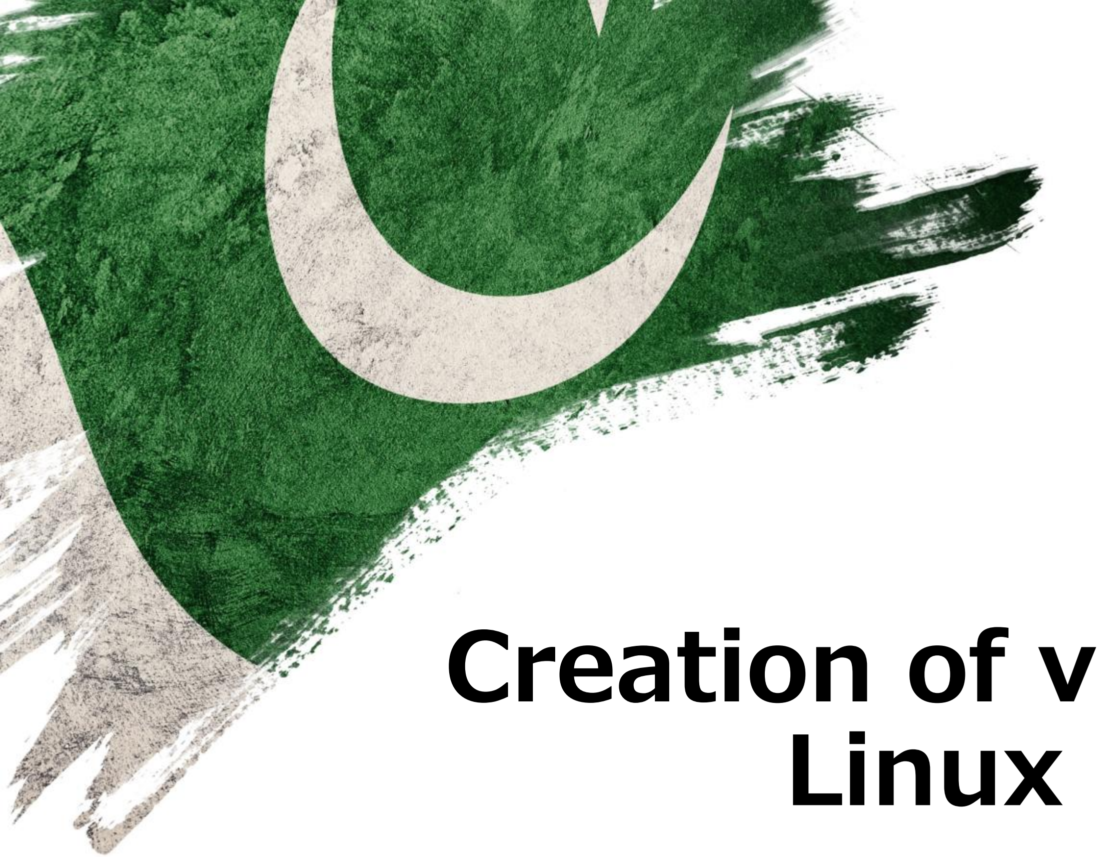

Creation of virtual server include Linux OS using AWS

Class 13

19/4/2025

Acknowledgement

The series of the IT & Japanese language course is Supported by AOTS and OEC.

Ministry of Economy, Trade and Industry

Overseas Employment Corporation

What you have Learnt Last Week

We were focused on following points.

- Usage of control and loop flow statement
- Performing Linear Algebra in Numpy
- Inspecting and Understanding Data
- Why Requirement Analysis is so important in the process?
- Requirement analysis practices
- Machine Learning algorithms
- Software development Life cycle
- Gitflow, Waterfall and agile methodologies
- Importance of Security compliance

What you will Learn Today

We will focus on following points.

- Introduction to AWS and EC2
- Comparison of popular Linux distributions available on AWS (e.g., Ubuntu, CentOS, Amazon Linux).
- Step-by-step guide to creating an AWS account. Understanding the AWS Free Tier and its benefits for new users
- Detailed steps for launching a new EC2 instance.
- Creating and managing key pairs for secure SSH access to your Linux server
- Quiz
- Q&A Session

Introduction to AWS: Powering the Cloud World

What is AWS & Its Core Service Offerings

AWS (Amazon Web Services) is a comprehensive cloud computing platform by Amazon Offers IaaS, PaaS, and SaaS-based solutions

Key service categories:

✅ Compute (e.g., EC2, Lambda)
✅ Storage (e.g., S3, EBS)
✅ Database (e.g., RDS, DynamoDB)
✅ Networking (e.g., VPC, Route 53)
✅ Machine Learning, Security, Analytics, DevOps tools

Example:

A startup launches a web app using EC2 for compute, S3 for media storage, and RDS for database needs.

Understanding AWS Global Infrastructure

Regions, Availability Zones, and Edge Locations

Region: A geographical area (e.g., US-East, Asia-Pacific)

Availability Zone (AZ): Multiple isolated data centers within a region

Edge Locations: Content delivery points for services like CloudFront (CDN)

Benefits:

✅ High availability & fault tolerance
✅ Low-latency global access
✅ Disaster recovery & backup options

Example:

A company deploys a website in the Singapore region with two Availability Zones for high availability, using Edge Locations in India for faster user access.

EC2: Scalable Virtual Servers in the Cloud

What is EC2 and Why Use It?

EC2 (Elastic Compute Cloud): Virtual machines for running applications on demand

Fully customizable: OS, CPU, memory, storage

Supports autoscaling and load balancing

Use cases:

✅ Web hosting
✅ Application backend
✅ Data processing
✅ Game servers
✅ Dev/test environments

Example:

A SaaS company hosts its backend on **EC2 Linux instances**, auto-scaling during traffic spikes.

EC2 Pricing Models Explained

Choose Based on Usage Pattern and Budget

On-Demand: Pay per hour/second, no commitment
✅ Good for short-term or unpredictable workloads

Reserved Instances: Up to 75% savings for 1–3 year commitments
✅ Ideal for steady, predictable usage

Spot Instances: Bid for unused capacity, up to 90% cheaper
✅ Great for fault-tolerant, flexible jobs

Savings Plans: Flexible pricing based on usage commitment
✅ Covers multiple EC2 types

Example:
A dev team uses **On-Demand** EC2 for testing, and **Reserved Instances** for production workloads to reduce costs.

Why Use EC2 for Linux Hosting?

Benefits for Developers and Businesses

- Cost-effective and scalable Linux environment
- Supports automation via SSH, scripts, and AMIs
- Integrates with other AWS services like CloudWatch, S3, IAM
- Easy to configure security and firewalls
- Offers wide variety of Linux distros (Amazon Linux,

  Ubuntu, CentOS, etc.)

Example:
An ecommerce store runs its Node.js app on Ubuntu EC2, uses CloudWatch for monitoring, and S3 for static assets.

Popular Linux Distributions Available on AWS

Overview of Commonly Used Linux Flavors in the Cloud

AWS offers a variety of pre-configured Linux AMIs (Amazon Machine Images)

Commonly used distros:

✅ Ubuntu
✅ CentOS
✅ Amazon Linux
✅ Red hat Linux

Example:

A developer may choose **Ubuntu** for quick deployment, while an enterprise may use **Rocky Linux** for legacy app compatibility.

Each serves different purposes: Dev, Enterprise, or AWS-optimized environments

Ubuntu on AWS: Developer-Friendly & Widely Adopted

Features, Use Cases, and Community Support

Use Cases:

✅ App development and testing
✅ Open-source projects
✅ Cloud-native environments

Package Manager: APT (Advanced Packaging Tool)

Benefits:

✅ Large community support
✅ Regular security updates
✅ Easily customizable

AWS Support: Available in official and community AMIs

Example:

A startup uses Ubuntu EC2 with Node.js and MongoDB for building and deploying a web app.

CentOS, Rocky Linux & AlmaLinux on AWS

Enterprise-Grade Linux Options for Long-Term Stability

CentOS: Previously popular for enterprise use; now replaced by alternatives due to lifecycle changes

Use Cases:
✅ Hosting legacy apps
✅ Stable, long-term enterprise support

Package Manager: YUM/DNF (RPM-based)

Benefits:
✅ High compatibility with RHEL (Red Hat Enterprise Linux)
✅ Secure and reliable
✅ Extended lifecycle support

Example:
An enterprise running a legacy ERP solution migrates from CentOS to Rocky Linux on AWS.

Amazon Linux: Built for AWS Performance & Security

Lightweight and Optimized OS for Cloud Workloads

Amazon Linux 2023:
✅ Maintained by AWS
✅ Tight integration with AWS tools (e.g., CloudWatch, IAM, SSM)

Benefits:
✅ Faster boot times
✅ Regular and timely updates
✅ Long-term support cycles

Use Cases:
✅ General-purpose workloads
✅ High-performance apps

Example:
A DevOps team uses Amazon Linux to deploy microservices with optimized networking and low overhead.

Comparing Security, Updates & Performance

Choosing Based on Operational and Security Needs

Distro	Security Updates	Package Manager	Performance	Best For
Ubuntu	Frequent	APT	Moderate	Dev & Open Source
Rocky/AlmaLinux	Enterprise-grade	YUM/DNF	Stable	Legacy & Enterprise Systems
Amazon Linux	AWS-Optimized	DNF	High	Native AWS Workloads

- Consider compliance requirements and app compatibility
- All offer SSH access, user role configs, and cloud-init support

Example:
A fintech firm requiring **FIPS** compliance uses **Rocky Linux**, while a game studio selects **Amazon Linux** for cost and speed.

How to Choose the Right Linux Distribution on AWS

Match Your Workload, Skillset, and Business Goals

Ask these questions:

✅ Is this for dev/testing or production?
✅ Do you need enterprise-grade support?
✅ Is compatibility with a legacy system required?
✅ Do you want deep integration with AWS services?

Best Picks:

✅ Ubuntu – For quick setup and dev agility
✅ Rocky/Alma – For compliance and enterprise workloads
✅ Amazon Linux – For cost-effective, AWS-native apps

Example:

A multi-cloud SaaS product uses Ubuntu for dev, Amazon Linux for staging, and Rocky Linux for production.

Creating an AWS Account – Step-by-Step Guide

How to Get Started with AWS in Minutes

Step 1: Visit aws.amazon.com and click "Create an AWS Account"

Step 2: Email Verification Enter email, account name, and verify via OTP

Step 3: Payment Method Add a credit/debit card (only 500Rs. authorization hold)

Step 4: Identity Verification Phone number confirmation via OTP

Step 5: Select Support Plan: Choose the Basic Free Tier plan

What is AWS Free Tier? – Overview of Free Usage Limits

Understanding What You Get for Free

⏳ **Duration**: Valid for 12 months post-signup

Popular Free Services:

- EC2: 750 hrs/month of **t2.micro/t3.micro**
  - **Some charges may be taken (only 10 to 20 Rs. / hour)**
- S3: 5 GB storage
- RDS: 750 hrs of db.t2.micro + 20 GB
- Lambda: 1M free requests/month
- CloudFront, DynamoDB, and more

Example:

A developer hosts a small WordPress site on t2.micro EC2 under Free Tier with no extra charges.

Staying Within AWS Free Tier – Best Practices

How to Avoid Surprises on Your AWS Bill

- 🔔 Set Billing Alerts using AWS Budgets
- 🛑 Stop/terminate EC2 instances when not in use
- 📊 Use AWS Cost Explorer to track usage
- ⏱️ Don’t exceed 750 EC2 hrs/month (for 1 instance only)
- 🗃️ Use free-tier eligible services only

Example:
A small startup uses Lambda and DynamoDB for their MVP, avoiding server costs while staying fully within the Free Tier.

Logging In & Navigating EC2 Dashboard

Getting Started with EC2 from the AWS Console

- Log in to AWS Management Console
- Select your region (e.g., Asia, Pakistan)
- Navigate to EC2 Dashboard Go to Services > Compute > EC2
- Click Launch Instance” to begin setup

Launching Your EC2 Instance – Step-by-Step

From AMI Selection to Security Configuration

- **Choose AMI**: e.g., **Ubuntu 22.04 LTS**, Amazon Linux 2025
- **Choose Instance Type**: e.g., **t3.micro** (Free Tier)
- **Add Storage**: e.g., 8 GB EBS (default)
- **Add Tags**: Name your instance (e.g “webserver”)
- **Configure Security Group**:
  1. Allow **SSH (22)** from your IP,
  2. Allow **HTTP (80)** for web traffic (optional)
- Review & Launch
- Create or choose an existing key pair (for SSH)

Post-Launch – Instance State & Access

Managing and Connecting to Your EC2 Instance

- After launch, go back to EC2 Dashboard > Instances
- **View:**
  1. Instance State (running, stopped, terminated)
  2. Public IPv4, instance ID, AMI ID, etc.
- **Common Actions:**
  1. Start/Stop: Pause/restart your server
  2. Terminate: Permanently delete instance
- **SSH Access (from terminal):**

```
ssh -i "your-key.pem" ubuntu@<Public-IP>
```

Understanding SSH Key Pairs and Authentication

How Key Pairs Secure EC2 Access

What is a Key Pair?

A key pair consists of a public and private key used for secure SSH login without passwords.

How SSH Works:

Public key is stored on EC2, private key stays with the user. Authentication happens when the private key proves identity.

✅ AWS Only Stores the Public Key

❌ If you lose the private key, you can't connect to the instance.

Creating, Using & Securing Key Pairs

Key Creation and Secure SSH Setup

Creating Key Pairs in AWS:

- Go to EC2 Dashboard > Key Pairs > Create key pair
- Choose format: .pem (for OpenSSH/Linux) or .ppk (for PuTTY/Windows)
- Choose algorithm: **RSA** (widely used), **ED25519** (faster, newer)
- **Download the private key immediately** (can’t be retrieved later)

Secure the Key File:

Run this below command

- chmod 400 your-key.pem (restricts read/write access)

Managing & Troubleshooting Key Pairs

Best Practices & Common Issues

Managing Multiple Key Pairs:

- Use different keys for dev/staging/prod
- Use naming convention: dev-key.pem, staging-key.pem

Troubleshooting SSH Access:

- Permission Denied (publickey) – Wrong key or user
- Key File Permissions Error – Run chmod 400
- Wrong IP Address – Confirm instance public IP
- Port 22 Not Open – Check EC2 security group

Installing Git on Your Local Machine

Set Up Git to Work with EC2 via SSH

Download and Install Git:

Windows: https://git-scm.com/download/win

Mac: Install using Homebrew: bre install git

Liux Ubuntu:

- sudo apt update
- sudo apt install git

Verify Installation:

- git --version

Connect to EC2 Using SSH Command in Git Bash

Use Git Bash (Windows) or Terminal (Mac/Linux)

Steps to Connect:

1. Open **Git Bash** (Windows) or **Terminal** (Mac/Linux)
2. Navigate to the directory where your .pem is saved
3. Set permissions for key (first time only):

```
chmod 400 your-key.pem
```

Use SSH to connect:

```
ssh -i "your-key.pem" ec2-user@<your-public-ip>
```

Tip: Default username depends on AMI

Amazon Linux/AMI: ec2-user
Ubuntu: ubuntu

Verifying EC2 Connection and Next Steps

Confirm Your EC2 Instance is Live & Connected

Successful Login Output:

__| __|_) Amazon Linux 2 AMI

_| (/   EC2 instance welcome message

📂 **Now You Can:**

- Run Linux commands
- Clone a Git repository
- Install software (e.g., Node.js, Docker, Nginx, etc.)

Example Command to Clone a Repo:

```bash
git clone https://github.com/your-repo-name.git
```

Quiz Section

Quiz

Everyone student should click on submit button before time ends otherwise MCQs will not be submitted

[Guidelines of MCQs]

1. There are 20 MCQs
2. Time duration will be 10 minutes
3. This link will be share on 12:25pm (Pakistan time)
4. MCQs will start from 12:30pm (Pakistan time)
5. This is exact time and this will not change
6. Everyone student should click on submit button otherwise MCQs will not be submitted after time will finish
7. Every student should submit Github profile and LinkedIn post link for every class. It include in your performance

Assignment

Assignment should be submit before the next class

[Assignments Requirements]

1. Create a post of today's lecture and post on LinkedIn.
2. Make sure to tag @Plus W @Pak-Japan Centre and instructors LinkedIn profile
3. Upload your code of assignment and lecture on GitHub and share your GitHub profile in respective

   your region group WhatsApp group
4. If you have any query regarding assignment, please share on your region WhatsApp group.
5. Students who already done assignment, please support other students

Q&A Session

ありがとうございます。

Thank you.

شکریا

+W

For the World with Diverse Individualities