Simple Kubernetes Deployment Flow (Helm + EKS)

1. Developer Pushes Code

2. Developer pushes changes to a branch (e.g., `main`) in GitHub.

3. GitHub Actions Workflow Triggers

4. The workflow automatically runs on push.

5. Run Tests

6. Unit tests or integration tests to ensure code is stable.

7. Build & Tag Docker Image

8. Docker image is built and tagged with the Git commit SHA (`$GITHUB_SHA`).

9. Push Docker Image to Registry

10. Authenticate and push the image to ECR (or any container registry).

11. Update EKS kubeconfig

12. Connect to the target EKS cluster using AWS CLI.

13. Deploy Using Helm

14. Run `helm upgrade --install` to deploy the app to Kubernetes.

15. Use Git SHA as the image tag to ensure versioned rollouts:

```
helm upgrade --install my-app ./helm-chart
  --namespace prod
  --set image.tag=$GITHUB_SHA
```

1. Application Running on EKS

2. Kubernetes pods start with the new image.

3. Helm can roll back automatically if deployment fails.

Flow Diagram:

Developer Push Code -> GitHub Actions Trigger -> (Optional Tests) -> Build & Tag Docker Image -> Push Image to Registry -> Update kubeconfig for EKS -> Helm Upgrade/Install -> Application Running on EKS (Git SHA versioned)