

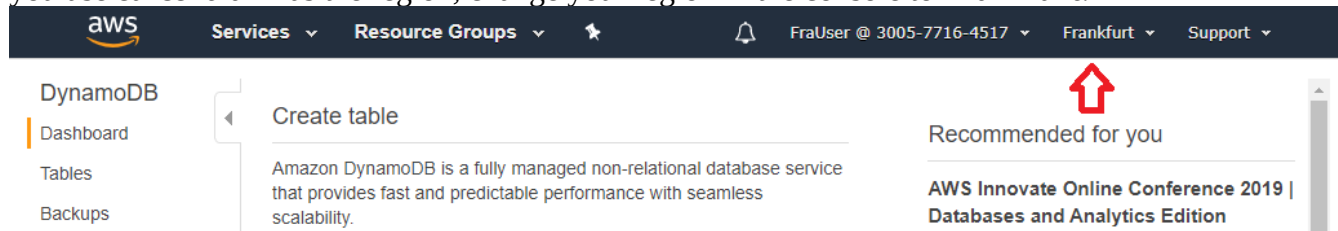
Frequently Asked Questions (FAQ) about the shop-1 example

List of questions:

- I used `./init-all.sh` to deploy the objects, but I don't see them in AWS...
- I used `./init-all.sh` to deploy the objects, but after a while I got an error... I used my own DNS domain to test with.
- I lost the long URL from AWS to use it as an URL for the client tools. Where can I find it back?
- I changed a Lambda function in one of the directories under `AMIS-Blog-AWS/shop-1/shop/lambda` and then tried to use the `init-shop.sh` script to upload the changed Python script to AWS. Why doesn't this work?
- I got a 500 error from the API gateway. What is going wrong?
- I got a 403 error from the API gateway. What am I doing wrong?
- I got a 200 message from `./encrypt-and-send.py`, but I don't see records with sold items in DynamoDB. What is going wrong?
- I destroyed my Vagrant box before destroying my AWS environment. Can I destroy the objects manually?
- You use a lot of objects – what will it cost me when I enroll and use your shop example?
- Can I use a free tier account to play along?

I used `./init-all.sh` to deploy the objects, but I don't see them in AWS...

Look at the region you are working in: is this the same region as you deployed your objects in? When you use **eu-central-1** as the region, change your region in the console to **Frankfurt**:



I used `./init-all.sh` to deploy the objects, but after a while I got an error... I used my own DNS domain to test with.

Probably, your request for a certificate is still pending. Go to Service > Certificate Manager and look at the state. When this is pending, you will have to wait until AWS does their checks.

In the mean time, you can continue using the shop example without a domain. Use the following commands again:

```
cd ~  
./init-all.sh
```

Answer “no” at the question if you own or want to use an own domain. When the certificate is issued, you can enroll again with “yes” and your own domain. AWS states, that issuing a public certificate can take hours or longer (<https://aws.amazon.com/certificate-manager/faqs/>), though with my domains (I tested it with a .nl and with a .org domain) this happened within minutes.

I lost the long URL from AWS to use it as an URL for the client tools. Where can I find it back?

The simplest way to get it back, is to go to the shop directory (`AMIS-Blog-AWS/shop-1/shop`) and type the command `../../terraform show`. The last lines two lines will be something like:

Outputs:

```
invoke_url = "https://cyk9ts3q31.execute-api.eu-west-1.amazonaws.com/prod/shop"
```

You can then use it by going to the client directory and using the `encrypt_and_send.py` script:

```
cd AMIS-Blog-AWS/shop-1/client
./encrypt_and_send AMIS1 https://cyk9ts3q31.execute-api.eu-west-1.amazonaws.com/prod/shop
```

I changed a Lambda function in one of the directories under AMIS-Blog-AWS/shop-1/shop/lambda and then tried to use the `init-shop.sh` script to upload the changed Python script to AWS. Why doesn't this work?

Terraform will not look at the timestamp of the zip file. So when there is a Lambda function with the correct name, it will assume that the content is also the same as it was.

Fastest solution: go to the shop directory and use the following commands to destroy all the shop objects and then redeploy the shop example:

```
cd AMIS-Blog-AWS/shop-1/shop
./destroy-shop.sh
./init-shop.sh
```

I got a 500 error from the API gateway. What is going wrong?

Every now and then, the API Gateway will raise a 500 error. When you look at the logs, you will see that this has something to do with permissions to run the Lambda. Or with permissions to get the key of the Lambda function. When you try again a few minutes later, this problem can either be gone or still be present. The problem is caused in the background of AWS: the deploy of IAM permissions isn't always successful or isn't always fast enough. The problem will be gone when you redeploy the environment:

```
cd ~
./destroy-all.sh
./init-all.sh
cd ~/AMIS-Blog-AWS/shop-1/client
<<./encrypt-and-send.py command with the parameters that are shown by the init-all.sh command>>
```

I got a 403 error from the API gateway. What am I doing wrong?

Probably, you forgot the `/shop` part of the URL:

```
[vagrant@localhost client]$ ./encrypt_and_send.py AMIS1 https://amis.retsema.eu
Shop id      = AMIS1
Key alias    = alias/KeyJ-AMIS1
URL          = https://amis.retsema.eu
Data         = {"shop_id": "AMIS1", "content_base64":
"k2/6G7NTdc7t52dMqHc8hIRiAmRc2Z87gOS3smp17+Y+6yKSXZRM+XBWBcshuqpmDDWI4R082Vex//
NEaea9dLb65P6maPJL19QnYk6BVlF8vvP4DRtOZmDtdPA19FdGZdr/
iGSYSFNCQJlX2r9EACBHI1iv87c7eftQ4YufqBj0o+PeYHByhq6FrISAvf/
1+0IvzutV5Z6jViQCSg99vy6ciC40LUSQYfyhFAA7sZIoeeG8EDVB5x9Q5In93DhXHONsyJNs26/
UVAUss8cwhkruBcf5Kvm4gE2+TM449wjEC/bx1uy/o+Jagpy07pPmjeTSuEBayNndpI/nUOuX9Q==" }
request_number = 1
Status code    = 403
Content        = b'{"message": "Missing Authentication Token"}'
```

Solution: add `/shop` to the URL:

```
[vagrant@localhost client]$ ./encrypt_and_send.py AMIS1 https://amis.retsema.eu/shop
Shop id      = AMIS1
Key alias    = alias/KeyJ-AMIS1
URL          = https://amis.retsema.eu/shop
```

```
Data          = {"shop_id": "AMIS1", "content_base64":
"ZPdNb+rliozwIUap20JkL0R3bYvvpZr9myvp1xYJANqPlV5M7+Ai7N0CrVI7dP7C/
pdgjpgRZAeCdChs+tkkrfpp8FTeyUsnu4WzEEU0Nqhc+6iZQtyaGoibEK4mctcMzrhfEI2wbbZuqVSm+XLogE0nxocQ
+Pj99DH6AQnI4xc9MBoSN+HFjst9hL7/
IhESYM10JyBYoxrfP40699fxhkp5piyN4DZpM2rxvw93PEmt7wQaPUNVP72amfXsrK21BZPpHypecx1BBSwkIaNE1Xu
OZar0BKx/KKd3LXZGvmZf0eFP2d91SdYHkaFLY3mmq6YKHwvBdHT0zb0WzMAS/g==" }
request_number = 1
Status code    = 200
Content        = b'"OK"'
```

I got a 200 message from ./encrypt-and-send.py, but I don't see records with sold items in DynamoDB. What is going wrong?

Every now and then, the SNS system is not able to deliver messages to Lambda functions. When you try again a few minutes later, this problem can either be gone or still be present. The problem is caused in the background of AWS: the deploy of IAM permissions isn't always successful or isn't always fast enough. The problem will be gone when you redeploy the environment:

```
cd ~
./destroy-all.sh
./init-all.sh
cd ~/AMIS-Blog-AWS/shop-1/client
<<./encrypt-and-send.py command with the parameters that are shown by the init-all.sh command>>
```

I destroyed my Vagrant box before destroying my AWS environment. Can I destroy the objects manually?

Yes, you can. This is a list of places where to look for objects:

init-cert.sh:

Certificate Manager > Certificate manager

Route53 > Hosted zones > your-domain-name >

- a CNAME record which is used to check if you owned the domain while requesting a certificate for example:

_737c89388ce141925aefa00b332f8de1.retsema.eu, with value
_37812d6ffa713cfcf8ebc1117fc2f271.nhqijqilxf.acm-validations.aws.

Please mind, that there might also be an A record amis.<your-domain-name> (f.e. amis.retsema.eu) in this domain, which is added by the init-shop.sh script.

Init-infra.sh:

IAM > policies (search for AMIS)
> roles (search for AMIS)

KMS > Customer managed keys

DynamoDB > Tables

init-shop.sh:

Please mind, that the objects in the API gateway have dependencies on each other. When you delete them in the order that they are below, then you shouldn't have problems:

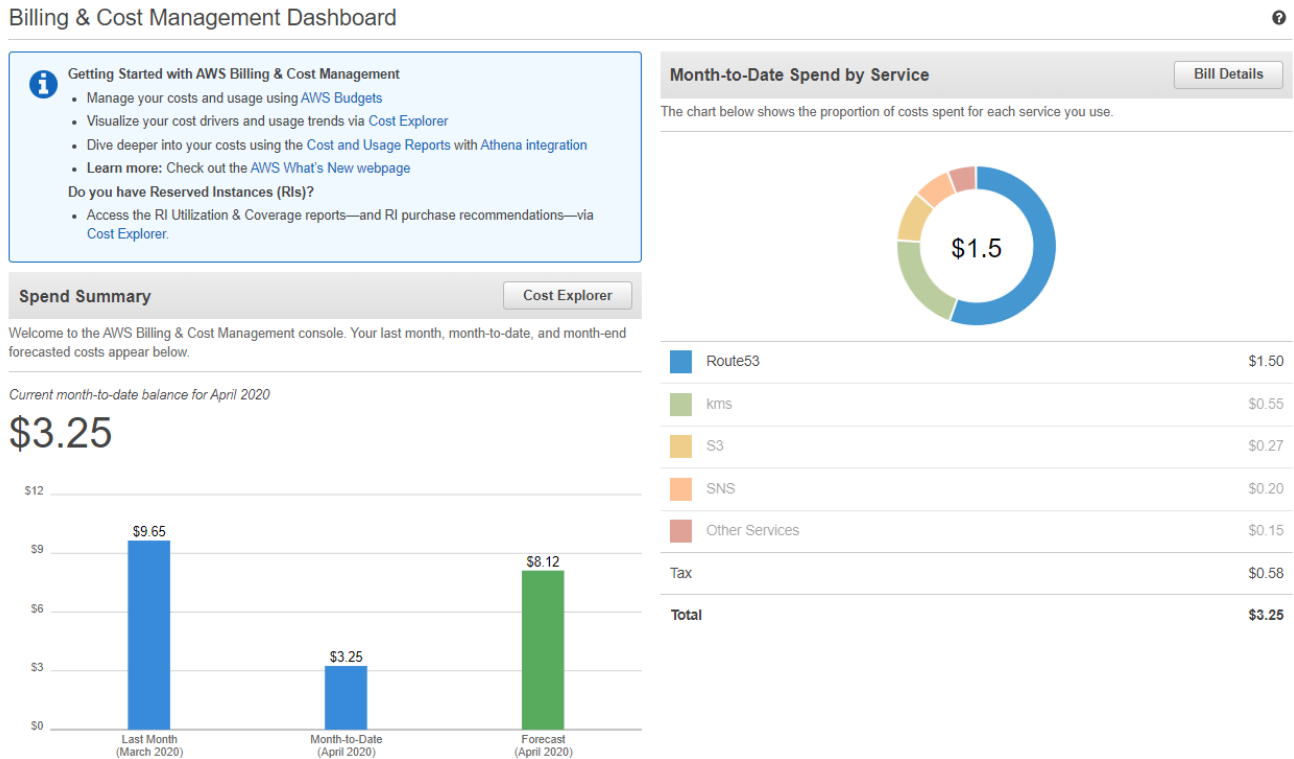
Route53 > Hosted zones > your-domain-name >

- an A record amis.your-domain-name. For example:

amis.retsema.eu, which has an alias target of d-7mbsbho886.execute-api.eu-west-1.amazonaws.com.
API Gateway > Custom Domain Names (if you used your own domain name)
> Resources
Lambda > APIs
SNS > Topics

You use a lot of objects – what will it cost me when I enroll and use your shop example?

I used AWS a lot to write this example and the blogs that are about this example. I can show you my bill: it's April 13 when I write this faq:



You see here, that most of my costs are for Route53 (I use three public domain names in Route53). I also use s3 for personal files, these costs have nothing to do with the shop example.

The costs of KMS are there because keys that are destroyed are not destroyed immediately, but after 7 – 30 days. On this moment, about 270 keys are waiting to be destroyed.

The costs for SNS and Other services (DynamoDB, Lambda) are spent on this example.

When I would continue to use AWS the same amount of time and resources as I am using it now, then the costs for this example shop will be around $(\$0.55 + \$0.20 + \$0.15) \times 30 / 12 = \$ 2,25$ per month. When you use this example, your costs will probably be much lower.

There is one way to keep the costs down: destroy all objects that you don't use anymore. So: every time you stop looking at this example, use the destroy-all.sh script in the home directory of the Vagrant user.

Can I use a free tier account to play along?

Yes, you can:

- The API gateway includes one million API calls
- Lambda provides 400 GB per second and one million requests for free
- The first million SNS push notifications are free
- DynamoDB offers 25GB of indexed storage, 25 write units and 25 read units
- Route53, Certificate Management don't offer free tier

When you look at this list, 25 write units and 25 read units are not that much, so you might have to pay a little bit. But see the answer to the previous question: you really have to do very much effort to get your monthly bill over \$2,25...