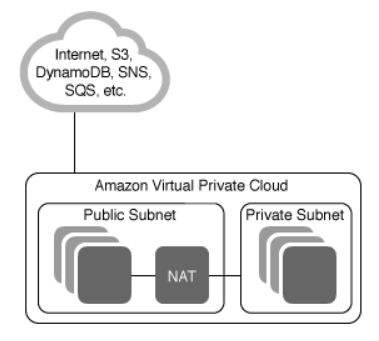
Create VPC with Public and Private subnet with access to the Internet

1. On AWS Services, go to **VPC**.
2. Click on **Start VPC Wizard**
3. On Select a VPC Configuration, choose **VPC with Public and Private subnets**

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

1. Click on **Select** button
2. You can specify VPC name: My VPC
3. Keep the default settings with a Public subnet and private subnet
4. On Specify the details of your NAT instance click on **Use a NAT gateway instead**
5. Instance type: t2.micro
6. Key pair name: create a key pair (.pem) with **PuttyGen** or if you have an already generated key pair, select its name into the list
7. A new page will display:

VPC Successfully Created

Your VPC has been successfully created.

You can launch instances into the subnets of your VPC. For more information, see Launching an Instance into Your Subnet.

1. Click on OK
2. Once the Public and Private subnets have been created, go to **Private subnet**
3. Select **Route Table**
4. Click on Edit
5. Change the Route table [rtb-c8ec96ae](https://eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#routetables:filter=rtb-c8ec96ae) to the Public subnet Route table [rtb-45e79d23](https://eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#routetables:filter=rtb-45e79d23) to connect it to the Internet Gateway
6. Then go to Services and select EC2
7. Launch instance
8. Choose AMI:
   1. We choose **Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type** because it is free and stable for R (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)
9. Click on Select
10. Choose instance type:
    1. We choose a small one and general purpose, **t2.micro**, which is free
11. Click on Next: Configure Instance Details
12. Configure Instance:
    1. In the Network menu, select My VPC
    2. In the **Subnet** menu, we select the Private Subnet
    3. We select Enable for **Auto-assign Public IP**
13. Click on Next: Add Storage
14. Keep the default values
15. Click on Next: Add Tags
16. Click on Next: Configure Security Group
17. Keep default rule:
    1. Type: SSH
    2. Protocol: TPC
    3. Port Range: 22
    4. Source : Anywhere
18. Click on Review and Launch
19. Check your Instance details and scroll to the bottom of the page
20. Click on Launch
21. It will appear a pop-up **Select an existing key pair or create a new key pair**
    1. Choose an existing key pair
    2. Select a previous generated key pair
    3. Select the acknowledge check box
22. Click on Launch Instance
23. Go to **Putty**
24. In Session paste the **Public DNS** of the created Instance with the prefix ec2-user@
25. Expand SSH section and select Auth
    1. Browse the private key file for authentication and select the previous generated key
26. Click on Open
27. In a new Web Browser window paste the **Public DNS** of your instance

My Public DNS

ec2-34-242-36-185.eu-west-1.compute.amazonaws.com