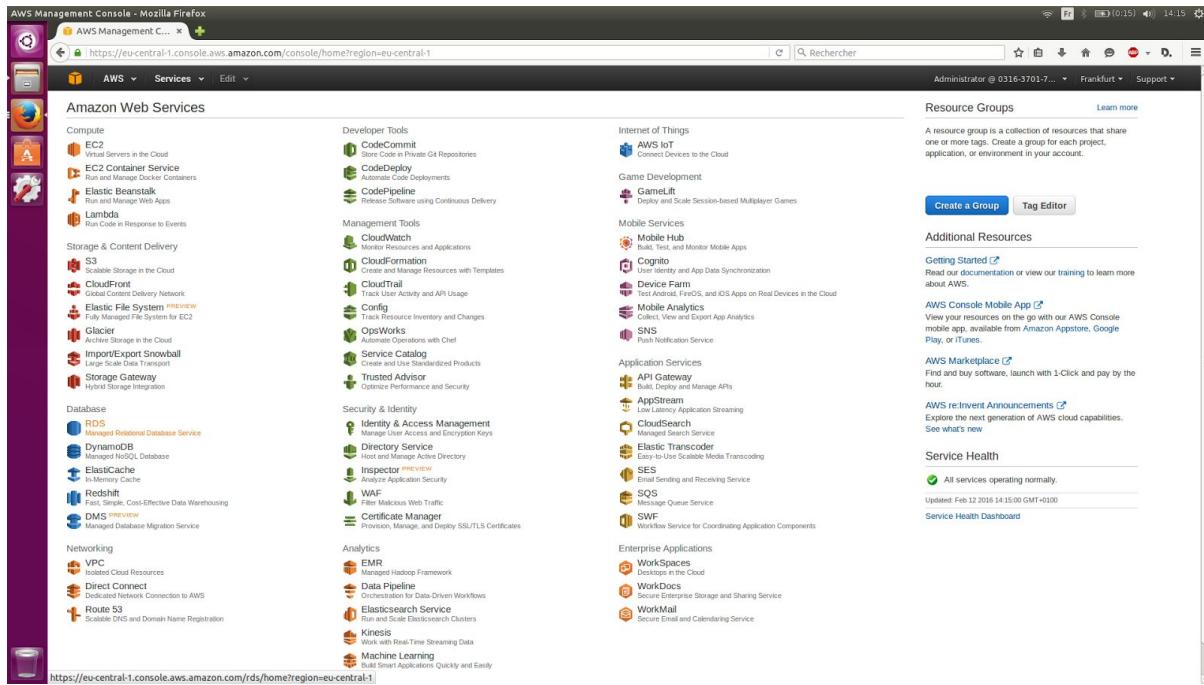


HOWTO: Setup AWS RDS

Create an AWS account and login to your AWS Console



1 - VPC configuration

[Create a new VPC](#)

[Internet Gateway](#)

[Then you need to create subnets.](#)

2- RDS configuration

[Create your RDS Instance](#)

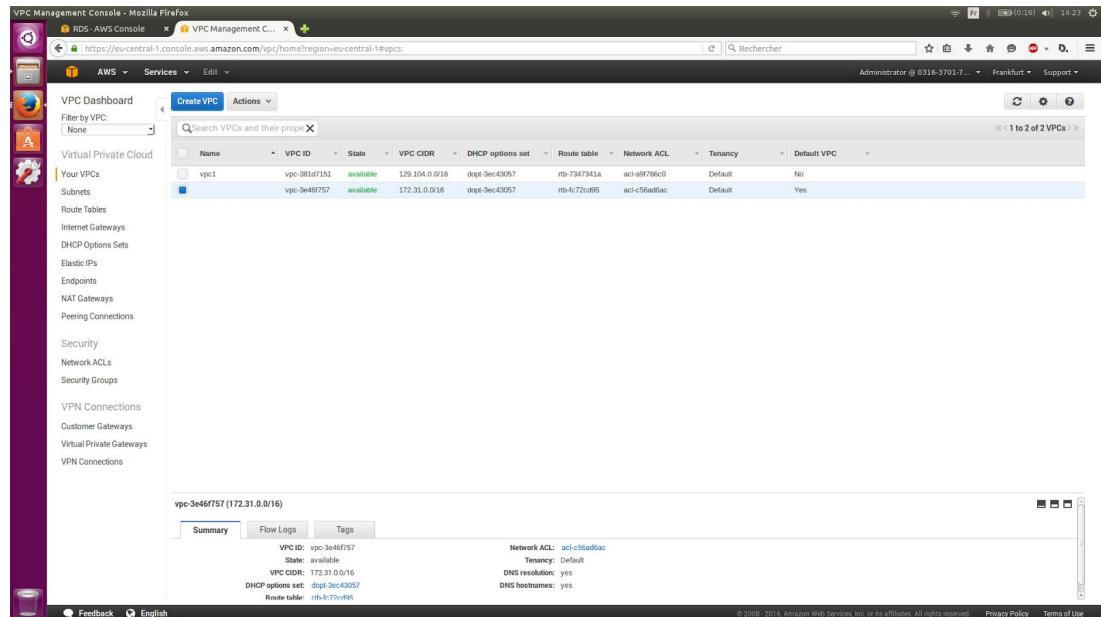
[Configure your security group to access your instance](#)

[Log in to your instance with a command line tool](#)

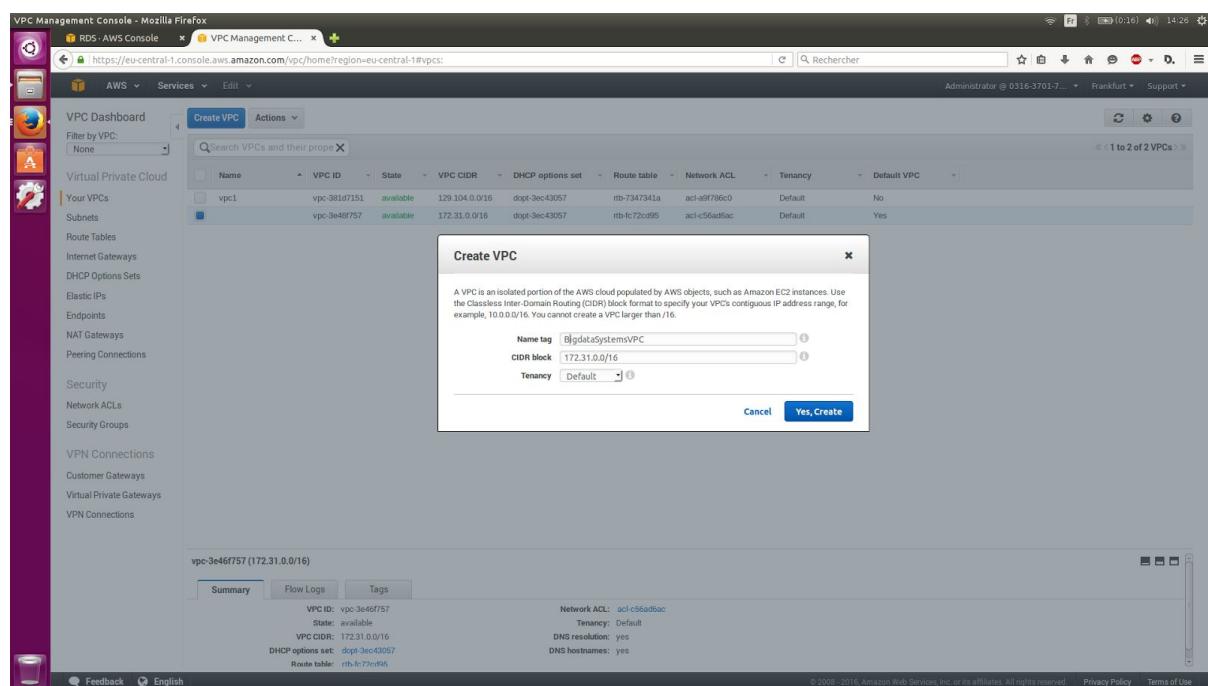
1 - VPC configuration

VPC is a virtual private network. It allows you to define a range of private IP addresses that will be given to the different tools that you'll plug in this network. For instance the default VPC has the IP range of 172.31.0.0/16. This means that your AWS Services that will run on this VPC will use one of the 2^{16} private IP addresses from 172.31.0.0 to 172.31.255.255.

IF YOU STILL HAVE YOUR DEFAULT VPC, EVERYTHING SHOULD BE FINE. You can still read this to get a better understanding and be able to debug if you have issues.



Create a new VPC



You need to activate DNS Hostname.

The screenshot shows the AWS VPC Management Console interface. In the top navigation bar, the URL is https://eu-central-1.console.aws.amazon.com/vpc/home?region=eu-central-1#vpcs;. The main content area displays a table of VPCs. One row for 'BigdataSystemsVPC' is selected, and a context menu is open over it, showing options like 'Delete VPC', 'Edit DHCP Options Set', 'Edit DNS Resolution', and 'Create Flow Log'. Below the table, a specific VPC summary is shown for 'vpc-39caa150 (172.31.0.0/16) | BigdataSystemsVPC'. The 'Summary' tab is selected, showing details such as VPC ID, State, Network ACL, and DNS resolution settings. A modal dialog box titled 'Edit DNS Hostnames' is overlaid on the page. It contains a single input field labeled 'DNS Hostnames' with two radio button options: 'Yes' (selected) and 'No'. At the bottom of the dialog are 'Cancel' and 'SAVE' buttons. The background of the modal is semi-transparent, allowing the underlying VPC summary to be partially visible.

Internet Gateway

You need to plug your VPC to internet by using an internet gateway. This basically says your network where is internet.

The screenshot shows the AWS VPC Dashboard. A search bar at the top contains the text 'Big'. Below it, a table lists VPCs. One VPC, 'BigdataSystemsVPC', is highlighted with a blue border. The table columns include Name, VPC ID, State, VPC CIDR, DHCP options set, Route table, Network ACL, Tenancy, and Default VPC.

The sidebar on the left shows navigation links for VPC Dashboard, Your VPCs, Subnets, Route Tables, Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and Peering Connections. The 'Internet Gateways' link is currently selected.

This screenshot shows the 'Create Internet Gateway' dialog box. It displays an instruction: 'An Internet gateway is a virtual router that connects a VPC to the Internet.' Below this is a 'Name tag' input field containing 'BigdataSystemIG'. At the bottom right are 'Cancel' and 'Yes, Create' buttons.

Then you need to attach it to your VPC

This screenshot shows the 'Attach to VPC' dialog box. It displays an instruction: 'Attach an Internet gateway to a VPC to enable communication with the Internet.' Below this is a 'VPC' dropdown menu showing 'vpc-39caa150 (172.31.0.0/16) | BigdataSystem...'. At the bottom right are 'Cancel' and 'Yes, Attach' buttons.

Then you need to create subnets.

Name	Subnet ID	State	VPC	CIDR	Available IPs	Availability Zone	Route Table	Network ACL	Default Subnet	Auto-assign Public IP
sub1	subnet-4ff55294	available	vpc-3e46f757 (172.31.0.0/16)	172.31.16.0/20	4096	eu-central-1b	rtb-fc72cd95	acl-c56ad5ac	Yes	Yes
sub2	subnet-7f36a04	available	vpc-381d7151 (129.104.0.0/16)	129.104.103.0/24	250	eu-central-1b	rtb-7347341a	acl-a9f786c0	No	Yes
sub3	subnet-b6ddaa7d	available	vpc-381d7151 (129.104.0.0/16)	129.104.0.0/24	250	eu-central-1a	rtb-7347341a	acl-a9f786c0	No	No
sub4	subnet-d56ee9bc	available	vpc-3e46f757 (172.31.0.0/16)	172.31.0.0/20	4096	eu-central-1a	rtb-fc72cd95	acl-c56ad5ac	Yes	Yes

You need two subnets in different timezones to create a RDS DB. Here we are working in Europe so there is only two possibilities.

First subnet :

Create Subnet

Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and a /28 netmask. Also, note that a subnet can be the same size as your VPC.

Name tag: BigdataSystemSubnet1

VPC: vpc-39caa150 (172.31.0.0/16) | BigdataSystem...

Availability Zone: eu-central-1a

CIDR block: 172.31.0.0/20

Cancel **Yes, Create**

And when you have two of them :

Name	Subnet ID	State	VPC	CIDR	Available IPs	Availability Zone	Route Table	Network ACL	Default	Auto-assign Publ
BigdataSystemSubnet1	subnet-81fabae8	available	vpc-39caa150 (172.31.0.0/16)	172.31.0.0/20	4091	eu-central-1a	rtb-b4d3aadd	acl-bc205ed5	No	No
BigdataSystemSubnet2	subnet-41dfb23a	available	vpc-39caa150 (172.31.0.0/16)	172.31.16.0/20	4091	eu-central-1b	rtb-b4d3aadd	acl-bc205ed5	No	No

subnet-81fabae8 (172.31.0.0/20) | BigdataSystemSubnet1

Summary **Route Table** **Network ACL** **Flow Logs** **Tags**

Subnet ID: subnet-81fabae8
BigdataSystemSubnet1
CIDR: 172.31.0.0/20
State: available
VPC: vpc-39caa150 (172.31.0.0/16)

Availability Zone: eu-central-1a
Route table: rtb-b4d3aadd
Network ACL: acl-bc205ed5
Default subnet: no

You should enable them to auto-assign public IP

Public IP are directly available from internet. You use the private IP to communicate within a VPC between different AWS tools you set up but you need to have a public IP to communicate with the outside world.

Modify Auto-Assign Public IP

Enable auto-assign public IP to automatically request a public IP address for instances launched into this subnet.

Enable auto-assign Public IP

Note: You can override the auto-assign public IP setting for each individual instance at launch time. Regardless of how you've configured the auto-assign public IP feature, you can assign a public IP address to an instance that has a single, new network interface with a device index of eth0.

Cancel **Save**

subnet-81fabae8 (172.31.0.0/20) | BigdataSystemSubnet1

Summary **Route Table** **Network ACL** **Flow Logs** **Tags**

Subnet ID: subnet-81fabae8
BigdataSystemSubnet1
CIDR: 172.31.0.0/20
State: available
VPC: vpc-39caa150 (172.31.0.0/16)

Availability Zone: eu-central-1a
Route table: rtb-b4d3aadd
Network ACL: acl-bc205ed5
Default subnet: no

Route table

The route table allows you to restrict communication within your VPC. You should configure them to allow the traffic to go to Internet.

You can find your route table by selecting one of your subnet and looking at his description at the bottom of your screen. Click on the route table (starting with "rtb-").

Name	Subnet ID	State	VPC	CIDR	Available IPs	Availability Zone	Route Table	Network ACL	Default	Auto-assign Publ
BigdataSystemSubnet2	subnet-41df5b23a	available	vpc-39caa150 (172.31.0.0/16) BigdataSystemsVPC	172.31.16.0/20	4091	eu-central-1b	rtb-b4d3aadd	acl-bc205ed5	No	Yes
BigdataSystemSubnet1	subnet-81f1abae8	available	vpc-39caa150 (172.31.0.0/16) BigdataSystemsVPC	172.31.0.0/20	4091	eu-central-1a	rtb-b4d3aadd	acl-bc205ed5	No	Yes

You need to edit the routes table.

Name	Route Table ID	Explicitly Associated	Main	VPC
rtb-b4d3aadd	0 Subnets	Yes		vpc-39caa150 (172.31.0.0/16) BigdataSystemsVPC

And add a new route from 0.0.0.0/0 (meaning everywhere), to your internet gateway (just type "igw" in the text field and they should automatically propose you the one attached to your VPC).

rtb-b4d3aadd

Destination	Target	Status	Propagated	Remove
172.31.0.0/16	local	Active	No	

[Add another route](#)

rtb-b4d3aadd

Destination	Target	Status	Propagated	Remove
172.31.0.0/16	local	Active	No	

[Add another route](#)

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

2- RDS configuration

RDS is a tool that let you deploy a database on demand quite easily.

Create your RDS Instance

Go to your console and select RDS

The screenshot shows the AWS RDS console in Mozilla Firefox. The left sidebar has links for RDS Dashboard, Instances, Reserved Purchases, Snapshots, Security Groups, Parameter Groups, Option Groups, Subnet Groups, Events, Event Subscriptions, and Notifications. The main content area displays information about Amazon Aurora (MySQL-compatible) and resources like DB Instances, Parameter Groups, and Option Groups. A warning message states "Aurora is not available in this region (EU Frankfurt). Use the region selector to switch to a supported region." Below this, there's a "Create Instance" section with a "Launch a DB Instance" button, a note about launching in the EU Frankfurt region, and a "Service Health" table showing the service is operating normally. The right sidebar contains links for Getting Started with RDS, Overview and Features, Documentation, Articles and Tutorials, Data import guide for MySQL, Data import guide for Oracle, Data import guide for SQL Server, Pricing, and Forums.

The screenshot shows the "Step 1: Select Engine" screen of the RDS instance creation wizard. It asks to choose a DB Engine below and click "Select". It lists MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server. MySQL is selected and highlighted. The "PostgreSQL" option is shown with its icon and name. A "Select" button is visible. A "Cancel" button is at the bottom right.

Launch a new PostgreSQL instance

We want a **free** instance

Step 1: Select Engine
Step 2: Production?
Step 3: Specify DB Details
Step 4: Configure Advanced Settings

Your current selection is eligible for the free tier.
[Learn More.](#)

Estimate your monthly costs for the DB instance using the [RDS Instance Cost Calculator](#).

Specify DB Details

Instance Specifications

DB Engine: postgres **License Model:** postgresql-license
DB Engine Version: 9.4.5 **DB Instance Class:** db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ Deployment: No
Storage Type: General Purpose (SSD)
Allocated Storage: 20 GB

Settings

DB Instance Identifier*: BidataSystemRDS
Master Username*: whatever
Master Password*: *****
Confirm Password*: *****

Specify a string that defines the password for the master user. Master Password must be at least eight characters long, as in 'MyPassword'.

*Required Cancel Previous Next Step

Here, be sure to get:

- the smallest deployment
- no A-Z deployment
- 20Gb of space

to keep it **free**.

Step 1: Select Engine
Step 2: Production?
Step 3: Specify DB Details
Step 4: Configure Advanced Settings

This instance will be created with the new Certificate Authority rds-ca-2015. If you are using SSL to connect to this instance, you should use the new certificate bundle. [Learn more here](#)

Network & Security

VPC: BigdataSystemsVPC (vpc-39caaf50) **Subnet Group:** Create new DB Subnet Group
Publicly Accessible: Yes **Availability Zone:** No Preference
VPC Security Group(s): Create new Security Group default (VPC)

Database Options

Database Name: bigdata **Database Port:** 5432
DB Parameter Group: default.postgres9.4

Specify a string of up to 8 alphanumeric characters that define the name given to a database that Amazon RDS creates when it creates the DB instance, as in 'myapp'. You can specify a database name. Amazon RDS does not create a database when it creates the DB instance.

Here:

- pick your VPC (default if you have one)
- select “create a new subnet group”
- you need your DB to be publicly accessible
- choose to create a new security group

Step 1: Select Engine
Step 2: Production?
Step 3: Specify DB Details
Step 4: Configure Advanced Settings

The selected Engine or DB Instance Class does not support storage encryption.

Network & Security

VPC: BigdataSystemsVPC (vpc-39caaf50) **Subnet Group:** Create new DB Subnet Group
Publicly Accessible: Yes **Availability Zone:** No Preference
VPC Security Group(s): Create new Security Group default (VPC)

Database Options

Database Name: bigdata **Database Port:** 5432
DB Parameter Group: default.postgres9.4 **Option Group:** default.postgres-9-4
Copy Tags To Snapshots: **Enable Encryption:**

The selected Engine or DB Instance Class does not support storage encryption.

Backup

Backup Retention Period: 7 days **Backup Window:** No Preference

The daily time range (in UTC) during which automated backups are created if automated backups are enabled.

Maintenance

Auto Minor Version Upgrade: Yes **Maintenance Window:** No Preference

*Required Cancel Previous **Launch DB Instance**

here choose the name of your database and keep default value for the rest (or read the documentation if you want to do some tuning).

Then you should be able to see this

The screenshot shows the AWS RDS Dashboard. On the left, there's a sidebar with options like Instances, Reserved Purchases, Snapshots, Security Groups, and Parameter Groups. The main area has tabs for Launch DB Instance, Show Monitoring, and Instance Actions. A search bar at the top says 'Filter: All Instances' and contains the text 'Q bida'. Below it, a table lists one instance: 'PostgreSQL bidatasystemrds' with status 'creating'. The table includes columns for Engine, Status, CPU, Current Activity, Maintenance, Class, VPC, Multi-AZ, Replication Role, and Encrypted.

This should take a while (like up to 15 minutes so you can go get a coffee break).

Configure your security group to access your instance

A Security Group is an AWS firewall that restricts the communication of the corresponding tool by setting some rules. In theory, it is better to restrict the access to a maximum, but since this is not a security class, we'll go with the easy solution. Don't do that in a professional environment.

Once your database is launched, go to your security group (you should put your cursor on the small "i") and click on the link that appears in my example as "rds-launch-wizard-3".

The screenshot shows the AWS RDS Dashboard with a PostgreSQL instance named 'bidatasystemrds' in the 'backing-up' state. A context menu is open over the instance, with a red box highlighting the 'rds-launch-wizard-3' option. The menu also includes 'Edit', 'Logs', and 'Tags'. The main dashboard shows monitoring data for CPU, Memory, and Storage, and a list of recent events.

Then we want to add a rule that'll allow us to connect from anywhere to our RDS instance. So click on the Inbound tab at the bottom and edit the rules

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. A search bar at the top right contains the text 'Rechercher'. The main area displays a table of security groups, with one row highlighted for 'sg-f4d5e19d'. Below this, a detailed view for 'sg-f4d5e19d' is shown, specifically the 'Inbound' tab of the 'Edit' dialog. The dialog lists an inbound rule for PostgreSQL on port 5432, with the source set to '92.156.69.224/32'.

Since most of the internet access providers gives floating IP, you don't want a single IP to be able to connect to your cluster so let's be extra generous and add the following rule.

The screenshot shows the 'Edit inbound rules' dialog box. It contains two entries:

Type	Protocol	Port Range	Source
PostgreSQL	TCP	5432	Custom IP: 156.69.224/32
All traffic	All	0 - 65535	Anywhere: 0.0.0.0/0

At the bottom of the dialog are 'Add Rule', 'Cancel', and 'Save' buttons.

Again, this is usually a bad idea but it's okay since it's just for a short time try. If you want to keep your DB running once the class is over, you should consider being more restrictive

Wait a bit to let everything to update (it's really quick so most of the time you don't have to wait but sometimes there is a delay)

Log in to your instance with a command line tool

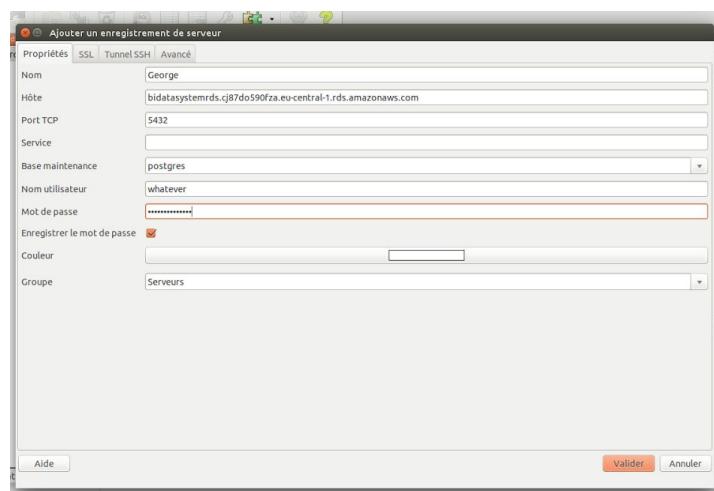
On the RDS instance view, you can select your instance and go to a detailed view of your database

You can use those information to log either using a command line with the following command (you'll have to install it first depending on your Unix distribution).

```
burgq@HPoux:~ 205x57
burgq@HPoux:~$ psql -h bidatasystemrds.cj87do590fza.eu-central-1.rds.amazonaws.com -p 5432 -U whatever -d bigdata
Password for user whatever:
psql (9.4.6, server 9.4.5)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

bigdata=>
```

Or with pgAdminIII if you're not used to command lines tool. It's working on most systems.



Congrat's you have a running DB. If you can't connect, two option :

- Your VPC is not well configured so please go through the first part of this HOWTO to check if everything is okay

- Your internet service provider is somehow blocking the protocol or the port (possible if you're using a free wifi access, not if you're using your own box to connect to internet).

3- EC2 configuration

We are going to add a small server on amazon webservices. We will use a free instance. That will allow us to download a “large” dataset and load it on your previously created RDS instance.

Go to the EC2 console

The screenshot shows the AWS EC2 Management Console dashboard. On the left, there's a sidebar with links for Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Commands, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Load Balancing, and Auto Scaling. The main area shows the following statistics:

- Resources:**
 - 3 Running Instances
 - 0 Dedicated Hosts
 - 3 Volumes
 - 2 Key Pairs
 - 0 Placement Groups
- Account Attributes:**
 - Supported Platforms: VPC
 - Default VPC: vpo-3e6d7757
 - Resource ID length management
- Additional Information:**
 - Getting Started Guide
 - Documentation
 - All EC2 Resources
 - Forums
 - Pricing
 - Contact Us

A message at the bottom encourages using Amazon Simple Queue Service. A "Launch Instance" button is visible on the right.

Launch a new instance and select your favorite distribution (I'll use RedHat here):

The screenshot shows the "Choose an Amazon Machine Image (AMI)" step of the EC2 instance creation wizard. The user has selected the "Red Hat" AMI, which is highlighted in blue. Other options listed include:

- Amazon Linux**: Free tier eligible, SSD Volume Type - ami-bc5b48d0
- Red Hat**: Free tier eligible, SSD Volume Type - ami-875042eb
- SUSE Linux**: Free tier eligible, SSD Volume Type - ami-6bd2ce07
- Ubuntu**: Free tier eligible, SSD Volume Type - ami-87564fe2
- Microsoft Windows Server**: Free tier eligible, SSD Volume Type - ami-0d5b4361
- Amazon RDS**: Launch a database using RDS

Each AMI entry includes a "Select" button and a "64-bit" link. A note at the bottom says "Are you launching a database instance? Try Amazon RDS".

Pick the free instance

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your application. [Learn more](#) about instance types and how they can meet your computing needs.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
General purpose	t2.micro	1	1	EBS only	-	Low to Moderate
General purpose	t2.small	1	2	EBS only	-	Low to Moderate
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
General purpose	t2.large	2	8	EBS only	-	Low to Moderate
General purpose	m4.large	2	8	EBS only	Yes	Moderate
General purpose	m4.xlarge	4	16	EBS only	Yes	High
General purpose	m4.2xlarge	8	32	EBS only	Yes	High
General purpose	m4.4xlarge	16	64	EBS only	Yes	High
General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit
General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High

Currently selected: t2.micro (Variable ECUs, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

Filter by: All instance types Current generation Show/Hide Columns

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Then select **THE SAME VPC AS YOUR RDS DATABASE (should be default or the one you painfully created at first)**. Also be sure to Enable the Auto-assign IP.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-39caa150 (172.31.0.0/16) [BigdataSystemsVPC]	<input type="checkbox"/> Create new VPC
Subnet	subnet-41d5b23a (172.31.16.0/20) [BigdataSystem]	<input type="checkbox"/> Create new subnet 4090 IP Addresses available
Auto-assign Public IP	Use subnet setting (Enable)	
IAM role	None	<input type="checkbox"/> Create new IAM role
Shutdown behavior	Stop	
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply	
Tenancy	Shared - Run a shared hardware instance	Additional charges will apply for dedicated tenancy.

Network interfaces

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses
eth0	New network interface	subnet-41d5b23a	Auto-assign	Add IP

Add Device Advanced Details

Next pick a storage (20Go are largely enough for the class purposes and for you if you want to play a bit).

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 4: Add Storage
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)

Volume Type	Device	Snapshot	Size (GB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/sda1	snap-01b5340a	20	General Purpose SSD (GP2)	60 / 3000	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Next you can add tag to differentiate between your different instances if you already have a few. If you're not, it's useless so don't bother with that. Go to the next step to select your security group. Select the same as your RDS database.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-0ceedaa5	default	default VPC security group	Copy to new
<input checked="" type="checkbox"/> sg-f4d5e19d	rds-launch-wizard-3	Created from the RDS Management Console	Copy to new

Inbound rules for sg-f4d5e19d (Selected security groups: sg-f4d5e19d)

Type	Protocol	Port Range	Source
PostgreSQL	TCP	5432	92.156.69.224/32
All traffic	All	All	0.0.0.0/0

[Cancel](#) [Previous](#) [Review and Launch](#)

Then you can launch your instance.

EC2 will ask you to choose an ssh key pair. If you don't already have one, create a new one and download it.

Keep preciously the key they'll give you. Otherwise you won't be able to login to your instance.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Type: Create a new key pair
Key pair name: bigdataKey
[Download Key Pair](#)

You have to download the **private key file** (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.

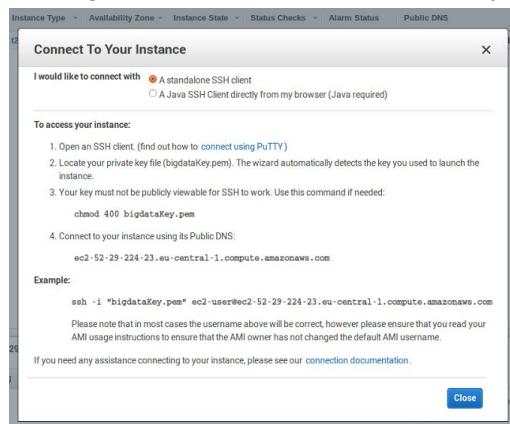
[Cancel](#) [Launch Instances](#)

You should be able to see your new EC2 instance creating. It is much faster than RDS so you don't have to take a coffee break here.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Spot Requests, Reserved Instances, Commands, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Auto Scaling, Launch Configurations, and Auto Scaling Groups. The main pane displays a table of instances. One instance, i-2ebc3092, is selected. Its details are shown in a modal window:

Description	Value
Instance ID	i-2ebc3092
Instance state	running
Instance type	t2.micro
Private DNS	ip-172-31-28-50.eu-central-1.compute.internal
Private IPs	172.31.28.50
Secondary private IPs	
VPC ID	vpc-39caa150
Subnet ID	subnet-41d5bc2a
Public DNS	ec2-52-29-224-23.eu-central-1.compute.amazonaws.com
Public IP	52.29.224.23
Elastic IP	-
Availability zone	eu-central-1b
Security groups	rds-launch-wizard-3, view rules
Scheduled events	No scheduled events
AMI ID	RHEL-7.2-HVM_GA-20151112-x86_64-1-Hourly2-GP2 (ami-875042eb)
Platform	-

Try to login using putty or ssh depending on your OS. The instructions to do so are given by clicking on the “connect” button once you select your instance



You'll have to install a postgresql client and the wget utilityon this machine so once you're logged in, if you choosed a RedHat distribution, just run the following command

```
> sudo yum install wget
> sudo yum install postgresql
```

You can then log into your RDS database using the same command as before.
For instance, for me :

```
> psql --host=bidatasystemrds.cj87do590fza.eu-central-1.rds.amazonaws.com --port=5432
--username=whatever --password --dbname=bigdata
```

If it's not working, there is probably an issue with your VPC so go please check the first part of this howto.

4- Download the data on your EC2 instance

Log in to your ec2 instance and use the following command

```
> wget https://s3-eu-west-1.amazonaws.com/bigdatatd/dblp\_dump
```

This should be quicker than downloading on your labtop and uploading it by hand after that.

log in to your database and create a user ‘student’

```
psql> CREATE USER student PASSWORD 'yourpassword';
psql>\q
```

Try to login to your database with the new user

```
> psql --host=bidatasystemrds.cj87do590fza.eu-central-1.rds.amazonaws.com --port=5432
--username=student --password --dbname=bigdata
```

If it works, you can log out and load the downloaded dblp_dump in your database

```
> psql --host=bidatasystemrds.cj87do590fza.eu-central-1.rds.amazonaws.com --port=5432
--username=student --password --dbname=bigdata < dblp_dump
```

You can now log to your RDS instance as the user ‘student’ and explore the data.