Cloudformation

Her seferinde ayağa kaldırılması gereken sistemleri teker teker ayağa kaldırmak çok zor olacaktır. Kurmak istediğimiz altyapıyı adım adım kurulum aşamaları ve kurmak istediğin servisler şeklinde yaxabileceğin bir template yazma imkanı sağlıyor. Daha sonra bu template deki tüm aşamaları tek bir tık ile kurabilmeni sağlıyor.

Avantajları

* Altyapımızı kod ile tanımlamamızı sağlamaktadır (IaC). Yani herhangi bir kaynağı (EC2 instance, S3 Bucket..) manuel olarak oluşturmamış oluyoruz, bunun da bize kontrol ve yönetim açısından büyük avantajları olmaktadır.
* Aynı altyapıyı farklı ortamlarda kurmamızı kolaylaştırmaktadır.
* Git gibi versiyon kontrol sistemlerini kullanarak altyapımızın versiyonlarını kod üzerinden takip edebilmemizi sağlamaktadır.
* CloudFormation template’i kullanarak oluşturduğumuz AWS kaynaklarının aylık tahmini maliyetini görebiliriz.
* Maliyetleri azaltmak için kaynakların kullanılmadığı zamanlarda Stack’leri silebiliriz(kritik olmayan servisler için). Örneğin dev ortamının Stack’ini gece 2’de silinip sabah 7’de tekrar ayağa kalkacak şekilde programlayabiliriz.

Temelde iki ana komopnete sahip

ilk komponent, template dosyaları -🡪 jason ya da yaml formatındadırlar… stack adı verilir.

Bu template cloud formationa aktarılır ve bu template deki yapıyı ayağa kaldır diyorsun. Bu bilgiye göre cloudformation bize bir stack yaratıyor sıra ile. Bu template daha sonra yeni bir stack yazmak için defalarca kullanılabilir. Daha sonra bu template de bir güncelleme yapılırsa bu güncellemenin altyapıya eklenmesi de sağlanır. Silmesi de kolaydır. Stack silindi mi o template e bağlı tüm komponentler bir anda silinir. Teker teker makineleri kapatacağım diye uğraşılmaz. Tek bir tuşla tüm stack kaldırılabiliyır. Cloudformation biz çözüm mimarlarının işleri oldukça kolaylaştırılabilir.

Bu template nasıl bir dosya

jason ya da yaml formatında yazılabilir.

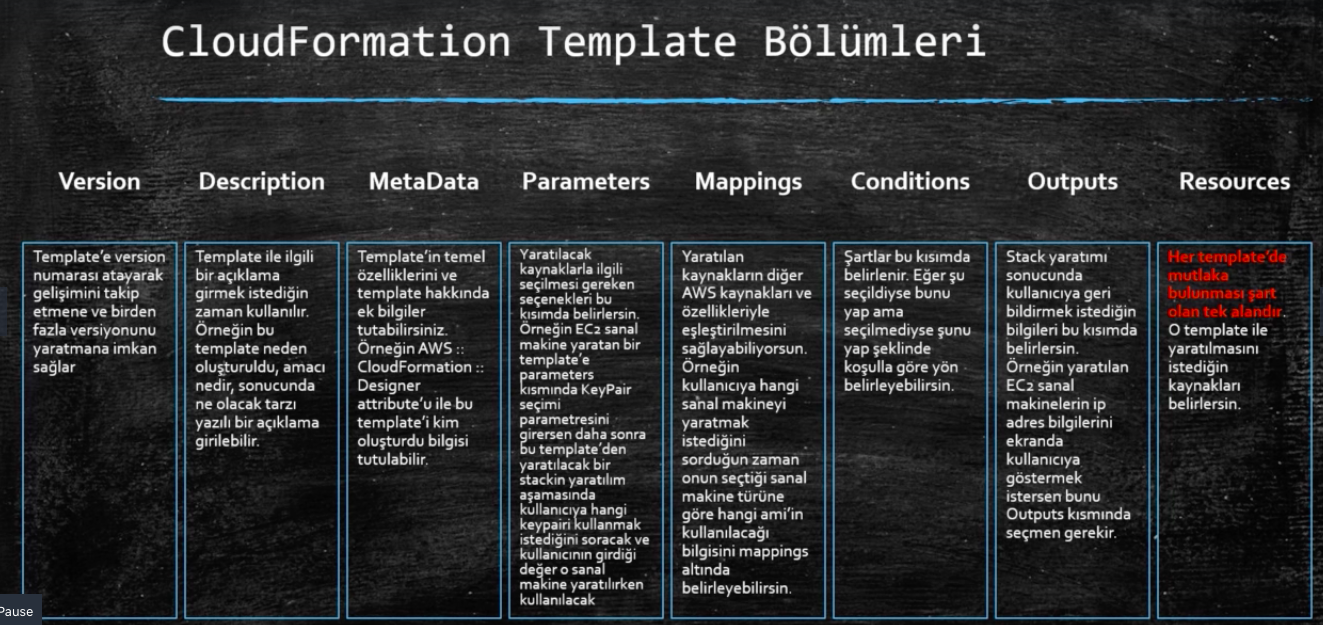
Bu dosyalar aşağıdaki ana bölümlerden oluşur

1. Versiyon

2. Description

3. Resources

4. Output



<https://s3-us-west-2.amazonaws.com/cloudformation-templates-us-west-2/VPC_AutoScaling_and_ElasticLoadBalancer.template>

Örnek tempate

Bu template ile bir VPC içerisinde Autoscaling group altında basit bir websitesi baroındıran sunucu ya da sunucular yaratıp bunu bir ELB arkasına alacak. Parametrelee bakılırsa öncesinde girilmesi gereken parametreleri görebilirsiniz. Mappings kısmında örneğin kullanıcı hangi Region da ise kullanıcı hangi AMI kullanacak onu belirlemiş. Outlook kısmında ise http ile web sitesi ismini oluşturacak

UYGULAMA

Cloud formation yöentim ekrarnına gel

4 adet seçenek çıkar

1. create new stack

Daha önceden oluşturduğumuz template den yeni bir stack oluşturuyoruz

2. Create a stack set

Birden fazla account ve regiona kurabildiğmiz türü .normalde elimizdeki tempate ile sadece kendi accountumuza işlemler yapabilirdik. geen seneden itibaren bu madde geldi . burada template i kendi hesabımızda tutup buradan hareketle başka hesaplarda stack oluşturabiliyoruz. Aynı template den birden çok mişterimizin hesabımızdan aldığımız tempate ile stack kurabiliyoruz.

ya da ortak bir template havuzu oluşturulabiliyoruz

3. Design template

Kendimiz template oluşturuken visualize edip GUİ den sürükle bırak yöntemi ile yapabiliyoruz.

4. Create a Template from your Existing Resource

Hazır bir sistemimiz varsa bu sistemi template haline getirmemizi sağlıyoruz.

Şimdi biz CREATE NEW STACK diyelim

Burada bize iki seçenek sunar

1. Template is Ready

Eğer ayağa kaldıracağımız stac e ait template elimizde hazır ise bu kullanılır

2.Use a sample template

Burada AWS in önceden oluşturmuş olduğu template leri seçme imkanımız olur.

3. Create a template in Designer

Burayı seçersek az önce bahsettiğimiz GUI’ye gidip bize template oluşturma imkanı verir.

Biz bu örneğimizde AWS den aldığımız template i kullanacağız. Buna bir bakarsak neler görürüz onu inceleyelim.

ilki bir versiyon numarasını verir

ikinnci kısım description kısmıdır. burada teplate’in descriptin ını bulunabilir.

Parameters kısmında ise ben bu stack ‘i yaratırken sonkullanıcının seçmesini istediğim parametreleri belirtirim. Bunları template içerisinden alabileceğim gibi buradan da belirlenmesini isteyebilirim.

Mappings kısmında ise, eğer kullanıcı şunu seçerse sunu yap bunu seçerse bunu yap gibisinden şeklne mapleme yani birbirine eşleştirme yapmış Outputs kısmında ise bu stack yaratıldığında son kullanıcıya çbilgi olarak ne verilsin onu belirliyoruz. Mesela bu template de son kullanıcıya bir web adresi verilecek

**Next**

İlk olarak bu bu stacke isim vermemiz isteniyor. Sonrasında da paramaters kısımları var

Burada cloud formation kısmında son kullanıcıdan bunların sorulmasını template de biz sorulmasını istedik. Burada template de bunu iki şekilde yapabiliriz. Ya hepsini biz belirleriz ya da burada son kullanıcıdan bilgileri alırız.

Next

tags

Bu yaratılan kaynaklara herhangi bir tag atamak isteyip istemeyeceğimizi sorar. Bu tag tüm kaynaklara atanır

permissiom: Eğer cloud formation bu kaynakları oluştururken kullanması gerektiği bir IAM role varsa burada o atanır.

Rollback Trigger:

Buraya bir değer girelim örneğin 90 ve gidip cloudwarch dan bir alarm ARN girebiliyorum. Bunlar yapılınca AWS bu stack yaratılırken 90 dk boyunca belirlediğim alarm gözleyecek, eğer bu alarm tetiklenirse AWS bu templatein yaratılmasını durduracak yaptığı tüm işlemleri geri alacak ve tüm kaynakları silecek. peki niye buna ihtiyaç var? Mesela biz bir cloud formation template i yaratırken oluşturulan senin template in başka bir birimdeki başka bir kaynağı bozdu ve tetikledi. İşte bu durumda stack oluşturulmuyor ve o ana kadar hangi template oluşmuş ise hepsi siliniyor. böyelikle hata daha büyümeden geriye dönük olarak tüm template siliniyor.

5 taneye kadar cloudwatch alarmı eklenebiliyor. Çeşitli metrikleri izleyebilirsin buradan

ADVANCE

Notificaitn

Bu stack in yaratılması esnasında bizin bir notificasyon alıp almayacağımız sorar. Sns topick oluşturulabilir ya da halihazırda oluşmuş olan topik burada atanabilir. Bize alarm gönderir

Termination protection: aynen EC2 daki gibi bu stack’in silinmesini engeller. Önce bunu açmak lazım

timeout:

ilginç bir özellik. mesela 60 dk girdim. AWS 60 dk boyunca bekleyecek. 60 dakika sonunda haa bu stack yaratılmadı ise buna time out verecek ve bütün o ana kadar yaartılan kaynaklar silinecek ve geri alıncak. Neden?

biz bir template yrattık ve içinde 10 larca kaynak var. Yaptığımız bu tempalte de bir hata var ve stack bir yerde tıkandı. o durumda 1 saat sonra hala stack yaratılmadı ise 1 saat sonra bu stacde o ana adar tüm kaynakları siler geriye dönük olarak.

Rollback on failuere:

Stack yaratımı sırasından herhangi bir sorun olduğunda bir sonraki adıma devam etmesini mi yoksa tüm işlemler geri almasını mı istersin ona karar veriyoruz. Mesela 10 instance yarat dedik ve 4 üncüde sorun çıktı. Ne yapsın 5incyi yaratmaya devam mı etsin yoksa orada kesip geriye dönük tüm kaynakları silsin mi?

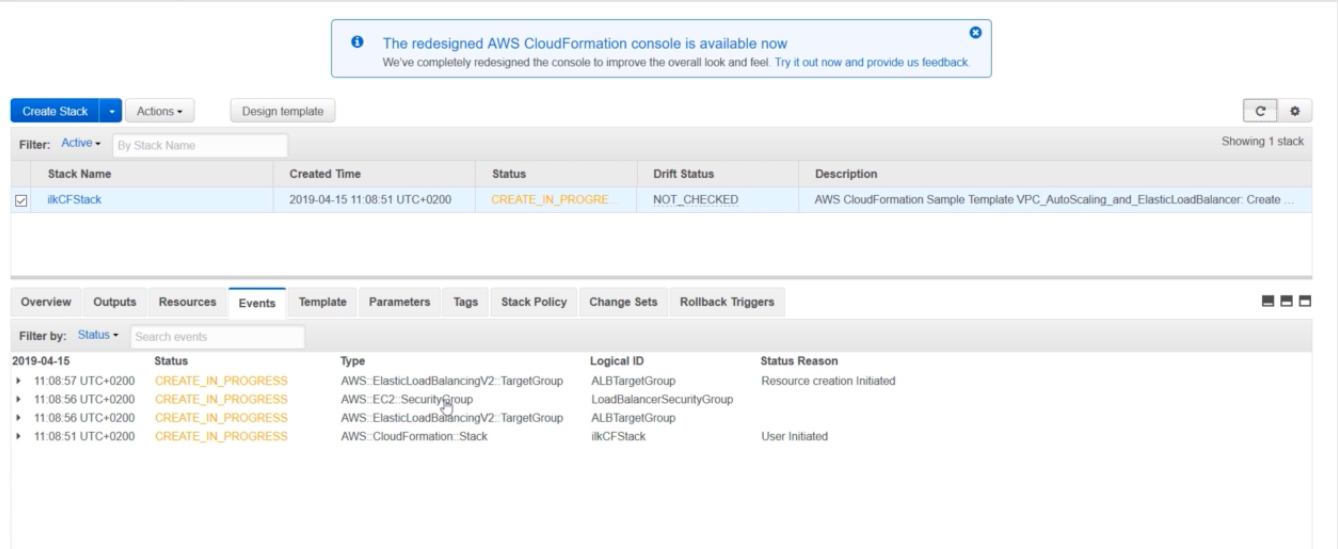
Stack policy

Stack yaratıldıktan sonra bu stack ile ilgili kimin ne işlem yapıp yapamayacağını belirleyeceimiz policy ler yaratılmasını sağlar. json formatında oluyor genelde.

Net

Review yaptırır ve create dnirse stack yaratılmaya başlar

create diyelim



Burada events kısmına gelinirse sıra ile neleri yaptığı görülebilir. Status olarak da create in progres olarak gözükür. Belirtilen herşey adım adım cloudformation servisi yaratır.

Tamamlanınca status create comlete denir.

Events kısmında create in progress gözüküyor. Bu bir AWS in bir bug ı. bunu önemseme. eğer status olarak create yazıyorsa gerisinin önemi yok.

Outputs kısmına geldiğimde burada bana bir url verdiğini görürüm. bunu vermesini ben template de istedim.

Sonrasında delete dendiğinde tüm stackden hangi kaynaklar yaratıldı ise hepsini siler. bizim uğraşmamıza gerek kalmaz.