Make Inclusive Data the Norm:

Country Landscape Assessment and Training Provider – Ghana

IT EXPERT TRAINING

MAY 6 -9, 2025

Ghana Federation of Disability Organisations

APC Colombia

STATISTICAL SERVICE GHANA

Global Partnership for Sustainable Development Data

Global Partnership for Sustainable Development Data

APC-COLOMBIA
Agencia Presidencial de Cooperación

Make Inclusive Data the Norm
A South-to-South learning project between Colombia, Ghana and Kenya

Make Inclusive Data the Norm

GHANA

IT Experts Training Workshop

May 06-09, 2025

REGISTRATION

Introduction

Why are we here?

Introduction

The “Leave no one behind” is the spine of the 2030 UN SDG 2030 Agenda which has been globally adopted by countries as the catalyst for transformative development.

With this commitment, the marginalized and most disadvantaged groups in our societies will be considered and represented in the design of solutions which will ultimately bridge the inequalities gaps within the society.

Decision makers need data and statistics that are accurate, timely, sufficiently disaggregated, relevant, accessible and easy to use to better understand where the people are, their circumstances, their needs, data collection strategies to enable tailor made solutions.

Objective & Outcome of the Project

Objective

To **carry out a landscape assessment** in Ghana and subsequently **develop a training course on inclusive data**, as well as the ramifications and approaches of inclusive data, including CGD, and focused on the issue of Female Genital Mutilation (FGM), as a dimension of gender-based violence (GBV).

Expected Outcome

The project will enable the three (3) collaborating counties (Ghana, Colombia and Kenya) **to learn from each other and to add to their knowledge resources and toolkits** on inclusive data systems and **increase the effectiveness of the development and implementation of urgent development policies.**

Project Sponsors & Partners

Implementation Process

Key Concepts

Citizen-Generated Data (CGD) – Data collected directly by individuals or communities rather than traditional government or institutional resources.

Inclusive Data – Data that accurately represents all social groups, including marginalized populations, ensuring fair policymaking.

FGM – Female Genital Mutilation (FGM) refers to “all procedures involving partial or total removal of the female external genitalia or other injury to the female genital organs for non-medical reasons.”

GBV – Gender-Based Violence (GBV) is violence committed against a person because of his or her sex or gender. It is forcing another person to do something against his or her will through violence, coercion, threats, deception, cultural expectations, or economic means.

Modules

MOUDUE 1: IT INFRASTRUCTURE & NETWORK SECURITY

MODULE 2: CYBERSECURITY & DATA PROPTECTION (Overview of Cybersecurity Threats)

MODULE 3: CLOUD & HYBRID INFRASTRUCTURE MANAGEMENT

MODULE 4: SOFTWARE DEVELOPMENT & DevOps

MODULE 5: DATA ANALYTICS & VISUALIZATION

MODULE 6: API INTEGRATION & OFFLINE DATA COLLECTION

MODULE 7: IT CAPACITY DEVELOPMENT ROADMAP

Introduction

By the end of this workshop, participants will:

- Understand best practices in IT infrastructure and network security.
- Strengthen cybersecurity knowledge and compliance with Ghana’s Data Protection Act.
- Gain hands-on experience in cloud and hybrid infrastructure management.
- Learn software development best practices, including DevOps and CI/CD pipelines.

Introduction

- Develop expertise in data analytics and visualization for decision-making.
- Understand API integration and offline data collection methodologies.
- Develop a structured IT capacity-building roadmap for long-term sustainability.
- Conduct an assessment of the current network security setup.
- Identify vulnerabilities and propose solutions.

Module 1

IT Infrastructure & Network Security

Assessment of the current network security setup is basically a health check for your digital systems. It's about making sure your network is safe from hackers, data leaks, and other threats.

You check things like:

- Are our passwords and access levels strong enough?
- Is our antivirus and firewall working?
- Are we updating our software regularly?
- Can we detect if something suspicious is happening?
- Do we have a plan if we get hacked?

In simple terms:

"it's making sure your **digital doors are locked**, your **alarms are on**, and **everyone knows what to do** if there's a break-in."

- Identifying vulnerabilities and proposing solutions means **finding the weak spots in your system that hackers could exploit** and **figuring out how to fix them**.

- Think of it like **checking your house for unlocked doors or broken windows**, then locking them up or replacing them.

- Look for what's not secure (e.g. weak passwords, outdated software, unprotected devices).
- Fix it by adding stronger passwords, updating systems, installing security tools, or training your team.

The goal is simple:

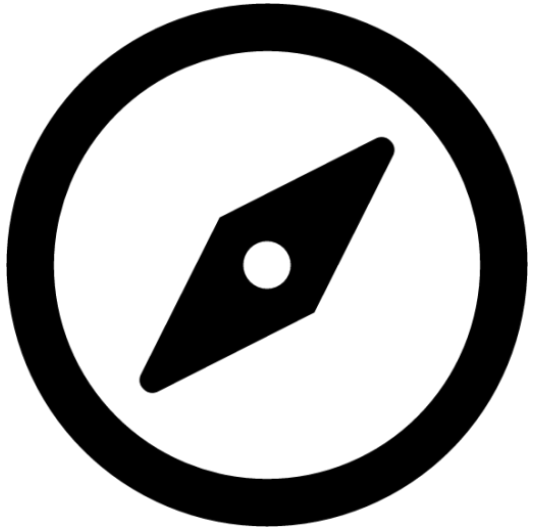
Spot the risks **before someone else does**, and patch them up to keep everything safe.

Module 2

Cybersecurity & Data Protection (Overview Of Cybersecurity Threats)

What we'll explore in this module:

- Ghana's Data Protection Act 2012 (Act 843)
- Endpoint Security & Penetration Testing
- Identifying and Mitigating Vulnerabilities
- Incident Response Planning



MINISTRY OF COMMUNICATION, DIGITAL TECHNOLOGY AND INNOVATIONS
REPUBLIC OF GHANA

DPC
DATA PROTECTION COMMISSION

The Ghana Data Protection Act, 2012 (Act 843)

A law that governs how personal data is collected, processed, stored, and shared in Ghana. Its main goal is to protect the privacy of individuals by setting legal standards for data protection and ensuring accountability from those who manage personal data.

Personal Data: Information about an individual that makes them identifiable.

Data Controller: A person or organization that determines the purposes for which and how personal data is processed.

Data Subject: The individual whose data is being collected or processed.

Data Protection Principles

Data controllers must ensure that personal data is:

- Processed lawfully and fairly.
- Obtained for specific, lawful purposes.
- Accurate and up to date.
- Retained only as long as necessary.
- Processed in accordance with the rights of data subjects.
- Protected against loss, unauthorized access, or damage.
- Not transferred outside Ghana unless the other country ensures adequate protection

Rights of Data Subjects

Individuals have the right to:

- Access their personal data.
- Correct or delete incorrect data.
- Object to the processing of their data.
- Be informed when their data is being collected.

Obligations of Data Controllers

- **Register** with the Data Protection Commission (DPC).
- **Implement security safeguards** (physical and digital).
- **Ensure** data processors they work with comply with the Act.

Role of the Data Protection Commission (DPC)

DATA PROTECTION COMMISSION

- Oversee and enforce compliance with the Act.
- Provide education and guidance on data protection.
- Investigate complaints and apply penalties for non-compliance.

Practical Implications for IT Departments

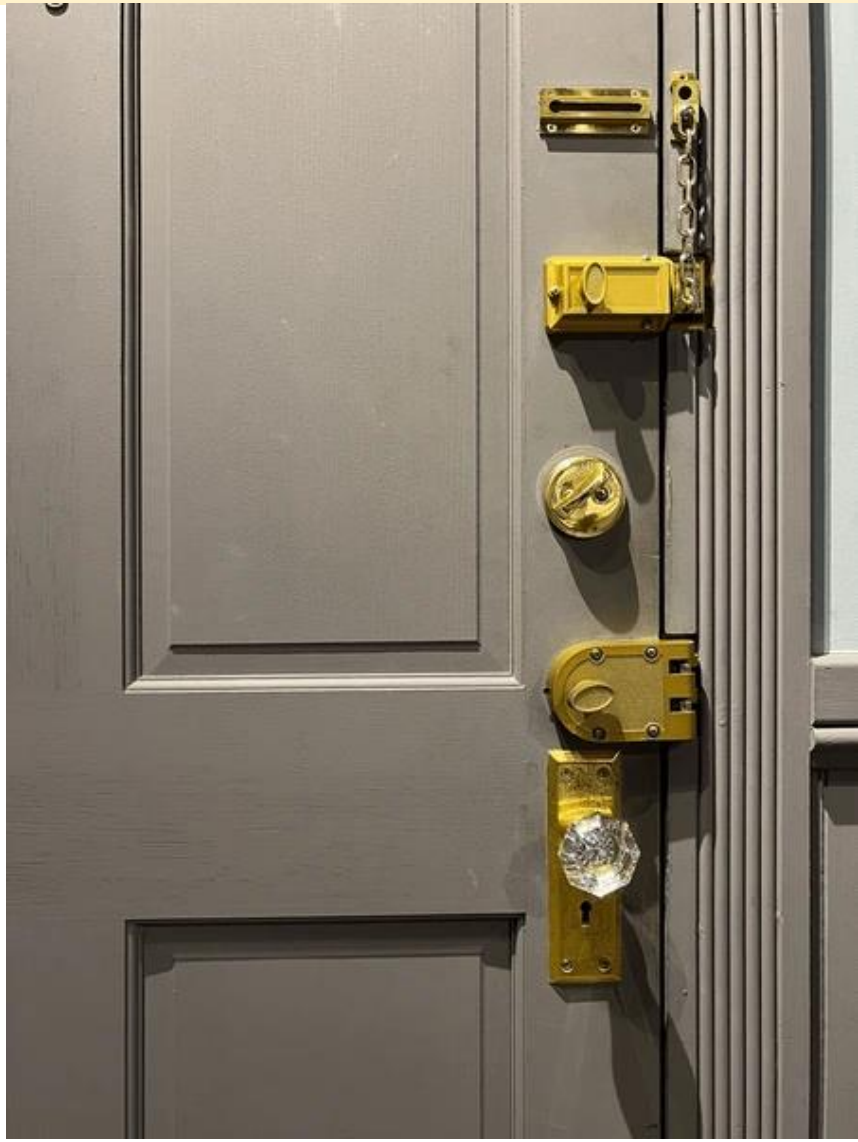
- Implement strong cybersecurity controls to safeguard personal data.
- Train staff on data privacy policies and procedures.
- Conduct regular audits to ensure data handling aligns with Act 843.
- Ensure cloud service providers meet data protection standards, especially if data is stored outside Ghana.

Example of Non-Compliance:

If a government agency collects health data from citizens without consent or shares it with third parties without approval, it violates Act 843 and may face penalties.

Endpoint security

Endpoint security is the practice of securing endpoints devices such as desktops, laptops, mobile phones, and servers that connect to a network. Each endpoint is a potential entry point for cyber threats, making protection critical.



Key Components of Endpoint Security

- **Antivirus/Antimalware:** Detects and removes harmful software.
- **Firewalls:** Controls incoming and outgoing network traffic.
- **Endpoint Detection and Response (EDR):** Advanced monitoring tools that detect suspicious activity and respond in real time.
- **Data Loss Prevention (DLP):** Prevents sensitive information from being sent outside the organization.
- **Disk Encryption:** Encrypts the hard drive (e.g., BitLocker) to prevent unauthorized access to data.
- **Access Control & MFA:** Ensures only authorized users can access systems, often using Multi-Factor Authentication.
- **Mobile Device Management (MDM):** Used to secure and control mobile devices accessing enterprise resources.

Best Practices

- Keep all endpoint software updated and patched.
- Use centralized endpoint management tools.
- Educate users on phishing and unsafe practices.
- Restrict admin privileges where possible.

Penetration Testing

Penetration testing is the process of simulating a cyberattack on a system, network, or application to find and fix vulnerabilities before malicious attackers can exploit them.

Types of Penetration Testing

Network Pen Testing: Tests firewalls, routers, and servers for weaknesses.

Web Application Pen Testing: Tests apps for vulnerabilities like SQL injection or cross-site scripting (XSS).

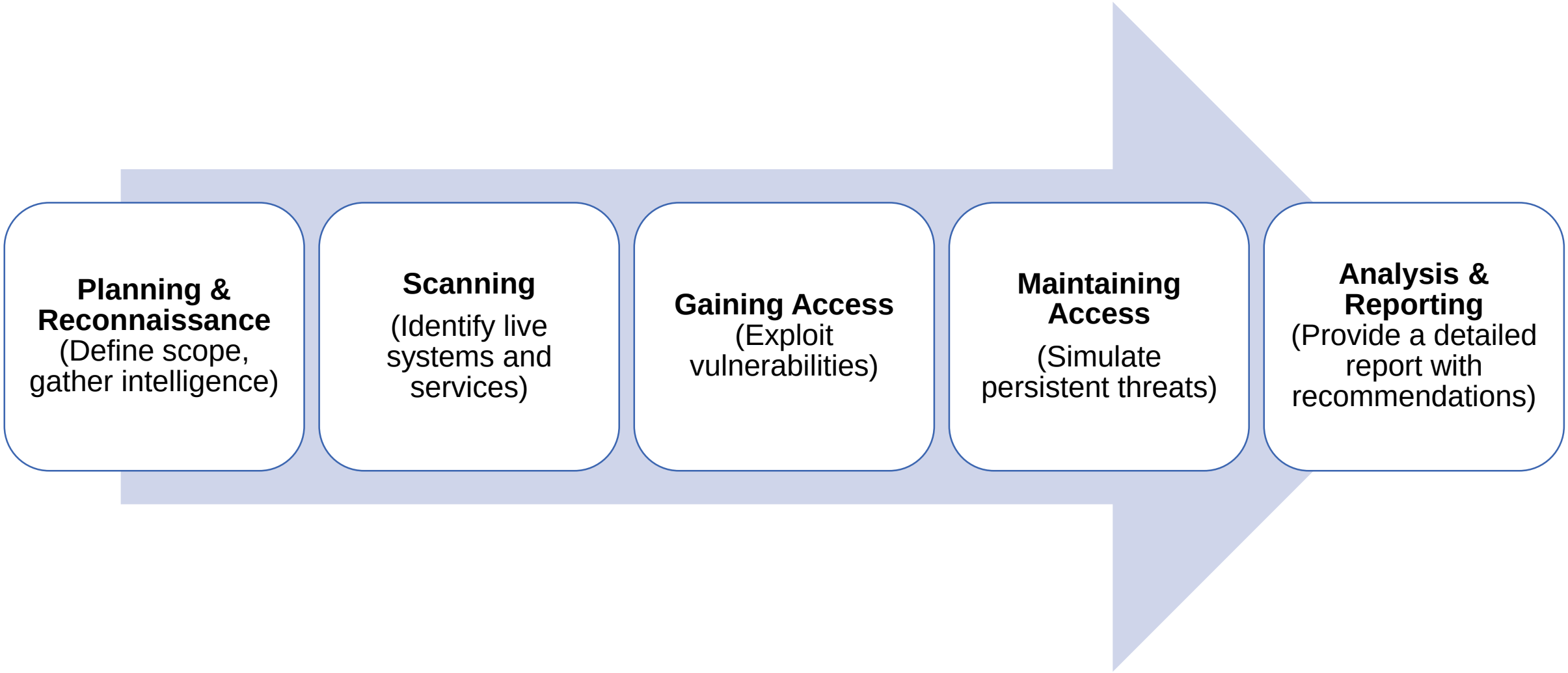
Wireless Pen Testing: Assesses Wi-Fi networks and access points.

Physical Security Testing: Attempts to gain physical access to devices or systems.

Social Engineering Testing: Simulates phishing or pretexting to test user awareness.

Common Tools

Typical Pen Testing Process



Real-World Example

Scenario:

An IT department runs a penetration test and discovers that several employee laptops lack disk encryption and antivirus is outdated. The pen tester also exploits an open RDP port with weak credentials to gain admin access.

Solution:

- Enforce strong password policies.
- Deploy encryption across all laptops
- Update all antivirus definitions.
- Disable RDP or restrict it via VPN and IP whitelisting

Every IT system has weak spots; what we call **vulnerabilities.** These are like open windows or unlocked doors that hackers can use to sneak in. **Our job is to find them before they do, and fix them fast.**

How Do We Spot Vulnerabilities?

Use Scanning Tools: Think of this like a digital security guard checking all your systems for open doors. Tools like Nessus or OpenVAS look for outdated software, weak settings, or missing patches.

Look Manually: Sometimes, it takes human eyes to spot bad habits like shared passwords or servers that give too much access.

Simulate Attacks (Pen Testing): Like a fire- drill; but for cyberattacks. We safely test our defenses to see where we'd break.

Stay Informed: New threats pop up daily. Check trusted sources like CVE reports or vendor updates to stay ahead.

How Do We Fix Vulnerabilities?

Patch It Up: Keep all software and systems updated. Patches are like security repairs.

Lock It Down: Turn off unused features, close open ports, and set strict access controls.

Use Strong Passwords + MFA: A good password + two-step verification = **a much harder system to crack.**

Train Your Team: Teach staff to spot phishing emails and avoid risky clicks.

Watch Everything: Set up alerts and logs so we know when something suspicious is happening.

Have a Plan: If something slips through, we need a solid incident response plan—like calling the fire brigade when there's smoke.

Real-Life Example:

A server still running old software gets flagged in a scan. We patch it immediately, turn off unused services, and make sure only the right people can access it.

Result: Problem solved before an attacker could take advantage.

Incident Response Planning

Incident response planning is about being ready for when things go wrong in your IT systems, like a security breach.

Incident Response Planning - Simplified breakdown

Preparation: Get your team trained and set up the tools you need (like monitoring systems) before an incident happens. Everyone should know their role and what to do if something goes wrong.

Identification: Detect issues early using your systems or reports from staff. Quickly figure out if it's a real problem or a false alarm.

Containment: Act fast to limit damage. For example, disconnect compromised systems to stop the problem from spreading.

Eradication: Find out what caused the issue, remove any threats (like malware), and fix the vulnerabilities to prevent it from happening again.

Recovery: Bring your systems back online carefully, using backups if needed, and watch closely to make sure the issue doesn't come back.

Incident Response Planning - Simplified breakdown

Lessons Learned: After the incident, review what happened, what worked, and what could be improved. Update your plan and train your team to be better prepared next time.

Documentation and Reporting: Keep a record of everything, report to the right authorities if needed, and make sure you've met legal and compliance requirements.

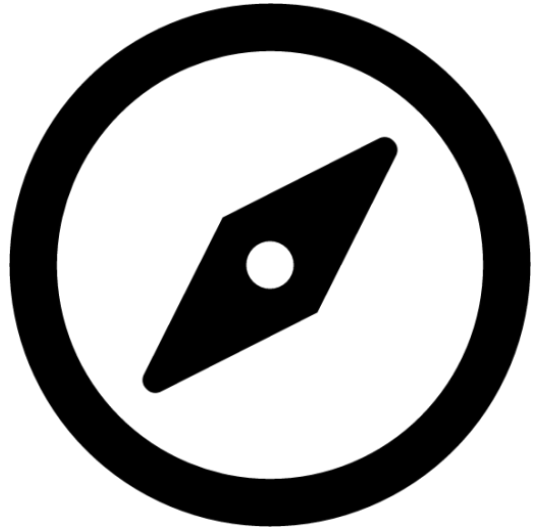
The key is being **quick, organized, and learning** from each incident to improve your response next time.

Module 3

Cloud & Hybrid Infrastructure Management

What we'll explore in this module:

- Introduction to Cloud Computing (AWS, Azure, Google Cloud)
- Cloud Security and Disaster Recovery Planning
- Managing Hybrid IT Systems (On-Site & Cloud Integration)
- Optimizing System Performance & Scalability



Google Cloud

AWS, Azure, and Google Cloud are the three biggest cloud platforms, each offering a wide range of services to help businesses store data, run applications, and scale their infrastructure without having to manage physical hardware.

AWS (Amazon Web Services):
The most widely used cloud platform.

Offers a massive selection of services like computing, storage, databases, machine learning, and more.

Known for its reliability and flexibility, but can be complex to navigate for beginners.

Azure (Microsoft Azure):
Strong integration with Microsoft products like Windows Server, Active Directory, and Office 365.

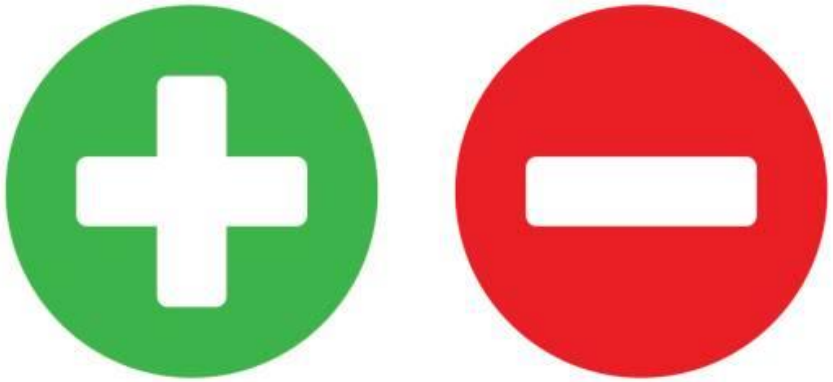
A good choice for companies already using Microsoft software.

Offers a wide variety of cloud services, similar to AWS, and focuses heavily on hybrid cloud setups.

Google Cloud (GCP):
Best known for its strengths in data analytics, machine learning, and open-source tools.

Great for businesses that want to leverage Google's big data capabilities and global infrastructure.

Offers competitive pricing and a focus on innovation in AI and data analytics.



- **Each cloud provider has its strengths**, so the right choice depends on your company's needs, existing software stack, and budget.
- Generally, AWS is the most established, Azure is great for Microsoft shops, and Google Cloud excels in analytics and AI.

Cloud Security

Cloud Security is all about protecting your data, applications, and services that live in the cloud. It includes the following:

Access Control: Making sure only authorized people can access sensitive information.

Encryption: Protecting data both when it's stored and when it's being transferred.

Threat Detection: Using tools to spot potential security issues (like unauthorized access or malware) early.

Cloud Security

Compliance: Ensuring that your cloud services meet industry regulations (like GDPR or HIPAA).

Disaster Recovery Planning (DRP) in the cloud is about making sure you can recover quickly if something goes wrong:

Backup: Regularly back up important data to prevent loss in case of an attack or failure.

Cloud Security

Redundancy: Using multiple cloud regions or availability zones so your systems stay online even if one part of the cloud goes down.

Recovery Procedures: Having a clear plan for how to get systems back up and running as fast as possible, including who to contact and what steps to take.

Testing: Regularly testing your disaster recovery plan to make sure it works when needed.

In short, cloud security focuses on keeping your data safe from threats, while disaster recovery planning ensures you can quickly recover if something goes wrong. Together, they help ensure your cloud environment is both secure and resilient.

Managing hybrid IT systems

Managing hybrid IT systems means combining your on-site (on-premise) infrastructure with cloud services to get the best of both worlds.

Managing hybrid IT systems

Integration: You need to ensure that your on-site systems (like local servers) can work smoothly with your cloud environment. This often involves using tools or services that allow both to communicate and share data seamlessly.

Flexibility: Hybrid systems let you choose where to run different workloads - keeping sensitive data on-site for security or compliance reasons while using the cloud for scalable, less critical tasks.

Security: You have to ensure both your on-site and cloud systems are secure, with strong access controls, encryption, and monitoring across both environments.

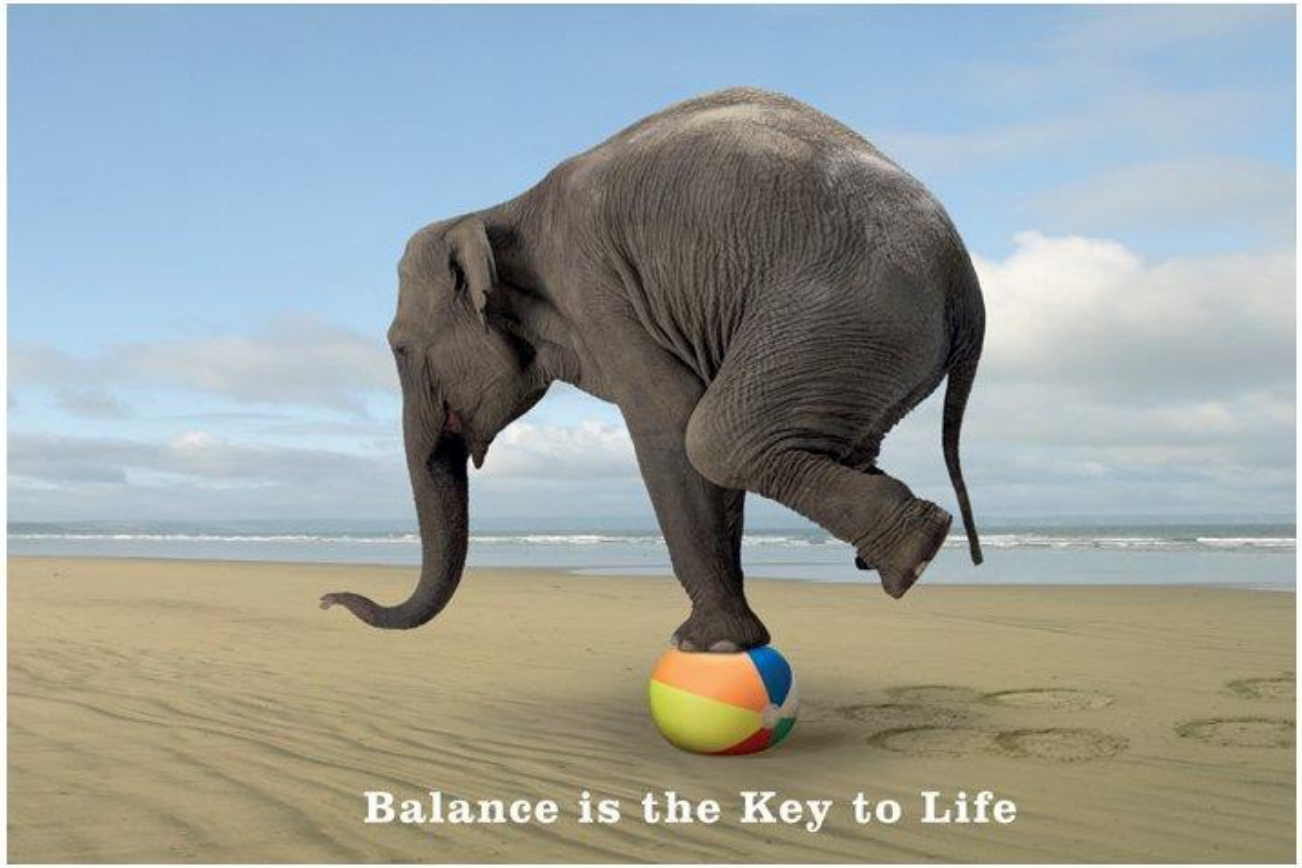
Managing hybrid IT systems

Management: Managing hybrid systems means handling both on-site hardware and cloud services. This might include using unified management platforms to monitor performance, manage resources, and maintain consistency across both environments.

Cost Efficiency: With a hybrid setup, you can optimize costs by moving certain workloads to the cloud when needed (like during peak demand) while keeping others on-site for stability and long-term use.

Managing hybrid IT systems

In short, managing hybrid IT systems involves **balancing** the control and security of on-site infrastructure with the flexibility and scalability of the cloud. It's about finding the right mix based on your company's needs.



Optimizing system performance and scalability

Optimizing system performance and scalability means **making sure your IT systems run efficiently and can grow to handle more work when needed.**

How to optimizing system performance and scalability

Monitoring and tuning: Regularly check system performance and adjust settings (like memory, CPU usage, or disk space) to keep things running at their best.

Reducing bottlenecks: Identify and remove any parts of the system that slow things down (e.g., underperforming servers or network issues).

Scalability: This is ensuring your system can handle more users or data as your business grows.

Vertical Scaling (Scaling Up): Add more power to your existing servers (e.g., more memory or CPU).

Horizontal Scaling (Scaling Out): Add more servers or instances to distribute the load.

How to optimizing system performance and scalability

Auto-scaling: In cloud environments, set up systems that automatically add resources when demand spikes and scale down when demand drops.

In short, **optimizing performance keeps things fast, while scalability ensures your system can grow without issues as demand increases.** Together, they help your systems run smoothly no matter the workload

Steps to deploy a small application in a cloud environment

1. **Create an Account:** Sign up for a cloud service like AWS, Azure, or Google Cloud.

2. **Prepare Your App:** Make sure your app's code and files (e.g., HTML, backend scripts) are ready to go.

3. **Set Up a Virtual Server:** Launch a virtual machine (VM) in the cloud (e.g., AWS EC2, Azure VM, Google Compute Engine). Choose your operating system (Linux or Windows).

4. **Add a Database (If Needed):** If your app uses a database, set one up in the cloud (e.g., Amazon RDS, Azure SQL).

Steps to deploy a small application in a cloud environment

5. Deploy the App: You can either manually upload your app to the VM or automate it using deployment tools like GitHub Actions.

6. Configure Networking: Open the necessary ports (e.g., HTTP/HTTPS) and ensure your app is publicly accessible.

7. Test: Go to your app's public IP or domain name to check if it's working.

8. Monitor (Optional): Set up monitoring tools to keep an eye on performance and get alerts if there are issues.

9. Scale (If Needed): If your app gets a lot of traffic, set up auto-scaling to handle demand automatically.

Steps to deploy a small application in a cloud environment

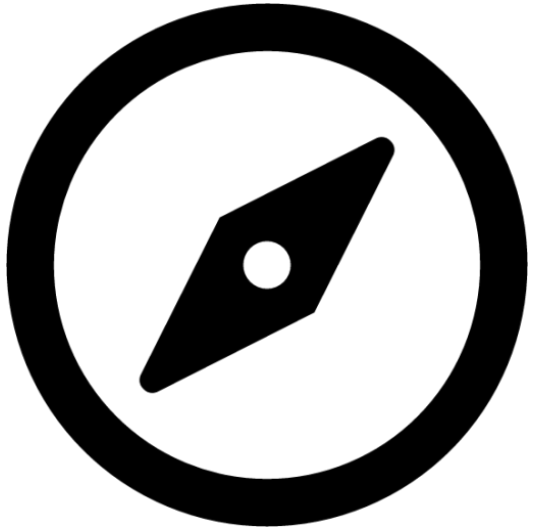
It’s all about **setting up the cloud environment, uploading your app, and making sure it’s secure and accessible**

Module 4

Software Development & DevOps

What we'll explore in this module:

- Full-Stack Development (Python, JavaScript, Flutter)
- Building Scalable Citizen-Generated Data Applications
- Understanding DevOps & CI/CD Pipelines
- Automating Workflows with GitHub Actions/Jenkins



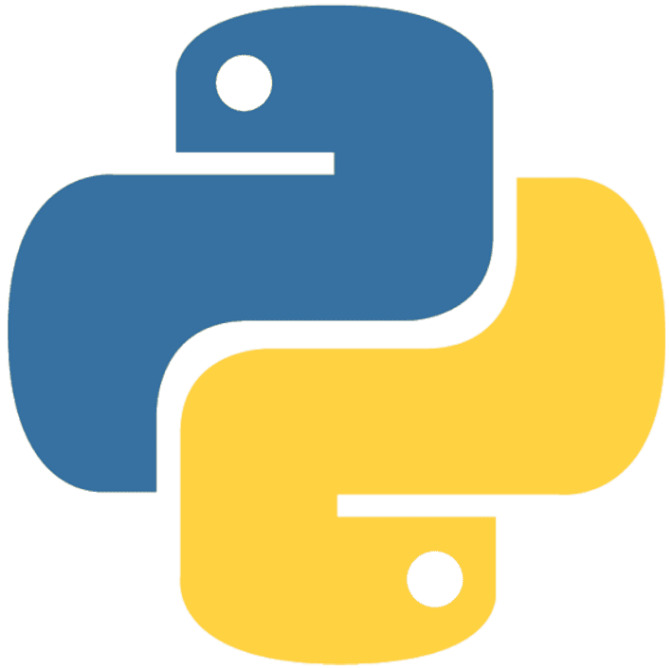
Full-stack development

Full-stack development means **building both the front-end (what users see) and the back-end (the server side) of an application.**

How Python, JavaScript, and Flutter fit into the process

Python (Back-End):
Python is used to build the server side of the app, which handles things like databases, user authentication, and processing logic.

Frameworks like Django or Flask help developers build robust, scalable back-end systems quickly.



How Python, JavaScript, and Flutter fit into the process

JavaScript (Front-End & Back-End)

On the front-end, JavaScript makes websites interactive. It's used in the browser to handle things like buttons, forms, and animations. React or Vue.js are popular libraries to build modern UIs.

On the back-end, JavaScript can also be used with Node.js, a runtime that lets you run JavaScript on the server, handling things like API requests, databases, and server logic.

How Python, JavaScript, and Flutter fit into the process

Flutter (Front-End for Mobile Apps)

Flutter is a framework by Google that lets you build beautiful mobile apps (iOS and Android) with a single codebase.

It’s written in Dart and allows you to create rich, responsive UIs that run smoothly on both platforms.

How Python, JavaScript, and Flutter fit into the process

Putting It All Together:

Back-End (Python): Handle data, servers, and business logic.

Front-End (JavaScript): Create interactive websites that users engage with.

Mobile Front-End (Flutter): Build mobile apps that run on iOS and Android.

How Python, JavaScript, and Flutter fit into the process

A full-stack developer **works with all three layers, managing the user experience, the server, and everything in between.**

This skill set allows you to build complete, end-to-end applications for both web and mobile.

Building scalable citizen data apps

Building scalable citizen-generated data applications means **creating systems that can handle large amounts of data generated by people, like surveys, feedback, or social media contributions.**

Building scalable citizen-generated data apps - KEY CAPABILITIES

Collect Data: The app allows citizens to submit data, like through forms, sensors, or social media. This could be anything from environmental reports to health data.

Store and Manage Data: The data needs to be stored in a secure and scalable way, often in cloud databases that can grow as the data increases. Cloud storage solutions like AWS, Azure, or Google Cloud can scale easily to handle large volumes of data.

Process and Analyze: Once collected, the data must be processed and analyzed. This could mean cleaning up the data, organizing it, or running analytics to extract useful insights (e.g., trends or patterns).

Building scalable citizen-generated data apps - KEY CAPABILITIES

Ensure Scalability: The system needs to be able to grow with more users and data. This is done by using cloud infrastructure that automatically adjusts resources (like server capacity) based on demand.

Ensure Security and Privacy: Since the data is often personal, it’s critical to have strong security measures, like encryption, to protect user information and comply with privacy laws.

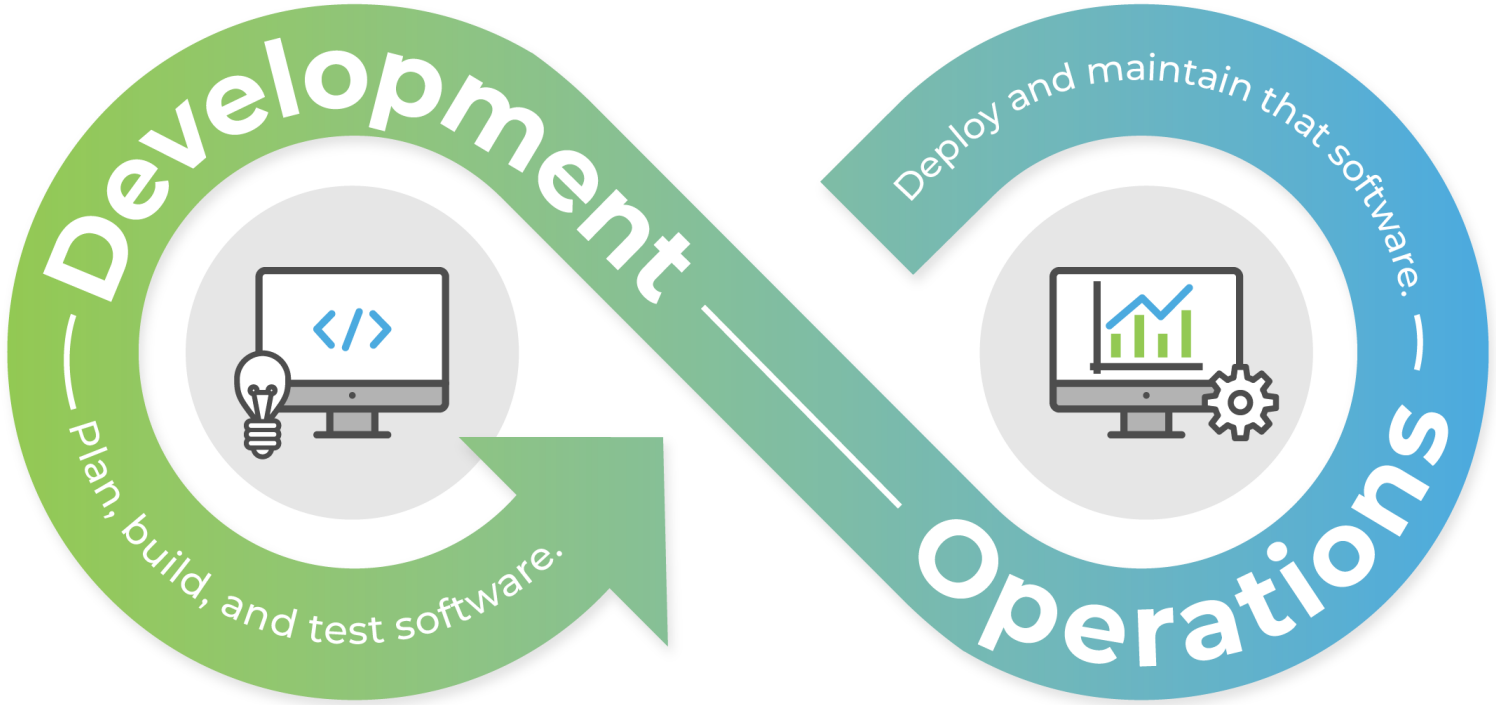
Building scalable citizen-generated data apps - KEY CAPABILITIES

Provide Insights and Reporting: After analyzing the data, the application should deliver useful insights to users, such as reports, dashboards, or visualizations, helping decision-makers or citizens themselves.

In short, building scalable citizen-generated data apps involves **creating a system that can easily handle increasing amounts of data, ensure privacy, and provide valuable insights to users.** It uses cloud tools and strong data processing to make sure the app can grow and adapt as needed.

DevOps

DevOps is a set of practices that **brings together software development (Dev) and IT operations (Ops) to improve collaboration, efficiency, and speed in delivering software.** It focuses on automating processes and improving communication between teams to deliver high-quality software faster.



DevOps - How it works

Continuous Integration (CI):

- Developers frequently push code changes to a shared repository (like Git).
- Every time code is added, it’s automatically built and tested to catch issues early.
- This ensures that new changes don’t break the app and keeps everything in sync.

DevOps - How it works

Continuous Delivery (CD):

- After CI, code is automatically deployed to a staging environment for further testing.
- Once it’s ready, it can be pushed to production (live) with minimal manual intervention.
- This speeds up the process of delivering new features, bug fixes, and updates to users.

DevOps - How it works

In Summary:

- DevOps: Brings teams together to automate and improve software delivery.
- CI/CD: Automatically tests, builds, and deploys code, making sure updates are fast and reliable.
- Together, they make software development smoother, faster, and more reliable.

Automating workflows with GitHub Actions/Jenkins

Automating workflows with GitHub Actions or Jenkins helps developers save time and reduce errors by automating repetitive tasks like building, testing, and deploying code. Here’s how they work:

GitHub Actions:
GitHub Actions is a feature within GitHub that automates workflows directly in your repository.

You can set up workflows to automatically run tasks whenever code changes are pushed (e.g., building, testing, or deploying your app).

Automating workflows with GitHub Actions/Jenkins

It uses **“workflows” (written in YAML)** to define tasks that can be triggered by events, like code commits or pull requests.

Jenkins:
Jenkins is a standalone automation tool widely used for continuous integration and delivery (CI/CD).

You set up Jenkins to monitor your code repository, and it automatically triggers workflows (build, test, deploy) when changes are detected.

Jenkins supports a wide variety of plugins, making it highly customizable for different needs.

Automating workflows with GitHub Actions/Jenkins

In Summary:

GitHub Actions: Automates tasks within your GitHub repository, simplifying workflows for building, testing, and deploying code.

Jenkins: A powerful, customizable tool that automates CI/CD processes and can work with multiple repositories and environments.

Both tools help speed up development by automating workflows, ensuring consistent code quality, and reducing manual steps in the process.

Creating and deploying a simple API for data submission.

Set up your environment: Use Python and Flask to create a basic web API. Flask will handle requests and responses.

Write the API: Create a POST endpoint that accepts data (like name and email), stores it temporarily in a list, and sends a success message in response.

Test Locally: Run the API on your local machine to make sure it works by sending test data through a tool like Postman or curl.

Deploy to the Cloud: Use Heroku to deploy your API. You'll prepare your app by creating a Procfile and requirements.txt file, then push your code to Heroku using Git.

Test the Deployed API: Once deployed, your API will be live. You can send requests to it via Heroku's URL.

THANK YOU.

Global Partnership for Sustainable Development Data

APC-COLOMBIA
Agencia Presidencial de Cooperación

Make Inclusive Data the Norm
A South-to-South learning project between Colombia, Ghana and Kenya

Make Inclusive Data the Norm

GHANA

IT Experts Training Workshop

May 06-09, 2025

FEEDBACK

Photo Credit

https://www.unicef.org/documents/review-technology-based-interventions-address-child-marriage-and-female-genital