



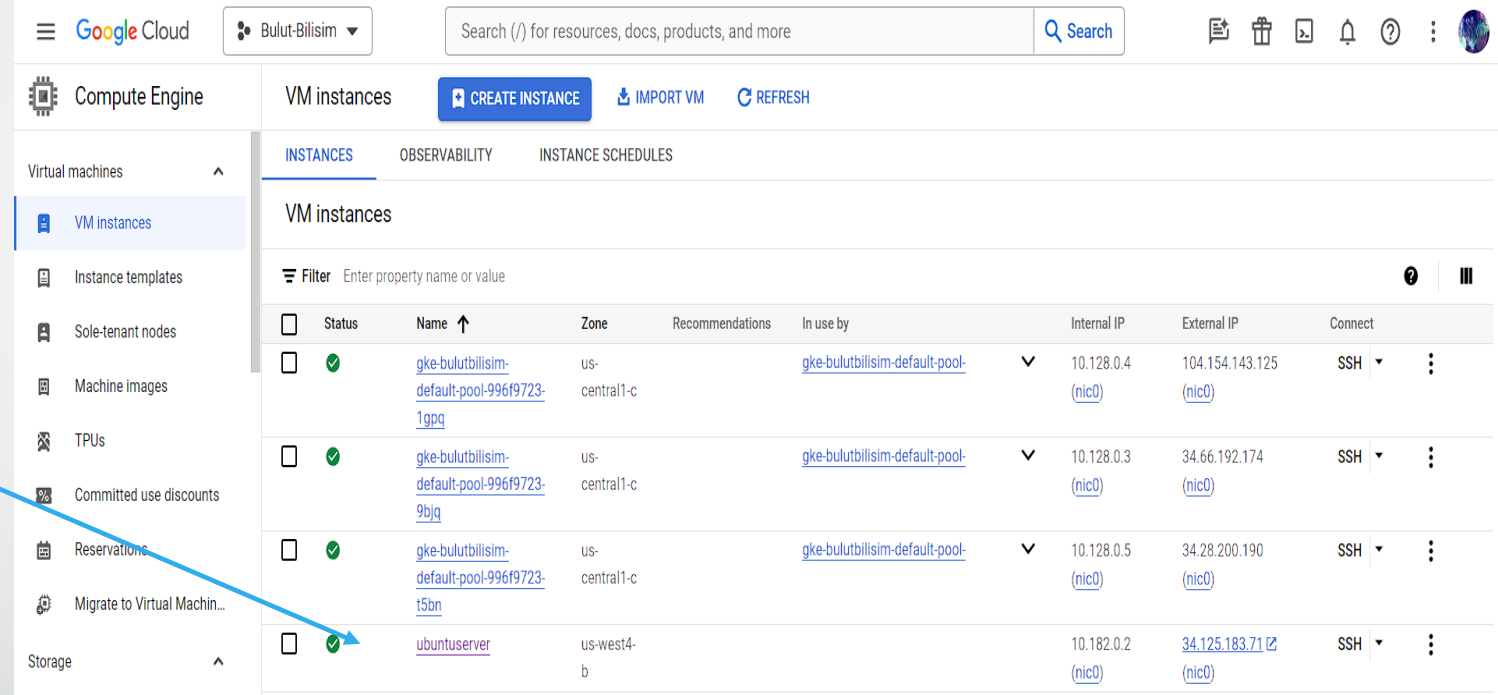
BULUT BİLİŞİM VE DEVOPS TEKNOLOJİLERİ

Muhammed Enes Sarıca

0220201050

Başlarken;

Google Cloud
Console üzerinde
oluşturduğumuz
sanal makineye
ubuntu kuruyoruz.



Google Cloud Bulut-Bilisim

Search (/) for resources, docs, products, and more

Compute Engine VM instances

CREATE INSTANCE IMPORT VM REFRESH

INSTANCES OBSERVABILITY INSTANCE SCHEDULES

VM instances

Filter Enter property name or value

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	gke-bulutbilisim-default-pool-996f9723-1gpq	us-central1-c		gke-bulutbilisim-default-pool-	10.128.0.4 (nic0)	104.154.143.125 (nic0)	SSH
✓	gke-bulutbilisim-default-pool-996f9723-9bjq	us-central1-c		gke-bulutbilisim-default-pool-	10.128.0.3 (nic0)	34.66.192.174 (nic0)	SSH
✓	gke-bulutbilisim-default-pool-996f9723-t5bn	us-central1-c		gke-bulutbilisim-default-pool-	10.128.0.5 (nic0)	34.28.200.190 (nic0)	SSH
✓	ubuntuserver	us-west4-b			10.182.0.2 (nic0)	34.125.183.71 (nic0)	SSH



Ubuntu Üzerinde Jenkins Kurulumu

- Java Development Kit (JDK) Kurulumunu yapıyoruz.
- Jenkins'in resmi Debian/Ubuntu deposunu ekleyerek Jenkins'i kurmamız gerekiyor. İlk olarak, Jenkins'in depo anahtarını sisteme ekleriz.
- Jenkins depo dosyasını ekleriz.
- Depo dosyasını ekledikten sonra, paket listesini güncelleyip ve Jenkins'i kurarız.
- Kurulum tamamlandıktan sonra, Jenkins servisini başlatıp sistem başlangıcında otomatik olarak başlaması için yapılandırırız.
- Tarayıcıyı açarız ve <http://localhost:8080/> adresine gideriz.
- İlk adımda, initialAdminPassword dosyasından alınan şifreyi yapıştırırız.
- Gerekli eklentileri seçerek kurulumu tamamlarız.
- Admin kullanıcısı oluşturmak için gerekli bilgileri gireriz.
- Jenkins URL'sini yapılandırırız.

```
sudo apt update
sudo apt install openjdk-11-jdk
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key \ | sudo apt-key add -
sudo mkdir -p /etc/apt/sources.list.d/
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt update
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5BA31D57EF5975CA
sudo apt install jenkins
sudo systemctl start jenkins
```

Python İçin Gerekli Ortamı Ubuntu içerisinde Kurma

- ■İlk olarak, Ubuntu paketlerinin güncel olduğundan emin olmalıyız. Terminali açıp paket listesini güncelleriz.
- ■Ubuntu genellikle Python'un bir sürümünü önceden yükler, ancak en son sürümü elde etmek için komut kullanabiliriz.
- ■Python'un doğru bir şekilde yüklendiğini kontrol etmek için kontrol edebiliriz.
- ■Python paketlerini yönetmek için pip adlı bir paket yöneticisine ihtiyacımız olur. Ubuntu'da pip'i yükleriz.
- ■Kurulum tamamlandıktan sonra, pip'in doğru bir şekilde yüklendiğinden emin olmak için versiyonu kontrol ederiz.
- Python projeleri için izole edilmiş sanal ortamlar oluşturmak genellikle iyi bir uygulamadır. Bu adım, proje bağımlılıklarımızı yönetmek ve çakışmaları önlemek için farklı Python projeleri için ayrı ortamlar oluşturmamıza olanak tanır ve Sanal ortamı etkinleştiririz

```
sudo apt install python3
sudo apt install python3-venv
python3 -m venv venv
source venv/bin/activate
curl -sSfL https://raw.githubusercontent.com/anchore/grype/main/install.sh | sudo sh -s -- -b /usr/local/bin
sudo apt update
sudo apt install -y python3 python3-pip
deactivate
```

Grype'i Ubuntuda Kurma



Docker Kullanarak Grype Çalıştırmak:

1. Docker'ı yükleyin ve Grype Docker görüntüsünü çekin.
2. Çalıştırmak istediğiniz projenizin bulunduğu dizini belirtin.
3. Grype komutunu Docker üzerinden çalıştırın.

Doğrudan Kurulum:

1. Grype GitHub deposuna gidin.
2. Sağ üst köşede bulunan "Code" düğmesine tıklayın ve ZIP dosyasını indirin veya git kullanarak depoyu klonlayın.
3. İndirilen ZIP dosyasını çıkarın veya depoyu klonladığınız dizine gidin.
4. Terminal veya Komut İstemcisinde aşağıdaki komutları sırasıyla çalıştırın:



```
chmod +x grype  
sudo mv grype /usr/local/bin/
```


Docker'ı Ubuntuya Kurma



- Ubuntu'nun 64-bit bir sürümünü çalıştırdığınızdan emin olun. Docker, 64-bit bir işletim sistemi gerektirir.
- Docker, varsayılan Ubuntu depolarında bulunmayabilir, bu yüzden Docker'ı resmi Docker depolarından kurmamız gerekir. Bunun için öncelikle, Docker'ı indirmek için gerekli depo anahtarını ve depo dosyasını ekleriz.
- Depolar güncellendikten sonra Docker'ı yükleyebiliriz.
- Docker'ın doğru bir şekilde yüklendiğini kontrol etmemiz gerekir.
- Docker komutlarını kullanabilmek için, mevcut kullanıcıyı docker grubuna ekleriz.
- Bu değişikliğin etkili olması için oturumu kapatıp tekrar giriş yaparız.

```
{
  .sudo apt update && \
  .sudo apt install -y apt-transport-https ca-certificates curl software-properties-common && \
  .curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-
archive-keyring.gpg && \
  .echo "deb [signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu
\$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null && \
  .sudo apt update && \
  .sudo apt install -y docker-ce docker-ce-cli .containerd.io && \
  .sudo systemctl start docker && \
  .sudo systemctl enable docker && \
  .sudo usermod -aG docker $USER
}
```



Jenkins ile Konfigürasyonların Ayarlanması

▪ Jenkins'e Giriş Yapma

Jenkins'in kurulu olduğu sunucunun IP adresini veya alan adını tarayıcınıza gireriz.

▪ Yeni Bir Proje Oluşturma

- Jenkins ana sayfasında, sol taraftaki menüde "Yeni İş"e tıklayıp yeni bir proje oluşturmak için uygun bir proje tipi seçip oluştururuz.

▪ Proje Yapılandırması

Projenizin ana sayfasında, sol menüde "Yapılandır" veya "Proje Yapılandır" gibi bir seçenek buluruz. Proje yapılandırma sayfasında, genel proje ayarlarını yapılandırıp tetikleyicilerini ve diğer yapılandırmaları ayarlarız. Yapılandırmayı kaydetmek için sayfanın altındaki "Kaydet" veya "Uygula" düğmesine tıklarız.

▪ Projenin Başlatılması ve İzlenmesi

Projenin ana sayfasına gidip "Şimdi yap" veya "Build Now" gibi bir butona tıklayarak projeyi başlatırız. Build işlemi başladıktan sonra, Jenkins ana sayfasındaki "Build History" bölümünden build'ın durumunu ve sonuçlarını izlememiz gerekir. Gerekirse, build'ın ayrıntılarını incelemek için build numarasına tıklanır.

Konfigürasyonu Düzenle

Genel

Kaynak Kodu Yönetimi

Yapılandırma Tetikleyiciler

Yapılandırma Ortamı

Yapılandırma Adımları

Yapılandırma Sonrası Aksiyonlar

Githubda Webhook Ayarlama ve Dosyaları Yükleme



- GitHub Depo Oluşturma Adımları Tamamlandıktan Sonra...
- Webhook Oluşturma
- ■ Oluşturduğunuz depoya gidip üst menüdeki "Settings" (Ayarlar) sekmesine tıklarız. Sol menüdeki "Webhooks" bölümüne gidip "New webhook" düğmesine tıklarız.
- ■ Payload URL alanına Jenkins'in webhook URL'sini gireriz "Content type" alanını application/json olarak ayarlarız. "Which events would you like to trigger this webhook?" veya benzer bir alan altında, hangi olayların webhook'u tetiklemesini istiyorsak onları seçeriz (örneğin, Push, Pull Request, etc.).



Clusterda Argocd Kurulması



- **Kubectl Yükleme**
 - Eğer henüz yüklü değilse, kubectl'i yükleriz. Kubectl, Kubernetes kümeleriyle etkileşimde bulunmak için kullanılır.
- **Argo CD YAML Dosyalarını İndirme**
 - Argo CD'nin YAML dosyalarını indirmek için GitHub deposuna gidip YAML dosyalarını indiririz veya kopyalarız.
- **Argo CD'yi Kubernetes Cluster'a Kurma**
 - İndirdiğiniz YAML dosyalarının bulunduğu dizine gidip Argo CD'nin tüm bileşenlerini Kubernetes cluster'ınıza yükleriz.
- **Argo CD'nin Durumunu Kontrol Etme**
 - Argo CD'nin başarıyla kurulup kurulmadığını kontrol ederiz.
- **Argo CD Hizmetine Erişim Sağlama**
 - Argo CD hizmetine erişmek için bir yönlendirme (service) oluşturmamız gerekebilir. Bunun için komut kullanırız. Bu komut, yerel makinemizde 8080 portunu kullanarak Argo CD web arayüzüne erişmemizi sağlar. Tarayıcımızı açıp ve <http://localhost:8080/> adresini ziyaret ederek Argo CD web arayüzüne erişiriz.
- **Argo CD'nin Kullanımı**
 - Argo CD web arayüzünden giriş yaparız. Varsayılan kullanıcı adı admin, varsayılan şifre ise password ile giriş yaparız.



APPLICATION DETAILS TREE

DETAILS

DIFF

SYNC

SYNC STATUS

HISTORY AND ROLLBACK

DELETE

REFRESH

Log out

APP HEALTH

Healthy

SYNC STATUS

Synced to main (7f670d9)

Auto sync is enabled.

Author: Muhammed Enes Sarica <muhammed_enes04@hotmail.com>

Comment: Update deployment.yaml

LAST SYNC

Sync OK to 7f670d9

Succeeded a day ago (Wed Jan 31 2024 18:34:14 GMT+0300)

Author: Muhammed Enes Sarica <muhammed_enes04@hotmail.com>

Comment: Update deployment.yaml

bulutbilisim

Healthy

a day

my-service

svc

Healthy

a day

bulutbilisim

deploy

Healthy

a day

rev:1

my-service

ep

a day

my-service-d5p5w

endpointslice

a day

bulutbilisim-757cfbf546

rs

a day

rev:1

bulutbilisim-757cfbf546-64wjt

pod

a day

running

1/1

bulutbilisim-757cfbf546-vtk4f

pod

a day

running

1/1



Clusterda Monitoring İçin Prometheus ve Grafana Kurma

- **Prometheus ve Grafana YAML Dosyalarını İndirme**
- Prometheus ve Grafana'nın YAML dosyalarını indirmek için GitHub depolarına gideriz. İlgili YAML dosyalarını indiririz veya kopyalarız.
- **Prometheus ve Grafana'nın Kurulumu**
- İndirdiğimiz YAML dosyalarının bulunduğu dizine gideriz. Prometheus ve Grafana'nın tüm bileşenlerini Kubernetes cluster'ınıza yükleriz.
- **Servislerin Durumunu Kontrol Etme**
- Prometheus ve Grafana servislerinin başarıyla kurulduğunu ve çalıştığını kontrol ederiz.
- **Grafana'ya Erişim Sağlama**
- Grafana'nın web arayüzüne erişmek için bir yönlendirme oluşturmamız gerekebilir. Yerel makinenizde 3000 portunu kullanarak Grafana web arayüzüne erişmenizi sağlar.
- **Grafana'da Prometheus Verilerini Görselleştirme**
- Grafana web arayüzünde, sol taraftaki menüden "Configuration" sekmesine gidip "Data Sources" (Veri Kaynakları) seçeneğine tıklarız. "Add data source" düğmesine tıklayarak yeni bir veri kaynağı ekleriz ve Prometheus'u seçeriz. Prometheus veri kaynağını yapılandırırız.
- Grafana'da istediğiniz görselleştirmeleri oluşturmak ve Prometheus verilerini izlemek için Grafana'ya başlamak için "Create" sekmesine gideriz.





Grafana'da Prometheus İle Grafik Oluşturma

- Grafana'ya Giriş Yapma

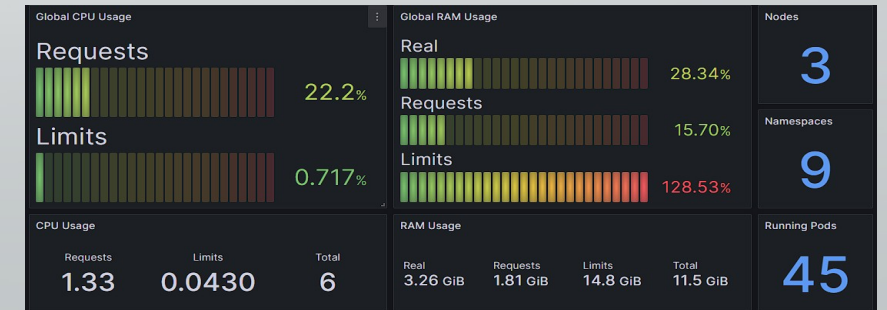
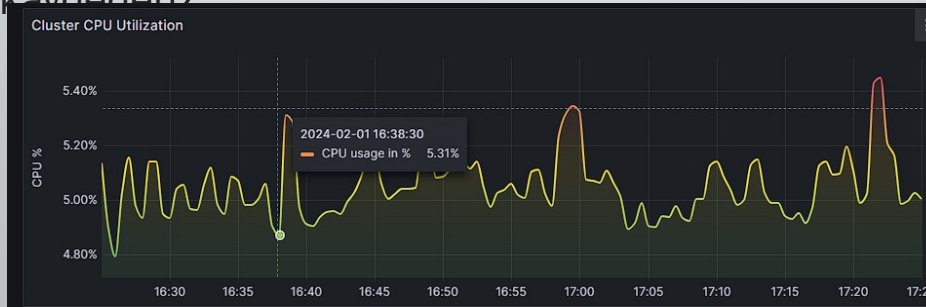
- Tarayıcımızı açıp Grafana web arayüzüne erişmek için adres çubuğuna Grafana URL'sini gireriz. Kullanıcı adı ve şifrenizi girerek Grafana'ya giriş yaparız.

- Veri Kaynağı Ekleme

- Grafana ana sayfasında, sol taraftaki menüden "Configuration" sekmesine gideriz. "Data Sources" seçeneğine tıklayarak yeni bir veri kaynağı ekleriz. "Add data source" düğmesine tıklayıp ve Prometheus'u seçin. Prometheus veri kaynağını yapılandırırız. Diğer gerekli bilgileri girip "Save & Test" düğmesine tıklayarak veri kaynağını kaydedip test ederiz.

- Grafik Oluşturma

- Grafana ana sayfasında, sol taraftaki menüden "Create" sekmesine gidip "Dashboard" seçeneğine tıklarız. "Add new panel" düğmesine tıklayıp "Graph" seçeneğini seçeriz. "Panel Title" bölümüne bir başlık gireriz. "Query" sekmesine gidip "Panel Data Source" altından Prometheus'u seçeriz. "Metric" sekmesine gidip Prometheus'dan veri almak için gerekli sorguyu yazarız. Grafikte görüntülemek istediğimiz metrikleri seçin veya sorguları yapılandırırız. Grafik ayarlarımızı tamamladıktan sonra, "Apply" veya "Save" düğmesine tıklayarak paneli kaydederiz.




Data Sources;

Data sources

View and manage your connected data source connections

Sort by A-Z



prometheus

Prometheus | http://34.42.237.138:9090 | default

Build a dashboard

Explore

Grafik;

Home > Dashboards > Kubernetes / Views / Global

Search or jump to...

ctrl+k

+

ⓘ

📶

🏠

🔍

🔄

Last 1 hour

🔍

🔄

30s

⬆

datasource prometheus cluster None resolution 30s

Overview

Global CPU Usage

Requests

22.2%

Limits

0.717%

Global RAM Usage

Real

28.60%

Requests

15.70%

Limits

128.53%

CPU Usage

Requests

1.33

Limits

0.0430

Total

6

RAM Usage

Real

3.29 GiB

Requests

1.81 GiB

Limits

14.8 GiB

Total

11.5 GiB

Nodes

3

Namespaces

9

Running Pods

45

Kubernetes Resource Count

Name	Min	Max	Mean
Running Containers	68	68	68
Running Pods	45	45	45
Configmaps	39	39	39
Endpoints	24	24	24
Services	23	23	23
Secrets	9	9	9
Persistent Volume Claims	3	3	3
Nodes	3	3	3

Resources

Cluster CPU Utilization

Cluster Memory Utilization