DevOps Pipeline Task: Automating Deployment to AWS

Overview:

You are given an application that requires a **database** and a **Nginx web server**. The goal is to **deploy** the application on **AWS**, ensuring that it is scalable, and can be easily replicated across multiple servers.

Your task is to design and implement an automation pipeline to handle the following:

1. **Containerize the application** along with its dependencies (database, Nginx, etc.) and deploy it to AWS.
2. **Ensure scalability** by enabling the ability to spin up **multiple servers** of the application.
3. **Automate the deployment process** as much as possible, from code to deployment.

You are free to choose the tools, services, and technologies that you feel are most appropriate for this task. However, here are the key requirements:

- **Database**: The application needs a database. You should decide what type of database to use and how to manage it (e.g., MySQL, PostgreSQL, etc.).
- **Nginx**: A Nginx server must be used to serve the application.
- **AWS Deployment**: The solution must be deployed on AWS, with the application and its services running on AWS infrastructure.
- **Scalability**: The system should be able to spawn multiple instances of the application, and its dependencies as required.
- **Automation**: You should automate as much of the process as possible, including building, deploying, and scaling the application.

Considerations:

- **Package**: How will you package the applications and ship them?
- **Deployment Strategy**: How will you deploy the application to AWS? What AWS services will you use?
- **Scaling**: How will you handle scaling to spawn multiple instances of the application? Will you use load balancers, auto-scaling groups, etc.?
- **Automation**: What tools will you use for automation?

You are free to choose the architecture, tools, and approach that you believe is the best solution to this problem.

Deliverables:

- A clear, documented solution with the steps you took to implement the automation pipeline.
- Source code, configuration files, and setup instructions for the entire solution.
- Documentation of how the solution handles scalability and deployment.