

## **İZLEME ve ALARM YÖNETİMİ**

**Proje:** Güvenilirlik, ölçeklenebilirlik ve performans gerektiren uygulamalar için izleme ve alarm yönetimi ihtiyaçlarını karşılamayı hedeflemektedir.

**Proje Açıklaması:** AWS CloudWatch hizmetini kullanarak uygulama ve altyapı bileşenlerinin izlenmesi ve alarm yönetiminin sağlanmasıdır. AWS CloudWatch, çeşitli metrikleri toplayarak sistem performansını, kullanımı ve kaynaklarını izlemek için kullanılır. Proje kapsamında, özel CloudWatch alarmları oluşturulacak ve önemli metrikler sürekli olarak izlenecektir. Bu alarmlar, belirli eşik değerlerinin aşılması durumunda bildirimler göndererek hızlı müdahale imkanı sağlayacaktır.

### **Projenin Temel Adımları:**

- AWS CloudWatch hizmetini kullanarak uygulama ve altyapı bileşenlerinin izlenmesi.
- Özel CloudWatch alarmları oluşturarak önemli metrikleri izleyin ve gerektiğinde bildirimler alınması.
- İzleme verilerini analiz etmek için AWS CloudWatch Logs ve AWS CloudWatch Insights gibi hizmetleri kullanılması.

### **Proje İçin Kullanılacak Amazon Hizmetleri**

- AWS CloudWatch
- AWS CloudWatch Logs
- AWS CloudWatch Insights

## AWS CloudWatch

Uygulama, hizmet ve altyapı bileşenlerin izlenmesi, logların toplanması ve olayların yönetimi için kullanılan bir hizmettir. CloudWatch, AWS kaynaklarının performansını, durumunu ve davranışını takip etmek ve analiz etmek için metrikleri, logları ve olayları toplar ve görselleştirir.

### Cloudwatch Alarm Oluşturma

CloudWatch konsoluna eriştikten sonra sol alanda bulunan “All Alarms” seçeneği seçilir. “Specify metric and conditions” sekmesinde istenilen Metric seçilir. Projede EC2 instance’ın CPU kullanımı %90’ı geçtiği takdirde alarm üretmesi için yapılandırma kullanılacaktır.

Select metric

Instance name 100/153

Instanceid

Metric name

<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	MetadataToken
<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	NetworkPacketsIn
<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	NetworkPacketsOut
<input checked="" type="checkbox"/>	No name specified	i-06cba95899fc8d...	CPUUtilization
<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	NetworkIn
<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	NetworkOut
<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	DiskReadBytes
<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	DiskWriteBytes
<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	DiskReadOps
<input type="checkbox"/>	No name specified	i-06cba95899fc8d...	DiskWriteOps

Cancel Select metric

Conditions

Threshold type

☒ Static  
Use a value as a threshold

☐ Anomaly detection  
Use a band as a threshold

Whenever CPUUtilization is...

Define the alarm condition.

☒ Greater  
> threshold

☐ Greater/Equal  
>= threshold

☐ Lower/Equal  
<= threshold

☐ Lower  
< threshold

than...

Define the threshold value.

90

Must be a number

► Additional configuration

Cancel Next

Configure Actions sekmesinde, alarm için gönderilecek bildirimin nereye gönderileceği bilgileri girilir.

# Configure actions

## Notification

### Alarm state trigger

Define the alarm state that will trigger this action.

☒ **In alarm**  
The metric or expression is outside of the defined threshold.

☐ **OK**  
The metric or expression is within the defined threshold.

☐ **Insufficient data**  
The alarm has just started or not enough data is available.

Remove

### Send a notification to the following SNS topic

Define the SNS (Simple Notification Service) topic that will receive the notification.

☐ Select an existing SNS topic

☒ Create new topic

☐ Use topic ARN to notify other accounts

### Create a new topic...

The topic name must be unique.

EC2-Cpu-Utilization-Alarm

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (\_).

### Email endpoints that will receive the notification...

Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

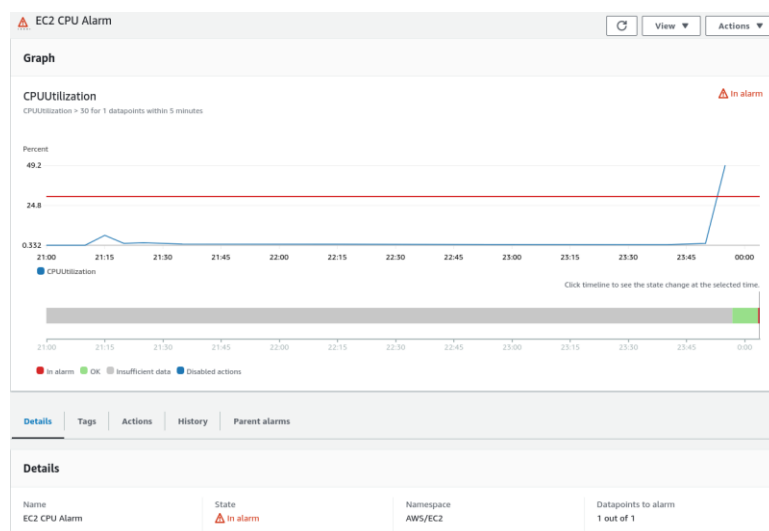
user@example.com

user1@example.com, user2@example.com

Create topic

Create notification

Proje kapsamında uygulama olarak CPU kullanımı %30 olarak ayarlanmıştır. Bash kodu ile CPU kullanımı %90 üzerine çıkartılmaya çalışılmış ve alarm test edilmiştir.



View this alarm in the AWS Management Console:  
<https://eu-central-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=eu-central-1#alarmsV2:alarms/EC2%20CPU%20Alarm>

<https://eu-central-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=eu-central-1#alarmsV2:alarm/EC2%20CPU%20Alarm>

```
- Name: EC2 CPU Alarm
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [49.
- Timestamp: Thursday 25 May, 2023 00:03:30 UTC
xxxx
- Alarm Arn: arn:aws:cloudwatch:eu-central-1:xxx:alarm:EC2 CPU Alarm
```

- The alarm is in the ALARM state when the metric is GreaterThanThreshold 30.0 for at least 1 of the last 1 period(s) of 300 seconds.

State Change Actions:

- OK:
- ALARM: [arn:aws:sns:eu-central-1:120703883523:Cpu-Anomaly]
- INSUFFICIENT\_DATA:

## CloudWatch Logs

CloudWatch Logs, uygulama günlüklerini merkezi bir konumda toplayarak günlük verilerinin izlenmesini ve analizini kolaylaştırır.

## EC2 Instance için CloudWatch Logs Oluşturma

CloudWatch hizmetinde sol tarafta bulunan “Log Groups” seçeneği seçilir ve grup ismi verildikten sonra Log grubu oluşturulur. Ardından ssh ile instance’a bağlantı sağlanır ve sanal sunucuya Amazon CloudWatch Logs Agent servisi kurulur ve sistem başlatılır. Amazon CloudWatch Logs Agent, Amazon Web Services (AWS) CloudWatch hizmetine log verilerini göndermek için kullanılan bir araçtır..

```
“sudo yum install -y awslogs;  
sudo systemctl enable awslogsd;  
sudo systemctl start awslogsd”
```

Log dosyasında değişiklik yapmak için aşağıdaki komut kullanılır.

```
“sudo vi /etc/awslogs/awslogs.conf”
```

Bu dosyada, log gruplarını ve izlenecek log dosyalarını yapılandırabilirsiniz. Örneğin, /var/log/messages dosyasını izlemek için aşağıdaki gibi bir örnek ekleyebilirsiniz:

```
[/var/log/messages]
```

```
datetime_format = %b %d %H:%M:%S
```

```
file = /var/log/messages
```

```
log_stream_name = {instance_id}/var/log/messages
```

```
log_group_name = MyLogGroup
```

*“nano /etc/awslogs/awscli.conf”* komutu ile region konum ile değiştirilir. Ardından CloudWatch loglardan loglar takip edilebilir.

CloudWatch > Log groups > Ec2-Log-Group > i-0c753b31322a6e89/var/log/messages	
<b>Log events</b> You can use the filter bar below to search for and match terms, phrases, or values in your log events. <a href="#">Learn more about filter patterns</a>	
<input type="text" value="Filter events"/> <span>Clear</span> <span>1m</span> <span>30m</span> <span>1h</span> <span>12h</span> <span>Custom</span> <span>Display</span> <span>⌵</span> <span>⊞</span>	
Timestamp	Message
There are older events to load. <a href="#">Load more</a> .	
▶ 2023-05-25T01:55:50.000+03:00	May 24 22:55:50 ip-10-0-1-218 dhclient[2928]: XMT: Solicit on eth0, interval 129160ms.
▶ 2023-05-25T01:57:59.000+03:00	May 24 22:57:59 ip-10-0-1-218 dhclient[2928]: XMT: Solicit on eth0, interval 112900ms.
▶ 2023-05-25T01:59:52.000+03:00	May 24 22:59:52 ip-10-0-1-218 dhclient[2928]: XMT: Solicit on eth0, interval 121460ms.
▶ 2023-05-25T02:00:01.000+03:00	May 24 23:00:01 ip-10-0-1-218 systemd: Created slice User Slice of root.
▶ 2023-05-25T02:00:01.000+03:00	May 24 23:00:01 ip-10-0-1-218 systemd: Started Session 877 of user root.
▶ 2023-05-25T02:00:01.000+03:00	May 24 23:00:01 ip-10-0-1-218 systemd: Removed slice User Slice of root.
▶ 2023-05-25T02:00:16.000+03:00	May 24 23:00:16 ip-10-0-1-218 dhclient[2879]: DHCPREQUEST on eth0 to 10.0.1.1 port 67 (xid=0x7d8d983e)
▶ 2023-05-25T02:00:16.000+03:00	May 24 23:00:16 ip-10-0-1-218 dhclient[2879]: DHCPACK from 10.0.1.1 (xid=0x7d8d983e)
▶ 2023-05-25T02:00:16.000+03:00	May 24 23:00:16 ip-10-0-1-218 NET: dhclient: Locked /run/dhclient/resolv.lock
▶ 2023-05-25T02:00:16.000+03:00	May 24 23:00:16 ip-10-0-1-218 dhclient[2879]: bound to 10.0.1.218 -- renewal in 1580 seconds.
▶ 2023-05-25T02:00:16.000+03:00	May 24 23:00:16 ip-10-0-1-218 ec2net: [get_meta] Querying IMDS for meta-data/network/interfaces/macsa/02:fc:d7:41:aba8/local-ipv4s
▶ 2023-05-25T02:00:16.000+03:00	May 24 23:00:16 ip-10-0-1-218 ec2net: [get_meta] Getting token for IMDSv2.
▶ 2023-05-25T02:00:16.000+03:00	May 24 23:00:16 ip-10-0-1-218 ec2net: [get_meta] Trying to get http://169.254.169.254/latest/meta-data/network/interfaces/macsa/02:fc:d7:41:aba8/local-ipv4s
▶ 2023-05-25T02:00:16.000+03:00	May 24 23:00:16 ip-10-0-1-218 ec2net: [remove_aliases] Removing aliases of eth0
▶ 2023-05-25T02:01:01.000+03:00	May 24 23:01:01 ip-10-0-1-218 systemd: Created slice User Slice of root.
▶ 2023-05-25T02:01:01.000+03:00	May 24 23:01:01 ip-10-0-1-218 systemd: Started Session 878 of user root.
▶ 2023-05-25T02:01:01.000+03:00	May 24 23:01:01 ip-10-0-1-218 systemd: Removed slice User Slice of root.
▶ 2023-05-25T02:01:53.000+03:00	May 24 23:01:53 ip-10-0-1-218 dhclient[2928]: XMT: Solicit on eth0, interval 113060ms.

## AWS CloudWatch Insight

AWS CloudWatch Logs Insights, Amazon CloudWatch Logs hizmetinin bir özelliğidir. Bu özellik, log verilerinizi sorgulamak, analiz etmek ve görselleştirmek için kullanılan bir hizmettir.

CloudWatch hizmetinde “Log Group” seçeneği seçildikten sonra sağ üstte bulunan “View in Insight” seçeneği seçilir. Sorgu alanına log verilerinizi sorgulayacak olan sorgu ifadesini girilir. Sorgu dili, SQL benzeri bir formattadır ve log verileri filtrelemek, gruplamak, hesaplamalar yapmak gibi birçok işlemi gerçekleştirilebilir. Örnek olarak, “*fields @timestamp, @message | filter @message like /error/*” gibi bir sorgu yazılabilir. “Run query” seçeneği seçilerek sorgu cevabı alınır.

