Over solid blue

endava

Introduction to DevOps

Internship
Spring 2025

Who am I?

Igor Bannicov
Senior DevOps Consultant
Igor.bannicov@endava.com

Agenda

- 01 DevOps timelines
- 02 Define DevOps
- 03 What does a DevOps do?
- 04 What should a DevOps know?
- 05 DevOps toolset
- 06 DevOps mindset
- 07 Q&A

01 DevOps Timelines

DevOps Timeline

DevOps is Introduced 01

Patrick Debois and Andrew Clay come with "DevOps" at "Birds of a Feather" Session

Agile Conference, Toronto

2008

2009

02 **1st DevOps Days**

Short Description

Ghent, Belgium

03 **State of DevOps report**

1st Report was drawn up and launched by Alanna Brown

Puppet

2012

2013

04 **The Phoenix Project Published**

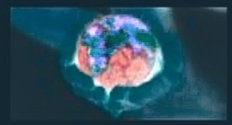
Fictional Novel about IT, DevOps and helping Organization adopt DevOps by Nicole Forsgren, Gene Kim, Jez Humble and others

IT Revolutions

05 **DevOps Impact "Enterprise"**

Many organization adopt the DevOps culture. DevOps adoption increasing worldwide.

2014



N..

06 **Next Gen of DevOps**

More organization using DevOps and SRE part of DevOps. DevSecOps approach introduced. More event like DevOps Days held Worldwide.

SRE

DevOps milestones

1

Pre-DevOps Era (pre-2000s)

- Dev and Ops were totally siloed.
- Waterfall model ruled – slow, rigid releases.
- "It works on my machine" was a lifestyle 😅

2

Agile Manifesto (2001)

- Agile dev sparked the idea of *collaboration and iteration*.
- Started to break down walls between teams.

3

Flickr Deploys 10x a Day (2009)

- John Allspaw & Paul Hammond's legendary talk at Velocity Conf:
  “10+ Deploys per Day: Dev and Ops Cooperation.”
- Massive influence – showed dev+ops working *together* was possible.

4

DevOps Term is Born (2009)

- Patrick Debois coined “DevOps” while organizing the **first DevOpsDays** in Belgium.
- A community-driven movement begins.

5

CI/CD Goes Mainstream (2010s)

- Jenkins, GitLab CI, Travis CI, CircleCI take off.
- Pipelines become the heart of modern delivery.

6

Docker Launches (2013)

- Huge for packaging apps reliably.
- Containers start dominating deployment stories.

7

Kubernetes + Cloud-Native Boom (2014–2018)

- Kubernetes (from Google) becomes the container orchestration king.
- Cloud-native patterns reshape infrastructure thinking.

8

DevSecOps Rises (~2017+)

- Security shifts *left* — integrated early in the pipeline.
- Tools like Snyk, Aqua, and policy-as-code grow.

9

GitOps, AIOps, Platform Engineering (2020s)

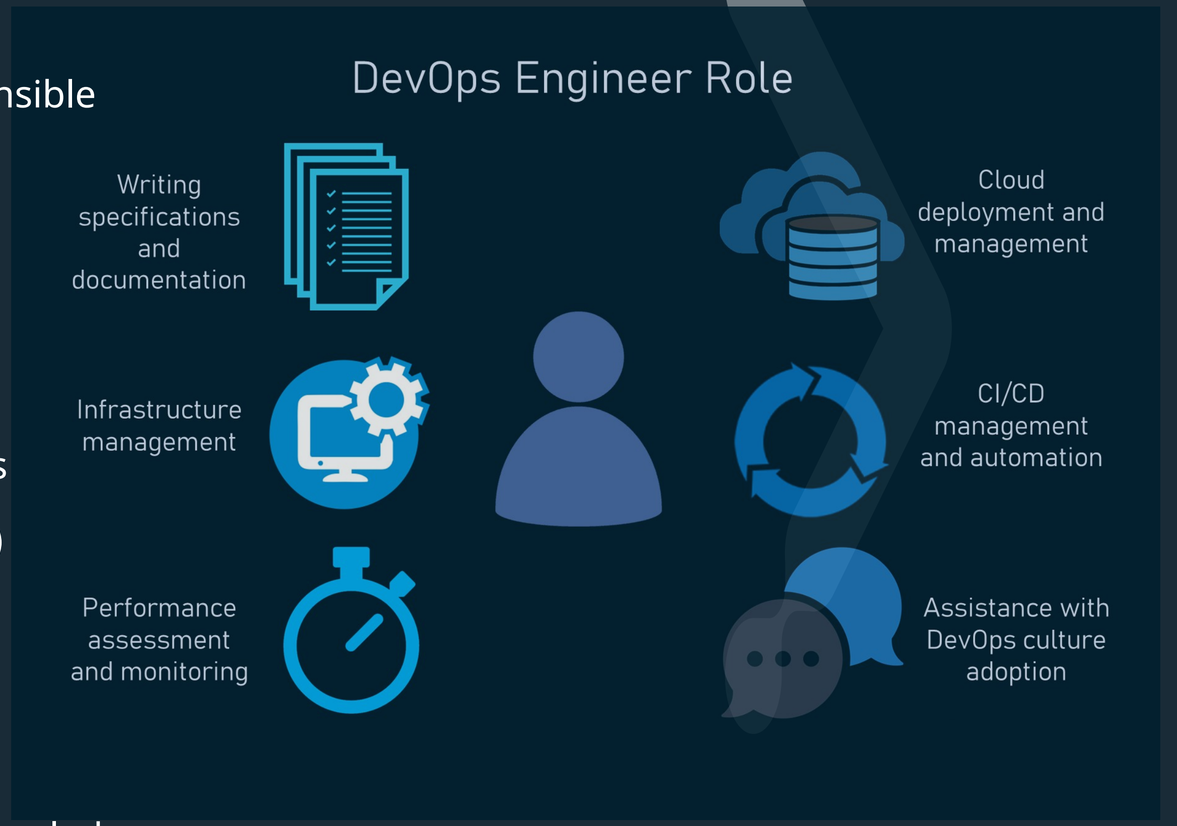
- Git becomes the source of truth for everything (GitOps).
- Observability and automation get smarter (AIOps).
- Platform teams support devs like product teams.

02 Define DevOps

DevOps is everything that happens to the code from developers IDE to the customer.

03 What should a DevOps do?

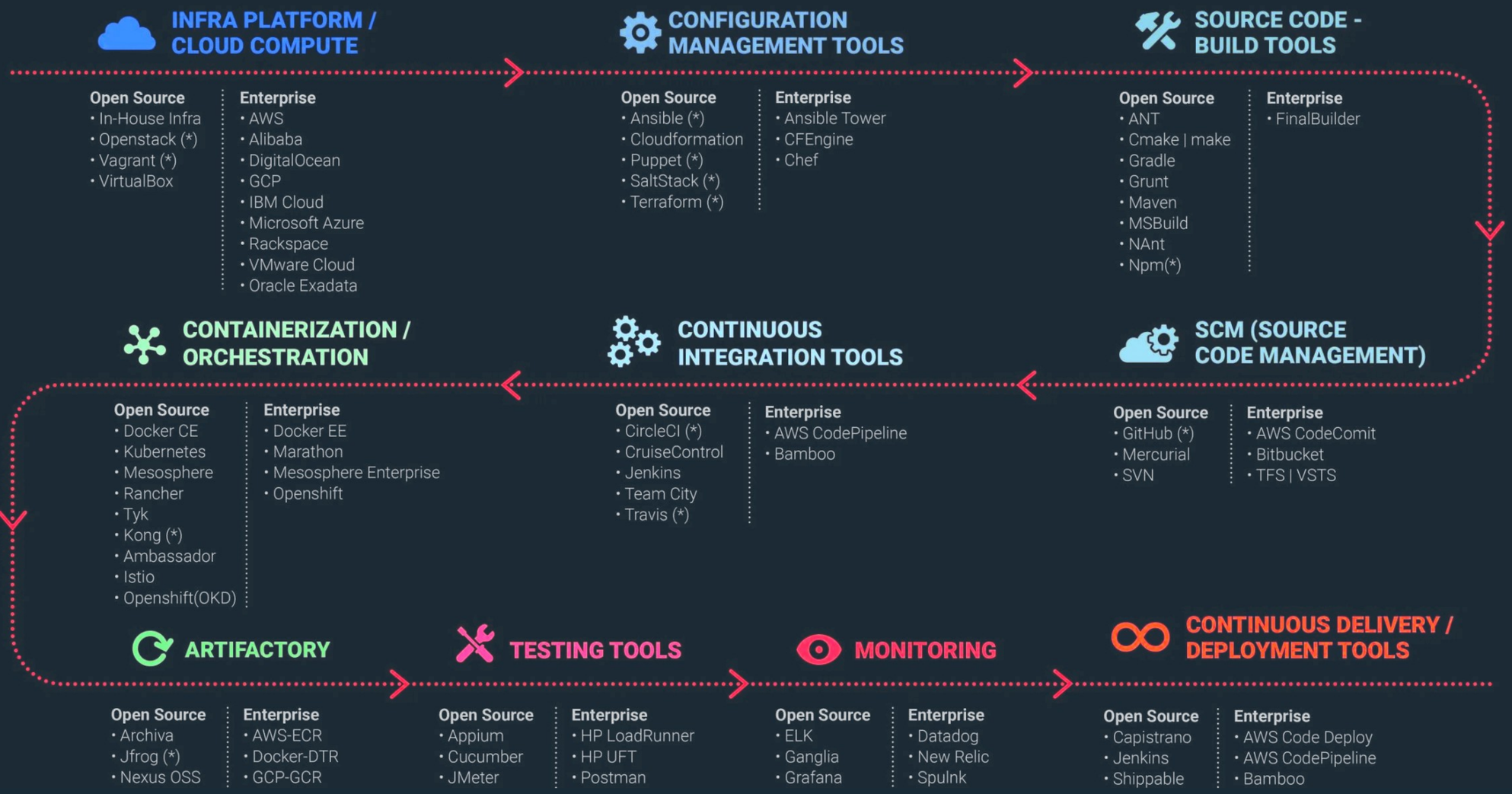
- **CI/CD pipeline building** – automate code build, test, and deploy
- **Infrastructure as Code (IaC)** – write infra using tools like Terraform, Ansible
- **Cloud management** – AWS, Azure, GCP, etc. setup & optimization
- **Monitoring & alerting** – set up tools like Prometheus, Grafana, ELK
- **Automation everywhere** – from testing to backups and scaling
- **Containerization** – Docker images, Kubernetes deployments
- **Version control support** – manage Git workflows, branching strategies
- **Security & compliance** – integrate security in the pipeline (DevSecOps)
- **Incident response** – logs, debugging, and fixing prod issues fast
- **Collaboration** – work closely with devs, QA, and ops teams
- **Performance tuning** – find bottlenecks, optimize systems
- **Documentation & knowledge sharing** – make sure stuff isn't tribal knowledge



04 What should a DevOps know?

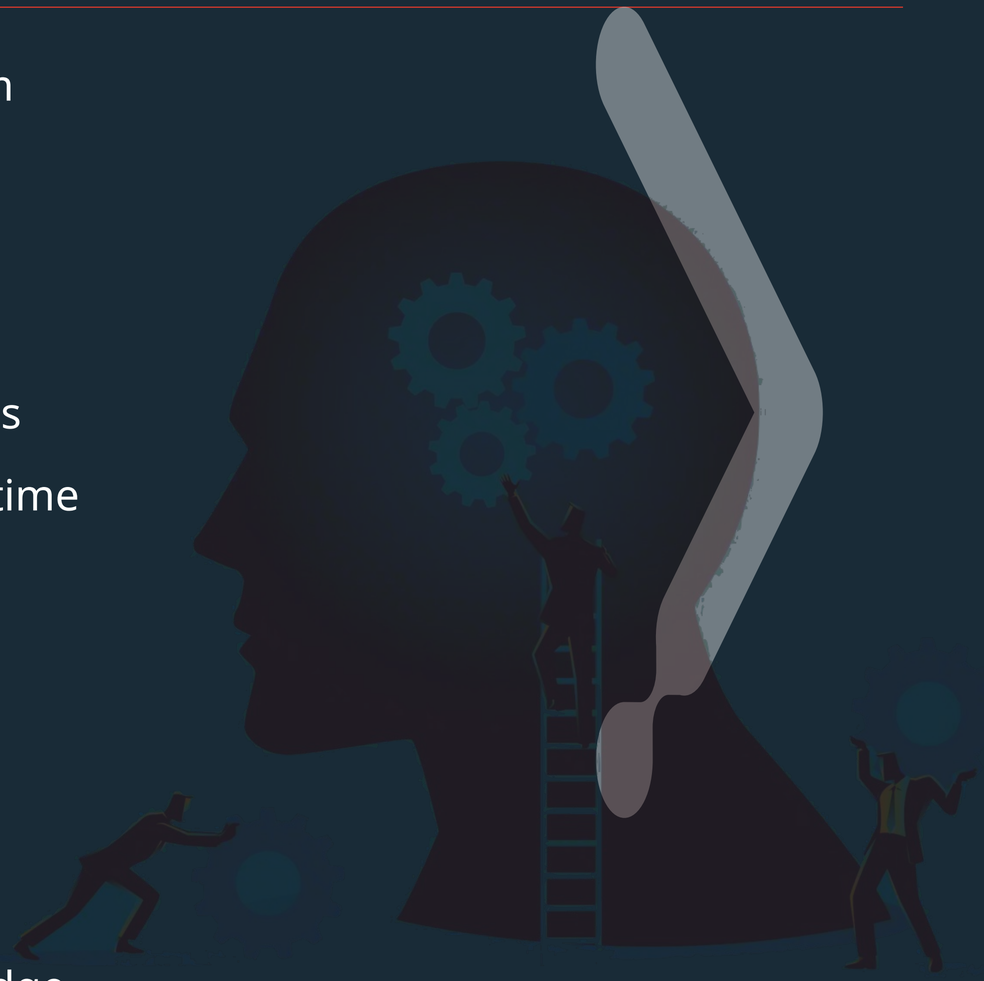
- **Linux fundamentals** – shell scripting, networking, file permissions
- **Git & version control** – Git workflows, branching, GitOps concepts
- **CI/CD tools** – Jenkins, GitLab CI, GitHub Actions, ArgoCD
- **Containers & orchestration** – Docker, Kubernetes (and Helm)
- **Infrastructure as Code (IaC)** – Terraform, Ansible, Pulumi
- **Cloud platforms** – AWS, Azure, GCP basics & core services
- **Monitoring & logging** – Prometheus, Grafana, ELK, Loki
- **Security basics** – secrets management, access control, DevSecOps mindset
- **Networking knowledge** – DNS, TCP/IP, load balancing, firewalls
- **Automation scripting** – Bash, Python, or Go
- **Agile/DevOps culture** – collaboration, feedback loops, fail fast
- **Incident response & troubleshooting** – logs, metrics, system recovery
- **Tooling ecosystem** – know what's out there & how to choose tools wisely

05 DevOps Toolset



06 DevOps Mindset

- **Collaboration over silos** – devs, ops, QA, and security = one team
- **Automate everything** – manual work is fragile and slow
- **Fail fast, recover faster** – embrace failure, build for resilience
- **Continuous improvement** – always look for ways to optimize
- **Measure what matters** – metrics drive decisions, not gut feelings
- **Customer-focused** – the end goal is delivering value, not just uptime
- **Own it end-to-end** – from code to prod, you’re responsible
- **Security is everyone's job** – bake it in from the start
- **Keep it simple** – complexity kills maintainability
- **Learn & adapt** – tech moves fast, stay curious and humble
- **Feedback loops** – short cycles for code, infra, and processes
- **Documentation is part of the product** – don’t gatekeep knowledge



Q&A

Thank you

Over solid blue