

1. Δημιουργία Postgres Βάσης Δεδομένων στο AWS

Αφού έχουμε κάνει login στην AWS console, επιλέγουμε την υπηρεσία RDS (Relational Database Service). Από το μενού αριστερά επιλέγουμε Databases.

Amazon RDS

Amazon Aurora
Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <\$1/day. Aurora supports up to 64TB of auto-scaling storage and up to 15 low-latency read replicas. [Learn more.](#)

[Create database](#)

Or, [Restore Aurora DB cluster from S3](#)

Resources [Refresh](#)

You are using the following Amazon RDS resources in the US East (N. Virginia) region (used/quota)

DB instances (1/40)	Parameter groups (1)
Allocated storage (20.00 GB/100.00 TB)	Default (1)
Click here to increase DB instances limit	Custom (0/100)
Reserved instances (0/40)	Option groups (1)
Snapshots (140)	Default (1)
Manual (0/100)	Custom (0/20)
Automated (4)	Subnet groups (1/50)
Recent events (4)	Supported platforms VPC
Event subscriptions (0/20)	Default network vpc-074af87d

Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

[Restore from S3](#) [Create database](#)

Note: your DB instances will launch in the US East (N. Virginia) region

Σε αυτή την οθόνη (RDS > Databases) βλέπουμε όλα τα instances βάσεων που έχουμε δημιουργήσει με τον λογαριασμό μας. Στο cloud ένα instance βάσης είναι ένα σύνολο πόρων που έχουν δεσμευτεί για την δημιουργία μιας βάσης δεδομένων. Προφανώς εφόσον δεν έχετε ακόμα δημιουργήσει κανένα Instance ο πίνακας είναι κενός.

Amazon RDS

Databases

[Filter databases](#)

[Group resources](#) [Modify](#) [Actions](#) [Restore from S3](#) [Create database](#)

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance
studentdb	Instance	PostgreSQL	us-east-1f	db.t2.micro	Available	4.50%	4 Connections	none

Επιλέγουμε Create database. Όπως βλέπετε, το AWS υποστηρίζει 6 διαφορετικά Engines βάσεων. Στις εργασίες του μαθήματος θα δουλέψουμε με Postgres επομένως επιλέγουμε αυτή.

Choose a database creation method [Info](#)

- ☒ **Standard Create**
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

- ☐ **Easy Create**
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

- ☐ Amazon Aurora



- ☐ MySQL



- ☐ MariaDB



- ☒ **PostgreSQL**



- ☐ Oracle



- ☐ Microsoft SQL Server



Version [Info](#)

PostgreSQL 10.4-R1



[i](#) If you want to create PostgreSQL 12 in the Preview environment, click [here](#).

Ακολουθως η Amazon μας ρωτάει εάν η βάση που θα φτιάξουμε θα είναι για κάποιο σύστημα που θα βγει στην παραγωγή ή θα είναι για δοκιμές και testing. Απαντάμε με το δεύτερο.

Templates

Choose a sample template to meet your use case.

☐ **Production**
Use defaults for high availability and fast, consistent performance.

☒ **Dev/Test**
This instance is intended for development use outside of a production environment.

☐ **Free tier**
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
[Info](#)

Ήρθε η ώρα να κάνουμε specify τα τεχνικά χαρακτηριστικά που θα έχει η βάση που θα δημιουργήσουμε. Για τις απαιτήσεις των εργασιών θα επιλέξουμε τα εξής:

- License model: postgresql-license
- DB engine version: PostgreSQL 10.4-R1
- DB instance class: db.t2.micro
- Multi-AZ deployment: No
- Storage type: General Purpose (SSD)
- Allocated storage: 20 GiB

Εν ολίγοις εκτός της DB instance class (όπου επιλέγουμε το πιο οικονομικό σε πόρους instance) αφήνουμε τις default τιμές.

DB instance size

DB instance class [Info](#)

Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

☐ Standard classes (includes m classes)

☐ Memory Optimized classes (includes r and x classes)

☒ Burstable classes (includes t classes)

db.t2.micro

1 vCPUs 1 GiB RAM Not EBS Optimized

☐ Include previous generation classes

Storage

Storage type [Info](#)

General Purpose (SSD) ▼

Allocated storage

20

GiB

(Minimum: 20 GiB, Maximum: 16384 GiB) Higher allocated storage **may improve** IOPS performance.

ⓘ Provisioning less than 100 GiB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. [Learn more](#) [↗](#)

Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application's needs.

☒ Enable storage autoscaling

Enabling this feature will allow the storage to increase once the specified threshold is exceeded.

Maximum storage threshold [Info](#)

Charges will apply when your database autoscales to the specified threshold

1000

GiB

Minimum: 21 GiB, Maximum: 16384 GiB

Availability & durability

Multi-AZ deployment [Info](#)

- ☐ Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
- ☒ Do not create a standby instance

Πιο κάτω στην φόρμα καλούμαστε να συμπληρώσουμε το όνομα του DB instance, τον master user της βάσης σας και το password του.

Settings

DB Instance Identifier [Info](#)
Specify a name that is unique for all DB instances owned by your AWS account in the current region.

InstructionsDB

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance". Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.

Master username [Info](#)
Specify an alphanumeric string that defines the login ID for the master user.

InstUser

Master Username must start with a letter. Must contain 1 to 63 alphanumeric characters.

Master password [Info](#) **Confirm password** [Info](#)

Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", " ", or "@".

Cancel Previous Next

Πατάμε Next και φτάνουμε σε ακόμη μία φόρμα προσδιορισμού settings. Η πρώτη ομάδα ρυθμίσεων έχει να κάνει με το Δίκτυο μέσα στο οποίο θα βρίσκεται το instance. Επιλέγουμε τις default ρυθμίσεις. Για να μπορούμε να έχουμε πρόσβαση στην βάση μας εκτός του εικονικού δικτύου πρέπει στο **Public accessibility** να είναι επιλεγμένο το **Yes**.

Connectivity

Virtual Private Cloud (VPC) [Info](#)
VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-4f506c35) ▼

Only VPCs with a corresponding DB subnet group are listed.

ⓘ

After a database is created, you can't change the VPC selection.

▼ Additional connectivity configuration

Subnet group [Info](#)
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default-vpc-4f506c35 ▼

Publicly accessible [Info](#)

☒ Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☐ No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group

Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

☒ Choose existing
Choose existing VPC security groups

☐ Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

default ✕

Availability zone [Info](#)

No preference ▼

Database port [Info](#)
TCP/IP port the database will use for application connections.

5432

5

Database authentication

Database authentication options [Info](#)

- ☒ Password authentication
Authenticates using database passwords.
- ☐ Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

Παρακάτω συμπληρώνουμε τα στοιχεία της πρώτης βάσης που θα φτιαχτεί μέσα στο instance μας.

Database options

Database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

Port [Info](#)

TCP/IP port the DB instance will use for application connections.

DB parameter group [Info](#)

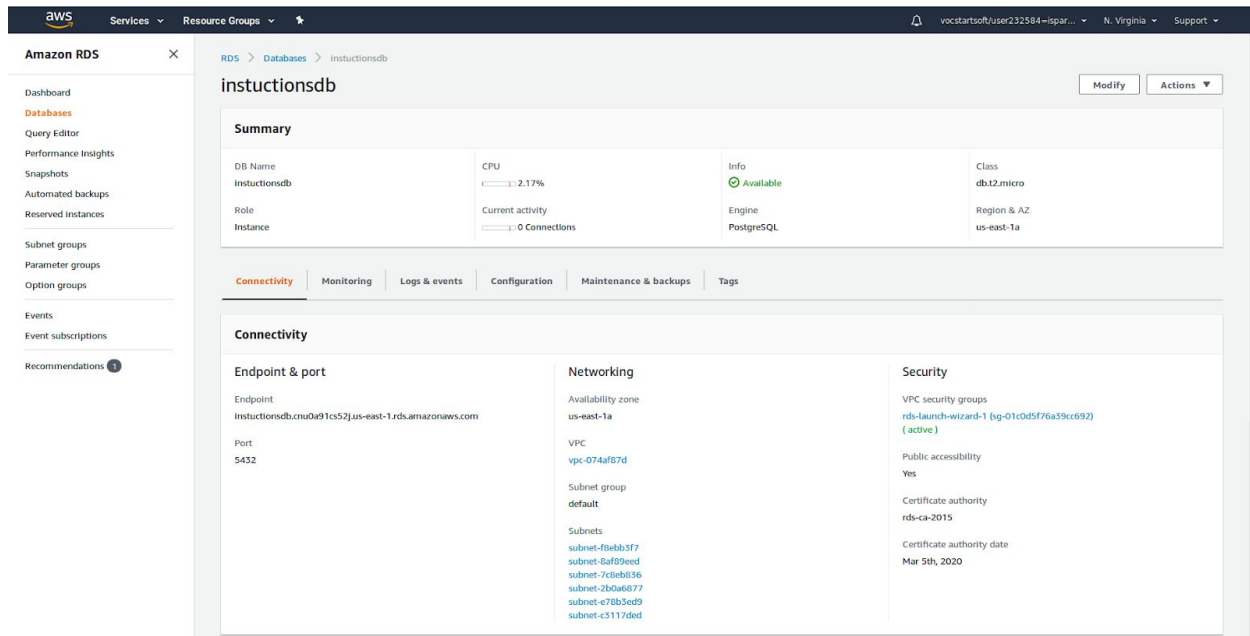
Option group [Info](#)

IAM DB authentication [Info](#)

☐ Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.

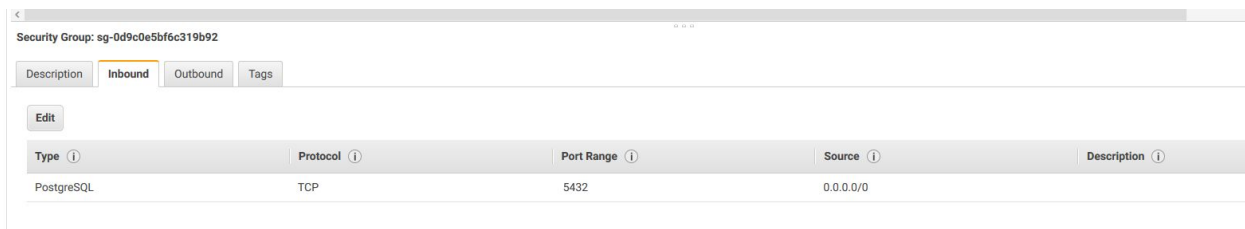
☒ Disable

Για τα υπόλοιπα αφήνουμε τις προεπιλεγμένες τιμές και πατάμε Create database. Είναι λογικό και αναμενόμενο η δημιουργία του DB instance να κάνει λίγα λεπτά για να ολοκληρωθεί. Όταν είναι έτοιμο, τότε θα έχει προστεθεί στον RDS > Databases πίνακα. Κάνουμε κλικ στον DB identifier του και οδηγούμαστε στη σελίδα με τα στοιχεία του (RDS > Databases > *db_identifier*).



Για να μπορούμε να έχουμε πρόσβαση στην βάση από τον υπολογιστή μας πρέπει να εξασφαλίσουμε τα εξής:

- A. Το **Public accessibility** να είναι **'Yes'**
- B. Στην οθόνη που οδηγούμαστε πατώντας στο link **VPC security groups** (βρίσκεται ακριβώς πάνω από το Public accessibility), στο tab **'Inbound'** η επιλογή στο Source να είναι **Custom '0.0.0.0/0'** ή **Anywhere** (οπότε συμπληρώνεται μόνο του). Αυτό σημαίνει να δέχεται συνδέσεις από όλες τις IP διευθύνσεις.



2. Σύνδεση σε Postgres Βάση και διαχείριση αυτής από τον υπολογιστή μας

Εάν ακολουθήσουμε επιτυχώς τα παραπάνω βήματα, τότε θα έχει δημιουργηθεί στο Amazon Cloud ένα db instance με μία postgres βάση δεδομένων. Προφανώς βάσεις δεδομένων μπορούμε να δημιουργούμε και τοπικά στον υπολογιστή μας. Τα βασικά πλεονεκτήματα της χρήση του Amazon RDS είναι ότι δεν ξοδεύουμε πόρους του δικού μας υπολογιστή για αυτό το σκοπό καθώς και το ότι με ελάχιστο διαχειριστικό κόπο συντηρείται η βάση στο Cloud και έχουμε την δυνατότητα πρόσβασης από οπουδήποτε.

Στην σελίδα RDS > Databases > *db_identifier* με τα στοιχεία της βάσης την οποία θέλουμε να διαχειριστούμε παρατηρούμε στο section Connectivity το Endpoint. Το Endpoint είναι το domain name το οποίο φαίνεται στον “έξω κόσμο” και θα χρησιμοποιήσουμε για να συνδεθούμε στην βάση μας.

Ένας τρόπος να συνδεθούμε σε αυτή είναι από την γραμμή εντολών μας με την *psql*¹. Για να γίνει αυτό πρέπει πρώτα να έχουμε εγκαταστήσει στον υπολογιστή μας την *postgres*². Εφόσον γίνει αυτό (και εφόσον έχουμε ενημερώσει κατάλληλα την *environment variable Path*³), για να συνδεθούμε με την βάση που έχουμε δημιουργήσει δίνουμε την εντολή στην γραμμή εντολών:

```
psql -h name_of_the_endpoint -p port_number -U db_user_name -d database_name
```

Για παράδειγμα, για την βάση που φτιάξαμε πριν θα δώσουμε την εντολή:

```
kostis@kostis-lab-pc:~$ psql -h instuctionsdb.cnv0a91cs52j.us-east-1.rds.amazonaws.com -p 5432 -U instUser -d instActualDB
Password for user instUser:
psql (9.5.14, server 10.4)
WARNING: psql major version 9.5, server major version 10.
         Some psql features might not work.
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

instActualDB=>
```

Ο *psql* client μας βρίσκει την απομακρυσμένη βάση η οποία ζητάει password. Εάν το δώσουμε σωστά έχουμε πρόσβαση στην βάση μας.

Για να δουλεύουμε ευκολότερα την *postgres* βάση δεδομένων υπάρχουν αρκετά συστήματα διαχείρισης, με το δημοφιλέστερο -ίσως- εξ’ αυτών το *pgAdmin*⁴. Για να συνδεθούμε με το *pgAdmin* στην Cloud βάση (ομοίως με πριν πρέπει πρώτα να το εγκαταστήσουμε) πατάμε δεξί κλικ στους Servers και επιλέγουμε Create > Server.

¹ <http://postgresguide.com/utilities/psql.html>

² <http://postgresguide.com/setup/install.html>

³ Στα Windows όταν έχουμε εγκαταστήσει ένα πρόγραμμα και θέλουμε να το καλούμε από την γραμμή εντολών από όλα τα folders, πρέπει να πούμε στον υπολογιστή που βρίσκεται το εκτελέσιμο. Έτσι και με την *psql*. Το εκτελέσιμο αρχείο της *psql*, μαζί με τα υπόλοιπα της *Postgres*, βρίσκεται συνήθως στο C:\Program Files\PostgreSQL\10\bin. Παίρνουμε αυτό το path και πάμε στις *Environment variables* των Windows μας. Το προσθέτουμε στην μεταβλητή *Path*. Με αυτόν τον τρόπο δίνουμε στο cmd από όπου κι αν είμαστε την εντολή *psql* κανονικά. –Εάν δεν φτιάχναμε την *Path* θα καλούσαμε την *psql* μόνο από τον folder bin...--

⁴ <https://www.pgadmin.org/download/>

Create - Server

General Connection SSL SSH Tunnel Advanced

Name: whateva name

Server group: Servers

Background: ☐

Foreground: ☐

Connect now?: ☒

Comments:

Either Host name, Address or Service must be specified.

Cancel Reset Save

Στην ουσία δεν φτιάχνουμε έναν νέο server αλλά μία νέα σύνδεση στον υπάρχοντα db server που έχουμε ήδη φτιάξει στο AWS RDS. Το όνομα στο tab General που θα βάλουμε δεν έχει ιδιαίτερη σημασία. Αυτό που έχει σημασία και είναι απαραίτητο είναι να συμπληρώσουμε σωστά στο tab Connection το Host name/address πεδίο όπου βάζουμε το end point μας, την θύρα όπου “ακούει” η βάση μας, το Username του χρήστη της βάσης και το password του.

Create - Server

General **Connection** SSL SSH Tunnel Advanced

Host name/address: tionsdb.cnu0a91cs52j.us-east-1.rds.amazonaws.com

Port: 5432

Maintenance database: postgres

Username: instUser

Password: [masked]

Save password? ☐

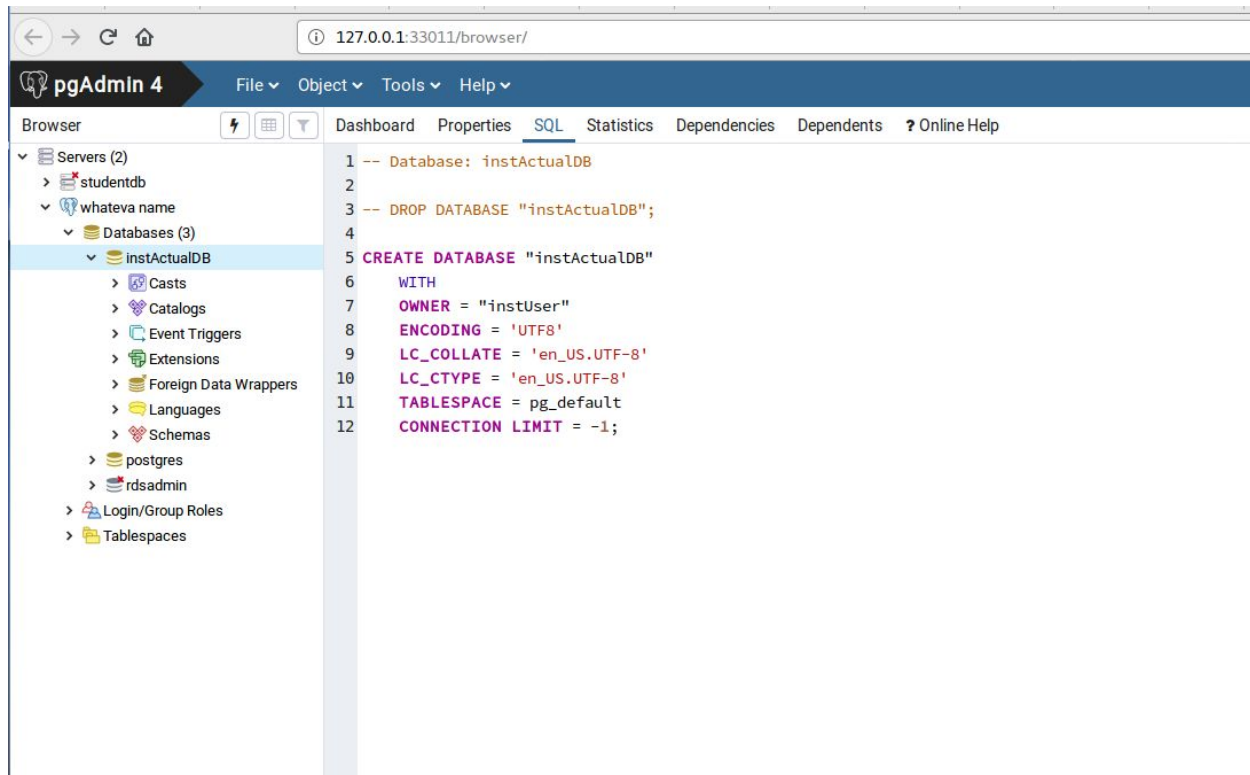
Role: [empty]

Service: [empty]

[i] [?] [Cancel] [Reset] [Save]

ΠΡΟΣΟΧΗ: Μην μπερδεύετε τον user της AWS πλατφόρμας με τον χρήστη της βάσης. Ο AWS user είναι σαν admin του όλου συστήματος και αυτό που κάνει είναι να δεσμεύει πόρους (χώρο στον δίσκο, CPUs, μνήμη κλπ) για το στήσιμο db instances και βέβαια για την παροχή όλων των άλλων υπηρεσιών στο AWS. Ο user της βάσης, από την άλλη, είναι αυτός που -αναλόγως των δικαιωμάτων του- φτιάχνει tables, κάνει insert δεδομένα, ρωτάει queries κλπ.

Εφόσον τα στοιχεία που δώσαμε είναι σωστά, το pgAdmin έχει αποκτήσει πρόσβαση στην Cloud βάση μας.



3. Δημιουργία χρηστών σε Postgres Βάση

Πέρα από τον master user μιας βάσης, στην πορεία υπάρχει συχνά η ανάγκη για δημιουργία και άλλων χρηστών με διαφορετικά δικαιώματα. Στην postgres υπάρχει η έννοια του ρόλου. Ο κάθε ρόλος ορίζεται ως ένα σύνολο δικαιωμάτων στην βάση. Ακολουθούν οι εντολές που θα πρέπει να δοθούν ώστε να δημιουργηθεί ένας read-only ρόλος και αυτός να εκχωρηθεί σε έναν νέο χρήστη της βάσης. Αυτό θα μας χρησιμεύσει και στις εργασίες καθότι για να σας βαθμολογήσουμε θα πρέπει να έχουμε πρόσβαση στην βάση που θα έχετε φτιάξει χωρίς όμως να έχουμε την δυνατότητα αλλαγών. Τις εντολές που φαίνονται στο screenshot τις δίνετε στην psql ή στο pgadmin εφόσον είστε ήδη συνδεδεμένοι στη βάση σας.

```

instActualDB=> CREATE ROLE examiner WITH LOGIN;
CREATE ROLE
instActualDB=> GRANT CONNECT ON DATABASE "instActualDB" TO examiner;
GRANT
instActualDB=> GRANT USAGE ON SCHEMA public TO examiner;
GRANT
instActualDB=> GRANT SELECT ON ALL TABLES IN SCHEMA public TO examiner;
GRANT
instActualDB=> \password examiner;
Enter new password:
Enter it again:
instActualDB=>

```

Προκειμένου να μπορεί ο εξεταστής να συνδεθεί στην βάση σας θα χρειαστεί να δώσετε στον αντίστοιχο ρόλο ένα password.