BULUT BİLİŞİM ve DEVOPS TEKNOLOJİLERİ:

Proje Raporu

DevOps, yazılım geliştirme ve işletme süreçlerinin uyumlu bir şekilde entegre edilmesini amaçlayan bir kültür ve metodoloji olarak öne çıkıyor. Bu rapor, DevOps'un temel prensiplerini vurgulamakta ve bu prensiplerin bir projede nasıl hayata geçirildiğine odaklanmaktadır. Projem, çeşitli süreçleri birleştirip otomatize etmeyi hedeflemektedir.

Proje kapsamında, geniş bir topluluk ve kaynak desteği sağlayan popüler araçlar tercih edilmiştir. Bu araçlar, projenin DevOps sürecine çeşitli avantajlar katmaktadır. Örneğin, popüler araç seçimi, proje ekibinin geniş bir destek ağına erişimini sağlamaktadır. Ayrıca, belirli adımlarda kullanılan araçlar, süreçleri otomatize ederek takım içi işbirliğini artırmaktadır. Bu da projenin verimliliğini ve güvenilirliğini önemli ölçüde iyileştirebilir.

Kendi projemde ilerleme sürecim aşağıda yer alıyor;

IDE(Visual Studio Code) - Git Repo(GitHub) - Code Scan(Snyk) - Image – (Docker File) - Image Registery(DockerHub) - Image Scan(Snyk) - Kubernetes Cluster - CI / CD(ArgoCD) - Monitoring(Prometheus, Grafana)

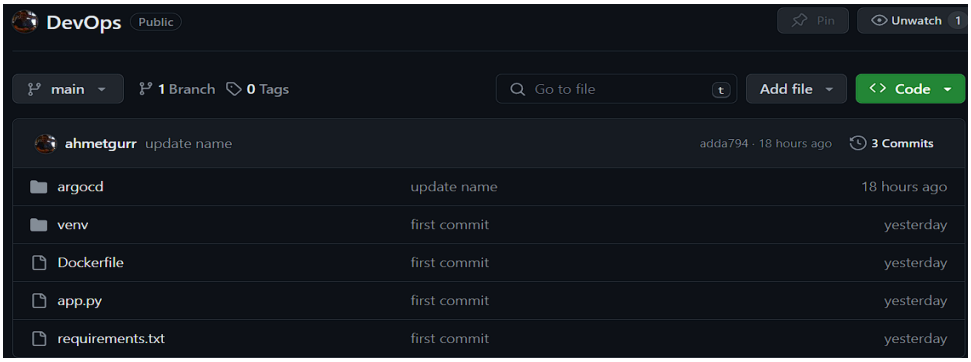
Bu entegrasyonun avantajları şunlar olabilir;

- Kolay izleme ve raporlama imkanı.
- Otomatik ve sürekli dağıtım.
- Hızlı yazılım dağıtımı.
- Tekrarlanabilir ve tutarlı dağıtım süreçleri.
- Modüler yapı sayesinde değişikliklerin sadece ilgili kısımlarda etkili olması.

Bu adımlarla birlikte DevOps sürecini oluşturduğumuzda daha verimli bir yazılım geliştirme ve dağıtım süreci elde edebiliriz.

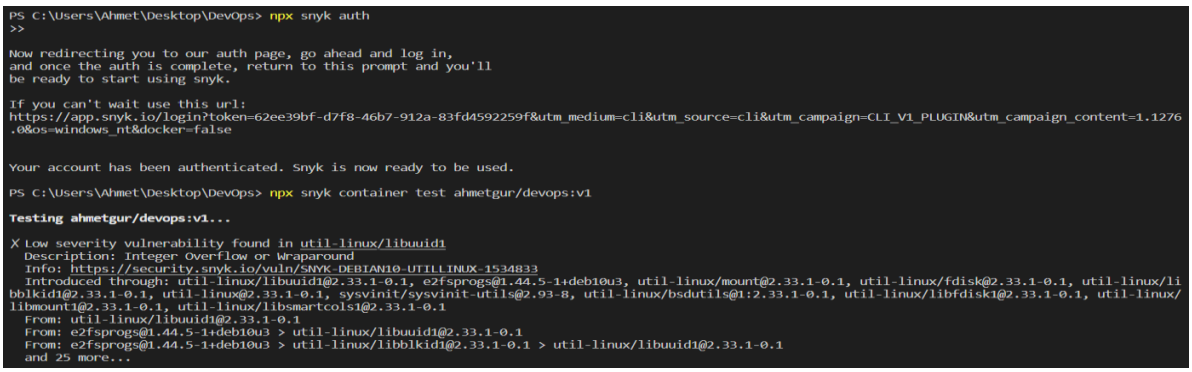
IDE Üzerinde Uygulama Yapma

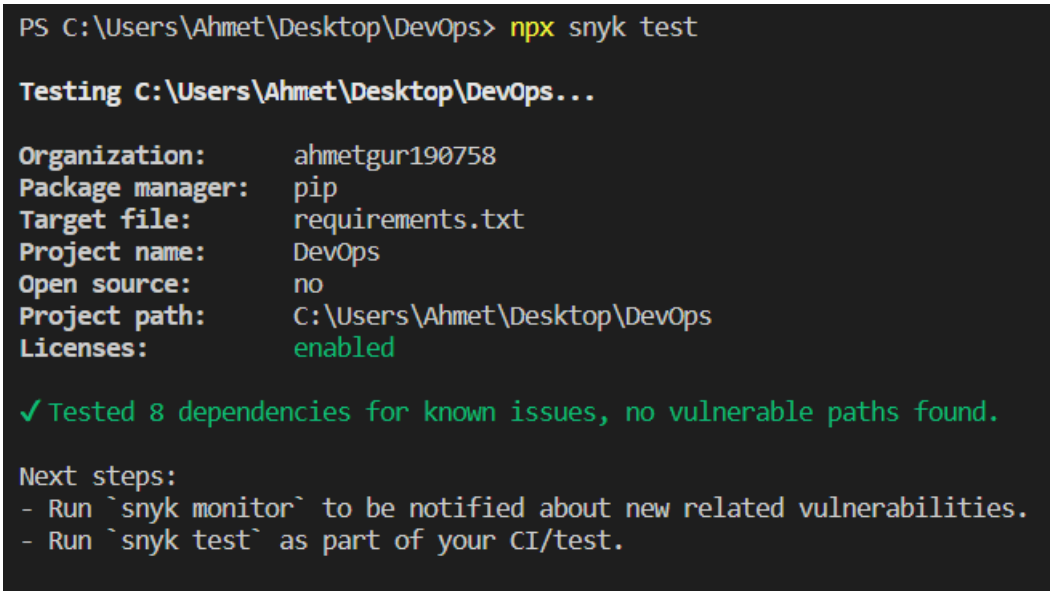
Visual Studio Code üzerinde kendim Container-based çalışan bir web sitesi yaptım. Uygulama içerisinde gerekli değişiklikleri hazır hale getirdikten sonra kodu GitHub repoma gönderdim.



Code Scan

Code Scan ürünü seçimi için Snyk kullanıyorum. Ardından Code seviyesinde (code scan ürünü ile) bir güvenlik taraması yapıyoruz. Bunun için synk ile entegre etmemiz gerekiyor. Daha sonra npx snyk auth ile terminalden giriş yapıp entegre ediyorum. Son olarak Snyk sitesinden analizini görüntüleyebiliyoruz.





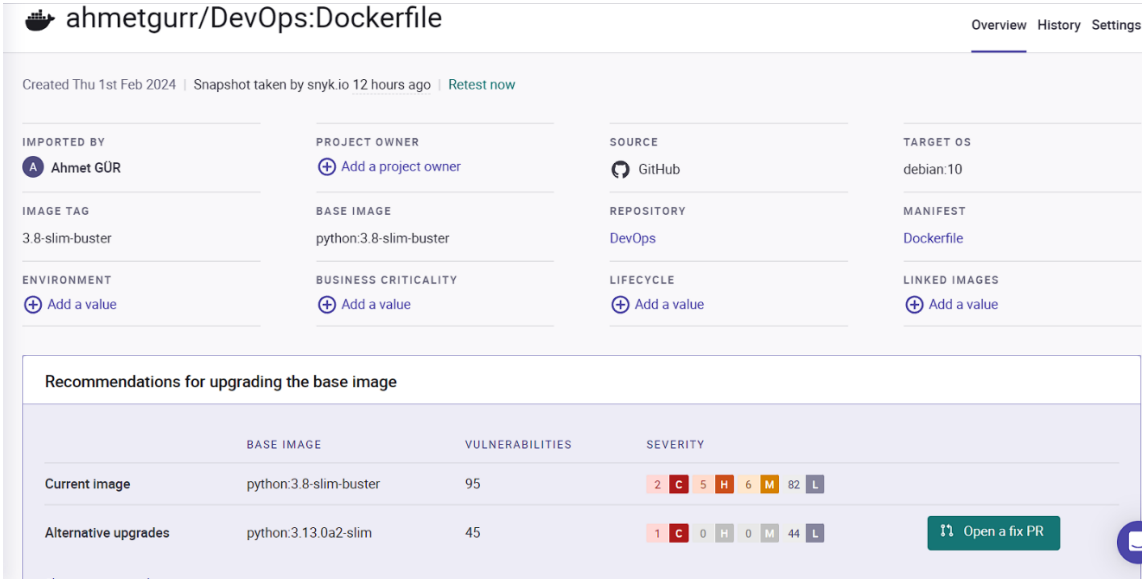


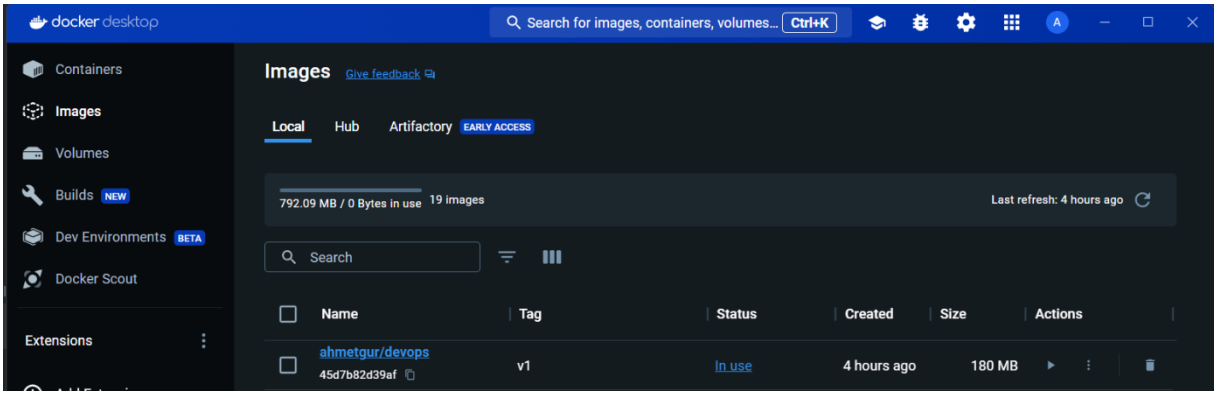
Image Oluşturma ve Docker Registery'ye Ekleme

Image oluşturma aracı olarak Dockerfile (manuel olarak) seçiyoruz. Dockerfile ile bir container image oluşturuyoruz. Image registery olarak Docker Hub'u kullanıyoruz ve oluşturulan image'yi Docker Hub'a yüklüyoruz.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Ahmet\Desktop\DevOps> docker build -t ahmetgur/devops:v1 .
[+] Building 20.6s (10/10) FINISHED                                                    docker:default
 => [internal] load .dockerignore                                                              0.1s
 => => transferring context: 2B                                                                0.0s
 => [internal] load build definition from Dockerfile                                           0.1s
 => => transferring dockerfile: 410B                                                           0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim-buster                      2.8s
 => [auth] library/python:pull token for registry-1.docker.io                                  0.0s
 => [1/4] FROM docker.io/library/python:3.8-slim-buster@sha256:8799b0564103a9f36cfb8a8e1c562e  8.1s
 => => resolve docker.io/library/python:3.8-slim-buster@sha256:8799b0564103a9f36cfb8a8e1c562e  0.0s
 => => sha256:8799b0564103a9f36cfb8a8e1c562e11a9a6f2e3bb214e2adc23982b36a04511 988B / 988B     0.0s
 => => sha256:90834dba6381dfc3957573dc7a3e6c5c8ed255cf60079329a6da2b5e6d4257b8 1.37kB / 1.37kB 0.0s
 => => sha256:addd6962740ab9fd79a788945daa24348c11adcec97d47a647e0a61c86cc9f60 6.87kB / 6.87kB 0.0s
 => => sha256:8b91b88d557765cd8c6802668755a3f6dc4337b6ce15a17e4857139e5fc964f3 27.14MB / 27.14MB 4.8s
 => => sha256:824416e234237961c9c5d4f41dfe5b295a3c35a671ee52889bfb08d8e257ec4c 2.78MB / 2.78MB 0.8s
 => => sha256:8f777578c172d018077d3dc22d6654911fff60066097943fe8c4697ecf8aac35 12.89MB / 12.89MB 2.4s
 => => sha256:cbfea27109a8b1136059a7973ccb8243889faf162ebc173a05909dcb0bec03c9 244B / 244B     1.2s
 => => sha256:276dfcf5defffff3c5d540a8e0d9a18656a4c03637a8b4f4eec1f4a147799c901 3.14MB / 3.14MB 2.4s
 => => extracting sha256:8b91b88d557765cd8c6802668755a3f6dc4337b6ce15a17e4857139e5fc964f3      1.8s
 => => extracting sha256:824416e234237961c9c5d4f41dfe5b295a3c35a671ee52889bfb08d8e257ec4c      0.2s
 => => extracting sha256:8f777578c172d018077d3dc22d6654911fff60066097943fe8c4697ecf8aac35      0.6s
 => => extracting sha256:cbfea27109a8b1136059a7973ccb8243889faf162ebc173a05909dcb0bec03c9      0.0s
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt                                   4.1s
 => exporting to image                                                                         0.6s
 => => exporting layers                                                                        0.6s
 => => writing image sha256:45d7b82d39af3ea5e73463679f699f28b04af843cab3ed9529e6e70c861288b8   0.0s
 => => naming to docker.io/ahmetgur/devops:v1                                                  0.0s
```

```
What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\Ahmet\Desktop\DevOps> docker login -u ahmetgur -p Acbirkokakolaic.01
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded
PS C:\Users\Ahmet\Desktop\DevOps> docker push ahmetgurr/devops:v1
The push refers to repository [docker.io/ahmetgurr/devops]
An image does not exist locally with the tag: ahmetgurr/devops
PS C:\Users\Ahmet\Desktop\DevOps> docker push ahmetgur/devops:v1
The push refers to repository [docker.io/ahmetgur/devops]
ad00604425ff: Pushed
e6ece6d329c7: Pushed
f7f6833d6276: Pushed
e6c5004ee77f: Mounted from library/python
997b8e79e84f: Mounted from library/python
3054512b6f71: Mounted from library/python
ae2d55769c5e: Mounted from library/python
e2ef8a51359d: Mounted from library/python
v1: digest: sha256:d5cabf3afeebd166baca7aebb81225819c10650987097cc1a80f1b80d2712fb3 size: 1999
```

```
C:\Windows\System32>kubectl get nodes
NAME             STATUS   ROLES           AGE    VERSION
docker-desktop   Ready    control-plane   3m1s   v1.28.2
```



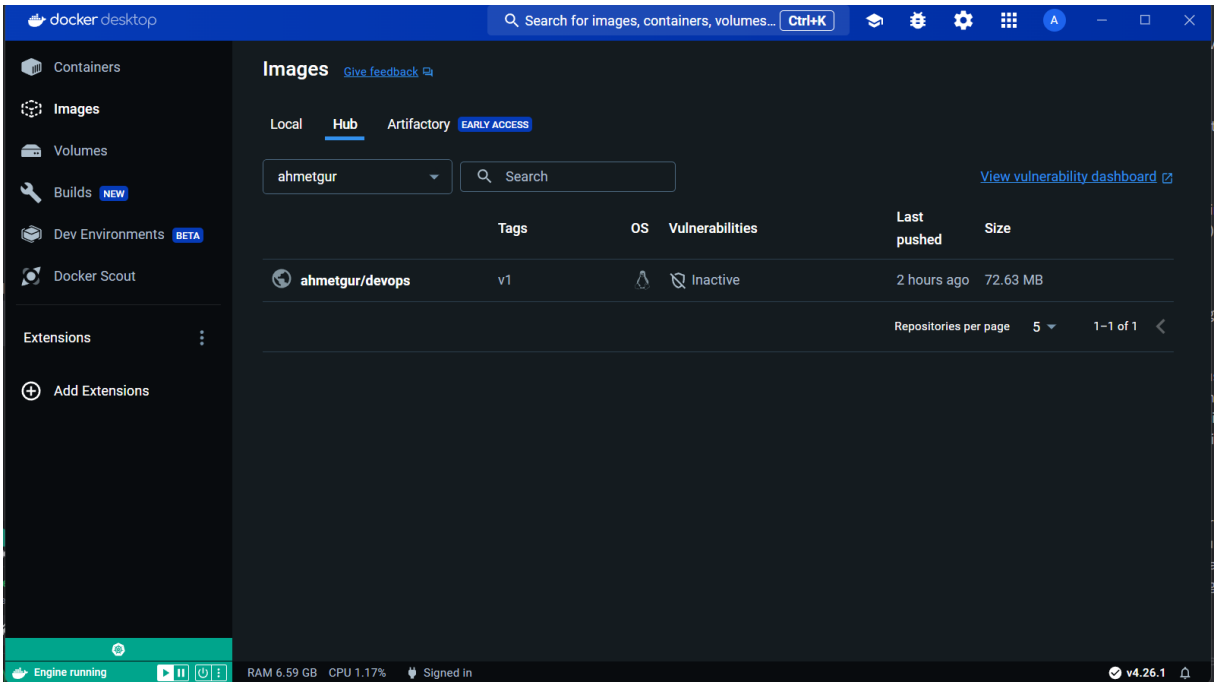


Image Güvenlik Taraması

Image Scan ürünü olarak Snyk seçiyoruz. Seçmiş olduğumuz Synk ile Image'mızı tarayarak güvenlik açıklarını kontrol ediyoruz.

```
Organization:      ahmetgur190758
Package manager:   deb
Project name:      docker-image|ahmetgur/devops
Docker image:      ahmetgur/devops:v1
Platform:          linux/amd64
Base image:        python:3.8.17-slim-buster
Licenses:          enabled

Tested 93 dependencies for known issues, found 95 issues.

Base Image               Vulnerabilities  Severity
python:3.8.17-slim-buster  95             2 critical, 5 high, 6 medium, 82 low

Recommendations for base image upgrade:

Alternative image types
Base Image                  Vulnerabilities  Severity
python:3.13.0a3-slim        45               1 critical, 0 high, 0 medium, 44 low
python:3.13.0a2-slim-bullseye 68             1 critical, 0 high, 0 medium, 67 low
python:3.11.7-bookworm      183              1 critical, 1 high, 2 medium, 179 low
python:3.11.5-slim-bookworm 59               1 critical, 4 high, 7 medium, 47 low


Learn more: https://docs.snyk.io/products/snyk-container/getting-around-the-snyk-container-ui/base-image-detection
```

Kubernetes Cluster'a Dağıtma

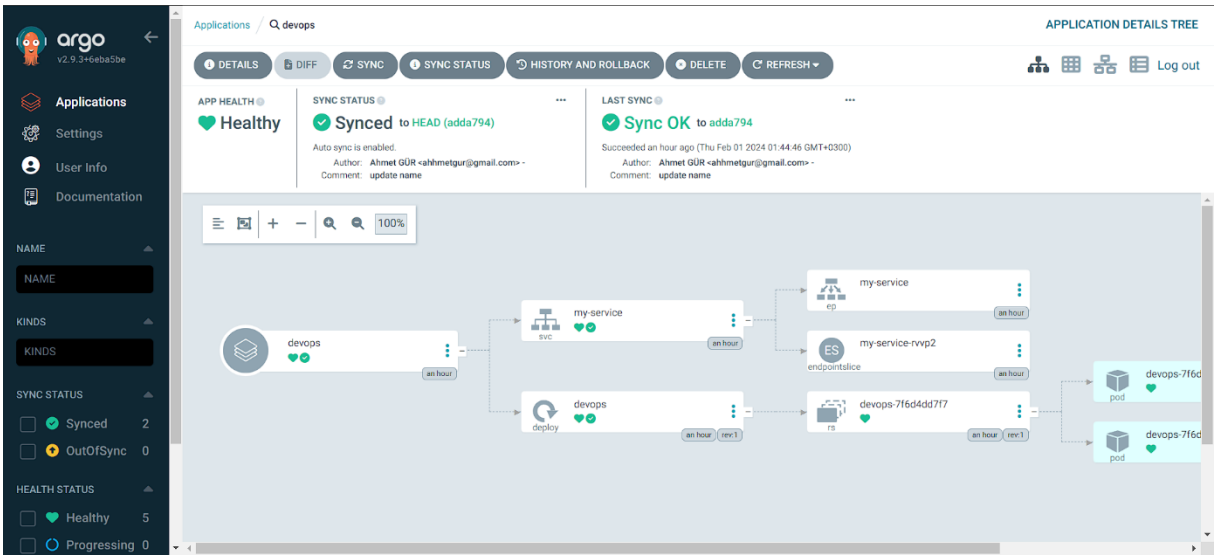
Kubernetes için CI/CD araçlarından ArgoCD kullanılarak Kubernetes Cluster'a dağıtma işlemini otomatize ettik. Daha sonra kubernates üzerine deployment objesini oluşturuyoruz. ArgoCD server kontrollerini gerçekleştiriyoruz. Aldığımız kullanıcı adı ve şifreyle Argoya giriş yapıyoruz daha sonra buradan ilgili GitHub proje linkini ekleyerek projeyi oluşturuyoruz ve proje analizini gerçekleştiryoruz.

```
PS C:\Users\Ahmet\Desktop\DevOps> kubectl apply -f argocd/deployment.yaml
deployment.apps/devops created
PS C:\Users\Ahmet\Desktop\DevOps> docker login
>>
Authenticating with existing credentials...
Login Succeeded
PS C:\Users\Ahmet\Desktop\DevOps> docker images ahmetgurr/devops
>>
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
PS C:\Users\Ahmet\Desktop\DevOps>
```

```
C:\Windows\System32>kubectl create ns argocd
namespace/argocd created

C:\Windows\System32>kubectl get ns
NAME              STATUS   AGE
argocd            Active   7s
default           Active   5m57s
kube-node-lease   Active   5m57s
kube-public       Active   5m57s
kube-system       Active   5m57s
```

```
C:\Windows\System32>kubectl edit svc -n argocd argocd-server
Edit cancelled, no changes made.

C:\Windows\System32>kubectl get po -n argocd
NAME                                                READY   STATUS    RESTARTS   AGE
argocd-application-controller-0                     1/1     Running   0          2m2s
argocd-applicationset-controller-78457b866d-szlvz   1/1     Running   0          2m2s
argocd-dex-server-579d4d7645-rrspg                  1/1     Running   0          2m2s
argocd-notifications-controller-d7559958c-h9bj2     1/1     Running   0          2m2s
argocd-redis-5f6b9df9-dg56h                         1/1     Running   0          2m2s
argocd-repo-server-7d96cf9b5f-zs5hj                 1/1     Running   0          2m2s
argocd-server-6f49c4bd54-xmn8b                      1/1     Running   0          2m2s

C:\Windows\System32>kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
Handling connection for 8080
Handling connection for 8080
```

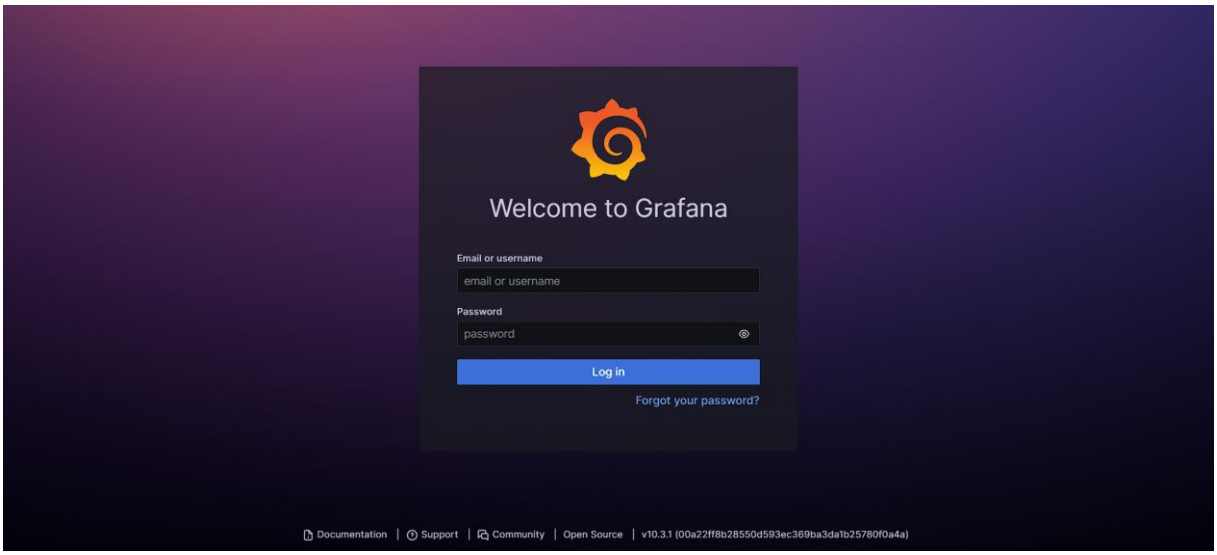
```
C:\Windows\System32>kubectl get all -n argocd
NAME                                                READY   STATUS              RESTARTS   AGE
pod/argocd-application-controller-0                 0/1     ContainerCreating   0          7s
pod/argocd-applicationset-controller-78457b866d-szlvz   0/1     ContainerCreating   0          7s
pod/argocd-dex-server-579d4d7645-rrspg              0/1     Init:0/1            0          7s
pod/argocd-notifications-controller-d7559958c-h9bj2     0/1     ContainerCreating   0          7s
pod/argocd-redis-5f6b9df9-dg56h                     0/1     ContainerCreating   0          7s
pod/argocd-repo-server-7d96cf9b5f-zs5hj             0/1     Init:0/1            0          7s
pod/argocd-server-6f49c4bd54-xmn8b                  0/1     ContainerCreating   0          7s

NAME                                               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
service/argocd-applicationset-controller           ClusterIP   10.102.98.20     <none>        7000/TCP,8080/TCP            7s
service/argocd-dex-server                          ClusterIP   10.103.191.215   <none>        5556/TCP,5557/TCP,5558/TCP   7s
service/argocd-metrics                             ClusterIP   10.106.183.51    <none>        8082/TCP                     7s
service/argocd-notifications-controller-metrics    ClusterIP   10.108.208.69    <none>        9001/TCP                     7s
service/argocd-redis                               ClusterIP   10.98.182.24     <none>        6379/TCP                     7s
service/argocd-repo-server                         ClusterIP   10.109.160.211   <none>        8081/TCP,8084/TCP            7s
service/argocd-server                              ClusterIP   10.109.46.48     <none>        80/TCP,443/TCP               7s
service/argocd-server-metrics                      ClusterIP   10.110.27.53     <none>        8083/TCP                     7s

NAME                                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/argocd-applicationset-controller   0/1     1            0           7s
deployment.apps/argocd-dex-server                  0/1     1            0           7s
deployment.apps/argocd-notifications-controller    0/1     1            0           7s
deployment.apps/argocd-redis                       0/1     1            0           7s
deployment.apps/argocd-repo-server                 0/1     1            0           7s
deployment.apps/argocd-server                      0/1     1            0           7s

NAME                                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/argocd-applicationset-controller-78457b866d   1         1         0       7s
replicaset.apps/argocd-dex-server-579d4d7645               1         1         0       7s
replicaset.apps/argocd-notifications-controller-d7559958c  1         1         0       7s
replicaset.apps/argocd-redis-5f6b9df9                      1         1         0       7s
replicaset.apps/argocd-repo-server-7d96cf9b5f              1         1         0       7s
replicaset.apps/argocd-server-6f49c4bd54                   1         1         0       7s

NAME                                          READY   AGE
statefulset.apps/argocd-application-controller   0/1     7s

C:\Windows\System32>kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "NodePort"}}'
service/argocd-server patched (no change)
Error from server (NotFound): services "{type:" not found
Error from server (NotFound): services "NodePort}}'" not found
```



Monitoring

Monitoring araçları seçimi olarak Prometheus ve Grafana seçildi. Prometheus ve Grafana kurulumları gerçekleştirildi ve otomatik verilen şifre yeni şifreyle değiştirildi ve 5000 portuna atandı.

```
C:\Windows\System32>kubectl get service -n monitoring
NAME                                  TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
grafana                               LoadBalancer   10.100.92.199    <pending>     80:32316/TCP   6h9m
prometheus-alertmanager               ClusterIP      10.110.175.206   <none>        9093/TCP       6h10m
prometheus-alertmanager-headless      ClusterIP      None             <none>        9093/TCP       6h10m
prometheus-kube-state-metrics         ClusterIP      10.97.15.19      <none>        8080/TCP       6h10m
prometheus-prometheus-node-exporter   ClusterIP      10.101.57.66     <none>        9100/TCP       6h10m
prometheus-prometheus-pushgateway     ClusterIP      10.107.41.187    <none>        9091/TCP       6h10m
prometheus-server                     ClusterIP      10.101.44.88     <none>        80/TCP         6h10m

C:\Windows\System32>
C:\Windows\System32>kubectl port-forward service/grafana 5000:80 -n monitoring
Forwarding from 127.0.0.1:5000 -> 3000
Forwarding from [::1]:5000 -> 3000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
```



Ahmet GÜR

02200201053

Örgün Öğretim