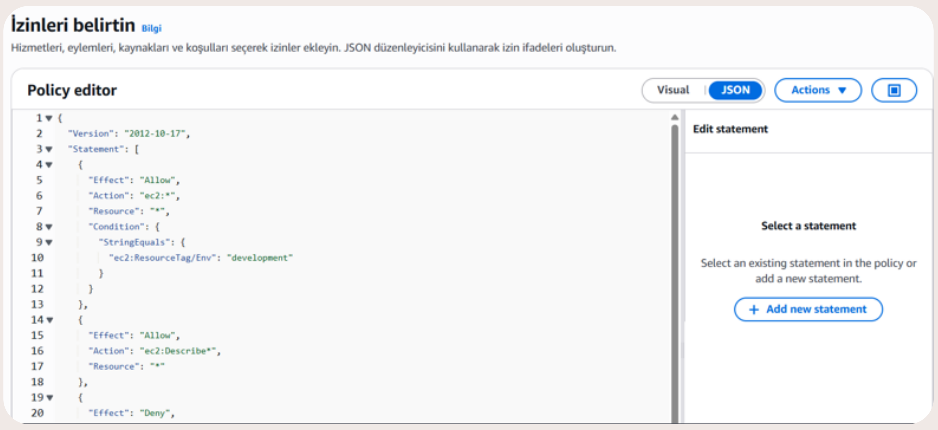
CLOUD SECURITY WITH AWS IAM



İzinleri belirtin

Hizmetleri, eylemleri, kaynakları ve koşulları seçerek izinler ekleyin. JSON düzenleyicisini kullanarak izin ifadeleri oluşturun.

Policy editor

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

Introducing Today's Project!

Project overview

In this project, I will demonstrate how AWS Identity and Access Management (IAM) is used to control authentication and authorization within an AWS account. I’m doing this project to gain hands-on experience with AWS security fundamentals.

Tools and concepts

In this project, I learned several key AWS services and core cloud security concepts:

AWS Identity and Access Management (IAM):
Creating users, user groups, policies, and managing authentication and authorization.

IAM Policies (JSON):
Understanding Effect, Action, and Resource, and how to control access using conditions based on tags.

Amazon EC2:
Launching and managing EC2 instances to scale computing power.

EC2 Tags:
Using tags to organize resources, separate environments (development vs production), and enforce access control.

IAM User Groups:
Managing permissions at scale by assigning policies to groups instead of individual users.

Account Alias:
Simplifying AWS login with a human-readable account identifier.

IAM Policy Simulator:
Testing and validating permissions safely without impacting real AWS resources.

Project reflection

This project took me approximately 1,5 hours.

Tags

What I did in this step

In this step, I will increase NextWork's computing capacity by launching additional servers in the cloud. By completing this step, NextWork gains the ability to handle more users simultaneously, improving performance and reliability during peak usage times.

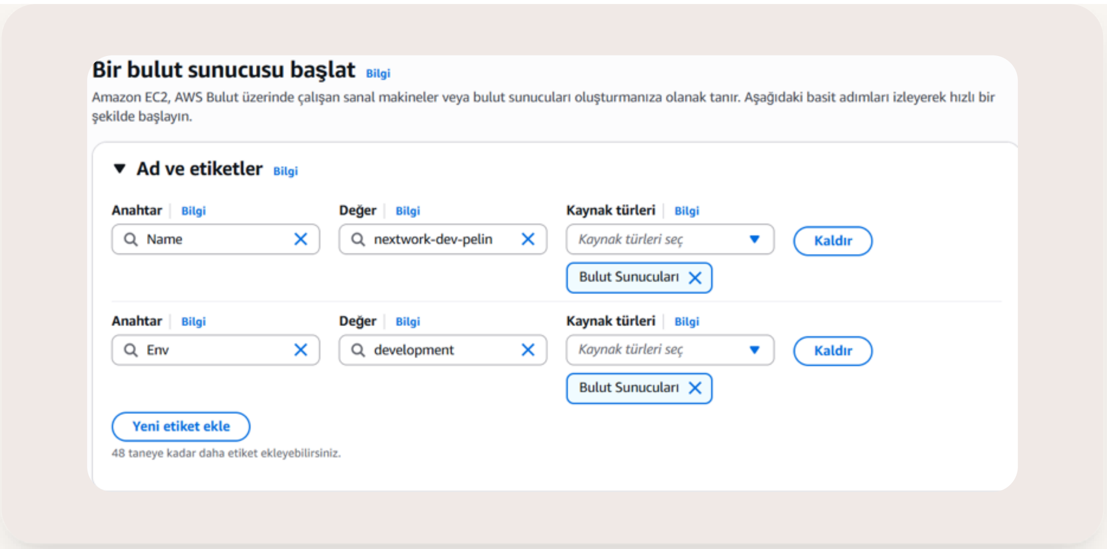
Understanding tags

Tags in Amazon EC2 are key–value pairs that you can assign to resources such as instances, volumes, and snapshots.

They are used to organize, identify, and manage resources in an AWS environment, especially when you have many resources running at the same time.

My tag configuration

I assigned tags to distinguish the purpose of each EC2 instance.One EC2 instance is tagged as Development.The other EC2 instance is tagged as Production. For both instances, the tag values were left as the default.


Bir bulut sunucusu başlat Bilgi

Amazon EC2, AWS Bulut üzerinde çalışan sanal makineler veya bulut sunucuları oluşturmanıza olanak tanır. Aşağıdaki basit adımları izleyerek hızlı bir şekilde başlayın.

▼ Ad ve etiketler Bilgi

Anahtar	Değer	Kaynak türleri	
Name	nextwork-dev-pelin	Bulut Sunucuları	Kaldır
Env	development	Bulut Sunucuları	Kaldır

Yeni etiket ekle

48 taneye kadar daha etiket ekleyebilirsiniz.

IAM Policies

What I did in this step

In this step, we are creating an IAM policy that controls access to the development EC2 instance. Specifically, we are:
Defining permissions that allow actions (such as viewing or managing EC2 resources)
Restricting access so that only the development instance can be accessed.Using this policy to ensure that users or groups can work on the development environment without affecting the production instance.

Understanding IAM policies

IAM policies define what actions are allowed or denied for a user, group, or role in AWS.

The policy I set up

For this project, I've set up a policy using JSON.

Policy effect

The effect of this policy is to allow limited EC2 access only to development instances while protecting resource tags. As a result, users can safely manage the development environment without affecting production resources, and they cannot modify tags to bypass access restrictions. This helps enforce security, consistency, and the principle of least privilege.

Understanding Effect, Action, and Resource

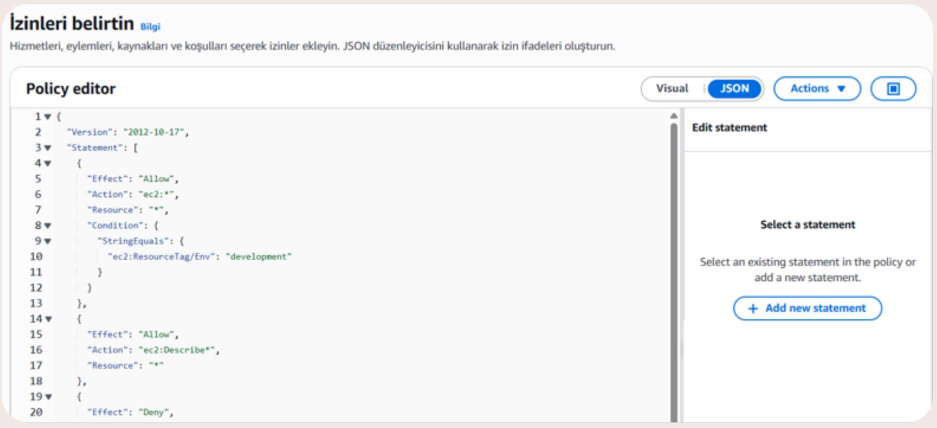
In an IAM JSON policy, Effect, Action, and Resource are the core elements that define permissions.

Effect
Specifies whether the policy allows or denies the listed actions.
Common values are Allow or Deny.

Action
Defines what actions are being controlled.
For example, ec2:StartInstances, ec2:StopInstances, or ec2:DescribeInstances.

Resource
Specifies which AWS resources the actions apply to.
This can be a specific resource (like a particular EC2 instance ARN) or all resources (*).

My JSON Policy



İzinleri belirtin

Hizmetleri, eylemleri, kaynakları ve koşulları seçerek izinler ekleyin. JSON düzenleyicisini kullanarak izin ifadeleri oluşturun.

Policy editor

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

Account Alias

What I did in this step

In this step, we are creating an AWS Account Alias to make logging in easier and more user-friendly.

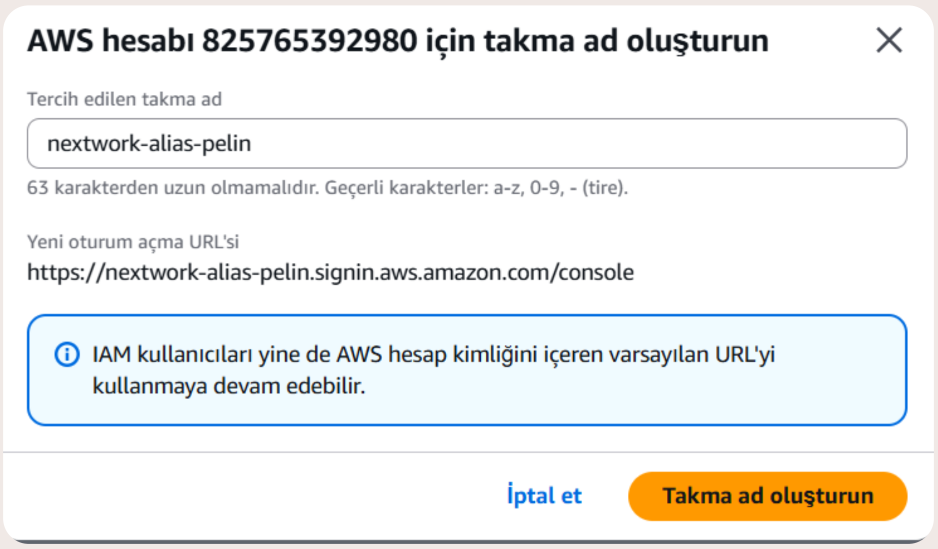
Instead of using the long numeric AWS account ID, we assign a custom, human-readable alias. This allows users to access the AWS Management Console through a simpler and more memorable login URL.

Understanding account aliases

An Account Alias is a custom name that represents your AWS account.
Instead of logging in with a long numeric AWS account ID, users can use this human-readable alias in the AWS Management Console login URL.

Setting up my account alias

It took less than a minute to set up the Account Alias, since it only involves choosing a unique name and saving the setting in the IAM console.


AWS hesabı 825765392980 için takma ad oluşturun

Tercih edilen takma ad

nextwork-alias-pelin

63 karakterden uzun olmamalıdır. Geçerli karakterler: a-z, 0-9, - (tire).

Yeni oturum açma URL'si

https://nextwork-alias-pelin.signin.aws.amazon.com/console

IAM kullanıcıları yine de AWS hesap kimliğini içeren varsayılan URL'yi kullanmaya devam edebilir.

IAM Users and User Groups

What I did in this step

In this step, we are setting up secure and centralized access for interns.

Specifically, we are:

Creating a dedicated IAM group for all NextWork interns to manage their permissions from one place

Defining permissions at the group level, instead of assigning them individually

Creating a dedicated IAM user for the new intern so they have their own login credentials

Understanding user groups

IAM user groups are collections of IAM users that share the same permissions.
Instead of assigning permissions to each user individually, you attach IAM policies to a group, and all users in that group automatically inherit those permissions.

Attaching policies to user groups

The effect of attaching the policy to the user group is that all users in the group automatically receive the permissions defined in the policy.
This means:
Every intern in the group can perform the allowed EC2 actions (such as starting, stopping, and describing instances)
Access is limited to the development instance only, based on the tag conditions in the policy
Users cannot modify or delete tags, preventing permission escalation

Understanding IAM users

IAM users are individual identities in AWS that represent a person or an application.

Logging in as an IAM User

Sharing sign-in details

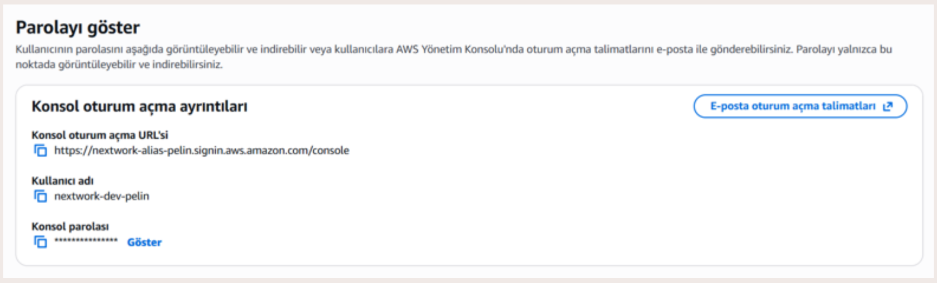
1)Send the sign-in information manually
Share the account alias (or login URL), username, and temporary password securely (for example, via a secure messaging tool).

2)Download the credentials file
AWS allows you to download a .csv file containing the user's sign-in

AWS allows you to download a .csv file containing the user's sign-in details, which can then be shared securely with the user.

Observations from the IAM user dashboard

In the new IAM user's AWS dashboard, I observed that access was limited based on the assigned permissions.



Parolayı göster

Kullanıcının parolasını aşağıda görüntüleyebilir ve indirebilir veya kullanıcılara AWS Yönetim Konsolu'nda oturum açma talimatlarını e-posta ile gönderebilirsiniz. Parolayı yalnızca bu noktada görüntüleyebilir ve indirebilirsiniz.

Konsol oturum açma ayrıntıları

Konsol oturum açma URL'si
https://nextwork-alias-pelin.signin.aws.amazon.com/console

Kullanıcı adı
nextwork-dev-pelin

Konsol parolası
*************** Göster

Testing IAM Policies

What I did in this step

In this step, we are verifying that IAM permissions are working as intended.
Specifically, we are:
Logging into AWS using the intern's IAM user credentials
Testing access to both the development and production EC2 instances

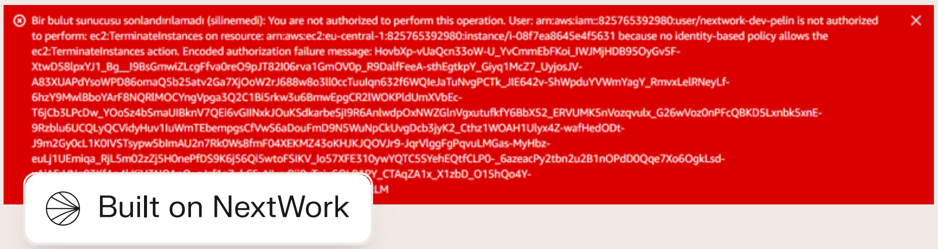
Testing policy actions

The action I performed on the two EC2 instances was testing instance control by attempting to stop them.

Stopping the production instance

When I tried to stop the production EC2 instance using the intern's IAM user, the action was denied.

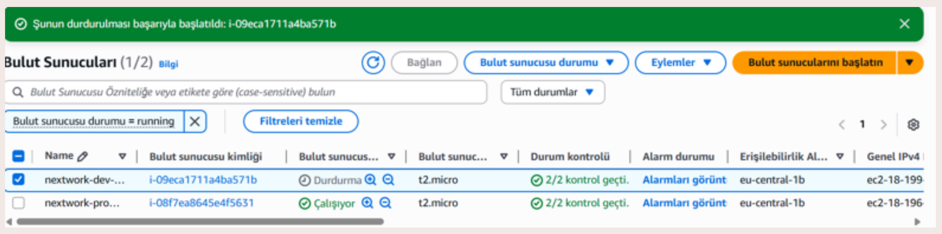
AWS displayed a permission error, indicating that the user was not authorized to perform this action. This happened because the IAM policy only allows EC2 actions on instances tagged with Env = development, and the production instance does not meet this condition.



```
Bir bulut sunucusu sonlandırılamadı (silinemedi): You are not authorized to perform this operation. User: arn:aws:iam::825765392980:user/nextwork-dev-pelin is not authorized
to perform: ec2:TerminateInstances on resource: arn:aws:ec2:eu-central-1:825765392980:instance/i-08f7ea8645e4f5631 because no identity-based policy allows the
ec2:TerminateInstances action. Encoded authorization failure message: HovbXp-vUaQcn33oW-U_YvCmmEbFKoi_IWJMjHDB95OyGv5F-
XtwD58lpxYJ1_Bg__I9BsGmwiZLcgFfva0reO9pJT82I06rva1GmOV0p_R9DalfFeeA-sthEgtkpY_Giyq1McZ7_UyjosJV-
A83XUAPdYsoWPD86omaQ5b25atv2Ga7XjOoW2rJ688w8o3ll0ccTuulqn632f6WQIeJaTuNvgPCTk_JIE642v-ShWpduYVWmYagY_RmvxLelRNeyLf-
6hzY9MwlBboYArF8NQRlMOCYngVpga3Q2C1Bi5rkw3u6BmwEpgCR2lWOKPldUmXVbEc-
T6jCb3LPcDw_YOoSz4bSmaUIBknV7QEi6vGIINxkJOuKSdkarbeSjI9R6AnlwdpOxNWZGlnVgxutufkfY6BbX52_ERVUMK5nVozqvulx_G26wVoz0nPFcQBKD5Lxnbk5xnE-
9Rzblu6UCQLyQCVidyHuv1IuWmTEbempgsCfVwS6aDouFmD9N5WuNpCkUvgDcb3jyK2_Cthz1WOAH1Ulyx4Z-wafHedODt-
J9m2Gy0cL1K0IVSTsypw5bImAU2n7Rk0Ws8fmF04XEKMZ43oKHJKJQOVJr9-JqrVlggFgPqvuLMGas-MyHbz-
euLj1UEmiqa_RjL5m02zZj5H0nePfDS9K6J56Qi5wtoFSIKV_Io57XFE310ywYQTC5SYehEQtfCLP0-_6azeacPy2tbn2u2B1nOPdD0Qqe7Xo6OgkLsd-
[illegible]PY_CTAqZA1x_X1zbD_O15hQo4Y-
[illegible]LM
```

Stopping the development instance

When I tried to stop the development EC2 instance, the action was successful. The instance stopped without any errors because the IAM policy explicitly allows actions like stopping EC2 instances that are tagged with Env = development. This confirmed that the intern had the correct permissions to manage development resources.



Şunun durdurulması başarıyla başlatıldı: i-09eca1711a4ba571b

Bulut Sunucuları (1/2) Bilgi
Bağlan
Bulut sunucusu durumu
Eylemler
Bulut sunucularını başlatın
Bulut Sunucusu Özniteliğe veya etikete göre (case-sensitive) bulun
Tüm durumlar
Bulut sunucusu durumu = running
Filtreleri temizle

	Name	Bulut sunucusu kimliği	Bulut sunucus...	Bulut sunuc...	Durum kontrolü	Alarm durumu	Erişilebilirlik Al...	Genel IPv4
✓	nextwork-dev-...	i-09eca1711a4ba571b	Durdurma	t2.micro	2/2 kontrol geçti.	Alarmları görünt	eu-central-1b	ec2-18-199
	nextwork-pro...	i-08f7ea8645e4f5631	Çalışıyor	t2.micro	2/2 kontrol geçti.	Alarmları görünt	eu-central-1b	ec2-18-196

IAM Policy Simulator

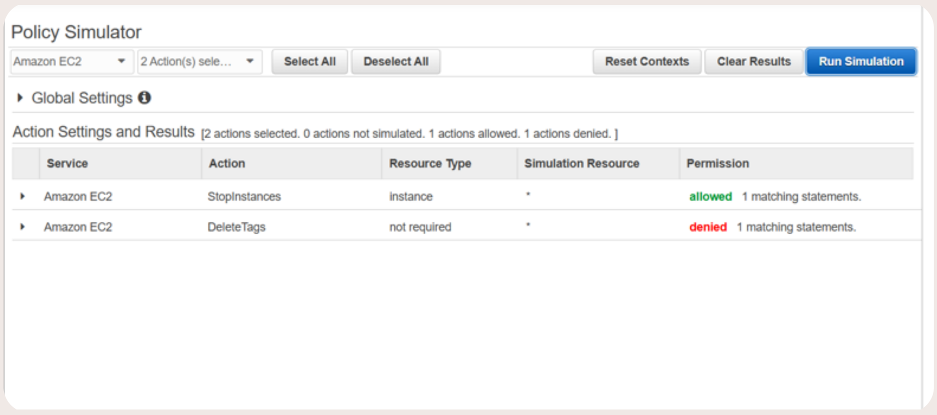
In this secret mission, we are testing and validating IAM permissions safely without making any real changes to AWS resources.

Understanding the IAM Policy Simulator

You would use the IAM Policy Simulator to test and verify IAM permissions safely before applying them in real environments.

How I used the simulator

The simulation results for the development EC2 instance showed that the requested EC2 actions (such as starting and stopping the instance) were allowed.



Action Settings and Results [2 actions selected. 0 actions not simulated. 1 actions allowed. 1 actions denied.]

Service	Action	Resource Type	Simulation Resource	Permission
Amazon EC2	StopInstances	instance	*	allowed 1 matching statements.
Amazon EC2	DeleteTags	not required	*	denied 1 matching statements.