

ArgoCD Implementation

The screenshot shows the ArgoCD web interface at <https://4.250.240.135/applications?showFavorites=false&proj=&sync=&autoSync=&health=&namespace=&cluster=&labels=>. The left sidebar includes links for Applications, Settings, User Info, Documentation, Application filters, SYNC STATUS, and HEALTH STATUS. The main content area displays the 'Applications' section with a card for 'lanik-app'. The card details the following information:

Project:	lanik-project
Labels:	
Status:	Healthy Synced
Reposit...	https://github.com/Oreire/aks-web-aut...
Target R...	main
Path:	AKS
Destinat...	in-cluster
NAMESP...	default
Created ...	10/11/2025 22:40:00 (22 minutes ago)
Last Sy...	10/11/2025 22:40:03 (22 minutes ago)

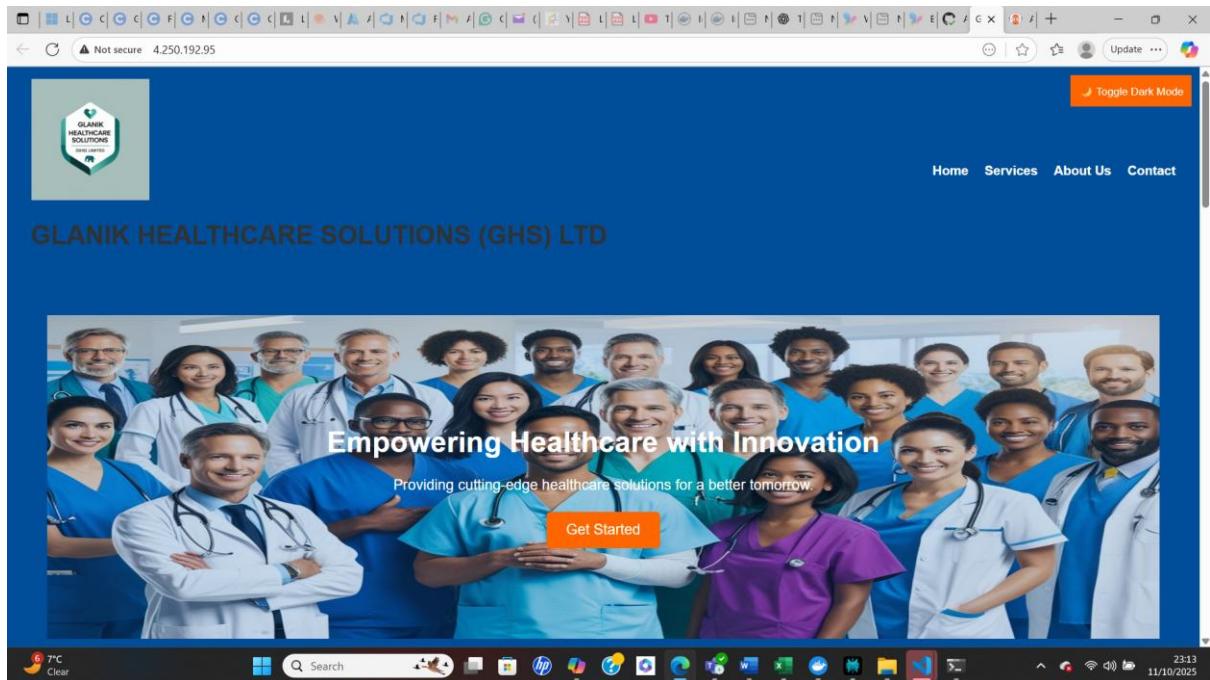
At the bottom of the card are three buttons: SYNC, REFRESH, and DELETE.

The screenshot shows the ArgoCD web interface at <https://4.250.240.135/applications?showFavorites=false&proj=&sync=&autoSync=&health=&namespace=&cluster=&labels=>. The left sidebar is identical to the first screenshot. The main content area displays the 'Applications' section with cards for 'lanik-app' and 'monitoring-stack'. The 'lanik-app' card is identical to the one in the first screenshot. The 'monitoring-stack' card shows the following information:

Project:	lanik-project
Labels:	
Status:	Unknown Synced
Reposit...	https://github.com/Oreire/aks-web-aut...
Target R...	main
Path:	MONITORING
Destinat...	in-cluster
NAMESP...	monitoring
Created ...	10/12/2025 00:16:56 (3 minutes ago)

At the bottom of each card are three buttons: SYNC, REFRESH, and DELETE.

Web Application



ArgoCD Resources

```

aayai@GLANIK MINGW64 /c/AZURE/aks-web-automation/ArgoCD (main)
● $ kubectl get pods -n argocd
NAME                               READY   STATUS    RESTARTS   AGE
argocd-application-controller-0     1/1     Running   0          36m
argocd-application-set-controller   1/1     Running   0          36m
argocd-dex-server-95477dd-r45r8    1/1     Running   0          36m
argocd-notifications-controller   1/1     Running   0          36m
argocd-redis-5746cc45fb-h28fg     1/1     Running   0          36m
argocd-repo-server-588cfaf4ab-4mk6 1/1     Running   0          36m
argocd-server-656b9b6c6c-jfjh      1/1     Running   0          36m

aayai@GLANIK MINGW64 /c/AZURE/aks-web-automation/ArgoCD (main)
● $ kubectl get svc -n argocd
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
argocd-applicationset-controller   ClusterIP  10.0.56.170 <none>        7000/TCP,8080/TCP   36m
argocd-dex-server                 ClusterIP  10.0.255.146 <none>        5556/TCP,5557/TCP,5558/TCP   36m
argocd-metrics                   ClusterIP  10.0.168.18  <none>        8082/TCP          36m
argocd-notifications-controller-metrics ClusterIP  10.0.45.123 <none>        9001/TCP          36m
argocd-redis                      ClusterIP  10.0.156.26  <none>        6379/TCP          36m
argocd-repo-server                ClusterIP  10.0.96.60   <none>        8081/TCP,8084/TCP   36m
argocd-server                     LoadBalancer 10.0.90.177  4.250.240.135  80:32222/TCP,443:32244/TCP   36m
argocd-server-metrics              ClusterIP  10.0.8.183  <none>        8083/TCP          36m

aayai@GLANIK MINGW64 /c/AZURE/aks-web-automation/ArgoCD (main)
● $ kubectl get deploy -n argocd
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
argocd-applicationset-controller   1/1     1          1          37m
argocd-dex-server                 1/1     1          1          37m
argocd-notifications-controller   1/1     1          1          37m
argocd-redis                      1/1     1          1          37m
argocd-repo-server                1/1     1          1          37m
argocd-server                     1/1     1          1          37m

```

Azure AKS Cluster and HPA

Azure Console Resources

https://portal.azure.com/#view/HubsExtension/BrowseAll.ReactView/resourceType/microsoft.resources%2Resources

Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > All resources

Default Directory

+ Create Manage view Refresh Export to CSV Open query Assign tags Delete Group by none

You are viewing a new version of Browse experience. Click here to access the old experience.

Filter for any field... Subscription equals all Resource Group equals all Type equals all Location equals all + Add filter

Name	Type	Resource Group	Location	Subscription
322b5ca3-cd2b-402f-afb2-a8a95360067e	Public IP address	MC_aks-resource-group_aks-cluster_uksouth	UK South	Azure subscription 1
aks-agentpool-25056853-nsg	Network security group	mc_aks-resource-group_aks-cluster_uksouth	UK South	Azure subscription 1
aks-cluster	Kubernetes service	aks-resource-group	UK South	Azure subscription 1
aks-cluster-agentpool	Managed Identity	MC_aks-resource-group_aks-cluster_uksouth	UK South	Azure subscription 1
aks-default-17884513-vmss	Virtual machine scale set	MC_aks-resource-group_aks-cluster_uksouth	UK South	Azure subscription 1
aks-vnet-25056853	Virtual network	MC_aks-resource-group_aks-cluster_uksouth	UK South	Azure subscription 1
k8scloudflstateuks	Storage account	aks-resource-group	UK South	Azure subscription 1
kubernetes	Load balancer	mc_aks-resource-group_aks-cluster_uksouth	UK South	Azure subscription 1
kubernetes-a86fbfd7e67ec64148a159e58aa7e04	Public IP address	mc_aks-resource-group_aks-cluster_uksouth	UK South	Azure subscription 1
kubernetes-a8afe9c9cc24d4708b684c55ea95158	Public IP address	mc_aks-resource-group_aks-cluster_uksouth	UK South	Azure subscription 1

Showing 1 - 10 of 11. Display count: auto < 1 2 > Give feedback

Monitoring Stack

```

ajayi@GLANIK MINGW64 /c/AZURE/aks-web-automation/MONITORING (main)
$ kubectl get pods -n monitoring
NAME                               READY   STATUS    RESTARTS   AGE
alertmanager-kube-prometheus-alertmanager-0   2/2     Running   0          7m5s
kube-prom-grafana-84d7d5987-hhk6d            3/3     Running   0          7m10s
kube-prom-kube-prometheus-operator-6689565ff-j2z4d 1/1     Running   0          7m10s
kube-prom-kube-state-metrics-5cbfc6bdb-7js4q    1/1     Running   0          7m10s
kube-prom-prometheus-node-exporter-bgjmw       1/1     Running   0          7m10s
kube-prom-prometheus-node-exporter-wrfv2       1/1     Running   0          7m10s
prometheus-kube-prometheus-prometheus-0      2/2     Running   0          7m5s

ajayi@GLANIK MINGW64 /c/AZURE/aks-web-automation/MONITORING (main)
$ kubectl get deploy -n monitoring
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
kube-prom-grafana 1/1         1           1           7m22s
kube-prom-kube-prometheus-operator 1/1         1           1           7m22s
kube-prom-kube-state-metrics 1/1         1           1           7m22s

ajayi@GLANIK MINGW64 /c/AZURE/aks-web-automation/MONITORING (main)
$ kubectl get svc -n monitoring
NAME              TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
alertmanager-operated   ClusterIP  <none>        <none>        9093/TCP,9094/TCP,9094/UDP   7m26s
kube-prom-grafana     ClusterIP  10.0.121.59  <none>        80/TCP          7m31s
kube-prom-kube-prometheus-alertmanager  ClusterIP  10.0.33.182 <none>        9093/TCP,8088/TCP   7m31s
kube-prom-kube-prometheus-operator    ClusterIP  10.0.22.200 <none>        443/TCP          7m31s
kube-prom-kube-prometheus-prometheus ClusterIP  10.0.73.168 <none>        9090/TCP,8080/TCP   7m31s
kube-prom-kube-state-metrics       ClusterIP  10.0.143.4  <none>        8080/TCP          7m31s
kube-prom-prometheus-node-exporter ClusterIP  10.0.90.92  <none>        9100/TCP          7m31s
prometheus-operated          ClusterIP  None          <none>        9090/TCP          7m26s

ajayi@GLANIK MINGW64 /c/AZURE/aks-web-automation/MONITORING (main)
$ 

```

A screenshot of a terminal window titled "bash - MONITORING". It shows the output of several Kubernetes commands: `kubectl get pods -n monitoring`, `kubectl get deploy -n monitoring`, and `kubectl get svc -n monitoring`. The terminal is part of a larger interface with tabs for "OUTPUT", "CODE REFERENCE LOG", "PROBLEMS 26", "DEBUG CONSOLE", "TERMINAL", "PORTS", "AZURE", and "SONARQUBE 1". The status bar at the bottom shows various system icons and the date/time.

Grafana Dashboard

The screenshot shows the Grafana interface with a dark theme. On the left, there's a sidebar with navigation links: Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Connections, Data sources (which is selected), and Administration. The main area displays a configuration page for a Prometheus data source. It includes sections for Performance (Prometheus type: Choose, Cache level: Low, Incremental querying (beta): off, Disable recording rules (beta): off), Other (Custom query parameters: Example: max_source_resolution=5m&in, HTTP method: POST, Series limit: 40000, Use series endpoint: off), and Exemplars (+ Add). A success message at the bottom states: "Successfully queried the Prometheus API. Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view." Buttons for "Delete" and "Save & test" are at the bottom. The status bar at the bottom shows system icons and the date/time.

Next Steps

DevSecOps Integration Strategy within the Workflow

Different Layers would be considered including:

1 Code & Dependency Security (before build)

Objective: Detect vulnerabilities, secrets, or policy violations before provisioning or deployment.

- ◆ **Static Application Security Testing (SAST)**
- ◆ **Secret Scanning / Credential Leakage**

Check for accidentally committed secrets.

2 Infrastructure-as-Code (Terraform) Security

Objective: Prevent insecure infrastructure definitions before they reach Azure.

- ◆ **Static IaC Scan (Terraform)**

Use **Checkov** or **tfsec** to enforce cloud security best practices.

This checks for: Publicly exposed AKS or storage accounts, missing encryption, insecure network rules & lack of RBAC / logging

3 Container Image Security

Objective: Scan your Docker image for OS/package vulnerabilities *before deployment*.

- ◆ **Container Vulnerability Scan**

Integrate **Aqua Trivy** (free and popular) **before pushing** or **before deploying to AKS**.

4 Dependency & Supply-Chain Security

Objective: Secure your build environment and dependencies.

- Enable **Dependabot** for automatic dependency updates:
- Usage of **signed commits** and enable **branch protection rules** (required reviews, CI checks, etc.)

5 Runtime & Kubernetes Security

Objective: Validate your AKS manifests and cluster configuration.

- ◆ **Kubernetes Manifest Scanning**

Use **kube-linter** or **kubescape** before kubectl apply to flag containers running as root, missing resource limits, unencrypted secrets & insecure host paths

- ◆ **Post-Deployment Checks (Optional)**

After deployment, integration of **kubescape** or **OPA Gatekeeper** for runtime policy compliance and **Azure Defender for Kubernetes** (native AKS integration)

6 Compliance & Reporting

Generate a summary or artifact of all security checks.

Optionally, integrate results into **GitHub Security Dashboard** (with CodeQL + Dependabot).

Putting It All Together: Enhanced Workflow Structure

You'd end up with a 4-stage pipeline:

Stage	Job	Key Tools
 Pre-Provision Security	security-scan	CodeQL, Semgrep, TruffleHog
 Infra Provisioning	provision	Terraform + Checkov
 Container Build & Scan	image-scan	Docker + Trivy
 Deploy to AKS	deploy	Kubectl + Kube-linter

Each stage can use needs: to enforce order:

security-scan → provision → image-scan → deploy

Bonus DevSecOps Enhancements

Category	Best Practice
Secrets Management	Use GitHub Actions Secrets or Azure Key Vault (not plaintext in workflow).
Policy as Code	Integrate OPA / Conftest to enforce compliance rules.
Auditing	Enable workflow artifact retention and audit logs in GitHub.
Least Privilege	Use minimal RBAC for your Azure SPN credentials (no Owner role).
Encryption	Ensure state files (Terraform backend) and images are encrypted at rest.