Real Project: Kubernetes + ArgoCD + Grafana Monitoring Automation

1. System Environment Overview

The project was developed and tested locally on a personal workstation running **Ubuntu 22.04 LTS** with the following configuration:

- ■ RAM: 22 GB (available memory ~19 GB free during runtime)
- ■■ OS: Ubuntu 22.04 LTS (64-bit)
- ■■ Minikube: Local single-node Kubernetes cluster (v1.34+)
- ■■ Kubectl: Installed and configured to access the local Minikube cluster
- ■ ArgoCD: Installed in namespace `argocd` and verified running

2. GitHub Project Setup

A new GitHub repository named **Real-Projects** was created to maintain all Kubernetes and automation configuration files. The repository was connected to the local environment using SSH for secure authentication.

- ■ Local Project Path: `~/Real-Project`
- ■ SSH key configured and verified with `ssh -T git@github.com`
- ■ Push access confirmed using secure SSH-based Git connection
- ■ GitHub Repository URL: https://github.com/nilmadhu/Real-Projects

3. Folder Structure

The following folder hierarchy was created to organize the project into logical modules for ArgoCD, Grafana, and Prometheus monitoring setup.

``` Real-Project/ ■■■ Argocd ■ ■■■ argocd/ ■ ■ ■■■ nginx-app.yaml ■ ■■■ deployment.yaml ■ ■■■ service.yaml ■■■ monitoring ■■■ argocd/ ■■■ grafana/ ■■■ monitoring/ ■■■ prometheus/ ```

4. Git Configuration and Commands Used

Below are the essential Git commands executed to initialize and push the project to GitHub:

``` cd ~/Real-Project git init git add . git commit -m "Initial commit - ArgoCD and Monitoring project setup" git branch -M main git remote add origin git@github.com:nilmadhu/Real-Projects.git git push -u origin main ```

SSH-based authentication was confirmed, ensuring no password prompts during `git push` operations.

5. ArgoCD Installation Verification

ArgoCD was deployed successfully and validated with the following command outputs:

``` kubectl get ns kubectl get pods -n argocd ``` All ArgoCD components (`argocd-server`, `argocd-repo-server`, `argocd-application-controller`, etc.) were in **Running** state.

6. Next Steps (Phase 2 Preview)

The next phase of the project will focus on automating the monitoring stack with Prometheus and Grafana, integrating them into ArgoCD for GitOps-based continuous deployment and observability.

- ■■ Deploy Prometheus Operator and configure service monitors
- ■ Automate Grafana dashboards deployment via Helm chart
- ■ Integrate Prometheus & Grafana manifests into GitOps workflow (ArgoCD)
- ■ Validate end-to-end automation with live metrics

7. Summary

This phase established the foundational structure of the **Kubernetes + ArgoCD + Grafana Monitoring Automation Project**, including GitHub integration, ArgoCD validation, and repository setup for subsequent automation. The environment is now fully prepared to proceed with observability and continuous monitoring integration.