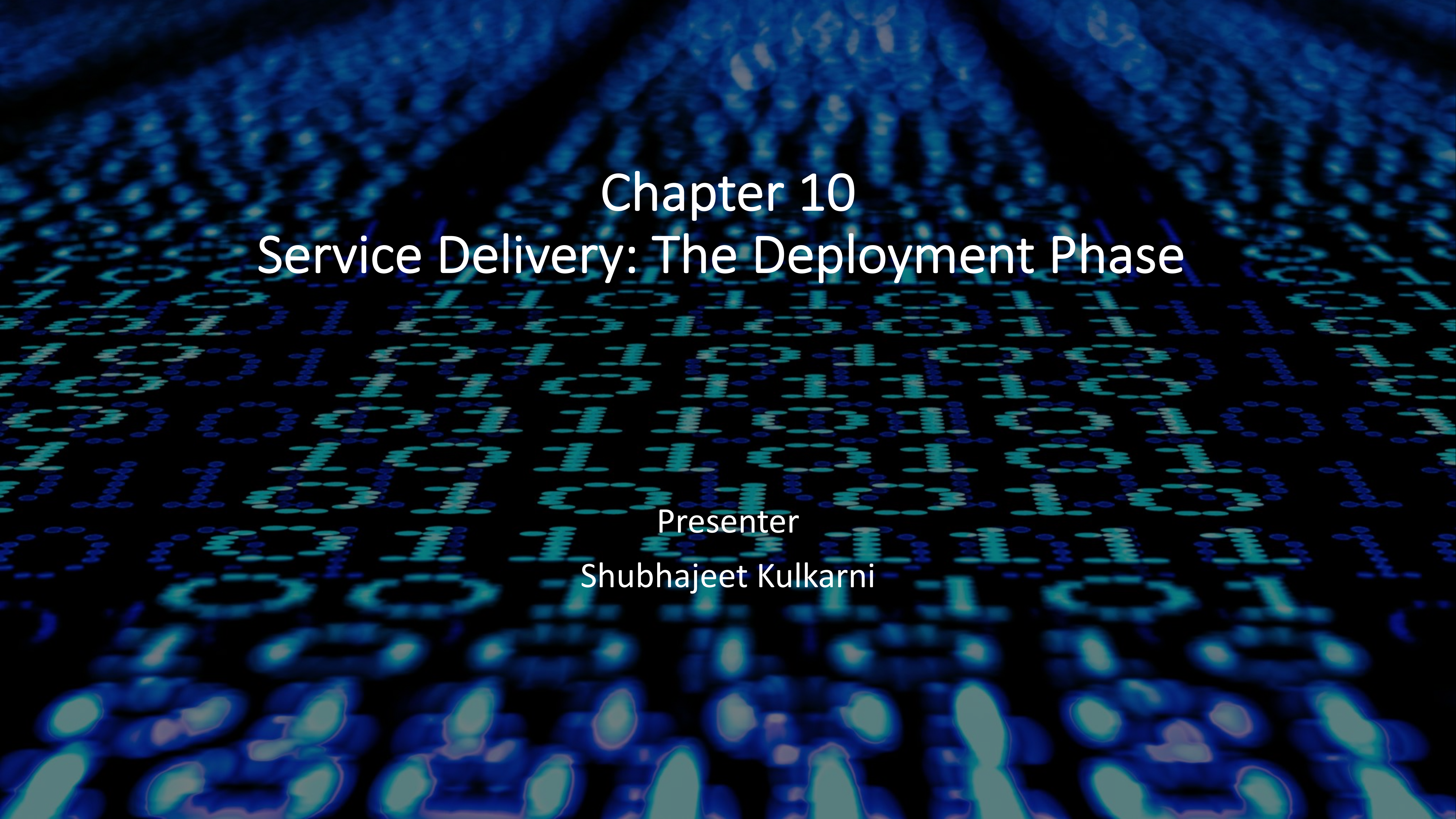

Chapter 10
Service Delivery: The Deployment Phase

Presenter

Shubhajeet Kulkarni

Topics to be covered

- Deployment phase & steps
- Testing & approval
- Operations console
- Continuous delivery
- Infrastructure as a Code
- Other platform services

Let er' Roll!

-Elvia Allman to Lucy and Ethel

Deployment phase

- The deployment phase creates the service in one or more testing and production environments. Deciding if a release used in the testing environment is ready to be used in the production environment requires approval.
- The goal of the deployment phase is to create a running environment. This environment is then used for testing or for live production services.

Step 1 : Promotion

- The **promotion** step is where a release is selected and promoted for use in the desired environment.
- The desired version is selected and marked as the right version for the environment being built.
- Package repository systems 'tags' at global level or 'pins' at environment level the latest version.
- All new packages will be put in a repository called “development.” When the package is ready to be used in the testing environment, it is copied to a repository called “testing.” All machines in the testing environment point at this repository. If the package is approved for use in production, it is copied to a third repository called “production”

Step 2: Installation

- In the **installation** step, the packages are copied to machines and installed. This is done by an installer that understands the package format. Most operating systems have their own installer software, each generally tied to its native package repository system.
- Installation process involves running the pre-install script, copying files from the package to their destination, and then running the post-install script.
- Pre-install scripts do tasks like creating directories, setting permissions, verifying preconditions are met, and creating accounts and groups that will own the files to be installed. Post-install scripts do tasks like copying a default configuration, enabling services, and registering the installation with an asset manager.

Step 3: Configuration

- In the **configuration** step, local settings and data are put in place to turn the installed package into the running service.
- While packages often include installation scripts that do some generic configuration, this step does machine-specific work required to create a working service.
- Convergent orchestration can be described as getting the environment to a particular state and keeping it there.
- Direct orchestration can be described as a method to execute a multistep process during which certain invariants hold true. For example, moving a database from one machine to another requires many steps that must happen in a certain order, all while the the invariant of “clients always have access to the database” remains true.

Testing & Approval

- Before a release is used in production, it must be tested and approved. First, automated testing is done. Next, manual testing, if there is any, is performed. Lastly, management approves or signs off on the release.
- The list of people or departments that must sign off on a release is called the **approval chain**. After all this activity is complete, the release can be promoted and pushed into production.

Step 1: Testing

- **System Testing:** This testing brings together all the various pieces of the service and tests the final product or system.
- **Performance Testing:** These tests determine the speed of the service under various conditions.
- **Load Testing:** This special kind of performance testing determines how much load the system can sustain.
- **User Acceptance Testing (UAT):** This testing is done by customers to verify that the system meets their needs and to verify claims by the producer.

Step 2: Approval

- If all the tests pass, the release is called a production candidate. Candidates are put through an approval process. If they are approved, they are installed in production.
- The production environment generally requires results of the preceding tests plus positive confirmation by the entire approval chain.
- Members of the approval chain are asked to sign off on the production candidate. For example, product and/or project manager. Often departmental approval is required from the security, legal, and privacy compliance for audit purposes.

Operations console

- The **operations console** is software that manages the operational processes, especially the deployment steps.
- A web-based system that makes it easy to view results, history and keeps statistics about success rates, process duration, and more.
- Security and authorization might be more important here than build phase because processes can affect live services. For example, tighter controls over who may initiate a launch for a new release into production.

Infrastructure Automation Strategies

- A few strategic tips will help you fully automate the deploy phase so that it can run unattended in the console
- Deploying the entire stack can be broken down even further: preparing and testing the physical or virtual machine, installing the operating system, installing and configuring the service.
- Each of these is a discrete step that can be automated separately.

Preparing physical machines

- Preparing a physical machine involves unboxing it, mounting it in a rack, cabling it, configuring BIOS settings, and testing.
- Blade servers are a technology made up of one chassis for many individual computers, each on a “blade,” which makes installation and maintenance easier.
- A quick-and-dirty solution is to manually install hardware and configure BIOS settings but automate the process of verifying that the settings are correct.
- Another strategy is to reduce complexity through standardization. Standardizing on a few hardware configurations makes machines interchangeable

Preparing virtual machines

- Preparing virtual machines should be a matter of making an API call.
- For example, one might allocate VMs in sizes such that either four small, two medium, or one large VM perfectly fills the physical machine. Then there is less chance that a physical machine may have some space unused, but not enough to create a new VM.
- One can also use multiples of Fibonacci numbers. If a 5-unit machine is deallocated, for example, that leaves room for five 1-unit machines, a 2-unit plus a 3-unit machine, and so on.
- This not only helps fully utilize the physical machines but also makes reorganizing them easier.

Installing OS & services

- The **image** method **(baked in)** involves creating a disk image for each kind of service. This image is then copied onto the disks and, after a reboot, the machine comes up preconfigured.
- The **configuration management (frying)** strategy involves using an installer or other mechanism to get a minimal operating system running.

Continuous delivery

- Continuous delivery (CD) is the technique in which testing is fully automated and triggered to run for each build.
- CD makes it economical and low risk to work in small batches, so that problems are found sooner and, therefore, are easier to fix.
- Continuous delivery is achieved when the deployment phase and all testing is automated. New release candidates are produced automatically, and confidence is improved as the testing becomes more and more extensive.

Infrastructure as a code

- The infrastructure for all environments should be built through automation that is treated just like any other service delivered by the SDP. This approach is called **infrastructure as code**.
- When the infrastructure as code technique is done correctly, the same code can build development, testing, and production environments.
- This minimizes the difference between the testing and production environments and makes testing more accurate.
- Infrastructure as code becomes easier to implement as hardware becomes more and more virtual.

KAHOOT TIME :)