DevOps Fundamentals

1. What is DevOps?

DevOps is a culture that improves the organization ability to deliver the application. DevOps is a process of improving the application delivery by ensuring that there is proper automation, quality, Continuous Monitoring/Observability and Continuous Testing.

1. Why DevOps?

Before 10 years, when the concept of DevOps was not available, still many teams involved in the process to deliver the code from Developer system to Production server.

Let’s say, we have system administrators, whose work is to create a server to deploy the application. Next, the BRE(Build & Release Engineer) comes to the picture and BRE used to fetch the code from the Git/CVS and deploy the application into the server. Now, the server administrator comes and creates an application server on top of the server, where our actual application will run. Still the process of Security, Monitoring etc. was there. This whole process is a manual process and will take around 10-15 days to complete. Hence, the concept of DevOps culture comes into the picture to automate the process by maintaining the quality with proper testing and monitoring.

Software Development Lifecycle (SDLC)

It is a process/standard followed by the software industry to design, develop and test high quality products. The end goal is to deliver a high quality product.

Planning -> Defining -> Designing -> Building -> Testing -> Deploy -> Planning…

1. Planning + Requirements -> Gather the requirements, Take inputs from your customers.
2. Defining Phase -> Defining means writing documents which called software requirement specification document.
3. Designing Phase -> Critical phase where our HLD(High Level Design) and LLD (Low Level Design) was specified. It is a critical phase because here everything needs to be related to the product should be designed, like which database you will use, which cloud provider services you need, how much scalable our software should be, etc.
4. Building Phase -> Developing phase. Developers reads the JIRA and starts writing the code and push the codes into the source code repository.
5. Testing Phase -> After the development, Testers/QA will test the application. They usually test in the Staging/Development server.
6. Deployment Phase -> Pre-Prod -> Production.

In the above phases of SDLC, where DevOps Engineer comes into the process?

DevOps comes mostly at the time of Building, Testing and Deployment phases, where DevOps fastens these processes by implementing Automation. So that these phases will smoothly run without any manual intervention.

Virtual Machines (VMs)

1. What is a server?

A server is a computer or system that provides resources, data, services, or programs to other computers, known as clients, over a network. Servers can offer a variety of functions, such as hosting websites, managing emails, storing files, running applications, and more.

Key characteristics of a Server:

1. Hardware or Software: Servers can be specialized hardware or software applications. In hardware form, servers are powerful computer designed to handle heavy workloads and run continuously without interruption. In software form, server applications run on standard computers to manage requests.
2. High Availability: Servers are typically designed for high uptime and reliability, as they often support critical services.
3. Networking: Servers operate within a network, often using protocols like HTTP, FTP, or SMTP to communicate with clients.

Types of Servers:

1. Web Server: Hosts and serves web pages to client via the internet. For example, Apache and Nginx are popular web server software.
2. Database Server: Manages and provides access to databases. Examples include MySQL, PostgreSQL, and Microsoft SQL Server.
3. File Server: Stores and manages files, allowing users on the network to share and access files.
4. Application Server: Runs applications and provides them to users on a network, such as ERP systems.
5. Email Server: Manages and routes email communication between clients.
6. What is a hypervisor?

A hypervisor also known as a Virtual Machine monitor, is software that creates and runs virtual machines (VMs). A hypervisor allows one of the host computer to support multiple VMs by virtually sharing its resources, such as memory and processing.

Popular Hypervisors are: VMware, Xen etc.

Example to understand Hypervisor:

Let us take an example from AWS, What AWS does is -> They creates physical servers in the different locations, for e.g., Mumbai, Singapore, Ohio etc. Suppose I am sitting in Hyderabad and I requested AWS that I want a VM of 10 GB RAM and 10 Core CPU in their Mumbai location Data center, so AWS takes my request and in one of their physical servers they send my request to the hypervisor to create the VM with the requested resource. Hypervisor, i.e., installed in the Physical server will create the VM with the requested resource and shares the IP address and the Key Value pair to the AWS and AWS sends us the same thing, so that we can login into the system and do our work.

1. How to create VMs in AWS, Azure, and on-premises?

You can create EC2 instances using AWS console (UI), but that is manual process and not efficient. Suppose you are DevOps engineer at your organization and you got 100 requests to create EC2 instances in AWS by different developers. Now it is not the efficient way to create those instances manually, rather that we can use the AWS EC2 API which is available and easy to use. You just need to use that API to send a request with the required details and the AWS will send you a response with an EC2 instance.

Ways to create EC2 instances using the API –

While writing the script to create EC2 instances, the request should be Valid (Follows the ways that API is expecting), you should be authenticated (should have account in AWS) and you should be authorized (Allows to create EC2 instances) to create EC2 instances.

You can write the script using:

* 1. AWS CLI
  2. AWS API (Any programming language you can use)
  3. AWS CFT (Cloud Formation Template)
  4. Terraform (Not only AWS, it supports Azure, GCP as well.) – If your organization follows hybrid cloud infrastructure, i.e., your VMs are in AWS, AI/ML stuffs are in GCP or Kubernetes in GCP. Likewise organizations follows hybrid model, then Terraform is best to use, in these cases.
  5. AWS CDK (Cloud Development Kit)

1. How to login to your EC2 instance?

You can connect to your EC2 instance with the connect button available in the AWS console, but that is not convenient.

Another way is to login through a terminal, because as a part of organization you must deal with many instances, and it is not convenient to go to the console every time to login to your instance.

Now, it is important that in which laptop/PC you are using. If you are using Windows laptop, install Mobaxterm which is widely use.

If you’re in mac -> Open Terminal

**ssh -i <PEM File Location> ubuntu@<PUBLIC\_IP\_ADDRESS>** -> Click Enter.

When you click enter if will give you an error, that your .pem file is too open, and it should not be this much of permission.

*To change the permission:* ***chmod 600 <PEM\_FILE>***

Again, when you enter the same command, it will allow you to login into your EC2 instance.