

Introduction to DevOps



By:
Goutham Kumar

what do we cover?

1. What is DevOps Engineer Role?
2. What is DevOps?
3. Current state of SDLC
4. Why we need DevOps?
5. DevOps implementation

Speaker details

13+ years of industry experience on

- DevOps
- AWS Cloud
- Automations
- Build and Release Technologies
- Middleware Technologies
- RHEL Administration

Who is a DevOps Engineer

DevOps Engineer is somebody who understands the Software Development Lifecycle and has the outright understanding of various automation tools for developing digital pipelines (CI/ CD pipelines).

But the **question still endures** “Who is a DevOps Engineer?”

DevOps Engineer works with developers and the Operations teams to oversee the code releases.

They are either **developers** who get interested in deployment operations or **sysadmins** who have a passion for configurations and coding.

Current Situation – Before DevOps

Prior to DevOps, the **Development** team and the **Operations** team worked in complete isolation. All were unaware of the current status of the project from each other's perspective. This led to some obvious and unavoidable problems that hindered the progress of IT firms significantly.

Lack of collaboration

Development team and operations team – Isolated

Testing team and Development team – Isolated

Lot of time spent in testing, deploying and designing

Manual Code deployments – Risky deployments

No synchronizing with teams

What is DevOps?

DevOps is not a tool, but it's an approach and a culture with an objective to optimize complete SDLC by bringing development and operations closer.

DevOps is a set of practices, where operations and development teams participate together in the entire application lifecycle, from design, development, testing and finally to the production support.

It emphasizes communication, collaboration, and integration between software developers and information technology (IT) operations personnel.

The feeling of 'oneness', is caused by the bridging of skill-sets & practices through the implementation of automation (DevOps) tools.

Agile Methodology

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.



How DevOps is related to Agile Delivery?

DevOps is built on the concepts of agile methodology.

It identifies the importance of extending agile principles beyond development by including operations and all other functions that support the application development life cycle.

Its more than just a **convergence** of Development + Operations

DevOps is getting involved in to all the phases of software development including quality assurance (QA), testing, security, release management and governance.

Why we need DevOps(reasons for adoption)

Shorter Development Cycles, Faster Innovation

Increase Efficiency & Reduce Maintenance

Reduced Deployment Failures

Faster Rollbacks & Less Recovery Time

Scalable infrastructure & Resiliency

Eliminate downtime

Reduced Costs & Cost Efficiency

Stable operating environment

DevOps Practices

DevOps will be completely achieved if every aspect of the below 5 practices are completed automated.

1: Continuous Development

2: Continuous testing

3: Continuous Integration

4: Continuous Delivery & Deployment

5: Continuous Monitoring

Continuous Development

This phase involves ‘**planning**’ and ‘**coding**’ of the software application’s functionality.

The development is achieved through Agile methodology, where its a software development methodology that emphasizes short, iterative planning and development cycles.

The code can be written in any language, but it is maintained by using **Version Control** tools.

Git, GitHub, Maven are some popular Continuous Development DevOps tools.

Continuous Testing

JUnit



cucumber

This phase involves ‘**testing**’ and ‘**verification**’ of the application functionality.

Continuous Testing is the process of executing automated tests as part of the software delivery pipeline

DevOps cannot be realized without Continuous Testing with **zero human intervention** in smoke/regression/acceptance test execution which will be accomplished through automation of testing through all the stages.

And the continuous use of these tools while developing the application is what forms the ‘Continuous Testing’ phase during DevOps lifecycle.

Continuous Integration

This phase involves ‘**integration**’ of the application functionality.

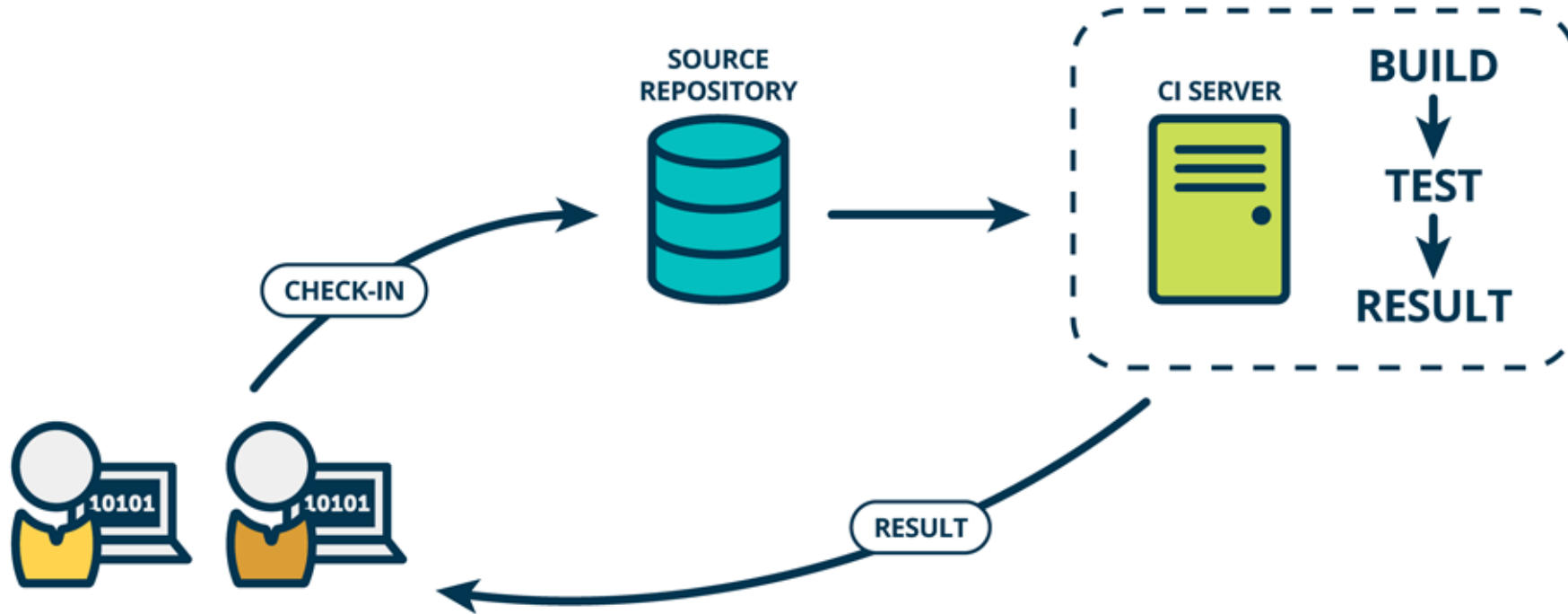
Continuous integration (CI), is a practice of **frequently integrating** the code of each developer into project integration branch and compile, build, and run unit tests on it.

Before CI, integration happened at the end of the creation process, all at once, and took an unknown amount of time; now with CI, it happens everyday, takes minutes and is just “the way we work”.

That way, you detect problems early, fix them when they’re easiest to fix, and get shiny new features to your users as early as possible.

This continual merging prevents a developer’s local copy of a software project from drifting too far and avoiding catastrophic merge conflicts.

Continuous Integration



CI Tools



Jenkins



Bamboo



Travis CI



circleci

Continuous Delivery & Deployment

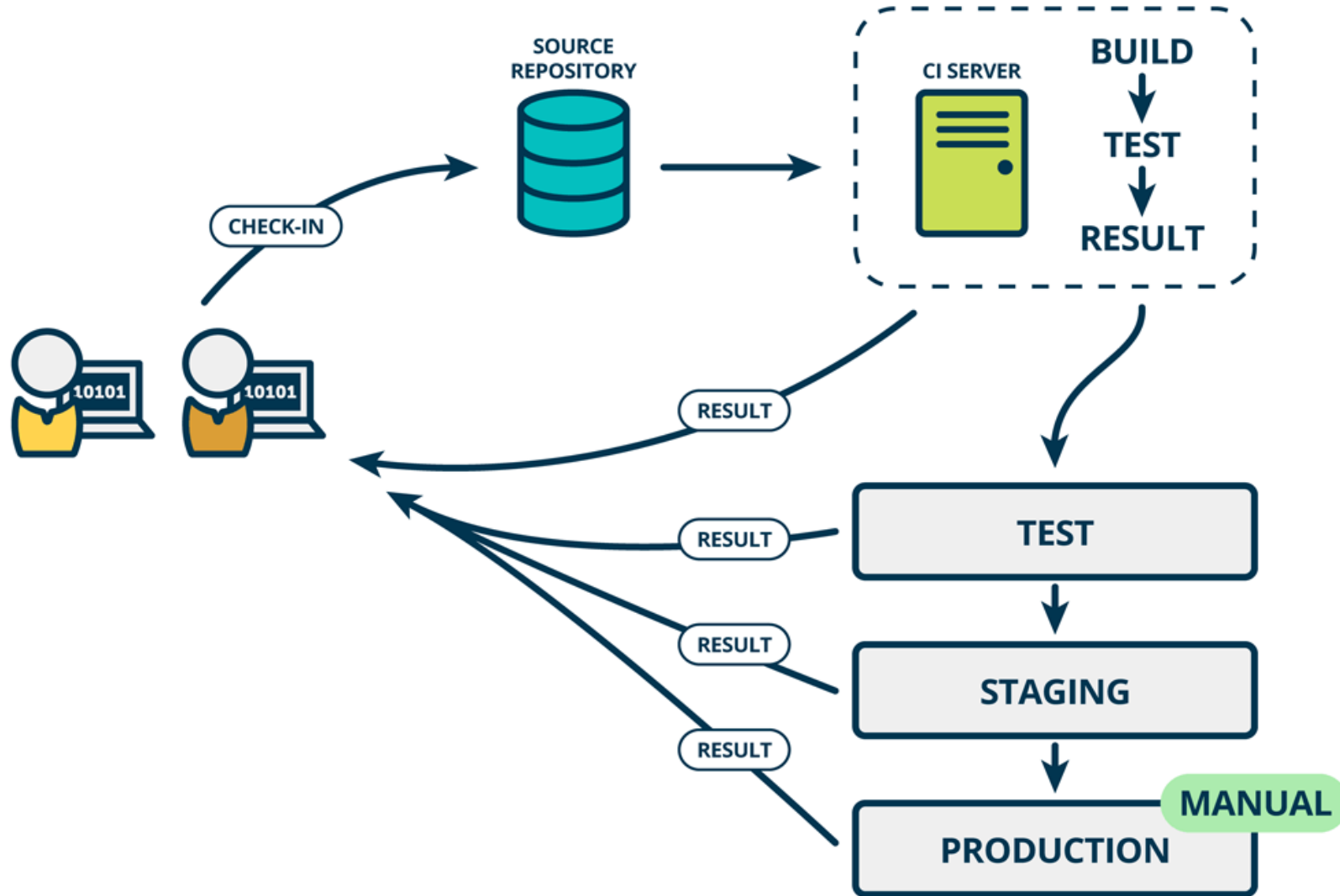
This phase involves **configuration** and **deployment** of the application functionality.

Is a series of practices designed to ensure that code can be **rapidly** and **safely** deployed to Production

By First delivering every change to a production-like environment and ensuring applications and services function as expected through rigorous automated testing.

Once delivered have confidence the application can be deployed to production with a push of a button when the business is ready.

Continuous Delivery & Deployment



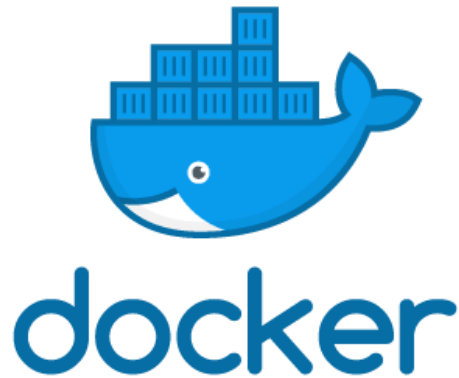
Continuous Delivery & Deployment



Container Tools

Containers have completely revolutionized the way IT runs its applications. And with the Rise of Microservices they became even more popular

**EVERY COMPANY WANT THEIR IT SYSTEMS RUN ON
CONTAINER BASED MICRO SERVICES ARCHITECTURE**

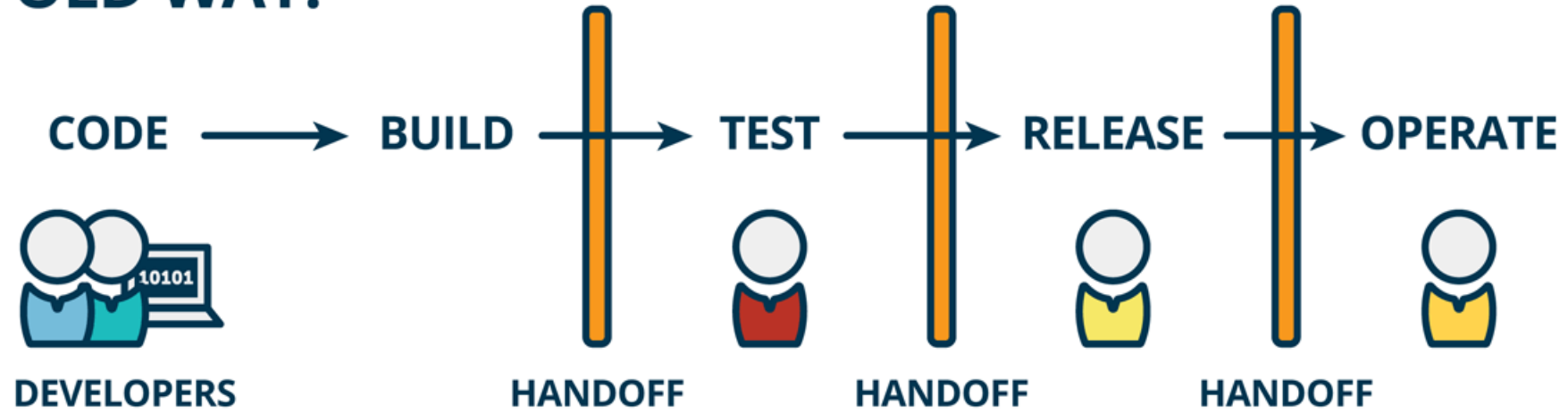


Continuous Monitoring

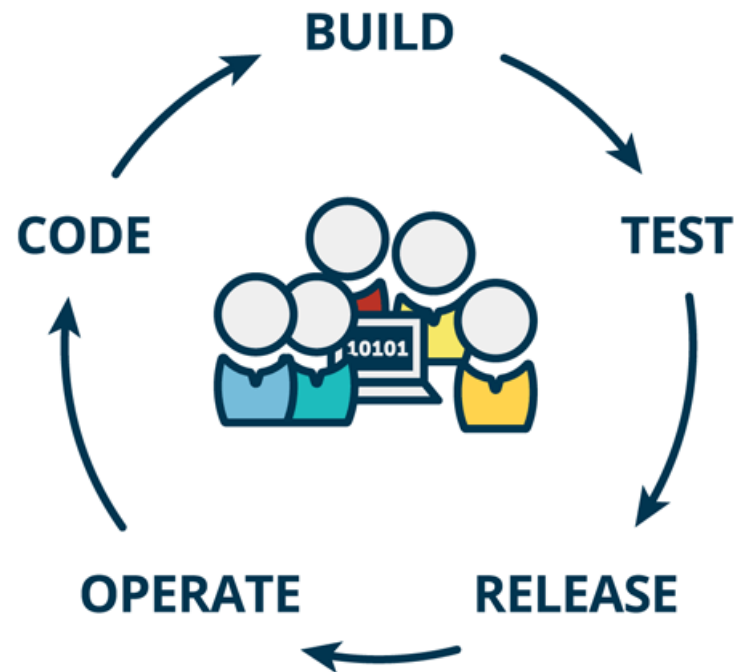
Monitoring is as important as developing the application because there will always be a chance of bugs which escape undetected during the testing phase.

The Nagios logo consists of the word "Nagios" in a bold, black, sans-serif font. The letter "N" is stylized with a horizontal underline. A registered trademark symbol (®) is located at the top right of the word.The ZABBIX logo is a red rectangular box. Inside the box, the word "ZABBIX" is written in large, white, bold, sans-serif capital letters. Below it, the words "MONITORING SYSTEM" are written in smaller, white, sans-serif capital letters.

OLD WAY:



NEW WAY:



List of core DevOps attributes:

1. Ability to use a wide variety of open source technologies and tools
2. Ability to code and script
3. Experience with systems and IT operations
4. Strong grasp of automation tools
5. Comfort with frequent, incremental code testing and deployment
6. Data management skills
7. A strong focus on business outcomes
8. Comfort with collaboration, open communication and reaching across functional borders

What to look for when hiring, recruiting



Developer

- › Customer-focused
- › Exposure to, or experience in, test, ops
- › Interest in taking on traditional testing, ops tasks
- › Adaptable



Tester

- › Customer-focused
- › Experience, or at least interest in, writing code
- › Adaptable



Ops

- › Customer-focused
- › Experience, or at least interest in, writing code
- › Adaptable



Thank You