Deployment Automation for Banking Systems

Team Leader:
Nourhan Tarek Gohary

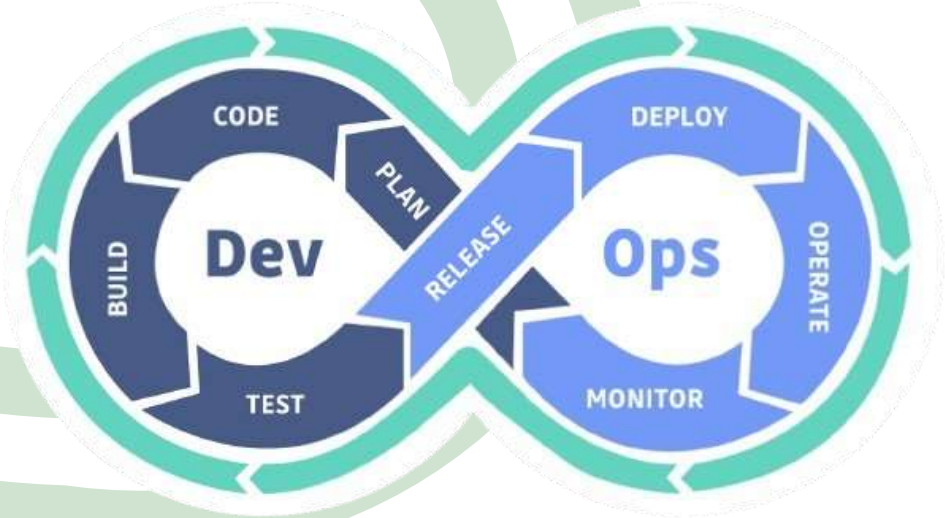
Prepared By:
Ahmed Ashour
Ahmed Mostafa
Ammar Ahmed

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

Coming sections

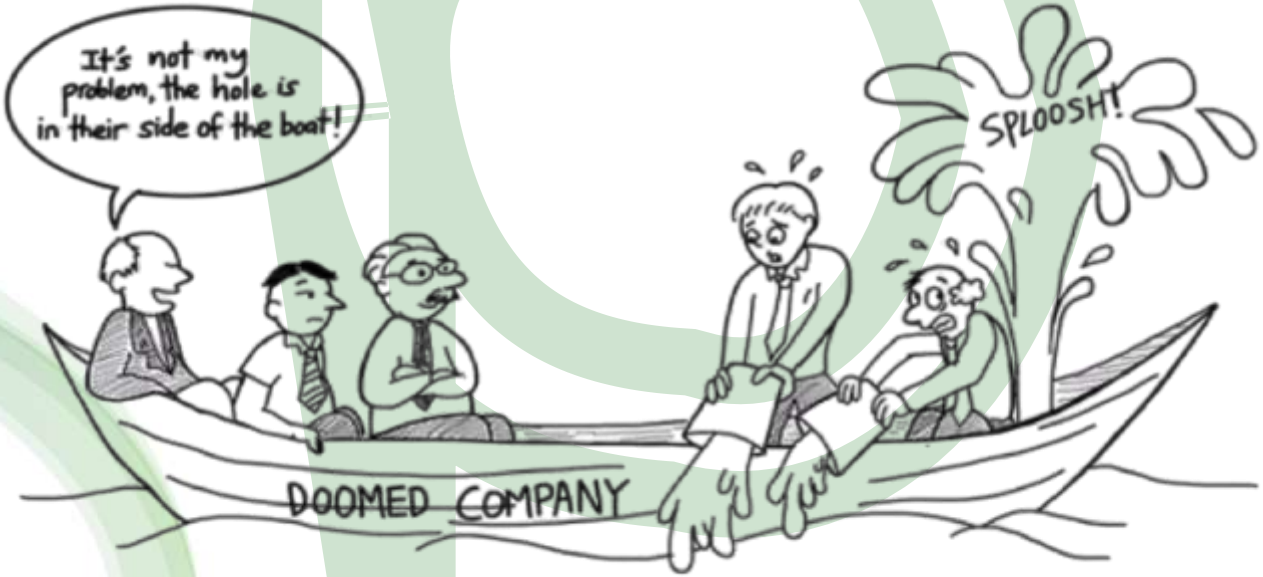
البنك الأهلي المصري
NATIONAL BANK OF EGYPT

1. What’s DevOps
2. Why DevOps Matters
3. Pre Deployment Check
4. Safe Cleanup
5. Full Backup
6. Smart and Git-Aware Backup
7. Conclusion



1.What’s DevOps

In the world of software, speed and stability are everything.
DevOps is a modern approach that brings developers and operations teams together to build, test, and deliver applications faster and more reliably.
It breaks down silos, automates repetitive work, and ensures that every update is smooth, secure, and customer-ready.



2.Why DevOps matters

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

In today's digital banking race, DevOps delivers speed, stability, and security—transforming how banks operate. It merges teams and automates workflows to launch features faster while maintaining ironclad compliance. With DevOps, banks achieve 99.99% uptime, recover from incidents in minutes, and prevent costly outages. More than just efficiency, it turns IT into a growth engine—keeping banks ahead of competitors and customers happy.

3.Pre-deployment Space Check

Before deploying any release to production—especially in systems like **OBDX SIT OHS servers**—it's critical to verify there's enough disk space to avoid crashes. This visual compares two scenarios:
⊘ On the left, skipping a space check leads to a failed deployment.
✓ On the right, using a simple script like `disk_check.sh` ensures enough space, allowing a smooth deployment.

3.Pre-deployment Space Check

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

We first need to know how much space the release will consume before deploying it:

```bash
set -o errexit   # Exit on errors
set -o pipefail  # Catch failures in pipes
set -o nounset   # Avoid undefined variable issues
```

```bash
function die()        # Standardized error handler
function human_readable()  # Formats size in readable units (e.g., MB, GB)
```

3.Pre-deployment Space Check

Unzips and **estimates the required disk space** before proceeding.

```bash
# Navigate and validate
[ -d "$path" ] || die "Directory missing"
[ -f "$required_zip" ] || die "Zip file missing"

# Get estimated extracted size
zip_info=$(unzip -l "$required_zip" | awk '/^---------/{getline; print $4}')
Final_size_M=$((zip_info / (1024 * 1024)))
```

This ensures **/OBDX** mount point has enough free megabytes.

```bash
fs_info=$(df -BM --output=avail "$mount_point" | awk 'NR==2 {print $1}')
available_space_M=${fs_info%M}
```

3.Pre-deployment Space Check

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

Script **proceeds or exits gracefully**, based on calculated space difference.

```bash
diff=$((available_space_M - Final_size_M))
if (( diff > 0 )); then
    echo "✅ Sufficient space, proceed"
else
    die "❌ Not enough space, aborting"
fi
```

4.Safe Cleanup: The Unsung Hero of Deployment

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

Before every smooth deployment, there's a critical backstage process that often goes unnoticed — **cleaning up old or unnecessary files**.

In enterprise environments like **banking platforms (e.g., OBDX)**, keeping the deployment directory clean ensures:

- **Faster deployments** with fewer errors
- **More available disk space** for future updates
- **Lower risk of leftover files** causing version conflicts
- **Less manual troubleshooting** for DevOps teams
- **Cleaner rollback points** and audit trails

4.Secure Cleanup

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

Base Configuration and Safety Check

```bash
# Configuration
BASE_DIR="/OBDX/OBDXui"
DIR_PATTERN="deploy_"
PROTECTED_EXT="zip"

# Safety check: ensure base directory exists
[ -d "$BASE_DIR" ] || {
    echo "ERROR: Base directory $BASE_DIR does not exist"
    exit 1
}
```

It searches recursively inside /OBDX/OBDXui for any directory with the exact name deploy_, and stores the full paths in the array target_dirs.

```
# Base directory and pattern
BASE_DIR="/OBDX/OBDXui"
DIR_PATTERN="deploy_"

# Make sure the base directory exists
[ -d "$BASE_DIR" ] || die "Base directory $BASE_DIR does not exist"

# Read all matching directories into an array
readarray -t target_dirs < <(find "$BASE_DIR" -type d -name "$DIR_PATTERN" | sort)
```

4.Secure Cleanup

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

If any of these checks fail, the script **stops immediately**.

```bash
# Enable strict error handling
set -o errexit   # Exit on command failure
set -o nounset   # Exit on use of unset variables
set -o pipefail  # Exit on pipeline failure
```


```bash
for dir in "${target_dirs[@]}"; do
    echo "Directory: $dir"
    echo "Would delete:"
    find "$dir" -type f ! -name "*.$PROTECTED_EXT" -printf "  %p\n" || true
    echo "Would keep:"
    find "$dir" -type f -name "*.$PROTECTED_EXT" -printf "  %p\n" || true
    echo "----------------------------------------"
done
```

It performs a dry run that lists which files in each matched directory would be deleted (all except .zip files) and which ones would be kept (only .zip files), without actually deleting anything.

4.Secure Cleanup

Before deleting anything, the script **asks the user** for final confirmation.

If the user types **Y/y**, the script **deletes all files** in the directory **except** `deploy.zip`.

If the user answers anything else (or nothing), the script **aborts safely**—nothing is deleted.

```
read -rp "Are you sure you want to delete all files except $PROTECTED_FILE? [y/N] " confirm

# If user confirms, proceed to delete and verify
if [[ "$confirm" =~ ^[Yy]$ ]]; then
    echo "Starting cleanup..."
    # Actual deletion (using extended globbing pattern)
    rm -rf !("$PROTECTED_FILE")
    echo "Cleanup complete."

    # Verify results
    echo "=== Remaining files ==="
    ls -ltr
else
    echo "Cleanup aborted by user."
    exit 0
fi
```

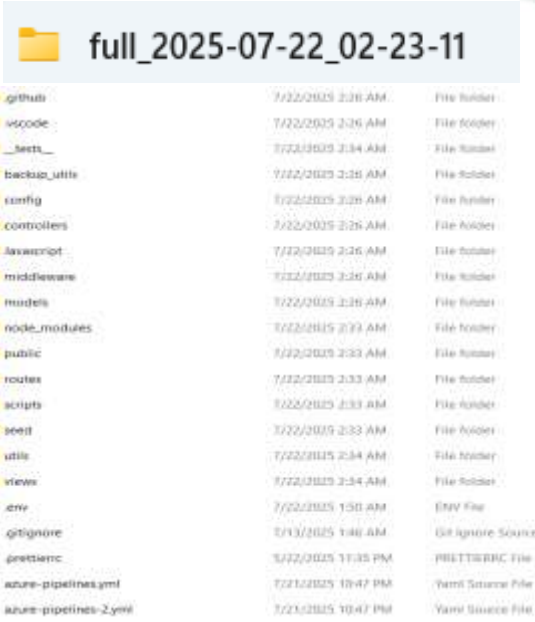
5.Full Backup

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\MIU\Semester 4\Web Development\Project\main\backup_utils> .\Full_backup
Creating full backup of repository at: D:/MIU/Semester 4/Web Development/Project/main
Backup destination: D:\MIU\Semester 4\Web Development\Project\main\backups\full_2025-07-22_02-23-11
Copied: D:\MIU\Semester 4\Web Development\Project\main\Full_backup.ps1 to D:\MIU\Semester 4\Web Development
ll_backup.ps1
```

powershell — Copy — Download

```powershell
# Copy each file individually with full path resolution
$FILES_TO_BACKUP | ForEach-Object {
    $sourceFile = Join-Path $REPO_ROOT $_
    $destFile = Join-Path $BACKUP_FOLDER $_

    if (Test-Path $sourceFile -PathType Leaf) {
        $destDir = [System.IO.Path]::GetDirectoryName($destFile)
        if (-not (Test-Path $destDir)) {
            New-Item -ItemType Directory -Force -Path $destDir | Out-Null
        }
        Copy-Item $sourceFile $destFile -Force
        Write-Host "Copied: $sourceFile to $destFile"
    }
    else {
        Write-Host "Warning: File not found - $sourceFile"
    }
}
```



5.Full Backup

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

The Pros and Cons of making full backup

✅ Upsides
- **Complete data snapshot**
- **Simplifies recovery process**
- **Easy to manage (no dependency on other backups)**

⚠️ Downsides
- **High storage consumption**
- **Time-intensive backup process**
- **Slower recovery for large data sets**
- **Increased network bandwidth usage**
- **Higher infrastructure costs**
- **Not ideal for frequent DevOps iterations**

6.Backup Strategy: Smart, Git-Aware & Clean

What This Part Does

Creates **automated timestamped backups** of the working directory.
Captures **Git branch name and latest commit message** for traceability.
Excludes unnecessary folders like `.git`, `bin/`, `obj/`, and `backups` to keep the package clean.

Why This Matters:

- **Traceability:** Links each backup to a specific point in the Git history.
- **Cleaner Backups:** Removes clutter, keeping only what's needed for deployment rollback.
- **Time-Saving:** Automates a previously manual and error-prone task.
- **Safety Net:** Provides a reliable restore point before major changes or deployments.

6.Smart and Git-Aware Backup

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

Define what to back up and where to store it — safely, cleanly, and organized.

```powershell
$REPO_ROOT = (git rev-parse --show-toplevel).Trim()
$BACKUP_DIR = Join-Path $REPO_ROOT "backups"
$TIMESTAMP = Get-Date -Format "yyyy-MM-dd_HH-mm-ss"
$BACKUP_FOLDER = Join-Path $BACKUP_DIR "incoming_$TIMESTAMP"
New-Item -ItemType Directory -Force -Path $BACKUP_FOLDER | Out-Null
```

Detect only incoming changes

```powershell
$BRANCH = git rev-parse --abbrev-ref HEAD
$INCOMING_FILES = git diff --name-only HEAD..origin/$BRANCH
```

6.Smart and Git-Aware Backup

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

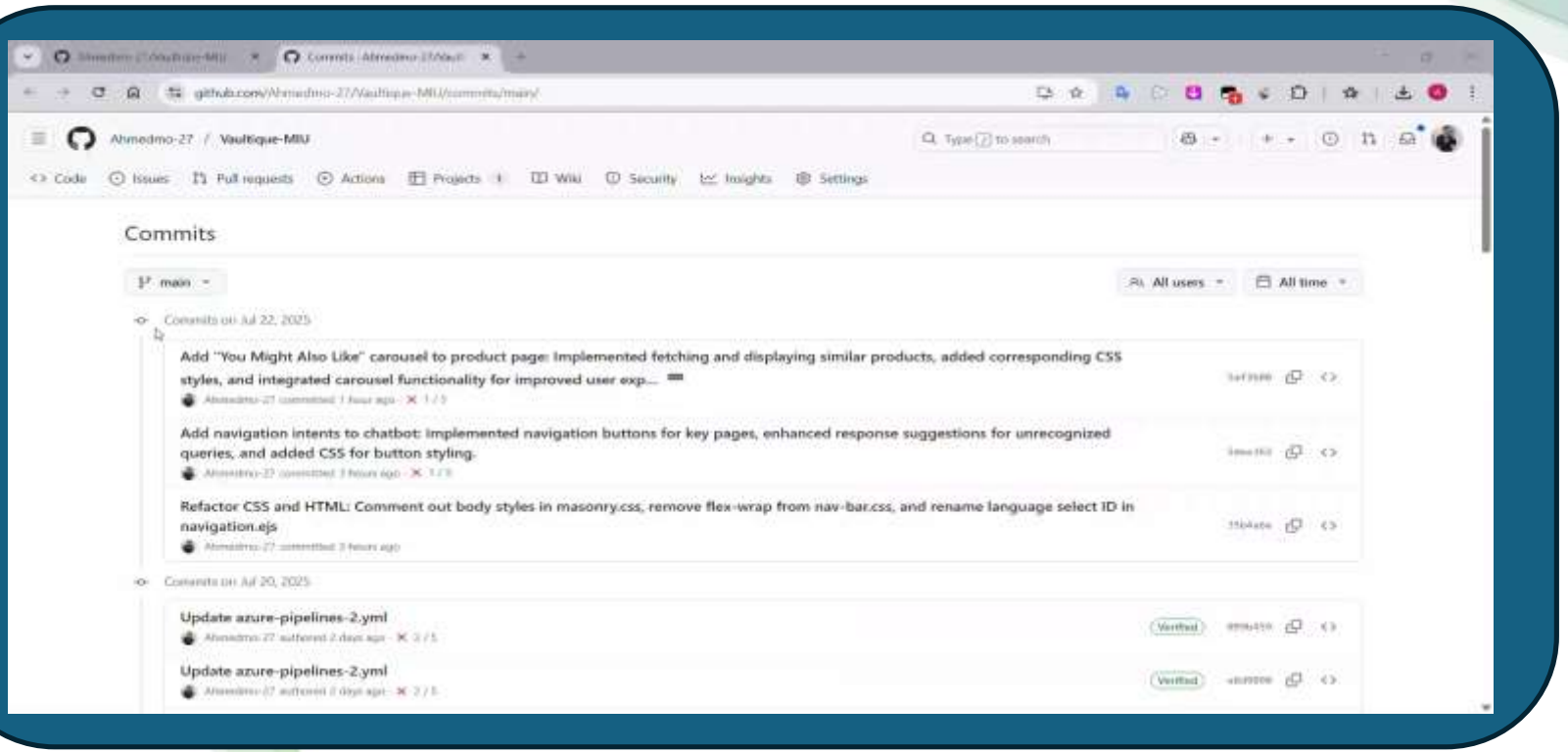
Create a timestamped backup folder and copies only the incoming changed files into it, preserving their directory structure.

```powershell
New-Item -ItemType Directory -Force -Path $BACKUP_FOLDER | Out-Null

$INCOMING_FILES | ForEach-Object {
    $sourceFile = Join-Path $REPO_ROOT $_
    $destFile = Join-Path $BACKUP_FOLDER $_

    if (Test-Path $sourceFile -PathType Leaf) {
        $destDir = [System.IO.Path]::GetDirectoryName($destFile)
        if (-not (Test-Path $destDir)) {
            New-Item -ItemType Directory -Force -Path $destDir | Out-Null
        }
        Copy-Item $sourceFile $destFile -Force
    }
}
```

6.Smart and Git-Aware Backup

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

Demo Video



7.Conclusion
Streamlined, Safe & Smarter DevOps

البنك الأهلي المصري
NATIONAL BANK OF EGYPT

1. Smart Space Check Before Deployment

Prevents mid-deployment crashes by **ensuring enough disk space** before starting the process. Saves teams hours of troubleshooting and rollbacks.

2. Safe Cleanup Practices

Cleans directories **without risking key deployment files**, reducing human error and supporting faster, cleaner releases.

3. Git-Aware Backup System

Keeps your code and context safe with **automatic backups** linked to Git history. Helps you roll back confidently at any time.

البنك الأهلي المصري

NATIONAL BANK OF EGYPT

Thank You