

Topics

Following are the topics covered in this module:

- Introduction to DevOps
- Benefits of working in DevOps environment
- DevOps Lifecycle
- DevOps Stages
- DevOps Delivery Pipeline

Objectives

At the end of this module, you will be able to:

- Understand the benefits of DevOps over other software development processes
- Gain insights of the DevOps environment
- Get overview of different DevOps Tools
- Get the picture of working of DevOps Delivery Pipeline

Issues Of A Boss From The Company



Every Team Is Happily Working In Silos

Meanwhile every team is happily working in Silos with their own objective...



Customer



Business Owner



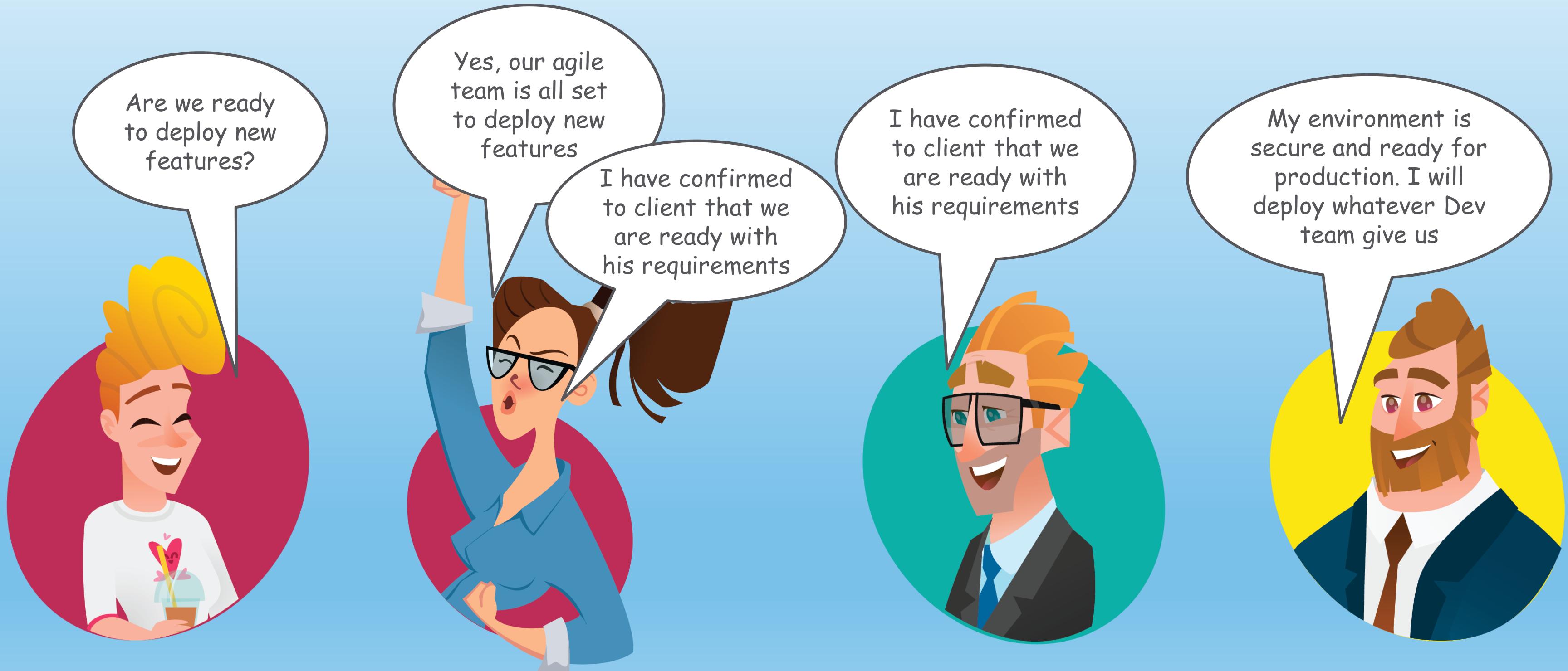
Agile Dev Team



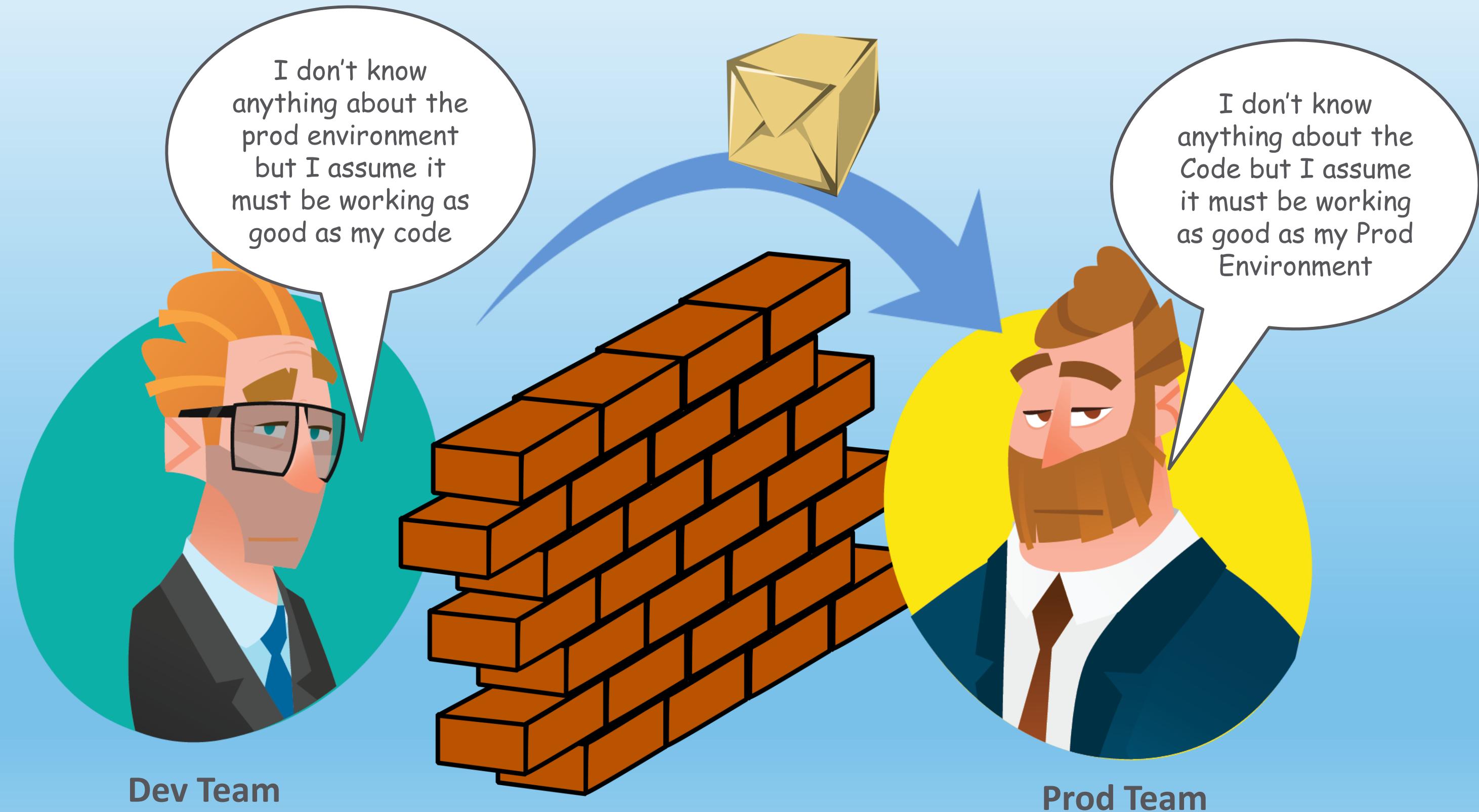
Ops Team

Agile Meets Deployment

Agile seems fine without DevOps culture till deployment comes into the picture...



Deployment By Teams Working In Silos



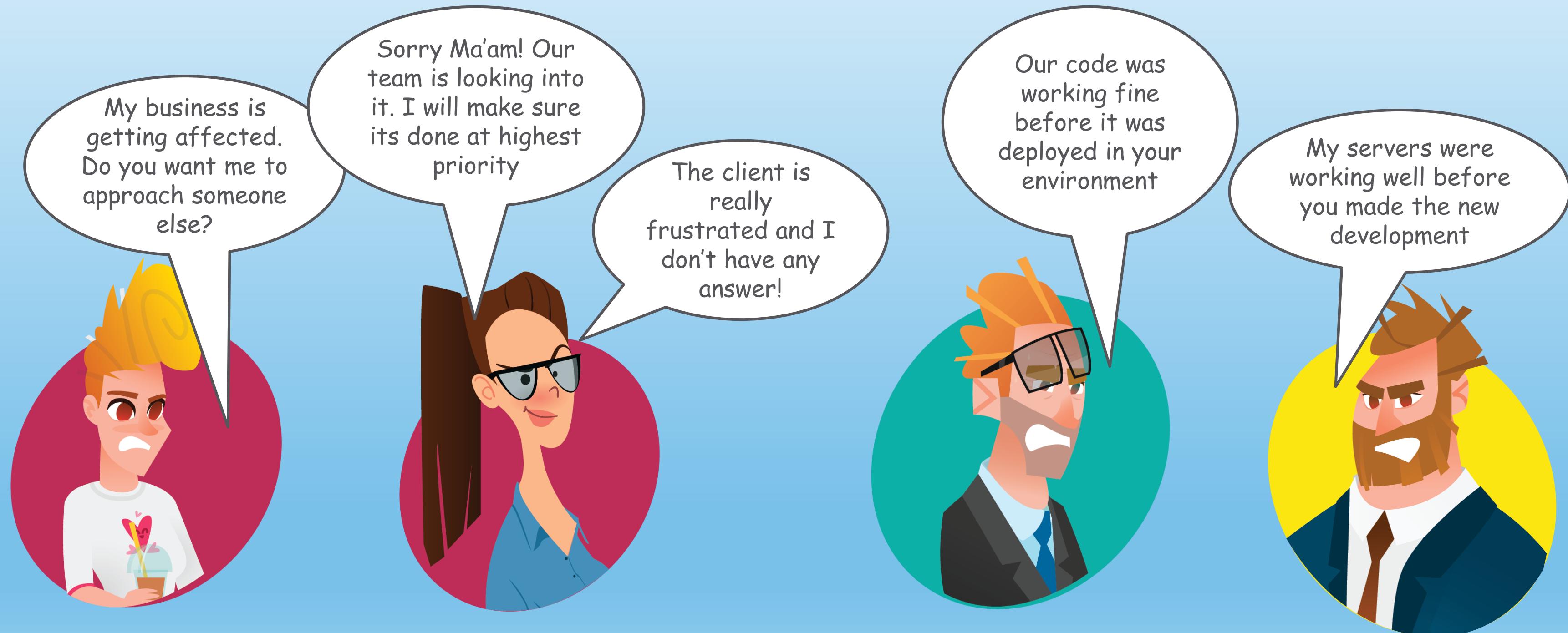
Agile Meets Deployment

Hunting for the defaulters begin with Client's Escalation...



Lack Of Coordination In Teams

Lack of Transparency and Coordination kick off the blame game and has left the Customer Furious....



Agile Development Without DevOps Culture



Release and Deploy Mismatch



Unpredictable Issues



Blame Game



Lack of monitoring and Feedback

Agile Development With DevOps Culture



Streamlined Deliveries



Continuous Monitoring and Feedback



Team Work in Collaboration

Traditional Development Models: Waterfall Model

“ Waterfall Model: New phase in the development process begins only if the previous phase is complete ”



Waterfall Model Challenges: Developers



Developers

- Huge waiting time for code deployment
- Pressure of work on old, pending and new code



Developers: What Needs To Be Done?

Developers

- Huge waiting time for code deployment
 - **Use system with little or no waiting time**
- Pressure of work on old, pending and new code
 - **Use system with updated code**



The Panacea - Developers

Dev Challenges	DevOps Solution
 <p>Waiting time for code deployment</p>	<ul style="list-style-type: none">▪ Continuous Integration ensures quick deployment, faster testing and speedy feedback mechanism
 <p>Pressure of work on old, pending and new code</p>	<ul style="list-style-type: none">▪ Since there is no waiting time to deploy the code. Hence the developer focuses on building the current code



Waterfall Model Challenges: Operations

Operations

- Difficult maintaining production environment uptime
- Ineffective tools for infrastructure automation management
- Continuous increase in the number of servers to be monitored



Operations: What Needs To Be Done?

Operations

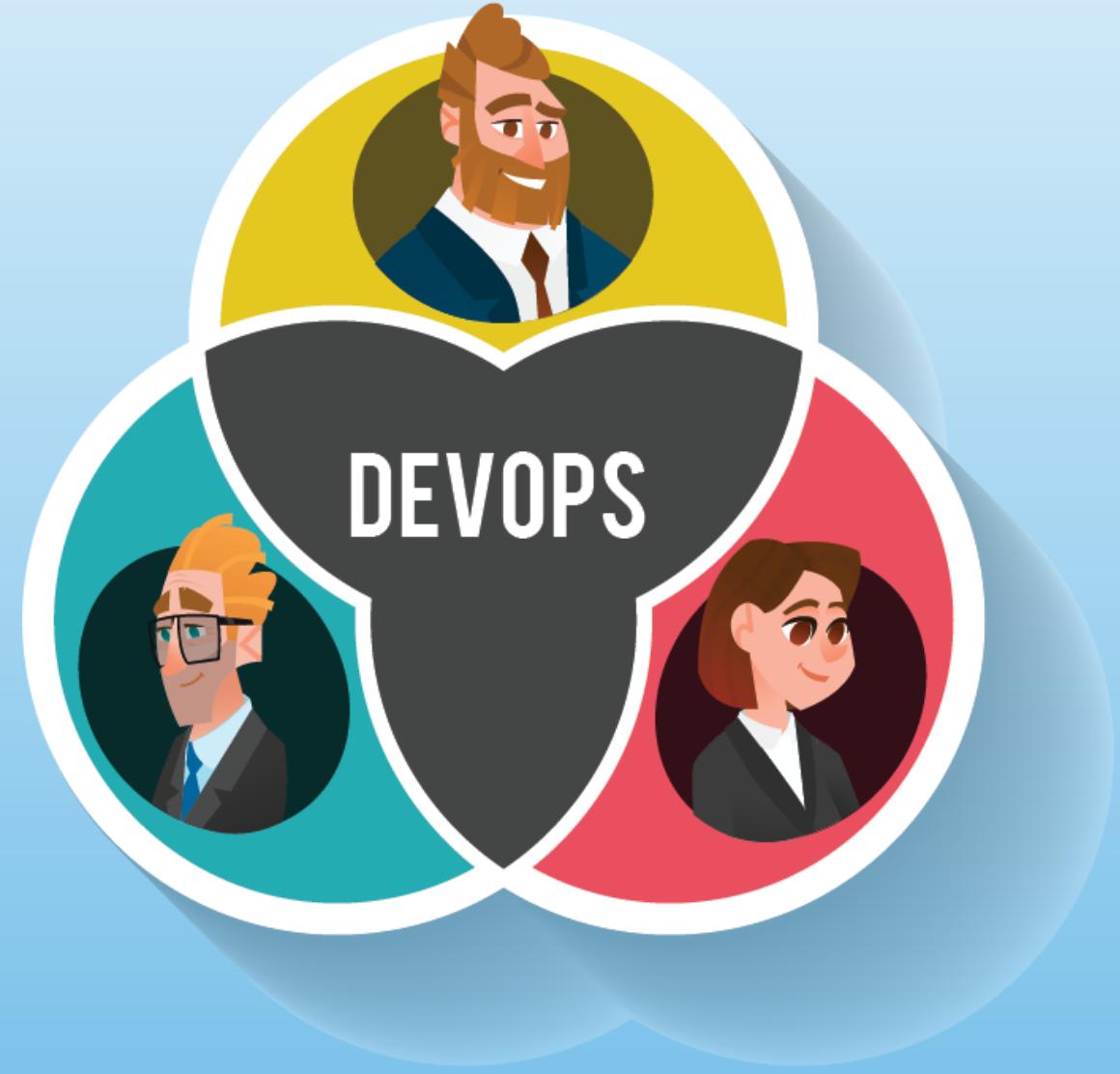
- Difficult maintaining production environment uptime
 - **Systems should be up and running most of the time**
- Ineffective tools for infrastructure automation management
 - **System required for easy administration**
- Continuous increase in the no. of servers to be monitored
 - **Effective monitoring and feedbacks system should be established**



The Panacea - Operations

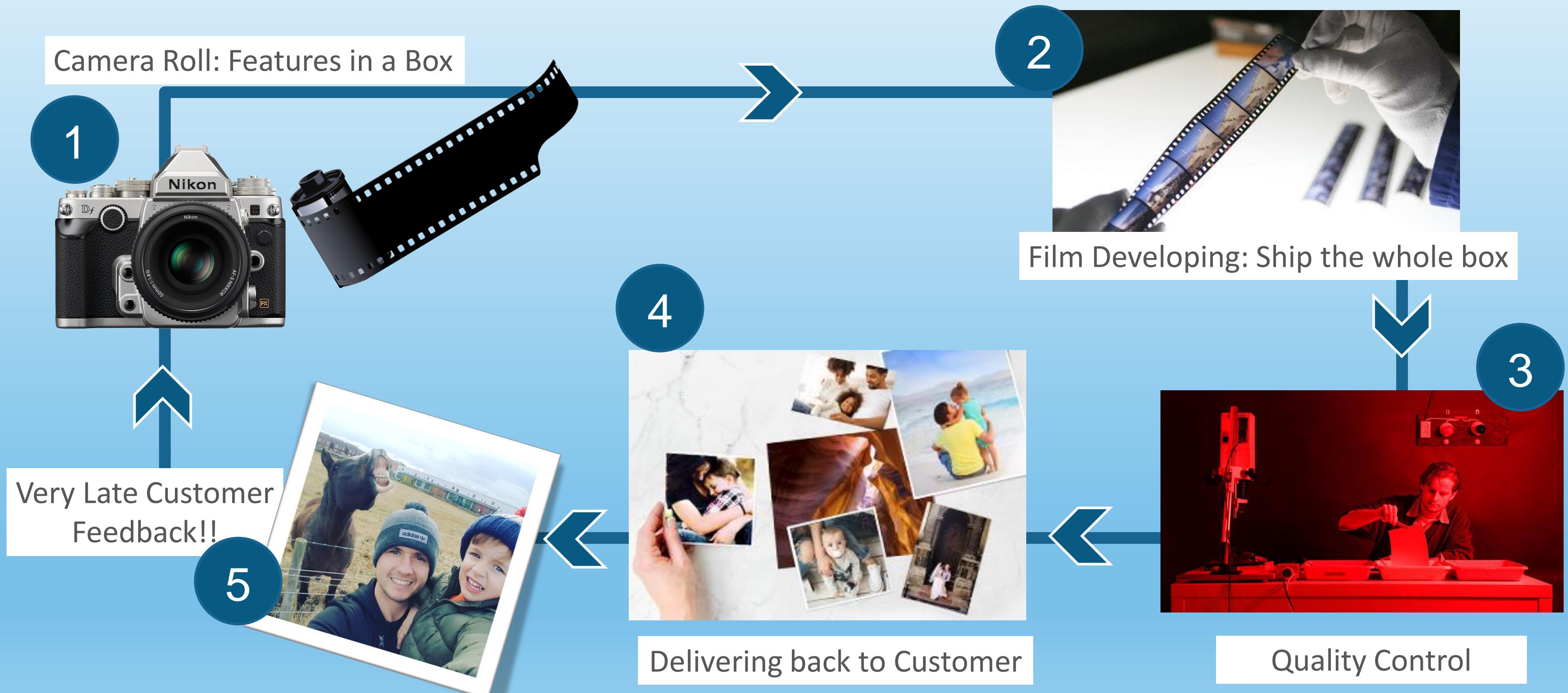
Ops Challenges	DevOps Solution
 <ul style="list-style-type: none">Difficult to maintain uptime of the production environment	<ul style="list-style-type: none">Containerization / Virtualization ensures simulated environment to run the software as containers in turn offering great reliability for service uptime
 <ul style="list-style-type: none">Tools to automate infrastructure management are not effective	<ul style="list-style-type: none">Configuration Management helps you to organize and execute configuration plans, consistently provision the system, and proactively manage their infrastructure
 <ul style="list-style-type: none">No. of servers to be monitored increases	<ul style="list-style-type: none">Continuous Monitoring
 <ul style="list-style-type: none">Difficult to diagnose and provide feedback on the product	<ul style="list-style-type: none">Effective monitoring and feedbacks system is established through NagiosThus effective administration is assured

DevOps: One Solution To All



Dev + Ops + QA = DevOps

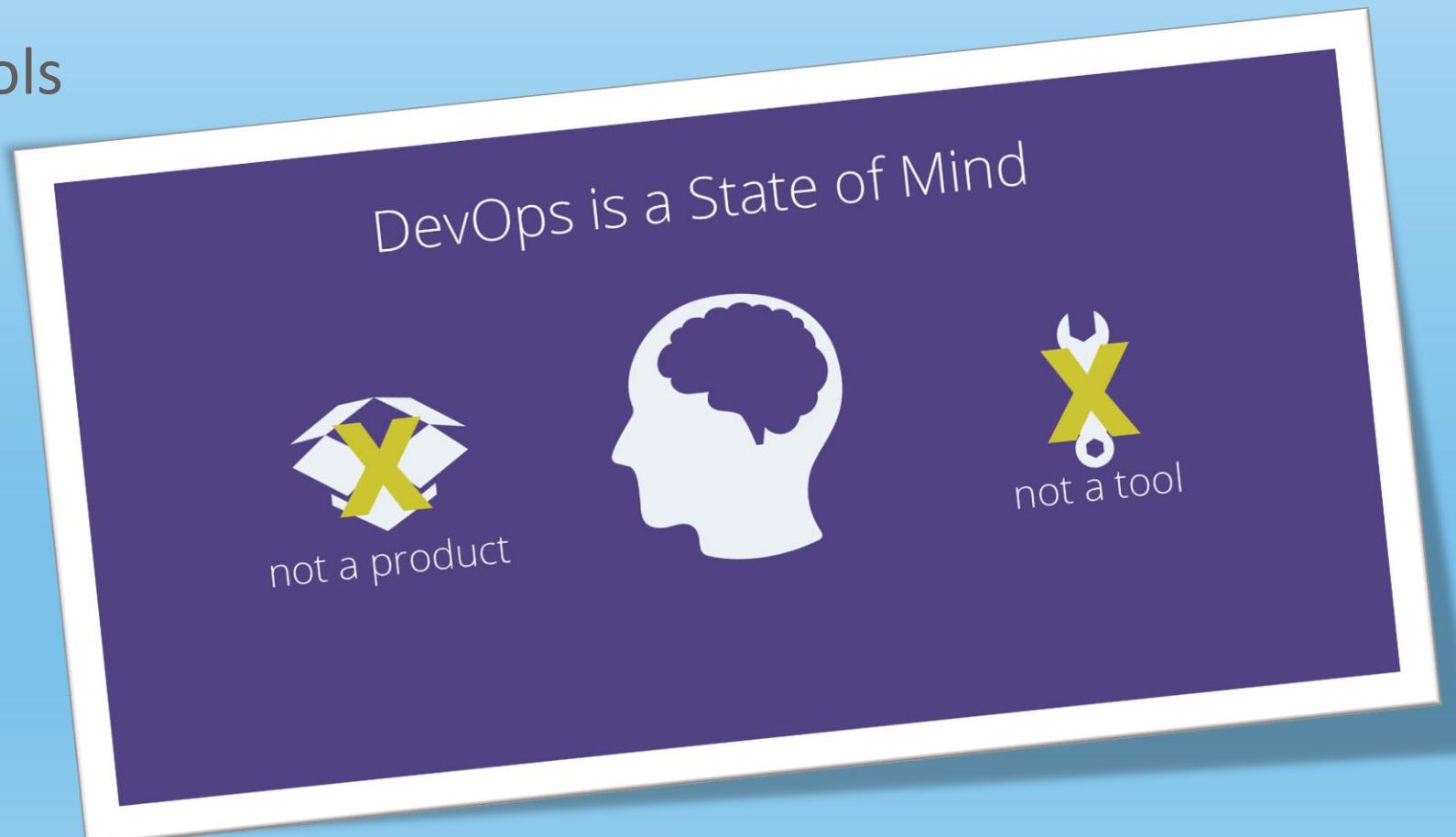
So What Exactly Is DevOps In Real Life?



DevOps Definition: What DevOps Is NOT!!

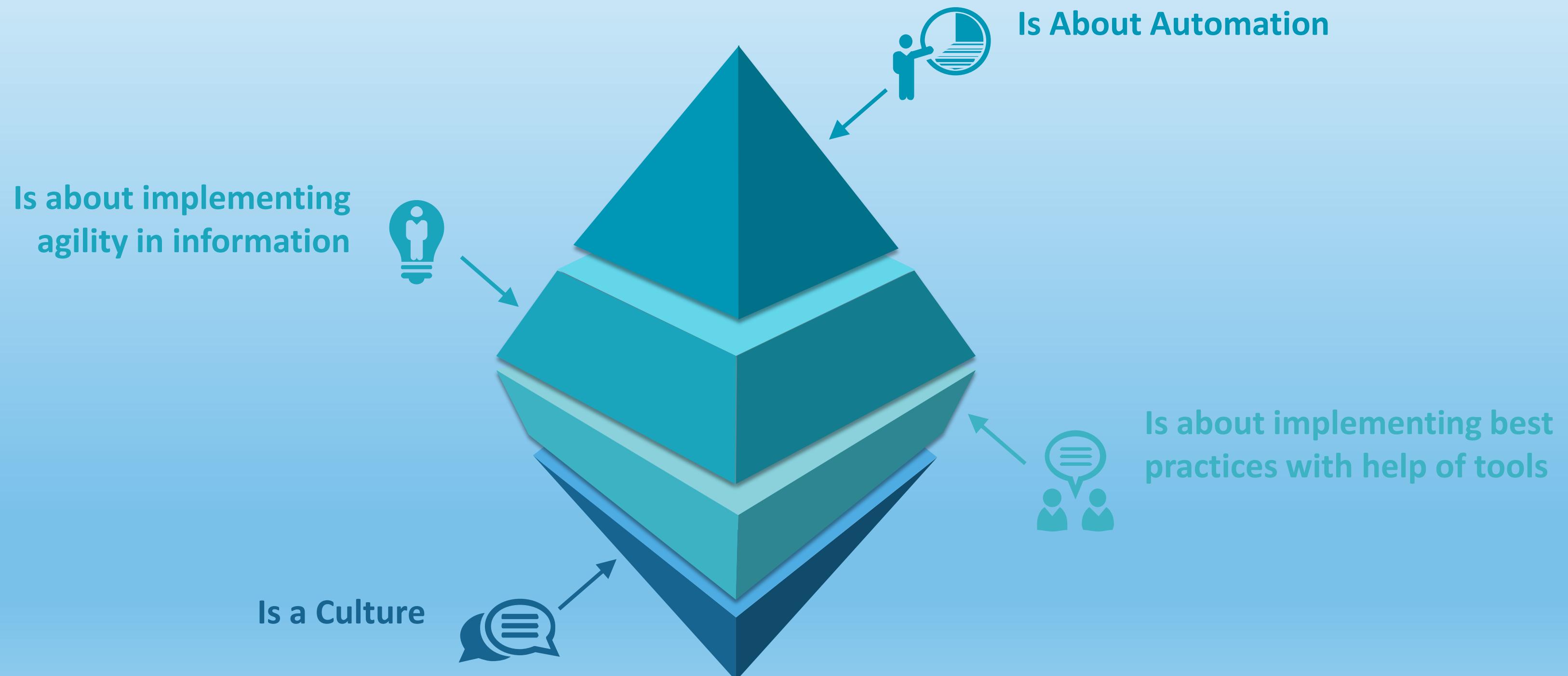
DevOps: Most Common Misunderstanding

- DevOps is not a role, person or organization
- DevOps is not a separate team
- DevOps is not a product or a tool
- DevOps is not about just writing scripts or implementing tools



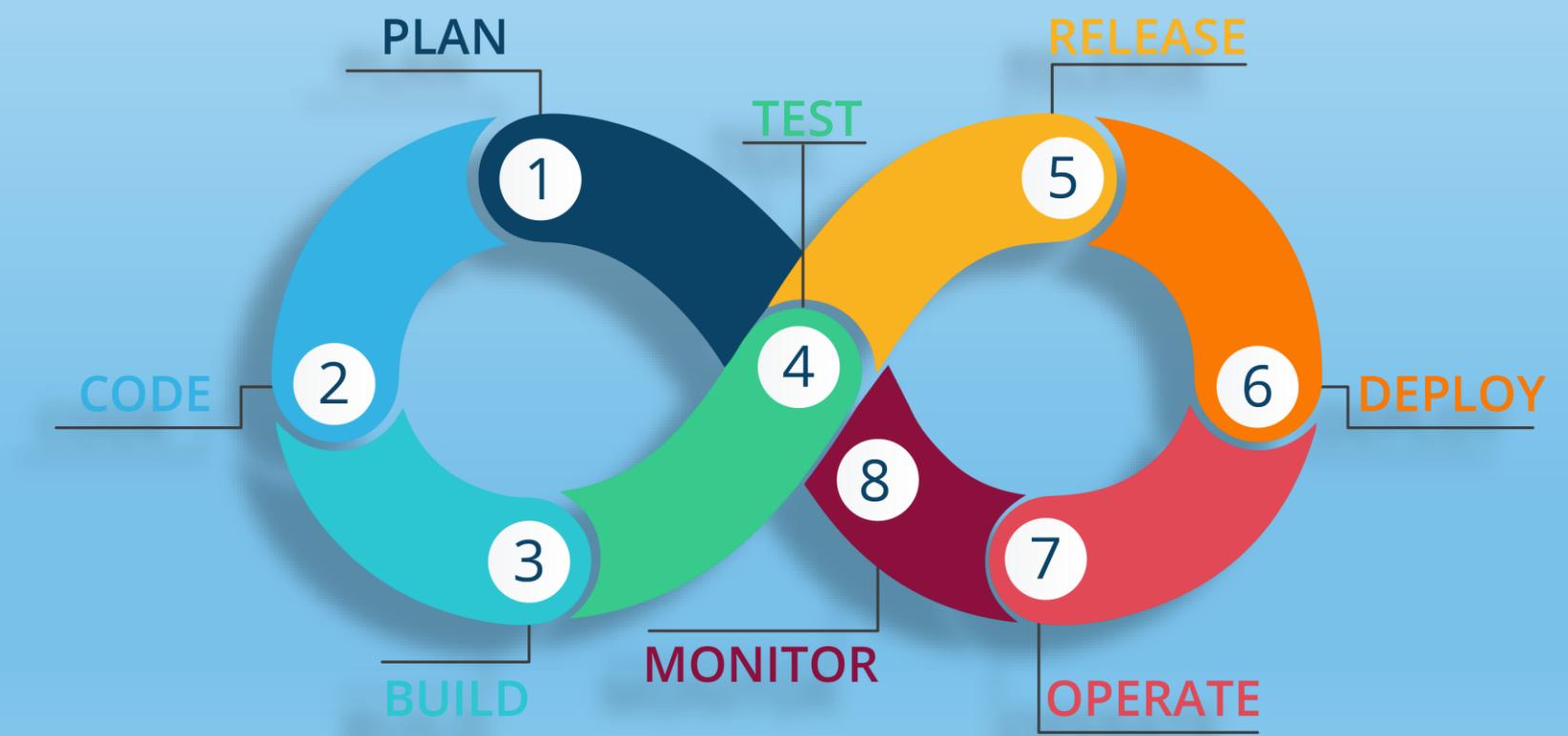
What Is DevOps?

“DevOps is not a goal, but a never-ending process of continual improvement.”

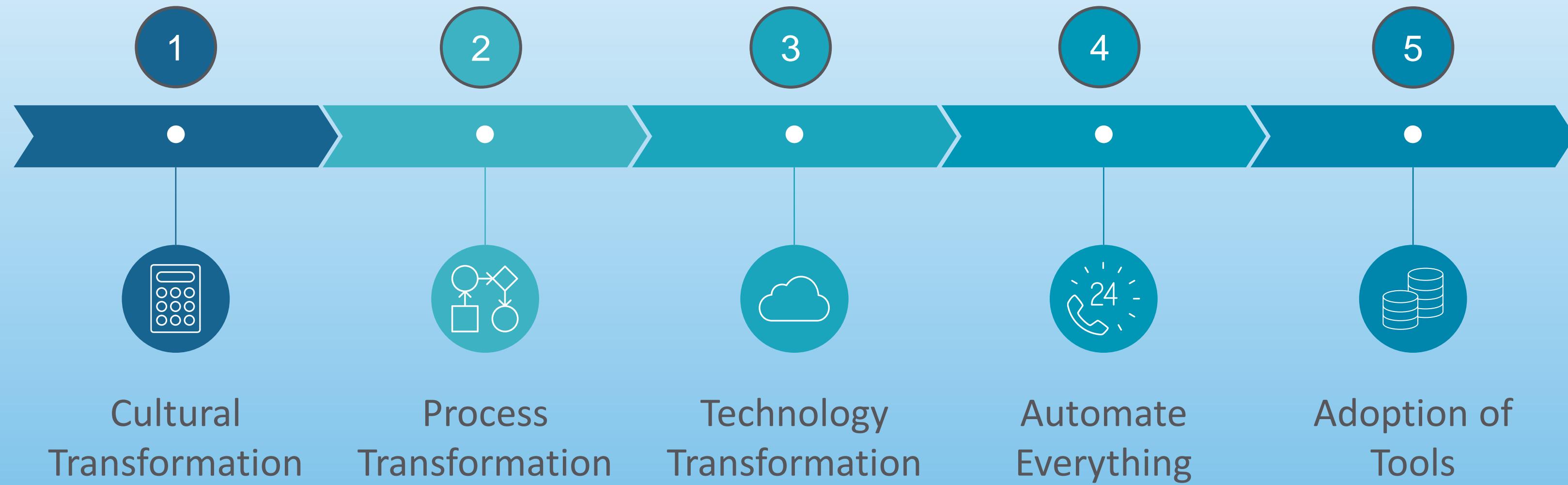


What Does DevOps Do?

- Integrates **developers** and **operations** teams
- Improves collaboration and productivity by:
 - Automating infrastructure
 - Automating workflows
 - Continuously measuring application performance



DevOps Adoption: 5 Step Journey



Agile Vs Waterfall Vs DevOps

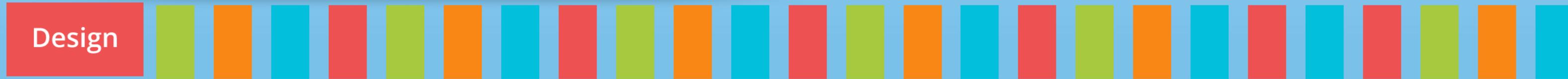
Waterfall



Agile

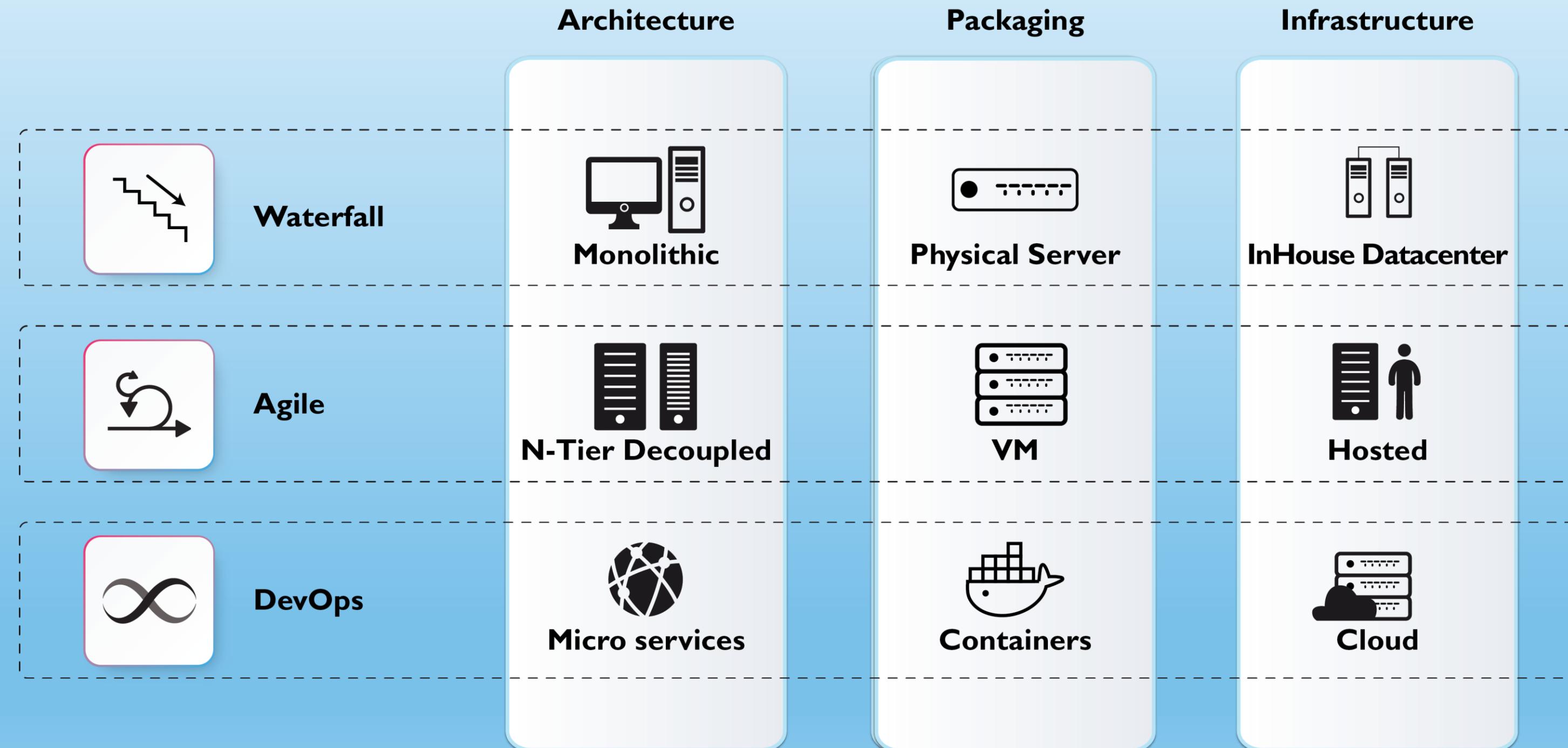


Agile with Continuous Deploy



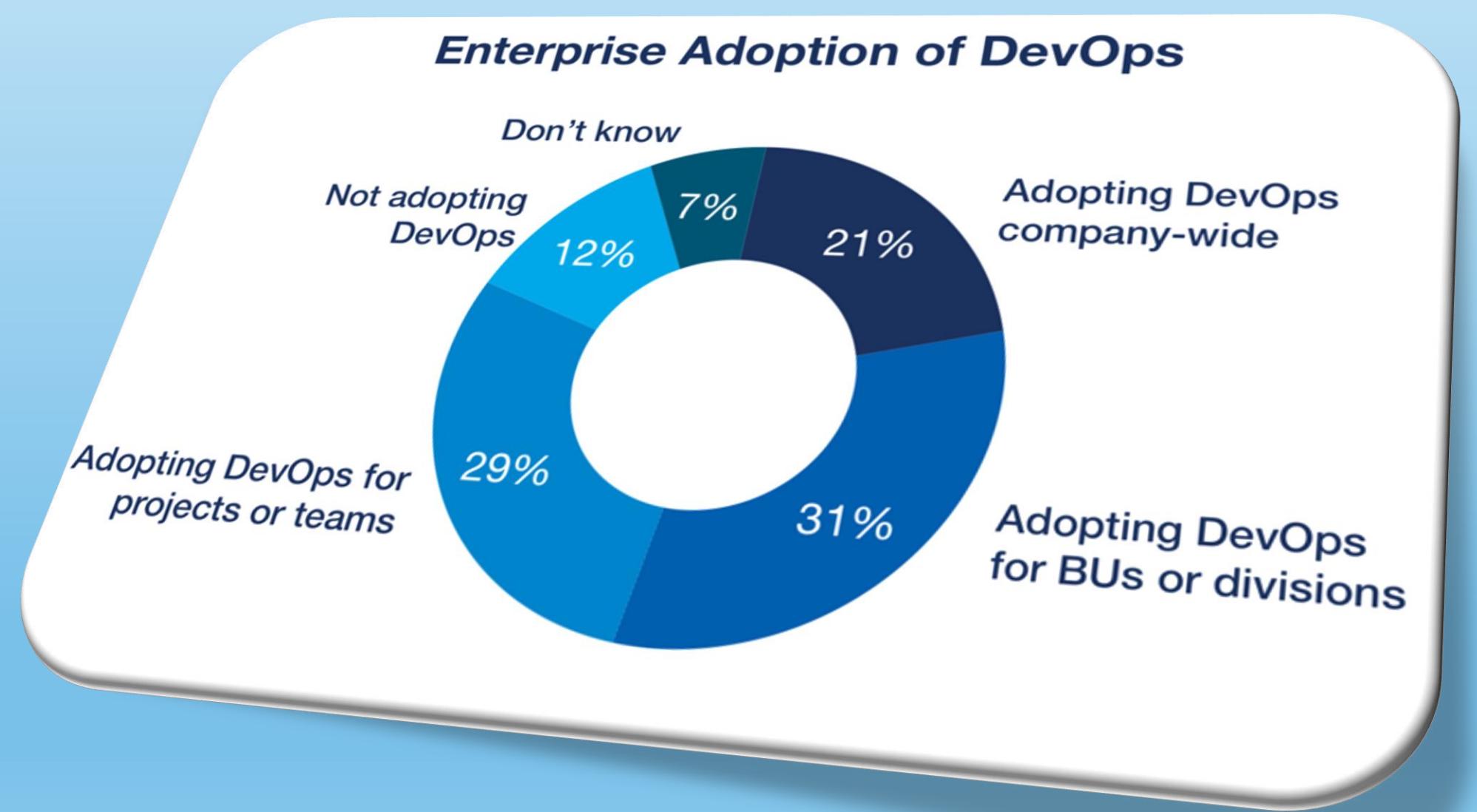
Continuous Deploy requires DevOps

Reactive And Scalable Infrastructure With DevOps



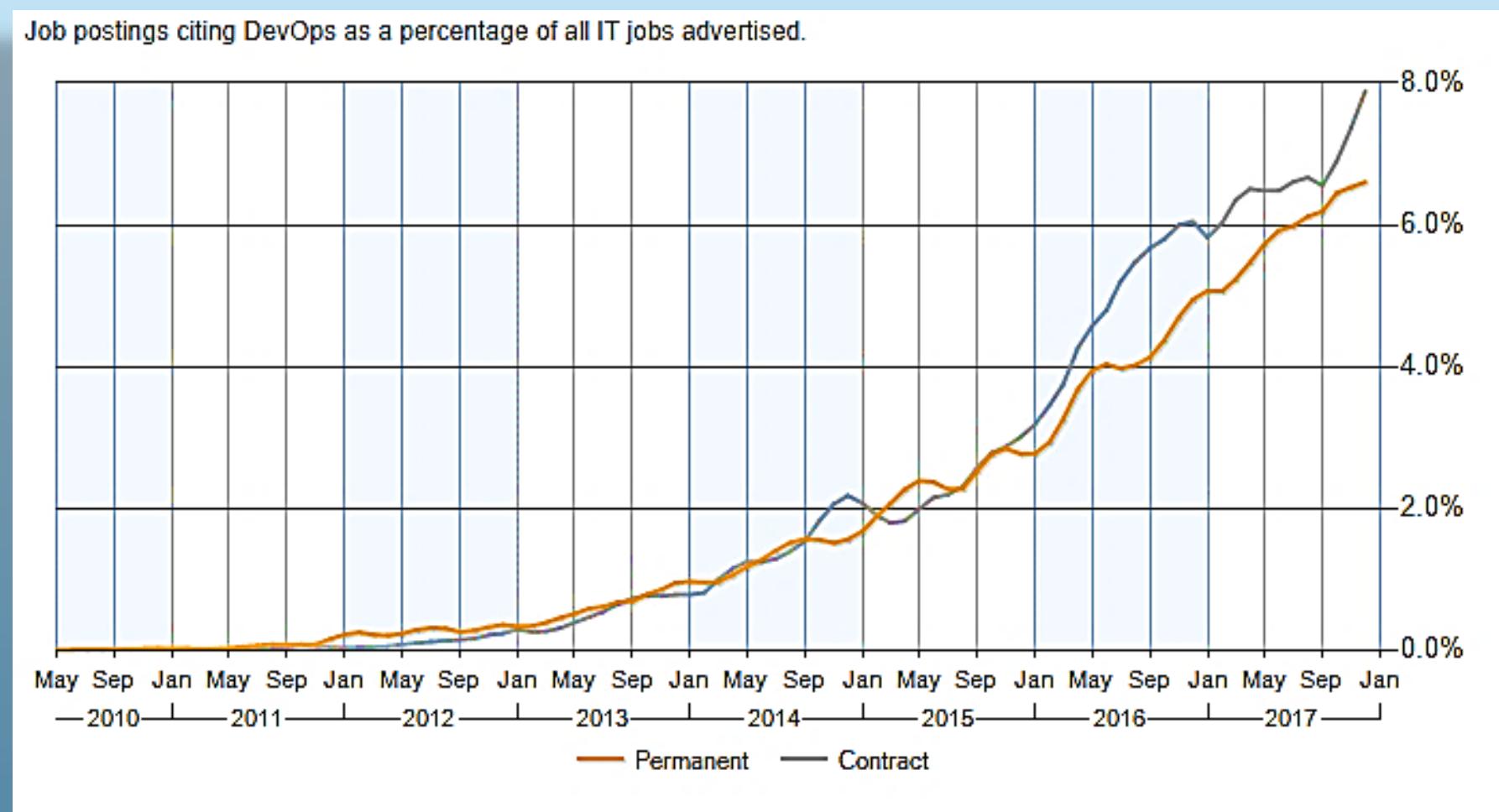
DevOps Current Scenario

“Considering the changing pace of IT landscape,
almost all the companies require fast paced
development environment”



DevOps Current Scenario: Job & Salary Trend

“The demand trend of job that features DevOps Engineer in the job title”



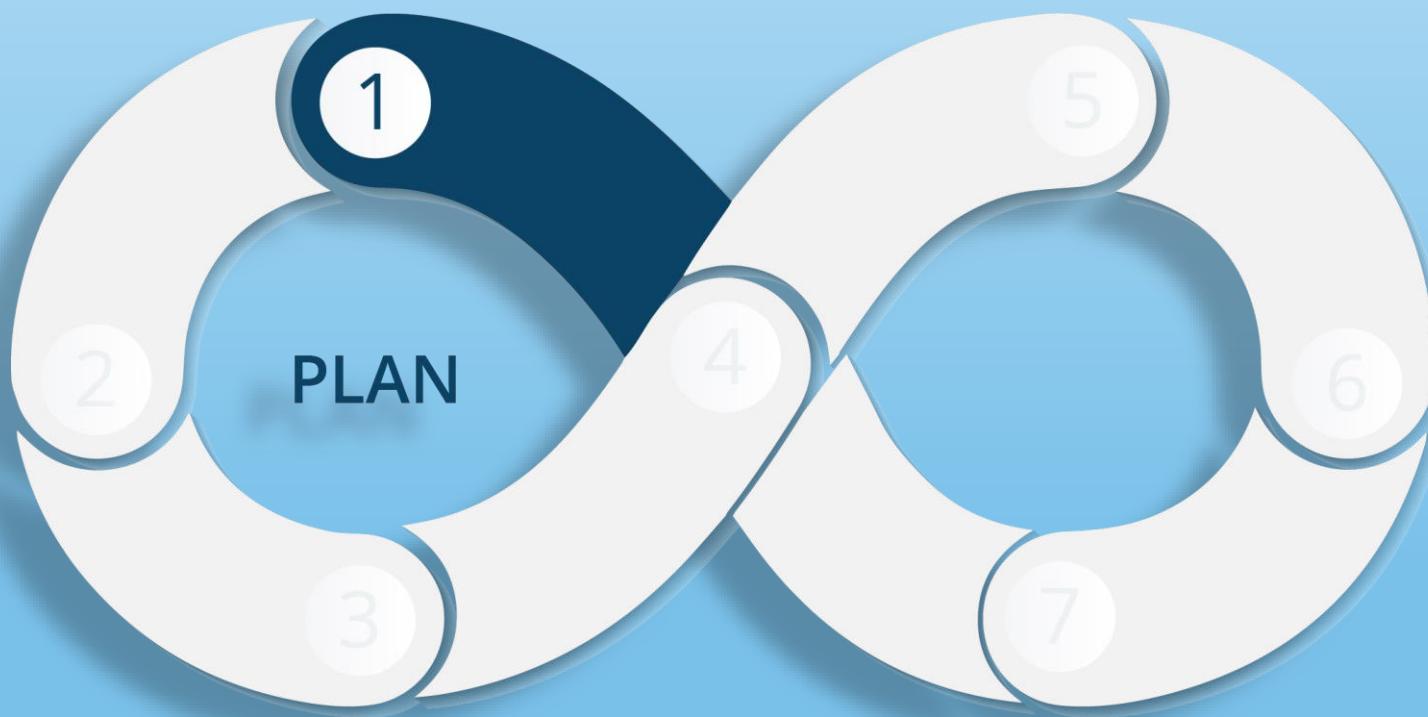
According to PayScale.com, salaries for professionals working on DevOps related jobs in California are upwards of \$94,000

Skills Of A DevOps Engineer

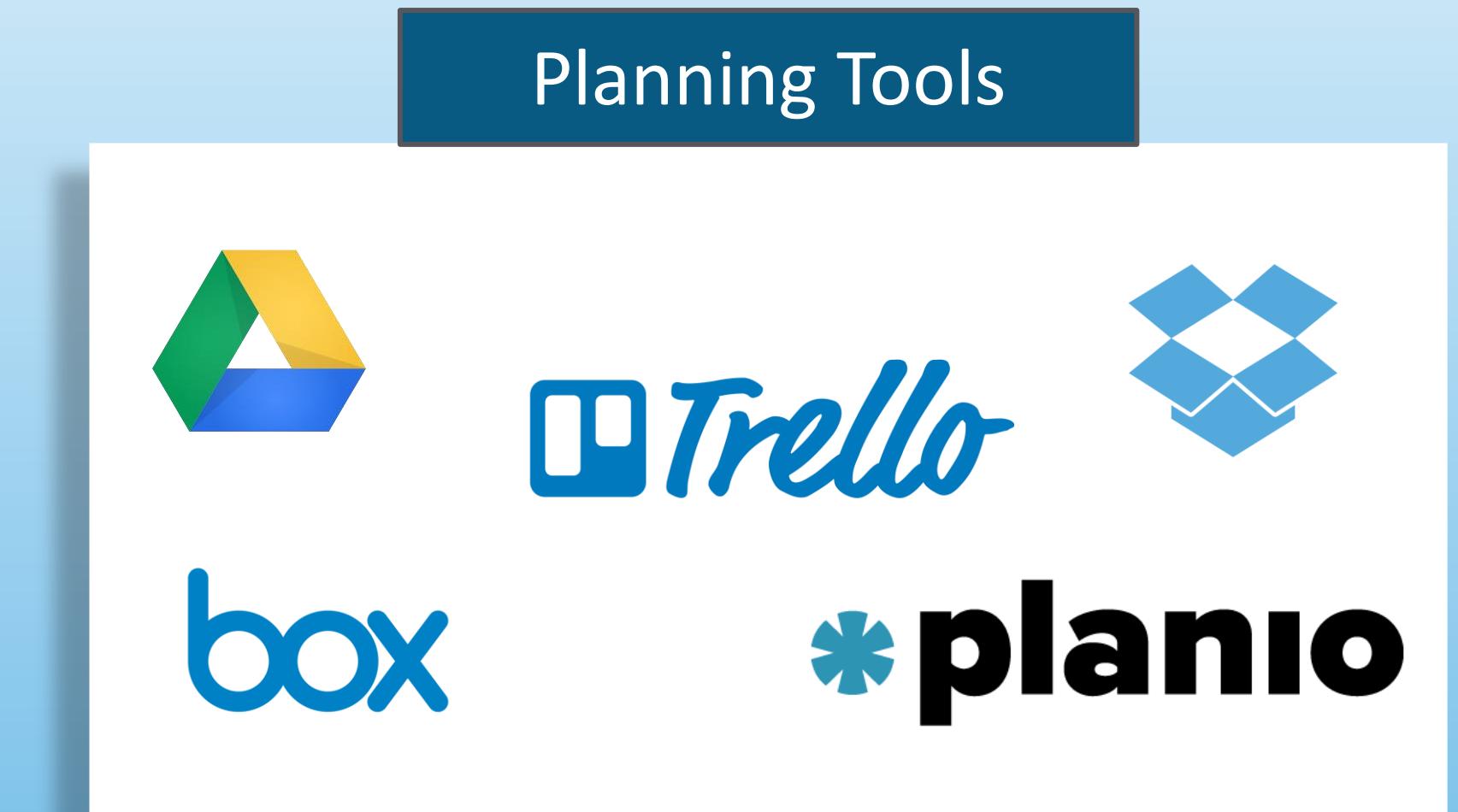
Skills	Description
Tools	<ul style="list-style-type: none">• Version Control - GIT• Continuous Integration - Jenkins• Virtualization/ Containerization – Docker/Kubernetes• Configuration Management – Puppet/Ansible• Monitoring – Nagios
Networking Skills	<ul style="list-style-type: none">• General networking skills – Establishing connection between the containers/Port Forwarding/ Container Orchestration
Other Skills	<ul style="list-style-type: none">• People Skills• Process Skill• Customer Skill and Empathy• Cloud Awareness

DevOps Lifecycle: Plan

“First stage of DevOps cycle where you
Plan, Track, Visualize and Summarize
your Project before working/starting it.”

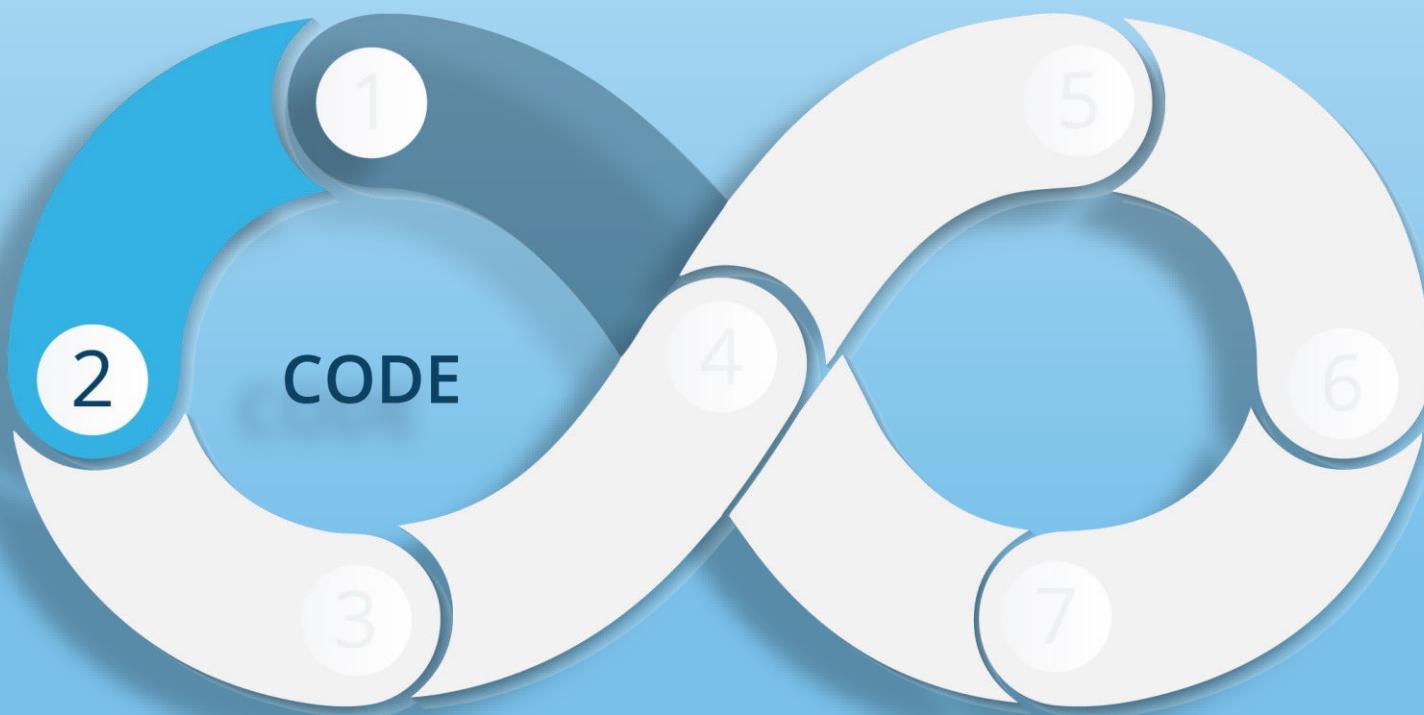


Planning Tools



DevOps Lifecycle: Code

“Second stage of DevOps cycle where
the developers write their code.”

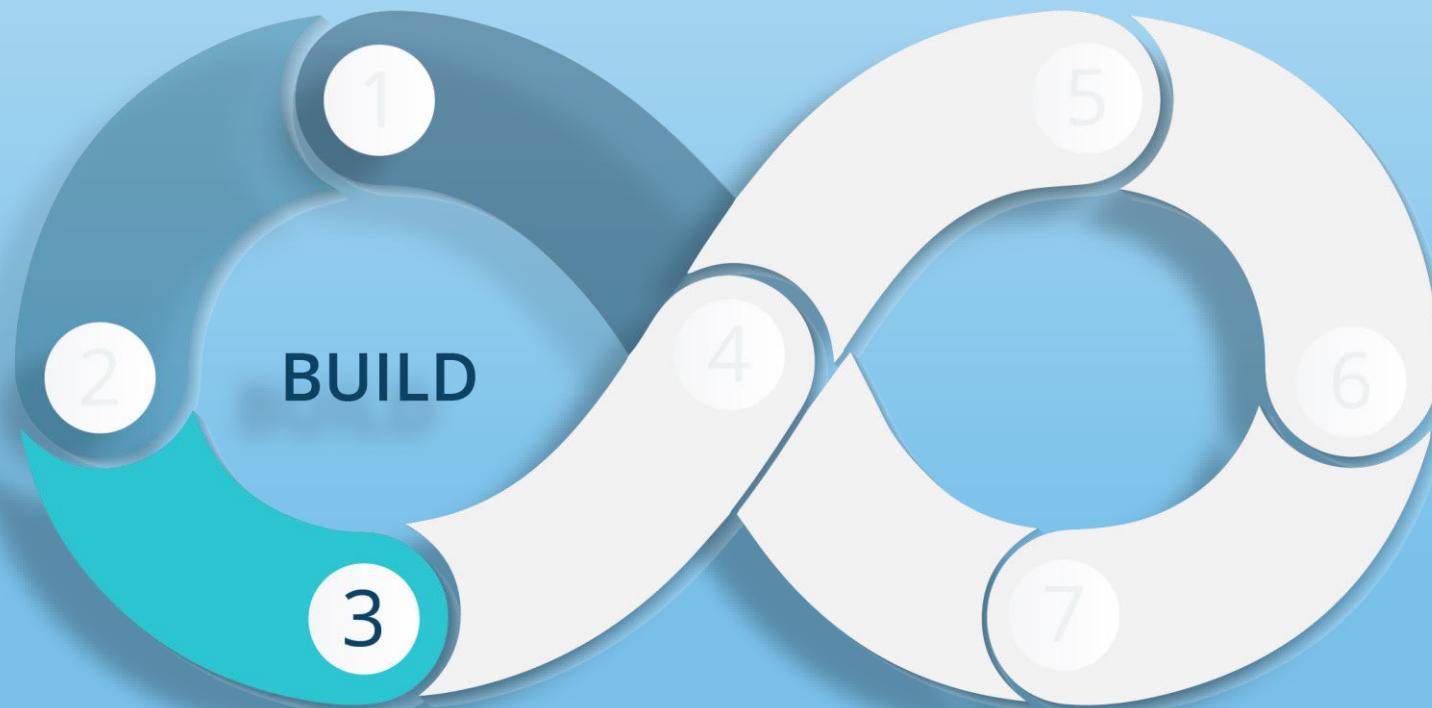


Coding Tools



DevOps Lifecycle: Build

“Build is a pre-release version and is identified by a build number, rather than by a release number”



Building Tools

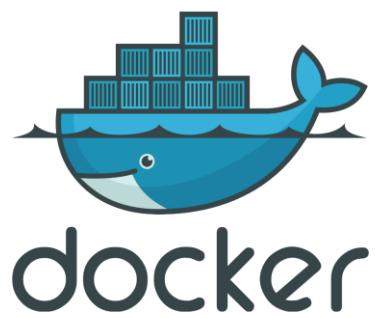


<APACHE ANT>



kubernetes

Maven™



docker



gradle
a better way to build

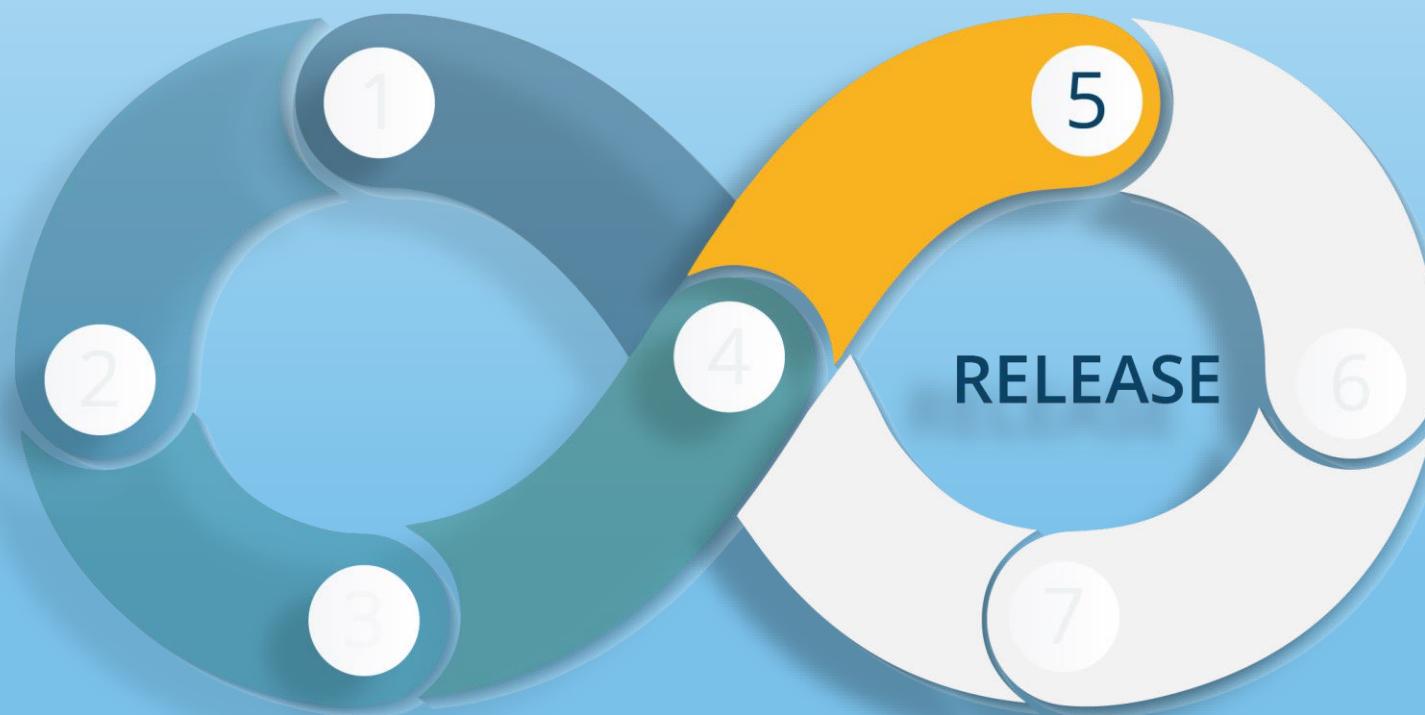
DevOps Lifecycle: Test

“ Process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release as rapidly as possible”

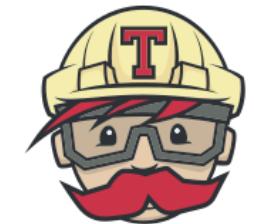


DevOps Lifecycle: Release

“This phase helps to integrate code into a shared repository using which, you can detect and locate errors quickly and easily”



Releasing Tools



Travis CI



Jenkins



CODESHIP



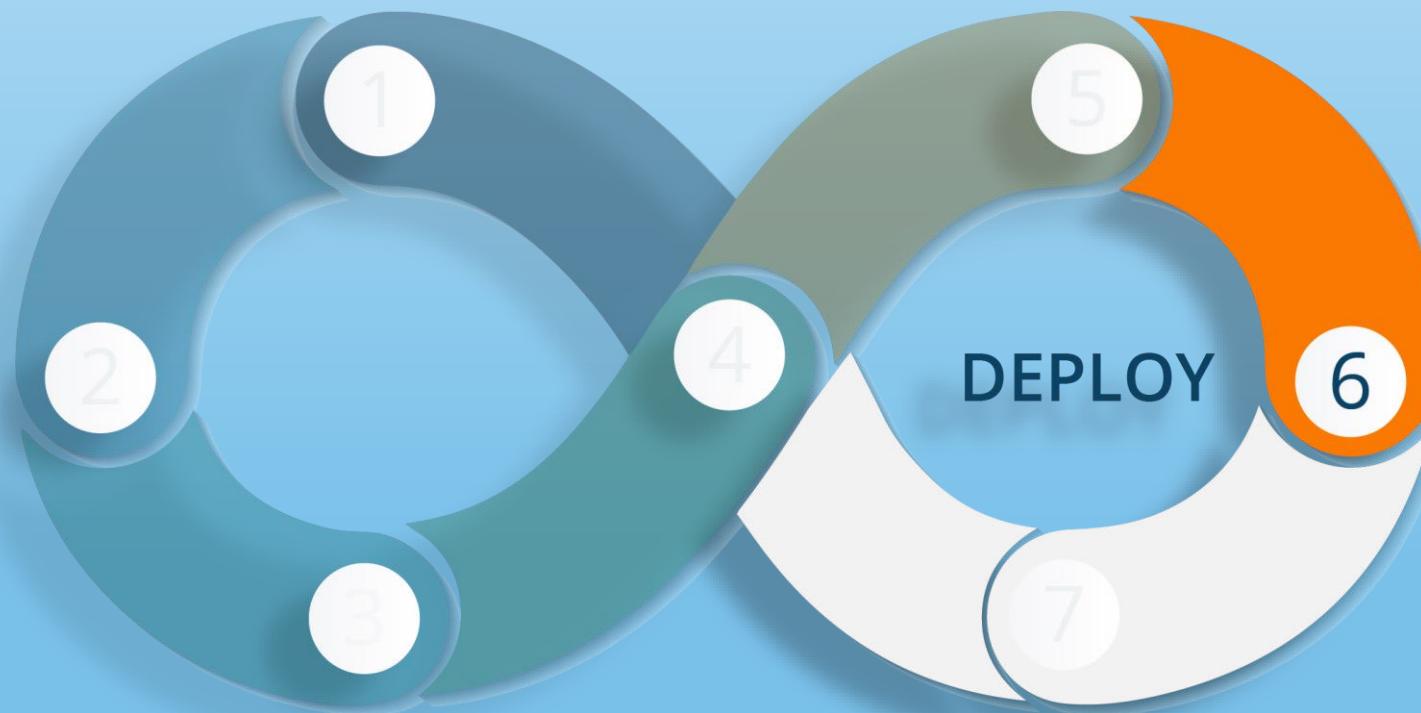
Bamboo



GitLab

DevOps Lifecycle: Deploy

“Manage and maintain development and deployment of software systems and servers in any computational environment”

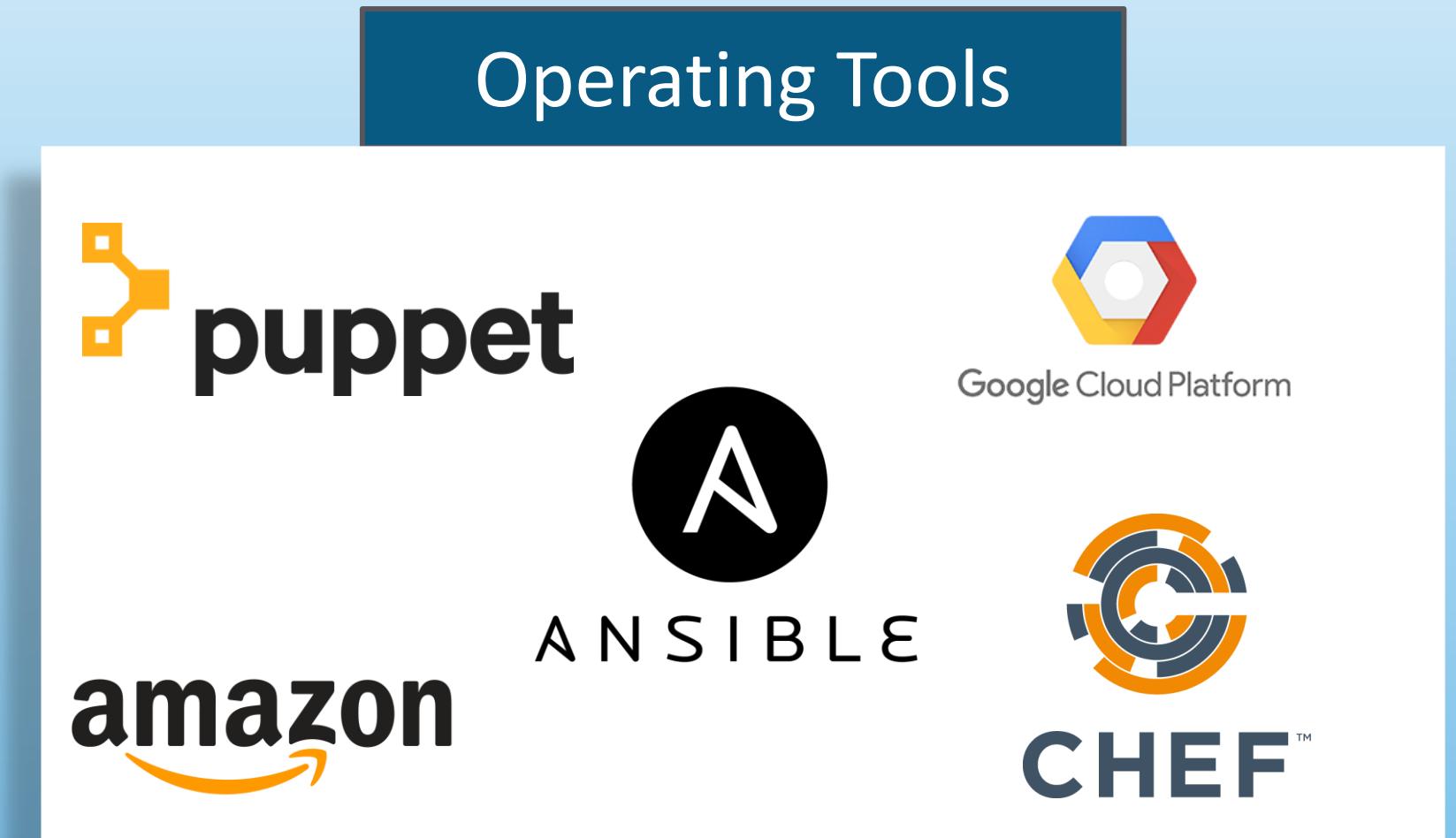
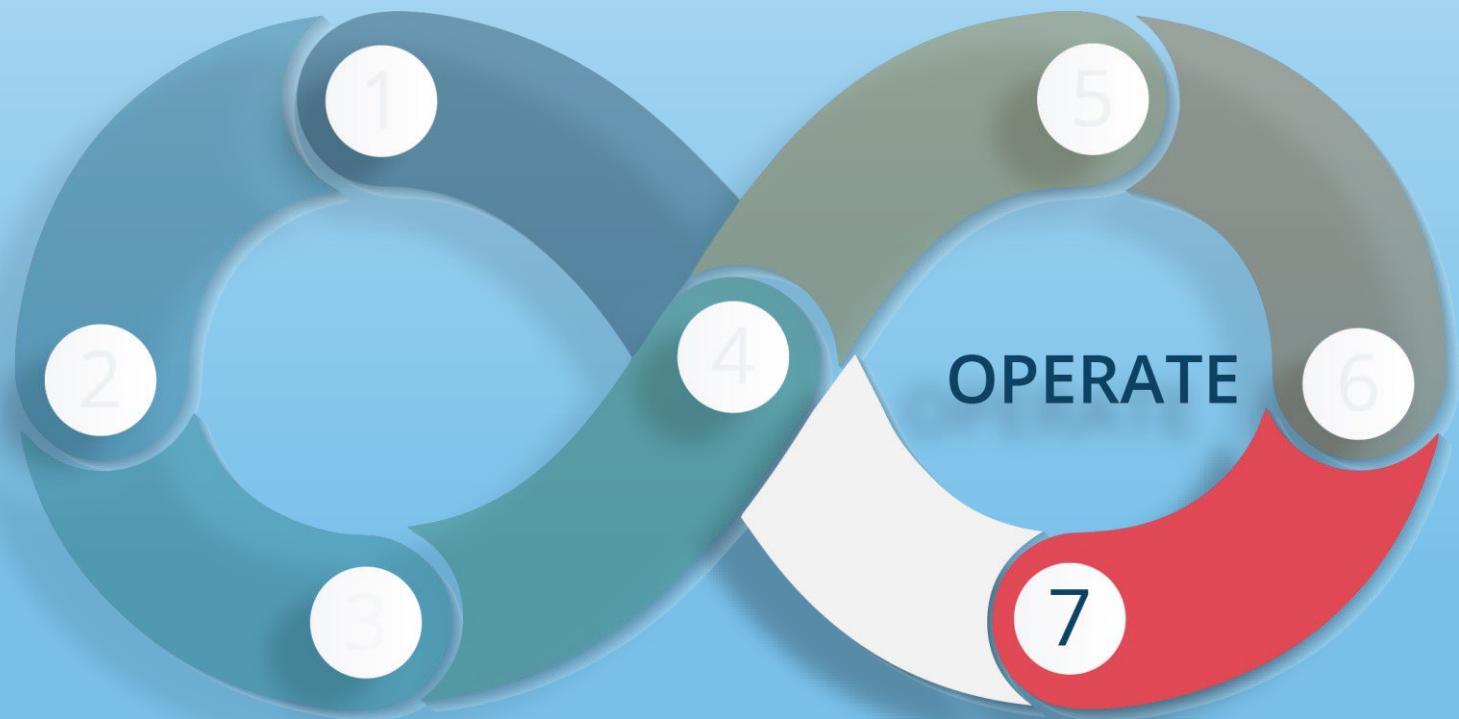


Deploying Tools



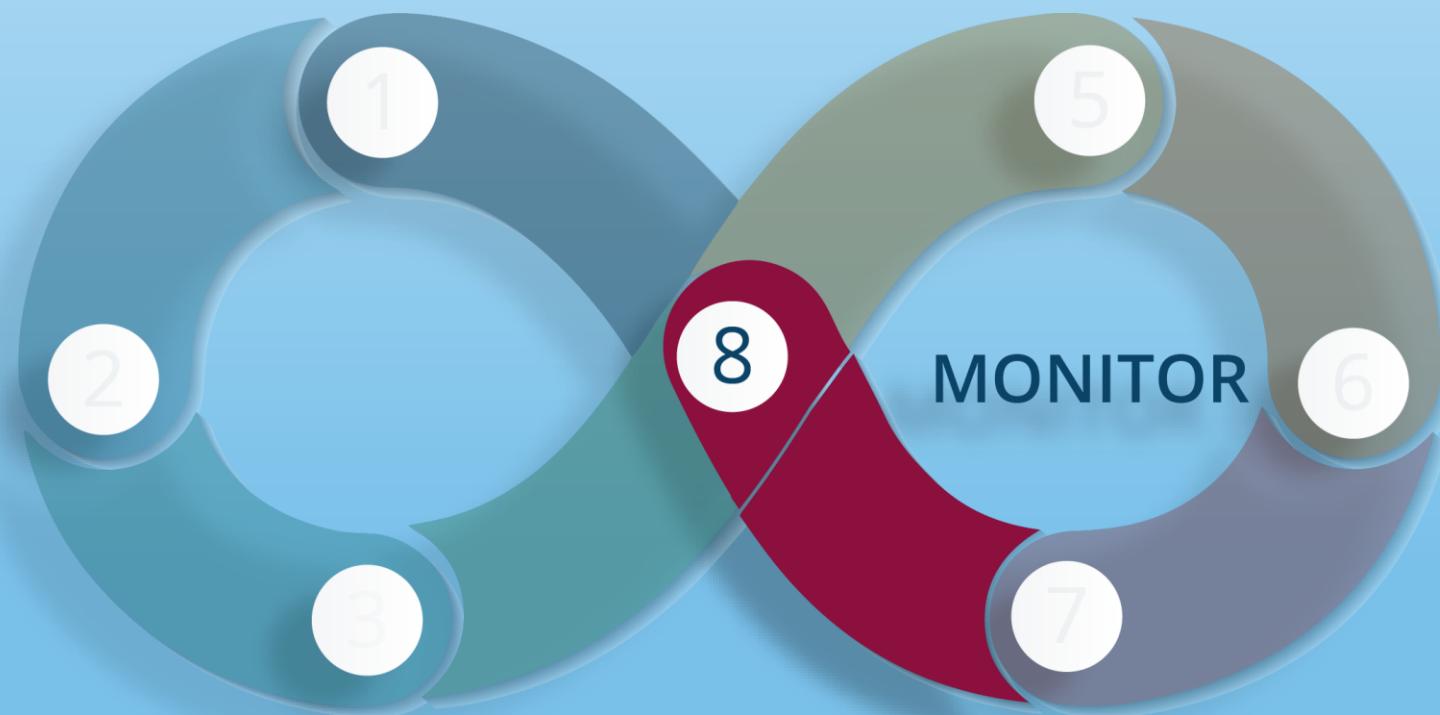
DevOps Lifecycle: Operate

“This phase is to keep the system upgraded with the latest update”



DevOps Lifecycle: Monitor

“It ensures that the application is performing as desired and the environment is stable. It quickly determines when a service is unavailable and understand the underlying causes”



Monitoring Tools

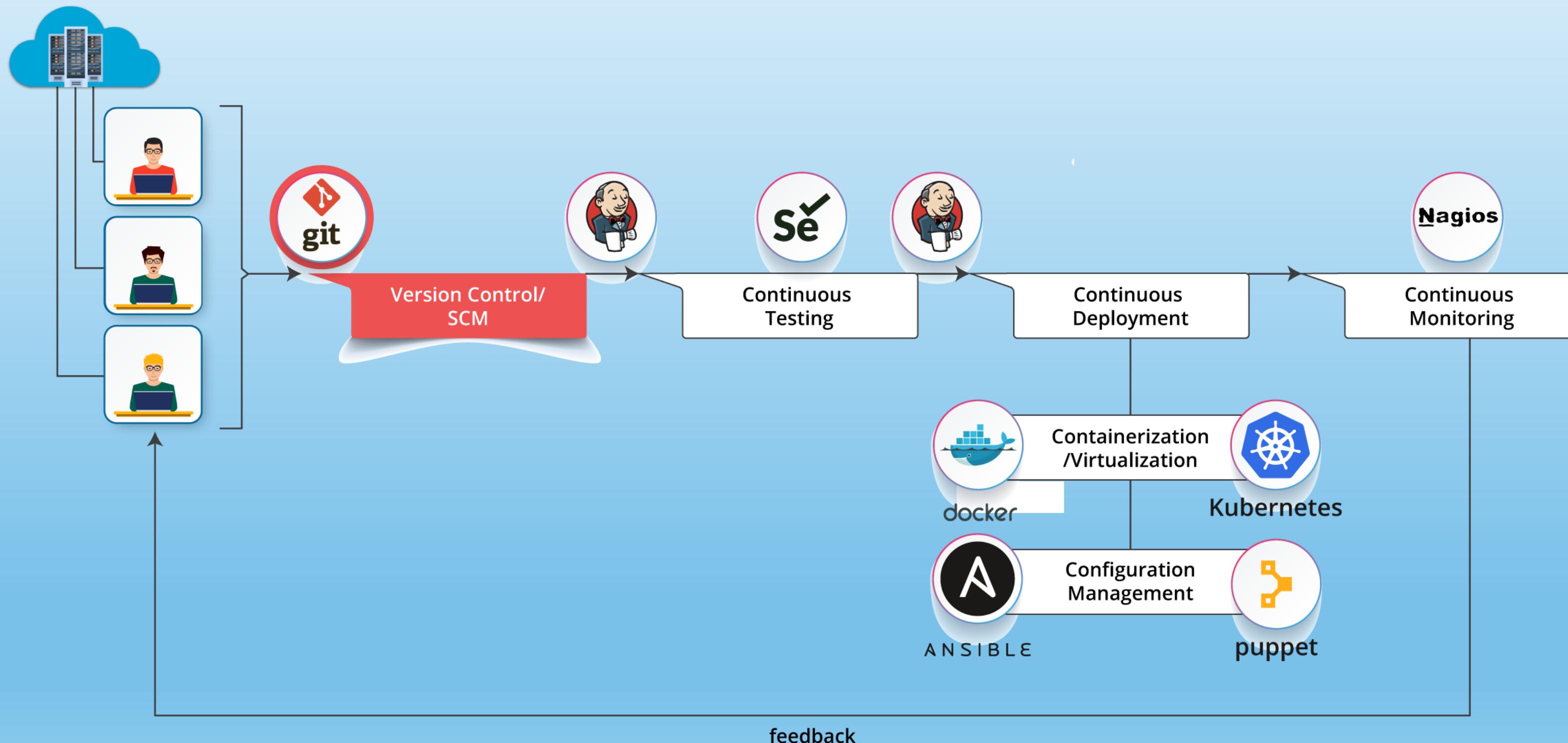
Nagios®



splunk®

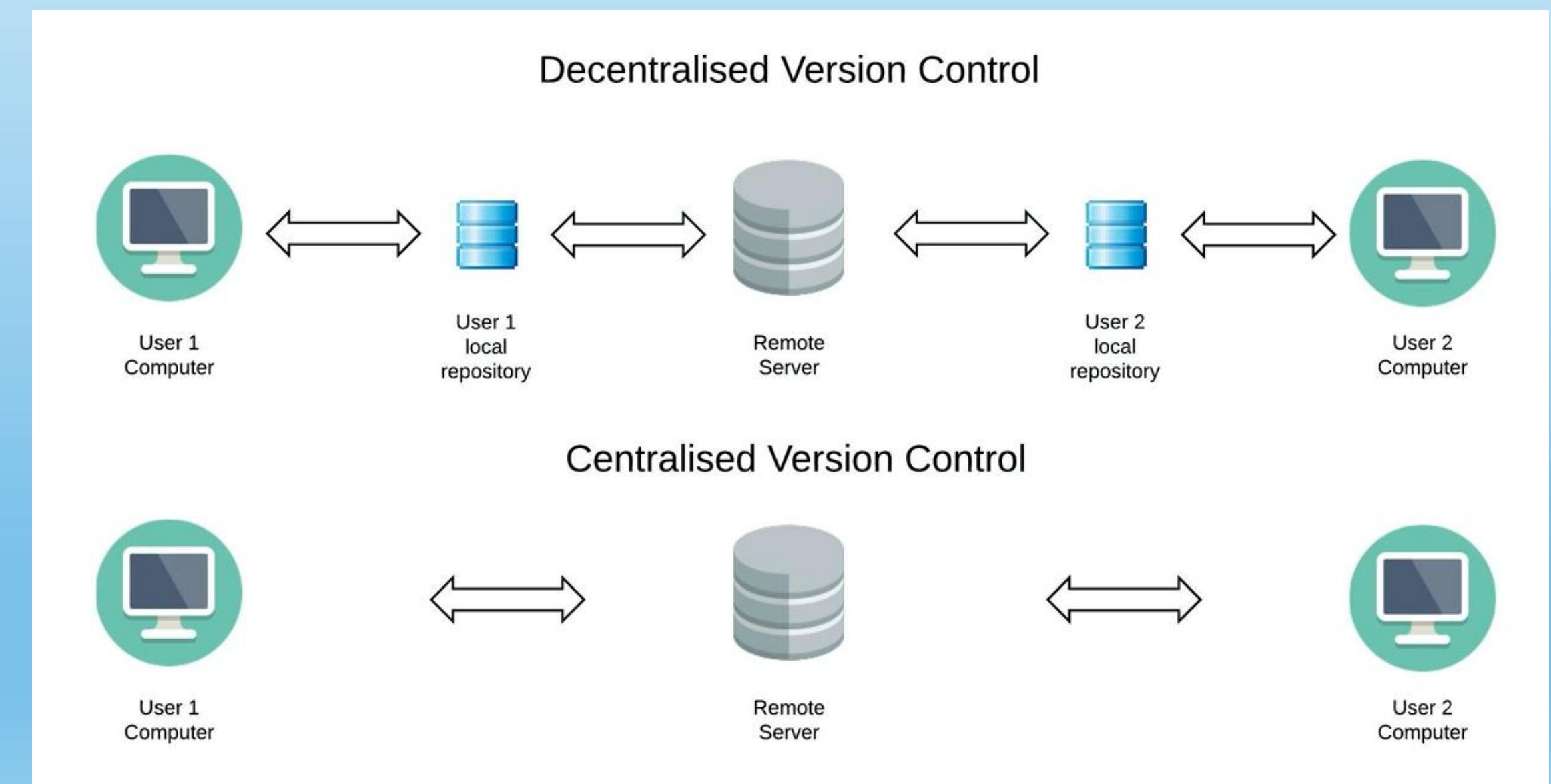


Version Control / Source Code Management

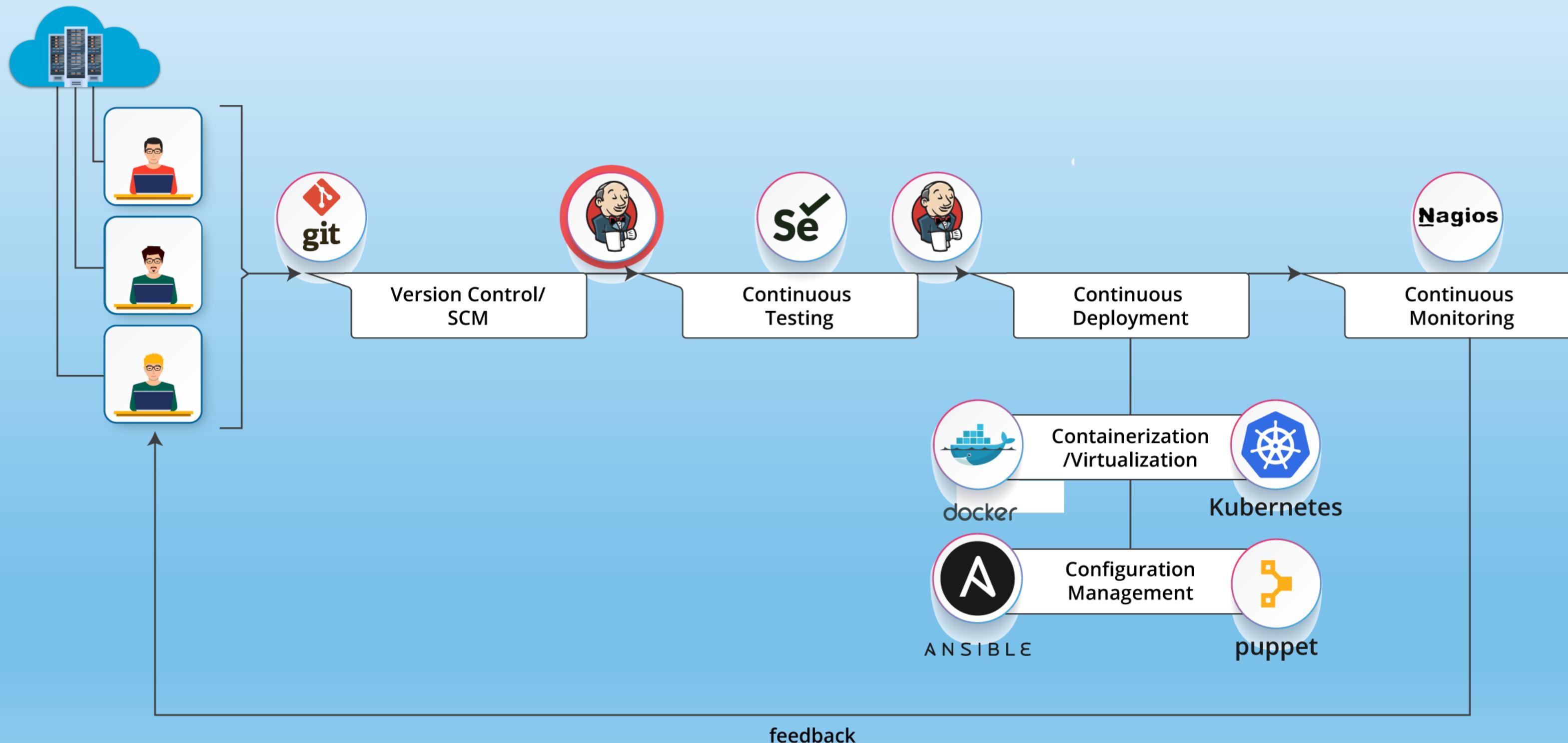


What Is Version Control?

System that records changes to documents, computer programs, large websites and other collection of information over time

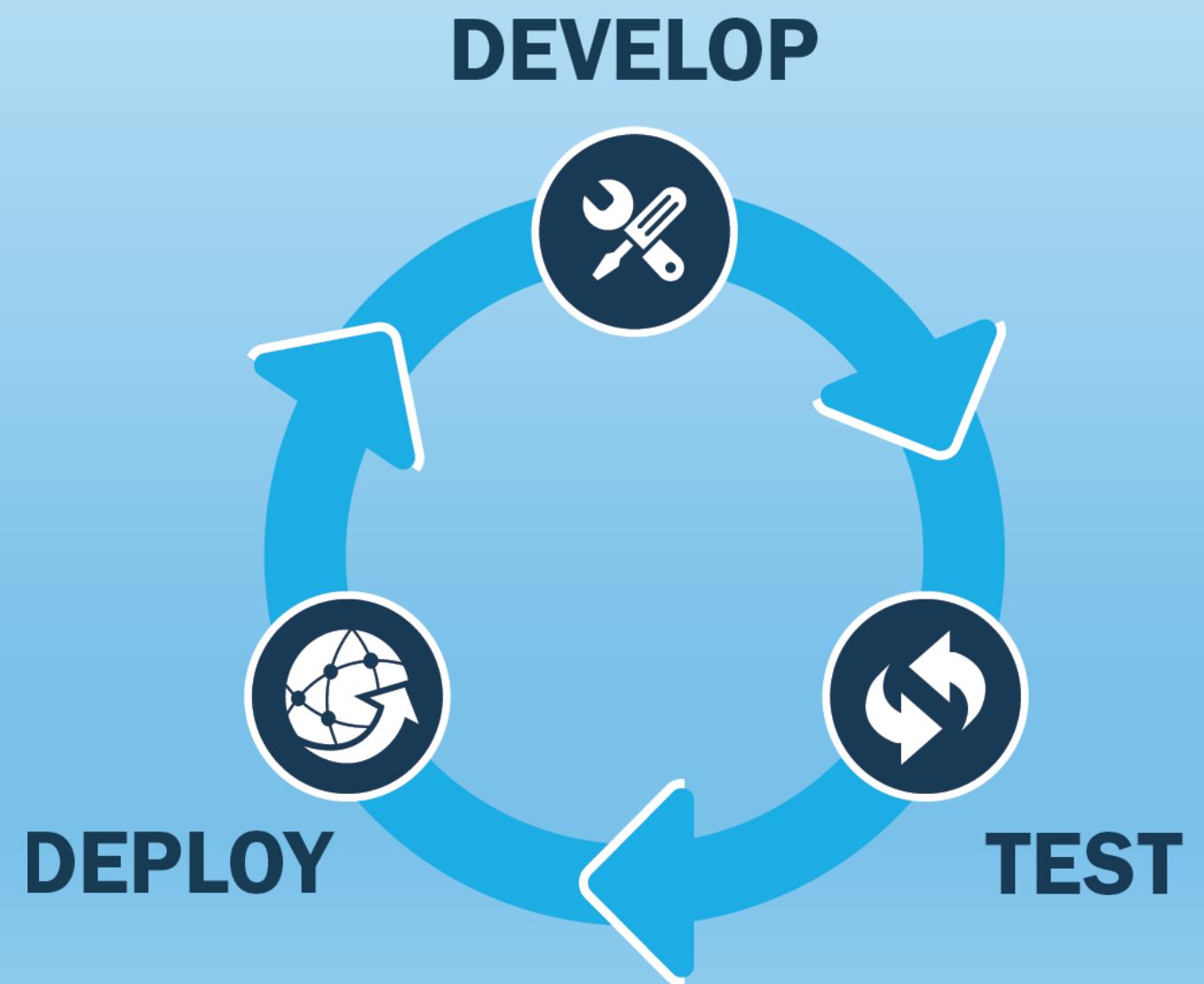


Continuous Integration

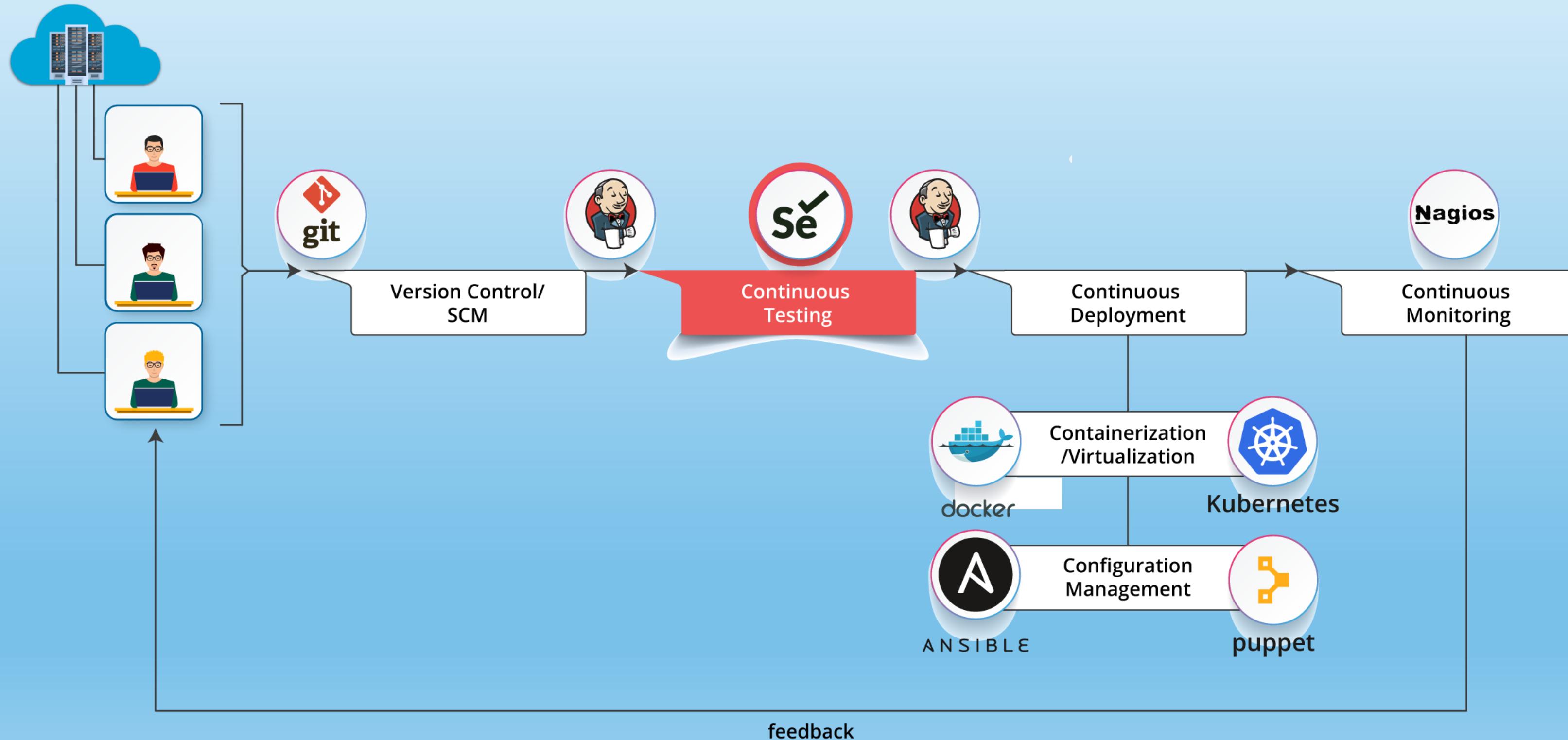


What Is Continuous Integration?

A development practice that requires developers to integrate code into a shared repository several times a day. By integrating regularly, you can detect errors quickly, and locate them more easily.

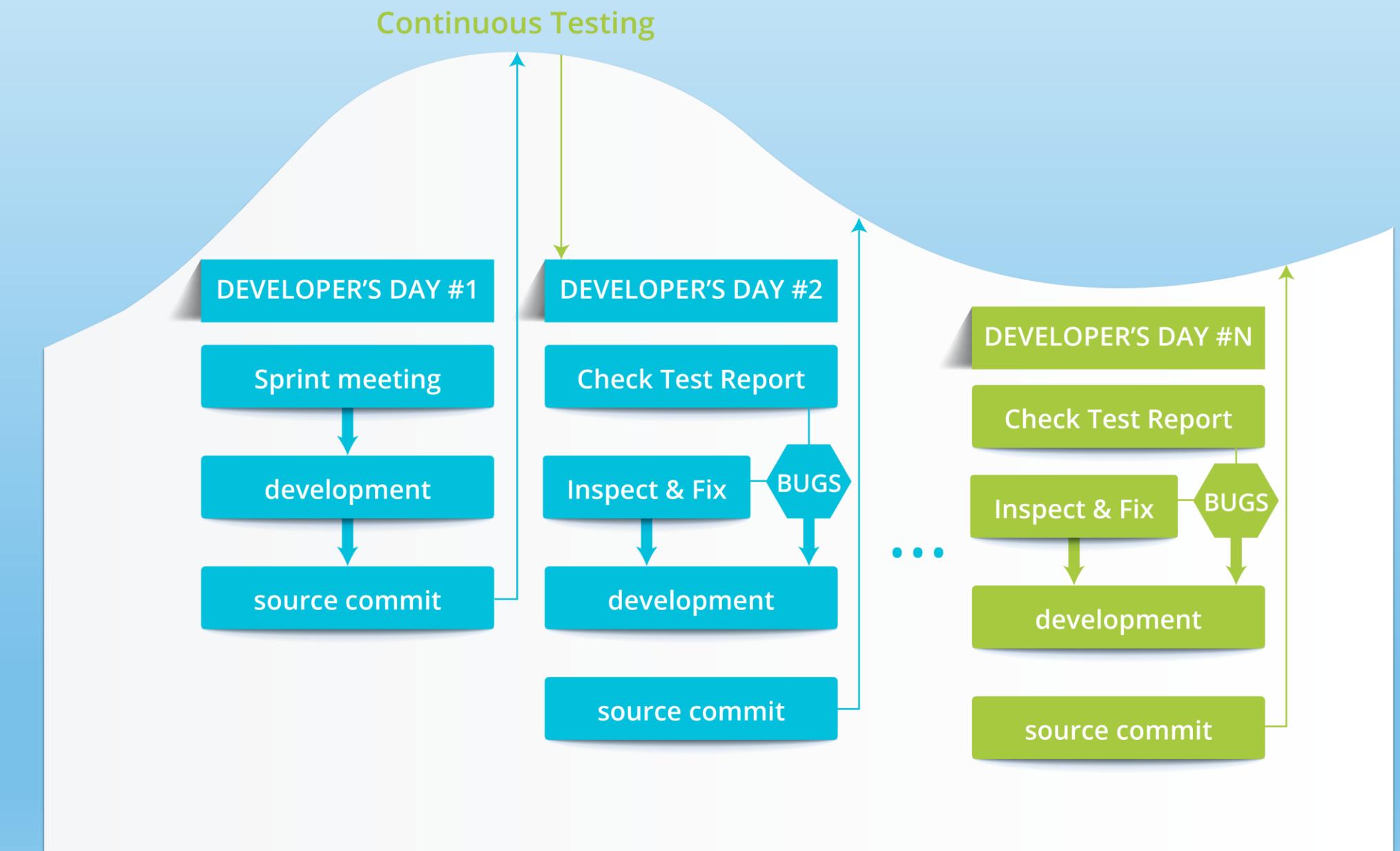


Continuous Testing

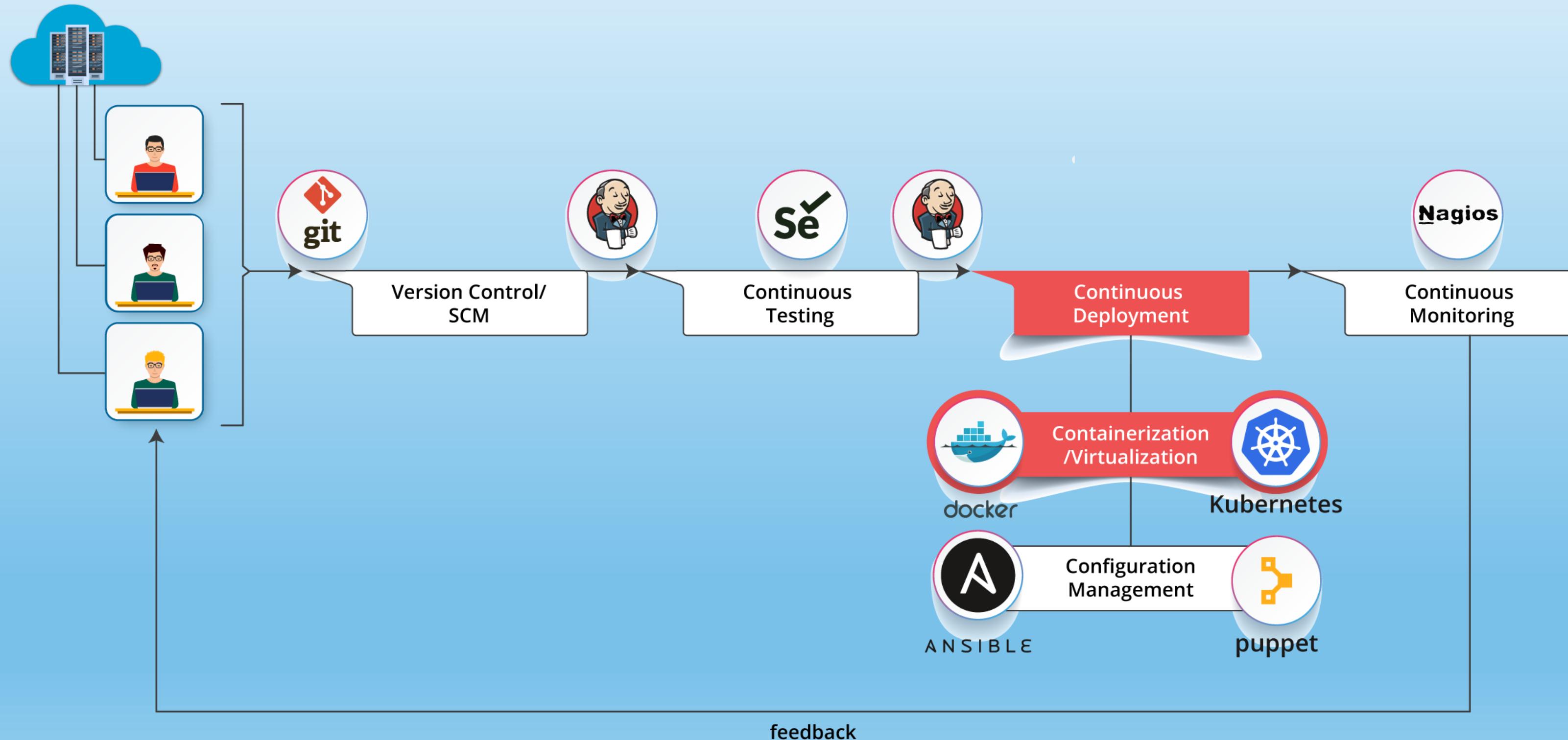


What Is Continuous Testing?

Process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback

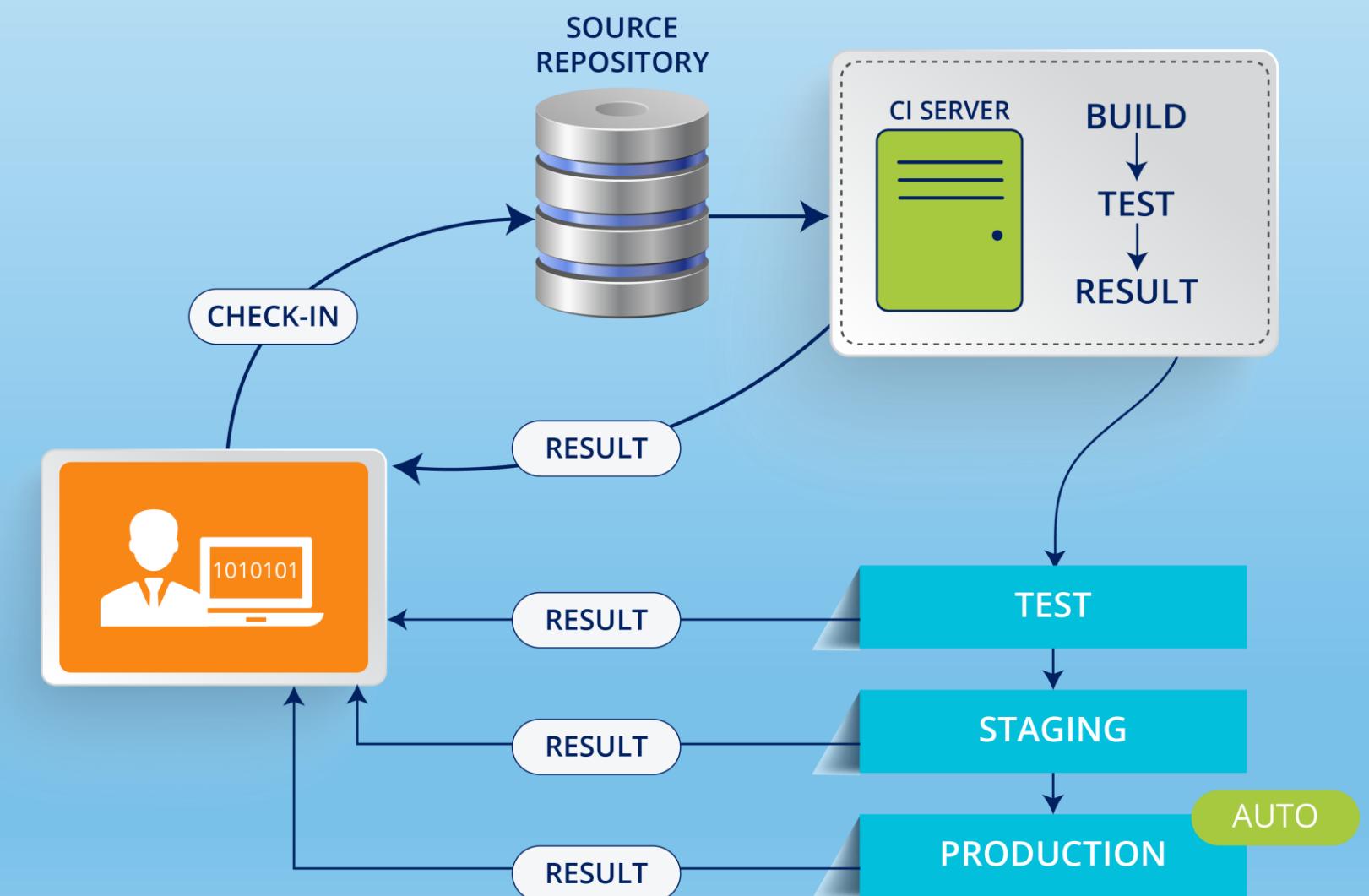


Continuous Deployment



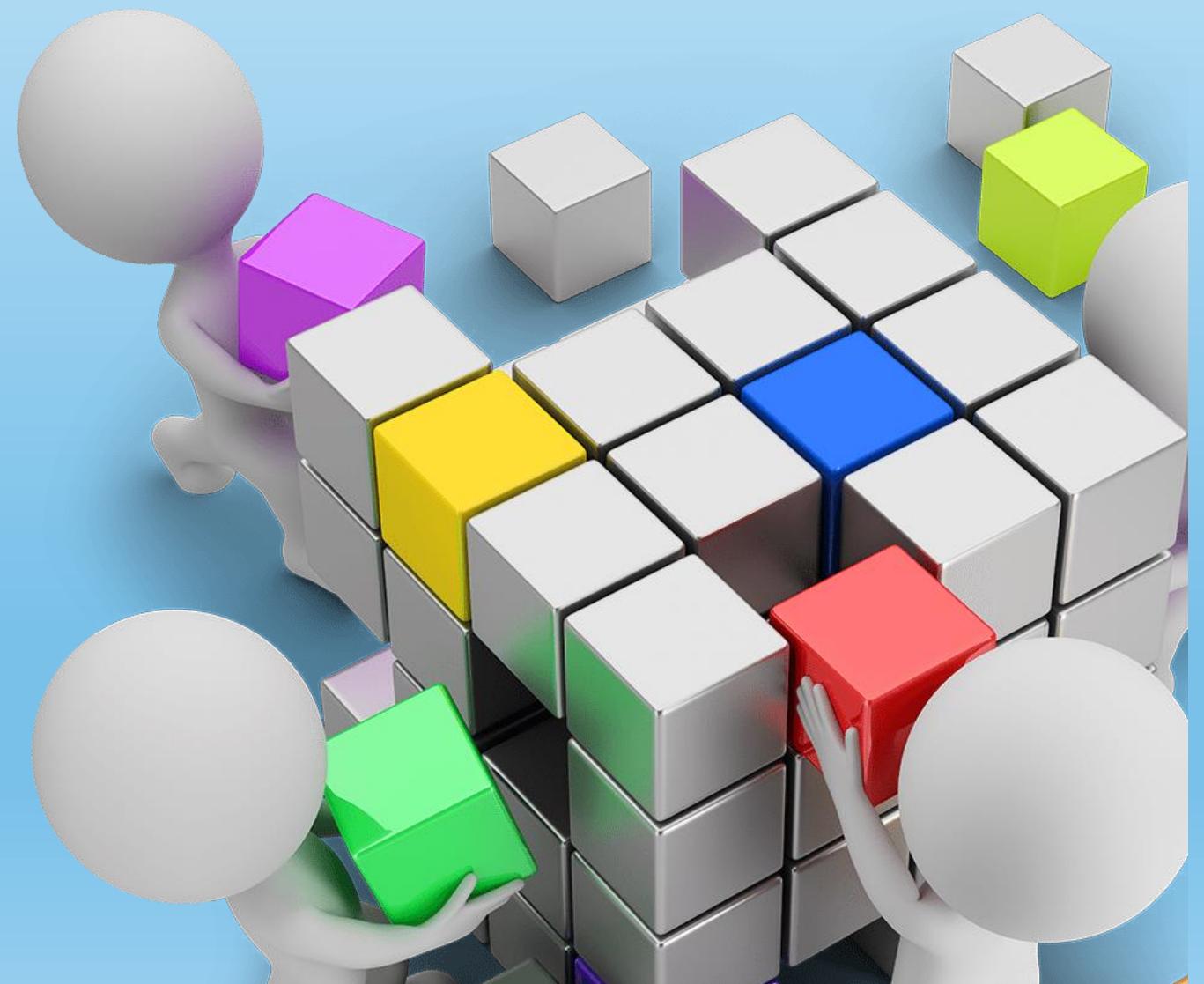
What Is Continuous Deployment?

Continuous Deployment is a DevOps practice where the code changes are automatically built, tested, and prepared for a release to production



What Is Configuration Management?

Process of standardizing resource configurations and enforcing their state across IT infrastructure in an automated yet agile manner.



Configuration Management Using Puppet

Puppet is a configuration management tool used to manage and maintain development and deployment of software systems and servers in any computational environment



puppet

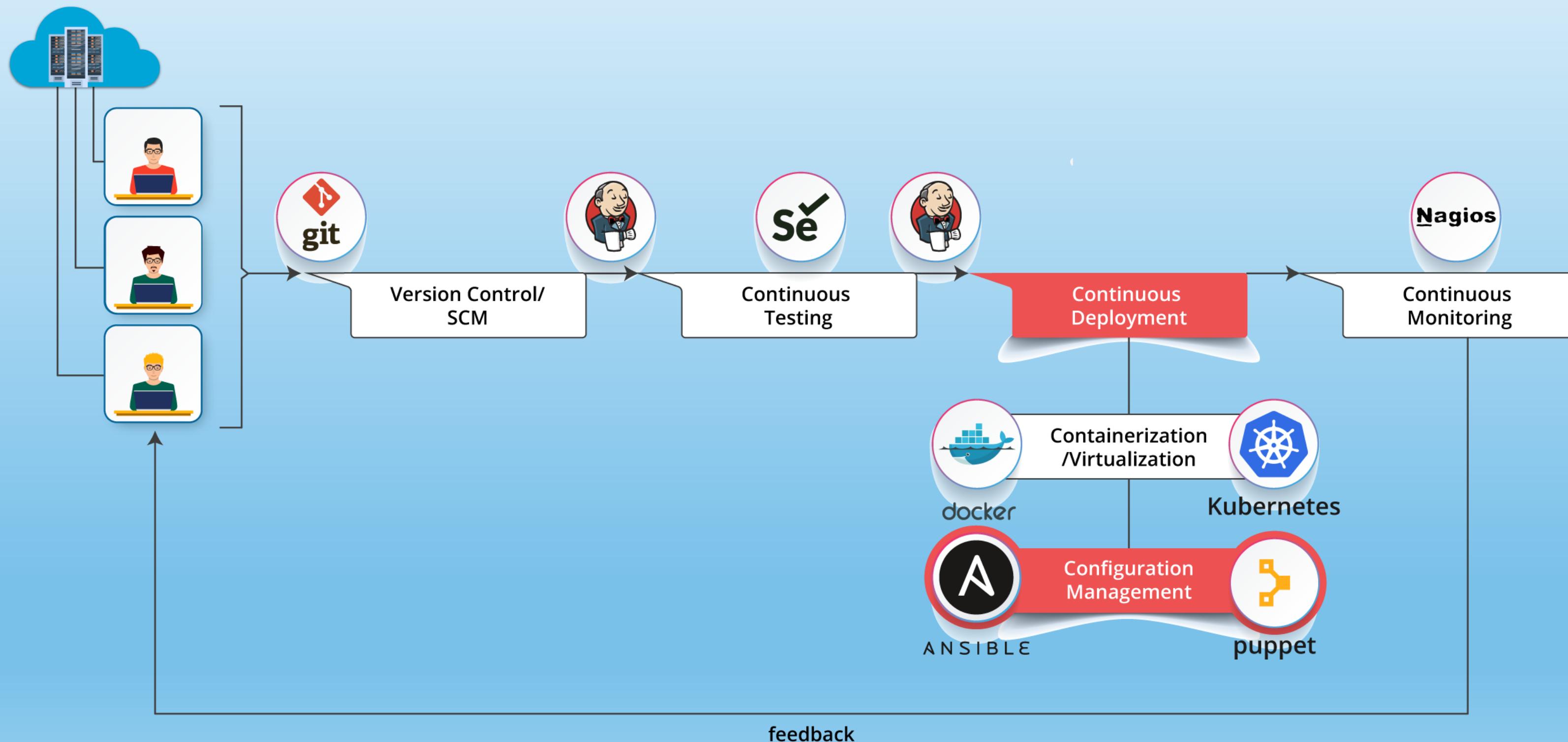
Configuration Management Using Ansible

- Ansible is an open source IT Configuration Management, Deployment & Orchestration tool
- It aims to provide large productivity gains to a wide variety of automation challenges



ANSIBLE

Continuous Deployment: Containerization

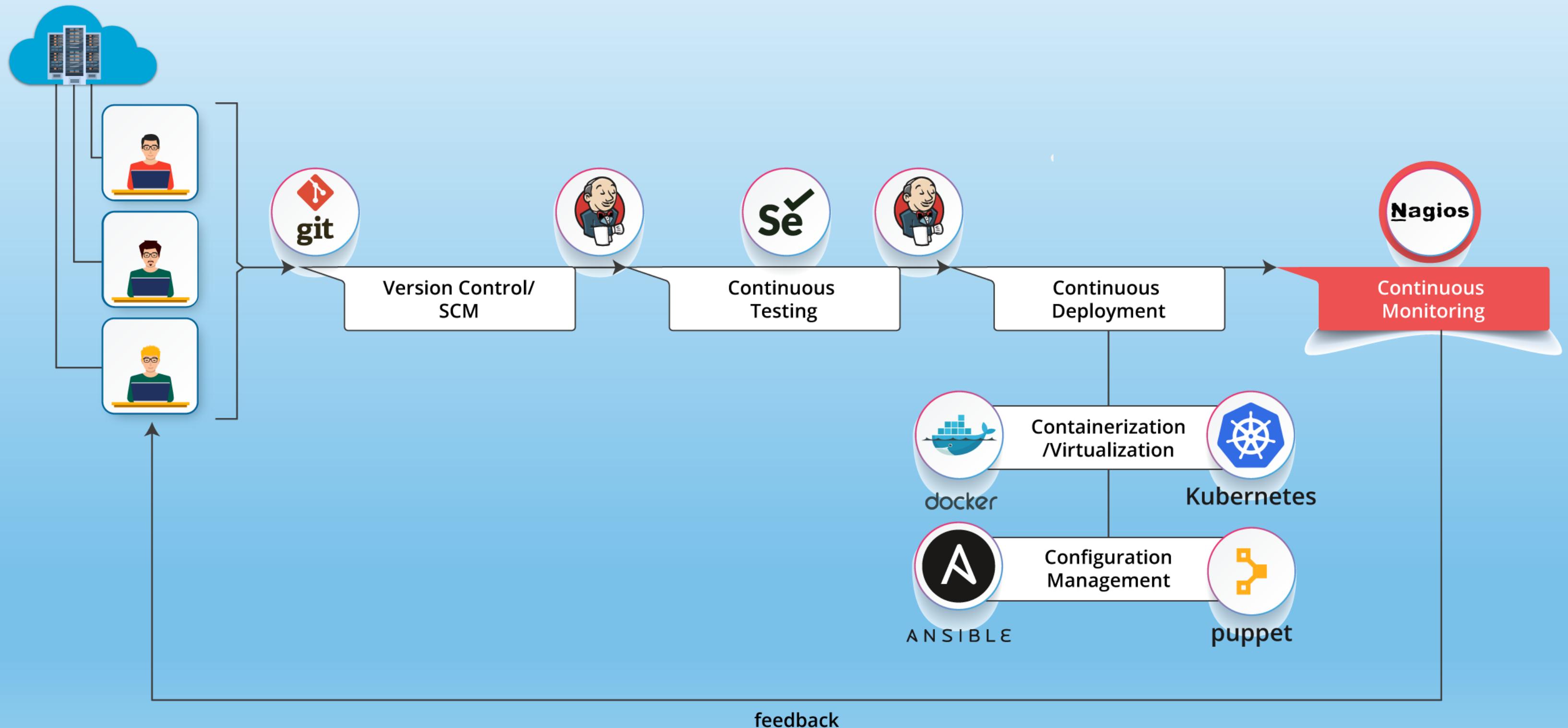


What Is Containerization?

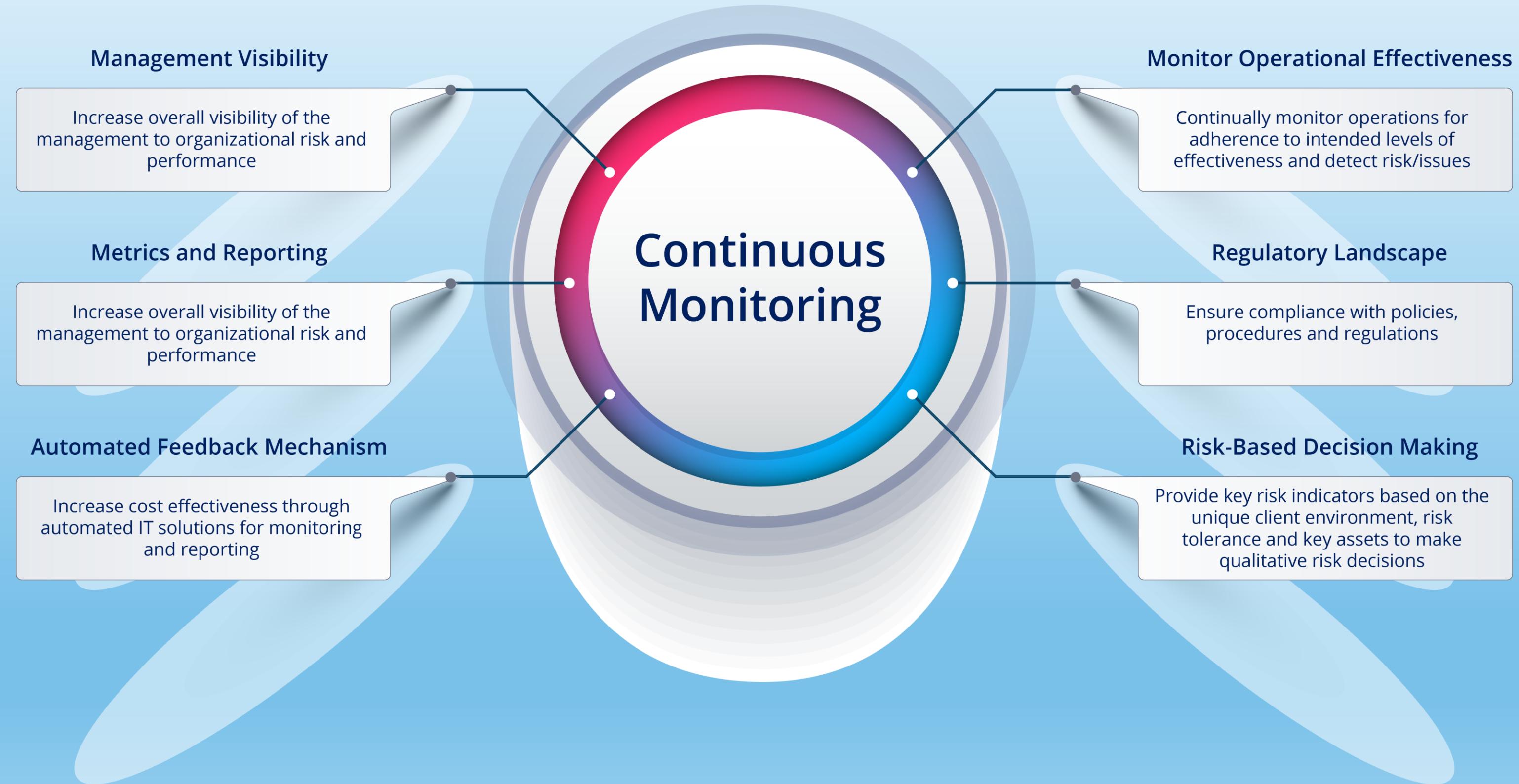
Containerization is a light weight alternative to full machine virtualization which involves encapsulating an application in a container with its working environment.



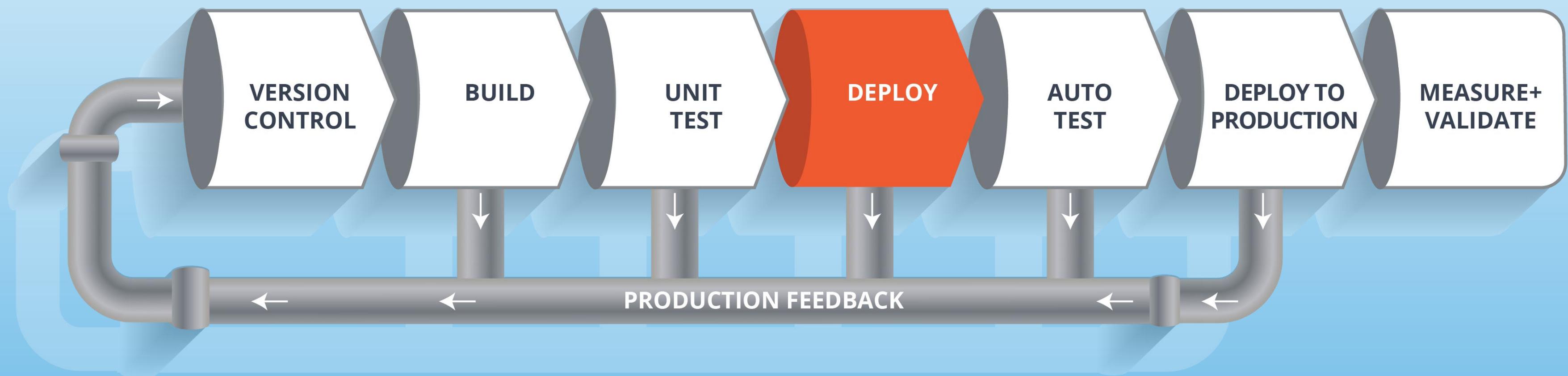
Continuous Monitoring



What Is Continuous Monitoring?



DevOps Delivery Pipeline



Summary

