While this migration is focused mainly on moving their web servers to the cloud, ABC also has new goals: enabling in-depth data analytics to gain insights into customer behavior and trends, and integrating an email service to support targeted marketing and customer communication. The tech team is fully supportive of this move, ready to manage the migration and implement these new capabilities on a scalable, cloud-based platform.

# Project Overview

In this exercise, we’ll design a highly available, redundant e-commerce and analytics solution on AWS for ABC Company using Cloudcraft.



# Step-by-Step Solution Design

1. **Add User Access**
   * Start by adding a user to your design by dragging a user icon to the perimeter of your diagram.

# Elastic Compute Instance (EC2) - 5 Points

* + Add an EC2 instance with the following specifications:
    - **Platform:** Linux
    - **CPU:** 3.1 GHz Intel Skylake E5-2686 v5 (4 vCPUs)
    - **Memory:** 16 GiB
    - **Storage:** EBS-only with up to 2780 Mbps throughput
    - **Network Performance:** Up to 5 Gbps
  + *Tip:* T-type instances may match these specifications. Use this instance as the web server and connect it to the user with the arrow tool.
  + **Screenshot Required:** Show your EC2 setup connected to the user.

# Primary Database (RDS) - 5 Points

* + Add an Amazon RDS instance as the primary database with a MySǪL engine, selecting an M5 large instance type.
  + **Connection:** Link the EC2 instance to this RDS instance.
  + **List RDS Specifications:** Include details of the RDS instance.
  + **Screenshot Required:** Show your setup of EC2 connected to the RDS instance.

# Caching with ElastiCache (Redis) - 15 Points

* + Add an ElastiCache instance with the Redis engine, M5 large type.
  + **Virtual Private Cloud (VPC):** Create a new VPC named “Cached Database,” assigning the RDS and ElastiCache instances to it.
  + **Customer Gateway:** Add a customer gateway to enable secure access to and from the VPC.
  + **Connections:** Connect the EC2 instance to the customer gateway, the customer gateway to the RDS, and finally connect ElastiCache to the path between the gateway and the RDS.
  + **ElastiCache Explanation:** Provide a brief explanation of ElastiCache’s role and benefits.
  + **Screenshot Required:** Show your VPC and connection setup.

# Auto Scaling - 5 Points

* + Duplicate your EC2 instance, creating two additional instances, and add Auto Scaling to manage these.
  + **Connections:** Draw connections from the user to the Auto Scaling setup and from Auto Scaling to the customer gateway.
  + **Screenshot Required:** Show your Auto Scaling configuration with connections.

# Load Balancer for Auto Scaling - 10 Points

* + Add an Elastic Load Balancer (Classic Load Balancer, 10 GB data processing) to manage traffic to the Auto Scaling group.
  + **Connections:** Route user access through the load balancer to the Auto Scaling instances.
  + **Screenshot Required:** Display the load balancer setup.

# Replica VPC for Redundancy - 10 Points

* + Create a replica of the “Cached Database” VPC with a similar RDS and ElastiCache configuration. Name this new VPC “Replica.”
  + **Customer Gateway:** Add a gateway to “Replica” for access management.
  + **Traffic Management:** Add a second Classic Load Balancer (10 GB data) to manage traffic between “Cached Database” and “Replica.”
  + **Screenshot Required:** Show the Replica VPC setup with connections.

# Email Service Setup - 5 Points

* + Add an email-sending service that ABC can use with their domain.
  + **Design Placement:** Decide where this component should be in the architecture and explain your choice.
  + **Screenshot Required:** Show the email service in the design.

# Analytics with Redshift and Lambda - 15 Points

* + Add Amazon Redshift for data warehousing and a Lambda function for analytics processing.
  + **Redshift Overview:** Describe key capabilities of Redshift.
  + **Connections:** Link Redshift and Lambda to relevant components in your architecture.
  + **Screenshot Required:** Show the analytics setup in your design.



# Final Tasks

1. **Design Summary - 10 Points**
   * Write a brief explanation of each step you took in your design process, focusing on how each part contributes to operation, application, storage, reliability, and elasticity.

# Cost Analysis - 10 Points

* + Go to the Budget tab and calculate the annual cost for all components of your design for US-EAST-2 (Ohio).
  + **3-Year Term Evaluation:** For the US-EAST-2 region, change all services to a 3- year term, upfront payment. Describe any cost differences and explain why this option might be beneficial for ABC.

# Multi-Region Redundancy - 10 Points

* + Duplicate your entire design to create a redundant infrastructure in a secondary region (i.e., create a replica in US-WEST-1). Ensure the replicated infrastructure matches the primary region’s specifications for high availability.
  + **Connections:** Set up Route 53 for failover between the regions to manage traffic routing in case of a primary region failure.
  + Repeat Step 2 above (Cost Analysis) and compare the costs between a single site solution and your redundant solution.
  + Please provide a screenshot