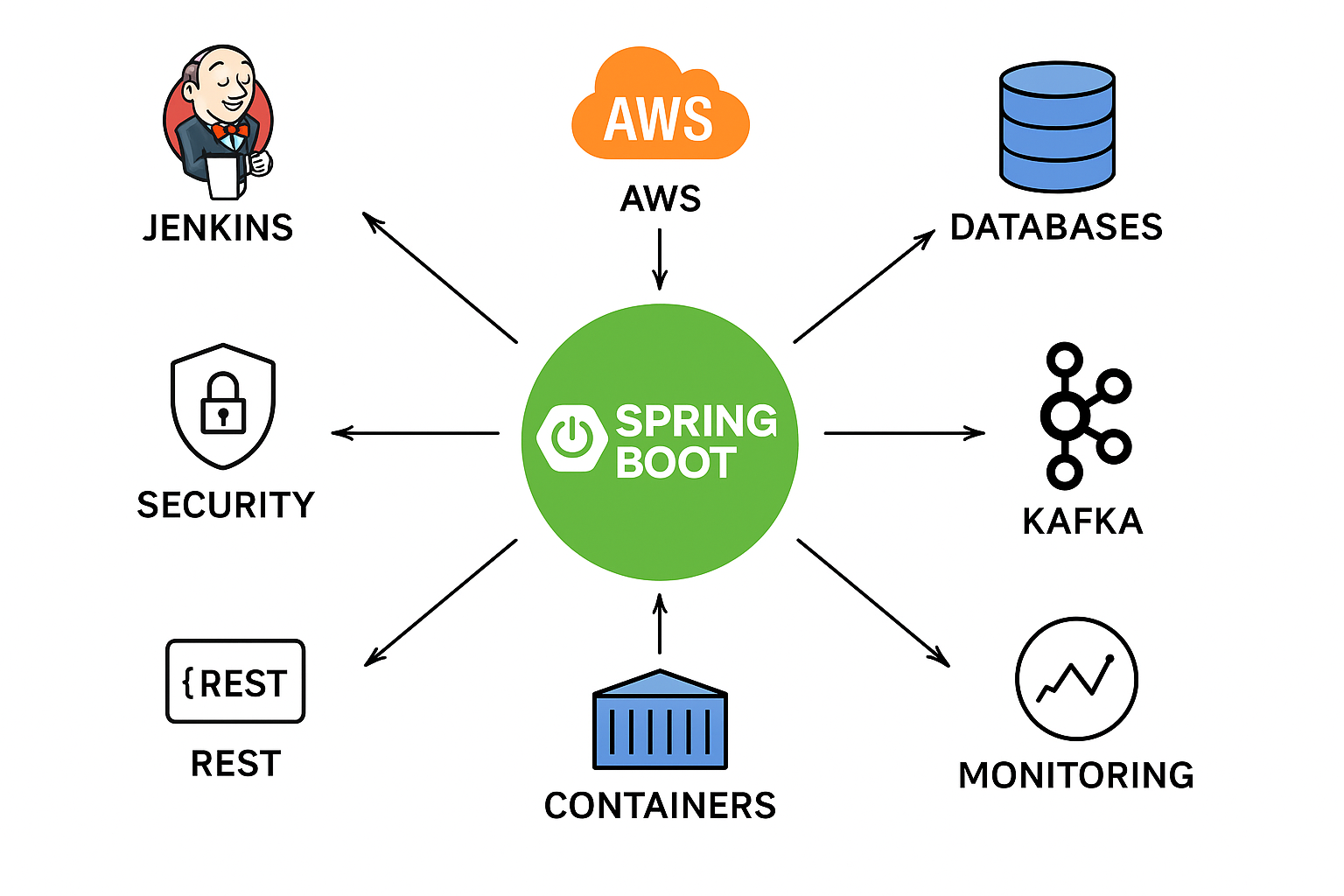
**

**Step 1:** **CI/CD Pipeline (Jenkins/ Github Actions)**

When developer commits code to GitHub, GitHub Actions triggers builds/tests. Then compiles the code (via Maven/Gradle), runs unit tests, builds a Docker image, and pushes it to a container registry (ECR, GCR, Docker Hub).

**Step 2:** **Cloud Hosting (AWS/GCP)**

After building the image, it deploys the Spring Boot app to any of the below

* AWS: Elastic Beanstalk, ECS, EKS, or Lambda.
* GCP: App Engine, Cloud Run, or Kubernetes (GKE).

These platforms run the application and handle scaling, networking, and load balancing.

**We can generally talk about the databases,** **messaging, APIs:**

**Databases:** Spring Boot uses Spring Data JPA or Spring Data MongoDB/Redis to read/write data. **Some example** DBs can be MongoDB, SQL.

**Messaging:** Kafka is used for event-driven communication between services. Spring Boot apps publish events to Kafka topics or consume events from them. **Some example** is ServiceNow pushes message to Kafka topic, spring boot will be designed to listen to those specific topics, consume them and store or send to DBs. And then publishes the message back to serviceNow as response. This happens asynchronously where it doesn’t wait until response is received.

**APIs:** Helps to fetch the data from Databases**. Some Example** is we used the graphQL with OAUTH to fetch data from Mongo DB database to backend micro services.