DEVOPS FOUNDATIONS

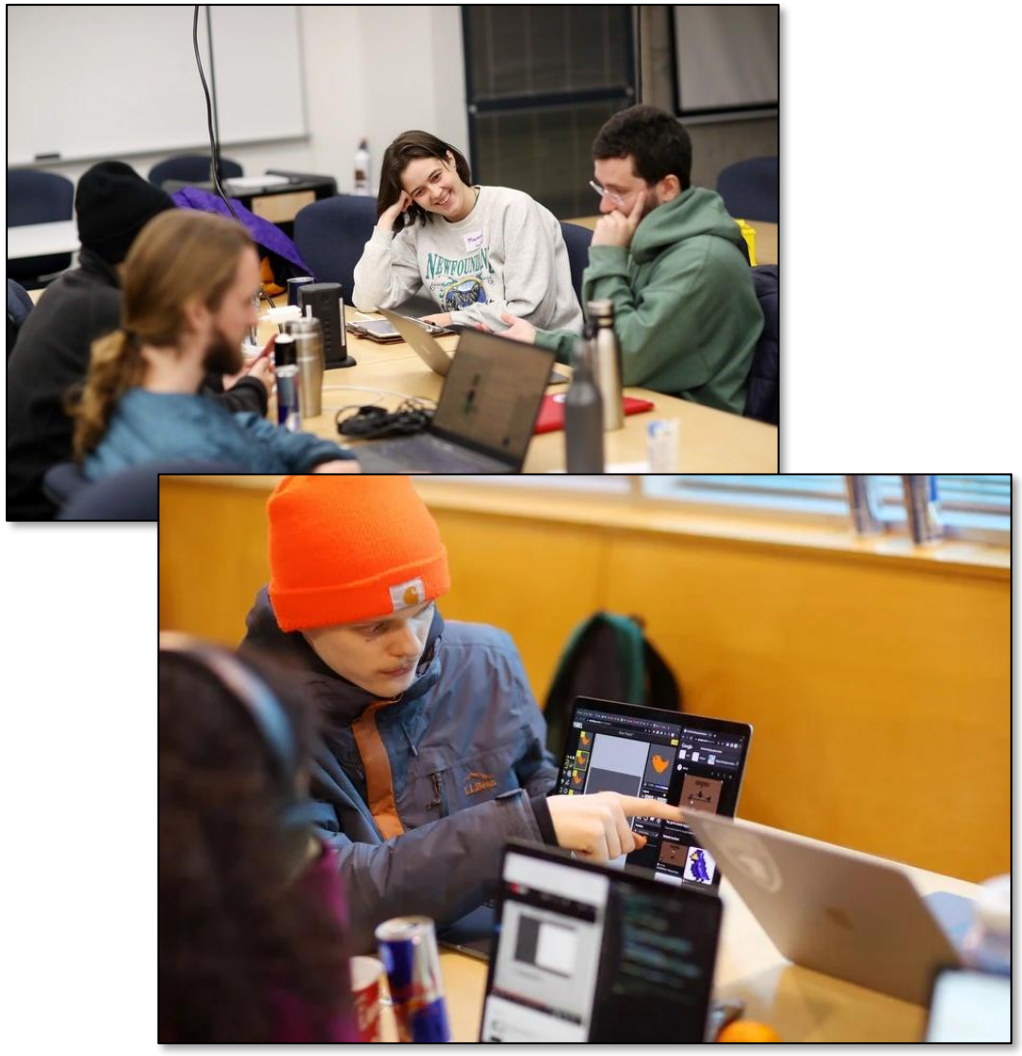
CERTIFICATE

- Zainuddin Saiyed -

SHIFTKEY ACADEMY

About ShiftKey Labs

From hackathons to the Startup Residency program, ShiftKey Labs makes sure students have the skills & support they need to pursue their most innovative ideas!



Highlights:

- NASA Space Apps Challenge
- Global Game Jam
- Generative AI Hackathon
- Workshops & Talks
- REACT | Blockchain | Prototyping | Startups | Freelancing | Cybersecurity

Contact: info@shiftkeylabs.ca | @ShiftKeyLabs | shiftkeylabs.ca

SHIFTKEY ACADEMY

Land Acknowledgement

ShiftKey operates in Mi'kma'ki, the unceded and ancestral territory of the Mi'kmaq People, protected under the Peace and Friendship Treaties of 1725. We acknowledge them as the rightful caretakers of this land and are thankful for the opportunity to be here. We are all Treaty People.

We also acknowledge the 50 African Nova Scotian communities as a distinct population and a crucial part of Mi'kma'ki's history, heritage, and legacy over the past 400 years. We are grateful for their contributions to our shared society.

SHIFTKEY ACADEMY

Course Timeline

- **Session 1**: 5th November
  - Introduction to DevOps and CI/CD Pipelines
- **Session 2**: 12th November
  - Containerisation and Orchestration using Docker and Kubernetes
- **Session 3**: 19th November
  - Cloud computing fundamentals, Infrastructure as Code, and container Deployment on AWS
- **Session 4**: 26th November
  - Final Exam & Project QnA session

Certificate Eligibility

1. Attend **at-least 2** out of the **first 3 sessions** (75% attendance).
2. Assessments and Deliverables:
    1. **In-person examination** on Session 4 (November 26).
    2. **Takeaway course project** released on Session 2 (November 12) due one week after Session 4 (November 26).

SHIFTKEY ACADEMY

SOFTWARE DEVELOPMENT LIFECYCLE

(SDLC)

Traditional SDLC

The traditional Software Development Life Cycle (SDLC) follows a structured and sequential approach, consisting of distinct phases such:

1. Planning
2. Analysis
3. Design
4. Implementation
5. Testing
6. Maintenance

Each phase is typically handled by separate teams, leading to a siloed environment where collaboration is limited.

Challenges with Traditional SDLC

Despite its structured nature traditional SDLC faces a lot of challenges such as:

- Lengthy development cycles and release cycles
- Slow time-to-market
- Lack of collaboration between development and operations teams (*impact on responsiveness*)
- Limited or Delayed feedback loops (*increased risk of errors in production*)

WHAT IS DEVOPS?

SHIFTKEY ACADEMY

What is DevOps?

DevOps is a set of practices that combines software development (**Dev**) and IT operations (**Ops**) to shorten the systems development lifecycle while delivering features, fixes, and updates frequently in close alignment with business objectives.

Emphasis: collaboration, automation, and continuous improvement.

- OR -

DevOps is a combination of cultural philosophies, practices and tools that:

- Enhances organisation's ability to deliver applications faster
- Improving products at a faster pace
- Enhancing customer service, while gaining competitive edge.

Read more: https://aws.amazon.com/devops/what-is-devops/

SHIFTKEY ACADEMY

What is the Aim/Goal of DevOps?

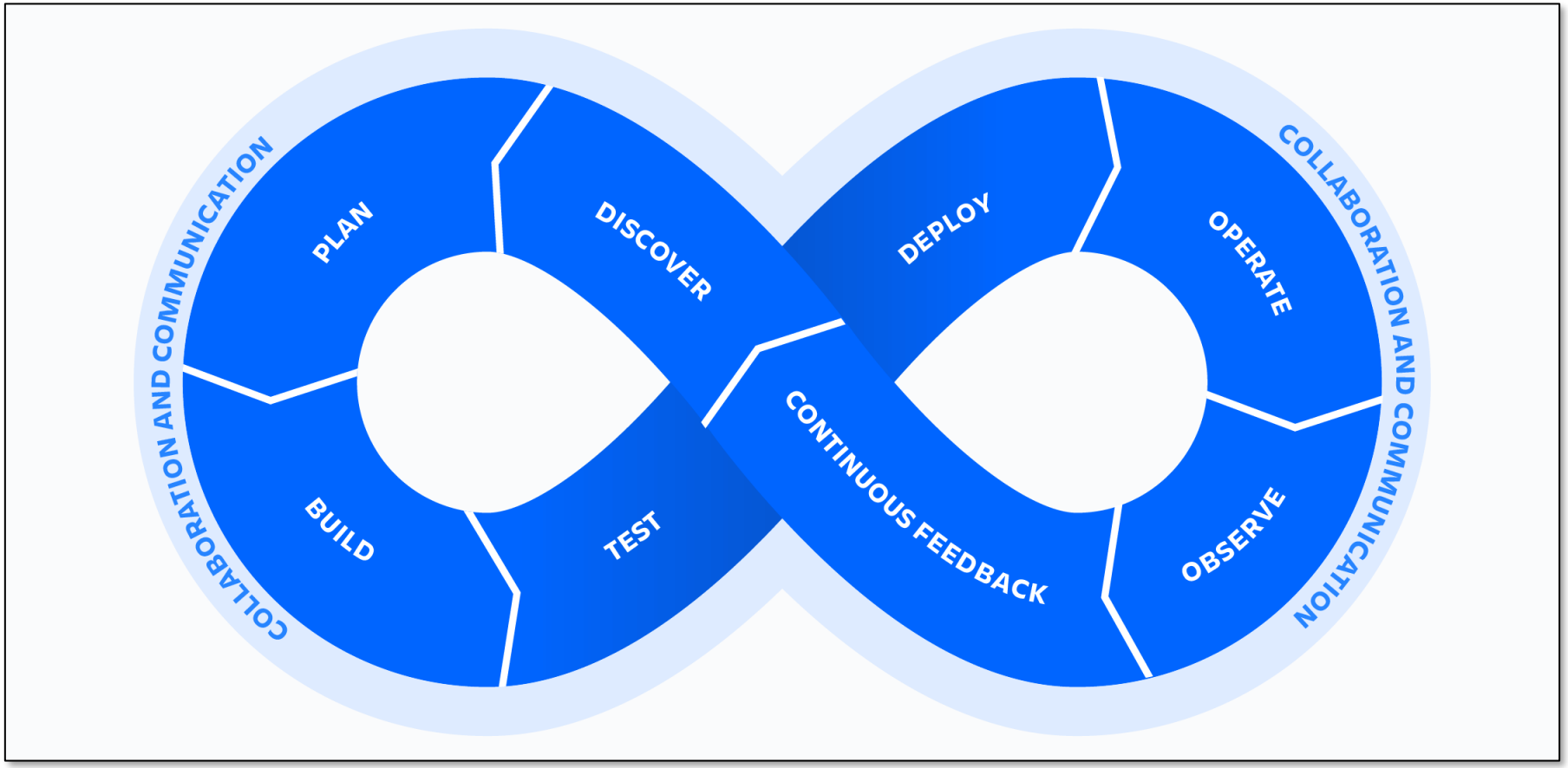
1. Increased Automation
2. Flexibility and tolerance to change in requirements
3. Process workflows are standardized
4. Reduced Cost and overall efforts
5. Time to market is reduced
6. Re-enforces Continous Improvement

Benefits of DevOps?

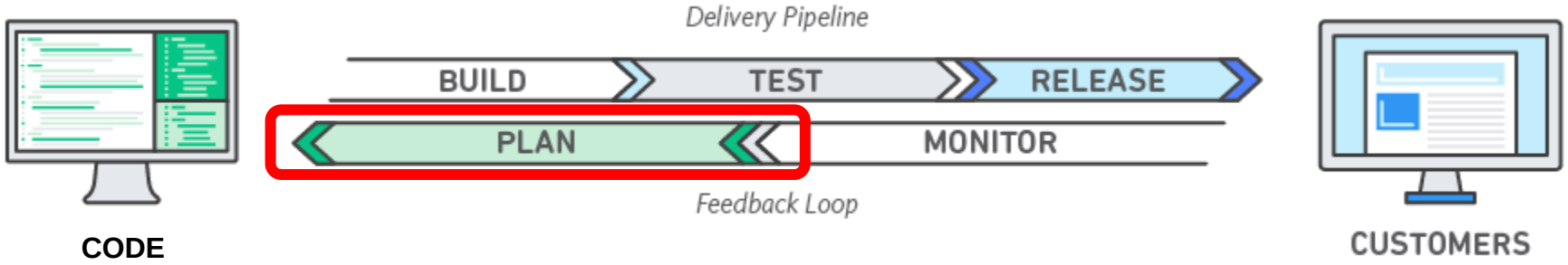
Key Benefits of DevOps:

- Faster delivery of software
- Improved collaboration between teams
- Enhanced quality and reliability
- Increased efficiency and productivity

DevOps Lifecycle



DevOps Lifecycle Phases

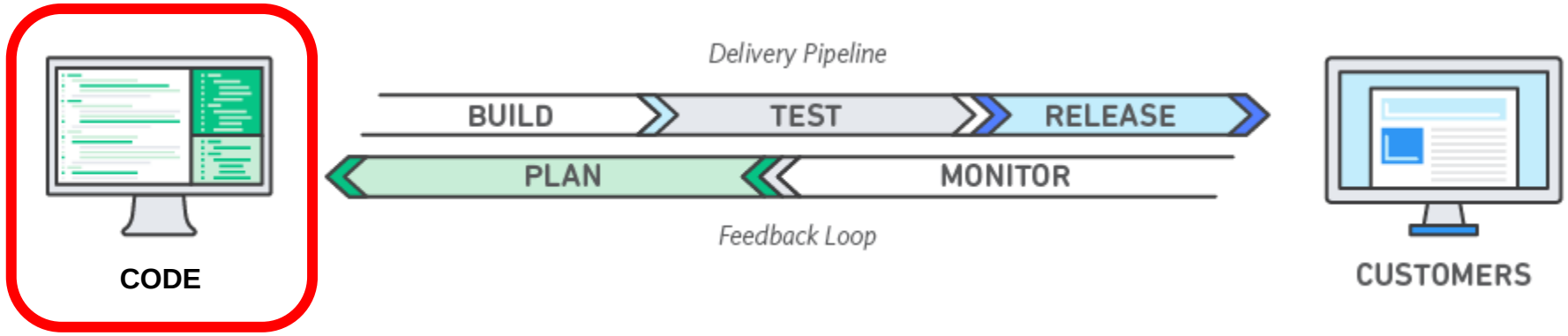


1. Plan

- Define project requirements and objectives
- Create a detailed roadmap and sprint plans
- Prioritize features and allocate resources
- **Tools**: Jira, Trello, … (*project management*)

Read more: https://aws.amazon.com/devops/what-is-devops/

DevOps Lifecycle Phases

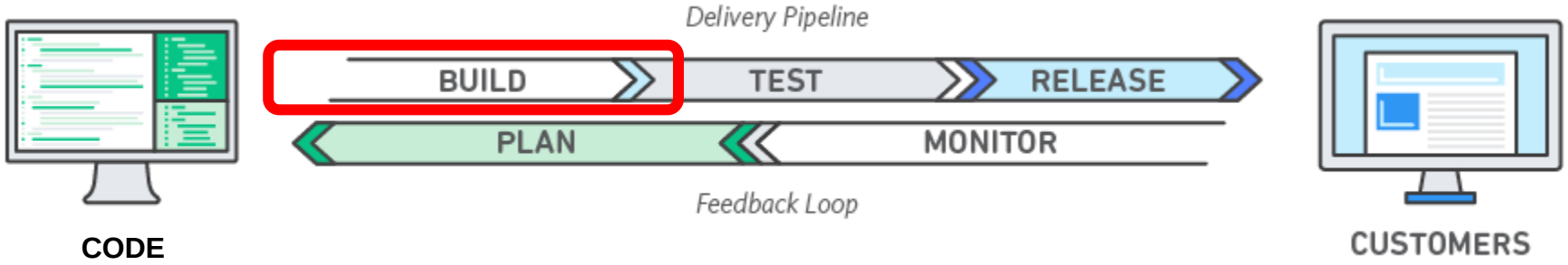


2. Code

- Develop software following industry coding standards
- Regular use of version control (like Git) for collaboration
- Conduct regular code reviews to ensure quality
- **Tools**: IDE, Python, Java-Springboot, Node.js, … (*Depending on Application*)

Read more: https://aws.amazon.com/devops/what-is-devops/

DevOps Lifecycle Phases

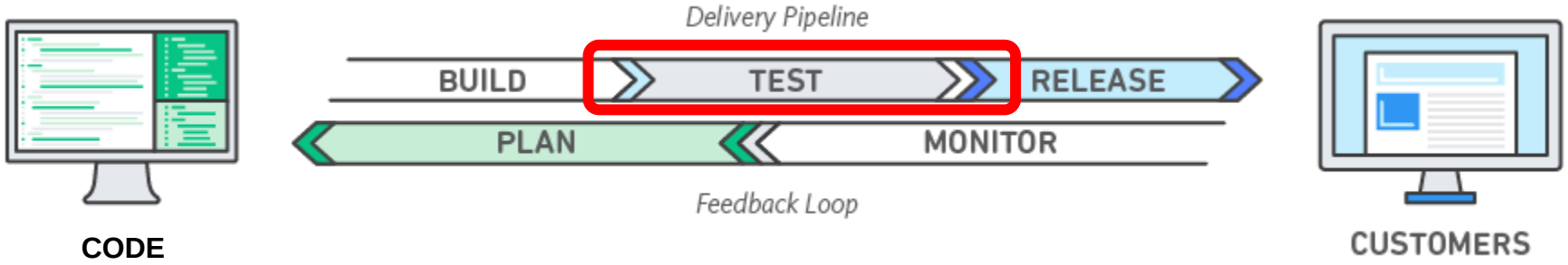


3. Build

- Compile source code into executable format
- Package application with modules/dependencies
- Create build artifacts for deployment
- **Tools**: Automated build tools like Jenkins, GitLab CI

Read more: https://aws.amazon.com/devops/what-is-devops/

DevOps Lifecycle Phases

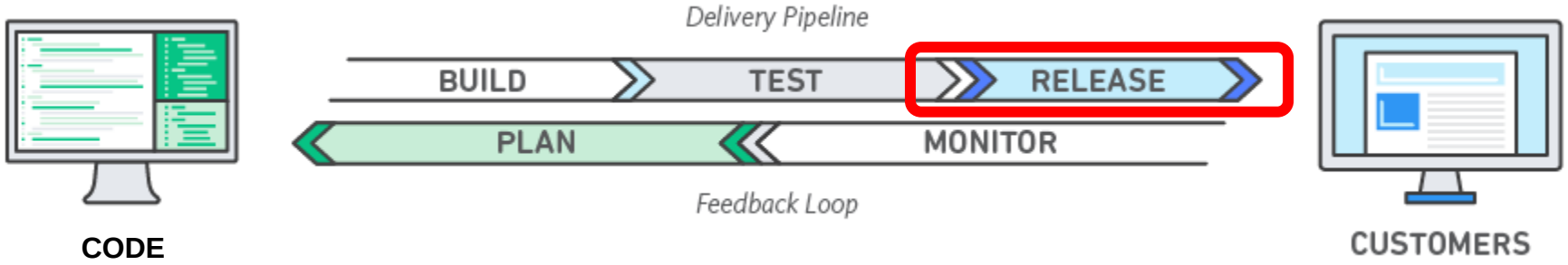


4. Test

- Implement automated testing (unit, integration, system)
- Perform quality assurance and security testing
- Conduct performance and load testing
- **Tools**: Selenium, JUnit, or pytest

Read more: https://aws.amazon.com/devops/what-is-devops/

DevOps Lifecycle Phases

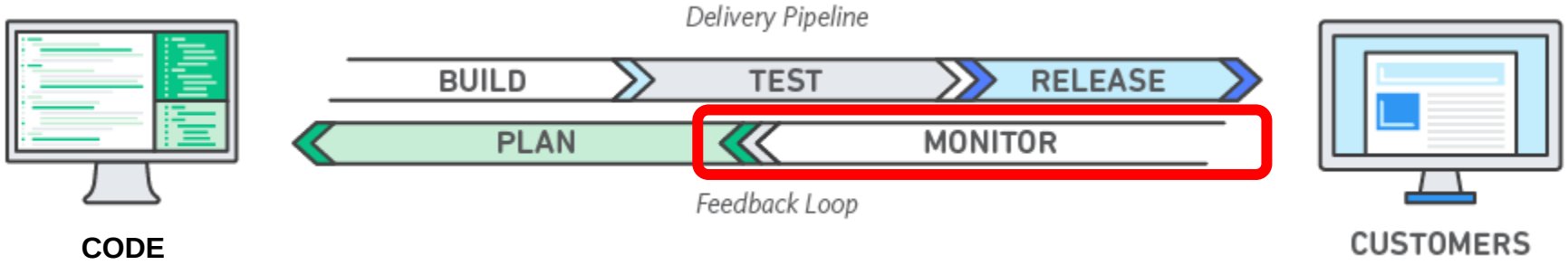


5. Release & Deploy

- Automate deployment processes for consistency
- Implement continuous delivery/deployment pipelines
- Use containerization (like, Docker) for portability
- Use infrastructure-as-code for ensuring program environment consistency

Read more: https://aws.amazon.com/devops/what-is-devops/

DevOps Lifecycle Phases

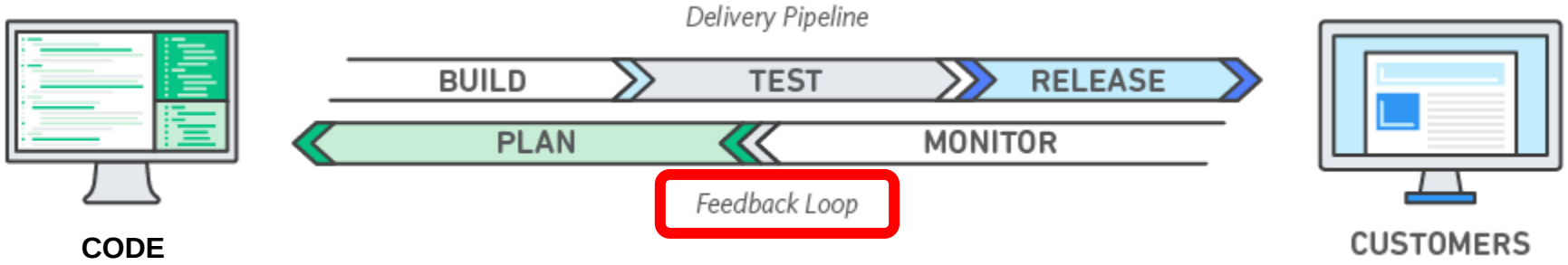


6. Monitor

- Track application and infrastructure performance
- Monitor user experience and business metrics
- Implement real-time alerting systems
- **Tools**: Prometheus, Grafana, or ELK stack

Read more: https://aws.amazon.com/devops/what-is-devops/

DevOps Lifecycle Phases



7. Feedback

- Collect and analyze data from all previous phases
- Get user feedback and feature requests
- Conduct analysis on incidents
- Use above insights to drive continuous improvement

Read more: https://aws.amazon.com/devops/what-is-devops/

All DevOps Practices

1. Continuous Integration (CI)
2. Continuous Delivery
3. Microservices
4. Infrastructure as Code
5. Monitoring and Logging
6. Communication and Collaboration
7. DevOps Tools★

★

SHIFTKEY ACADEMY

1. Continuous Integration (CI)

- **Definition**:

  A software development practice where developers frequently merge their code changes into a central repository, followed by automated builds and tests.

- **Benefits**:
  - Regularly merges code changes (often several times a day)
  - Automated builds and tests run after each merge
  - Aims for early bugs detection and resolution
  - Improves software quality
  - Reduces time to validate and release updates

2. Continuous Delivery (CD)

- **Definition**:

  A software development practice where code changes are automatically built, tested, and prepared for release to production.

- **Benefits**:
  - Automates the entire software release process (*to production*)
  - Ensures code is always in a deployable state
  - Enables faster and more frequent releases
  - Reduces manual errors in the deployment process

DEMO?

ANY QUESTIONS?

THANK YOU!

SHIFTKEY ACADEMY