**ELB (Elasticity Load Balancer)** distributes load to all instances equally (EC2 instances)

**EC2 (Elasticity Cloud Computing)** for creating multiple instances (virtual machines)

**EBS (Elasticity Bean Stack)** everything is automatically setup for application,

No need to worry about ELB, Storage, etc.

**EKS (Elasticity Kubernetes Service)** used to run microservices.

**ECS (Elasticity Container Service)** used to run microservices.

**RDS (Relational Database Service)** provides database services.

**S3 (Object Storage Service)** used to store files (static websites, images, videos etc.)

**Regions**  places where datacenters are built.

**Zones** multiple places in a region

**ASG** auto scaling of resources.

**Route53** Domain naming service

**CloudFront** to cache and distribute data across the world.

**VPC (Virtual Private Cloud)** we can control both incoming and outgoing traffic.

**Route Tables** To define rules of VPC.

**Container:**

Before deploying an application, we need to setup **VM, OS, AR, Code**. But now everything is simplified by **Container Image.**

**Container Image** has **VM, OS, AR, Code.** We can run this image anywhere (i.e., local, cloud)­

**Microservices Architecture:**

Container Orchestration tools (**EKS, ECS**) are used to run microservices. **EKS (Elasticity Kubernetes Service)** & **ECS (Elasticity Container Service)** are used in AWS.

First, we need to create a cluster, in the cluster there are multiple nodes. To this cluster we can deploy our services.

Adding cluster & nodes is our responsibility. If we need serverless, then we can use **Fargat.**

**Create Load Balancer, Cluster, Deploy, Install Application on server:**

Serverless is one of the approaches, here we don’t need to worry about server. For this serverless approach we use **AWS Lambda. (Handles auto scaling, pay for what you use)**

**Setting Up Database:**

Setting up database by EC2 instance is a hectic process, we must manage data, DB runtime, OS, regions everything by ourselves.

So, **Managed Database Services** are a way to escape from such a hectic process.

**RDBMS RDS (Relational Database Service)**

**Aurora (**Global Relational Database**)**

**Nonrelational DynamoDB**

**Analytical Databases Hadoop, RedShift**

**AWS Data Pipeline (BI)** reads data from **Transactional DB** and loads it to **Analytical DB. (AWS Quick Sight)**

Attach storage to virtual machines [**EBS (Elastic Block Store), EFS (Elastic File System)**]

EBS is for single instance, EFS is shared among all instances.

To **Store Files**, databases are not optimized. So, we use **Object Storage (S3)** where files are stored in key-value pairs. We can store any files such as images, videos, audio etc.

**Key unique id for file**

**Value actual files**

**Security:**

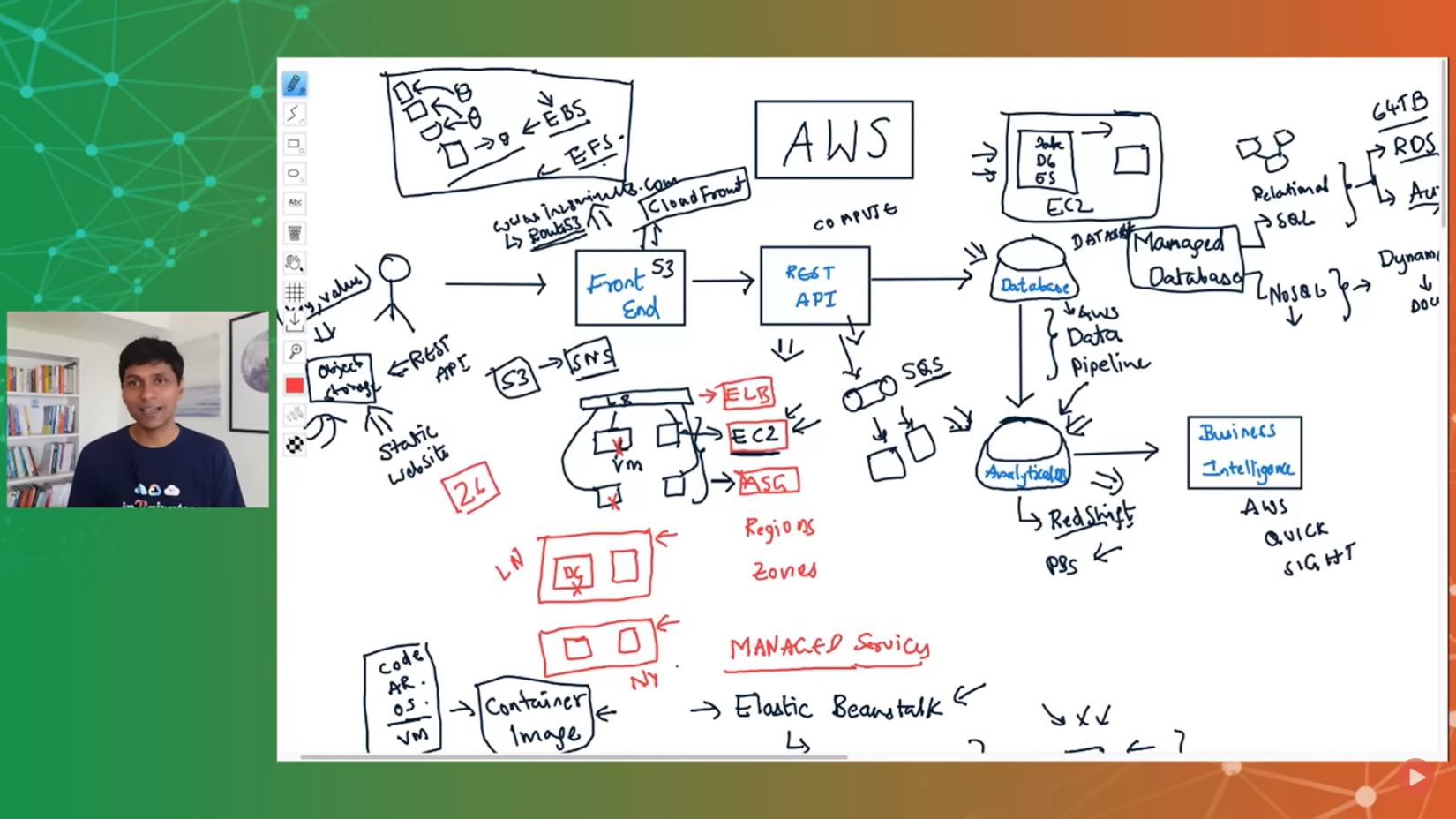
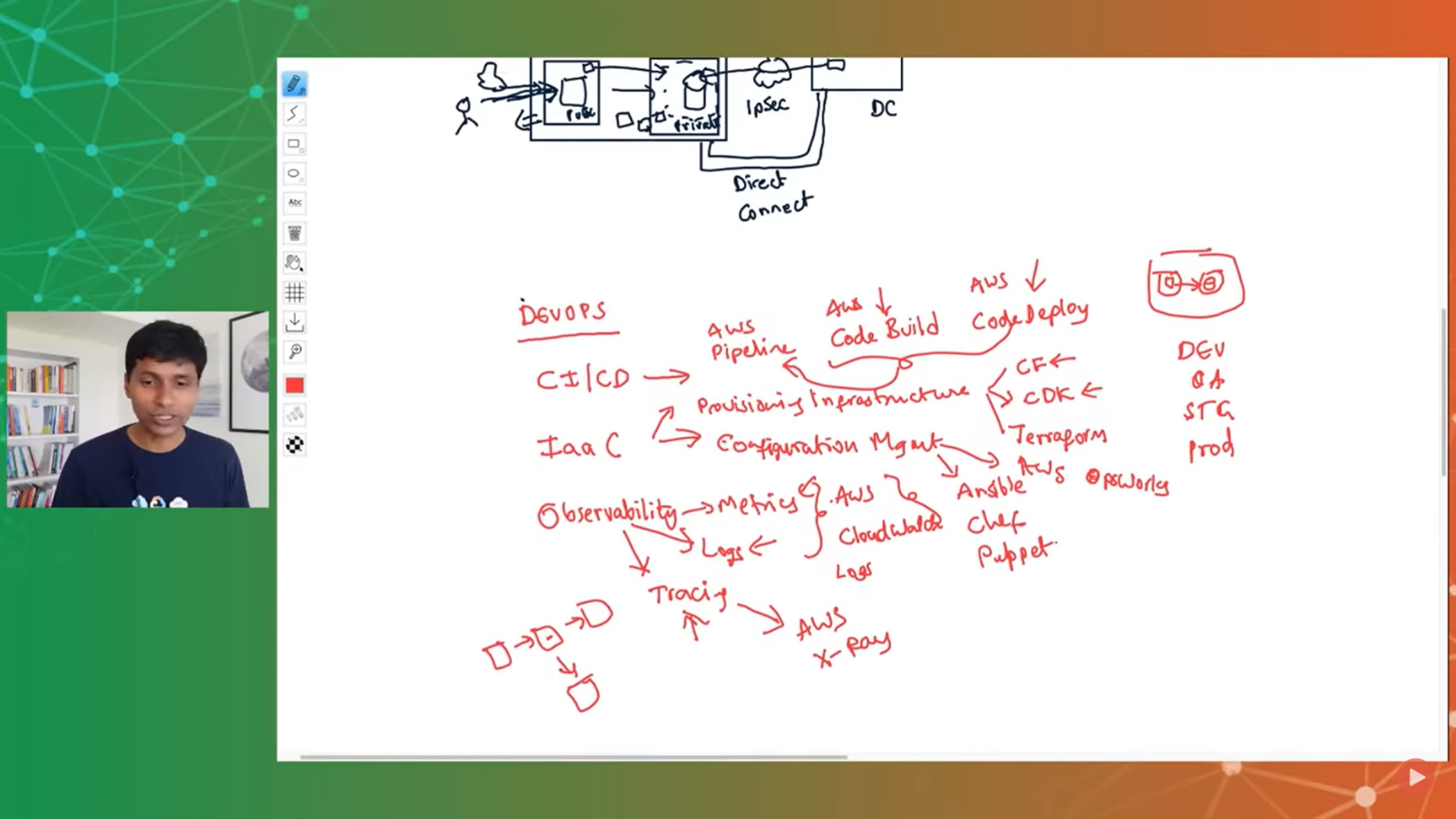
We can handle security using **VPC** & **Subnets**.

**VPC (Virtual Private Cloud)** we can control both incoming and outgoing traffic.

**Subnets** allows us to configure different rules for different things.

**Private Subnet** can be accessed by certain users.

**Public Subnet** can be accessed from the outside world.

****