DevOps Foundations

with Ernest Mueller and James Wickett

DevOps Foundations Glossary

This handout provides definitions of terms used in the course.

A/B Deployment

A deployment strategy where you release feature flagged software and then use a system to open the new features only to a certain subset of users

AIOps

Using machine learning and artificial intelligence to automate IT operations processes

Andon Cord

A device used by any worker to stop a production line in case of a significant issue

Artificial Intelligence (AI)

Creating computing systems that mimic human cognitive abilities

Blameless Postmortems

Retrospectives of an incident that avoid blaming individuals for errors and are focused on learning about your system and organization, which typically contain:

1. A description of the incident
2. A description of contributing causes including participant assumptions and perceptions
3. How the incident was stabilized or fixed
4. A timeline of events including all actions taken to resolve the incident
5. How the incident affected customers
6. Lessons learned, remediations, and corrective actions

Blue-Green Deployment

A deployment strategy where you create an entire new version of the system and then cut user traffic over from the current, or “blue,” system to the new, or “green,” system

Build-Measure-Learn

A feedback loop that is a core component of the Lean Startup methodology

- Build – minimum viable product
- Measure – outcome and internal metrics
- Learn – your problem and your solution
- Repeat – go deeper where it’s needed

Culture, Automation, Measurement, and Sharing (CAMS)

A set of core DevOps values, sometimes known as CALMS, with the addition of lean

Canary Deployment

A deployment strategy where you only upgrade one of many identical systems and let it run for a while to see if it exhibits any problems under production load

Chaos Engineering

The discipline of experimenting on a system to build confidence in the system's capability to withstand real-world conditions in production

Chaos Monkey

The first chaos engineering tool, built by Netflix to intentionally disable cloud servers to ensure their systems continued to operate correctly when that happened

Cloud Computing

"A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"

—NIST Special Publication 800-145

Cloud Native

Something related to Kubernetes

Configuration Management

A process to create and maintain computer systems and software in a desired, consistent state

Continuous Delivery

Automatically releasing an application into its production environment after it passes all test and approval stages

Continuous Deployment

Automatically deploying every change to a production-like test environment and performing automated integration and acceptance testing so that the application is always release ready

Continuous Integration

Automatically building and unit testing the entire application frequently, ideally on every source code check-in so that the application is always in a working state

Declarative (Functional)

Takes a definition of a specific state and executes commands to bring the system to that state

Deployment

Installing and upgrading applications on a system

Developers

Engineers principally tasked with writing applications; also known as programmers, coders, or software engineers

DevOps

The practice of operations and development engineers participating together through the entire service lifecycle, from the design and development process all the way to production support; also characterized by operations engineers using development techniques for their systems work

DevSecOps

The practice of integrating security as a shared responsibility throughout the entire DevOps lifecycle; an extension of DevOps culture for the benefit and inclusion of security

Domain Specific Language (DSL)

A programming language designed not for general purpose coding but for a specialized domain; any infrastructure-as-code tools use a DSL to provide definitions somewhere between a simple template and full programming in complexity and power

Drift

When the configuration of an environment changes and comes out of compliance with its defined state

Fault Injection

Intentionally applying an outage or performance degradation to a live system component

Five Whys

A kaizen continuous improvement technique used to get to the root of a problem by iteratively asking “why” something happened to uncover successive layers of detail and discover underlying factors that contributed

Game Day

A coordinated event where your organization plans to respond to a real or simulated incident to learn and improve

Gemba (xiànchǎng)

The “real place” where work happens

Generative AI

A type of artificial intelligence which uses machine learning tools like large language models to generate new content

Idempotent

Executing the same procedure repeatedly results in the same end state each time

Immutable

An approach to provisioning and deployment of IT resources where components are replaced rather than changed

Imperative (Procedural)

Executing specified commands intended to produce a specific state

Incident

An unplanned event that disrupts business services; also known as an outage

IaaS (Infrastructure as a Service)

A type of cloud computing in which you use infrastructure made available over the Internet, without having to manage the underlying hardware

Infrastructure as Code

Provisioning and managing infrastructure through writing automation code instead of through manual processes

Kaizen (Gǎishàn)

Improvement; in a business context, continuous improvement; a process that encourages everyone to look for ways to improve all parts of the job as an ongoing part of everyday work, and then make small, iterative changes for the better

KISS Principle

"Keep it simple, stupid" –an axiom reminding you to not introduce unnecessary complexity because it causes additional maintenance and failures in the long term

Kubernetes

An open-source container orchestration system for automating software deployment, scaling, and management

Large Language Model (LLM)

A type of ML model trained on large amounts of textual data so that it can generate dynamic intelligent-seeming responses based on prompt questions about that data

Lean Management

A DevOps methodology focusing on using small batches of work, work in progress limits, feedback loops, and visualization.

Machine Learning (ML)

A subset of artificial intelligence (AI) that allows a computer system the ability to learn from and make predictions based on data without having that use case explicitly programmed in

Machine Learning Model

A file containing an algorithm that has been trained to recognize certain kinds of patterns; its size is based on the number of parameters but can range from a few MB to many GB

MLOps

The extension of DevOps for machine learning systems, performing deployment and maintenance of machine learning models and including the needs of data scientists in the development process

Muda (Muda)

Useless activity, or work that absorbs resources but adds no value

Type 1: necessary but nonvalue add

Type 2: unnecessary

Mura (Bān)

Irregularity, or work coming in unevenly instead of a constant or regular flow, leading to delays and wait times

Muri (Muri)

Overwork, or unreasonable work imposed on workers and machines, leading to fatigue and breakdowns

Observability

A measure of how well the internal states of a system can be inferred from knowledge of its external outputs

Operations Engineers

Engineers principally tasked with build, deployment, maintenance, and monitoring of running computer systems; also known historically as system administrators

Orchestration

Coordination and management of computer systems and software

PaaS (Platform as a Service)

A type of cloud computing in which you use an application hosting platform made available over the internet, without having to manage the underlying systems

People over Process over Tools

A DevOps methodology that focuses on identifying who’s responsible for a job function first, and then defining the process that needs to happen around those people, and then selecting a tool to perform that process – in that order

Platform Engineering

The discipline of designing and building toolchains and workflows that enable self-service capabilities for software engineering organizations

Production Environment

The live end-user–facing installation of a service, as opposed to a development or test environment

Prompt Engineering

The process of structuring a text query that can be interpreted and understood by a generative AI model

Provisioning

Creating a system of computing infrastructure and making it ready for operation

Reliability

The ability of a system to perform its intended function correctly and consistently when it’s expected to; it includes performance, availability, and security

Resilience

The ability of a system to maintain or regain a stable state and continue operations after a major mishap and/or in the presence of a continuous stress

Rolling Deployment

A deployment strategy where you upgrade one of many identical systems at a time, to allow seamless shifting of traffic to make the upgrade invisible to the user

SaaS (Software as a Service)

A type of cloud computing in which you use an application made available over the internet, without having to manage the infrastructure or software

Self-Service

Activities safely automated to the point that an end user can perform them on demand

Shift Left

Performing functions earlier in the software delivery lifecycle as part of the development phase – testing, security, reliability engineering – instead of leaving them till after the software is delivered

Serverless Architecture

“Serverless … is run in stateless compute containers that are event-triggered, ephemeral (may only last for one invocation), and fully managed by a 3rd party.”

—Martin Fowler

Site Reliability Engineer (SRE)

A job title for an engineer tasked with operational support but who performs that role by taking a software engineering approach

The Three Ways of DevOps

A set of principles for DevOps

1. Systems Thinking
2. Amplify Feedback Loops
3. A Culture of Continuous Experimentation and Learning

—Gene Kim and Mike Orzen

Toolchain Approach

Using multiple smaller tools that can be easily composed with each other into a set of steps

Transparent Uptime

Communicating frankly with your customers about system outages; the four prerequisites for doing this successfully:

1. Admit failure.
2. Sound like a human.
3. Have a communication channel.
4. Above all else, be authentic.

Value Stream Mapping

Diagramming the steps in your path of delivering a product

Version Control (a.k.a. Source Control)

Tools that track and manage changes to software code over time

Visible Ops Change Control

A DevOps methodology that implements change control with an emphasis on eliminating fragile artifacts, creating a repeatable build process, and then building an environment of continual improvement

Well-Behaved Tools

Tools that can be manipulated in a developer-friendly way – they or their config can be checked into source control, tested, automatically deployed, and integrated with other tools in a toolchain