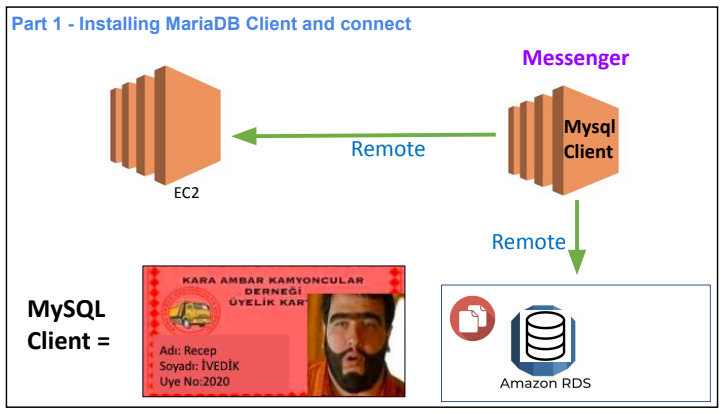
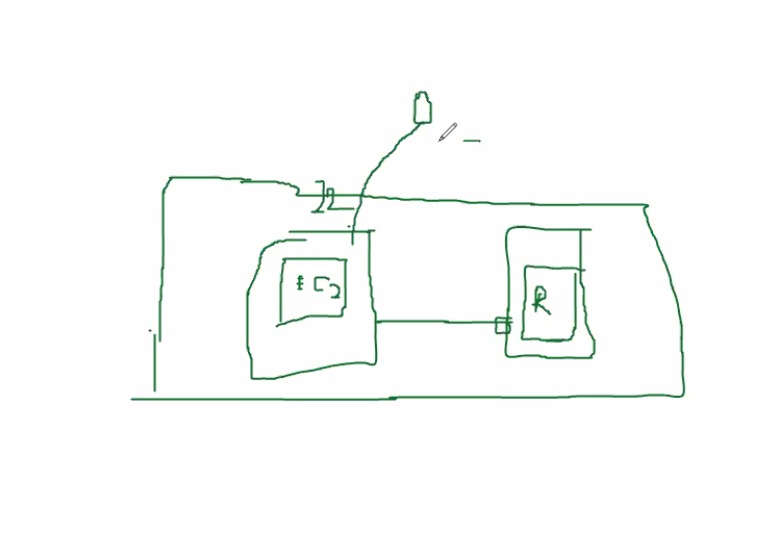
RDS

Database de saklanan veriler önemli olduğu için ekstra güvenlik katmanı gerekir. Bu amaçla önce uygun mimari yapı oluşturulur sonra yapıda belli erişimler verebilmek için uygun security Group oluşturulur (2 tane SG oluşturacağız) .

RDS ve tüm databaseler 3306 portunu kullanır. Database kurulum sırasında **PUBLIC** yapmadığımız sürece database workbench ile bağlanamayız.

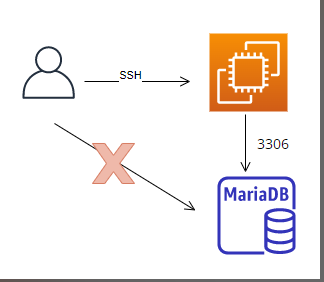


RDS ye client ile bağlanacağız. Bunun sebebi de hem database’in yoğunluğunu hazırlamak hem de girişlerdeki güvenliği çok daha üst seviyelere taşımaktır.

#### HANDS ON :

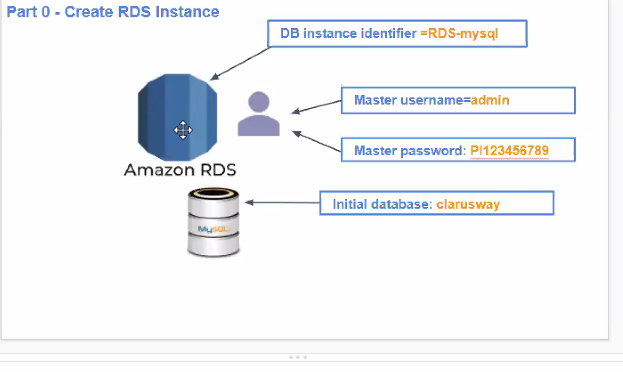
Önce Sec group oluşturup. RDS, Ubuntu, MariaDB oluşturup tablo yapacağız. Sonrasında snapshot alacağız.

Ec2 üzerine kurduğumuz RDS den bağlanacağız.



Multi-AZ dediğimizde aklımıza **Failover/ Continious Data Flow**, Read Replica dediğimizde ise **Performans** gelecek.

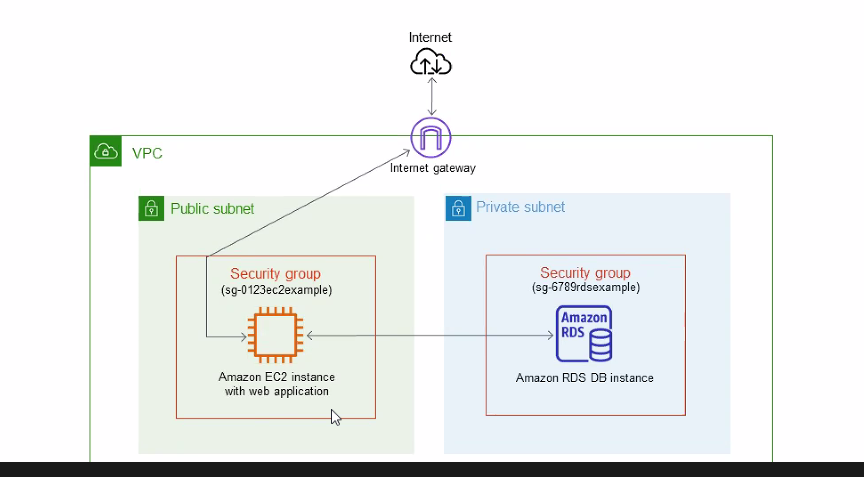
RDS oluşturuyoruz.



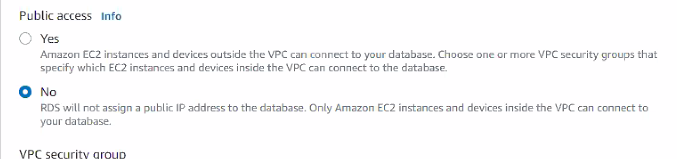
RDS içine EBS volume konuluyor. Onun için storage bölümünde 20 GB bize yeterli. Ancak bunu otomatik hala dönüştürebiliyoruz ve bir balonun şişmesi gibi ihtiyaç halinde düşünün. İhtiyaçla artabilir.

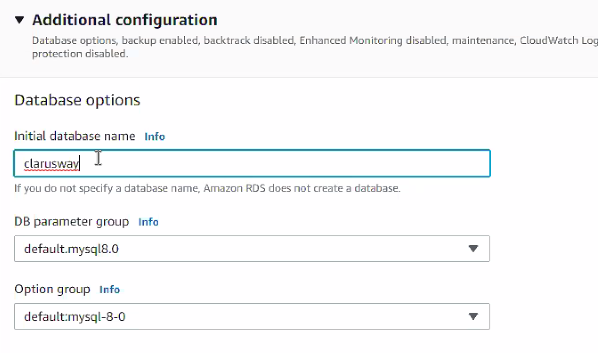
Connectivity de VPC ye dikkat etmek gerekir.

DataBASE leri özel alanlarda saklamamak gerekir. En az 2 tane subnet grubu olmasını tavsiye ediyor AWS. AWS kendi yönettiği için bunu istiyor.



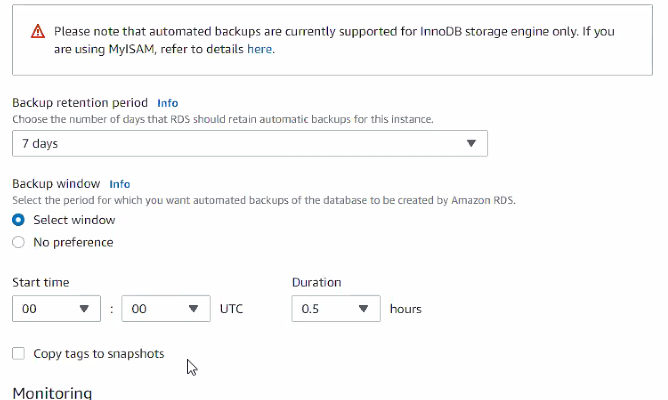
Bunlar aynı AZ de olduğu için Public Access i yes veya no seklinde ayarlar iken bunu dikkate almak gerekiyor.





DB yi EC2 gibi düşünün. Buna ram, storage vs eklemek gerek. Engin’e de ilave etmek gerek default Mysql i seçiyoruz. Bu kısımda daha fazla DB de ilave edebiliriz. Bunu siyah ekrandan da yapabiliriz.

Mevzu DB in yönetimi, sorunsuz hayatına devam etmesi çok ama çok önemlidir. AWS bizim için backup alıyor. 1-35 günlük backup sunuyor. İlk gün alınan data 36. Gün siliniyor. Bunu s3 te tutuyor. İhtiyaç halinde buradan bir maine kaldırabilirsin.

Bunu konsoldan seçerken 7 gün ile başlatıyor. 

Maintenance kısmında bizim için bir bakım yapıyor. Backup saatinden sonra maintenance koyabiliriz.

AWS nın her region için bir backup aldığı saat var. Sizin için alıyor. Yarım saati geçerse bakım ona göre ayarlanır.

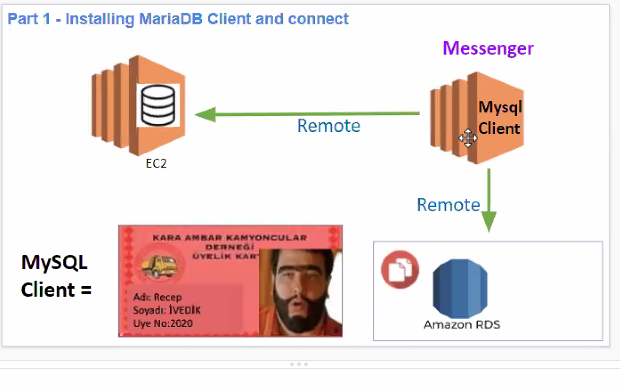
Maintenance ile backup arası fark: 1 gün daha büyük backup dır. Diğer günler üzerine ekler. Maintenance update gibi düşünün.

‘’

[Fatih Yildiz](https://app.slack.com/team/U02SREREQFN)  [9:55 PM](https://awsdevops-10.slack.com/archives/C02QVKUM33J/p1646333715343719)

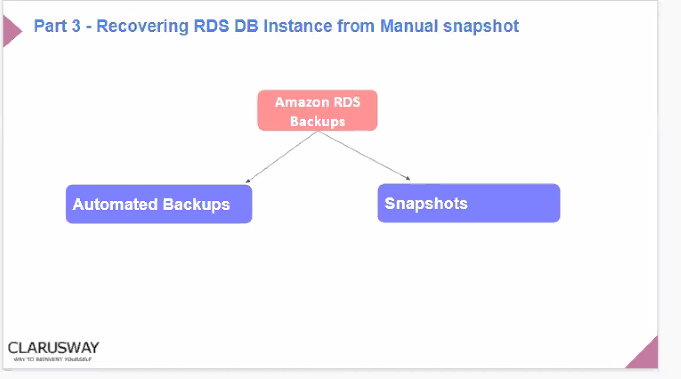
Sanırım su şekilde hocam. Sistem backup alırken sadece backup almak için uğraşıyor. Yeni data yazmıyor backup alana kadar. Backup bittikten sonra tekrardan normal düzenine donuyor. Sistemin büyüklüğüne göre back up suresini arttırıp tek seferde backup alınabilir.

‘’  
  
  
RDS ayağa kalırdık. Şimdi client ı ayarlayacağız. Ubuntu seçiyoruz.



Client e oluşturduktan sonra RDS e bağlanıp içine MariaDB ekleyeceğiz. SSH ile client e girip EC2 you client olarak kullanacağız.

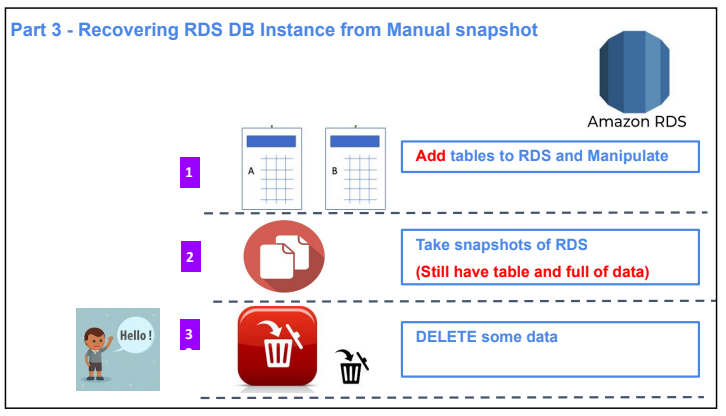
### PART 2 :



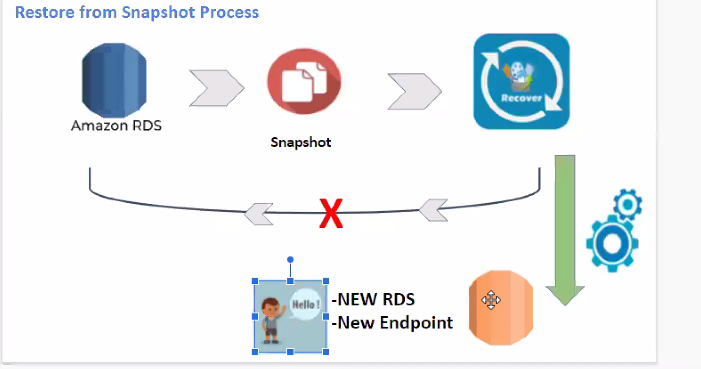
Tablolar güncel iken snapshot alacağız.

RDS ye gelip point time ile manuel Snapshot alacağız. action bölümünden take a Snapshot diyoruz. Yani mevcut durumun fotosunu çekiyor. Siz bir region da 40 Snapshot tutabiliyorsunuz. Storage ile bir kısıt var. İlk belirler iken 20 gb ı seçmiştik. Toplama snapshot ve backuplar 20 GB ı geçerse onu ücretlendiriyor.

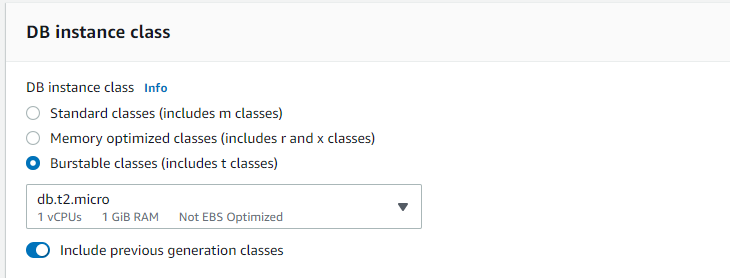
### PART 3:



DB yi kurduk. İlgilisine teslim ettik. Junior imiş. Yanlışlıkla bazı şeyleri sildi. Senaryo bu şekilde.

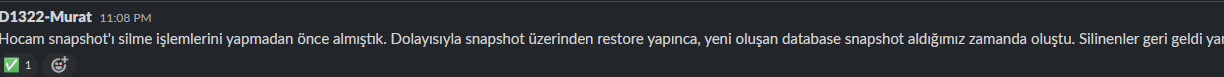


Yeni makine kurulmuş oldu. Endpoint i farklı olacak.

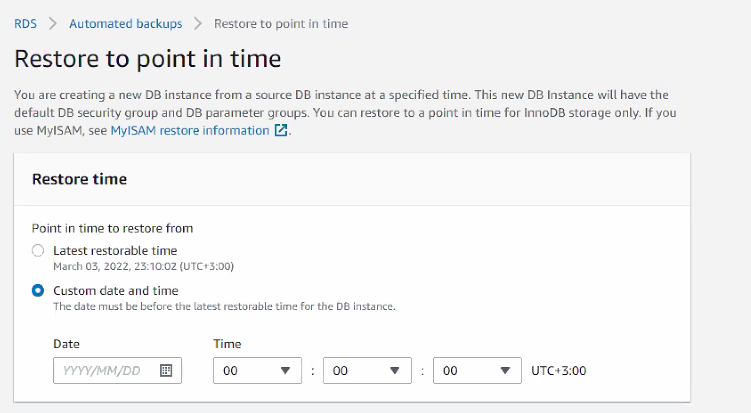


Burada AWS nin tuzağı var dikkat edelim. T2 micro yu seçelim.

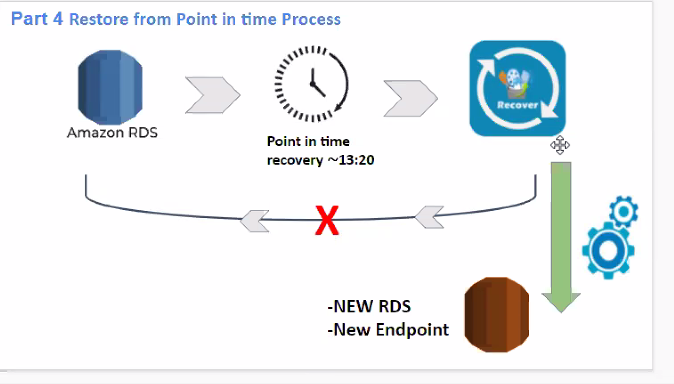
Buraya kadar yaptıklarımızı kısaca özetlersek; tabloyu oluşturduk. Db yi teslim ettik. Junior sildi. Once Snap shot aldık. Sonra restore yaptık. Yeni bir instance oluştu. Bunun end pointi farklı durumda.



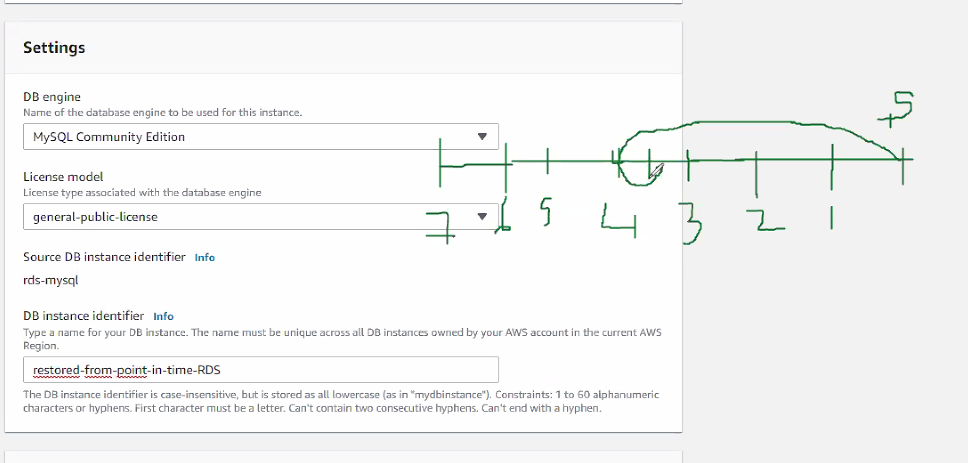
Şimdi Restored from man snapshot ı siliyoruz.



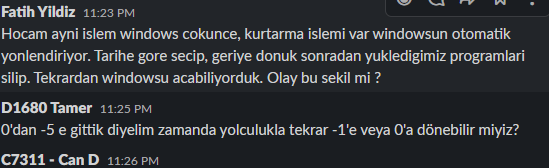
Şimdi not aldığımız saat i baz alarak point time ile ayarlıyoruz. Sanki yeni makine kuruyor gibi set ediyor.



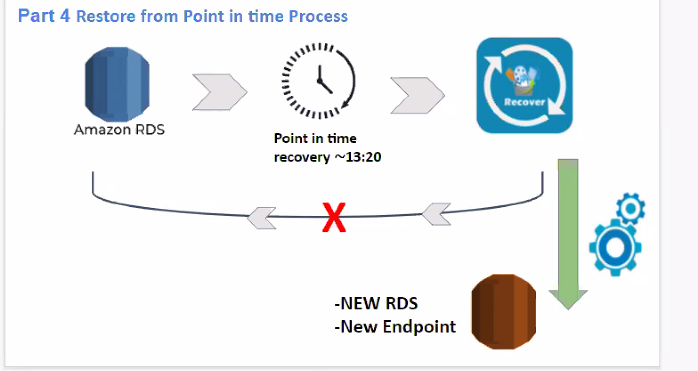
4uncu günün herhangi bir zamanına donuyorum.

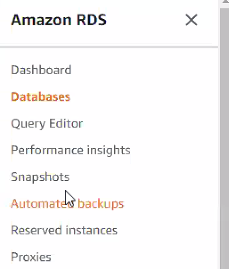


Automated Backup ile daha önce ayarladık isek transasion log ile geriye gidip alabiliyoruz. Eğer bu yok ise ve eğer snapshot almış isek buradan geriye gidip işlem yapabiliyoruz.

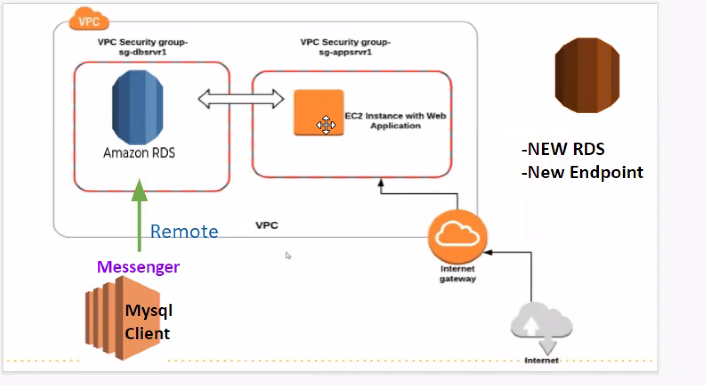


### PART 4:





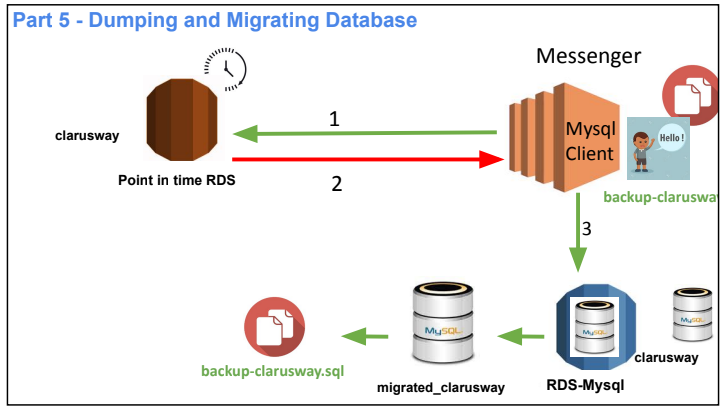
Burada Snapshot ve Automated Backup ile görebiliyoruz.

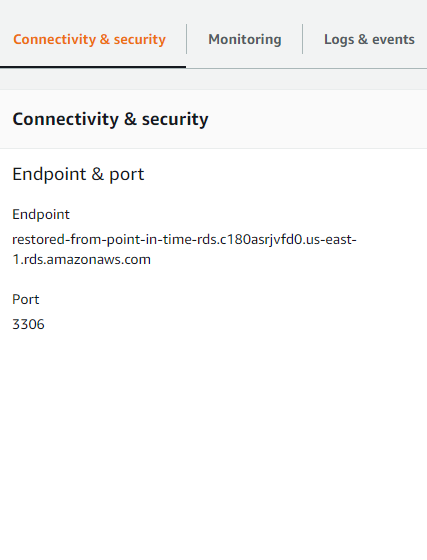


Burada restore yapıldı ama daha problemi çözemedik. Aplikasyon eski DB yi kullanmaya devam ettiği için problem devam ediyor. Akabinde dumping işlemini yapmak gerekir.

### PART 5:

Dump işlemi ise aşağıdaki gibi yapılır.





NOTE

[D1322-Murat](https://app.slack.com/team/U02S2316SDU)  [11:44 PM](https://awsdevops-10.slack.com/archives/C02QVKUM33J/p1646340243407959)

Birden fazla database migrate etmek istersek;

mysqldump -h ENDPOINT -u username -p --databases DB1 DB2 DB3.. > file\_name.sql

[D1322-Murat](https://app.slack.com/team/U02S2316SDU)  [11:45 PM](https://awsdevops-10.slack.com/archives/C02QVKUM33J/p1646340321215589)

Tüm database leri migrate etmek istersek;

mysqldump -h ENDPOINT -u username -p --all-databases > file\_name.sql

[D1197-Alparslan](https://app.slack.com/team/U02S5R4J8DT)  [11:46 PM](https://awsdevops-10.slack.com/archives/C02QVKUM33J/p1646340375835089)

Mysqldump komutunda > işareti kullandık, restore ederken ise < işareti kullandık.

