IAM (Identıty and management)

TAM ANLAMI İLE AWS İN KALBİ. REGİONDAN BAĞIMSIZ BİR SEVİS. BURADA YARATILAN HERŞEY REGION DAN BAĞIMSIZ İŞLER

Tüm kğmlük ve erişim objelerini yönetmemizi sağlayan seviş

users

groups

polcies

roller

burada yaratılır

User:

AWS e bağlanıp işlem yapan kullanıcıya verilen isim

Bir kullanıcı adı ve şifre ya da acces key ve secret kombinasyonu ile sisteme bağlanıp AWS dünyasıda işelm yapmayı sağlayan roldür.

AWS dünyasında owner olan kişi aynı zamanda root yetkisine de sahiptir ve e-mail ve şifre kombinasyonu ile hesabına rahtaça girebilir. MFA gibi ek security check de eklenebilir.

Bu root sistemde en önemli ve en üst yetkiye sahip roldür.

Sadece ilk userlar oluşturulurken ve gruplar kurulurken kullanılır. Sonrasında tüm işlemle bu roller üzerinden sürdürülür. Nedeni ise root en yüksek yetkiye sahip kullanıcıdır. hesap silmek de dahildir bu yetkiye. Dolayısı ile yetkisi en üstte olduğundan hata yapmamak adına AWS sadece en başta bunu kullanmayı sonrasında oluşturulan hesaplar üzerinden işlemlerin devam etmemizi tavsye ediyor.

Gruplar: Birden fazla kullanıcıya tek sefer de istenile yetkisi atamamızı sağlayan objeler. 20 kullanıcı var ve onlara admin yetkisi verilmek isteniyor. Bir grup oluştururuz, 20 kişiyi bu gruba ekleriz ve sonrasında da sadece gruba yetki devri yaparak 20 kişiye de yetkinin verilmesini sağlarız.

Policy: AWS dünyasında kimin nereye hangi yetki ile erişebileceğini belirlememizi policy sağlar. Belirlediğimiz özelliklerde bir policy hazırlayıp bu policy i yetkilendirmek istediğimiz kişiye ya da gruba atar sonrasında gerekli yetki devrini sağlamış oluruz.

Mesela br kişiye sadece S3 e erişme yetkisi verebiliriz bu policy ler ile. Bu policy ler Json formatında olur. Policy ler iki çeşittir.

1. kendı ısteklerımıze uygun oluşturduğumuz polıcyler. ki bu json ile oluştururuz

2. AWS in öncesinde belirlediği policy ler

ROL: AWS kaynaklarımıza başka AWS kaynaklarının ya da diğer AWS hesaplarından gelen kullanıcıların ya da dış dünyadaki bir identity provider ile kurduğumuz güven ilişkisi sonucu sistemimize erişme yetkisi vermek istediğimizde nerelere hangi yetkilerle ulaşacağını belirlediğimiz yerdir.

Mesela EC2 ile oluşturlan bir sanal makinenin bir S3 servisine erişip oradan kullanacağı veriyi çekmesini bu sanal makineye atayabileceğimiz roller ile belirler. Ya da başka bir AWS hesabında bir kullanıcının bizim hesabımızda işlem yapmamızı istiyorsak o hesaba bizim kaynaklarımızla çeşitli işlem yapabilmeyi sağlayan bir role atanabilir.

1. Rol nereye atanacak

2. Rolün atanacağı kaynağın neler yapacağı belirlenir.

İşte IAM bize bu 4 obje tipini yaratıp yönetmemize izin veren merkezi kimlik yönetim sistemidir. AWS dünyasında sahip olunan tüm yetki tipleri bu servisle ayarlanır. Atrıca bu servis güvenlik katmanlarının da yararlandığı yerdir.

ÖZET

1. Merkezi AWS hesap yönetim servisidir

2. Tüm hesap ayarları bu servis aracılığı ile yönetilir

3. User, Group, Role, ve Policy yaratıp yönetmemize imkan tanır

4. Kullanıcıların her türlü şifre kısıtlamaları ve MFA (Multi Factor Authantication) IAM aracılığı ile yapılır

5. Dış kimlik doğrulama mekanizmaları ile “Ör:Active Directory” kurulacak bağlantılar IAM üstünde yönetilir

User oluşturmada

adduser a basılırsa kendimize kullanıcı adı seçeriz

bu noktada AWS bize iki seçenek sunuyor. Bu oluşturulan kullanıcı AWS kaynaklarına ne şekilde ulaşabilsin. Bunlar

1. Programatic acces

2. AWS Management Console accsess

Kullanıcı aynen bizim şu anda yaptığımız gibi AWS kaynaklarına konsol üzerinden erişebilir ve işlem yapabilir.

Bunun üzerine programmatic acces seçilirse bu durumda bu kullanıcı bu konsol dışında AWS kaynaklarına komut satırı üzerinden AWS API, CLI, SDK ve diğer development toll lar ile de erişebilir. Şimdi bu konsolda biz aslında kullanıcıya bir kullanıcı adı ve şifre belirliyoruz . Ancak kullanıcının komut satısı üzerinden erişim sağlayabilmesi için acces key ID ve secret Accesskey denilen başka bir şifre katmanına ihtiyacı var. Yani kullanıcı normal konsola bağlandığı kullanıcı adı ve passsword ile API ya da SDK e erişemiyor o yüzden be bu yeni şifre katmanı devreye giriyor. Biz bu kullanıcının bu altyapılara erişmesini istiyorsak o zaman programatic access yetkisini de vermek lazım.

User oluşturmada, add user kısmında Tag lar şunun için kullanılır:

Tag lar ke value pair şeklinde yaratılan ve AWS ortamında tüm kaynaklara atanabilen etiketlerç. Bu syaede kaynakları bir araya toplayabiliyoruz bir arad gösterebiliyoruz. Böylece kaynağın ne ile ilgili olduğunu görebiliyoruz.

Mesela

Key: depertment

value :bilgiişlem

key lokasyon

value: İstanbul

desem, sonrasında lokasyonu İstanbul olan tüm kullanıcıları buradan süzebilirim. Bir çk kaynağı bir arada görmemi sağlıyor.

Kullanıcı oluşturulduğunda girilen tüm bilgileri bir review etme imkanı verir. sonrasında create user denildiğinde Acces key ID ve Secret Access key oluşturu. ayrıca bu bilgileri içeren bir .cvs dosyasını indirme imkanı verir. Bu download ımutlaka yapın, zira bu key’lere konsol üzerinden tekrar erişilemiyor. Yenisi tekrar yaratılıyor ama mevcut olan ilk yaratılana tekrar ulaşamıyoruz. Buradaki sent email, kullanıcıya bu account ile ilgili bir email gönderilmesine sağlar.

IAM Policies

NOT: HERHANGİ BİR POLİCY ATARKEN KARŞI TARAFA SADECE KULLANACAĞI KADAR YETKİ VERMEK ESASTIR. BUNA LEAST PRİVİLEGE PRİNCİPLES ADI VERİLİR.

Policies AWS de kimlik bilgileri ve kaynaka tanımlamaları ile ilişkili izinlere verilen genel isimdir. Bu izinler hangi hizmetleri kullanabileceğinizi ya da hangilerine giriş izninizin olmadığını belirler. Bu polcy ler JSON formatında saklanır. JASON stands for javascrıpt Access polıcy

Aşağıdaki gösterime statement blok denir. version, effect, action ve resourse dan oluşur

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "\*",

"Resource": "\*"

}

]

}

Burada;

Version: Jason un kullanılan versiyonunu belirtir

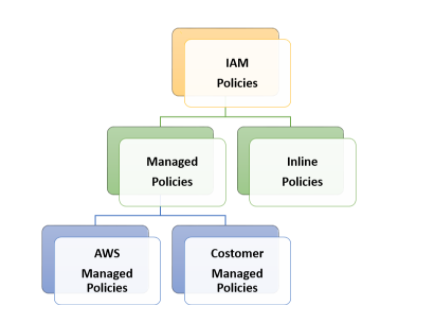
Statement Block: Üç bilgi içerir, effect, action and resources

Efect: sadece allow ve deny değerlerını alır. statement ın ne yapacağını belırler.

actıon effect keyden hareketle hangı actıon un ızın verılıp hangısıne ızın verılmeyecegını belırtır.

resources. hangı aws kaynaklarının statement tarafından dıkkate alınacağını belırler.

Policy oluşturmak için IMA a giriş yaptıktan sonra solda policy kısmı tıklanır. burada default tanımlanmış policy ler olduğu gibi, kendimizin de policy tanımlamamıza imkan veren bir bölüm vardır.



2 tür IAM policy mevcut. Bunlardan ilki Managed policy, ikincisi inline policy

Managed policy ikiye ayrılıyor.

a. AWS managed policie:

Bunlar AWS tarafından doluşturulan policiy lerdir. Daha yeni başlıyorsanız AWS bu hazırlanmış policy leri kullanmamızı öneriyor

b. Managed Policies: Bizim kendi amacımıza yönelk yazdığımız policy ler. Bunlar daha specific kontrol imkanı sağlayan policy oluşturabileceğimiz türdür.

Inline Policies: Bunlar direk bir user, group ya da role atayabileceğimiz ve erişimini yönetebileceğimiz policy lerdir. Bu policy’ler, user bir policy ile geçerli olduğu ana başlık (user, group veya role) arasında bire bir sıkı bir ilişki sürdürmek isterse yararlıdır. Bu nedenle, işletmenin veya kaynağın silinmesi aynı zamanda satır içi politikanın silinmesine de yol açacaktır.

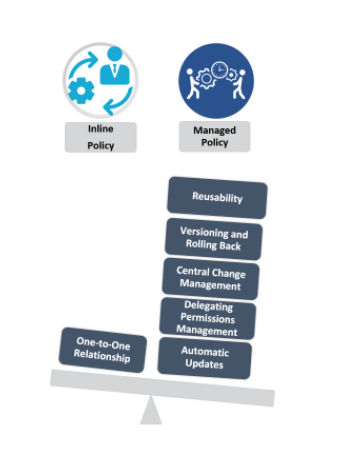
Inline policy ler yukarıda belirtildiği gibi user, group ya da rollere direct atandığından genel policy listesinde görülmezler. Sadece entity ye ekli görülür ve entity silindiğinde silinirler.

Peki inline mı managed mi diye sorulursa verilecek cevap nedir? Burada AWS managed policy lerin kullanılmasını tavsiye ediyor. Çünkü daha kullanışlı ve tekrar kullanılabilirlik, geri alınabilir

Not: Policy belirlemede visual editör kullanılıyor ise Actions da istersek seçtiğimiz servisin tüm kaynaklarını kullandırabiliriz ya da manuel yetki vermeyi seçip Access level olarak listeleme, okuma yazma ya da izin yönetim privilage ını verebiliriz. Ayrıca, Resources kısmından kaynak belirleyebiliriz. Eğer alt kısımda Request conditions kısmında MFA ya da source IP seçersek giren kişi için MFA ya da source IP erişim için bir ön koşul haline gelecektir. Source IP şudur: Bu seçilirse, kullanıcının bu kaynaklara erişebileceği IP bilgisayarları tanımlama hakkını bana verir ki ondan başka bilgisayardan bağlantı kuramaz.

Örnek221.12.12.0/24 gibi bir IP aralığı verebiliriz

Not: Description kısmına muhakkak açıklayıcı bir açıklama eklemeyi unutma



AWS dünyasında 4 tür policy vardır

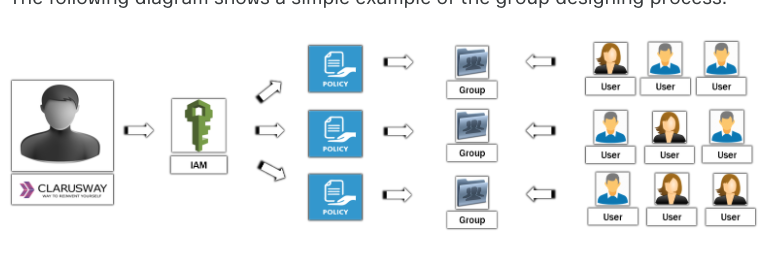
1. Inline policy (can be created for specific user, group or role)(is created by yourself)

2. Customer Managed (can be created for multiple user or groups)(is created by yourself)

3. Job descriptive (created by AWS)

4. AWS Managed (crated by AWS)

IAM Groups



An IAM group is essentially a collection of users that makes administration easier over IAM users. Some important features of IAM groups as follows:

* Groups can contain many users, and a user can be in multiple groups.
* Groups have no credentials.
* Managed IAM policies can be attached to groups.
* Inline IAM policies can be added to groups.
* Groups can contain only users, but not other groups.
* There's no default group that automatically includes all users in the AWS account.
* The limit of IAM users in a group is equal to the user quota for the account, that is,  max 5000.
* An IAM user can be a member of the max. 10 different IAM groups.

IAM Roles

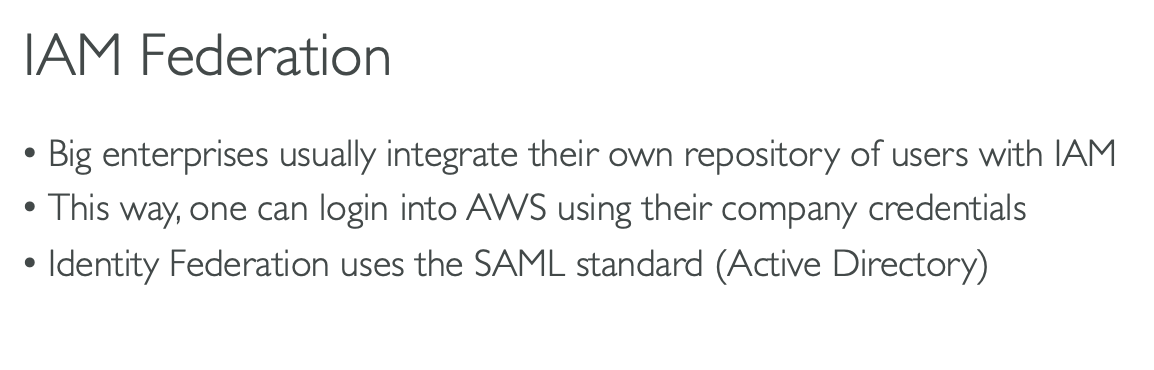
Roller AWS kaynaklarının bir başka AWS kaynağına veya dışarıdan bir kullanıcının bizim AWS kaynaklarımıza erişebilmeleri için yarattığımız yetki seviyeleri.

Yeni rol oluşturma:

+ farklı rol yaratma seçeneği var. Bir AWS servisinin bir başka AWS servisine ulaşmasını istiyorsak AWS service seçilir.

Bir başka hesaptaki AWS kullanıcısınız bizim hesabımıza erişmesi için bir rol atamak istiyorsak bu sefer Another AWS account ile devam edilir. Mesela her iki tarafın da AWS hesabı var ve bir taraf diğer tarafın AWS hesabı üzerinde bazı işlemler yapmasını istiyorsunuz. Mesela bir sanal makine yaratmayı istiyor. Bu durumda bir rol yaratıp orada da bir yetki verilirse o zaman bir taraf diğer tarafa erişir ve verilen yetki sahibi olur.

Kendi aktif directory mizdeki kullanıcıların bir AWS hesabına bağlanmsını istiyorsak bu durumda SAML 2.0 federation butonundan yürünür. SAML NEREDE KULLANILIR:



Rol atamada önce hangi kaynaklarımız için rol atamak istediğimizi seçtik. Sonrasında Bu servisiin hangi izne sahip olacağı seçilir mesela s3 araması yapılır ve s3 full Access verilirse bu durumda EC2 makinesi atandığı kaynağın s3 makinelerine tam yetki ile erişme yetkisi verilir.

* An IAM role, similar to an IAM user, is an IAM identity that has specific permissions that you can create in your account.
* A role is meant for anyone who needs it to be assumable. An entity is allowed to assume the role and when the role is assumed, the entity gains the permission that that role has.
* By creating a role, in fact, you define entities that can assume this role via policies.

Every role has two policies: A trust policy and a permission policy.

* The trust policy defines who can assume the role. It is a JSON policy document in which you define the principals that you trust to assume the role.
* The permission policy is just an IAM policy that gives that role some permissions on some things. It can be thought of as the same as an IAM policy on a group. A permissions document in JSON format in which you define what actions and resources the role can use. The document is written according to the rules of the IAM policy language.

Roles basically contain 2 main components.

* First of all, when creating the role, it is determined who can use this role, that is, where this role can be assigned.
* Then, it is determined what powers the reliable resource to which this role is assigned will have and what it can do. This is done through the IAM policy files we have learned.