

Inlämningsuppgift 1

Skapa en robust, säker, skalbar hosting-miljö för en webbapplikation.

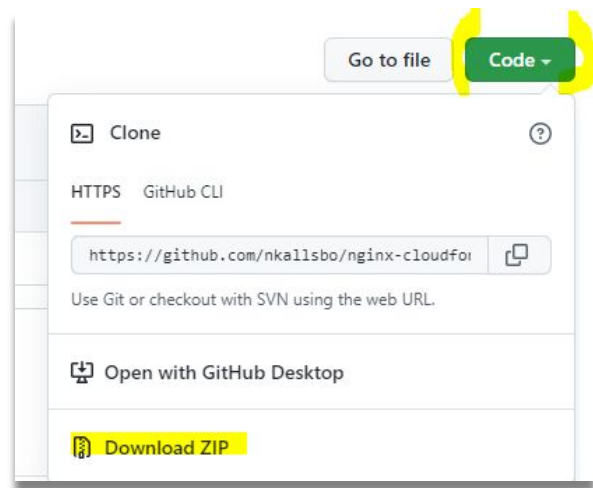
Vi sätter upp en Nginx server med hjälp utav AWS Console och Cloudformation.
Vissa delar är automatiserade för att underlätta.

Vi använder oss utav auto scaling med en lastbalanserare emellan för att säkra vår miljö.

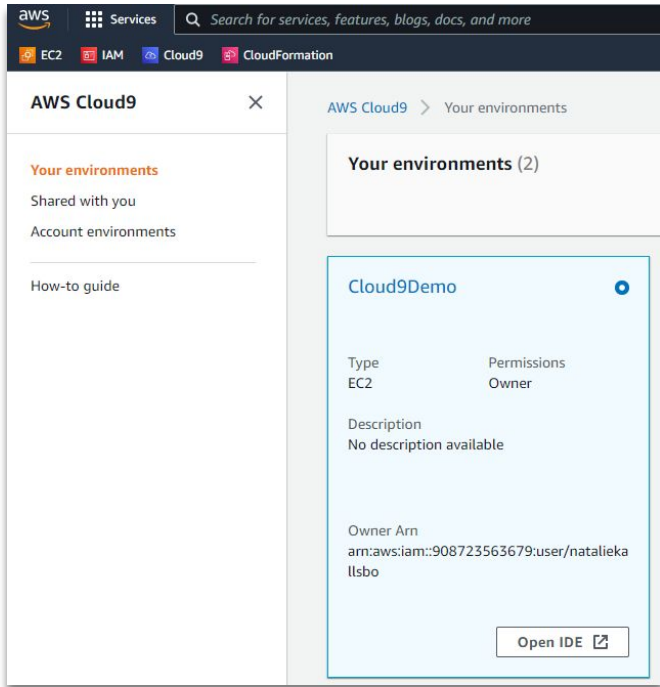
1. Skapa Security Group och Launch Template med Cloudformation

Vi kommer att börja med att ladda ner filer från github. Dessa filer är skrivna med hjälp utav YAML och kommer att användas i CloudFormation. Detta för att ta bort några manuella steg för denna installationen.

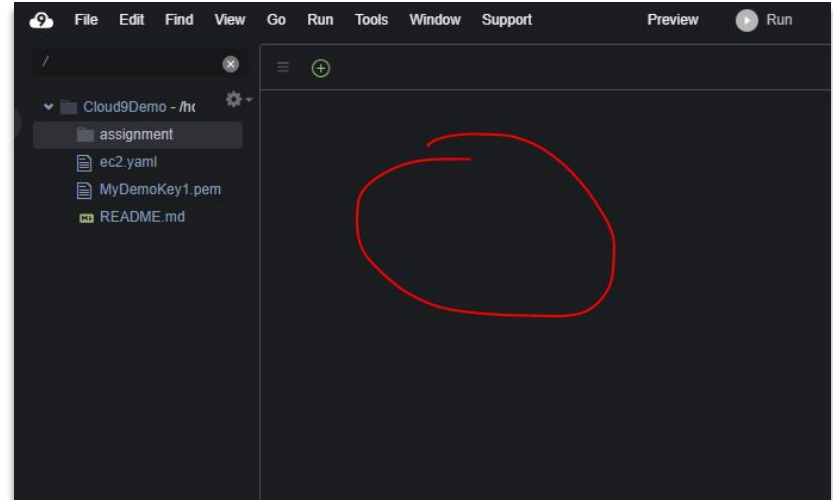
1. Gå till länken - <https://github.com/nkallsbo/nginx-cloudformation>
2. Tryck på **Code** och välj **Download ZIP**. Där finns alla filer som behövs, även denna guide.
3. Navigera till din **Download** mapp (eller där dina nedladdade filer sparas)
Höger klicka på mappen och välj **Extract All**
4. Välj mapp som det skall sparas i och välj **Extract**
5. Öppna mappen för att se alla filer



6. Navigera till Cloud9 och öppna en miljö du kan jobba i



6. Vi skall börja med att flytta över `nginx-part1.yaml` till vår miljö. För över din fil med Drag & Drop. Släpp filen i det markerade området



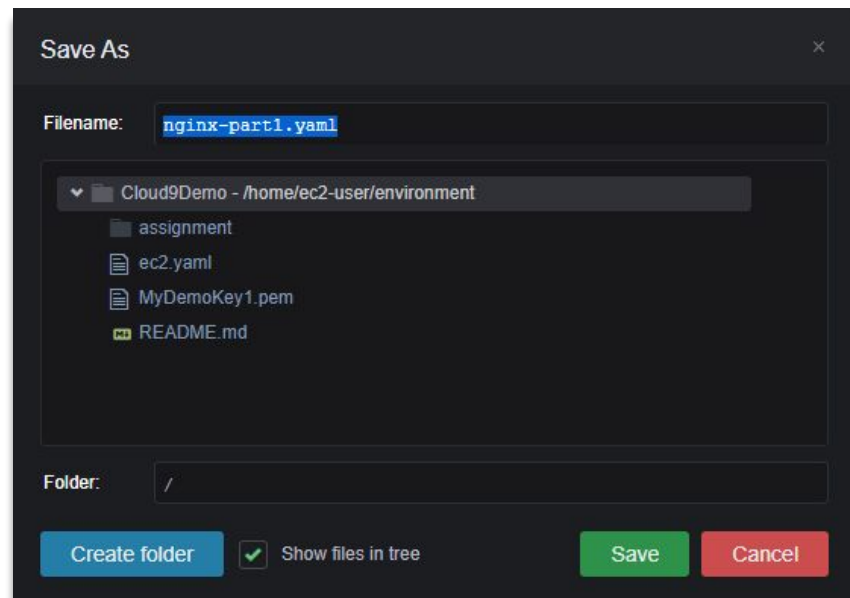
6. Tryck **Ctrl+S** för att spara

7. Tryck på **Save**

Nu är filen uppladdad. Vi skall nu med kommando i terminalen köra vår kod som kommer skapa en stack i CloudFormation

```
---
AWSTemplateFormatVersion: "2010-09-09"
Description: "Creates a security group and a launch template"
Resources:

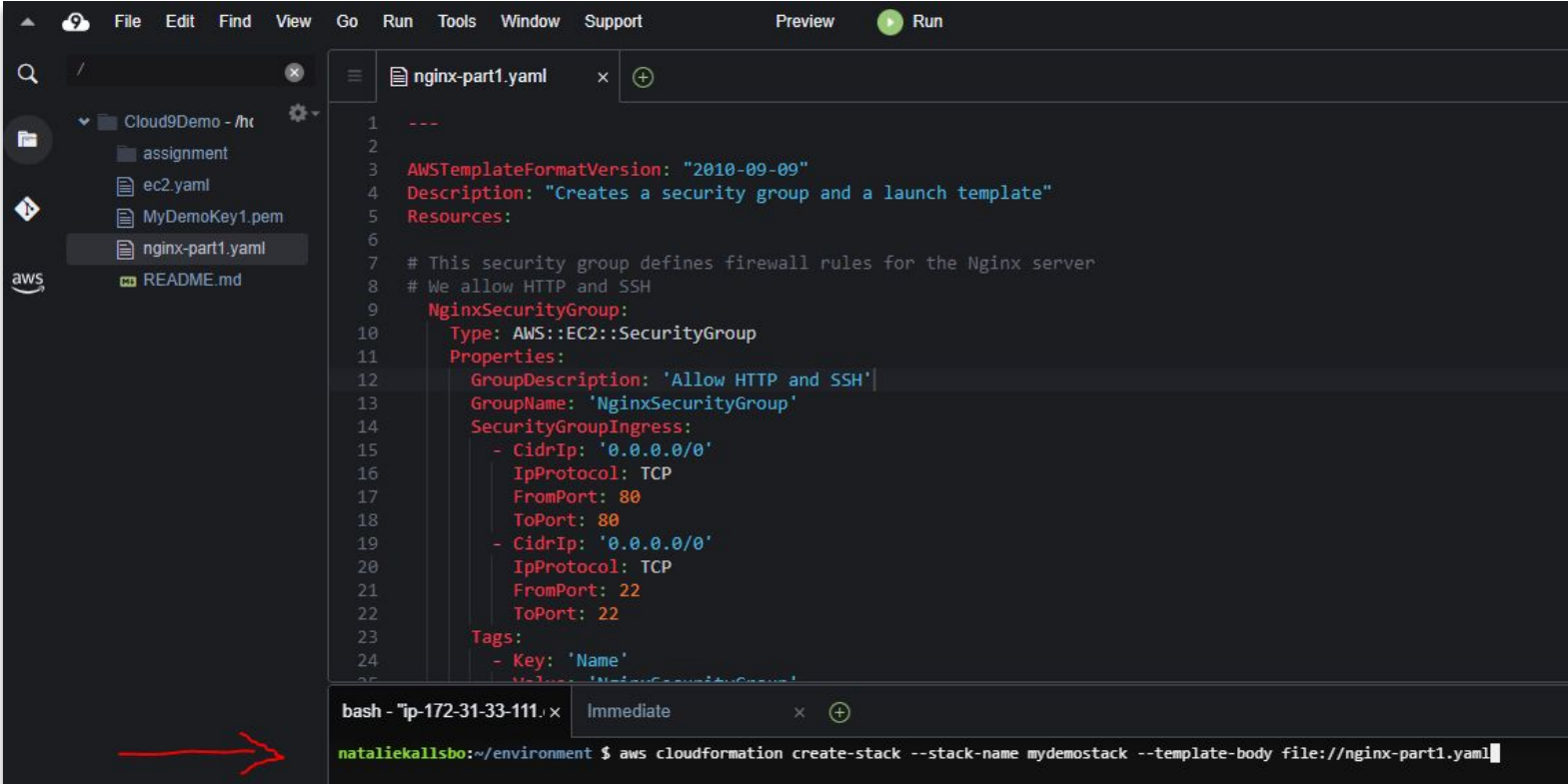
# This security group defines firewall rules for the Nginx server
# We allow HTTP and SSH
  NginxSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: 'Allow HTTP and SSH'
      GroupName: 'NginxSecurityGroup'
      SecurityGroupIngress:
        - CidrIp: '0.0.0.0/0'
          IpProtocol: TCP
          FromPort: 80
          ToPort: 80
        - CidrIp: '0.0.0.0/0'
          IpProtocol: TCP
          FromPort: 22
          ToPort: 22
```



8. Skriv följande kommando i terminalen:

```
aws cloudformation create-stack --stack-name nginxstack --template-body file:///nginx-part1.yaml
```

och tryck **Enter**



The screenshot shows a code editor with a dark theme. The left sidebar displays a file explorer for a project named 'Cloud9Demo - /h...'. The files listed are 'assignment', 'ec2.yaml', 'MyDemoKey1.pem', 'nginx-part1.yaml' (which is selected), and 'README.md'. The main editor area shows the content of 'nginx-part1.yaml', which is an AWS CloudFormation template. The template includes a version, a description, and resources for a security group and a launch template. The terminal window at the bottom shows a bash prompt with the command `aws cloudformation create-stack --stack-name mydemostack --template-body file:///nginx-part1.yaml` entered. A red arrow points to the terminal window.

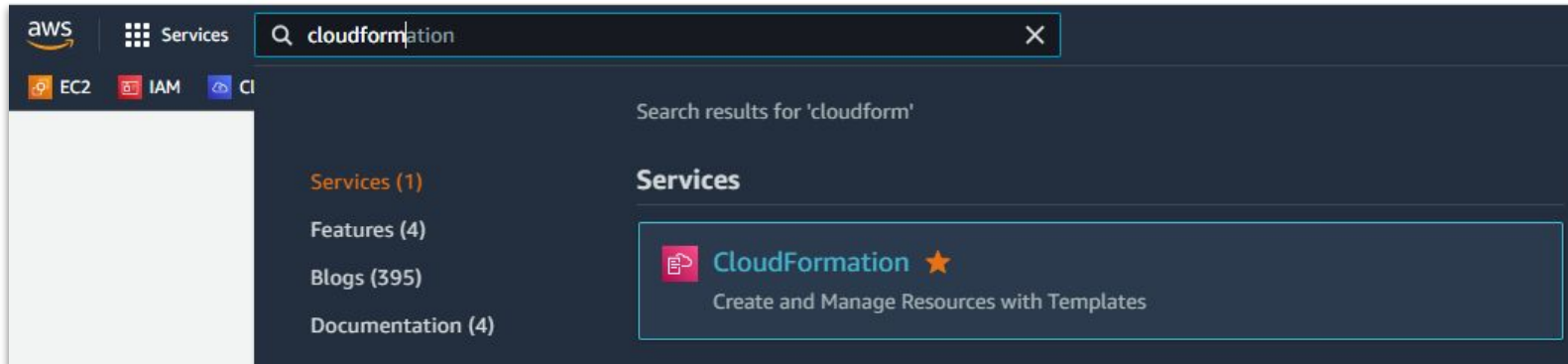
```
1 ---
2
3 AWSTemplateFormatVersion: "2010-09-09"
4 Description: "Creates a security group and a launch template"
5 Resources:
6
7 # This security group defines firewall rules for the Nginx server
8 # We allow HTTP and SSH
9 NginxSecurityGroup:
10   Type: AWS::EC2::SecurityGroup
11   Properties:
12     GroupDescription: 'Allow HTTP and SSH'
13     GroupName: 'NginxSecurityGroup'
14     SecurityGroupIngress:
15       - CidrIp: '0.0.0.0/0'
16         IpProtocol: TCP
17         FromPort: 80
18         ToPort: 80
19       - CidrIp: '0.0.0.0/0'
20         IpProtocol: TCP
21         FromPort: 22
22         ToPort: 22
23   Tags:
24     - Key: 'Name'
25       Value: 'NginxSecurityGroup'
```

```
bash - "ip-172-31-33-111" x Immediate x +
nataliekallsbo:~/environment $ aws cloudformation create-stack --stack-name mydemostack --template-body file:///nginx-part1.yaml
```

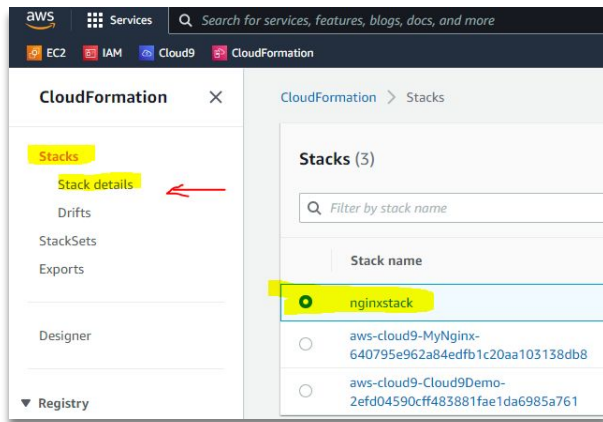
Du får en output med ditt stack id. För att kontrollera att allt gick bra kan vi gå in i Cloudformation

```
nataliekallsbo:~/environment $ aws cloudformation create-stack --stack-name mydemostack --template-body file://nginx-part1.yaml
{
  "StackId": "arn:aws:cloudformation:eu-west-1:908723563679:stack/mydemostack/d37d01f0-d38c-11ec-8522-0ab8e1c4b7d5"
}
nataliekallsbo:~/environment $
```

9. Om du inte redan har det sparat som favorit, sök på CloudFormation och Välj



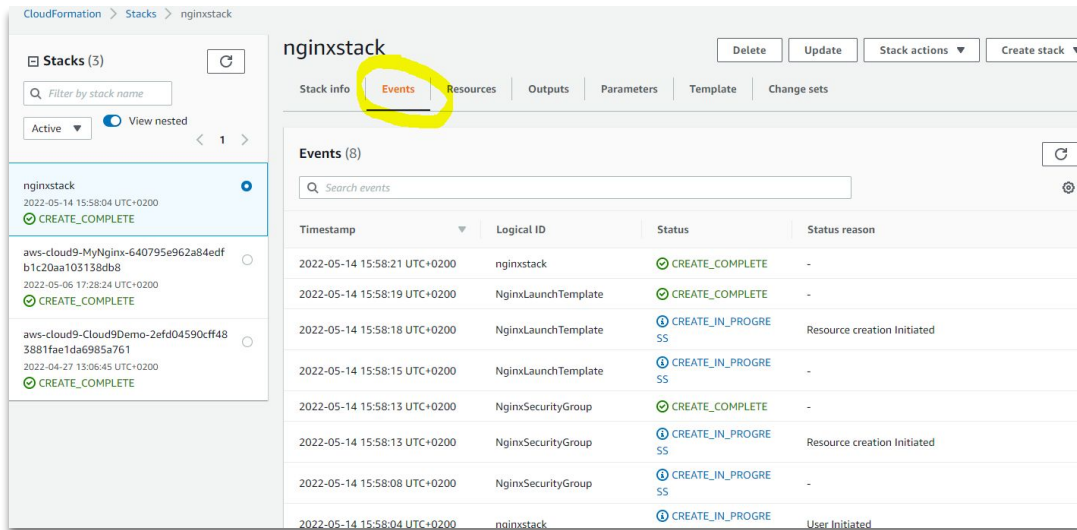
10. Markera **nginxstack** som vi precis skapat
11. Välj sedan **Stack details**



12. Tryck på events
Här kan du nu se att allt har gått igenom.
NginxLaunchTemplate och **NginxSecurityGroup** har skapats.

Om man vill kontrollera att dem faktiskt finns kan man gå in i navigeringsfönstret, under Load Balancing, välj **Load Balancers**.

Eller i navigeringsfönstret, under Network & Security, välj **Security Groups**.



2. Skapa "Key Pair"

1. I navigeringsfönstret, under Network & Security, välj **Key Pairs**.
2. Välj **Skapa nyckelpar**.
3. För **Namn** ange **NginxKey** till nyckel paret.
4. Fyll i enligt bilden nedan och tryck "Create key pair"

Create key pair [Info](#)

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

NginxKey

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)

☒ RSA

☐ ED25519

Private key file format

☒ .pem

For use with OpenSSH

☐ .ppk

For use with PuTTY

Tags (Optional)

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel

Create key pair

5. En fil kommer att laddas ner med din privata nyckel. Spara denna på ett säkert men åtkomligt ställe.



NginxKey.pem



3. Skapa “Load Balancer” och “Target Group”

Lastbalanseraren distribuerar inkommande HTTP- och HTTPS-trafik över flera mål, såsom Amazon EC2-instanser, mikrotjänster och “containers”, baserat på begäran attribut. Din kund gör en förfrågan till din applikation. Sedan får “the listeners” i din lastbalanserare förfrågningar som matchar protokollet och porten som du konfigurerar. Den mottagande lyssnaren utvärderar den inkommande förfrågan mot de regler du anger, och om tillämpligt dirigerar den förfrågan till lämplig “target group”. Trafiken dirigeras baserat på lastbalansering algoritmen och de routing regler som anges i “the listener”.

1. I navigeringsfönstret, under Load Balancing, välj **Load Balancers**.
2. Välj **Create Load Balancer**.
3. För “Load balancer types” väljer du **Application Load Balancer** genom att trycka på **Create**.

4. För Namn ange **NginxLoadBalancer**.

5. Fyll i enligt nedan

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

NginxLoadBalancer

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)

Scheme cannot be changed after the load balancer is created.

☒ Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#) [↗](#)

☐ Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)

Select the type of IP addresses that your subnets use.

☒ IPv4

Recommended for internal load balancers.

☐ Dualstack

Includes IPv4 and IPv6 addresses.

6. Bocka i enligt nedan för att välja olika “subnets” i olika “Availability Zones”

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after you confirm the VPC for your targets, view your [target groups](#).

-
vpc-0a6772e33f5905273
IPv4: 172.31.0.0/16



Mappings [Info](#)

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic to the selected Availability Zones. Zones that are not supported by the load balancer or VPC cannot be selected. Subnets can be added, but not removed, once a load balancer is created.

☒ eu-west-1a

Subnet
subnet-0a5c6fc92d9be4cf3

IPv4 settings
Assigned by AWS

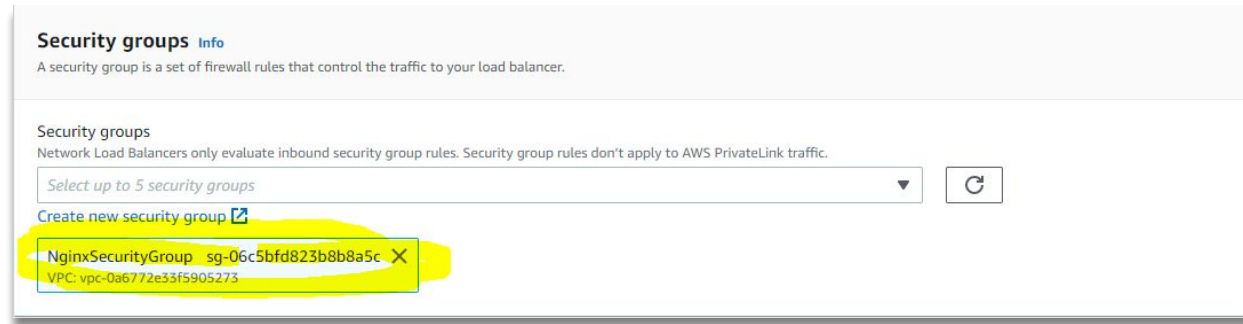
☒ eu-west-1b

Subnet
subnet-09cc5b35d80ff355a

IPv4 settings
Assigned by AWS

☐ eu-west-1c

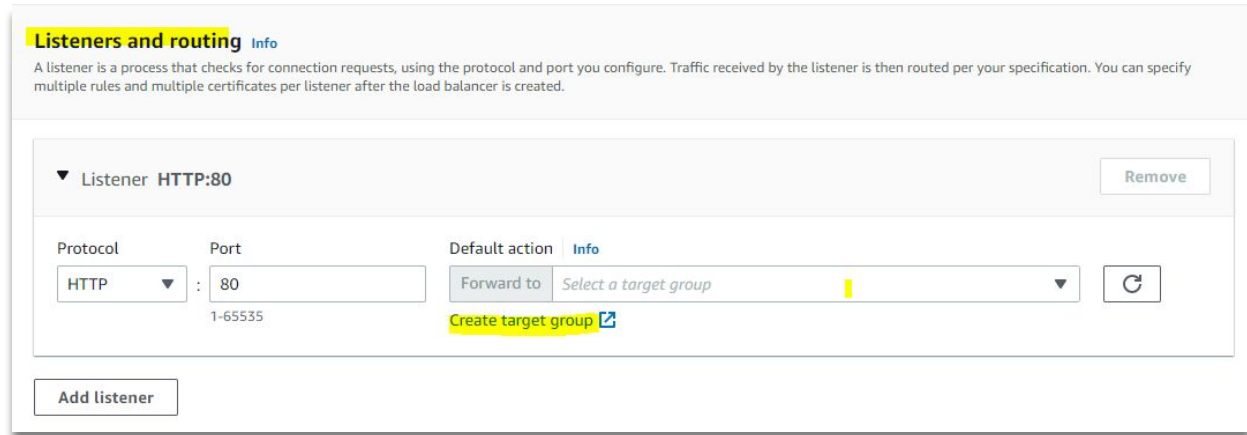
7. I Security groups väljer vi **NginxSecurityGroup** som skapades med hjälp utav CloudFormation



“The listener” är en process som söker efter anslutningsförfrågningar, med hjälp av protokollet och porten du konfigurerar. Trafik som tas emot av lyssnaren dirigeras sedan enligt din specifikation. Du kan ange flera regler och flera certifikat per lyssnare efter att lastbalanseraren har skapats.

Vi behöver skapa en Listener, och till detta behöver vi även en “Target Group”

7. Tryck på **Create target group**
(Ett nytt fönster kommer öppnas)



8. Välj här **Instances**

Detta för att vi använder en EC2 som backend.

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.


Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type



Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#)  to manage and scale your EC2 capacity.



IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.



Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.



Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

9. För **Namn** ange **NginxALBTargetGroup**.

10. Tryck sedan på **Next**

11. Välj **Create target group**

12. Gå tillbaka till fönstret för att skapa lastbalanseraren och fortsätt konfigurationen.

13. Tryck på uppdatera pilen

14. Välj **NginxALBTargetGroup** i listan

15. Välj sedan **Create load balancer**

Target group name

NginxALBTargetGroup

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol Port

TCP : 80

VPC

Select the VPC with the instances that you want to include in the target group.

-
vpc-0a6772e33f5905273
IPv4: 172.31.0.0/16

Listeners and routing [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener HTTP:80 Remove

Protocol Port Default action [Info](#)

HTTP : 80 Forward to Select a target group

1-65535

Create target

NginxALBTargetGroup HTTP

Target type: Instance, IPv4

Add listener

4. Skapa Autos Scaling Group

1. I navigeringsfönstret, under Auto Scaling, välj **Auto Scaling Groups**.
2. Välj **Skapa Create Auto Scaling group**.
3. För **Namn** ange **NginxASG**.

Med hjälp utav Cloudformation har vi sedan tidigare skapat vår Launch Template.

4. I listan välj **NginxLaunchTemplate**
5. Välj sedan **Next**

Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.



Name


Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.



Launch template [Info](#) [Switch to launch configuration](#)


Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.


 

[Create a launch template](#) 

Version

[Create a launch template version](#) 

Description	Launch template	Instance type
-	NginxLaunchTemplate  lt-03fd1f0e39605a2d1	t2.micro

6. Välj de två "Availability Zones" enligt bilden
7. Tryck på **Next**

Choose instance launch options [Info](#)

Choose the VPC network environment that your instances are launched into, and customize the instance type options.

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across multiple Availability Zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0a6772e33f5905273
172.31.0.0/16 Default



[Create a VPC](#)

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets



eu-west-1a | subnet-0a5c6fc92d9be4cf3 X
172.31.16.0/20 Default

eu-west-1b | subnet-09cc5b35d80ff355a X
172.31.32.0/20 Default

[Create a subnet](#)

I detta steg skall lastbalanseraren konfigureras.

8. Välj enligt bilden

Vi väljer en lastbalanserare som redan existerar, den vi precis skapat. Samma sak med vår "Target Group" för "The Listener"

9. Tryck på **Next**

Configure advanced options [Info](#)

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Load balancing - optional [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.



No load balancer

Traffic to your Auto Scaling group will not be fronted by a load balancer.



Attach to an existing load balancer

Choose from your existing load balancers.



Attach to a new load balancer

Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.



Choose from your load balancer target groups

This option allows you to attach Application, Network, or Gateway Load Balancers.



Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups



NgixALBTargetGroup | HTTP



Application Load Balancer: NgixLoadBalancer

Här väljer vi hur många instanser vi önskar, minimum och max. Så om det kommer mycket trafik på till vår applikation så kommer vår Auto Scaling group max starta 3 instanser. Och vi kommer alltid att ha minst 1 instans igång.

10. Fyll i enligt bilden

11. Tryck på **Next**

12. Vi kommer här till “Add notifications” vi hoppar över det i detta exempel.

Configure group size and scaling policies [Info](#)

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

Group size - *optional* [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

Minimum capacity

Maximum capacity

Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☐

Target tracking scaling policy

Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☒

None

13. För att våra insatser skall få ett namn som är överskådligt vill vi lägga till ett. Detta gör vi under "tags" - Fyll i enligt bild.

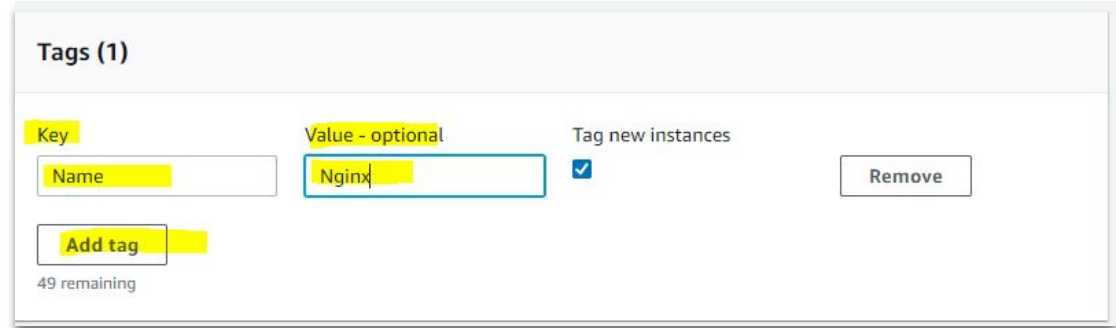
14. Tryck **Next**

15. Välj **Create Auto Scaling Group**

Om man nu går till **Instances** kan man se att en ny instans skapas med namnet **Nginx**. Skulle belastningen bli hög så kommer en till instans att skapas med hjälp utav vår **Auto Scaling Group**.

16. Markera instansen

17. Kopiera IP adressen



Tags (1)

Key: Name

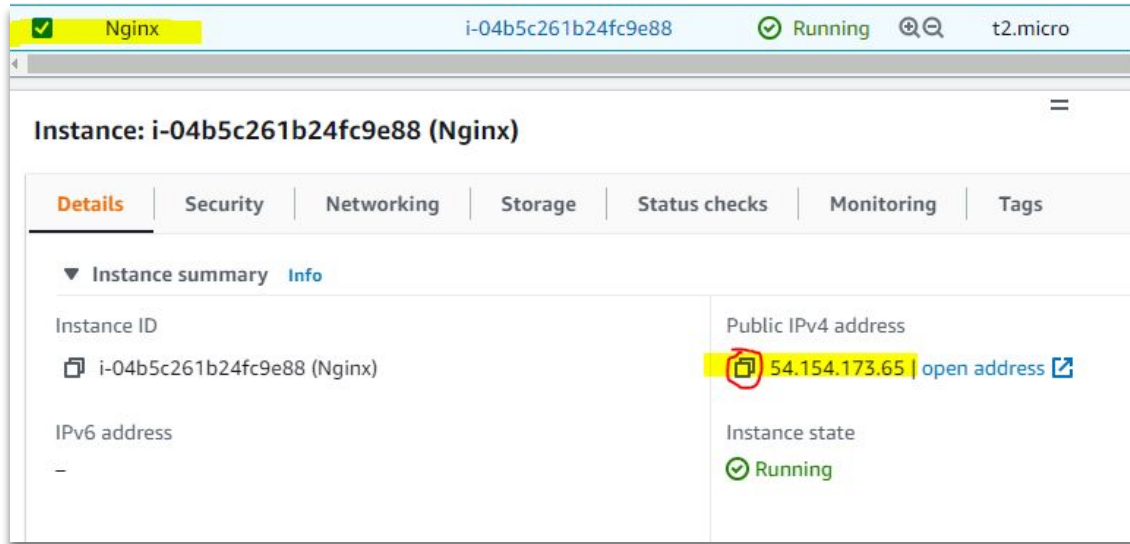
Value - optional: Nginx

Tag new instances: ☒

Remove

Add tag

49 remaining



Instances

Instance: i-04b5c261b24fc9e88 (Nginx)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

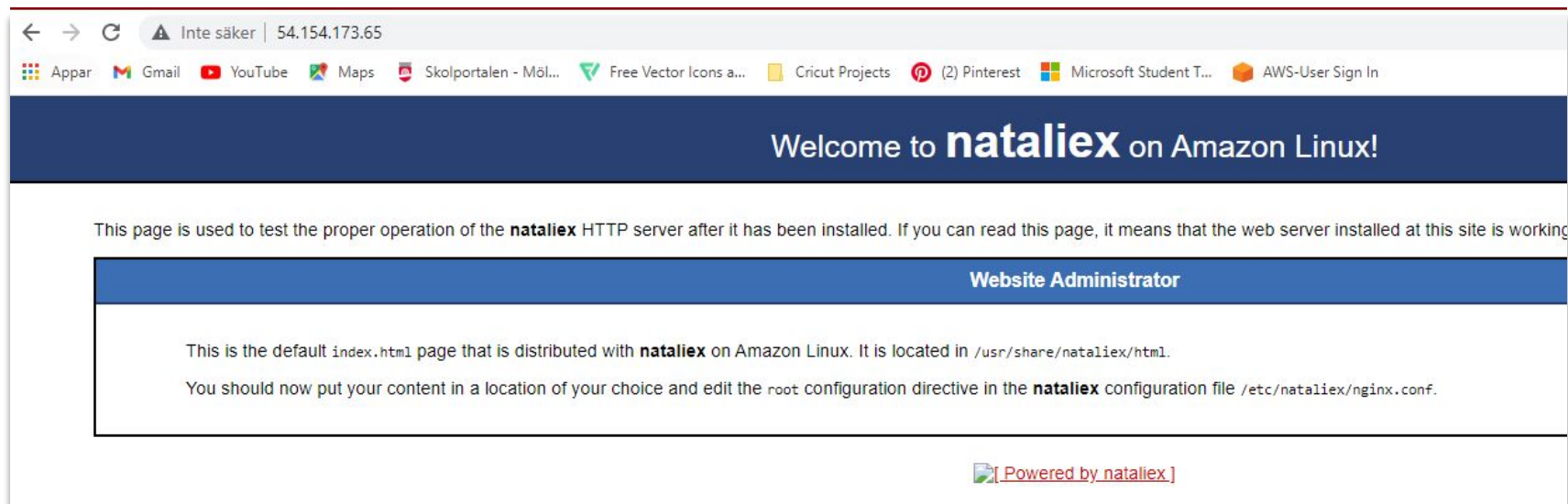
Instance summary Info

Instance ID: i-04b5c261b24fc9e88 (Nginx)

Public IPv4 address: 54.154.173.65 | open address

Instance state: Running

18. Kopiera in adressen i din browser och gå till den.



Här är vår Nginx sida. Som man kan se så står det "nataliex" lite varstans. Detta är för att jag har i ett bash script som installerar och startar Nginx även lagt till att alla ord där det står nginx skall bytas ut till nataliex

I CloudFormation har vi ett bash skript inlagt under "user data". Detta gör att när instansen startas upp så uppdateras maskinen > installerar nginx > startar nginx

Men sedan så navigerar jag till index.html filen som skapas när nginx installeras.

med hjälp utav kommandot:

```
sed -i 's/nginx/nataliex/'  
index.html
```

byts **nginx** ut mot **nataliex**

```
nginx-part1.yaml  
README.md  
19 - CidrIp: '0.0.0.0/0'  
20 IpProtocol: TCP  
21 FromPort: 22  
22 ToPort: 22  
23 Tags:  
24 - Key: 'Name'  
25 Value: 'NginxSecurityGroup'  
26  
27 # Creates a launch template  
28 NginxLaunchTemplate:  
29 Type: AWS::EC2::LaunchTemplate  
30 Properties:  
31 LaunchTemplateName: 'NginxLaunchTemplate'  
32 LaunchTemplateData:  
33 ImageId: "ami-0c1bc246476a5572b"  
34 InstanceType: "t2.micro"  
35 UserData:  
36 Fn::Base64: |  
37 #!/bin/bash  
38 yum update -y  
39 amazon-linux-extras install nginx1 -y  
40 systemctl start nginx  
41 systemctl enable nginx  
42 cd /usr/share/nginx/html  
43 sed -i 's/nginx/nataliex/' index.html  
44 SecurityGroups:  
45 - Ref: 'NginxSecurityGroup'  
46
```

Vi kan också finna detta efter att vår Launch Template skapats.

19. I navigeringsfönstret, under Instances, välj **Launch Templates**.

20. Markera **NginxLaunchTemplate**

21. Tryck på **Advanced details**

The screenshot displays the AWS Management Console interface for a Launch Template. At the top, a table lists the template with ID 'lt-0847d9b0b1bb3e0be', name 'NginxLaunchTemplate', and version '1'. Below this, a summary section provides key details: Launch template ID, Launch template name, Default version, and Owner. The 'Details' tab is selected, showing 'Launch template version details'. This section includes a dropdown for the version (set to '1 (Default)'), a description, the date created, and the user who created it. At the bottom, a navigation bar allows switching between different views, with 'Advanced details' currently selected and highlighted.

Launch template ID	Launch template name	Default version	Owner
lt-0847d9b0b1bb3e0be	NginxLaunchTemplate	1	arn:aws:iam::908723563679:user:nataliekallsbo

Launch template version details

Version	Description	Date created	Created by
1 (Default)	-	2022-05-14T14:42:55.000Z	arn:aws:iam::908723563679:user:nataliekallsbo

Navigation: Instance details | Storage | Resource tags | Network interfaces | **Advanced details**

22. Scrolla längst ned

Detta kan man om man vill ändra i efterhand.
Kanske skapa flera versioner utav det.



```
User data
❏

#!/bin/bash
yum update -y
amazon-linux-extras install nginx1 -y
systemctl start nginx
systemctl enable nginx
cd /usr/share/nginx/html
sed -i 's/nginx/nataliex/' index.html

Base64-encoded user data has been decoded for readability.
```

Vi har i detta exempel skapat en robust och säker miljö för en webbapplikation.

Denna kommer under tid att utvecklas och uppdateras så att allt blir automatiserat så långt det går.