

DevSecOps for ML: Securing Your Model Supply Chain

Building Trustworthy AI Systems



by **Nnenna Ndukwe**

Developer Advocate, Software Engineer





AI is Everywhere, But is it Secure?



ML Everywhere

Development speed outpaces
security



MLOps Complexity

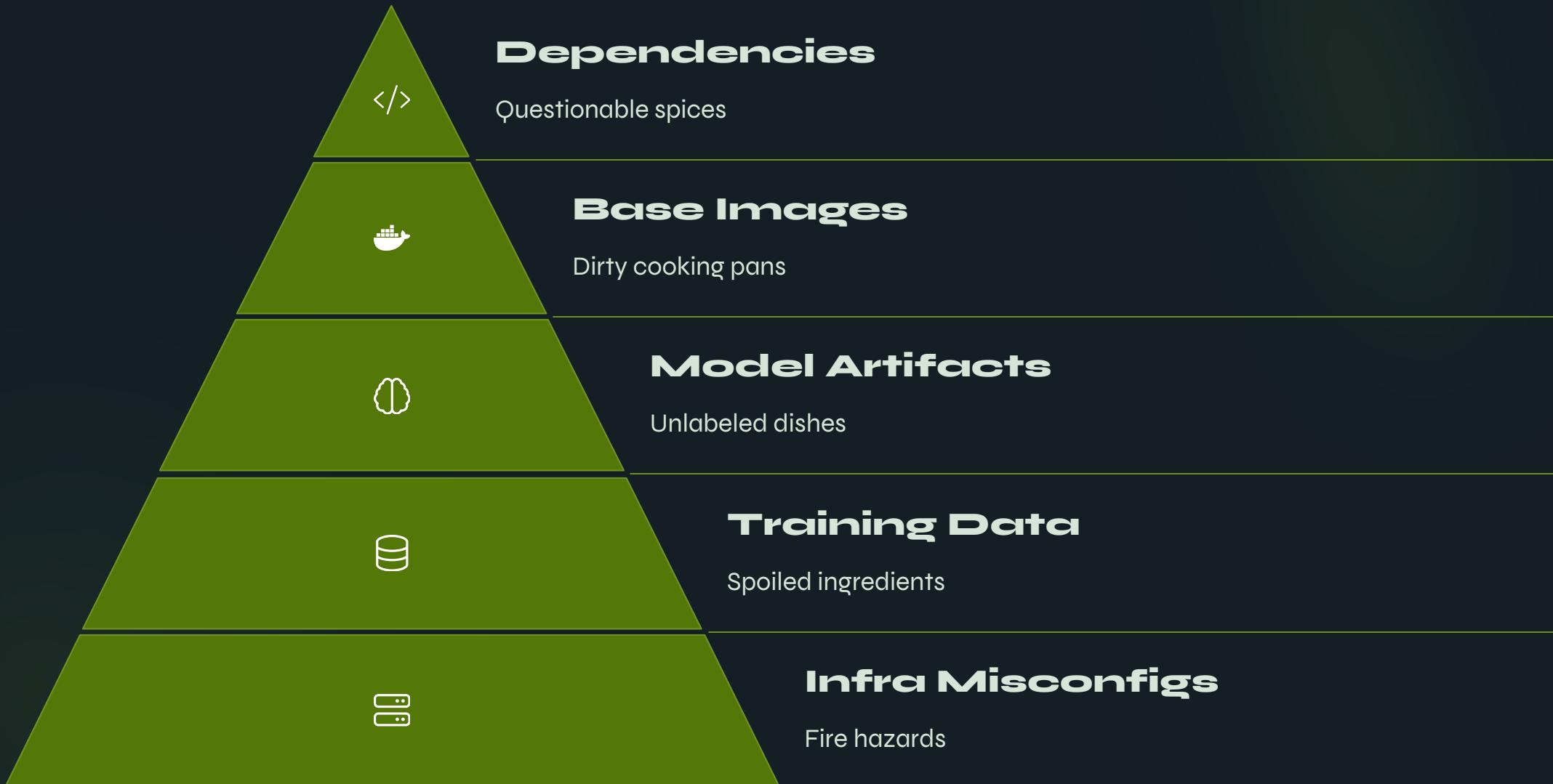
DevOps for ML with more
moving parts



Safety Risk

Like running a restaurant with no food safety rules

Understanding the ML Attack Surface



Shifting Left: Integrating Security into MLOps

- 1 Plan**
Security requirements defined early
- 2 Code**
Secure coding practices
- 3 Build**
Automated security testing
- 4 Deploy**
Verified secure artifacts only



The Right Tools for Secure MLOps

Artifactory

Secure, universal pantry for all artifacts

Xray

Continuous scanner for vulnerabilities

JFrog Platform

Smart pantry + ingredient scanner





Taming Artifact Chaos with Artifactory



Development

Create models and code

CI

Build and package

Artifactory

Central trusted repository

Deployment

Verified artifacts only



JFrog Xray

IPog a verified - un fetchord Dependencies

```
< JFrog Xray
  < IPog a verified - un fetchord Dependencies
    < JFrog Xray
      < Effectors is null vulnerable:
        < Let's consider common accelerators:
          < If we want to do something with ML dependencies (e.g.):
            < Is there a way to do it?
              < A security analysis tool like GitHub Dependency Advisor:
                < Dependency is too slow (e.g. / (ML) artifacts mostly available) (M)
                  < Other security analysis (e.g. ML) for static analysis or fuzzing:
                    < Activity analysis: Run coverage while performing various tests and reviews:
                      < Cover = ML interpretation (M);
                      < Let's get vulnerabilities (M);
                        < Detection, ML (Dependencies, ...):
                          < Description: Day detection (M);
                          < If we want to do something with ML dependencies (M):
                            < A ML attack (M);
                            < If there is no ML dependency (M):
                              < ML detection (M);
                              < ML detection is taking (M);
                              < ML detection is taking (M);
                            < ML detection is taking (M);
                          < ML detection is taking (M);
                        < ML detection is taking (M);
                      >
                    >
                  >
                >
              >
            >
          >
        >
      >
    >
  >
< Vulnerability Scanning:
  < Vulnerability Scanning (M):
    < You can run a regular vulnerability scan
    < Let's do it:
      < Configuration (M):
        < Configuration (M):
          < Configuration (M):
            < Configuration (M):
              < Configuration (M):
                < Configuration (M):
                  < Configuration (M):
                    < Configuration (M):
                      < Configuration (M):
                        < Configuration (M):
                          < Configuration (M):
                            < Configuration (M):
                              < Configuration (M):
                                < Configuration (M):
                                  < Configuration (M):
                                    < Configuration (M):
                                      < Configuration (M):
                                        < Configuration (M):
                                          < Configuration (M):
                                            < Configuration (M):
                                              < Configuration (M):
                                                < Configuration (M):
                                                  < Configuration (M):
                                                    < Configuration (M):
                                                      < Configuration (M):
                                                        < Configuration (M):
                                                          < Configuration (M):
                                                            < Configuration (M):
                                                              < Configuration (M):
                                                                < Configuration (M):
                                                                  < Configuration (M):
                                                                    < Configuration (M):
                                                                      < Configuration (M):
                                                                        < Configuration (M):
                                                                          < Configuration (M):
                                                                            < Configuration (M):
                                                                              < Configuration (M):
                                                                                < Configuration (M):
                                                                                  < Configuration (M):
                                                                                    < Configuration (M):
                                                                                      < Configuration (M):
                        < Configuration (M):
                          < Configuration (M):
                            < Configuration (M):
                              < Configuration (M):
                                < Configuration (M):
                                  < Configuration (M):
                                    < Configuration (M):
                                      < Configuration (M):
                                        < Configuration (M):
                                          < Configuration (M):
                                            < Configuration (M):
                                              < Configuration (M):
                                                < Configuration (M):
                                                  < Configuration (M):
                                                    < Configuration (M):
                                                      < Configuration (M):
                                                        < Configuration (M):
                                                          < Configuration (M):
                                                            < Configuration (M):
                                                              < Configuration (M):
                                                                < Configuration (M):
                                                                  < Configuration (M):
                                                                    < Configuration (M):
                                                                      < Configuration (M):
                                                                        < Configuration (M):
                                                                          < Configuration (M):
                                                                            < Configuration (M):
                                                                              < Configuration (M):
                                                                                < Configuration (M):
                                                                                  < Configuration (M):
                                                                                    < Configuration (M):
                                                                                      < Configuration (M):

```

Uncovering Risks with Xray



Auto-Scan

Scans on
upload/download



Vulnerability Detection

Finds CVEs and security
issues



Deep Inspection

Recursive scanning of
dependencies

Building Security Gates in Your Pipeline

Define Policies

Set security thresholds and rules

Integrate CI/CD

Add Xray scan steps to pipeline

Enforce Gates

Block unsafe artifacts from deployment



The Secure MLOps Loop in Action

Code Commit

Developer pushes ML code

Xray & Policy

Scans and enforces security

Build

CI system creates artifacts

Artifactory

Stores all artifacts securely



Secure MLOps: Key Takeaways



MLOps needs DevSecOps. Principles scale to ML. Artifactory centralizes. Xray inspects. Automation enables security at speed.

Explore Secure MLOps Hands- On

Scan the QR code to access our detailed tutorial.

Learn how to set up Artifactory, Xray, and secure your ML pipeline step-by-step.

New to MLOps? Check out our blog for deeper insights and beginner guidance.

Thanks for joining this talk! Reach out anytime for more examples or questions.

