

Reference on : <https://aws.amazon.com/fr/blogs/big-data/running-r-on-aws/>

## TASK 1 VPC Creation

1 In the **Services** console select **VPC Start VPC WIZARD**

Select : **VPC with a Single Public Subnet**

Click **Select**

**Change the VPC name with loadbalancer**

Step 2: VPC with a Single Public Subnet

IPv4 CIDR block:\* (65531 IP addresses available)

IPv6 CIDR block:  
☒ No IPv6 CIDR Block  
☐ Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR:\* (251 IP addresses available)

Availability Zone:\*

Subnet name:

You can add more subnets after AWS creates the VPC.

Service endpoints

Enable DNS hostnames:\*☒ Yes ☐ No

Hardware tenancy:\*

Click **OK**

Check the route table of the public subnet (modify the name by clicking on the Name):

Create Subnet

Subnet Actions

Search Subnets and their pro X

<< 1 to 5

	Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4
<input type="checkbox"/>		subnet-d763a79f	available	vpc-601fc306   RosettaHUB VPC	172.30.2.0/24	250
<input type="checkbox"/>		subnet-6e97a535	available	vpc-601fc306   RosettaHUB VPC	172.30.0.0/24	250
<input type="checkbox"/>		subnet-9b05f4fd	available	vpc-601fc306   RosettaHUB VPC	172.30.1.0/24	250
<input checked="" type="checkbox"/>	Public subnet Elastic Load	subnet-c76ce38f	available	vpc-ea8ff48c   Elastic load	10.0.0.0/24	251
<input type="checkbox"/>	default	subnet-99f60bc3	available	vpc-ddea36bb	172.31.0.0/16	65531



Create Route TableDelete Route TableSet As Main Table

Search Route Tables and the X

<input type="checkbox"/>	Name	Route Table ID	Explicitly Associat	Main	VPC
<input type="checkbox"/>		rtb-7fc02206	0 Subnets	Yes	vpc-ddea36bb
<input type="checkbox"/>		rtb-29e38650	1 Subnet	No	vpc-ea8ff48c   Elastic load
<input checked="" type="checkbox"/>		rtb-0ae28773	0 Subnets	Yes	vpc-ea8ff48c   Elastic load
<input type="checkbox"/>		rtb-0bc52772	0 Subnets	Yes	vpc-601fc306   RosettaHUB VPC

rtb-0ae28773

SummaryRoutesSubnet AssociationsRoute PropagationTags

CancelSave

View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	

Add another route

3/ Define the same IGW in the route table of the second public VPC than the first one.

<input checked="" type="checkbox"/>	rtb-0ae28773	0 Subnets	Yes	vpc-ea8ff48c   Elastic load
<input type="checkbox"/>	rtb-0bc52772	0 Subnets	Yes	vpc-601fc306   RosettaHUB VPC

rtb-0ae28773

SummaryRoutesSubnet AssociationsRoute PropagationTags


Edit

View: All rules

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	<a href="#">igw-43eea924</a>		No

## Task 2 : lauch the EC2 instance

1/ Go to **Services** console and select **EC2** , on left pane select **instance** then **Launch Instance**.

**Amazon Linux**  
Free tier eligible

**Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-ca0135b3**  
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.  
Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

**Select**  
64-bit

Select the Amazon Linux and then select t2.micro.

Filter by: All instance types Current generation [Show/Hide Columns](#)

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

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

Select the good VPC and apply this for the 2 subnets

### 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 low price to the instance, and more.

Number of instances

1

[Launch into Auto Scaling Group](#)

Purchasing option

☐ Request Spot instances

Network

vpc-ea8ff48c | Elastic load

[Create new VPC](#)

Subnet

subnet-6258d72a | Public subnet 2 elastic Load | eu-  
251 IP Addresses available

[Create new subnet](#)

Auto-assign Public IP

Enable

Placement group

☐ Add instance to placement group.

IAM role

None

[Create new IAM role](#)

Shutdown behavior

Stop

Enable termination protection

☐ Protect against accidental termination

Monitoring

☐ Enable CloudWatch detailed monitoring  
[Additional charges apply.](#)

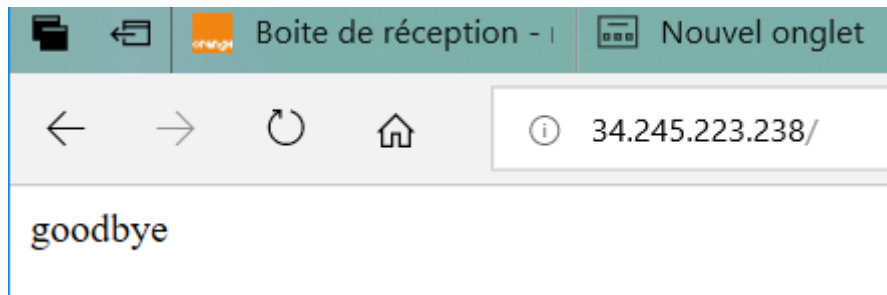
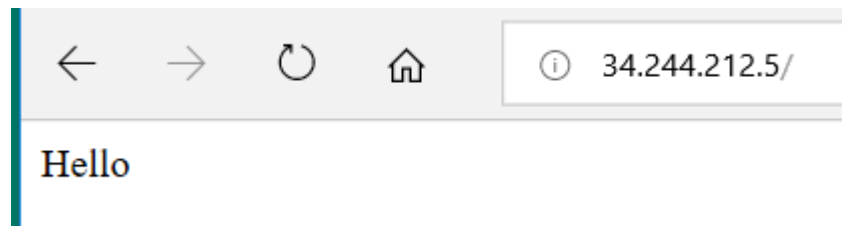
```
#!/bin/bash
```

```
sudo yum update -y
```

```
sudo yum -y install httpd
```


```
sudo service httpd start
```

```
sudo bash -c 'echo Hello> /var/www/html/index.html'
```



Task Create Load balancer

On EC2 instance console

Services ▾Resource Groups

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

LOAD BALANCING

**Load Balancers**

Create Load Balancer

Filter by tags and attribu

☐ Name


Select HTTP HTTPS

## Select load balancer type

---

Elastic Load Balancing supports three types of load balancers: , and . Choose the type that best fits your needs. [Learn more about which load balancer is right for you](#)

**Application Load Balancer**



**Create**

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing, TLS termination and visibility features targeted at application architectures, including microservices and containers.

[Learn more >](#)