

▶ AWS CloudWatch



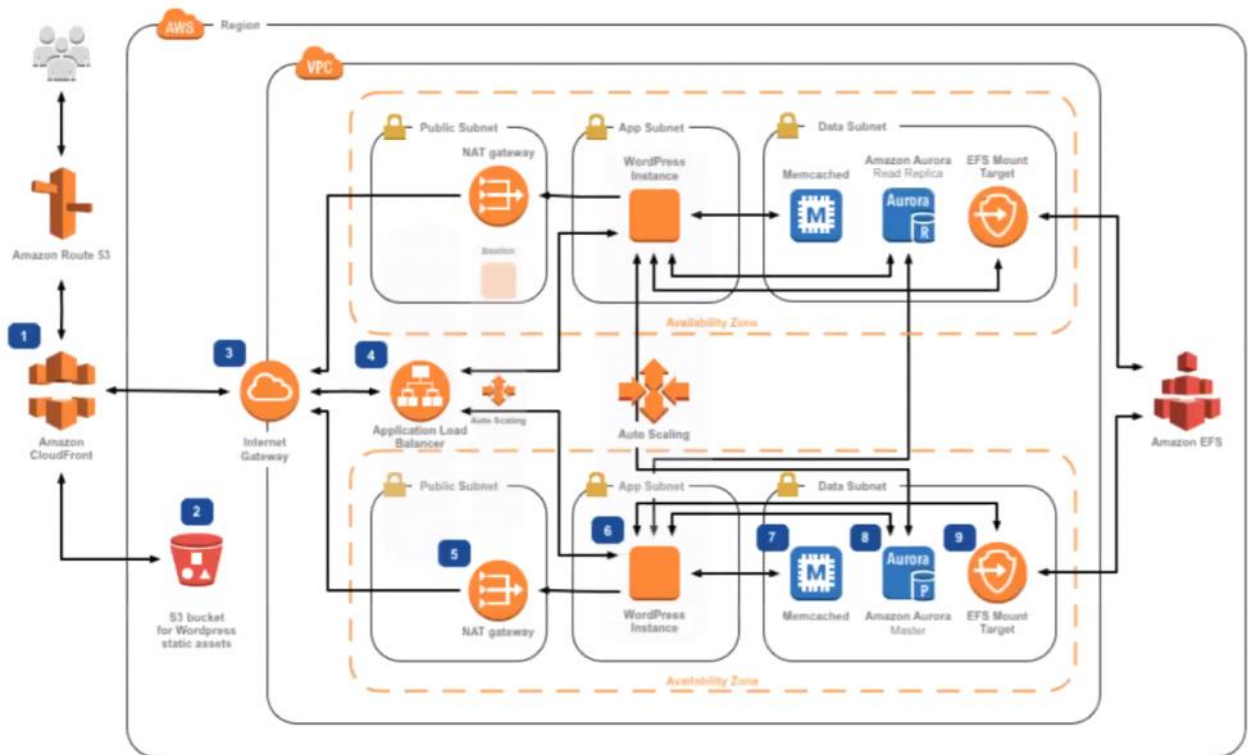
Table of Contents

- ▶ What is CloudWatch?
- ▶ CloudWatch Basic Components
- ▶ How Does CloudWatch Work?
- ▶ CloudWatch Pricing

1

What is CloudWatch?

Calling Manager



✓ [What is CloudWatch?](#)

CloudWatch is the monitoring service of AWS. It enables you to monitor metrics of resources and applications that run on AWS as well as on-premise servers. It also lets you get logs, set alarms, trigger events and analyze utilization.

► CloudWatch

What is CloudWatch?



- You can get logs, set alarms, trigger events and analyze utilization by CloudWatch.

AWS nin configurationdan sorumlu monitoring servisi diyebiliriz
Log kayıtlari 10 yila kadar metricleri ise 15 aya kadar saklayabiliriz

CloudWatch

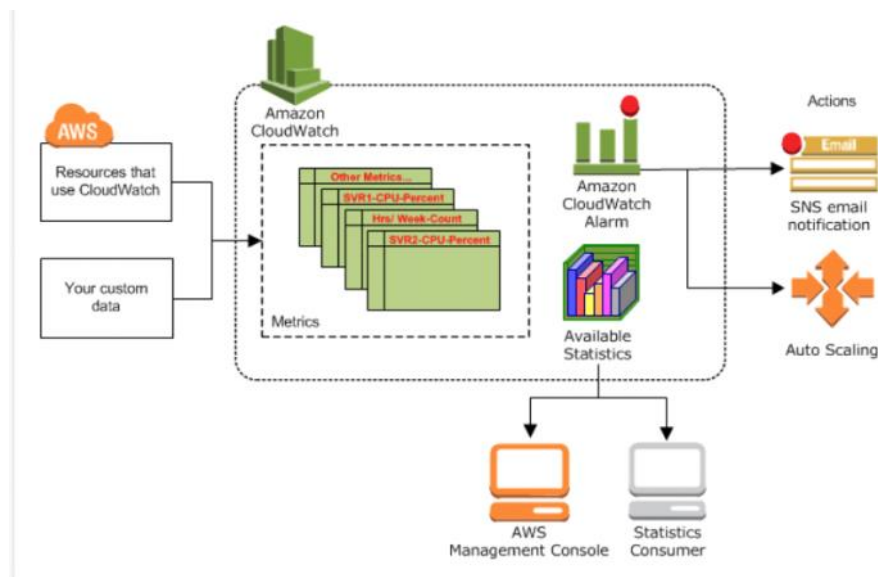
What is CloudWatch?



- CloudWatch is the **monitoring service** of AWS.
- It enables you to monitor **metrics of resources and applications** that run **on AWS** as well as **on-premise servers**.

✓ [How does CloudWatch work?](#)

CloudWatch collects metrics and logs from AWS resources, applications and services running on AWS and on-premises servers. You can visualize applications and infrastructure using **Dashboards**; correlate logs and metrics side by side to troubleshoot and set **Alarms**. It also enables you to automate response to operational changes with **Events** and **Auto-Scaling**. You can leverage metrics (up to 1 second), extended data retention (15 months) and real-time analysis with **Metric Math**.



2

CloudWatch Basic Components

✓ [Basic Components of CloudWatch](#)

CloudWatch

Basic Components



- **Metrics:** Metrics are the **measurable data** about your resources, apps or services like EC2 instance CPU utilization or the number of objects in an S3 bucket.

Namespace : Verilen gorsel bakimdan ayni baslik altinda toplanmasi

Metrics: Metrics are the measurable data about your resources, apps or services like EC2 instance CPU utilization or the number of objects in an S3 bucket.

CloudWatch

Basic Components



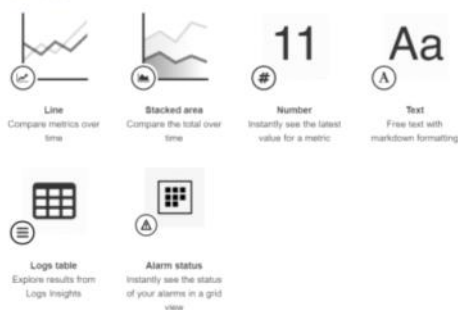
- **Dashboards:** Dashboards enable you to create graphs and visualize your cloud resources and applications according to your needs.

Global olarak dusunebiliriz. Farkli regionlardaki resourcelerin metriclerini ekleyebiliriz

Refresh time : 10 saniyede bir en az olarak guncelleme imkani verebilmektedir

CloudWatch


Basic Components



- **Dashboards:** You can leverage dashboards as customizable home pages in the CloudWatch console to view metrics, logs and alarms.

Dashboards: Dashboards enable you to create graphs and visualize your cloud resources and applications according to your needs. You can leverage dashboards as customizable home pages in the CloudWatch console to view metrics, logs and alarms.

CloudWatch Basic Components



- Alarms:** Alarms enables you to monitor CloudWatch metrics and to receive notifications if the metrics fall outside of the thresholds (higher or lower) that you set.

Alarms: Alarms enables you to monitor CloudWatch metrics and to receive notifications if the metrics fall outside of the thresholds (higher or lower) that you set.

"Cloudwatch in gucu alarmlardan geliyor denilebilir."

Thresholds: ORN==> EC2 nun CPUkullanimi- %60 in üzerine gectigi zaman alarm gonder diyebilirsiniz bir nevi **esik** degeri

Alarmın 3 seviyesi vardır

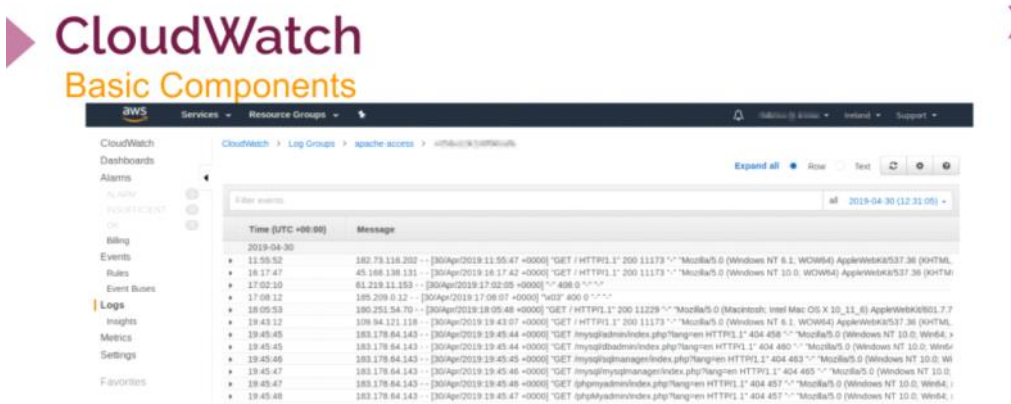
Alarm	Insufficient	OK
1	-	-

İlk asaması insufficient

Diger asama OK ==> %60 in altındaki durumlarda(yukarıdaki orn)

ALARM ==> %60 in üzeri

CloudWatch Basic Components



- Logs:** Logs let you monitor, store, and access your log files from AWS or other resources. It also allows you to centralize these logs for querying and analyzing.

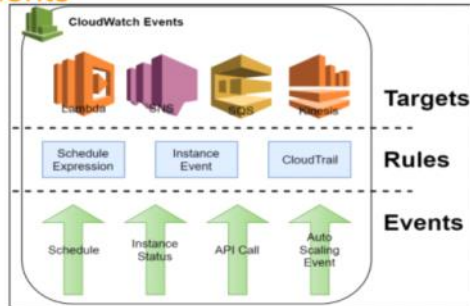
Logs: Logs let you monitor, store, and access your log files from AWS or other resources. It also allows you to centralize these logs for querying and analyzing.

Troubleshoot ve **analysis** aklımıza gelebilir. EC2 üzerinde app çalıştırıyoruz ve her bir request log kaydı olarak geldiğini düşünebiliriz

Log incelemesi önemli bir iş onun için piyasada özel toollar var mesela Splunk en iyisi ve piyasada çok kullanılanı genel de sitemci ve Siberçiler logları inceler

CloudWatch

Basic Components



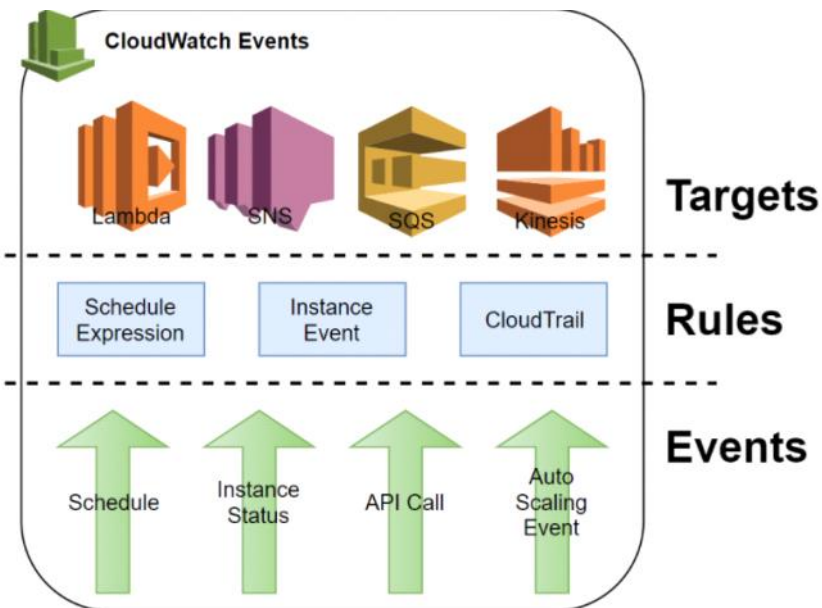
- **Events:** An event indicates changes in your AWS environment. AWS resources can generate events when their state changes, or you can create rules that self-trigger on an automated schedule.

Events: An event indicates changes in your AWS environment. AWS resources can generate events when their state changes, or you can create rules that self-trigger on an automated schedule. For example, you can receive an email via SNS if one of your EC2 instances goes to the "stop" state.

Gercege yakin olarak bizim sistemimizdeki degisikleri tutan **kurallar** butunudur

Mesela her sabah 8 de lambda fonksiyonu calissin ve verecegimiz kurala gore instance olustursun veya olusani start etsin gibi

is baslamadan 10 dakika once sistemi acsin bittikten 10 dk sonra kapat gibi



✓ [Accessing and Pricing](#)

CloudWatch Pricing

CloudWatch Pricing

- No up-front commitment or minimum fee
- Pay for what you use
- Pricing varies by region
- Enables limited **Free Tier** usage

CloudWatch Pricing

Free Tier:

	Basic Monitoring Metrics (at 5-minute frequency)
Metrics	10 Detailed Monitoring Metrics (at 1-minute frequency)
	1 Million API requests (not applicable to GetMetricData and GetMetricWidgetImage)
Dashboard	3 Dashboards for up to 50 metrics per month
Alarms	10 Alarm metrics (not applicable to high-resolution alarms)
Logs	5GB Data (ingestion, archive storage, and data scanned by Logs Insights queries)
Events	All events except custom events are included
Contributor Insights	1 Contributor Insights rule per month
	The first one million log events that match the rule per month
Synthetics	100 canary runs per month

CLARUSWAY
WAY TO REINVENT YOURSELF

Accessing:

You can access CloudWatch using any of the following methods:

- Amazon CloudWatch console - <https://console.aws.amazon.com/cloudwatch/>
- AWS CLI
- CloudWatch API - For more information, see the [Amazon CloudWatch API Reference](#).
- AWS SDKs - For more information, see [Tools for Amazon Web Services](#).

Pricing: [here](#)

CloudWatch doesn't require any up-front commitment or minimum fee. As many other Amazon Web Services, you simply pay for *what you use* and will be charged at the *end of the month* for your usage. Pricing varies by region.

Many services vend metrics for free, so that you can leverage

within **Free Tier**:

	Basic Monitoring Metrics (at 5-minute frequency)
Metrics	10 Detailed Monitoring Metrics (at 1-minute frequency) 1 Million API requests (not applicable to GetMetricData and GetMetricWidgetImage)
Dashboard	3 Dashboards for up to 50 metrics per month
Alarms	10 Alarm metrics (not applicable to high-resolution alarms)
Logs	5GB Data (ingestion, archive storage, and data scanned by Logs Insights queries)
Events	All events except custom events are included
Contributor Insights	1 Contributor Insights rule per month The first one million log events that match the rule per month
Synthetics	100 canary runs per month

3

How does CloudWatch work?

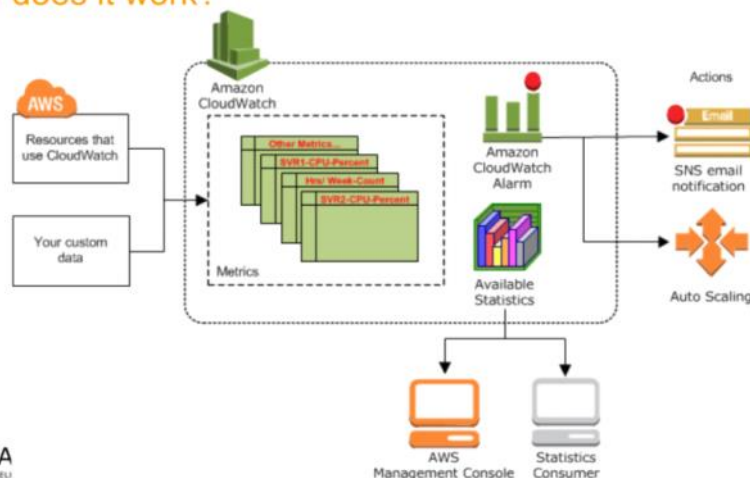
CloudWatch

How does it work?

- CloudWatch collects **metrics and logs** from AWS resources, applications and services running **on AWS and on-premises servers**.
- You can visualize applications and infrastructure using **Dashboards**; correlate logs and metrics side by side to troubleshoot and set **Alarms**.
- It also enables you to automate response to operational changes with **Events**.

CloudWatch

How does it work?




```

*****
*****
## Part 1 - Prep - Launching an Instance
STEP 1 : Create a EC2

```

```

AMI : Amazon Linux 2
Instance Type : t2.micro
Configure Instance Details:
- Monitoring --> Check "Enable CloudWatch detailed monitoring"
Tag :
  Key : Name
  Value : Cloudwatch_Instance
Security Group --> Allows ssh, http to anywhere
...
- Set user data.

```bash
#!/bin/bash
yum update -y
amazon-linux-extras install nginx1.12
chkconfig nginx on
cd /usr/share/nginx/html
chmod o+w /usr/share/nginx/html
rm index.html
wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/index.html
wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/ken.jpg
service nginx start
```

```

Userdata

```

#!/bin/bash
yum update -y
amazon-linux-extras install nginx1.12
chkconfig nginx on
cd /usr/share/nginx/html
chmod o+w /usr/share/nginx/html
rm index.html
wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/index.html
1
wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/ken.jpg
service nginx start

```

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of t1 management role to the instance, and more.

Number of instances ⓘ
1
Launch into Auto Scaling Group ⓘ

Purchasing option ⓘ
☐ Request Spot instances

Network ⓘ
vpc-282cb155 (default)
Create new VPC

Subnet ⓘ
No preference (default subnet in any Availability Zone)
Create new subnet

Auto-assign Public IP ⓘ
Use subnet setting (Enable)

Placement group ⓘ
☐ Add instance to placement group

Capacity Reservation ⓘ
Open

Domain join directory ⓘ
No directory
Create new directory

IAM role ⓘ
None
Create new IAM role

Shutdown behavior ⓘ
Stop

Stop - Hibernate behavior ⓘ
☐ Enable hibernation as an additional stop behavior

Enable termination protection ⓘ
☐ Protect against accidental termination

Monitoring ⓘ
☒ Enable CloudWatch detailed monitoring
Additional charges apply.

☒ Cloudwatch_Instance ⓘ
i-044f8d04d83cb40a5
Running ⓘ


A detailed illustration of Ken Masters from the Street Fighter series. He is shown in a dynamic, ready-to-fight pose, wearing his signature red gi with a black belt. His blonde hair is spiky, and he has a determined expression. He is barefoot, and his muscular physique is clearly defined. Below the character, the name 'KEN' is written in white capital letters inside a red rectangular box.

The screenshot shows the AWS CloudWatch console interface. On the left, a sidebar contains the following elements: a 'CloudWatch' header with a close icon, a toggle for 'New menu experience', a 'Favorites' section with a right-pointing arrow, a 'Dashboards' section in orange, and an 'Alarms' section with a dropdown arrow and three status icons (yellow triangle, green circle, grey circle). Below 'Alarms' are links for 'In alarm' and 'All alarms'. The main content area on the right features a blue-bordered box with an information icon, text about a new design, and a 'Try out the new interface' link. Below this is the 'Dashboards' heading and a blue 'Create dashboard' button.


Add to this dashboard

✕


Select a widget type to configure



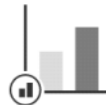
Line
Compare metrics over time




Stacked area
Compare the total over time



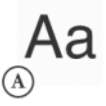
Number
Instantly see the latest value for a metric




Bar
Compare categories of data




Pie
Show percentage or proportional data



Text
Free text with markdown formatting



Logs table
Explore results from Logs Insights



Alarm status
Instantly see the status of your alarms in a grid view

Cancel

Next

Add to this dashboard

From which data source would you like to create the widget?

☒ **Metrics**
Create widget based on Metrics and configure your widget on the next step.

☐ **Logs**
Create widget based on query results from CloudWatch Logs Insights.

Cancel
Configure

All metrics

N. Virginia

Metriçler belirli başlıklar altında toplanmış ==> **namespace**

All metrics
Graphed metrics
Graph options
Source

N. Virginia
Search for any metric, dimension or resource id
Graph

1,980 Metrics

| | | |
|---------------------------------------|--------------------------------------|------------------------------|
| ApiGateway
31 Metrics | ApplicationELB
228 Metrics | Billing
16 Metrics |
| CertificateManager
1 Metric | CloudFront
6 Metrics | EBS
468 Metrics |
| EC2
841 Metrics | Lambda
36 Metrics | Logs
15 Metrics |
| NATGateway
28 Metrics | RDS
242 Metrics | Route 53
5 Metrics |
| S3 | States | Usage |

EC2
841 Metrics

Sonra

Per-Instance Metrics
710 Metrics

▼ Instance summary
info

Instance ID
i-044f8d04d83cb40a5
(Cloudwatch_Instance)

Ec2 id yi kopyalayalım

i-044f8d04d83cb40a5
Search for any metric, dimension or resource id
Graph search

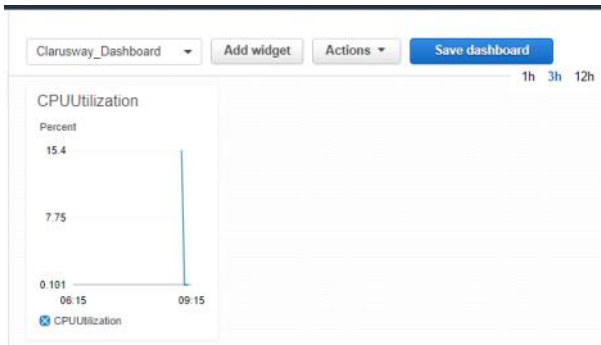
Alana yapistirip enter

All metrics
Graphed metrics (1)
Graph options
Source

N. Virginia
All > EC2 > Per-Instance Metrics
i-044f8d04d83cb40a5
Search for any metric, dimension or resource id
Graph search

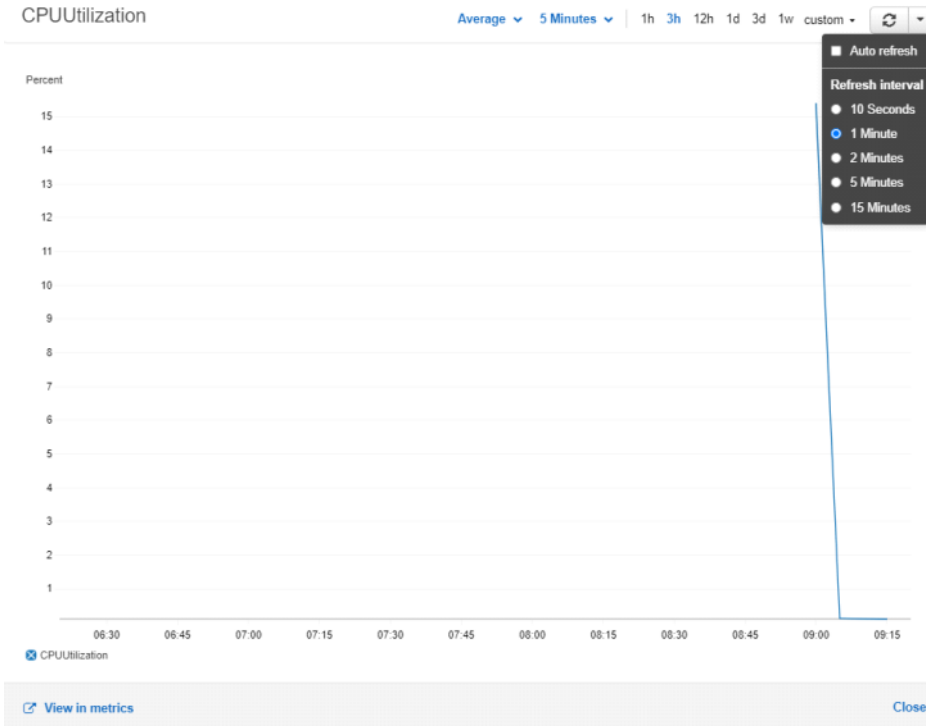
| Instance Name (17) | InstanceId | Metric Name |
|---|---------------------|----------------|
| <input checked="" type="checkbox"/> Cloudwatch_Instance | i-044f8d04d83cb40a5 | CPUUtilization |
| <input type="checkbox"/> Cloudwatch_Instance | i-044f8d04d83cb40a5 | NetworkIn |

Create widget



Sag ustten save edelim

Actigimiz ec2 icin cpu utilization basligi altinda metric ayarladik
Cloudwatch ile RAM degerlerini de gosterebiliriz



Refresh time i sag ustte gorebiliriz

Instance ye ssh ile baglanalim

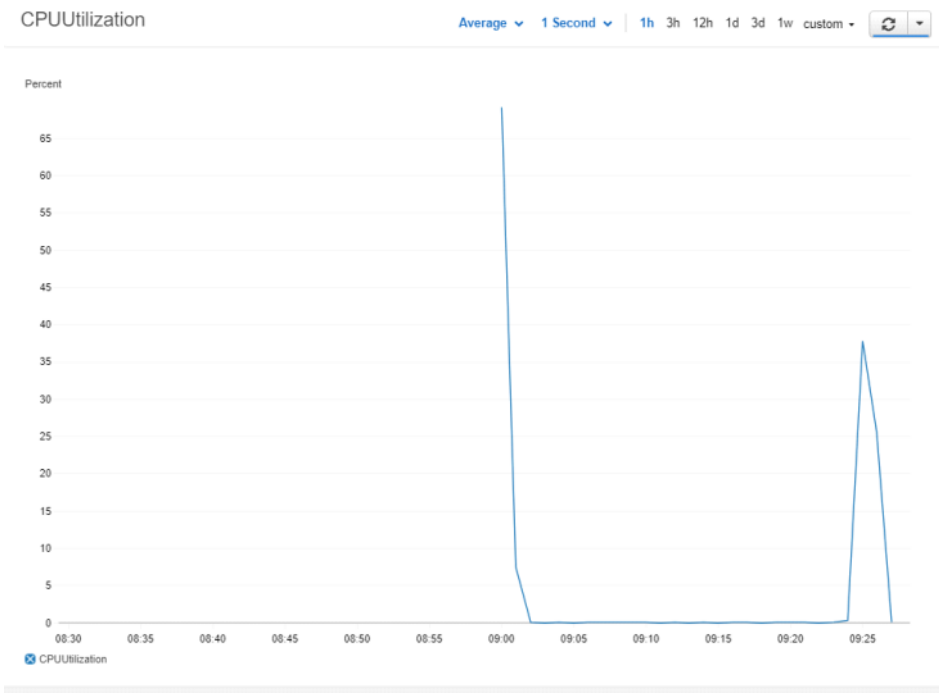
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[ec2-user@ip-172-31-53-250 ~]$
```

Asagidaki komutlari yazalim

```
sudo amazon-linux-extras install epel -y
sudo yum install -y stress
stress --cpu 80 --timeout 20000 #(optionally using 3000 for timeout)
```

Son komutla birlikte asagidaki gorseli gormemiz gerekmektedir

```
[ec2-user@ip-172-31-53-250 ~]$ stress --cpu 80 --timeout 20000
stress: info: [7700] dispatching hogs: 80 cpu, 0 io, 0 vm, 0 hdd
```



Ctrl c ile terminalden cikalim

Part 3 - Create an Alarm.

| Name | State | Conditions |
|-------------------|-------|----------------------------------|
| AWS_billing_alarm | OK | EstimatedCharges > 10 fr 6 hours |

Alarms

Alarm ile ilgili 3 asamayi gorebiliriz

Alarms (1)

Hide Auto Scaling al

Create alarm

CloudWatch > Alarms > Create alarm

Step 1
Specify metric and conditions

Step 2
Configure actions

Step 3
Add name and description

Step 4
Preview and create

Specify metric and conditions

Metric

Graph
Preview of the metric or metric expression and the alarm threshold.

Select metric

EC2 841

Per-Instance Metrics 710

Metrics (17) Graph search View graphed metrics (1)

All > EC2 > Per-Instance Metrics

| <input type="checkbox"/> | Instance Name (17) | InstanceId | Metric Name |
|-------------------------------------|---------------------|-----------------------|---------------------|
| <input type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 ▼ | NetworkPacketsOut ▼ |
| <input checked="" type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 ▼ | CPUUtilization ▼ |
| <input type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 ▼ | NetworkIn ▼ |
| <input type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 ▼ | NetworkOut ▼ |
| <input type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 ▼ | DiskReadBytes ▼ |

Cancel Select metric

Ilk sayfa gorunumu

Specify metric and conditions

Metric

Edit

Graph

This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

Percent

20

15

10

5

07:00 08:00 09:00

■ CPUUtilization

Namespace

AWS/EC2

Metric name

CPUUtilization

InstanceId

i-044f8d04d83cb40a5

Instance name

Cloudwatch_Instance

Statistic

Q Average

Period

5 minutes

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.

10000

Must be a number

► Additional configuration

Cancel

Next

degisiklikler

Period

1 minute

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.

60

Must be a number

► Additional configuration

Cancel

Next

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](#)

Select an 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

Create a new topic...
The topic name must be unique.

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.

user1@example.com, user2@example.com

Create topic

Add notification

Create topic diyelim

Mail imize gidip confirm etmeliyiz

Subscriptions (1)

[Edit](#) [Delete](#) [Request confirmation](#) [Confirm subscription](#) [Create subscription](#)

[<](#) [1](#) [>](#) [⚙](#)

| | ID | Endpoint | Status | Protocol |
|-----------------------|----------------------|----------------------|----------------------|----------|
| <input type="radio"/> | Pending confirmation | hamidgokce@gmail.com | Pending confirmation | EMAIL |

AWS Notifications <no-reply@sns.amazonaws.com> 12:40 PM (3 minutes ago) ☆ ↩ ⋮

to me ▾

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:000667629202:Clarus-alarm-sns

To confirm this subscription, click or visit the link below (if this was in error no action is necessary):
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

Confirm

Subscriptions (1)

[Edit](#) [Delete](#) [Request confirmation](#) [Confirm subscription](#) [Create subscription](#)

[<](#) [1](#) [>](#) [⚙](#)

| | ID | Endpoint | Status | Protocol |
|-----------------------|--------------------------------------|----------------------|-----------|----------|
| <input type="radio"/> | b1abdbff-55ec-4e73-980f-d7b3f6ce2cb1 | hamidgokce@gmail.com | Confirmed | EMAIL |

EC2 action

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](#)

Take the following action...
Define what will happen to the EC2 instance with the Instance ID i-044f8d04d83cb40a5 when this alarm is triggered.

☐ Recover this instance
You can only recover certain EC2 Instance types. [See documentation](#)

☒ **Stop this instance**
You can only stop an instance if it is backed by an EBS volume. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

☐ Terminate this instance
You will not be able to terminate this instance if termination protection is enabled. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

☐ Reboot this instance
An instance reboot is equivalent to an operating system reboot. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

Add EC2 action

Alarm olduğu zaman instanceyi stop et
Next

Add name and description

Name and description

Alarm name
My First Cloudwatch Alarm

Alarm description - optional
My First Cloudwatch Alarm

Up to 1024 characters (25/1024)

Cancel Previous Next

Next

Create alarm

CloudWatch > Alarms

Alarms (2) ☐ Hide Auto Scaling alarms Clear selection Create composite alarm Actions

Search Any state Any type

| <input type="checkbox"/> | Name | State | Last state update | Conditions | Actions |
|--------------------------|---------------------------|-------------------|---------------------|---|---------------------|
| <input type="checkbox"/> | My First Cloudwatch Alarm | Insufficient data | 2021-08-14 12:45:43 | CPUUtilization > 60 for 1 datapoints within 1 minute | 2 action(s) enabled |
| <input type="checkbox"/> | AWS_billing_alarm | OK | 2021-07-20 01:00:54 | EstimatedCharges > 10 for 1 datapoints within 6 hours | 1 action(s) enabled |

Biraz bekledikten sonra ok geldigini gorebiliriz

| <input type="checkbox"/> | Name | State |
|--------------------------|---------------------------|-------|
| <input type="checkbox"/> | My First Cloudwatch Alarm | OK |
| <input type="checkbox"/> | AWS_billing_alarm | OK |

Stress toolun tekrar aktif edelim

```
168 #### Step 1: Create Role
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[ec2-user@ip-172-31-53-250 ~]$ stress --cpu 80 --timeout 20000
stress: info: [4687] dispatching hogs: 80 cpu, 0 io, 0 vm, 0 hdd
```

| | | | |
|-------------------------------------|---------------------|---------------------|------------|
| <input type="checkbox"/> | Cloudwatch_Instance | i-092aa69f7ddcea89a | Terminated |
| <input checked="" type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 | Running |



Ec2 ekranı

| <input type="checkbox"/> | Name | Instance ID | Instance state |
|--------------------------|---------------------|---------------------|----------------|
| <input type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 | Stopped |

Cloudwatch ekranı

| <input type="checkbox"/> | Name | State | Last state update | Conditions |
|--------------------------|---------------------------|----------|---------------------|--|
| <input type="checkbox"/> | My First Cloudwatch Alarm | In alarm | 2021-08-14 13:03:37 | CPUUtilization > 60 for 1 datapoints within minute |

Stresstool u calistirdigimiz icin ec2 stop etti

Part 4 - CloudWatch Events with Lambda

Simdi bir tane event olusturacagiz

Lambda ve IAM servislerine gidelim

Policy olusturup lambda ya attach edecegiz

IAM > Policies

Policies (837) Info
A policy is an object in AWS that defines permissions.

Filter policies by property or policy name and press: < 1 2 3 4 5 6 7 ... 42 > @

| | Policy Name | Type |
|-----------------------|---|------------------|
| <input type="radio"/> | AWSLambdaBasicExecutionRole-0aa0883a-f646-4063-98b2-613523148a4f | Customer managed |
| <input type="radio"/> | AWSLambdaBasicExecutionRole-4deb3cd8-435c-47b1-907c-7c2bb5d82b... | Customer managed |

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Start*",
        "ec2:Stop*"
      ],
      "Resource": "*"
    }
  ]
}
```

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor JSON Import managed policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Start*",
        "ec2:Stop*"
      ],
      "Resource": "*"
    }
  ]
}
```

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Character count: 235 of 6,144.

Cancel

Next: Tags

Add tags (Optional)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

| | | |
|-----------------------------------|--|-----------------------------|
| Key | Value - optional | |
| <input type="text" value="name"/> | <input type="text" value="start-stop-instance"/> | <button>Remove tag</button> |

Add tag

You can add up to 49 more tags

Cancel Previous Next: Review

Name*

Use alphanumeric and '+', '@', '-' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

Summary

| Service | Access level | Resource | Request |
|---|----------------|--------------------|---------|
| Allow (2 of 289 services) <a>Show remaining 287 | | | |
| CloudWatch Logs | Limited: Write | arn:aws:logs:*:*:* | None |
| EC2 | Limited: Write | All resources | None |

Tags

| Key | Value |
|------|---------------------|
| name | start-stop-instance |

Required

Cancel Previous Create policy

Create role

Create role

1 2 3 4

Select type of trusted entity

| | | | |
|--|---|---|--|
| AWS service
EC2, Lambda and others | Another AWS account
Belonging to you or 3rd party | Web identity
Cognito or any OpenID provider | SAML 2.0 federation
Your corporate directory |
|--|---|---|--|

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

| | | | | |
|-----------------|------------------------|----------------------------|------------------|-------------|
| API Gateway | CodeBuild | EMR Containers | IoT SiteWise | RDS |
| AWS Backup | CodeDeploy | ElastiCache | IoT Things Graph | Redshift |
| AWS Chatbot | CodeGuru | Elastic Beanstalk | KMS | Rekognition |
| AWS Marketplace | CodeStar Notifications | Elastic Container Registry | Kinesis | RoboMaker |
| AWS Support | Comprehend | Elastic Container Service | Lake Formation | S3 |
| Amplify | Config | Elastic Transcoder | Lambda | SMS |
| AppStream 2.0 | Connect | ElasticLoadBalancing | Lex | SNS |

* Required

Cancel Next: Permissions

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy



| Filter policies ▼ | | Q start | Showing 1 result |
|-------------------|---------------------|---------|------------------|
| | Policy name ▼ | Used as | |
| ✓ ▶ | start-stop-instance | None | |

► Set permissions boundary

* Required

Cancel

Previous

Next: Tags

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name* start-stop-instance_role
Use alphanumeric and '+-=_.' characters. Maximum 64 characters.

Role description Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+-=_.' characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies start-stop-instance

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

Cancel

Previous

Create role

Lambda ya gecelim

Create function

Lambda > Functions > Create function

Create function [Info](#)

Choose one of the following options to create your function.

Author from scratch ☒
Start with a simple Hello World example.

Use a blueprint ☐
Build a Lambda application from sample code and configuration presets for common use cases.

Container image ☐
Select a container image to deploy for your function.

Browse serverless app repository ☐
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

[View the start-stop-instance_role role on the IAM console.](#)

► Advanced settings

[Cancel](#) [Create function](#)

Olusturdugumuz rolü sectik

Lambda ekranına python kodunu yapıştırılim

```
import boto3
region = 'us-east-1'
instances = ['i-02c107da60f5dad15'] #DON'T FORGET TO CHANGE ME
ec2 = boto3.client('ec2', region_name=region)
def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

Ec2 id sini unutmayalim



Ve deploy diyelim



Testi tiklayalim

Ve ec2 yu start edelim test edebilmek icin

Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

☒ Create new test event
 ☐ Edit saved test events

Event template

hello-world

Event name

teststop

1

Cancel Format JSON Create

Teste baselim

lambda_function

Execution result

Status: Succeeded Max memory used: 84 MB Time: 708.35 ms

Test Event Name

teststop

Response

null

Function Logs

START RequestId: 25b83e4e-8dbb-4276-9118-ab0d1f0c5c49 Version: \$LATEST
 stopped your instances: ['i-044f8d04d83cb40a5']
 END RequestId: 25b83e4e-8dbb-4276-9118-ab0d1f0c5c49
 REPORT RequestId: 25b83e4e-8dbb-4276-9118-ab0d1f0c5c49 Duration: 708.35 ms Billed Duration: 709 ms Memory Size: 128 MB Max Memory Used: 84

Request ID

25b83e4e-8dbb-4276-9118-ab0d1f0c5c49

Instances (1/1) Info

Filter instances

| <input checked="" type="checkbox"/> | Name | Instance ID | Instance state | Instance type | Stat |
|-------------------------------------|---------------------|---------------------|----------------|---------------|------|
| <input checked="" type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 | Running | t2.micro | - |

Test otomatik stop etti instanceyi

Instances (1) Info

Filter instances

| <input type="checkbox"/> | Name | Instance ID | Instance state | Instance typ |
|--------------------------|---------------------|---------------------|----------------|--------------|
| <input type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 | Stopping | t2.micro |

Aynı işlemleri start için de yapalım

Create function

Lambda > Functions > Create function

Create function [Info](#)

Choose one of the following options to create your function.

Author from scratch ☒

Start with a simple Hello World example.

Use a blueprint ☐

Build a Lambda application from sample code and configuration presets for common use cases.

Container image ☐

Select a container image to deploy for your function.

Browse serverless app repository ☐

Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.

start_instance

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.8

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

start-stop-instance_role

[View the start-stop-instance_role role on the IAM console.](#)

► **Advanced settings**

Cancel **Create function**

Kodumuzu ekleyelim ve ec2 id yi ekleyelim

Code source [Info](#)

File Edit Find View Go Tools Window **Test** **Deploy** **Changes not deployed**

Go to Anything (Ctrl-P)

Environment

- start_instance
 - lambda_function.py

```

1 import boto3
2 region = "us-east-1"
3 instances = ["i-044f8094d83cb40e2"]
4 ec2 = boto3.client('ec2', region_name=region)
5
6 def lambda_handler(event, context):
7     ec2.start_instances(InstanceIds=instances)
8     print('started your instances: ' + str(instances))
  
```

Deploy edelim

Configure test event ✕

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

☒ Create new test event
☐ Edit saved test events

Event template
hello-world ▼

Event name
teststart

1 {}

Cancel Format JSON Create

Upload from ▼

Go Tools Window Test + Deploy Changes deployed

Execution result

| | |
|-----------------|---|
| Test Event Name | teststart |
| Response | null |
| Function Logs | <pre>START RequestId: a7144ba5-5f9e-49f3-be1b-99a4061a59b9 Version: \$LATEST started your instances: [{"i-044f8d04d83cb40a5"}] END RequestId: a7144ba5-5f9e-49f3-be1b-99a4061a59b9 REPORT RequestId: a7144ba5-5f9e-49f3-be1b-99a4061a59b9 Duration: 830.24 ms Billed Duration: 831 ms Memory Size: 128 MB Max Memory Used: 85</pre> |
| Request ID | a7144ba5-5f9e-49f3-be1b-99a4061a59b9 |

Status: Succeeded Max memory used: 85 MB Time: 830.24 ms

Kapanan instance yinin lambda sayesinde acilabilecegini gorebiliriz

Instances (1) Info

Filter instances

| Name | Instance ID | Instance state | Instance type |
|---------------------|---------------------|----------------|---------------|
| Cloudwatch_Instance | i-044f8d04d83cb40a5 | Pending | t2.micro |

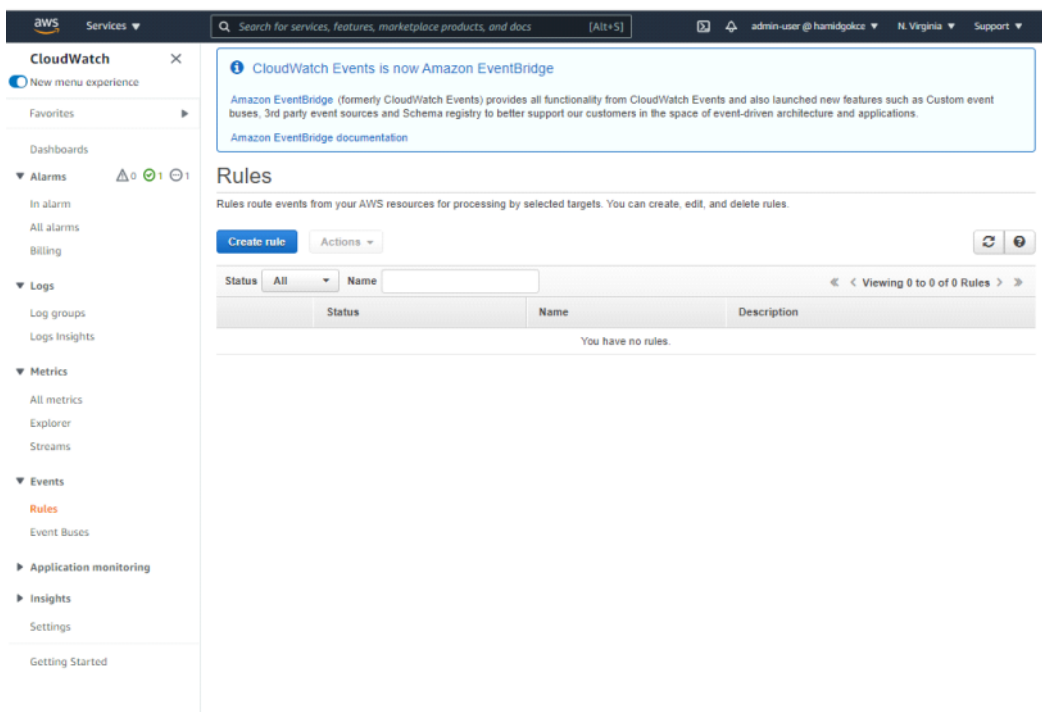
Instances (1) Info

Filter instances

| Name | Instance ID | Instance state | Instance type |
|---------------------|---------------------|----------------|---------------|
| Cloudwatch_Instance | i-044f8d04d83cb40a5 | Running | t2.micro |

Bu islemlerin herbirini manuel olarak yaptik. Eger biz bunu cloudwatch e tanimlarsak bir zamana endexlemis oluruz ve istedigimiz zamanda acilip kapanmasini saglayabiliriz

Cloudwatch ekranina geelim



Create rule

```
#### Step 6 Creating Stop-Cloudwatch Event

- Go to the CloudWatch Console and from left hand menu under the Event
sub-section
Rules-----> Create Rule

- Select the Event Source parameters:
...
Event Source:
- Select "Schedule"
- Cron expression: 0 18 ? * MON-FRI *
...
- Explain the cron parameters.

- Select the Target parameters:
...
Targets:
- Function: Stop_Instance
...
- Click "Configure Details"
...
- Name: Event_Stop
- Description: This event provides stop action.
- State: Enabled
...
- Click "Create Rule."
```

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

☐ Event Pattern **i** ☒ Schedule **i**

☒ Fixed rate of

☐ Cron expression

[Learn more about CloudWatch Events schedules.](#)

► Show sample event(s)

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

* Required

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

☐ Event Pattern ☒ Schedule

☐ Fixed rate of Minutes

☒ Cron expression

Next 10 Trigger Date(s)

1. Sat, 14 Aug 2021 18:00:00 GMT
2. Mon, 16 Aug 2021 18:00:00 GMT
3. Tue, 17 Aug 2021 18:00:00 GMT
4. Wed, 18 Aug 2021 18:00:00 GMT
5. Thu, 19 Aug 2021 18:00:00 GMT
6. Fri, 20 Aug 2021 18:00:00 GMT
7. Sat, 21 Aug 2021 18:00:00 GMT
8. Mon, 23 Aug 2021 18:00:00 GMT
9. Tue, 24 Aug 2021 18:00:00 GMT
10. Wed, 25 Aug 2021 18:00:00 GMT

[Learn more about CloudWatch Events schedules.](#)

[Show sample event\(s\)](#)

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function*

[Configure version/alias](#)

[Configure input](#)

* Required

Cancel

[Configure details](#)

Step 2: Configure rule details

Rule definition

Name*

Description

State ☒ Enabled

CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.

* Required

Cancel

[Back](#)

[Create rule](#)

Stop kurali hazır şimdi start kurali

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

☐ Event Pattern ☒ Schedule

☐ Fixed rate of Minutes

☒ Cron expression

Next 10 Trigger Date(s)

1. Mon, 16 Aug 2021 08:00:00 GMT
2. Tue, 17 Aug 2021 08:00:00 GMT
3. Wed, 18 Aug 2021 08:00:00 GMT
4. Thu, 19 Aug 2021 08:00:00 GMT
5. Fri, 20 Aug 2021 08:00:00 GMT
6. Sat, 21 Aug 2021 08:00:00 GMT
7. Mon, 23 Aug 2021 08:00:00 GMT
8. Tue, 24 Aug 2021 08:00:00 GMT
9. Wed, 25 Aug 2021 08:00:00 GMT
10. Thu, 26 Aug 2021 08:00:00 GMT

[Learn more about CloudWatch Events schedules.](#)

[Show sample event\(s\)](#)

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function*

[Configure version/alias](#)

[Configure input](#)

* Required

Cancel

[Configure details](#)

Step 2: Configure rule details

Rule definition

Name*

Description

State ☒ Enabled

CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.

* Required

Cancel

[Back](#)

[Create rule](#)

Olusturulan 2 rules

| Rules | | |
|--|-----------------------------|-----------------------------------|
| Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules. | | |
| Create rule Actions | | |
| Status | All | Name |
| Viewing 1 to 2 of 2 Rules | | |
| Status | Name | Description |
| <input type="radio"/> | event_start | This event provides start action. |
| <input type="radio"/> | event_stop | This event provides stop action. |

EC2 olusturma saati

| |
|------------------------|
| Launch time |
| 2021/08/14 13:20 GMT+3 |


Stop a geelim

| | | |
|-----------------------|----------------------------|----------------------------------|
| <input type="radio"/> | event_stop | This event provides stop action. |
|-----------------------|----------------------------|----------------------------------|

Edit diyelim

Rules > event_stop

Summary

ARN  arn:aws:events:us-east-1:000887820202:rule/event_stop

Schedule Cron expression @ 18 ? * MON-SAT *

Next 10 Trigger Date(s)

1. Sat, 14 Aug 2021 18:00:00 GMT

2. **Mon, 16 Aug 2021 18:00:00 GMT**

3. Tue, 17 Aug 2021 18:00:00 GMT

4. Wed, 18 Aug 2021 18:00:00 GMT

5. Thu, 19 Aug 2021 18:00:00 GMT

6. Fri, 20 Aug 2021 18:00:00 GMT

7. Sat, 21 Aug 2021 18:00:00 GMT

8. Mon, 23 Aug 2021 18:00:00 GMT

9. Tue, 24 Aug 2021 18:00:00 GMT

10. Wed, 25 Aug 2021 18:00:00 GMT

Status Enabled

Description This event provides stop action.

Monitoring [Show metrics for the rule](#)

Targets

Filter:

<< < Viewing 1 to 1 of 1 Targets > >

| Type | Name | Input | Role | Additional parameters |
|-----------------|-------------------------------|---------------|------|-----------------------|
| Lambda function | Stop_instance | Matched event | | |

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets:

☐ Event Pattern
 ☒ Schedule

☐ Fixed rate of Select

☒ Cron expression

Next 10 Trigger Date(s)

1. Mon, 16 Aug 2021 10:23:00 GMT
2. Tue, 17 Aug 2021 10:23:00 GMT
3. Wed, 18 Aug 2021 10:23:00 GMT
4. Thu, 19 Aug 2021 10:23:00 GMT
5. Fri, 20 Aug 2021 10:23:00 GMT
6. Sat, 21 Aug 2021 10:23:00 GMT
7. Mon, 23 Aug 2021 10:23:00 GMT
8. Tue, 24 Aug 2021 10:23:00 GMT
9. Wed, 25 Aug 2021 10:23:00 GMT
10. Thu, 26 Aug 2021 10:23:00 GMT

[Learn more about CloudWatch Events schedules](#)

[Show sample event\(s\)](#)

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

[Configure version/alias](#)
[Configure input](#)

[Add target*](#)

* Required

[Cancel](#)
[Configure details](#)

Step 2: Configure rule details

Rule definition

☒ State ☒ Enabled

CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.

* Required

[Cancel](#)
[Back](#)
[Update rule](#)

Ec2 yu kontrol ettigimizde lambda sayesinde stop edildigini gorebiliriz

| Instances (1) Info | | | | |
|--|---------------------|---------------------|----------------|---------------|
| <div> <input type="text" value="Filter instances"/> </div> | | | | |
| <input type="checkbox"/> | Name | Instance ID | Instance state | Instance type |
| <input type="checkbox"/> | Cloudwatch_Instance | i-044f8d04d83cb40a5 | Stopping | t2.micro |

Start rule de degisiklik yapalim gorseldeki gibi / update edelim

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

☐ Event Pattern
 ☒ Schedule

☐ Fixed rate of Select

☒ Cron expression

Next 10 Trigger Date(s)

1. Sat, 14 Aug 2021 10:55:00 GMT
2. Mon, 16 Aug 2021 10:55:00 GMT
3. Tue, 17 Aug 2021 10:55:00 GMT
4. Wed, 18 Aug 2021 10:55:00 GMT
5. Thu, 19 Aug 2021 10:55:00 GMT
6. Fri, 20 Aug 2021 10:55:00 GMT
7. Sat, 21 Aug 2021 10:55:00 GMT
8. Mon, 23 Aug 2021 10:55:00 GMT
9. Tue, 24 Aug 2021 10:55:00 GMT
10. Wed, 25 Aug 2021 10:55:00 GMT

[Learn more about CloudWatch Events schedules.](#)

[Show sample event\(s\)](#)

* Required

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Function*

[Configure version/alias](#)
[Configure input](#)

[Add target*](#)

[Cancel](#)
[Configure details](#)

<https://crontab.cronhub.io/>

<https://crontab.guru/> ==> tarih formatlarini ayarlayabilirsiniz

Yeni ec2 olusturalim-

```

### Part 5 - Configure Logging with Agent

STEP 1 : Create second EC2 Instance

- Go to EC2 menu using AWS console

- Launch an Instance

- Configuration of instance.

***text
AMI           : Amazon Linux 2
Instance Type : t2.micro
Tag           :
  Key         : Name
  Value       : Cloudwatch_Log
Security Group ---> Allows all traffic ---> anywhere
***

- Set user data.

***bash
#!/bin/bash
yum update -y
amazon-linux-extras install nginx1.12
chkconfig nginx on
cd /usr/share/nginx/html
chmod o+w /usr/share/nginx/html
rm index.html
wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/index.html
wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/ken.jpg
service nginx start
***

```

Monitoring [i](#) ☒ Enable CloudWatch detailed monitoring
Additional charges apply.

Userdata

```

#!/bin/bash
yum update -y
amazon-linux-extras install nginx1.12
chkconfig nginx on
cd /usr/share/nginx/html
chmod o+w /usr/share/nginx/html
rm index.html
wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/index.html
wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/ken.jpg
service nginx start

```


Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)

Value (256 characters maximum)

Instances

Volumes

Network Interfaces

Name

Cloudwatch Log

Add another tag (Up to 50 tags maximum)

| Type | Protocol | Port Range | Source | Descrip |
|------|----------|------------|-----------|---------|
| HTTP | TCP | 80 | 0.0.0.0/0 | |
| HTTP | TCP | 80 | :::0 | |
| SSH | TCP | 22 | 0.0.0.0/0 | |
| SSH | TCP | 22 | :::0 | |

Bu instancenin cloud watch e log kayitlarini tanimlayacagimiz rol ile gonderecegiz

IAM servise gidelim

IAM > Roles

Roles (8)

Info

Refresh

Delete

Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

< 1 >

Role name

Trusted entities

Last activity

Create role

1 2 3 4

Select type of trusted entity

AWS service

EC2, Lambda and others

Another AWS account

Belonging to you or 3rd party

Web identity

Cognito or any OpenID provider

SAML 2.0 federation

Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway

CodeBuild

EMR Containers

IoT SiteWise

RDS

AWS Backup

CodeDeploy

ElastiCache

IoT Things Graph

Redshift

AWS Chatbot

CodeGuru

Elastic Beanstalk

KMS

Rekognition

AWS Marketplace

CodeStar Notifications

Elastic Container Registry

Kinesis

RoboMaker

AWS Support

Comprehend

Elastic Container Service

Lake Formation

S3

Amplify

Config

Elastic Transcoder

Lambda

SMS

AppStream 2.0

Connect

ElasticLoadBalancing

Lex

SNS

AppSync

DMS

EventBridge

License Manager

SWF

Application Auto Scaling

Data Lifecycle Manager

Forecast

MQ

SageMaker

Application Discovery

Data Pipeline

GameLift

Machine Learning

Security Hub

* Required

Cancel

Next: Permissions

Create role

1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

Refresh

Filter policies

CloudWatchLogsFullAccess

Showing 1 result

Policy name

Used as

☒

CloudWatchLogsFullAccess

None

► Set permissions boundary

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.


Role name*

Use alphanumeric and '+', '@', '_' characters. Maximum 64 characters.

Role description

Maximum 1000 characters. Use alphanumeric and '+', '@', '_' characters.

Trusted entities

Policies  CloudWatchLogsFullAccess [Link](#)

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

[Cancel](#) [Previous](#) [Create role](#)

Create role

Search for services, features, marketplace products, and docs [Alt+S]

admin-user @ hanidgokce N. Virginia Support

Instances (1/2) Info

Filter instances

| Name | Instance ID | Instance state |
|---------------------|---------------------|----------------|
| Cloudwatch_Instance | i-044f8d04d83cb40a5 | Stopped |
| Cloudwatch_Log | i-05154b494fa6be8a | Running |

Actions

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Image and templates
- Monitor and troubleshoot

Change security groups

Get Windows password

Modify IAM role

Rolu attach edelim

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

[Create new IAM role](#)

[Cancel](#) [Save](#)

Olusturdugumuz rolu secelim ve save

Ec2 ya baglanalim

```
420 - select CloudwatchLogsFullAccess --> NE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[ec2-user@ip-172-31-54-98 ~]$
```

```
basli
sudo yum install -y awslogs
sudo systemctl start awslogsd
sudo systemctl enable awslogsd.service
...
```

Yukaridaki komutlari sirayla girelim

```
[ec2-user@ip-172-31-54-98 ~]$ sudo systemctl enable awslogsd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/awslogsd.service to /usr/lib/sy
stemd/system/awslogsd.service.
```

Cloudwatch ekraninda instance olusturmadan once en alttaki yoktu

CloudWatch > Log groups

Log groups (5)
By default, we only load up to 10000 log groups.

Filter log groups or try prefix search

Exact match

| Log group | Retention | Metric Fil... | Contributor Insig... | Subscri... |
|-----------------------------|--------------|---------------|----------------------|------------|
| /aws/lambda/hamidnumbers | Never expire | - | - | - |
| /aws/lambda/NumberGenerator | Never expire | - | - | - |
| /aws/lambda/start_instance | Never expire | - | - | - |
| /aws/lambda/Stop_instance | Never expire | - | - | - |
| /var/log/messages | Never expire | - | - | - |

CloudWatch > Log groups > /var/log/messages

/var/log/messages

Actions View in Logs Insights Search log group

Log group details

| | | | |
|---------------------------|--------------------------------|---------------------------|---|
| Retention
Never expire | Creation time
2 minutes ago | Stored bytes
- | ARN
arn:aws:logs:us-east-1:000667629202:log-group:/var/log/messages* |
| KMS key ID
- | Metric filters
0 | Subscription filters
0 | Contributor Insights rules
- |

Log streams Metric filters Subscription filters Contributor Insights Tags

Log streams (1)

Filter log streams or try prefix search

| Log stream | Last event time |
|---------------------|---------------------------------|
| i-05154b494fca6be8a | 2021-08-14 13:50:49 (UTC+03:00) |

Instancenin ilk acilistaki log kayitlari

CloudWatch > Log groups > /var/log/messages > i-05154b494fca6be8a

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

View as text Actions Create Metric Filter

Filter events

Clear 1m 30m 1h 12h Custom

| Timestamp | Message |
|-------------------------------|--|
| 2021-08-14T13:47:15.000+03:00 | There are older events to load. Load more . |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: v1.8.0_242 =>stable |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 28 firecracker available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 29 golantg.11 available \ |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: [v1.11.3 v1.11.11 v1.11.13 =>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 30 squid4 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 31 php7.3 available \ |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: [v7.3.2 v7.3.3 v7.3.4 v7.3.6 v7.3.8 v7.3.9 v7.3.10 =>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: v7.3.11 v7.3.13 =>stable |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 32 lustre2.10 available \ |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: [v2.10.5 v2.10.8 =>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 33 java-openjdk11 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 34 lynix available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 35 kernel-ng available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 36 gcc available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:15 ip-172-31-54-98 cloud-init: 37 mono available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 38 nginx available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 39 ruby2.6 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 40 rock available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 41 postgresql11 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 42 php7.4 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 43 livepatch available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 44 python3.8 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 45 haproxy2 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 46 collectd available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 47 aws-nitro-enclaves-cli available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 48 m4 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 49 kernel-5.4 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 50 selinux-ng available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 51 php8.0 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 52 tomcat9 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 53 unbound1.13 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 54 mariadb10.5 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 55 kernel-5.10 available [=>stable] |
| 2021-08-14T13:47:15.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 56 redis6 available [=>stable] |

Su an sistem loglarini gonderiyor ayarlama sonrasinda nginx log kayitlarini da gonderecegiz

```
[ec2-user@ip-172-31-54-98 ~]$ cd /etc/awslogs/
[ec2-user@ip-172-31-54-98 awslogs]$ ls
awscli.conf  awslogs.conf  config  proxy.conf
[ec2-user@ip-172-31-54-98 awslogs]$ sudo nano awslogs.conf
```

```
ec2-user@ip-172-31-54-98 awslogs]$ sudo nano awslogs.conf
```

Dosya icerigi

```
GNU nano 2.9.8 awslogs.conf
#-----$
# %S Second as a zero-padded decimal numbers. $
# %f Microsecond as a decimal number, zero-padded on the left. $
# %z UTC offset in the form +HHMM or -HHMM (empty string if the object is naive). $
# %j Day of the year as a zero-padded decimal number. $
# %W Week number of the year (Sunday as the first day of the week) as a zero padded $
# decimal number. All days in a new year preceding the first Sunday are considered $
# to be in week 0. $
# %W Week number of the year (Monday as the first day of the week) as a decimal number. $
# All days in a new year preceding the first Monday are considered to be in week 0. $
# %c Locale's appropriate date and time representation. $
#-----$

[/var/log/messages]
datetime_format = %b %d %H:%M:%S
file = /var/log/messages
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = /var/log/messages
```

```
[/var/log/nginx/access.log]
datetime_format = %b %d %H:%M:%S
file = /var/log/nginx/access.log
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = AccessLog
```

```
[/var/log/nginx/error.log]
datetime_format = %b %d %H:%M:%S
file = /var/log/nginx/error.log
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = ErrorLog
```

```
GNU nano 2.9.8
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = /var/log/messages

[/var/log/nginx/access.log]
datetime_format = %b %d %H:%M:%S
file = /var/log/nginx/access.log
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = AccessLog

[/var/log/nginx/error.log]
datetime_format = %b %d %H:%M:%S
file = /var/log/nginx/error.log
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = ErrorLog
```

Kaydedip cikalim asagidaki komutlari girelim
Once calisan sistemi durdurmak gerekir stop komutu ile

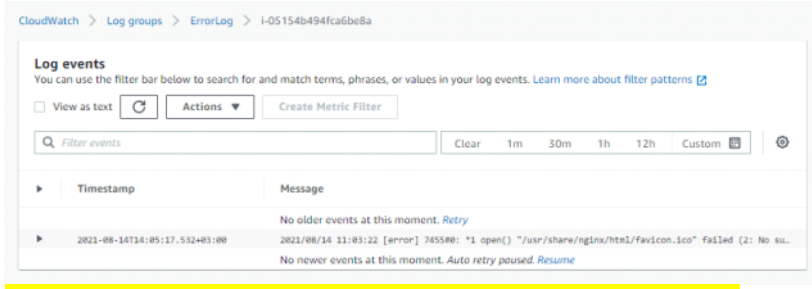
```
sudo systemctl stop awslogsd
sudo systemctl start awslogsd
```

```
[ec2-user@ip-172-31-54-98 awslogs]$ sudo systemctl start awslogsd
[ec2-user@ip-172-31-54-98 awslogs]$ sudo systemctl enable awslogsd.service
```

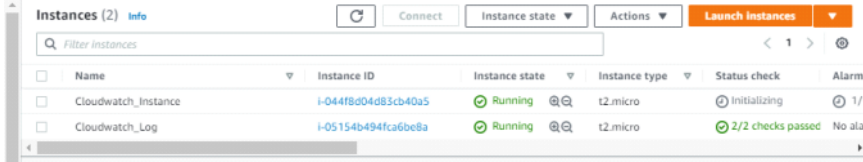
Log kayitlarina geldigini gorebilirsiniz

| | | | | |
|--------------------------|-----------|--------------|---|---|
| <input type="checkbox"/> | AccessLog | Never expire | + | + |
| <input type="checkbox"/> | ErrorLog | Never expire | + | + |

| | |
|-------------------------------|--|
| 2021-08-14T13:47:16.000+03:00 | Aug 14 10:47:16 ip-172-31-54-98 cloud-init: 38 nginx available [-stable] |
|-------------------------------|--|



Bu arada kapali instancenin tekrar calistigini gorebiliriz ayarladigimiz RULES sayesinde



Silinecekler

- Lambda function
- Role durabilir
- Ec2 lari silebiliriz
- Rules lari silebiliriz
- Log gruplarini silebiliriz
- Bugun olusturdugumuz alarmi silebiliriz
- Dashboardin icerisini silebiliriz (3 e kadar ucretsiz)