**Pipeline image explanation:**

1. The engineer codes using an IDE
2. The IDE uses several plugins to help during coding
   1. Security plugins
      1. HashiCorp Vault to connect to a HashiCorp Server and store password
      2. Linters plugins for code linting
      3. Snyk plugin to provide real-time insight and alerts when vulnerable code is imputed
   2. API plugin
      1. OpenAPI (Swagger) plugin to help develop APIs according to best practices.
3. Pre-commit hooks to perform linting checks. If checks pass, commits can pursue.
4. Commit goes to a remote GitHub repository (code repository or configuration repository)
   1. The GitHub repository will perform security processes such as storing code to HashiCorp Vault, branch protection, linting, static code analysis (scan by ShiftLeftSecurity), software composition analysis (Dependency Track)
5. When code (web application, APIs) goes to GitHub Code repository, if there a pull request to main branch is accepted, then Jenkins will be triggered on the main branch
6. Jenkins will first run a build stage using Gradle
   1. Gradle will compile the code, then perform unit testing using popular testing tools that are relevant to the code being tested (e.g. JUnit if code is in Java).
   2. Gradle will archive the file (e.g. WAR file if it is in Java. WAR files are for web applications)
7. Jenkins will integrate the archive into a Docker manifest file
   1. Docker manifest file will be used to build a Docker image
8. Jenkins will use Syft and Grype plugins to perform vulnerability scanning on the built Docker image
   1. Any failed checks will cancel the build
9. Jenkins will perform integration and functional tests on the Docker image using Selenium plugin
   1. Any failed checks will cancel the build
10. Jenkins will push the hardened docker image to GitHub Container Registry
11. Jenkins Will update the Kubernetes Manifest File with the latest Docker image tag
12. GitHub configuration repository will receive either updates to Docker images, Terraform files or Kubernetes Files
13. ArgoCD will detect that the code in the configuration repository is different from what is in the real world (deployment environment) or vice versa.
    1. If the code is different, ArgoCD will pull changes from the GitHub configuration repository and apply the changes to the environment.
    2. If the environment is different, ArgoCD will revert the environment back to the initial state specified in the code.
14. Kubernetes will manage multiple pods that are load balanced and will undergo advanced deployments methods (canary deployment, Blue-green, etc.)
    1. Web Application microservices to build the web application
    2. API microservices for APIs
    3. Database microservices for storage
       1. Store data necessary for the microservices to run
       2. Uploads batch jobs to AWS environment
15. AWS environment will receive batch jobs through AWS Direct Connect
    1. AWS S3 bucket receives batch jobs.
    2. The event is picked up by S3 trigger
    3. Automatic and cost-effective triggering of batch jobs to AWS Lambda
       1. If a batch job is too long to execute inside AWS Lambda, it will be processed by AWS batch, thus reducing cost.
    4. AWS CloudWatch will monitor S3 bucket and send metrics to Prometheus
    5. AWS IAM will run to achieve identity and access management.
16. Prometheus pod will collect metrics from the AWS environment as well as the Kubernetes cluster
    1. Prometheus forwards metrics to Grafana pod
    2. Grafana will be used by an Engineer for metric visualization
17. KubeFlow will be managed by Kubernetes and will use PyTorch or TensorFlow to facilitate operations predictions capabilities (AI/ML).
18. Kubecost pod will be managed by Kubernetes and will be used for FinOps purposes.
19. OpenSCAP, Nmap, and OWASP ZAP pods managed by Kubernetes will be used for dynamic application security testing and compliance as code.
20. The development environment will run inside a virtual server, which will be hosted in a physical server.
21. By using infrastructure as code, Test and Production environments will be similar to the development environment.