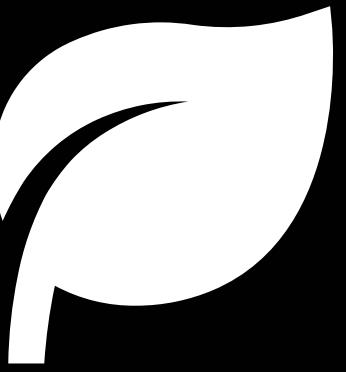


# Cloud Foundations

## Session 02

≈



Presented by Vansh Sood





From hackathons to the Startup Residency program,  
ShiftKey Labs makes sure students have the skills & support  
they need to pursue their most innovative ideas!



## HIGHLIGHTS

**NASA Space Apps  
Challenge**

**Global Game Jam**

**Generative AI  
Hackathon**

**Workshops & Talks**

REACT  
Blockchain  
Prototyping  
Startups  
Freelancing  
Cybersecurity

shiftkeylabs.ca

@ShiftKeyLabs

info@shiftkeylabs.ca

# About Me

---

- Third-year BCS, Researching in AI, TA at Dalhousie University
- Tech coordinator for Shiftkey Labs and Faculty of Health
- Freelancing for over 6 years
- Worked with 25+ startups and small businesses
- Helped startups raise over \$100,000 in funding
- Founder of a 6 figure web and mobile development agency
- Building a food tech startup Bean
- AWS Solutions Architect
- Google Code-In Grand Prize Winner (Global Top 50)



# Course Overview



Over these 4 weeks, you'll learn about cloud computing and AWS services.

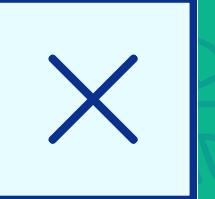
By the end of the course, you'll have the knowledge needed for the AWS Cloud Practitioner certification.

You will get knowledge to practice AWS and I highly recommend giving the AWS Certification and get Reimbursed by ShiftKey

Get ready for hands-on activities, real-world examples, and fun learning experiences!



# HEY DAL STUDENTS!

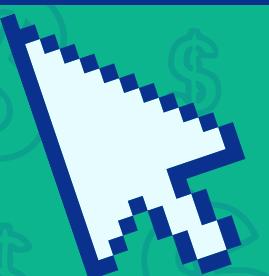


**ShiftKey's Reimbursement  
Program can put MONEY  
BACK IN YOUR POCKET!**

**SCAN HERE**



**Questions? Email Or Denemark at [Or@dal.ca](mailto:Or@dal.ca) or check our website for more**  
***\*This program is only eligible for Dalhousie students\****



# How is this going to help?

---

## Development Skills

---

Learn the necessary tools and techniques in cloud, specifically focused on AWS.

## Certification Completion

---

Walk away with a certificate you've completed from start to finish, demonstrating your ability to turn ideas into workable applications.

## Career Boost

---

Reliable, practical skills, and a polished certificate to showcase to potential employers, will be a highlight on your resume.

# What are you taking away from this?



Apart from the food? A whole lot of knowledge.

You can choose to just listen, or you can choose to actually work on an application that will teach you more than two development courses in university will.

Work on this for four days, and I guarantee you will have an amazing project with a whole lot of skills up your sleeve.

And even though I'm not a big fan of resumes, this will be an excellent addition to your resume.

# Getting the Certificate



What is absolutely necessary?

- Join 3/4 sessions
- Go home and practice!!!
- Give the final exam (date to be announced)
  - Can be replaced with AWS CCP
- Submit 1 min video solution (mock interview)

What can be done?

- Use Cloud regularly
- Give AWS CCP / Other Foundational exams
- Have fun

Which of the following best describes an AWS Region?

- A) A collection of multiple data centers located in the same city
- B) A single physical data center used by AWS customers
- C) A group of Availability Zones within a geographical area
- D) A single Availability Zone within AWS

Which of the following best describes an AWS Region?

- A) A collection of multiple data centers located in the same city
- B) A single physical data center used by AWS customers
- C) A group of Availability Zones within a geographical area
- D) A single Availability Zone within AWS

1. True or False? AWS allows customers to replicate data across multiple AWS Regions for increased resilience.
2. True or False? Each AWS Region operates completely independently of other Regions for fault isolation.
3. True or False? An AWS Availability Zone consists of one or more physical data centers.
4. True or False? AWS provides root-level access to its physical servers for customers.
5. True or False? The AWS Free Tier allows users to try certain services at no cost for 12 months.

1. True or False? AWS allows customers to replicate data across multiple AWS Regions for increased resilience.
2. True or False? Each AWS Region operates completely independently of other Regions for fault isolation.
3. True or False? An AWS Availability Zone consists of one or more physical data centers.
4. True or False? AWS provides root-level access to its physical servers for customers.
5. True or False? The AWS Free Tier allows users to try certain services at no cost for 12 months.

# Why Cloud

## Scalability

Easily scale resources up or down based on demand.

## Elasticity

Automatically adjust resources to match workload.

## Cost Efficiency

Pay for only the resources you use, reducing upfront costs.

## Global Reach

Access cloud services from anywhere in the world.

This is Bill.

Bill uses AWS.



He creates users with least privileges in IAM and enforces 2FA.

Be like Bill.

# How are we going to proceed?

---

## Type

---

We will discuss what types are available for every service provided by AWS, what each type's use case is and where each type gets used.

## Use Cases

---

What is the service's use cases and where it is generally used

## Pricing

---

What different pricing models are for each service, when to decide what pricing to go for and

# Compute

---



# Elastic Cloud Compute

---

## Use Cases:

- Web Hosting: Host websites or web apps with varying loads.
- Big Data Processing: EC2 is flexible and scalable for analytics workloads.
- Enterprise Applications: Suitable for running software like SAP, Oracle, etc.



# EC2

---

## Types of EC2 Instances:

- General Purpose: For balanced compute, memory, and networking (e.g., t3, m5).
- Compute Optimized: For high-performance processors (e.g., c5).
- Memory Optimized: For memory-intensive applications (e.g., r5).
- Storage Optimized: For large amounts of data storage and high I/O performance (e.g., i3).
- Accelerated Computing: For hardware acceleration (e.g., p3 for ML, inf1 for inference).
- High Performance Computing (HPC): For large-scale parallel processing (e.g., hpc6id).



# EC2

---

## Pricing:

- Pay-as-you-go: Pay only for the compute resources you use (on-demand).
- Reserved Instances: Save by committing to usage for 1 or 3 years.
- Spot Instances: Lower cost by bidding for unused capacity.
- Savings Plans: Flexible pricing model offering up to 72% savings for long-term commitment.

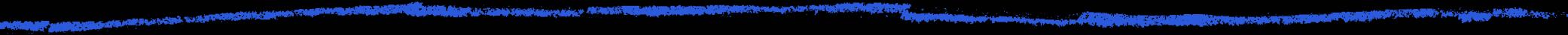


# AWS Auto Scaling



- Purpose: Automatically adjust the number of EC2 instances or containers based on demand.
- Use Cases:
  - Web Traffic Spikes: Scale EC2 instances for sudden increases in web traffic.
  - Cost Optimization: Automatically reduce the number of instances when demand decreases.
  - Background Tasks: Scale worker instances based on job queue lengths or CPU/memory usage.
- Pricing:
  - There are no charges for Auto Scaling itself; costs are based on the instances and resources it launches.

# Elastic Load Balancing (ELB)



**Purpose:** Distributes incoming traffic across multiple resources to ensure high availability and reliability.

## Use Cases:

- **High Availability:** Distribute traffic to EC2 instances across multiple AZs for fault tolerance.
- **Microservices:** Route traffic to specific containers or services.
- **SSL Termination:** Handle SSL/TLS encryption and decryption for secure communication.

## Pricing:

- Charged by the number of hours the load balancer is running and the amount of data processed.

# ELB Types

---

## Application Load Balancer (ALB)

Best for HTTP/HTTPS traffic, with routing based on content.

## Network Load Balancer (NLB)

Best for handling high-throughput and low-latency traffic at the network layer.

## Classic Load Balancer (CLB)

Older version; best for simple web applications.

# Serverless?

---

No server? No problem?



# Serverless?



## What is this?

- Cloud computing model where the cloud provider manages infrastructure.
- Developers focus on writing code without managing servers.
- Resources scale automatically based on demand.

## Key Features

- No Server Management: No provisioning, maintenance, or scaling of servers.
- Auto-Scaling: Scales up and down based on real-time usage.
- Pay-as-You-Go: Charges based on actual execution time and resources consumed.
- Event-Driven Execution: Functions trigger based on events (e.g., API calls, database updates).

# Serverless?



## Use Cases

- Web applications and APIs
- Event-driven automation
- Data processing pipelines
- IoT applications
- Chatbots and AI inference



### Pros:

- Lower operational cost
- High scalability
- Faster time to market



### Cons:

- Cold start latency
- Limited execution time
- Vendor lock-in risks

# AWS Lambda

---

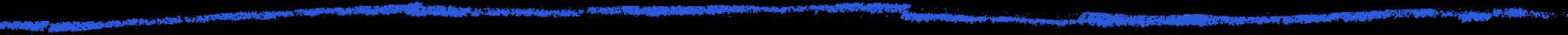
- Type: Serverless compute service for executing code in response to events.
- Use Cases:
  - Event-driven Applications: Trigger functions in response to events (e.g., changes in S3, DynamoDB).
  - Real-time File Processing: Automatically process files as soon as they are uploaded to S3.
  - API Backends: Use Lambda functions as lightweight APIs.
- Pricing:
  - Based on the number of requests and duration (execution time) of the function.
  - Free tier available: 1M free requests per month and 400,000 GB-seconds of compute time.

# AWS Fargate

---

- Type: Serverless compute engine for containers (works with ECS and EKS).
- Use Cases:
  - Microservices: Easily deploy and manage microservices architectures without managing servers.
  - Batch Jobs: Run background processes without worrying about the infrastructure.
  - Web Applications: Containerized apps running in isolated environments.
- Pricing:
  - Charged based on the requested vCPU and memory resources that are used during the execution of tasks.
  - No charge for idle resources, unlike EC2.

# Elastic Container Service



Type:

- Managed service for running Docker containers

When you hear the word container, you may associate it with Docker. Docker is a popular container runtime that simplifies the management of the entire operating system stack needed for container isolation, including networking and storage. Docker makes it easy to create, package, deploy, and run containers.

# ECS



## Use Cases:

- **Microservices Architecture:** Efficiently run containerized applications.
- **Batch Jobs:** Run periodic, event-driven, or batch processing tasks.
- **Web Applications:** Deploy containers to run dynamic web applications.

# ECS



## Pricing:

- Based on the underlying EC2 instances or Fargate (serverless) compute resources.
- Additional costs for data transfer, storage (EBS volumes), and other resources used.

# Amazon EKS (Elastic Kubernetes Service)

---

- Type: Managed Kubernetes service for container orchestration.
- Use Cases:
  - Containerized Applications: Run applications that require Kubernetes for orchestration.
  - Multi-cloud: Manage workloads across on-premises and AWS environments.
  - High Availability: Automatically scale applications across multiple availability zones.
- Pricing:
  - Control Plane: Fixed monthly fee per cluster.
  - Worker Nodes: Pricing based on EC2 instance types used for worker nodes.

# Questions?



# Databases



# Relational Database Service

---

- Type: Managed relational database service.
- Supported Engines:
  - MySQL, PostgreSQL, MariaDB, Oracle, SQL Server.
  - Amazon Aurora (also part of RDS).
- Pricing:
  - Pay-as-you-go based on instance size, storage, and data transfer.
  - Reserved instances offer cost savings for long-term usage.

# RDS

---

## Use Cases:

- Enterprise Applications: Run high-availability and scalable relational databases for enterprise apps.
- Data Warehousing: For structured data that requires complex queries.
- Web Applications: Backend databases for dynamic, high-traffic websites.

# Amazon Aurora

---

- Type: MySQL and PostgreSQL-compatible relational database engine designed for the cloud.
- Use Cases:
  - High Availability & Scalability: High performance with automatic scaling.
  - Global Applications: Applications requiring low-latency, cross-region replication.
  - Data-Intensive Applications: Suitable for applications with heavy read/write loads.
- Pricing:
  - Charged based on the instance size, storage consumed (in GB), and I/O requests.
  - Aurora Serverless: Scales automatically based on demand, charged by the second.

# Amazon DynamoDB

---

- Type: Managed NoSQL database service.
- Use Cases:
  - High Throughput, Low Latency: Ideal for applications requiring consistent low-latency reads and writes.
  - Mobile & Web Apps: Store and manage data for mobile apps, e-commerce, and social apps.
  - Serverless Applications: Works seamlessly with AWS Lambda for serverless applications.
- Pricing:
  - Provisioned Mode: Pay for the read and write capacity units.
  - On-demand Mode: Pay for data reads and writes without provisioning capacity.
  - Optional for data storage, backups, and DynamoDB Streams.

# Amazon ElastiCache

---

- Type: Managed in-memory caching service (supports Redis and Memcached).
- Use Cases:
  - Caching Layer: Improve application performance by caching frequently accessed data.
  - Session Storage: Store user session data in memory for faster access.
  - Real-Time Analytics: Cache data for fast, low-latency analytics.
- Pricing:
  - Based on the node type and the number of nodes used.
  - Additional costs for data transfer and backups.

# Amazon Redshift

---

- Type: Managed data warehouse service for running complex queries on large datasets.
- Use Cases:
  - Data Analytics: For large-scale data analysis, aggregation, and reporting.
  - Business Intelligence: For companies needing to analyze vast amounts of structured data quickly.
  - Data Warehousing: Ideal for consolidating data from different sources for analytics.
- Pricing:
  - Based on the type and number of nodes in the cluster.
  - Additional storage and backup charges.

# Bear with me

---

MY DAD SAYS  
THE CLOUD IS  
THE ANSWER TO  
EVERYTHING,  
MISS.



# Amazon Neptune

---

- Type: Managed graph database service for highly connected datasets.
- Use Cases:
  - Social Networks: Model relationships between users and content.
  - Recommendation Engines: Build systems that recommend products, services, or content based on connections.
  - Fraud Detection: Analyze relationships and connections in financial transactions.
- Pricing:
  - Based on instance type, storage usage, and I/O operations.

# Amazon DocumentDB

---

- Type: Managed document database service that is compatible with MongoDB.
- Use Cases:
  - Content Management Systems: Manage flexible, schema-less data.
  - Mobile Apps: Store JSON-like documents for easy retrieval and updates.
  - Real-Time Data: Ideal for use cases that require flexible, rapid access to data.
- Pricing:
  - Based on instance type, storage, I/O requests, and data transfer.

# AWS Database Migration Service (DMS)

---

- Type: Service to migrate databases to AWS securely and with minimal downtime.
- Use Cases:
  - Database Migration: Move databases to AWS with minimal downtime.
  - Database Replication: Replicate data for disaster recovery or for running a hybrid database setup.
  - Data Integration: Integrate data between on-premises and AWS databases.
- Pricing:
  - Charged based on replication instances used and data transferred.
  - Charges for the storage of logs and backups.

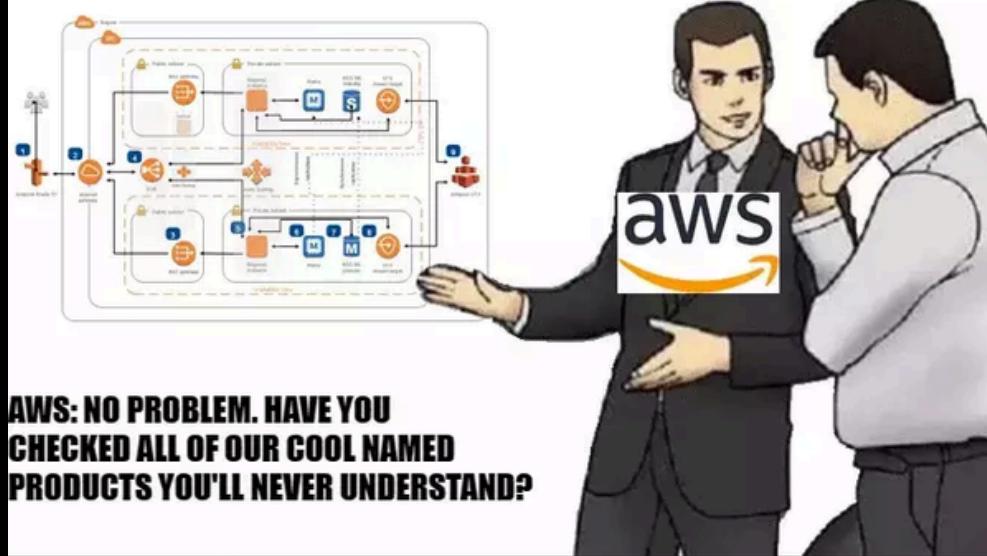
# AWS Schema Conversion Tool (SCT)

---

- Type: Tool to convert database schemas from one database engine to another.
- Use Cases:
  - Schema Conversion: Convert from one database engine (e.g., Oracle) to another (e.g., MySQL).
  - Database Modernization: Migrate legacy database architectures to modern cloud-based services.
  - Database Consolidation: Combine multiple databases into a single, scalable platform.
- Pricing:
  - AWS SCT itself is free, but there may be costs for the underlying migration services or resources used.

# Questions?

**ME: I JUST NEED TO HOST  
'HELLO WORLD' ON THE CLOUD.**



# EC2-Hosted Databases vs. AWS Managed Databases

---

- You manually manage the database, software patches, backups, and scaling. Best for specific configurations or when using unsupported database engines.
- Use Case: Custom database configurations, legacy database applications, and full control over database settings.
- Managed services like RDS, DynamoDB, and Aurora handle maintenance, backups, scaling, and security. Ideal for scalability, availability, and ease of use.
- Use Case: When you prefer less management overhead, need high availability, and scalability.

# Storage

Remind me to take a break here



# Amazon S3 (Simple Storage Service)

---

- Type: Object storage service.
- Use Cases:
  - Data Archiving: Store data for long-term retention and easy retrieval.
  - Backup and Restore: Store backups of data for disaster recovery.
  - Big Data Storage: Store large volumes of unstructured data like logs, images, and videos.
- Pricing:
  - Based on storage used, number of requests (PUT, GET, etc.), and data transfer.
  - Different pricing tiers for different storage classes (e.g., Standard, Intelligent-Tiering, Glacier).

# S3

---

- Storage Classes:
  - Standard: Frequently accessed data.
  - Intelligent-Tiering: Automatically moves data between two access tiers based on changing access patterns.
  - Standard-IA (Infrequent Access): For data that is accessed less frequently.
  - One Zone-IA: Lower-cost storage for infrequently accessed data that doesn't need multi-AZ resilience.
  - Glacier: Low-cost archival storage with retrieval times from minutes to hours.
  - Glacier Deep Archive: The lowest-cost storage for long-term data archiving.

# The Lifecycle

---

- Type: S3 feature for managing the lifecycle of objects in a bucket.
- Use Cases:
  - Archiving Data: Automatically transition objects to lower-cost storage classes (e.g., Glacier) after a specific period.
  - Data Retention: Automatically delete data after a defined period, useful for regulatory compliance.
  - Cost Optimization: Use lifecycle policies to move infrequently accessed data to cheaper storage classes.
- Pricing:
  - There is no additional cost for lifecycle policies, but transitions between storage classes may incur data transfer costs.

# Amazon Elastic Block Store (Amazon EBS)

---

- Type: Block storage service.
- Use Cases:
  - Persistent Storage for EC2 Instances: Provides storage volumes that can be attached to EC2 instances for storing OS, applications, and data.
  - Database Storage: Store database files and transaction logs with high performance.
  - Disaster Recovery: Store critical data that needs to be recovered in case of failure.
- Pricing:
  - Based on the volume size, provisioned IOPS (for high-performance use cases), and the snapshot storage.
  - Charges for data transferred and the provisioned throughput.

# Instance Store

---

- Type: Temporary block storage.
- Use Cases:
  - Ephemeral Data: Store temporary data, such as caching, buffer storage, and scratch data.
  - High-Performance Use Cases: For workloads that need very low-latency and high-throughput storage (e.g., batch processing, temporary file systems).
- Pricing:
  - Free, but only available on specific EC2 instance types.
  - Data is lost if the instance is stopped or terminated.

# Amazon Elastic File System (Amazon EFS)

---

- Type: Managed file storage service that is scalable.
- Use Cases:
  - Shared File Storage: Multiple EC2 instances can access the same file system concurrently.
  - Content Management: Store shared files that need to be accessed by multiple users or applications.
  - Lift-and-Shift Applications: Migrate on-premises applications that require file storage.
- Pricing:
  - Pay-as-you-go based on the amount of storage consumed.
  - Pricing tiers for Standard (low latency) and Infrequent Access (lower-cost, infrequent access data).

# Amazon FSx

---

- Type: Managed file storage for Windows (FSx for Windows File Server) or Lustre (FSx for Lustre).
- Use Cases:
  - Windows File Storage: FSx for Windows File Server provides fully managed Windows-based file storage with support for the SMB protocol.
  - High-Performance Computing: FSx for Lustre is designed for compute-heavy workloads, providing high-speed storage for data-intensive applications.
- Pricing:
  - Based on the type of file system (Windows or Lustre), storage consumed, and throughput performance.

# AWS Storage Gateway

---

- Type: Hybrid cloud storage service that integrates on-premises environments with AWS storage services.
- Use Cases:
  - Backup and Archiving: Store backups from on-premises applications directly to AWS cloud.
  - Disaster Recovery: Create backups and disaster recovery solutions for on-premises data.
  - Cloud Storage Integration: Use AWS storage for on-premises workloads without changing the architecture.
- Pricing:
  - Charges are based on the type of gateway (e.g., file, volume, tape), and the amount of data transferred to the cloud.

# AWS Backup

---

- Type: Managed backup service to automate and centralize data backup across AWS services.
- Use Cases:
  - Automated Backups: Schedule and automate backups for EC2 instances, RDS, EFS, DynamoDB, and other AWS services.
  - Regulatory Compliance: Ensure that backup data is stored securely and complies with industry standards.
  - Disaster Recovery: Centralized backup management to recover from outages or accidental data loss.
- Pricing:
  - Based on the amount of data backed up and stored, as well as the backup frequency.

# Lets play



# Thank you



Reach out to me:

<https://www.linkedin.com/in/vanshsood/>