Welcome to DevOps and Continuous Integration.

What you will learn

At the core of the lesson

You will learn how to:

- Define DevOps
- Identify the goals of DevOps
- Identify the challenges that DevOps solves
- Describe the culture of DevOps

3

aws re/start

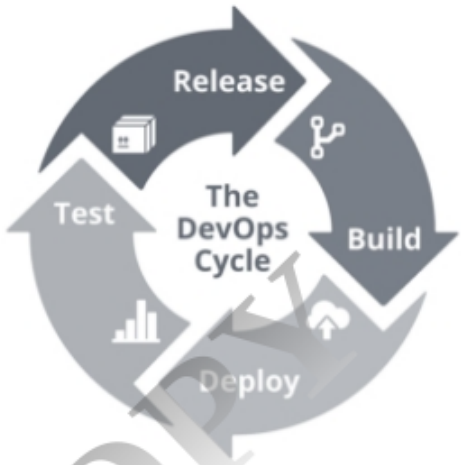
In this module, you will learn how to:

- Define DevOps
- Identify the goals of DevOps
- Identify the challenges that DevOps solves
- Describe the culture of DevOps

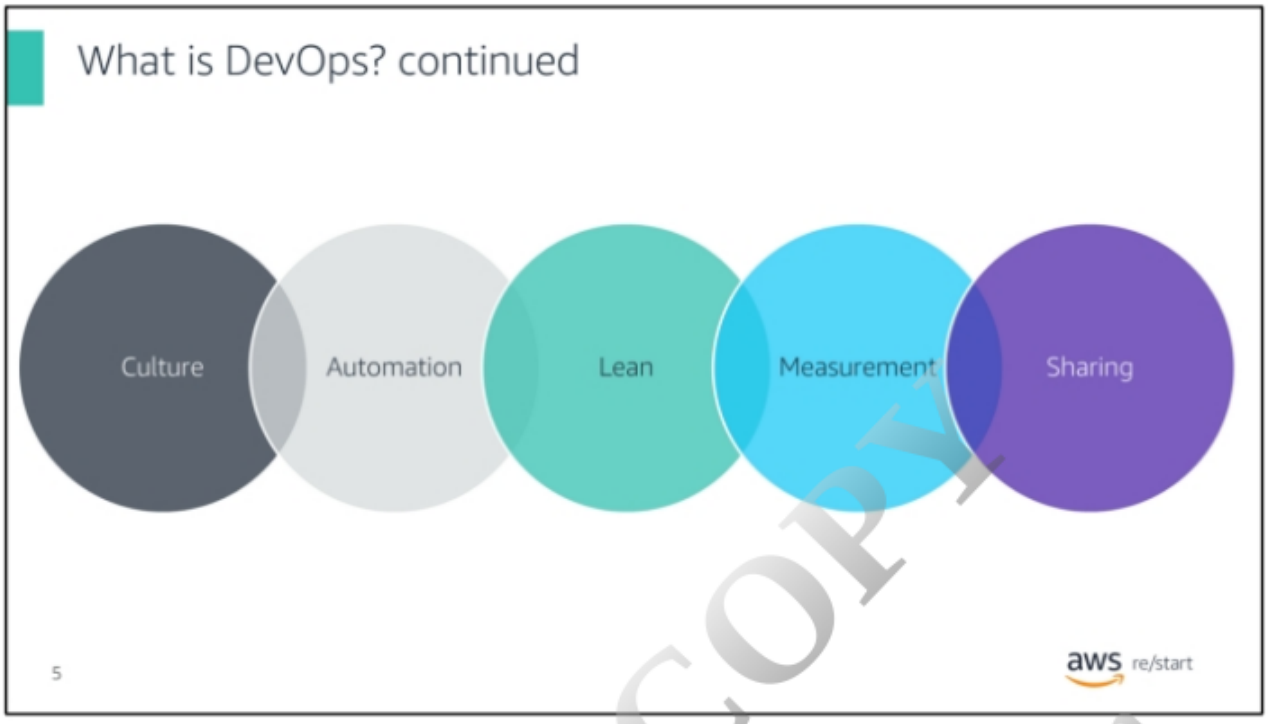
What is DevOps?

DevOps is a software engineering culture and practice that aims to unify software development (Dev) and software operation (Ops).

The main characteristic of the DevOps movement is to advocate for automation and monitoring at all steps of software construction. These steps range from integration, testing, and releasing to deployment and infrastructure management.



aws re/start

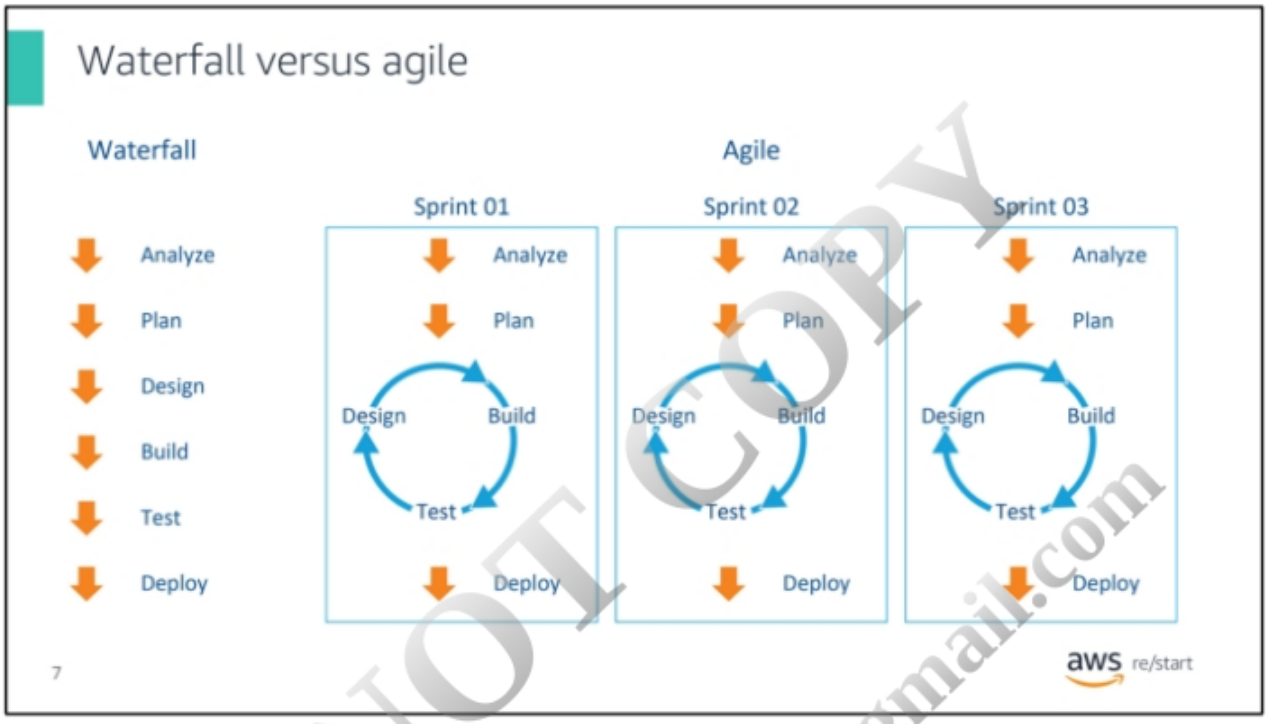


Culture of DevOps

1. Consider the performance of the **entire system – FLOW**
2. Create and amplify the right-to-left **feedback** loops
3. Create a culture that fosters:
   a. Continual experimentation, taking risks, and **learning**
   b. **Repetition and practice** – the prerequisite to proficiency

6

aws re/start



Goals of DevOps

- DevOps is meant to bridge the gaps between traditional IT, software development, and quality assurance (QA).
  - The most difficult part for beginners is the QA part. The appearance of your code is important.
- DevOps is meant to be faster and more flexible.
  - The challenge is better integration of QA and security in these quicker, shorter cycles (see the *Waterfall versus agile* diagram).
- DevOps is meant to bridge or reduce the need for specialized individual work.
  - When you begin to develop, you might notice that it is easy to become immersed in your own work. DevOps is meant to make it easier to do development work in a team.

aws re/start

What you will learn

At the core of the lesson

You will learn how to:

- Explain the need for automation
- Identify the states of CI/CD pipeline
- Describe continuous integration
- Describe continuous delivery
- Identify important features to look for in CI/CD tools

12

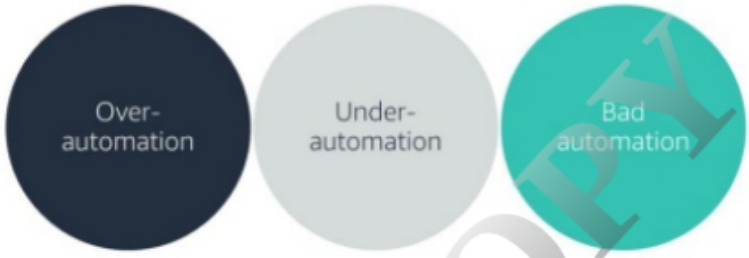
aws re/start

In this lesson, you will learn how to:

- Explain the need for automation
- Identify the states of CI/CD pipeline
- Describe continuous integration
- Describe continuous delivery
- Identify important features to look for in CI/CD tools

Automation

- When you develop software, it can be tedious and inefficient to perform the same tasks repeatedly. Automation can help solve this issue.
- The goal of automation is creative efficiency. However, automation has several risks that can undermine this goal:



- These concepts are explained in the next slide.

aws re/start

Automation: Risks

Over-automation

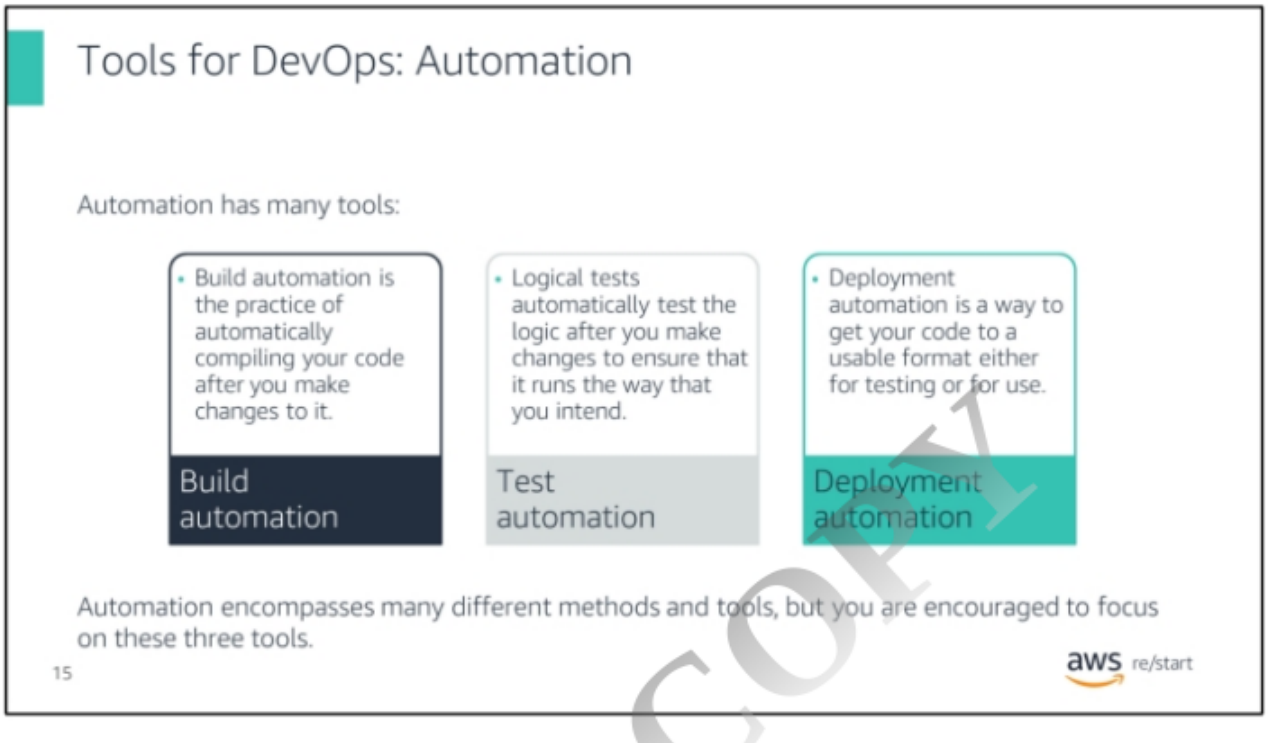
- Over-automation happens when you automate steps in the development process so that it reduces creativity. If you must think about and consider specific steps in a different way each time that you do them, you probably should not automate them—for example, analyzing, planning, and designing.

Under-automation

- Under-automation occurs when you avoid automation to make sure that things are handled correctly, or because it is helpful to find exactly where code stops working. Processes that are good to automate include building, testing, and deploying.

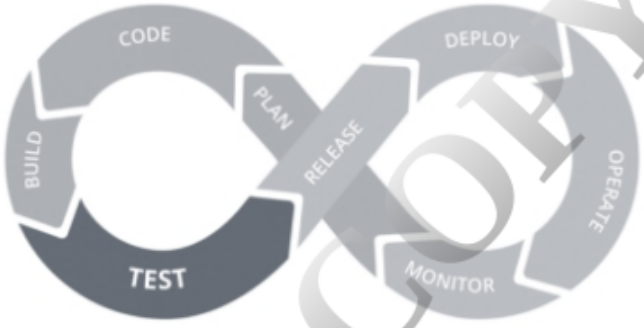
Bad automation

- Bad automation happens when you automate a process that does not work well. Bad automation can be fixed by revisiting the planning stage of development.

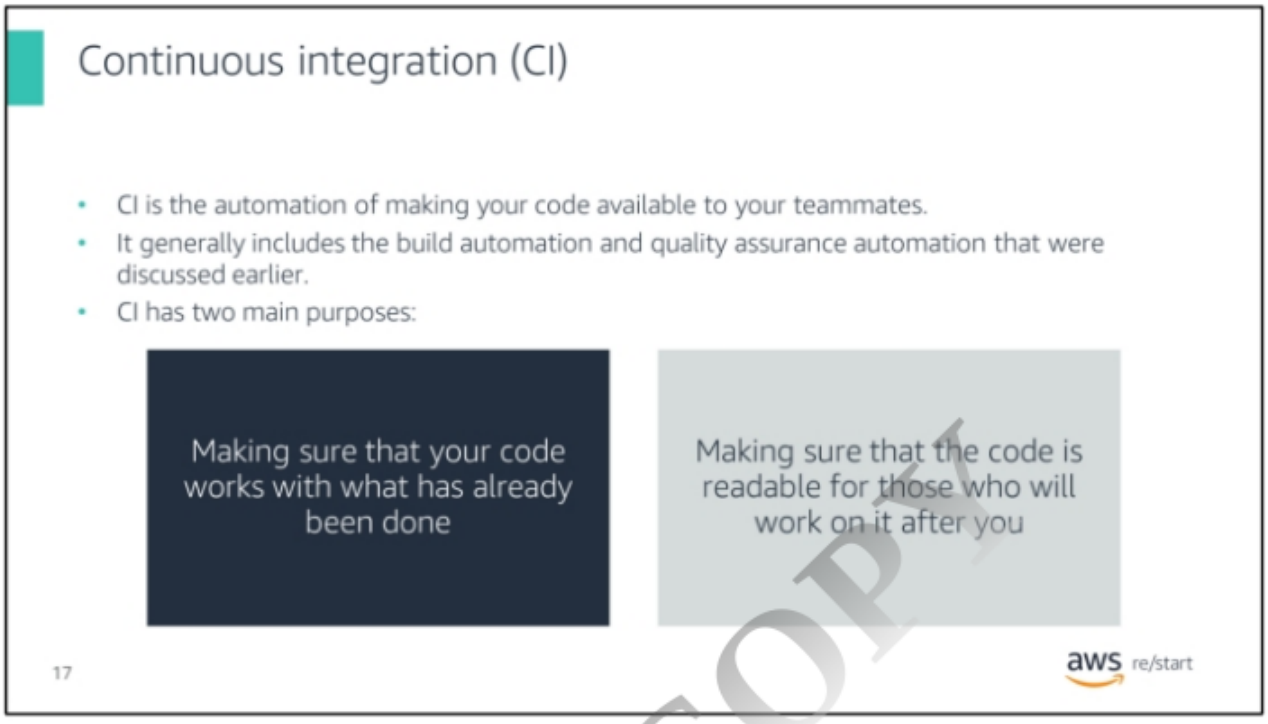


CI/CD pipeline

- The CI/CD pipeline is another tool for automation. The two parts are –
  - Continuous integration (CI)
  - Continuous delivery (CD)
- The next two slides cover CI and CD in detail.

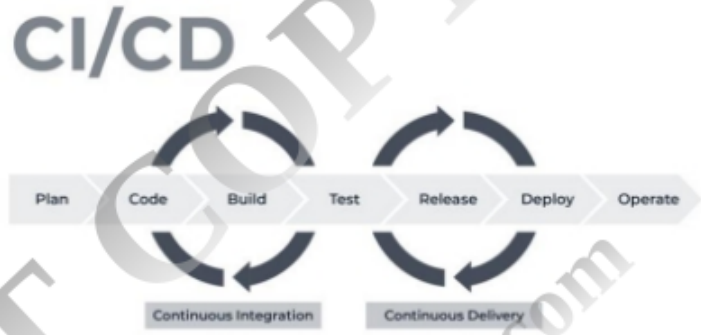


aws re/start

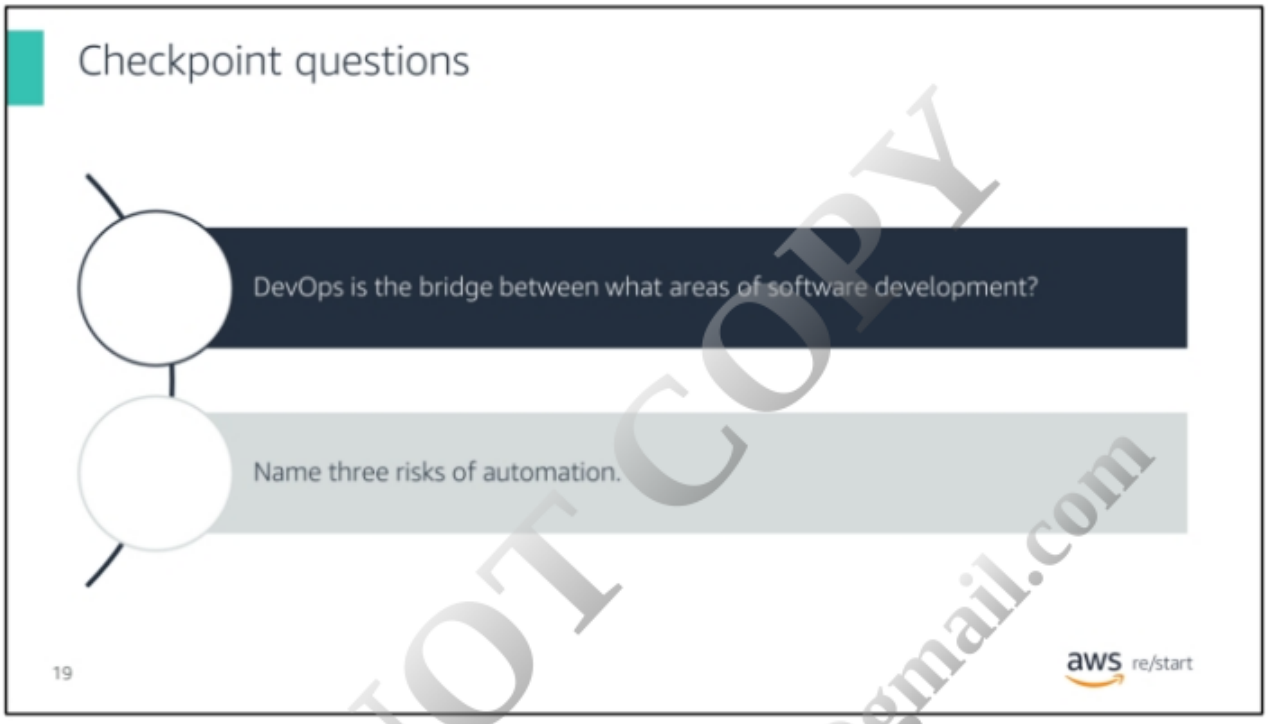


Continuous delivery (CD)

- CD is the extension of CI.
- CD includes a test automation for all code that is submitted. Its purpose is to ensure that the code works –
  - In the way that it was intended.
  - In a way that makes sense.
- CD also ensures that at any point in the development process, a working version of the code can be produced immediately.
  - This part is the deployment automation.

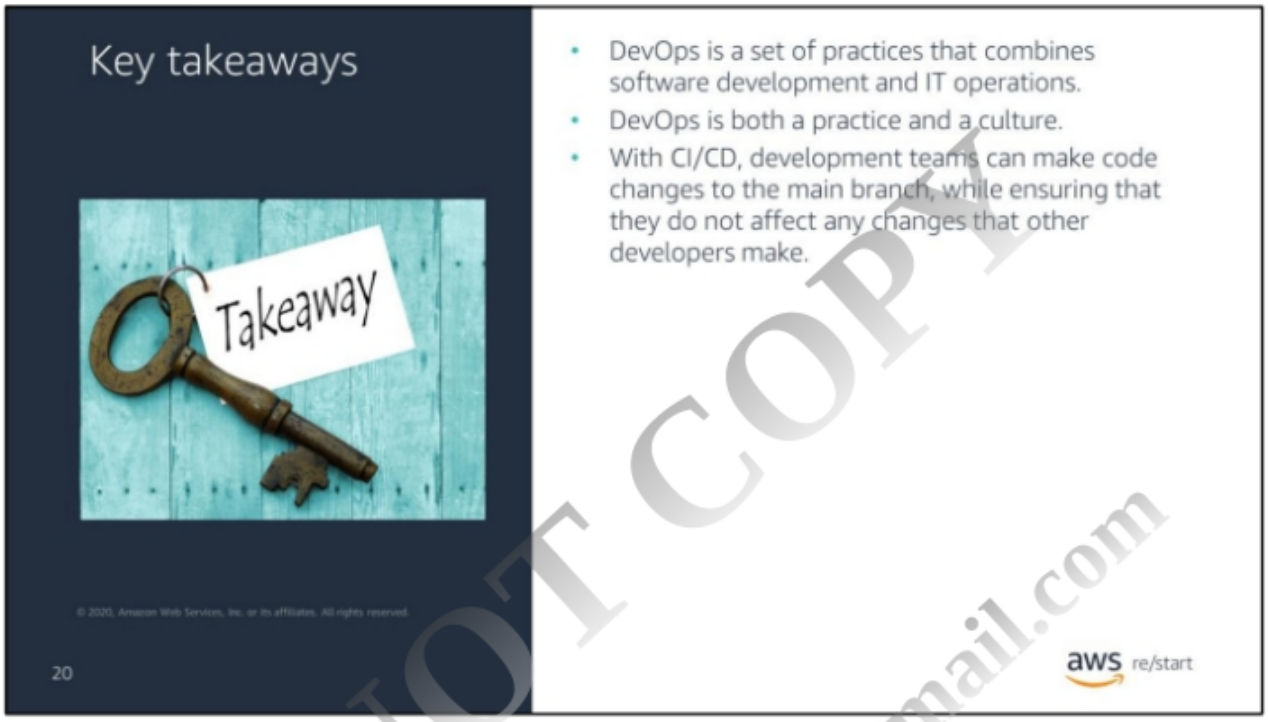


aws re/start



Answers:

1. DevOps bridges the gap between traditional IT, software development, and QA
2. Three risks of automation are:
   - Over-automation
   - Under-automation
   - Bad automation



Some key takeaways from this lesson include:

- DevOps is a set of practices that combines software development and IT operations.
- DevOps is both a practice and a culture.
- With CI/CD, development teams can make code changes to the main branch, while ensuring that they do not affect any changes that other developers make.