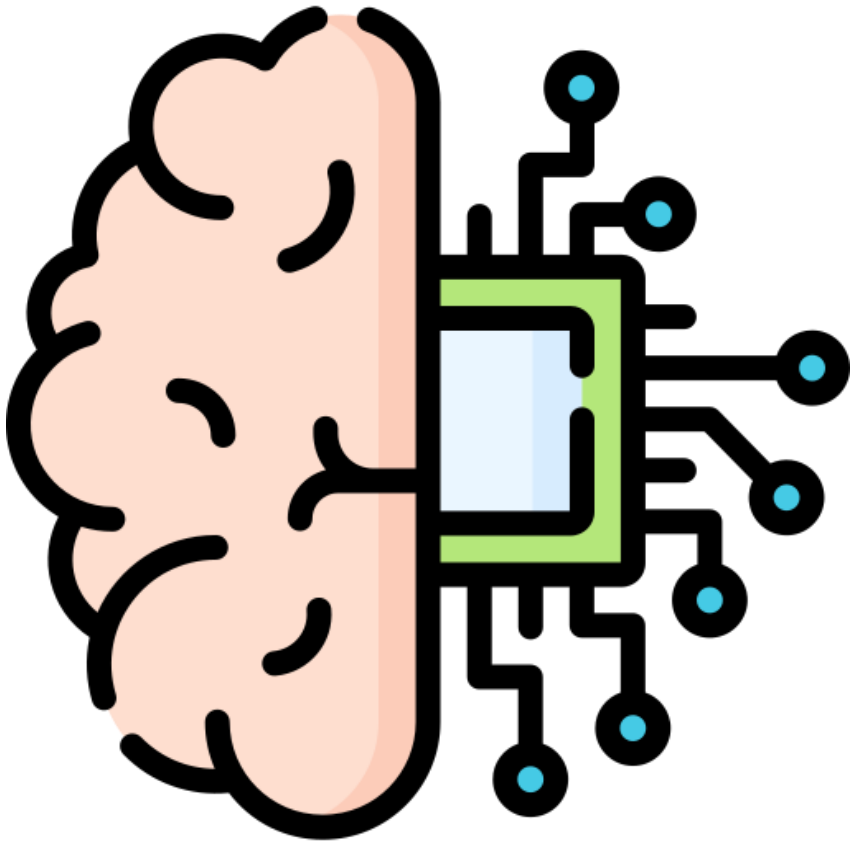
4-weeks Bootcamp

Introducing

Amazon Machine Learning



Instructor : THET SU WIN

On Boarding

Course Schedules

Details Schedules

No	Week	Course			
		Course Titles	Theory	Practical Labs	Discussion
1	Week 1 02-aug-2025	Intro to Cloud Computing	AWS services overview for ML (S3, EC2, IAM) ML lifecycle on AWS	AWS Free Tier setup (IAM, EC2, S3) Hands-on: launching EC2 , Setup IAM and creating S3 bucket, and usage of boto3	Details in Lecture Slides and Assignments
2	Week 2	Containerization & Serverless ML Pipelines	Intro to Docker + AWS ECR (Elastic Container Registry) Serverless Concepts	Build a simple ML inference container using Lambda, API gateway	Details in Lecture Slides and Assignments
3	Week 3	Big Data Processing with PySpark & EMR	What is Big Data? PySpark concepts and EMR overview	Local Spark Setup and data preprocessing using pyspark	Details in Lecture Slides and Assignments
4	Week 4	Model Development with SageMaker	Amazon SageMaker intro Built-in algorithms & Estimators & Inferencing	Tabular model training with built-in XGBoost , Creating the inferencing pipeline	Details in Lecture Slides and Assignments
5	Optional Lecture	Creating the RAG Application using AWS Services Creating the Orchestration pipelines of model training & Inferencing using AWS Stepfunction			

Declaimer : This 4-week bootcamp will not cover to understand 100% workflow of machine learning using AWS, but will have a sense of which AWS services are using and how to use it as a Data Scientist or Machine Learning Engineer.

On Boarding

Weekly Course Schedules

Overall Schedules

09/06	09/13	09/20	09/27	10/04
On Boarding	Week 1	Week 2	Week 3	Week 4 & Closing

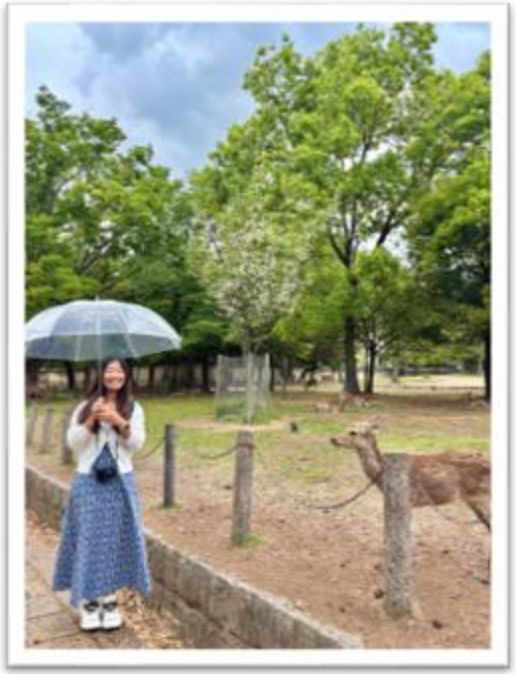
Course References

Udemy: AWS Machine Learning Specialty by Frak Kane and Stephane Maarek
AWS Documentation
Detailed citations and links will be provided in the lecture slides for each module.

Weekly Schedules

Sunday	Happy Weekend !!! (Nothing to do on this day)
Monday	New theory content and hands-on materials are released at 9:00PM (MMT)
Monday – Wednesday	study the theory and complete the case study assignment.
Thursday	Submit the case study for instructor feedback.
Friday	Instructor prepares for feedback
Saturday	Includes hands-on, discussion, feedback, and Q&A.

GET TO KNOW ME 👋



Hi, I'm Thet Su Win (Your Instructor for this bootcamp)

- **Data Scientist** at a cybersecurity company in Singapore
- **Master's Student** at School of Computing (AI Specialization), National University of Singapore
- **Instructor** at DATAVERSITY Myanmar, where I taught Python, AI, and Data Science
- Got AWS Machine Learning Specialty Certificate from AWS in 2023
- Over **5 years of experience** in AI & Data Science, with **7+ years** in the software industry
- Occasionally share content on **Medium** about AI and tech
- Passionate about **teaching**, especially to kids, and enjoy **traveling, reading**, and watching dramas
- **Fun Fact:** I left SageMaker running for 2 days and burned around **SGD 8,000**. (And yes, *I still somehow kept my job.)*

Connect with me

https://www.linkedin.com/in/thet-su-win-169221172/

Medium

https://medium.com/@thetsuwin.tsw6

I'M HERE TO GUIDE, BUT ALSO TO LEARN WITH YOU — LET'S GROW TOGETHER AND MAKE THE MOST OF THESE 4 WEEKS!!

On Boarding

What You'll need for this Bootcamp

1. AWS Account (Required)

- Please sign up at aws.amazon.com if you don't have one
- Free Tier is enough for most of our hands-on exercises
- Make sure you can access **SageMaker, S3, Lambda, and IAM**

2. Basic Tools

- A laptop or desktop with **internet access**
- A modern browser (Chrome, Firefox, or Edge recommended)
- A GitHub account (optional, but useful for code sharing)
- Visual Studio Code or any code editor (optional for advanced practice)

3. Optional, but Helpful

- AWS CLI installed
- Python 3.x installed locally
- An IDE with terminal access (like VS Code or JupyterLab)

Things to Be Aware of

1. AWS is not 100% FREE

- While AWS **Free Tier** covers many services (S3, SageMaker Studio, Lambda), some actions (e.g., large training jobs, real-time endpoints) can incur **charges**
- We'll **stay within the Free Tier** in this bootcamp — but always check your usage on the **Billing Dashboard**

2. Don't share your AWS credentials

- Never expose or hardcode your **Access Keys or Secrets** in code
- If using notebooks, **use IAM roles or environment variables** to authenticate

3. Clean Up Resources After Use

- Some services like **SageMaker Notebook Instances** and **EC2** keep running and **incur cost** even when idle
- Always **stop or delete** your resources after each session
- I'll remind you at the end of every hands-on lab

4. Region Matters

- Always use the same **AWS Region** (e.g., us-east-1) to avoid confusion
- Services and pricing vary by region — we'll stick to one region throughout

5. Use Bootcamp Resources ONLY for Learning

- Please don't use the AWS account for unrelated production tasks or heavy workloads
- This is for educational purposes only

I trust you'll use this access responsibly so everyone can benefit — let's make this smooth for all of us 

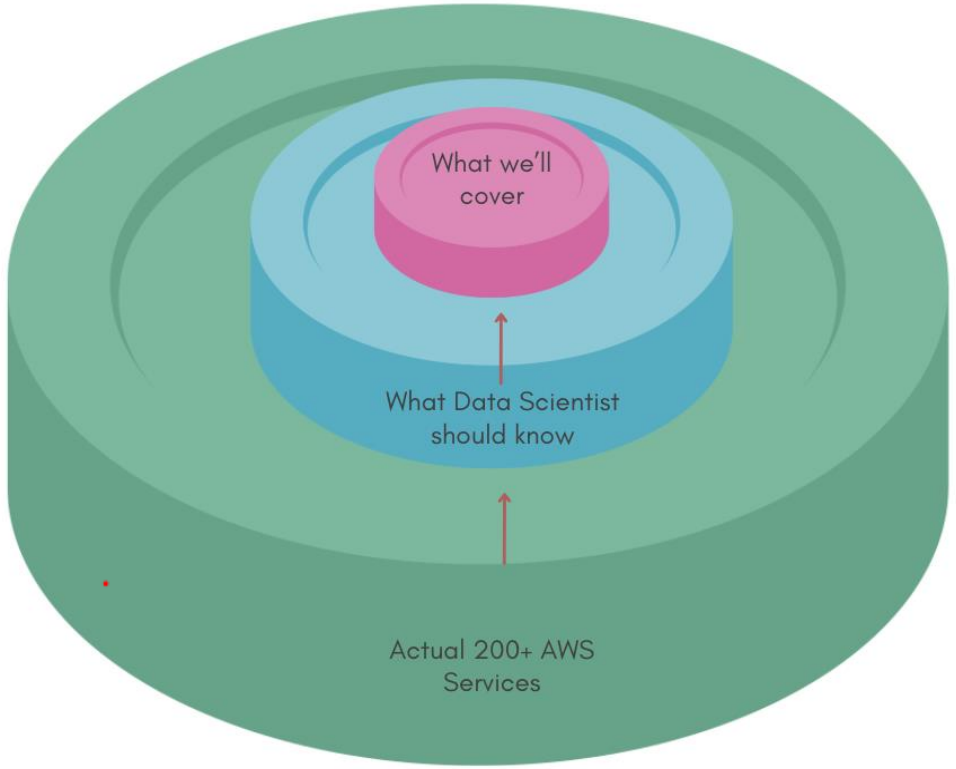
Let’s Dive In! 🚀

Ready to explore AWS Machine Learning?

What is Amazon Web Services (AWS) ?

AWS offers over **200+ services** across:

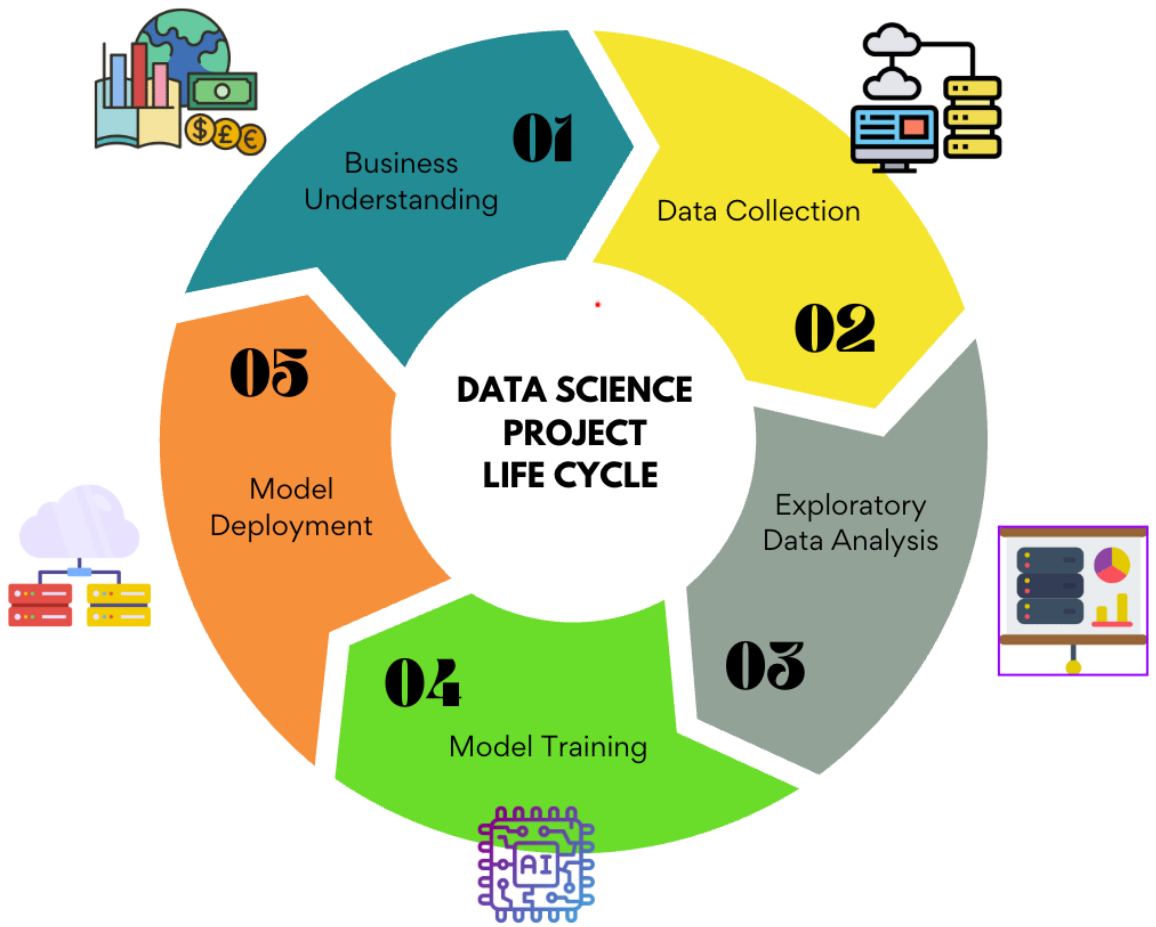
- Compute
- Storage
- Machine Learning
- Analytics
- Security
- IoT and more...



How many AWS services can you name off the top of your head?

(Drop it in the chat or shout it out!)

Data Science/Machine Learning Project Lifecycle



1. Business Understanding: Define the client's problem, background, use cases, and target users.

2. Data Collection: Gather data (in-house or through crawling), clean, sample, and possibly manually label it.

3. Exploratory Data Analysis (EDA): Analyze data quality, identify noise/outliers, and ensure suitability for training ("Garbage in, Garbage out").

4. Model Training: Select algorithms, train the model, and refine performance through **hyperparameter tuning**. If accuracy is low, revisit earlier stages for data or requirement issues.

5. Model Deployment: Release the finalized model to production (on-premise or cloud) and consider incremental training for new data.

https://medium.com/@thetsuwin.tsw6/data-science-machine-learning-project-life-cycle-8ec9e17d9d07

Data Science/Machine Learning Project Lifecycle *in AWS Environment*

	Life Cycle	AWS Services
1	Data Collection	Amazon S3, Amazon GLUE, Amazon Kinesis, Amazon IoT core
2	Data Processing	Amazon SageMaker, Amazon GLUE, Amazon Athena, Amazon EMR, Amazon RedShift
3	Exploratory Data Analysis	Amazon SageMaker, Amazon Athena, Amazon EMR, amazon QuickSight
4	Model Training & Model Evaluation	Amazon SageMaker, Amazon EC2, Amazon EMR, Amazon EKS
5	Model Deployment	Amazon SageMaker, Amazon API Gateway, AWS Lambda, Amazon EKS, Amazon StepFunction

In this course, we'll cover

- Amazon S3
- Amazon SageMaker
- Amazon EC2
- Amazon EMR (partially)
- Amazon Lambda
- Amazon API Gateway
- Amazon StepFunction

Due to some cost limitation.

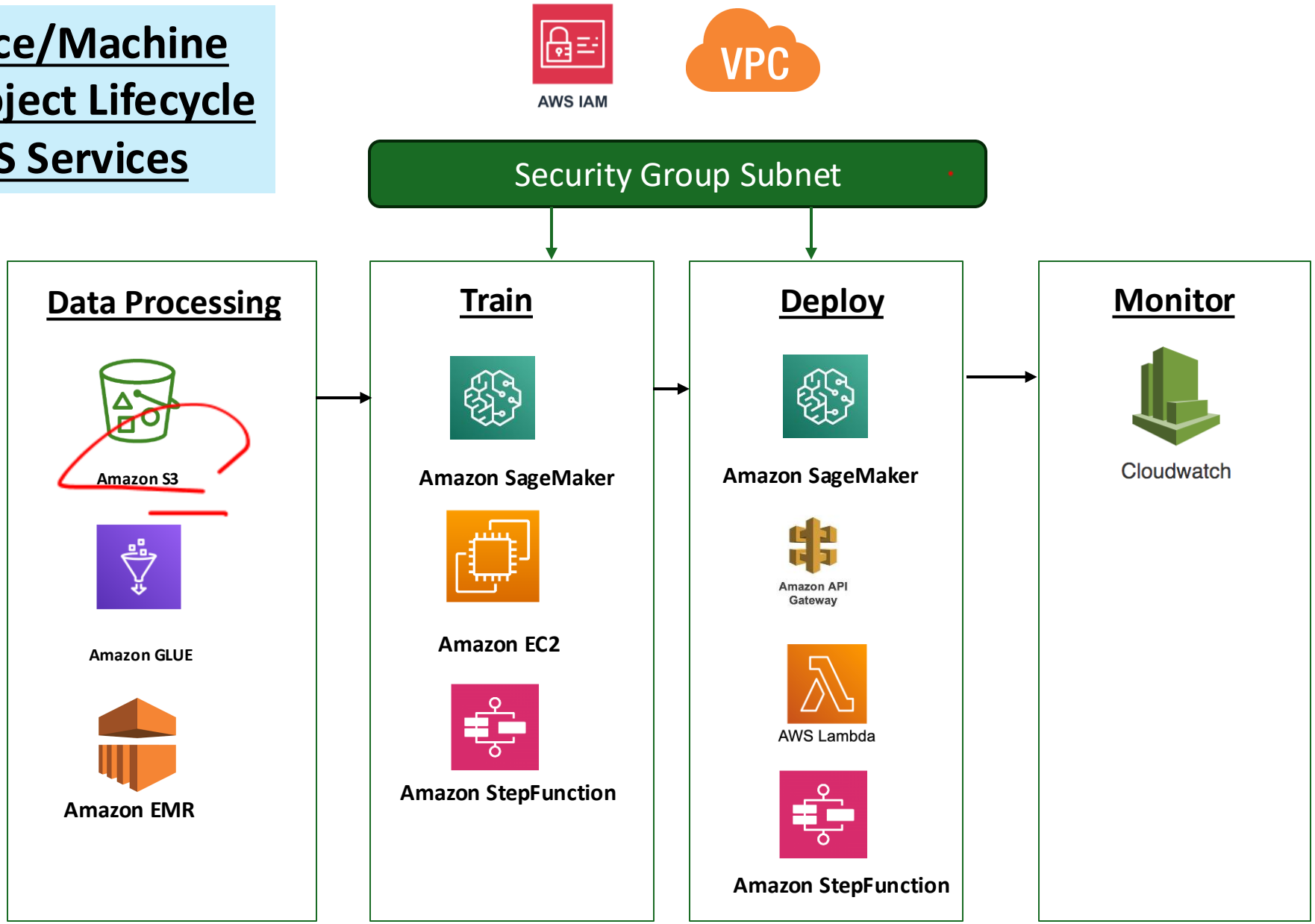
Don't worry if you didn't know the services, you *will* by the end of this bootcamp. For now, just start thinking like a Data Scientist using AWS.

Additionally, you should know

Additional AWS Services

1	Security and Access Management
2	Networking & Infrastructure
3	Workflow and Automation
4	Other Useful Tools

Data Science/Machine Learning Project Lifecycle with AWS Services



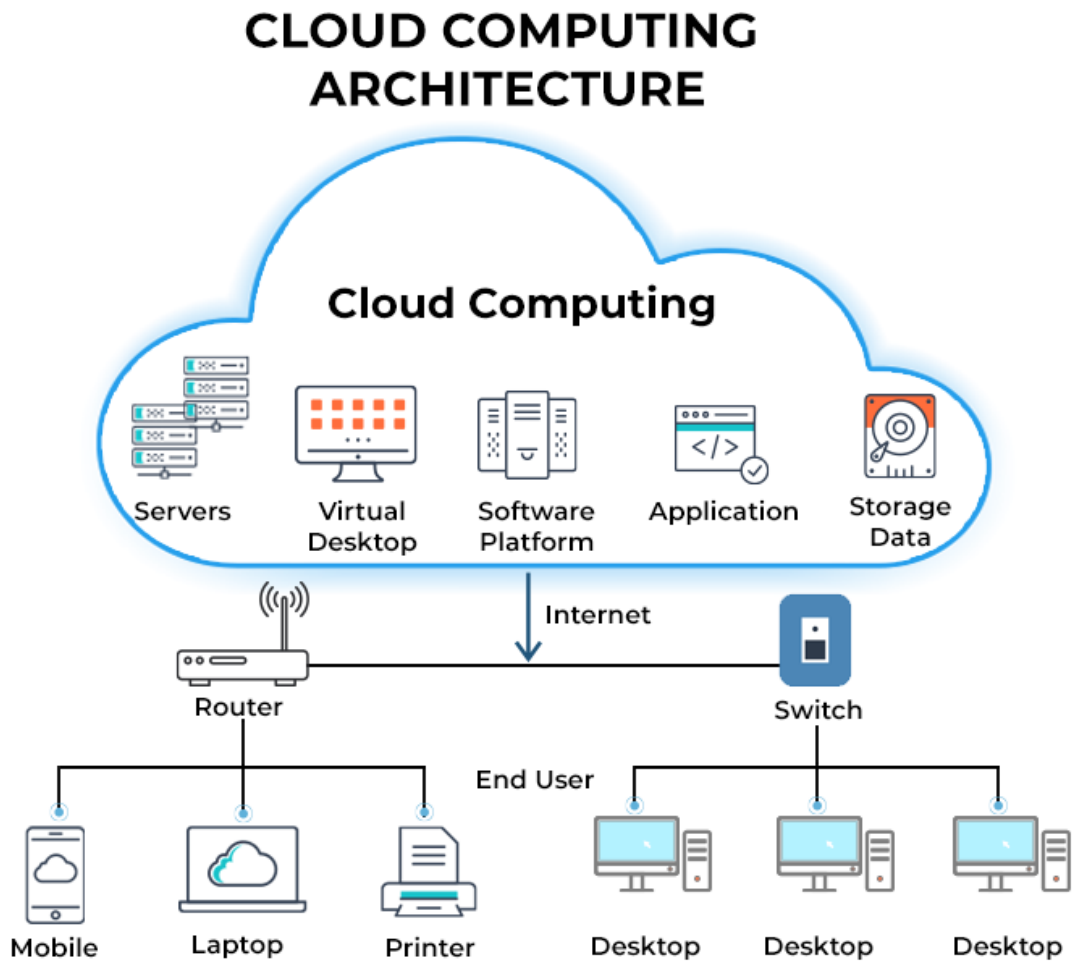
WEEK 1 : Intro to Cloud Computing

Course Schedules

Details Schedules

No	Week	Course			
		Course Titles	Theory	Practical Labs	Discussion
1	Week 1 06-Sep-2025	Intro to Cloud Computing	AWS services overview for ML (S3, EC2, IAM) ML lifecycle on AWS	AWS Free Tier setup (IAM, EC2, S3) Hands-on: launching EC2 Jupyter setup on EC2	
2	Week 2	Containerization & Serverless ML Pipelines	Intro to Docker + AWS ECR (Elastic Container Registry) Serverless Concepts	Build a simple ML inference container using Lambda, API gateway	
3	Week 3	Big Data Processing with PySpark & EMR	What is Big Data? PySpark concepts and EMR overview	Local Spark Setup and data preprocessing using pyspark	
4	Week 4	Model Development with SageMaker	Amazon SageMaker intro Built-in algorithms & Estimators	Tabular model training with built-in XGBoost Model tuning and logging	

Why Cloud Computing ?



Resource Limitation

1. If we try to train a large deep learning model on a personal laptop, what computational bottlenecks will we face?

Cost

2. Suppose training is not frequent; maybe once a month. From an economic standpoint, is it efficient to invest in expensive hardware, or is a rental model better?

Scalability

3. Now imagine an application experiences a sudden 100x increase in users. How can we design infrastructure to elastically adjust to demand instead of collapsing?

Compliance

4. In certain industries, data must remain in specific countries due to regulatory requirements. How can infrastructure adapt to such constraints?

These four problems ; compute bottlenecks, cost, scaling, compliance — are exactly what cloud computing was invented to solve

1. What is Cloud Computing ?

""Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centres and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS)."" ***FROM AWS***

IN SHORT

- *"Cloud computing = on-demand delivery of IT resources via the internet with pay-as-you-go pricing."*
- Keywords => **on-demand**, **pay-as-you-go**, **scalable**, **global**.

resources when you need them

you rent resources as you go instead of buying them

scale up or down quickly

access services from data centers worldwide

Ref : https://aws.amazon.com/what-is-cloud-computing/

Types of Cloud Computing

The three main types of cloud computing -

On Premises
You own and manage everything: servers, storage, networking, OS, applications

IaaS (Infrastructure as a Service)
You rent raw infrastructure (compute, storage, networking). You install OS, frameworks, apps.

PaaS (Platform as a service)
You rent a managed environment — provider handles OS, runtime, scaling. You just deploy code or models.

SaaS (Software as a Service)
You use the software directly via the internet. No need to manage infra, runtime, or code.

Every day life examples -

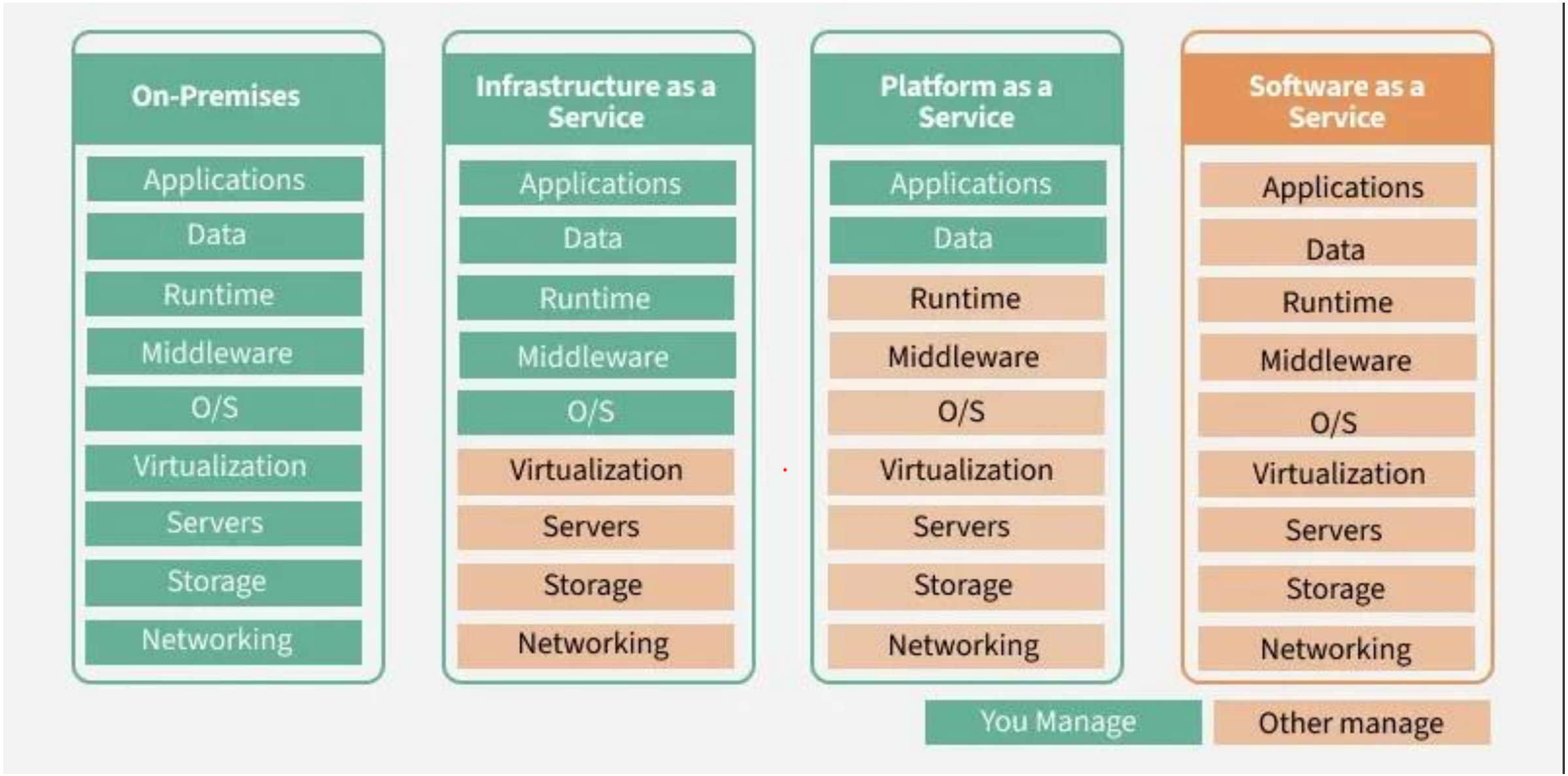
Cooking at home from scratch
- You buy your own stove, fridge, ingredients.
- You cook, serve, and clean everything.
- *Full control, full responsibility.*

Renting an empty kitchen
- The landlord gives you the kitchen space (electricity, water)
- You bring your own oven recipes and cook your own meal.
- *You control the cooking, but someone else owns the building*

Using a ready-to-use kitchen (with tools & oven)
- The kitchen comes with oven, mixer, utensils.
- You just bring ingredients and bake your cake.
- *You focus on the recipe, not setting up the kitchen.*

Ordering food via GrabFood / Foodpanda
- You just order, food arrives ready to eat.
- No cooking, no cleaning.
- *Everything is handled for you. you only consume the service.*

Each type of cloud computing provides different levels of control, flexibility, and management so that you can select the right set of services for your needs.



Ref : https://www.geeksforgeeks.org/software-engineering/difference-between-iaas-paas-and-saas/

Now that you get the food analogy… what do you think the equivalent examples would be for **Machine Learning**?

Machine Learning Scenario

On Premises | **IaaS (Infrastructure as a Service)** | **PaaS (Platform as a service)** | **SaaS (Software as a Service)**

- *You buy your own GPU servers, install TensorFlow, set up networking, manage everything yourself.*
- *You rent AWS EC2 GPU instances and use S3 for storage. You install ML libraries yourself and run training.*
- *You use AWS SageMaker — AWS gives you Jupyter notebooks, pre-installed ML frameworks, and managed training jobs. You focus only on the model and data.*
- *You call AWS Rekognition API for image classification, or AWS Translate for language translation. You don't train anything; you just consume the ML service.*

No	Types of Cloud Computing	AWS Services
1	**On Premises**	-
2	**IaaS**	EC2 (Compute), S3 (Storage), IAM (Security)
3	**PaaS**	SageMaker (end to end ML platform), AWS Glue (ETL Preprocesing)
4	**SaaS**	Rekoginition (image/video analysis), Comprehend (NLP), Translate, Polly (tts)

DISCUSSION 1: Cloud Strategy

You are building a real-time fraud detection system for a mobile payment app serving **500,000 monthly users** across Southeast Asia. The app requires **low-latency predictions (<300ms)** and needs to comply with **local data residency laws** (e.g., in Singapore and Indonesia).

Would you choose a **cloud-based ML pipeline** or an **on-premise setup**? Justify your decision with respect to **latency, security, DevOps complexity, cost, and data compliance**.

My Response (Cloud is Preferable)

Latency

- **<300ms prediction latency** is realistic using AWS services like **Lambda**, **SageMaker endpoints**, or even **Edge deployment (SageMaker Neo)**.
- AWS supports regional endpoint deployment (e.g., Singapore, Jakarta) to minimize network roundtrip delays.
- On-prem latency could be slightly lower, but managing low-latency infrastructure in multiple countries is operationally painful.

Security & Data Compliance

- Local data residency laws can be met using **AWS region-specific S3 buckets**, **VPC controls**, and **KMS** for encryption.
- IAM roles and logging help maintain strong access control and auditing.
- On-prem would offer tighter control but would **increase complexity and staffing needs**, especially across multiple jurisdictions.

DevOps & Maintenance

- In the cloud, I can automate model deployment and versioning with **CI/CD pipelines (e.g., CodePipeline + Lambda + S3)**.
- Scaling to 500K users becomes trivial with **autoscaling EC2 or Lambda**, which would be much harder on-prem without large DevOps investment.

Cost

- **Cloud is more cost-efficient** for variable workloads. I can use **spot instances for training** and **serverless inference** to reduce idle cost.
- On-prem would require purchasing hardware upfront for peak capacity — expensive and inefficient for a mid-scale use case.

A **cloud-first strategy with regional AWS configurations** gives the best balance of **performance, security, and compliance** for this specific use case. On-prem might be justified for *highly classified or ultra-low latency* systems, but not here.

DISCUSSION 2: Cloud Strategy

You're working for a **national healthcare agency** building a long-term ML system for **medical image classification (e.g., tumor detection)**. The system processes **sensitive MRI/CT scan data** from multiple hospitals, must operate in a **secure and air-gapped environment**, and is expected to run continuously for **at least 5 year**.

Due to regulatory policies, **patient data cannot leave the physical infrastructure**. Would you choose a **cloud-based setup** or an **on-premise architecture**? Justify your choice.

My Response (On Prem is Preferable)

Data Privacy & Compliance

- Health data is **extremely sensitive**. The scenario explicitly states that **data cannot leave the infrastructure**, which makes most public cloud options (even regionally hosted ones) **non-compliant**.
- AWS and other cloud providers offer HIPAA-compliant services, but **regulatory approval** for handling medical images often requires **physical air-gapped systems** in national infrastructure.

Security Control

- On-prem allows **total control over network traffic**, **access permissions**, and **hardware-based encryption**.
- There's no reliance on third-party APIs or internet access; important for **zero-trust or classified healthcare systems**.

Cost Predictability

- For a **5-year deployment**, on-prem hardware amortized over time could be **cheaper** than cloud usage (especially for **GPU-heavy inference or training workloads** like medical imaging).
- Cloud's pay-as-you-go becomes **less attractive** when usage is **steady and high-volume** for years.

Integration with Hospital Infrastructure

- Hospitals may already use **local PACS servers**, proprietary image formats, and specific network configurations that integrate better with an on-prem setup.
- On-prem simplifies integration and avoids costly data transfer or translation layers required in cloud.

Risk Tolerance

- In critical systems (like national health), **dependency on external vendors** for model hosting, service uptime, or API changes may be unacceptable.
- On-prem gives full-stack control and avoids vendor lock-in.

For sensitive, regulated, long-term medical ML systems, on-premise deployment offers compliance, security, control, and cost stability that cloud infrastructure cannot match.

DISCUSSION 3: Cloud Strategy

Scenario-based Question:

You're building an ML pipeline to **detect phishing websites** by taking screenshots of URLs and comparing them to known company logos and UI layouts (e.g., fake login pages).
Each day, the system ingests **thousands of URLs**, renders them (headless browser), takes a screenshot, and passes it through a **CV model (e.g., ResNet or ViT)** to assess visual similarity.

You're deciding whether to run this **on-prem** or in the **cloud**.

My Response (On Prem is Preferable)

Scalability

- Screenshot rendering + image inference is compute-intensive.
- Cloud (e.g., AWS Lambda + ECS + SageMaker) scales up quickly for bursty, parallel inference.

URL Access and Threat Intel

- Since you're scraping **live URLs**, running in the cloud may avoid local network exposure.
- You can route traffic through **VPC NAT gateways**, restrict domains, and avoid hitting the internal corp network.

Pay-per-user Cost Model

If you're only running inference during peak attack windows or using event-driven triggers (e.g., AWS EventBridge), cloud billing stays low.

My Response (On Prem is Preferable)