AWS Architecting and SysOps

Scalability and DevOps in AWS

June-July 2019

Contents

Scalability
DevOps in AWS

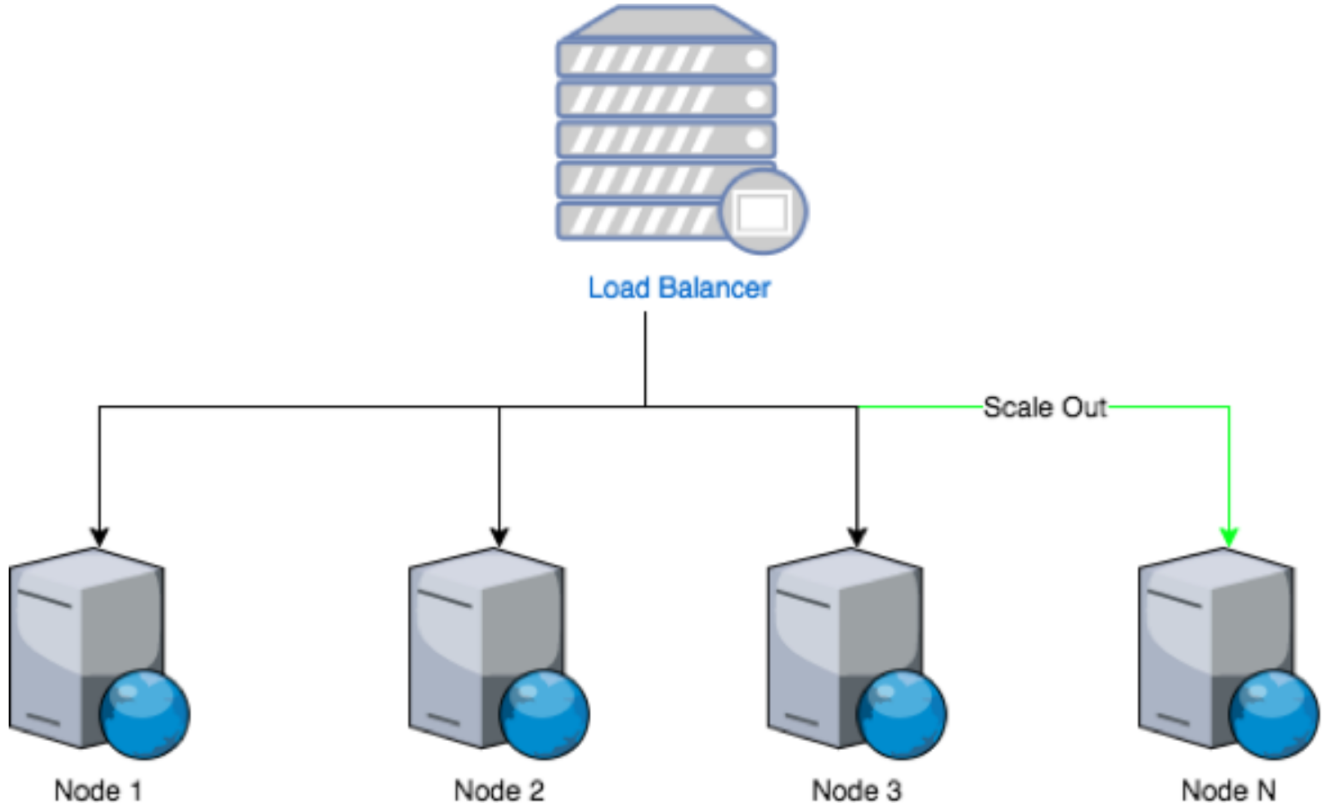
Scalability

Some concepts on grow from zero to big

Scalability

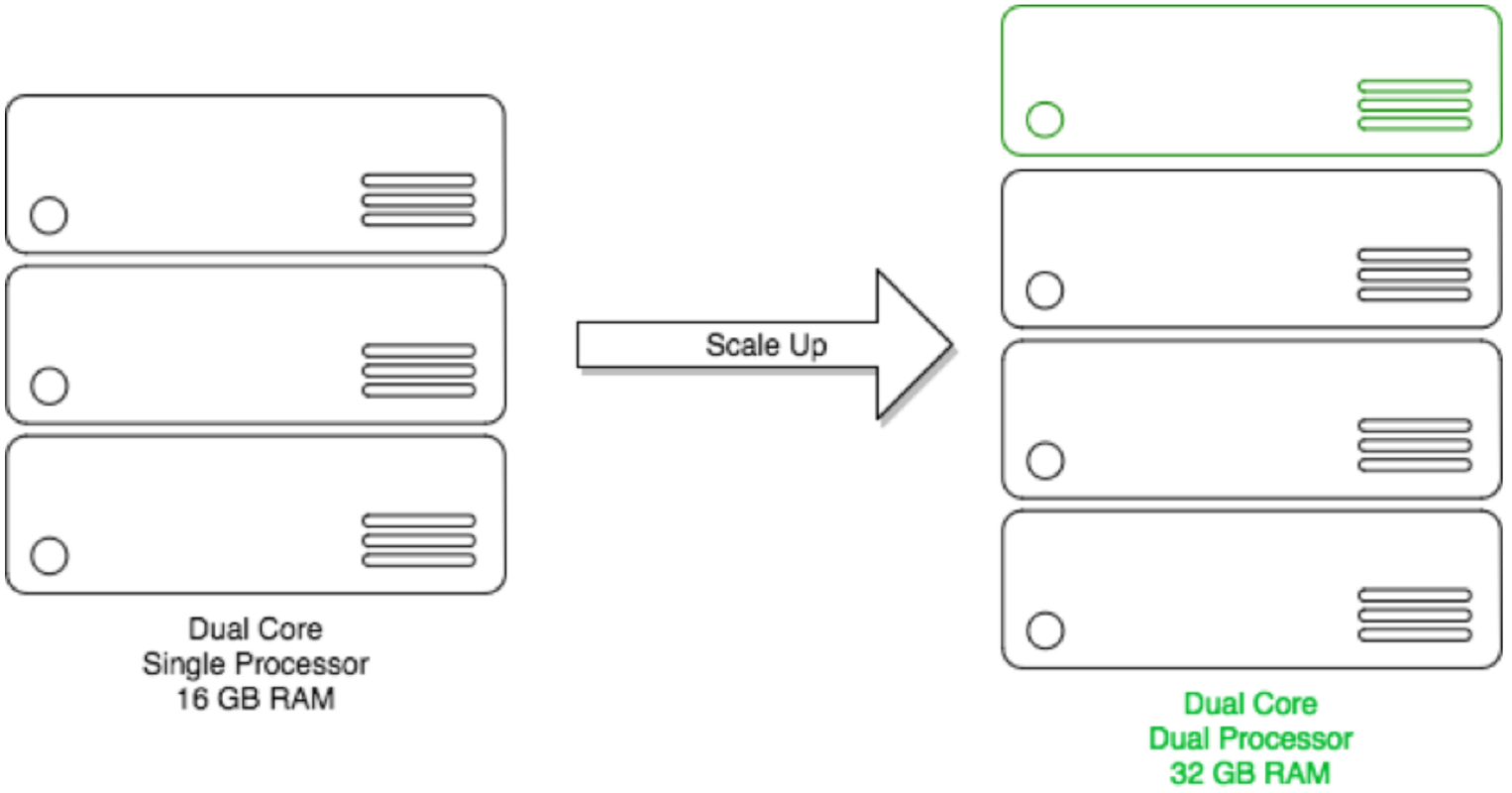
- Property of a system or application to handle big amounts of work, or to be easily expanded, in response to increases demand for network, processing, database access or file system resources

- ✓ Horizontal
  - You scale out by adding more nodes with identical functionality to your existing ones, redistributing the load among all of them
  - Cluster is a common term for describing a scaled out processing system



Scalability

- ✓ Vertical
- You scale up by adding processing, memory, storage, or network interfaces to a node to satisfy more requests per system
- Virtualization is a way to scale up systems



- Scaling on-premises can be hard but deploying your application in the cloud can soften your deployment

High Availability (HA)

- Availability: how well a system provides useful resources over a period of time

- High availability guarantees an absolute degree of functional continuity within a time window expressed as the relationship between uptime and downtime

  A = 100 – (100*downtime / uptime)

Availability %	Downtime in Minutes	Downtime per Year	Vendor Jargon
90	52,560.00	36.5 days	one nine
99	5,256.00	4 days	two nines
99.9	525.60	8.8 hours	three nines
99.99	52.56	53 minutes	four nines
99.999	5.26	5.3 minutes	five nines
99.9999	0.53	32 seconds	six nines

- HA depends on the expected uptime defined for system requirements and is a direct function of an SLA
  - Service Level Agreement (SLA): negotiated terms that outline the obligations of the two parties involved in delivering and using a system
- Availability goes up when factoring planned downtime, such as a monthly 8-hour maintenance window

Scalability & High Availability in AWS

- Horizontal scalability
  - Autoscaling
  - Elastic load balancing
- Vertical scalability
  - Autoscaling
- Remember:
  - Scaling horizontally: thousands of Minions will do the work together for you
  - Scaling vertically: one big Hulk will do all the work for you
- High availability
  - Redundancy through different availability zones

Scalability & High Availability Guidelines

- Design for failure
- Treat hosts as ephemeral and dynamic
  - Hosts are cattle, not pets
- Consider a Serverless Architecture
- Scales automatically
- Sparse your servers across availability zones
- Automate things
  - Everything can be managed as code, so do it
- Firewall your resources, breaches happen
- Monitor everything, alarm carefully
- Chaos engineering??

DevOps in AWS

What AWS offers to enforce DevOps culture

What is DevOps?

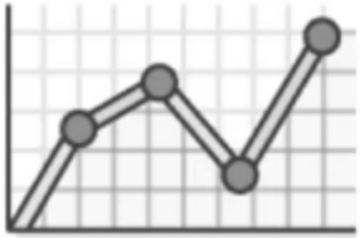
- Development + Operations = DevOps
- A combination of cultural philosophies, practices, and tools
- Increase an organization's ability to deliver applications and services at high velocity
- Enables product improvements at faster pace than organizations using traditional software development and infrastructure management processes
- DevOps culture requires collaboration between the teams involved in a product lifecycle
  - Enables rapid development and delivery products
- DevOps goal: a continuous delivery model that is repeatable, reliable, stable, resilient, and secure while improving operational efficiency
  - We need to create a framework that is going to do all this work for us

DevOps benefits

Speed

Rapid Delivery

Reliability



Scale

Improved Collaboration

Security

DevOps culture

- DevOps is about removing the barrier between development and operations
  - Getting your different teams to communicate to each other
  - You do not want to be locked in “this is the way we have always done things” as you will not be able to keep up with the demands of a changeable market
- With DevOps, teams work together to optimize both the productivity of developers and the reliability of operations
  - And if necessary adapt new tools and new ways of working
- DevOps motto: People over Process over Tools
  - It is more a mentality, that a specific set of tools or resources
  - It is a way of doing things

Why AWS for DevOps

- AWS services simplify provisioning and managing infrastructure, deploying application code, automating software release processes and monitoring your application and infrastructure performance

- Get started fast
  - In a few clicks, a service is running
- Fully managed services
  - Reduce setting up your system
- Built for scale
  - Flexible resources
- Programmable
  - AWS CLI, APIs and SDKs
- Large partner ecosystem
  - Using the most recent platforms
- Pay-as-you-go
  - No minimum fees, no upfront commitments
- Secure
  - IAM allows access granular control
- Automation
  - Manual tasks or processes

DevOps benefits in AWS

Continuous Integration

Continuous Delivery

Microservices

Infrastructure as Code

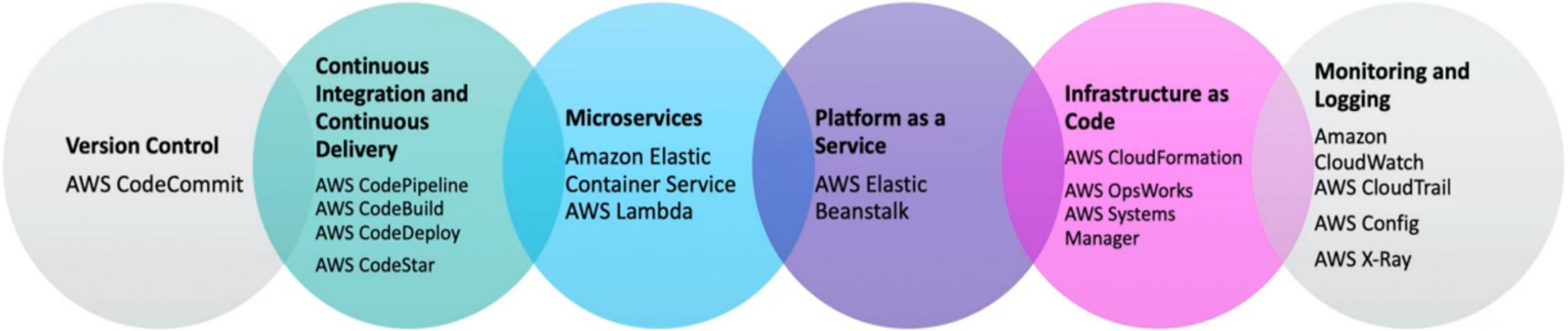
Monitoring and Logging

Communication and Collaboration

DevOps benefits in AWS

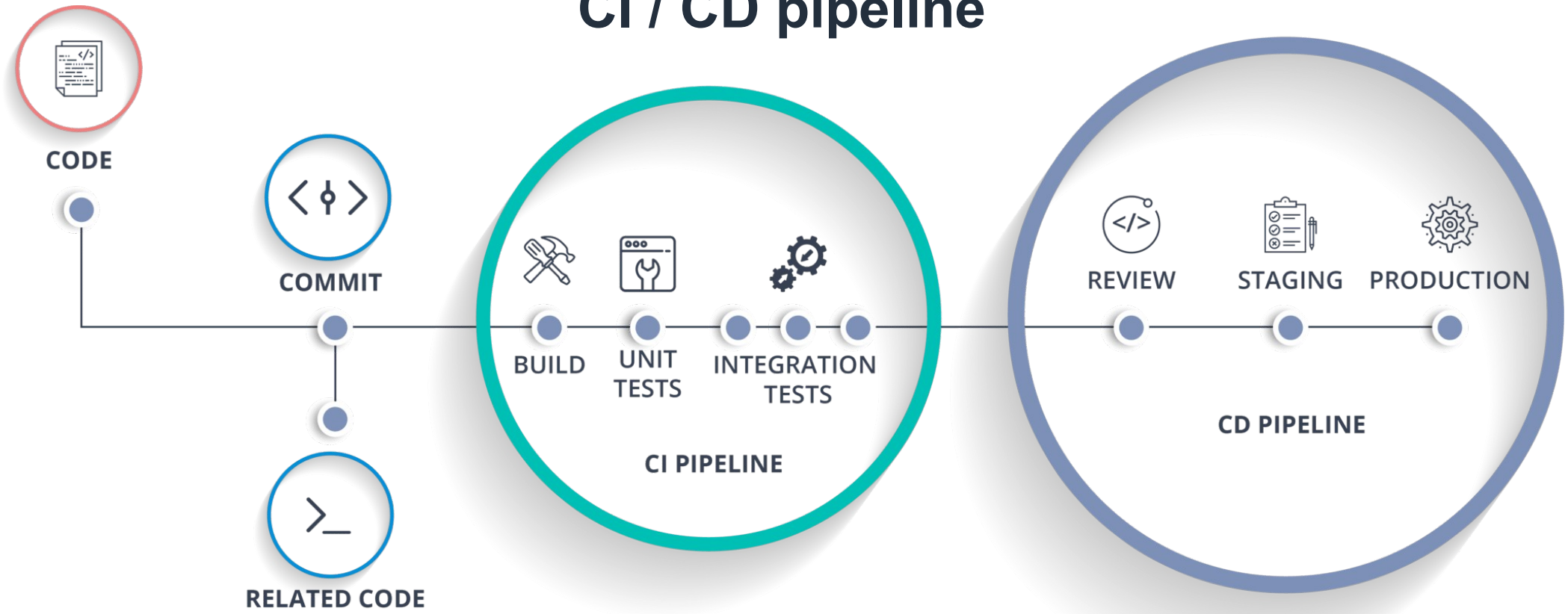
- Continuous integration (CI)
  - You are writing and building your code
- Continuous delivery (CD)
  - When it is up and going, you need to deploy it somewhere effectively
- Microservices
  - A way of breaking down that monolithic application into small pieces
- Infrastructure as code
  - Provision, configure, and manage your AWS infrastructure using code and templates
- Monitoring and logging
  - Record logs and monitor application an infrastructure performance in near real-time
- Communication and collaboration
  - Single point where you can view the health of your application

DevOps tools in AWS



- This is how AWS tools can play in your DevOps pipeline
- Any of these services can be combined with your own tools

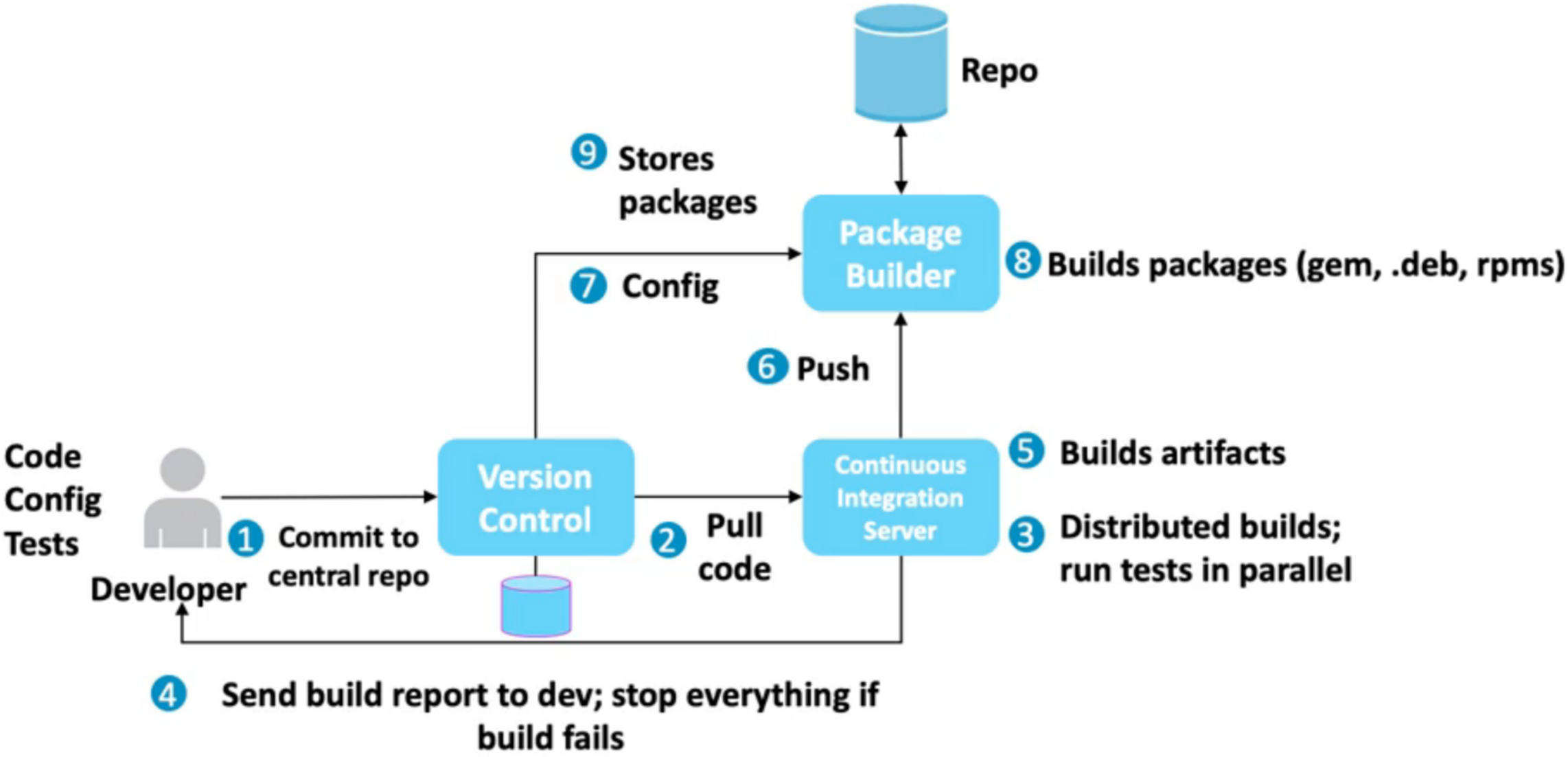
CI / CD pipeline



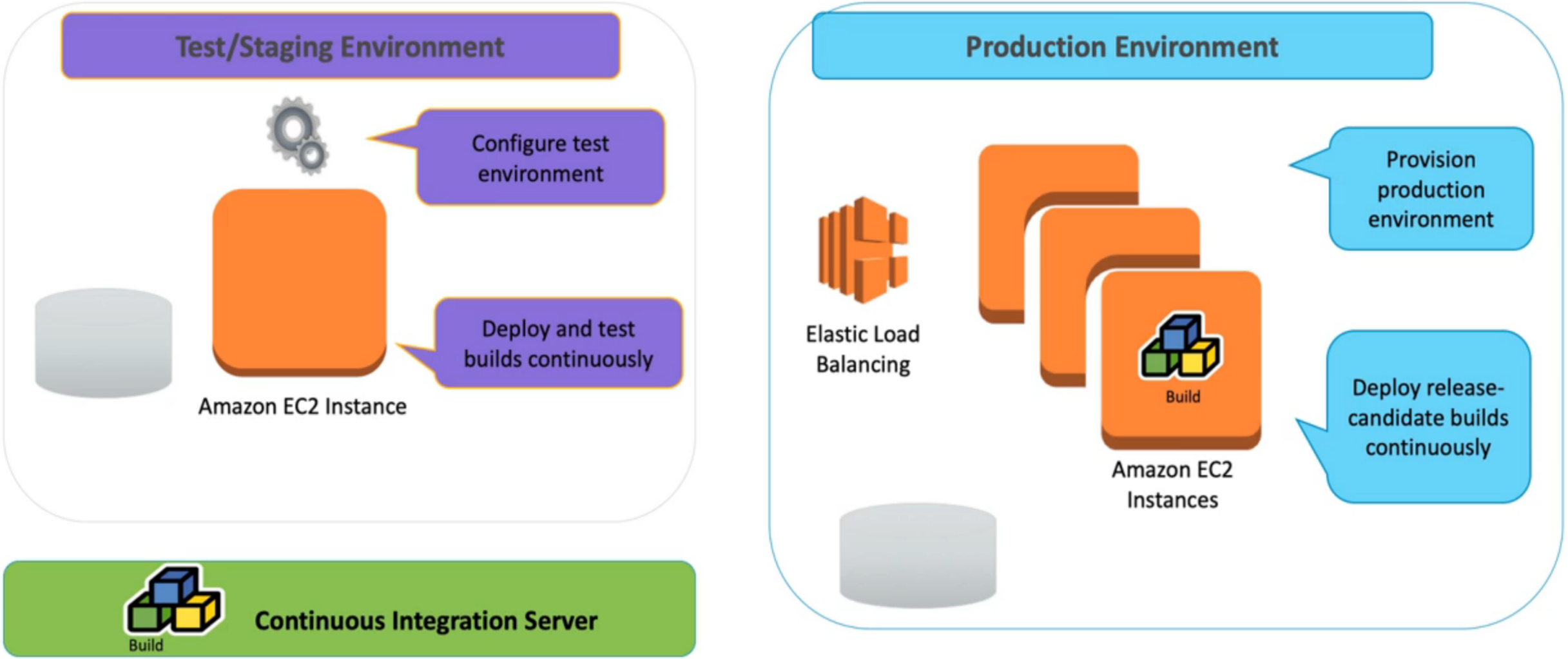
- Integration
  - ✓ Use code repositories
  - ✓ Test and commit code frequently
  - ➢ Do not break the build!
  - Automate CodeCommit to trigger the build phase

- Delivery
  - ✓ Automate build and test
  - ➢ Never do it manually!
  - ✓ Push code to non-production environment
  - ✓ Deploy production environment

Continuous Integration



Continuous Delivery



Continuous Monitoring and Improvement

- Process of:
  - Capturing, categorizing, and analyzing data and logs generated by applications and infrastructure
  - To understand the performance, impact and issues
  - Constant improvement based on the metric analysis

- DevOps metrics:
  - Changes volume
  - Deployment frequency
  - Lead time from development to deployment
  - % of failed deployments
  - Availability
  - Customer ticket volume
  - % change in user volume

- Application performance metrics:
  - Changes volume
  - Monitor application performance
  - Scale resources up or down as needed
  - Communicate performance bottlenecks or under-utilized resources to development team

DevOps references

- DevOps and AWS
  - https://aws.amazon.com/devops/