1. **Agenda**:
   1. What is Docker?
   2. Architecture?
   3. How does it help to create images and container instances?
   4. Different components inside the Docker?

Diagram

Description automatically generated

1. **Docker Client (Component)**:
   1. You need a client to issue command to **Docker Server/Host**.c
   2. **Clients**: Using either of these two clients, you can issue docker commands to Docker Host/Server.
      1. **Docker CLI**: Like Git CLI or AWS CLI.
      2. **Docker Remote API**:
2. **Docker Host/Server**:
   1. Installing Docker on our OS will give us Docker Server.
   2. Responsible to maintain all the images and containers on our system.
   3. **How does it work**?
      1. First it will build an image of the app we have in hand.
      2. Then, it creates container inside which our app runs and it also exposes an endpoint if it is web-app.
      3. Image is like Java Class and container is like that java class instance.
      4. **Advantage**:
         1. **Portability & Compatibility**: Can move image from one env to another and run there without any issue.
3. **Docker Registry**:
   1. Now the question is that **how to share the images** with your colleagues/friends?
   2. Docker has its own **registry** which we call **Docker Hub** which is a public place/registry.  
      Anyone can push and pull the images from **public registry**.
   3. **Private Registry**: Images are secured from public.