



- EC2 stands for Elastic Compute Cloud in AWS.

- EC2 is a virtual server that is physically in the AWS Data Center.

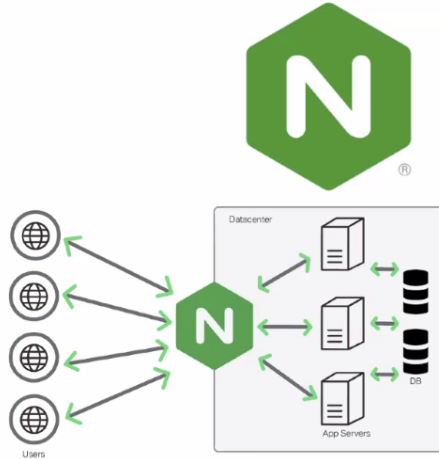
EC2 makinelerimi hangi amaçla kullanmak istiyorsa o amaçla dizayn edip çalıştırabiliriz. NGINX nedir? Bilgisayara yüklediğimiz bir programdır. Server yazılımı olarak geçer.

Bu tür programların amaçları, browserdan gelen requestlerimize serverın cevap vermesini sağlayan yazılımlardır.

What is NGINX?

NGINX, which is pronounced as 'engine ex,' is a widely used open-source web server software. It was a project started by Igor Sysoev in 2002. It got officially released in October 2004. It was created to solve the problem of C10k, which is defined as a challenge to manage the ten thousand connections all at the same time.

CLARUSWAY



(enginex olarak telaffuz ediyoruz.)

Bugün NGINX i statik bir web sayfasını serve edebilmek için kullanacağız.

Apache ilk başta ortaya çıkıyor. Sonra yazılımlar çeşitleniyor. Çünkü internet hızı ve içeriklerin çeşitliliğin artmasıyla, statik içerikten dinamik içeriğe doğru evrilme gerçekleşmesi neticesiyle ihtiyaçlar değişti. İnterneti talep eden kişiler fazlaştı. Bu nedenle Apache gibi diğer yazılımlar gitgide yetersiz kalmaya başladı.

10 binden fazla requesti aynı anda handle edebilecek bir yazılım olarak NGINX 2004 yılında Igor Sysoev tarafından geliştirildi.

Bugüne dek yazdığımız basit program ve fonksiyonları terminallerde yazdık ve sonuçlarını orada aldık. Peki bu basit uygulamaları internetten sunmak istersek programı öncelikle web

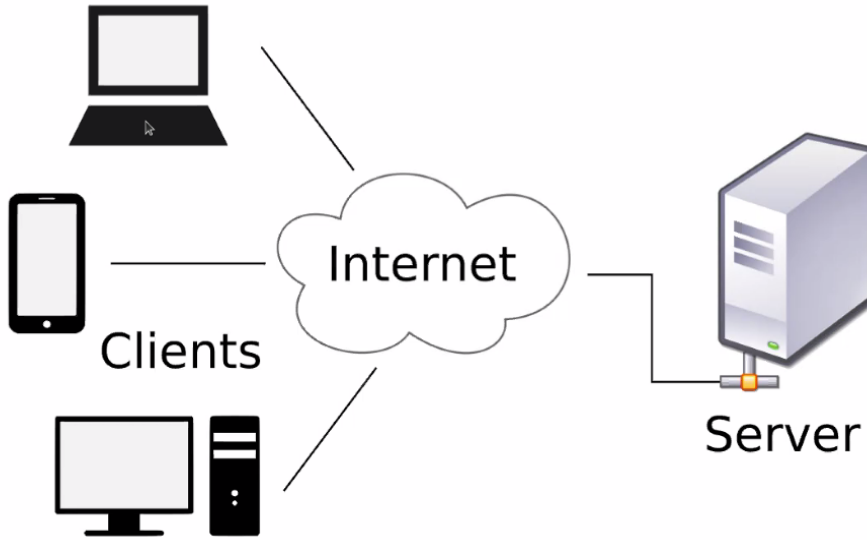
tabanlı olarak yazabilmemiz gerekir. Bunun için programa dilleri için frameworkler var. Python için Flask, Java için Spring gibi.

Bu şekilde yazsak dahi browserımızda adresini yazdığımız zaman sonucu görebilmek için bir cevap almamız gerekiyor. Browser (kullanıcı) ile serverlar arasındaki ilişkiyi http requestlerle gerçekleşiyor. Bu http requestlerin gerçekleşmesi NGINX gibi programlarla gerçekleşir. Request response olarak çalışır.

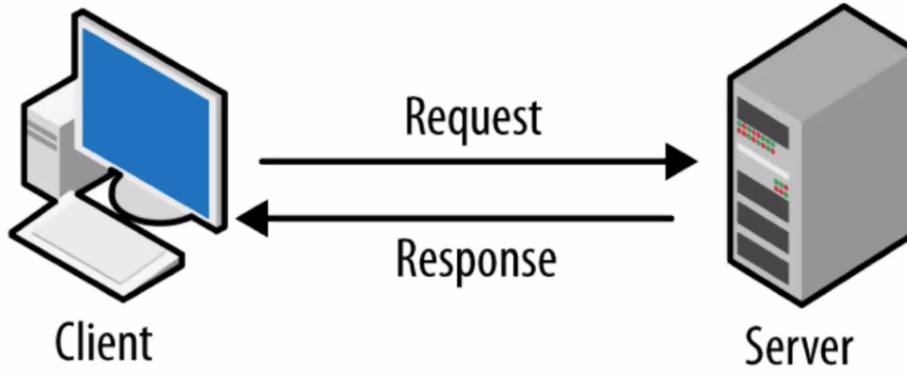
NGINX in statik bir kaynağı browser dan görebilecek şekilde bir setup kurmaya çalışacağız bugün.

Bir server ayağa kaldırıp içine NGINX yükleyeceğiz ve bu server HTTP server haline gelmiş olacak.

Bir server birden fazla görevi de yapar (database server, HTTP server görevlerini aynı anda yapar.) Ancak tek server olursa ve bu servera bir şekilde ulaşılamazsa sitede de ulaşılamaz. Tek server yerine yapı front end, back end ve database olarak üçlü yapıya bölünür. P server görevlerini aynı anda



Bu requestler farklı



HTTP Requesti browser vasıtasıyla gerçekleştiriyoruz. Server de o adreste ne varsa browserımıza yansıtmış oluyor.

EC2 ayağa kaldırırken, image, volume, instance type ve security group belirleyip instance ayağa kaldırıyoruz. Bunları konsoldan yapıyoruz.

Ancak serverlar artacak çünkü ihtiyaçlar çeşitlenecek. Bunları her seferinde konsoldan mı yapcaz. Bunları otomasyon yapcaz. Bizi arandır kılan da otomasyon yapabiliyor olmamız olacak. Bunu launch template ile yapacağız.

EC2 Launch Template

Launch Template EC2 ya ilişkin bir template tir. Virtual makinemizin template ini içinde barındırıyor olacak.

Bir template oluşturmamız demek bir EC2 ayağa kaldırdığımız anlamına gelmiyor.

İhtiyacımız olduğunda oluşturduğumuz o template dosyasına giderek o özelliklerde bir EC2 ayağa kaldıracağız. Yani özellikleri hazır bir şablon olacak.

Kendimiz ayağa kaldırmayacağız. AWS e diyeceğiz ki şu durumlarda şu kadar makine ayağa kaldır, şu durumlarda şu kadarını ayakta tut, şu durumlarda da makine sayısını azalt gibi.

EC2 Launch Template



template ?

Öte yandan bir şirketin hesabını yönettiğimizi düşünelim. Çalışanların hangi template'leri kullanabileceğini kısıtlayabiliriz. Böyle bir özelliği de var.

Bir template i sıfırdan oluşturabileceğimiz gibi başka bir template ten de çıkarabiliriz. Ya da hazır çalışan EC2 nun template ni çıkar da diyebiliriz ve kenarda saklarız.

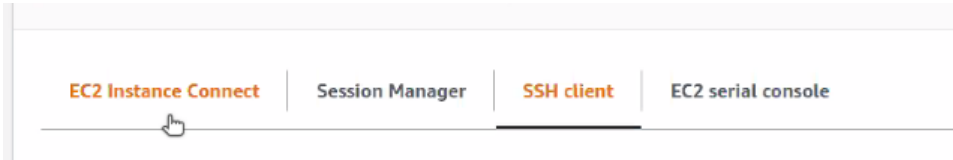
Autoscaling launch configuration da yapılıyor ancak eskide kalan bir yöntem. AWS bizi autoscaling için launch template e yönlendiriyor.

Launch template te autoscaling yaparken versiyonlama yapabiliyoruz. Ancak launch configuration da bu mümkün değil.

Şimdi Amazon Linux EC2 ayağa kaldırıyoruz.

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

HTTPS kullanmamız için bazı sertifikaların yüklü olması gerekiyor. Bunu da ileride göreceğiz.



Soldaki EC2 Instance Connect AWS konsolu üzerinden bağlantı yapmayı sağlar.

HTTP requestlerimiz 80 port üzerinden gönderiliyor.

Hands-on'a geçiyoruz.

2. Sıradaki işlemde:

2. Install the Nginx Web Server.

```
sudo amazon-linux-extras enable nginx1  
sudo yum info nginx --showduplicates  
sudo yum install nginx-1.20.0
```

or

```
sudo amazon-linux-extras install nginx1
```

En üstte yazan amazon-linux-extras amazonun bazı programlar için tuttuğu bir repo gibidir.

```
6 sudo amazon-linux-extras list
```

komutuyla hangi programların available olduğunu görebiliyoruz.

Önce NGINX i enable hale getireceğiz sonra üstteki ya da alttaki yöntemle NGINX i kurabiliyoruz.

--showduplicates bize yükleyebileceğimiz diğer sürümleri gösterir.
Aksi takdirde en son sürümü yükler.

```
Verifying : 1:openssl11-libs-1.1.1g-12.amzn2.0.9.x86_64  
  
Installed:  
  nginx.x86_64 1:1.20.0-2.amzn2.0.5  
  
Dependency Installed:  
  gperftools-libs.x86_64 0:2.6.1-1.amzn2  nginx-filesystem  
  
Complete!  
ec2-user@ip-172-31-81-42:~$
```

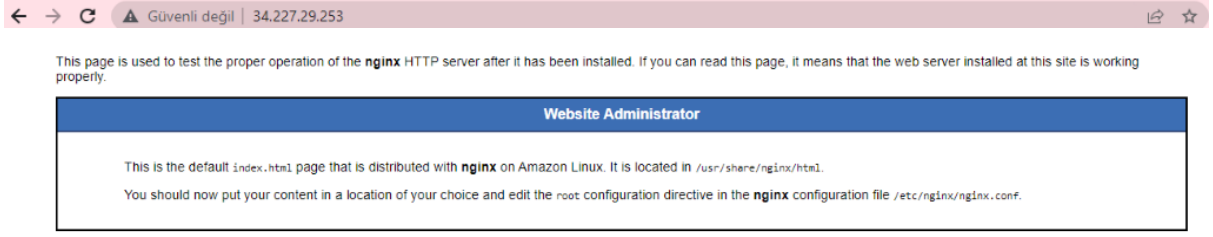
NGINX i yükledik ve sıradaki komutla

```
sudo systemctl start nginx
```

start ediyoruz.

```
sudo systemctl status nginx
```

komutuyla aktif hale geldiğini kontrol ediyoruz. Şimdi Instance ın public ip si web serverda çalışmaya başlayacaktır:



NGINX

index.html page nginx Amazon Linux ta /usr/share/nginx/html. adresindeki default page dir. Bu statik sayfayı bu klasör altındaki index.html den yayınılıyor.

```
71 sudo chmod -R 777 /usr/share/nginx/html
72 ***
73 7. Remove existing `index.html`.
74 ***
75 sudo rm index.html
76 ***
77 8. Upload new `index.html` and `ken.jpg` files with `wget` command. Show the github and explain the RAW .
78 ***
79 wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/index.html
80 wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/ken.jpg
```

bu klasördeki index.html dosyasını sildik ve kendi index.html ve ken.jpg dosyamızı yükledik (komutlar hands-onda) Browser'da sayfayı yeniledik:

Street Fighter - Winner



bazı durumlarda NGINX değişiklikleri hemen işlemeyebilir bu durumda restart etmemiz gerekebilir, eğer web browserda cevabı aldıysak restart etmeye gerek yok:

```
82 9. restart the Nginx Web Server.
83 ~~~
84 sudo systemctl restart nginx
85 ~~~
86 10. configure to start while launching
87 ~~~
88 sudo systemctl enable nginx
89 ~~~
```

NGINX ayarını markette DevOps çular yapar. Bu işleri otomasyona dökmek gerekir, otomasyon söz konusu olduğunda da DevOps çu devreye girer.

Server stop ederse NGINX i tekrar start etmemiz gerekir. Bu nedenle servisi disable dan enable hale getireceğiz ki kendisi start olsun:

```
[ec2-user@ip-172-31-83-108 html]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: d
  isabled)
   Active: active (running) since Sal 2022-12-13 19:28:11 UTC; 22min ago
```

```
💡
sudo systemctl enable nginx
~~~
```

komutuyla enable hale getiriyoruz:

ikinci bir html dosyası ekleyerek web browser da açıyoruz:

```
13. Add another index.html file
💡
echo "Second Page" > /usr/share/nginx/html/index_2.html
~~~

14. add "/index_2.html" at the end of the the public DNS
http://ec2-54-144-132-10.compute-1.amazonaws.com/index_2.html
```

← → ↻ ⚠ Güvenli değil | 34.227.29.253/index_2.html

📦 AWS Management... ➡ Course: AWS 13-TU... 📅 Google Takvim - 13..

Second Page

Şimdi bunları otomasyon ile yapacağız. Yeni bir EC2 ayağa kaldırıp uyguladığımız kodların script hallerini User Data kısmına yazacağız:

User data [Info](#)

```
#!/bin/bash
yum update -y
amazon-linux-extras install nginx1
systemctl start nginx
cd /usr/share/nginx/html
chmod -R 777 /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
systemctl restart nginx
systemctl enable nginx
```

☐ User data has already been base64 encoded

⚠ Güvenli değil | 44.211.174.0

Street Fighter - Winner



ikinci instance ımızda da KEN sayfasını gördük.

ikinci makinamızda istediğimiz sayfanın çalışıp çalışmadığını browser açmadan terminal üzerinden curl komutuyla görebiliriz:

localhostu deneyelim:

```
ec2-user@ip-172-31-81-42:/usr/share/nginx/html$ curl 127.0.0.1
<html>
  <head>
    <title> KEN </title>
  </head>
  <body>
    <center><h1> Street Fighter - Winner</h1><center>
    <center>
  </body>
</html>
```

dns ile deneyelim:

```
ec2-user@ip-172-31-81-42:/usr/share/nginx/html$ curl http://ec2-44-211-174-0.compute-1.amazonaws.com
<html>
  <head>
    <title> KEN </title>
  </head>
  <body>
    <center><h1> Street Fighter - Winner</h1><center>
    <center>
  </body>
</html>
```

Şimdi Launch Templates hands-on a geçiyoruz:

AWS konsoldan launch template sayfasına gidiyoruz ve Create Launch Template diyoruz bu sayfayı görecez :

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '\$', '!', '@'.

Template version description

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto

Summary

- Software Image (AMI)
- Virtual server type (instance type)
- Firewall (security group)
- Storage (volumes)

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the future)

Cancel **Create launch template**

Biz burada ilgili bilgileri gireceiz ve Create Launch Template dediğimizde bize bir şablon oluşturacak.

Burada template isim veriyoruz:

Launch template name and description

Launch template name - *required*

MyClaruswayTemplate

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

origin

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Template tags kısmından template e tag ekleyebiliriz.

Source template de başka bir templateten template oluşturmak içindir.

Bu kısımdan işletim sistemi seçebiliriz.

▼ Instance type [Info](#) [Advanced](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0116 USD per Hour

On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible

▼

[Compare instance types](#)

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

first-key

▼

[Create new key pair](#)

buralarda da instance type ve key.pair i seçiyoruz.

▼ **Network settings** [Info](#)

Subnet Info

Don't include in launch template ▼ [Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Select existing security group ☐ Create security group

Security groups [Info](#)

Select security groups ▼

ders-SSH sg-0b93018bdeac5c768 X
VPC: vpc-07e0f2d1e19abc930

[Compare security group rules](#)

► Advanced network configuration

security grup seçiyoruz.

Resource tag ekliyoruz. (Bu templatı kaynak olarak kullanmak için ileride)
Create LAunch Template diyoruz.

Launch Template ana sayfasına girince listede oluşturduğumuz template i görüyoruz:

EC2 > Launch templates

Launch templates (1) [Info](#) [Actions](#) [Create launch template](#)

Filter by tags or properties or search by keyword

Launch template ID	Launch template name	Default version	Latest version
lt-073dd7d3f3003dded	MyClaruswayTemplate	1	1

Template e tıklayıp giriyoruz:

Sağ tarafındaki her iki Action dan da Launch Instance from template dediğimizde bu template kullanarak instance ayağa kaldırır:

MyClaruswayTemplate (lt-073dd7d3f3003dded) [Actions](#) [Delete template](#)

Launch template details

Launch template ID	Launch template name	Default version	Owner
lt-073dd7d3f3003dded	MyClaruswayTemplate	1	am:aws:iam::976907656638:user/emreaydeniz

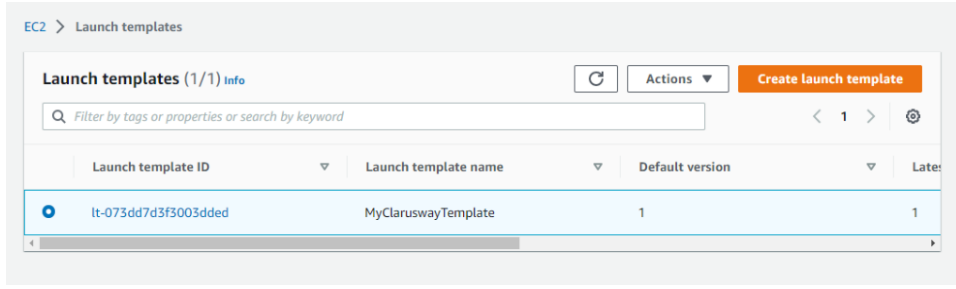
[Details](#) [Versions](#) [Template tags](#)

Launch template version details

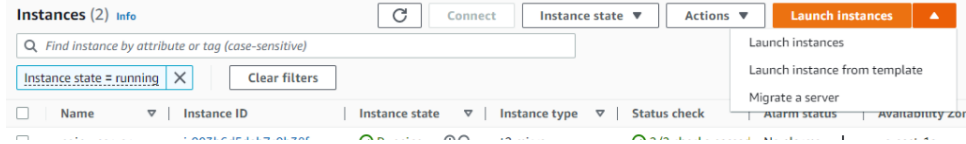
Version	Description	Date created	Actions
1 (Default)	origin	2022-12-13T20:...	Launch instance from template Modify template (Create new version) Set default version

[Delete template version](#)

Launch Templates ana sayfasında da action var:

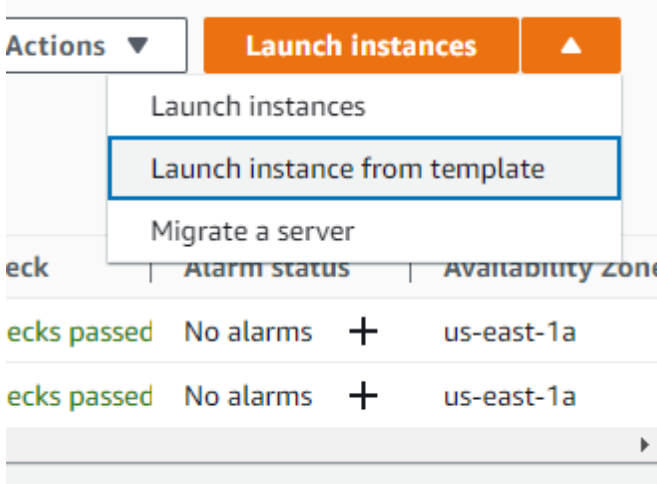


Instances ana sayfasında da template kullanarak instance kaldırma butonu var:



Launch Templates ücretsiz bir hizmet.

Instances sayfasından bu template ile instance kaldırmak için tıklayalım:



Template seçiyorum:

[EC2](#) > [Launch templates](#) > [Launch instance from template](#)

Launch instance from template

Launching from a template allows you to launch from an instance configuration that you would have saved in the past. These saved configurations can be reused and shared with other users to standardize launches across an organisation.

Choose a launch template

Source template

Select a launch template

Q |

MyClaruswayTemplate
ID: lt-073dd7d3f3003dded

Ayarladığımız özellikleri bize gösteriyor, instance kaldırmadan önce bunları burada değiştirebiliriz de:

Burada yaptığımız değişiklikler template versiyonunu değiştirmez sadece oluşturduğumuz bu instance in özelliklerini değiştirir.

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

AMI from catalog

Recents

Quick Start

Amazon Machine Image (AMI)

amzn2-ami-kernel-5.10-hvm-2.0.20221103.3-x86_64-gp2
ami-0b0dcb5067f052a63

Free tier eligible

Verified provider

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Published	Architecture	Virtualization	Root device type	ENA Enabled
2022-11-14T23:11:49.000Z	x86_64	hvm	ebs	Yes

▼ Instance type [Info](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour

Free tier eligible

[Compare instance types](#)

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)
ami-0b0dcb5067f052a63

Virtual server type (instance type)

t2.micro

Firewall (security group)

default-ssh

Storage (volumes)

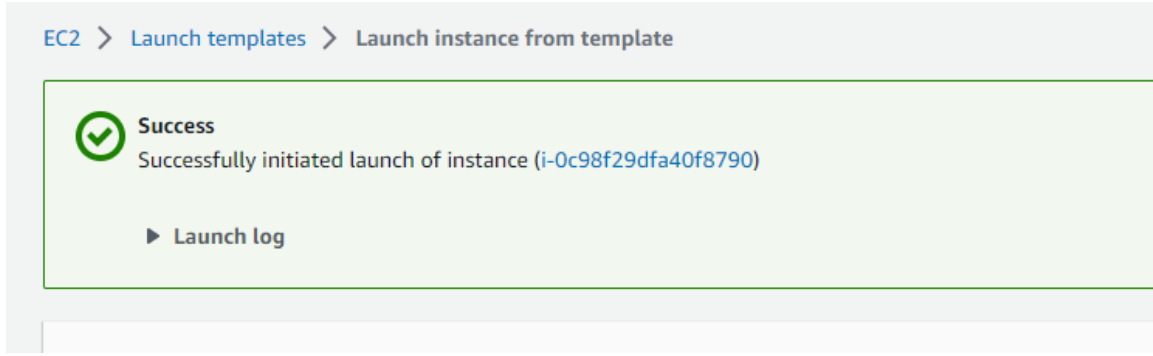
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Launch Instance diyoruz:



Instance ımız oluştu:

The screenshot shows the 'Instances (3)' page in the AWS Management Console. It displays a table with three instances. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability zone. The instances are: 'nginx_server_auto' (Running), 'Webserver-Origin' (Running), and 'nginx-server' (Running). The 'Webserver-Origin' instance is highlighted.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
nginx_server_auto	i-093b6d5deb7c9b38f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a
Webserver-Origin	i-0c98f29dfa40f8790	Running	t2.micro	Initializing	No alarms	us-east-1a
nginx-server	i-054c8072d56f6e59c	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

Resource tag olarak girdiğimiz isim Webserver-Origin olarak instance ismi oldu.

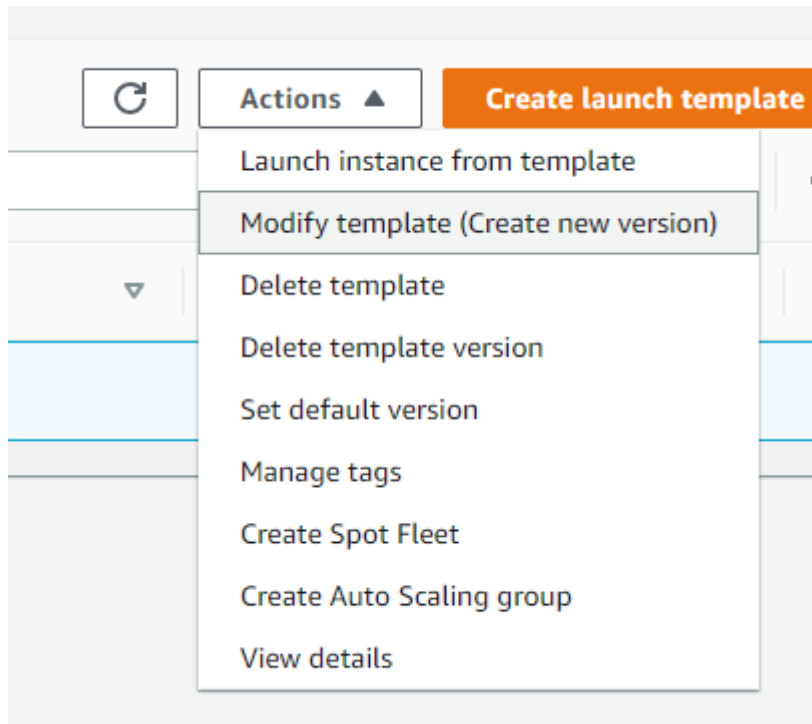
Şimdi version oluşturalım:

Launch Templates ana sayfasında default versionu 1 olarak görüyoruz:

The screenshot shows the 'Launch templates' page in the AWS Management Console. It displays a table with one launch template. The table has columns: Launch template ID, Launch template name, Default version, and Latest version. The launch template is 'lt-073dd7d3f3003dded' with the name 'MyClaruswayTemplate' and default version '1'.

Launch template ID	Launch template name	Default version	Latest version
lt-073dd7d3f3003dded	MyClaruswayTemplate	1	1

Actions tan modify template diyoruz:



description giriyorum:

Launch template name and version description

Launch template name

MyClaruswayTemplate (lt-073dd7d3f3003dded)

Template version description

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Recourse tag a geliyorum:

çünkü diğer özellikleri değiştirmek istemiyorum.

Instance tag ekliyorum:

▼ Resource tags [Info](#)

Key [Info](#)

Value [Info](#)

Resource types [Info](#)

X

X

Select resource ty... ▼

Instances X

Add tag

49 remaining (Up to 50 tags maximum)

Advanced detailsten User Data ya kod script i ekleyecez:

User data [Info](#)

```
#!/bin/bash

yum update -y
amazon-linux-extras install nginx1
systemctl enable nginx
systemctl start nginx
```

☐ User data has already been base64 encoded

ders-SSH

[Storage \(volumes\)](#)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the

Cancel

Create template version

Create Template Version diyoruz.

Template imize tıklayarak özelliklerine bakalım:

The screenshot shows the 'Launch template details' page for 'MyClaruswayTemplate (lt-073dd7d3f3003dded)'. The page has tabs for 'Details', 'Versions', and 'Template tags'. The 'Versions' tab is active, showing a table of two versions. Version 2 is the default version, and Version 1 is the origin.

Version	Default version	Description	Creation time	Created by	AMI ID
2	No	V2 nginx	2022-12-13T20:44:05.0...	arn:aws:iam::976907656638:user/emreaydeniz	ami-0b0dcb5067f052a63
1	Yes	origin	2022-12-13T20:32:03.0...	arn:aws:iam::976907656638:user/emreaydeniz	ami-0b0dcb5067f052a63

Versions sekmesine basınca İkinci versiyonu görüyoruz. Burada versionlardan herhangi birini seçip Action sekmesine tıklarsak burada şu seçenekler çıkıyor:

The screenshot shows the 'Versions (2)' page with the 'Actions' dropdown menu open. The menu options are: 'Launch instance from template', 'Modify template (Create new version)', 'Set default version', 'Create Auto Scaling group', and 'Create Spot Fleet'. The 'Launch instance from template' option is highlighted.

Buradaki versionlardan birini default yapabiliriz, bu versionlardan herhangi birisiyle instance ayağa kaldırabiliriz. Yeni bir version daha oluşturabiliriz.

Version 2 seçiliyken Launch instance from template diyelim. Sayfa açıldığında da version 2 olduğunu görebiliyoruz:

The screenshot shows the 'Launch instance from template' page. It has a heading 'Launch instance from template' and a sub-heading 'Choose a launch template'. Below this, there are two dropdown menus: 'Source template' and 'Source template version'. The 'Source template' dropdown is set to 'MyClaruswayTemplate' and the 'Source template version' dropdown is set to '2'. There are also buttons for 'Refresh' and 'Cancel' next to each dropdown.

Bu sayfada sağ alttan Launch Instance diyoruz.

Instance ıımızı ayağa kaldırdık version 2 yle.

3. bir version oluşturacağız. Versionu anlamak adın description ekledik. Resource tagı değiştireyoruz:

▼ Resource tags Info

Key Info

Value Info

Resource types Info

Q Name X

Q Webserver-V3 X

Select resource ty... X

Instances X

Add tag

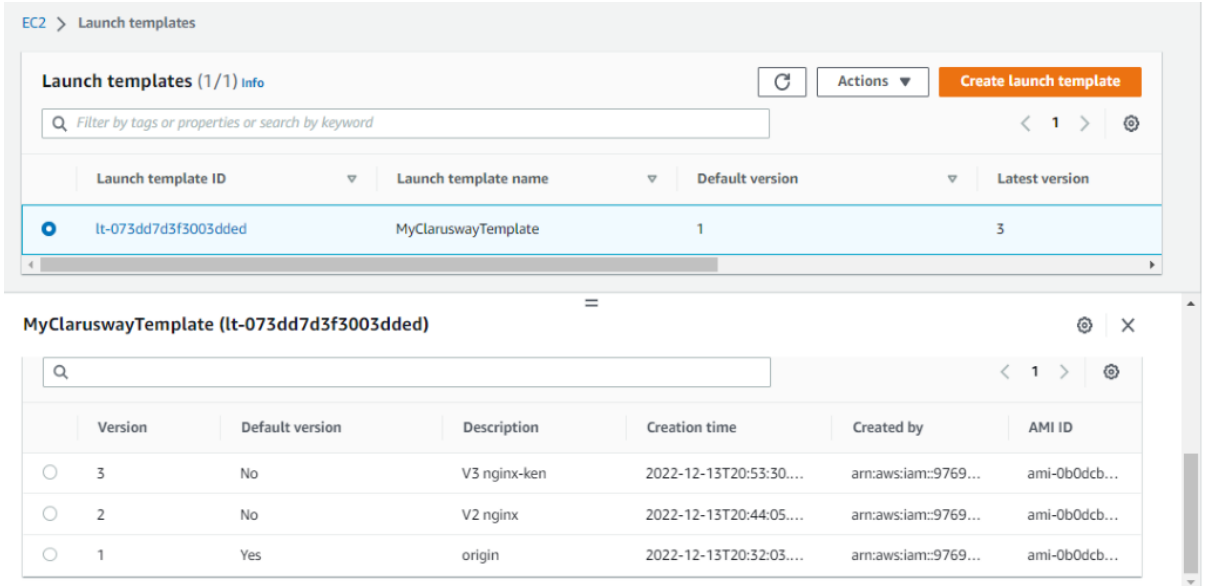
49 remaining (Up to 50 tags maximum)

Bu kez User Data ya şunları yazacağız:

User data Info

```
#!/bin/bash
yum update -y
amazon-linux-extras install nginx1
systemctl start nginx
cd /usr/share/nginx/html
chmod -R 777 /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
systemctl restart nginx
systemctl enable nginx
```

3 tane versionumuz oldu:



Launch templates (1/1) Info

Filter by tags or properties or search by keyword

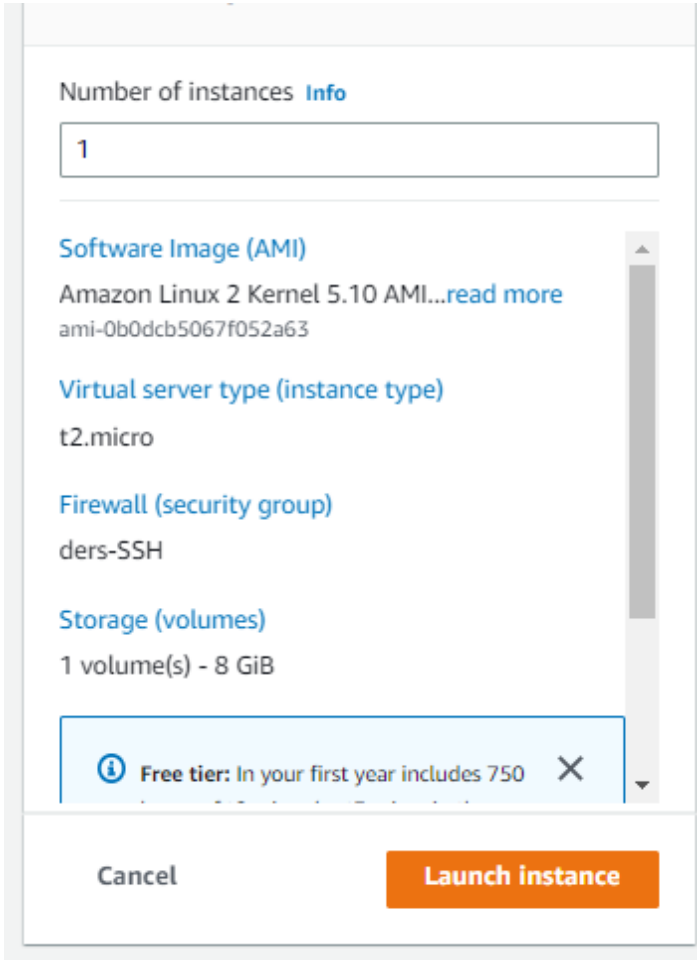
Launch template ID	Launch template name	Default version	Latest version
lt-073dd7d3f3003dded	MyClaruswayTemplate	1	3

MyClaruswayTemplate (lt-073dd7d3f3003dded)

Version	Default version	Description	Creation time	Created by	AMI ID
3	No	V3 nginx-ken	2022-12-13T20:53:30...	arn:aws:iam::9769...	ami-0b0dcb...
2	No	V2 nginx	2022-12-13T20:44:05...	arn:aws:iam::9769...	ami-0b0dcb...
1	Yes	origin	2022-12-13T20:32:03...	arn:aws:iam::9769...	ami-0b0dcb...

1. si boştu. 2.sinde nginx yüklüydü. 3.sü çalıştırmak istediğimiz programı içeriyor.

3. templateten bir instance ayağa kaldırıyoruz.



Number of instances Info

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...read more

ami-0b0dcb5067f052a63

Virtual server type (instance type)

t2.micro

Firewall (security group)

ders-SSH

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750

Cancel Launch instance

sağ taraftaki number of instances tan bu template ile ayağa kaldırmak istediğimiz kadar instance kaldırabiliriz.

2. template ile kaldırdığımız instance in public dns ini web browser a yazalım :

ec2-44-212-41-45.compute-1.amazonaws.com

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

nginx sayfasını gördük.

3.template ile kaldırdığımız instance in public dns ini web browser a yazalım :

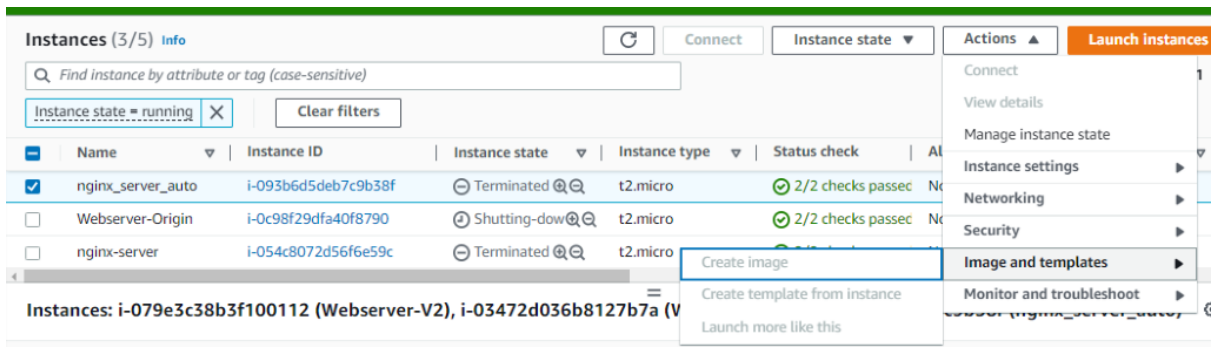
ec2-54-234-164-48.compute-1.amazonaws.com

Street Fighter - Winner



Ayağa kaldırdığımız bir EC2 dan da template oluşturabiliriz. Yani bu EC2 nun özelliklerini template haline getirebiliriz:

Instance sayfasında sağ taraftaki Actions sekmesine bastığımızda:



Name	Instance ID	Instance state	Instance type	Status check
nginx_server_auto	i-093b6d5deb7c9b38f	Terminated	t2.micro	2/2 checks passed
Webserver-Origin	i-0c98f29dfa40f8790	Shutting-down	t2.micro	2/2 checks passed
nginx-server	i-054c8072d56f6e59c	Terminated	t2.micro	2/2 checks passed

Instances: i-079e3c38b3f100112 (Webserver-V2), i-03472d036b8127b7a (V2)

- Create image
- Create template from instance
- Launch more like this

Image and templates in üstüne tıklıyoruz: burada da Create template from this instance a bastığımızda bu instance özelliklerinde bir template oluşturabiliriz

Templateleri silebiliriz:Launch Template sayfasına gelip template kutusu işaretli haldeyken Actions a bastığımızda delete template diyerek siliyoruz.