Introduction to Deployment

Learn how developers transform a problem into a developed solution and deliver it to end-users through the concept of deployment!

Introduction

Imagine we have just finished creating an e-commerce web application that allows users to buy and sell clothing. The application code, database, and assets (e.g., images) are located on a personal computer (our "machine"). How do we make our application available to users to interact with over the internet? That is where deployment comes into play!

We can think of **deployment** as a set of activities that make a piece of software available for other users. Before the invention of the internet, deployment looked like storing software on floppy disks or CD-ROMs, shipping them to users, and having those users manually install the software on their own devices. This process was slow and expensive, and many bugs slipped through the cracks. Today, software can be deployed via the internet with greater ease and speed of delivery than ever before. However, deployment isn't as simple as clicking a big red button labeled "deploy". There are multiple activities and processes involved to ensure that deployment occurs with no issues.

To explore deployment, in this article, we will explore:

Let's jump into the world of deployment!

Deployment in the Software Development Life Cycle (SDLC)

Before an application can be deployed, several steps must occur. For example, for our e-commerce application, we likely spent a large chunk of time planning features, writing code, and testing that the code works. These steps are part of a process known as the **Software Development Life Cycle (SDLC)**. The SDLC is a structured cycle of steps used to create high-quality software. While a few slightly different variations of the life cycle are used by software engineering teams, the phases typically include:

1. **Planning**: This first phase of the SDLC involves defining the problem to solve, and any objectives or requirements the software should meet are gathered.
2. **Defining/Analysis**: After developing a solid plan, information must be gathered before software engineers can create the new software. This could include defining what resources (like hardware or network) will be needed to run a prototype of the software or even research to find existing or similar software.
3. **Design**: In this phase, the technical details of the project are designed. The requirements gathered in the planning phase are transformed into concrete specifications.
4. **Development/Implementation**: The software starts to come alive within this stage as the code is built. This is when code is written to meet the specifications and goals of the software.
5. **Testing/Integration**: Testing is a crucial step in the SDLC. This step confirms that all of the software components are working seamlessly together. Any major issues or bugs are ideally caught during this stage prior to the application reaching the hands of the users.
6. **Deployment**: In this phase, a version of the software is packaged and made available so it can be used by other members of the development team (e.g., QA engineers), non-development team members (e.g., project managers), or real users. During the deployment process, the software can be tried out on different environments, like, for example, a testing environment only available to beta users (more on this later).
7. **Maintenance**: Lastly, once the software is out in the world, it is crucial to maintain it. This phase involves fixing bugs, as well as the continued development of new features. Any changes follow the same SDLC cycle of defining the problem (bug/feature), designing a solution, implementing the fix, testing, and deployment.

When visualized, the process looks like this:

Each phase of the SDLC can be broken down into a series of activities needed to complete that phase. For example, in the development phase, teams might engage in the following activities:

- Team members splitting up and assigning the work of specific features
- Team members writing the code for their assigned features
- Team members reviewing each other's code when it is complete

Similarly, the deployment phase also includes various activities. Let's dive a bit deeper into the deployment phase and explore what the typical deployment process looks like.

Typical Deployment Process

In most software engineering teams, developers deploy code across multiple **environments** during the development process. An environment is the subset of infrastructure resources (e.g., computers, memory) used to execute a program under specific constraints. Though the names of environments may vary, a common set of environments includes:

- The local development environment: This is where software is first written and tested, typically on a developer's own computer.
- The staging environment: This is where the software can be tested in a production-like environment, but before real users are involved.
- The production environment: This is where software is accessible by real users!

If we refer to the above SDLC process, this means that even though deployment is listed after the development and testing phases, in most cases, developers also deploy software between development and testing, typically into the staging environment. Once the code is properly tested in the staging environment, it can be deployed into the production environment. The deployment at this stage is typically also referred to as the **software release**, where it becomes available for non-development team users.

Additionally, most software engineering teams use automated deployment tools to handle the heavy lifting of moving software between environments and can deploy code multiple times a day (or on a custom schedule)! There are a vast number of deployment tools available. Some popular options include Render, Amazon Web Services, and Google Cloud Platform.

Before moving on, let's review:

Free response

In your own words, explain what deployment is and why it is important.

Your response

Deployment is the process of releasing software for being used by testers in the development team, by stakeholders of the project and finally by the real users.

Our answer

Deployment refers to the process of releasing software in a target environment (e.g., staging, production) so that it can be accessed and used by a set of end-users (e.g., project managers, customers). Without deployment, only development team members would be able to access the software.

Wrap-Up

In this article, we familiarized ourselves with what deployment is, how it fits into the SDLC, and what the typical deployment process looks like. Let's recap the main points:

- Deployment is a set of activities that make a piece of software available for use to a set of users
- The Software Development Life Cycle (SDLC) is a structured cycle of steps used to create high-quality software
- Deployment is one of the major phases of the SDLC and involves multiple activities
- Developers deploy code across multiple environments during the development process
- Software release is one of the activities of the deployment process
- Software engineering teams use a variety of tools to make deployment automated and efficient

Overall, understanding deployment is crucial for any software development project to ensure that high-quality software is delivered in a timely and efficient manner. By incorporating deployment activities into the SDLC, software engineering teams can streamline the process and minimize errors. With the use of automation tools, deployments can be done quickly and consistently. One great way to continue learning about deployment and related technologies is by exploring DevOps, which can provide in-depth insights and practical guidance on how to optimize the deployment process for maximum efficiency and effectiveness.