

Dockerizing Spring Boot Applications

Common developer frustration

"It works on my machine, but fails in production."

Docker resolves this by isolating applications with everything they need (runtime, libraries, dependencies) to run consistently across environments.

The Problem

Redundant Configurations

Manual setup of each production server (dependencies, libraries).

Time-consuming and **prone to errors.**

Dependency Hell:

Applications requiring conflicting dependency versions.

Forced to use **outdated or unsupported versions to avoid conflicts.**

Inconsistent Environments:

Apps working on a developer's machine but failing in production due to **incompatible processes and configurations.**

Slow Deployment:

Manual configurations delay deployment cycles.

Difficult to scale applications rapidly.

Scalability Issues:

Spinning up or tearing down environments to match demand is cumbersome.

The Solution - Docker

What is Docker?

Open-source containerization platform transforming application deployment.

Packages applications with their dependencies, libraries, and runtime into lightweight, isolated containers.

How Docker Solves Problems:

Eliminates the "It works on my machine" problem with consistent environments.

Avoids dependency conflicts by isolating applications.

Streamlines deployment – production servers need only Docker installed.

Enables easy scaling with lightweight containers.

Basic Docker Terminologies

Image:

Lightweight, immutable package containing application code, dependencies, and runtime.

Built using a **Dockerfile** with defined instructions.

Container:

Running instance of a Docker image.

Self-contained, lightweight, and portable across environments.

Docker Registry:

Central storage for Docker images.

Examples: **Docker Hub** (public and private repositories).

Containers vs Virtual Machines

Virtual Machines:

Full operating systems running on virtual hardware.

Resource-intensive and slower to start.

Containers:

Lightweight environments sharing the host OS kernel.

Faster, more efficient, and easier to manage.

Benefits of Docker

Consistency:

Applications behave the same way across development, testing, and production.

Portability:

Containers can run anywhere, from a developer's laptop to cloud servers.

Isolation:

Prevents conflicts between applications and dependencies.

Collaboration:

Fosters better teamwork between developers and operations teams.

Scalability:

Spin up or tear down instances quickly to match user demand.

Conclusion

Docker eliminates deployment pain points like "dependency hell" and inconsistent environments.

Ensures faster, more reliable deployments with scalability and portability.

Problem Statement Before Docker

Inconsistent Environments

It works on my machine!!

Heavy Virtual Machines (VMs)

Slow Deployment

Dependency Conflicts

Portability Issues

How Docker Resolved These Problems

Docker is an open-source containerization platform.

Lightweight Containers: Containers share the host OS kernel, using fewer resources and starting faster than VMs.

Consistent Environments: Docker images package the application with its dependencies, ensuring consistent behavior across environments.

Portability:

Containers can run anywhere, from a developer's laptop to cloud servers.

Isolation:

Prevents conflicts between applications and dependencies.

Collaboration:

Fosters better teamwork between developers and operations teams.

Scalability:

Spin up or tear down instances quickly to match user demand.

Conclusion

Docker eliminates deployment pain points like "dependency hell" and inconsistent environments.

Ensures faster, more reliable deployments with scalability and portability.

Problem Statement Before Docker

Inconsistent Environments

It works on my machine!!

Heavy Virtual Machines (VMs)

Slow Deployment

Dependency Conflicts

Portability Issues

How Docker Resolved These Problems

Docker is an open-source containerization platform.

Lightweight Containers: Containers share the host OS kernel, using fewer resources and starting faster than VMs.

Consistent Environments: Docker images package the application with its dependencies, ensuring consistent behavior across environments.

Portability:

Containers can run anywhere, from a developer's laptop to cloud servers.

Isolation:

Prevents conflicts between applications and dependencies.

Collaboration:

Fosters better teamwork between developers and operations teams.

Scalability:

Spin up or tear down instances quickly to match user demand.

Conclusion

Docker eliminates deployment pain points like "dependency hell" and inconsistent environments.

Ensures faster, more reliable deployments with scalability and portability.

Problem Statement Before Docker

Inconsistent Environments

It works on my machine!!

Heavy Virtual Machines (VMs)

Slow Deployment

Dependency Conflicts

Portability Issues

How Docker Resolved These Problems

Docker is an open-source containerization platform.

Lightweight Containers: Containers share the host OS kernel, using fewer resources and starting faster than VMs.

Consistent Environments: Docker images package the application with its dependencies, ensuring consistent behavior across environments.

Portability:

Containers can run anywhere, from a developer's laptop to cloud servers.

Isolation:

Prevents conflicts between applications and dependencies.

Collaboration:

Fosters better teamwork between developers and operations teams.

Scalability:

Spin up or tear down instances quickly to match user demand.

Conclusion

Docker eliminates deployment pain points like "dependency hell" and inconsistent environments.

Ensures faster, more reliable deployments with scalability and portability.

Problem Statement Before Docker

Inconsistent Environments

It works on my machine!!

Heavy Virtual Machines (VMs)

Slow Deployment

Dependency Conflicts

Portability Issues

How Docker Resolved These Problems

Docker is an open-source containerization platform.

Lightweight Containers: Containers share the host OS kernel, using fewer resources and starting faster than VMs.

Consistent Environments: Docker images package the application with its dependencies, ensuring consistent behavior across environments.

Portability:

Containers can run anywhere, from a developer's laptop to cloud servers.

Isolation:

Prevents conflicts between applications and dependencies.

Collaboration:

Fosters better teamwork between developers and operations teams.

Scalability:

Spin up or tear down instances quickly to match user demand.

Conclusion

Docker eliminates deployment pain points like "dependency hell" and inconsistent environments.

Ensures faster, more reliable deployments with scalability and portability.

Problem Statement Before Docker

Inconsistent Environments

It works on my machine!!

Heavy Virtual Machines (VMs)

Slow Deployment

Dependency Conflicts

Portability Issues

How Docker Resolved These Problems

Docker is an open-source containerization platform.

Lightweight Containers: Containers share the host OS kernel, using fewer resources and starting faster than VMs.

Consistent Environments: Docker images package the application with its dependencies, ensuring consistent behavior across environments.

Portability:

Containers can run anywhere, from a developer's laptop to cloud servers.

Isolation:

Prevents conflicts between applications and dependencies.

Collaboration:

Fosters better teamwork between developers and operations teams.

Scalability:

Spin up or tear down instances quickly to match user demand.

Conclusion

Docker eliminates deployment pain points like "dependency hell" and inconsistent environments.

Ensures faster, more reliable deployments with scalability and portability.

Problem Statement Before Docker

Inconsistent Environments

It works on my machine!!

Heavy Virtual Machines (VMs)

Slow Deployment

Dependency Conflicts

Portability Issues

How Docker Resolved These Problems

Docker is an open-source containerization platform.

Lightweight Containers: Containers share the host OS kernel, using fewer resources and starting faster than VMs.

Consistent Environments: Docker images package the application with its dependencies, ensuring consistent behavior across environments.

Portability:

Containers can run anywhere, from a developer's laptop to cloud servers.

Isolation:

Prevents conflicts between applications and dependencies.

Collaboration:

Fosters better teamwork between developers and operations teams.

Scalability:

Spin up or tear down instances quickly to match user demand.