(a) Describe the three primary cloud service models in cloud computing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Provide specific examples of how each model can be applied in the context of software development.

Infrastructure as a Service (IaaS):

IaaS provides virtualized computing resources such as virtual machines, storage, and networking over the internet. Users can rent these resources and have full control over the operating systems and applications they install, making it suitable for organizations that need flexibility in managing their infrastructure.

Example: Amazon Web Services (AWS) EC2. Developers can use EC2 to deploy virtual machines that run their applications or development environments. This allows them to focus on building and scaling their applications without managing the physical hardware.

Platform as a Service (PaaS):

PaaS provides a platform that allows developers to build, deploy, and manage applications without worrying about the underlying infrastructure. It abstracts away the complexity of managing servers, databases, and networking, giving developers more time to focus on application development.

Example: Google App Engine. Developers can deploy web applications to App Engine, and the platform handles all the server management, scaling, and load balancing. Developers don’t need to worry about managing hardware or server configurations.

Software as a Service (SaaS):

SaaS delivers fully functional software applications via the internet. Users can access these applications on-demand, typically through a web browser, without the need for installation or maintenance on their own systems. The service provider manages the software and infrastructure.

Example: Google Workspace (e.g., Gmail, Google Docs). Software developers and teams can use these tools for email communication, document collaboration, and project management, without needing to manage or maintain the underlying software.

(b) What is Docker? Describe a scenario where you would use containerization technologies such as Docker in software development. How does containerization contribute to the development and deployment process?

Docker is an open-source platform that automates the deployment of applications inside lightweight, portable containers. Containers package an application and its dependencies into a single, isolated unit that can run consistently across different computing environments. Docker simplifies the process of moving applications from development to production without worrying about environment discrepancies.

Scenario:

Imagine you are developing a web application on your local machine. Everything works perfectly in your development environment. However, when you deploy it to the production environment, you encounter issues related to dependencies and configurations that were not present on your local machine. By using Docker, you can package the application and all its required dependencies into a container. This container can be deployed consistently on any environment—whether it’s your local machine, a test server, or a cloud platform—ensuring that the application behaves the same everywhere.

Benefits of Containerization:

Consistency: Containers ensure that the application runs the same way in all environments, avoiding the "it works on my machine" problem.

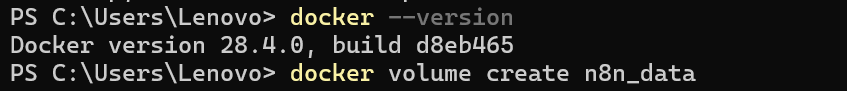
Portability: Docker containers can run on any system that supports Docker, including cloud servers, local machines, or on-premises servers.

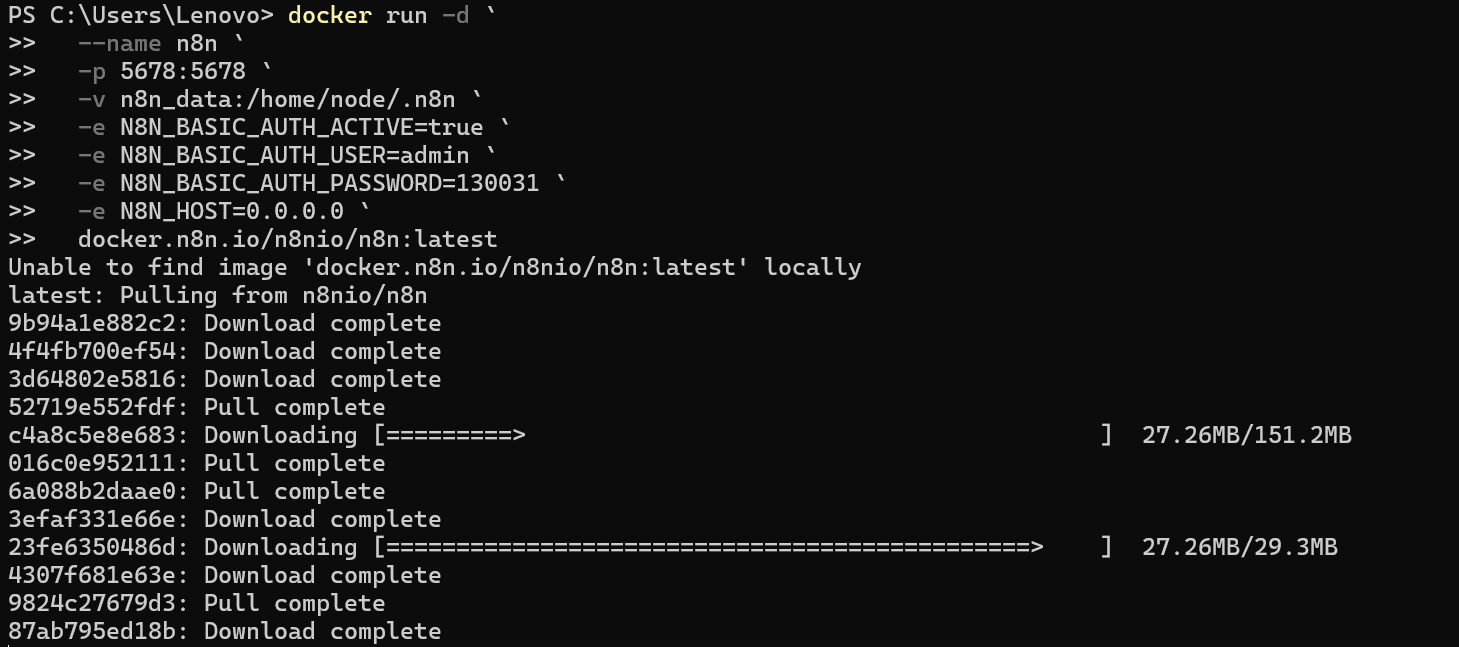
Efficiency: Docker containers are lightweight compared to virtual machines, allowing you to run multiple containers on a single host without excessive overhead.

By using Docker, developers can ensure that their applications are portable, consistent, and easier to deploy, leading to smoother workflows and faster development cycles.

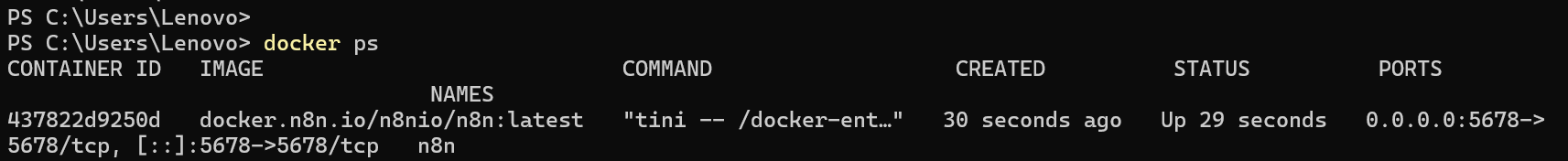
(c)

1. Pull the n8n Docker image





2. Start the n8n container



3. Visit n8n

