



Passo a Passo:

Pré-requisitos

- Ter conta AWS ativa
 - Saber em qual **região** vai trabalhar (ex: us-east-1)
 - Ter um **par de chaves SSH** ou criar um na hora do EC2
-

◆ PARTE 1 — Criar a VPC

1. Acessar a VPC

1. No Console AWS, procure por **VPC** no search bar.
 2. Clique em **VPC** (serviço “Amazon VPC”).
-

2. Criar a VPC

1. Menu lateral esquerdo → **Your VPCs** (Suas VPCs).
2. Clique em **Create VPC**.
3. Em **Resources to create**, escolha:
 - **VPC only** (pra ensinar tudo na mão)
4. Em **Name tag**:
 - demo--vpc -ec2
5. Em **IPv4 CIDR block**:
 - 10.0.0.0/16
 - 172.16.0.0/16
6. Mantenha o resto padrão (IPv6 desabilitado se quiser simplificar).
7. Clique em **Create VPC**.

Pronto: você tem uma VPC vazia, sem subnet, sem rota, sem nada.

3. Criar a Subnet Pública

1. No menu esquerdo → **Subnets**.



2. Clique em **Create subnet**.
3. Em **VPC ID**, selecione sua VPC: demo-vpc-vpc-ec2.
4. Em **Subnet name**:
 - demo-public-a
5. Em **Availability Zone**:
 - Escolha uma (ex: us-east-1a).
6. Em **IPv4 CIDR block**:
 - 10.0.1.0/24
 - 172.16.1.0/24
7. Clique em **Create subnet**.

Habilitar IP público automático na subnet pública

1. Na lista de subnets, selecione demo-public-a.
 2. Clique em **Actions** → **Edit subnet settings** (ou “Editar configurações”).
 3. Marque **Enable auto-assign public IPv4 address**.
 4. Salvar.
-

4. Criar a Subnet Privada

1. De novo em **Create subnet**.
2. **VPC ID**: demo-vpc-vpc-ec2.
3. **Subnet name**: demo-private-b.
4. **Availability Zone**: outra, se quiser (ex: us-east-1b).
5. **IPv4 CIDR block**:
 1. 10.0.2.0/24.
 2. 172.16.2.0/24
6. **Create subnet**.

(Não habilitar IP público nessa.)



5. Criar e anexar o Internet Gateway (IGW)

1. Menu esquerdo → **Internet Gateways**.
 2. Clique em **Create internet gateway**.
 3. **Name tag**: demo-igw.
 4. Clique em **Create internet gateway**.
 5. Com ele selecionado, clique em **Actions** → **Attach to VPC**.
 6. Selecione a VPC demo-vpc-vpc-ec2 → **Attach internet gateway**.
-

6. Route Table da Subnet Pública

1. Menu esquerdo → **Route tables**.
2. Clique em **Create route table**.
3. **Name**: demo-public-rt.
4. **VPC**: demo-vpc-vpc-ec2.
5. **Create route table**.

Adicionar rota para a internet

1. Selecione demo-public-rt.
2. Aba **Routes** → **Edit routes** → **Add route**.
3. Em **Destination**: 0.0.0.0/0.
4. Em **Target**: selecione **Internet Gateway** → demo-igw.
5. Clique em **Save changes**.

Associar a subnet pública a essa route table

1. Ainda na demo-public-rt, vá na aba **Subnet associations**.
2. Clique em **Edit subnet associations**.
3. Marque a subnet demo-public-a.
4. **Save associations**.

Agora: tudo na demo-public-a tem rota de saída para a internet.



7. (Opcional) NAT Gateway para subnet privada

Se quiser ensinar **subnet privada com saída para internet**:

1. Menu → **NAT Gateways**.
2. **Create NAT gateway**.
3. **Name**: demo-nat-gw.
4. **Subnet**: escolha demo-public-a.
5. **Connectivity type**: Public.
6. Clique em **Allocate Elastic IP**.
7. Clique em **Create NAT gateway**.

Depois, crie uma **route table privada**:

1. **Route tables** → **Create route table**.
 2. demo-private-rt → VPC demo-vpc-vpc-ec2.
 3. Em **Routes** → Add route:
 - Destination: 0.0.0.0/0
 - Target: NAT Gateway → demo-nat-gw
 4. Em **Subnet associations** → associe demo-private-b.
-

◆ PARTE 2 — Criar a EC2 na Subnet Pública

1. Acessar EC2

1. No Console AWS, procure por **EC2** e clique no serviço.
 2. Clique em **Instances** → **Launch instances**.
-

2. Configurar Instância EC2

1. **Name and tags**:
 - Name: demo-ec2-public.
2. **Application and OS Images (AMI)**:
 - Selecione **Amazon Linux 2023** (ou Amazon Linux 2).



3. Instance type:

- t2.micro (Free Tier, se estiver disponível).

4. Key pair (login):

- Escolha uma key existente ou crie uma (ex: demo-key).

5. Network settings:

- **VPC:** selecione demo-vpc-vpc-ec2.
- **Subnet:** selecione demo-public-a.
- **Auto-assign public IP: Enable** (se não estiver automático).

6. Firewall (security group):

- Escolha **Create security group**:
 - Name: demo-sg-ssh-http
 - Inbound rules:
 - **SSH** → Port 22 → Source: **My IP** (ou seu IP fixo)
 - **HTTP** → Port 80 → Source: 0.0.0.0/0 (pra demo de web).
- Outbound: deixe All traffic (padrão).

7. Clique em Launch instance.

3. Conectar na EC2 (SSH)

Depois da instância ficar em **running**:

1. Clique na instância.
2. Copie o **Public IPv4 address**.

Em Linux/macOS:

No terminal:

```
chmod 400 demo-key.pem
```

```
ssh -i demo-key.pem ec2-user@SEU_IP_PUBLICO
```

Exemplo:

```
ssh -i demo-key.pem ec2-user@54.123.45.67
```



Em Windows:

- Use o **Windows Terminal** com OpenSSH ou o **PuTTY** (convertendo a chave).
- Com PowerShell (se tiver OpenSSH):

```
ssh -i .\demo-key.pem ec2-user@SEU_IP_PUBLICO
```

4. Instalar NGINX pra testar HTTP

Já logado na EC2 (Amazon Linux):

```
sudo yum update -y      # opcional, mas bom pra aula  
sudo yum install -y nginx  
sudo systemctl enable nginx  
sudo systemctl start nginx
```

Agora, no navegador:

http://SEU_IP_PUBLICO

Se tudo estiver certo, vai aparecer a tela padrão do NGINX.

◆ PARTE 3 — (Opcional) EC2 na Subnet Privada + Bastion

Se quiser fazer “modo arquiteto” completo na aula:

1. Criar uma segunda EC2 (demo-ec2-private) usando:
 - VPC: demo-vpc-vpc-ec2
 - Subnet: demo-private-b
 - Sem IP público
 - SG permitindo SSH **apenas** a partir do SG do bastion (demo-sg-ssh-
http, por exemplo).
2. Usar a EC2 pública como **bastion host**:
 - Conectar na pública via SSH.
 - De dentro da pública, conectar na privada via IP privado.

Exemplo de comando:



AWS UG Triângulo Mineiro

```
ssh -i demo-key.pem ec2-user@10.0.2.10 # IP privado da EC2 privada
```



🎯 Boas Práticas que você deve ensinar

- Nunca abrir SSH para 0.0.0.0/0
- Usar EC2 Connect
- Sempre separar subnets públicas e privadas
- EC2 backend sempre na privada
- Usar Load Balancer + Autoscaling
- Usar IAM Roles (sem Access Key na EC2!)
- Habilitar logs de VPC Flow Logs

Fluxo no console:



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#Home:

Create VPC Launch EC2 Instances

Note: Your instance will launch in the United States region.

Resources by Region

VPCs N. Virginia 1 NAT Gateways N. Virginia 0
▶ See all regions ▶ See all regions

Subnets N. Virginia 4 VPC Peering Connections N. Virginia 0
▶ See all regions ▶ See all regions

Route Tables N. Virginia 1 Network ACLs N. Virginia 1
▶ See all regions ▶ See all regions

Service Health View complete service health details ↗

Settings Block Public Access
Zones ↗
Console Experiments

Additional Information VPC Documentation
All VPC Resources

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateVpc:createMode=vpcOnly

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
demo-vpc-vpc-ec2

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16

IPv6 CIDR
10.0.0.0/16

CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy [Info](#)
Default

VPC encryption control (\$) - new | [Info](#)
Monitor mode provides visibility into encryption status without blocking traffic. Enforce mode prevents unencrypted traffic. [Additional charges apply ↗](#)

None Monitor mode
See which resources in your VPC are unencrypted but allow the creation of unencrypted resources.

Enforce mode
Requires all resources, except exclusions, in your VPC to be encryption-capable and blocks creation of unencrypted resources.

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Screenshot of the AWS VPC creation process:

VPC encryption control (\$ - new | Info)

Monitor mode provides visibility into encryption status without blocking traffic. Enforce mode prevents unencrypted traffic. [Additional charges apply](#)

None

Monitor mode
See which resources in your VPC are unencrypted but allow the creation of unencrypted resources.

Enforce mode
Requires all resources, except exclusions, in your VPC to be encryption-capable and blocks creation of unencrypted resources.

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional

Name demo-vpc-vpc-ec2

You can add 49 more tags

VPC dashboard <

AWS Global View

Filter by VPC

Virtual private cloud

Your VPCs

- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists

Details [Info](#)

VPC ID	State	Block Public Access	DNS hostnames
vpc-0e2c8001ff2057a80	Available	Off	Disabled
DNS resolution	Tenancy default	DHCP option set dopt-9e0101fb	Main route table rtb-0f8937ff90025a737
Main network ACL acl-089ca107c9454b07a	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
IPv6 CIDR (Network border group)	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 147397866377
Encryption control ID	Encryption control mode -		

Actions

VPC dashboard <

AWS Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Subnets (4) Info

Last updated 12 minutes ago

Name	Subnet ID	State	VPC	Block Publ
-	subnet-c2cc50e8	Available	vpc-e7de3280	Off
-	subnet-a112449c	Available	vpc-e7de3280	Off
-	subnet-5e7f3028	Available	vpc-e7de3280	Off
-	subnet-da197c82	Available	vpc-e7de3280	Off

Select a subnet



Screenshot of the AWS VPC Subnet creation interface. The top navigation bar shows the AWS logo, search bar, and various services like EC2, VPC, DynamoDB, etc. The top right shows the account ID (1473-9786-6377), region (United States (N. Virginia)), and user (Rogerio Fontes).

The main page title is "Create subnet". It has sections for "VPC ID" (vpc-0e2c8001ff2057a80 (demo-vpc-vpc-ec2)), "Associated VPC CIDRs" (10.0.0.0/16), and "Subnet settings".

Subnet settings: Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1:

- Subnet name:** demo-public-a (maximum 256 characters)
- Availability Zone:** United States (N. Virginia) / us-east-1a (us-east-1a)
- IPv4 VPC CIDR block:** 10.0.0.0/16
- IPv4 subnet CIDR block:** 10.0.1.0/24 (256 IPs)
- Tags - optional:** A single tag named "Name" with value "demo-public-a".
- Actions:** Buttons for "Add new tag", "Remove", "Add new subnet", "Cancel", and a large orange "Create subnet" button highlighted with a green box.



Screenshot of the AWS VPC Subnets page showing a successful subnet creation:

You have successfully created 1 subnet: subnet-094d562ea49c4f0d3

Subnets (1) Info Last updated less than a minute ago Actions Create subnet

Name	Subnet ID	State	VPC	Block Publ
demo-public-a	subnet-094d562ea49c4f0d3	Available	vpc-0e2c8001ff2057a80 demo...	Off

Select a subnet

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences Account ID: 1473-9786-6377 Rogerio Fontes

Screenshot of the "Create subnet" wizard - Step 1: Subnet settings:

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
demo-private-b
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / use1-az4 (us-east-1b)

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.2.0/24 256 IPs

Tags - optional
Key Name Value - optional demo-private-b Add new tag Remove Add new subnet

Create subnet (button highlighted with a green box)



Screenshot of the AWS VPC Subnets page showing a successfully created subnet:

You have successfully created 1 subnet: subnet-0a89f513632db23ed

Subnets (1)

Name	Subnet ID	State	VPC	Block Public
demo-private-b	subnet-0a89f513632db23ed	Available	vpc-0e2c8001ff2057a80 demo...	Off

Select a subnet

Screenshot of the AWS VPC Subnet details page for subnet-0a89f513632db23ed / demo-private-b:

Details

Subnet ID	subnet-0a89f513632db23ed	Subnet ARN	arn:aws:ec2:us-east-1:147397866377:subnet/subnet-0a89f513632db23ed	State	Available	Block Public Access	Off
IPv4 CIDR	10.0.0.0/24	IPv6 CIDR	-	IPv6 CIDR association ID	-	Route table	-
Availability Zone	use1-az4 (us-east-1b)	Available IPv4 addresses	251	VPC	vpc-0e2c8001ff2057a80 demo-vpc-vpc-ec2	Auto-assign public IPv4 address	No
Network ACL	-	Network border group	us-east-1	Outpost ID	-	IPV4 CIDR reservations	-
Auto-assign customer-owned IPv4 address	No	Default subnet	No	Hostname type	IP name	Resource name DNS A record	Disabled
IPv6 CIDR reservations	-	Customer-owned IPv4 pool	-				

Screenshot of the AWS VPC Create Internet Gateway page:

Create internet gateway

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.
demo-igw

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="demo-igw"/> <input type="button" value="Remove"/>

You can add 49 more tags.



Screenshot of the AWS VPC Internet Gateways page. The 'Create internet gateway' button is highlighted with a green circle.

Screenshot of the 'Create internet gateway' wizard. The 'Name tag' field contains 'demo-igw'. The 'Tags - optional' section shows a single tag 'Name: demo-igw'. The 'Create internet gateway' button is highlighted with a green circle.

Screenshot of the newly created Internet Gateway details page. The 'Attach to a VPC' button is highlighted with a green circle. The 'Actions' dropdown menu is also circled in green.



Screenshot of the AWS VPC Attach Internet Gateway wizard. The page title is "Attach to VPC (igw-08201ab40ba882497) [Info]". It shows a search bar with "vpc-0e2c8001ff2057a80" and a large orange "Attach internet gateway" button at the bottom right.

Screenshot of the AWS VPC Internet gateway details page for "igw-08201ab40ba882497 / demo-igw". A green box highlights the "Actions" dropdown menu, which contains options like "Edit", "Delete", and "Create route table".

Screenshot of the AWS VPC Route tables page. A green box highlights the "Create route table" button in the top right corner of the "Route tables (1) [Info]" section.

Screenshot of the AWS VPC Route tables page. A green box highlights the "Create route table" button in the top right corner of the "Route tables (1) [Info]" section.



Screenshot of the AWS VPC Route Tables creation process across three steps:

- Create route table (Step 1):** Shows the "Route table settings" section where a name ("demo-public-rt") and VPC ("vpc-0e2c8001ff2057a80 (demo-vpc-vpc-ec2)") are specified. A "Tags" section is also present.
- Create route table (Step 2):** Similar to the first step, showing the "Route table settings" and "Tags" sections. The "Create route table" button is highlighted with a green oval.
- VPC dashboard (Step 3):** Shows the newly created route table "rtb-0746fb5a5c074dbad / demo-public-rt". The "Details" section shows the route table ID, VPC, and owner ID. The "Routes" tab is selected, displaying one route entry. The "Actions" dropdown menu is visible.



Screenshot of the AWS VPC Route Tables page showing the details of route table rtb-0746fb5a5c074dbad. The 'Routes' tab is selected, displaying one route: Destination 10.0.0.0/16 Target local Status Active Propagated No Route Origin CreateRouteTable. A green circle highlights the 'Edit routes' button.

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable

A green circle highlights the 'Add route' button at the bottom left of the table.

Buttons: Cancel, Preview, Save changes.

Screenshot of the AWS VPC Route Tables page showing the details of route table rtb-0746fb5a5c074dbad. The 'Routes' tab is selected, displaying two routes: Destination 10.0.0.0/16 Target local Status Active Propagated No Route Origin CreateRouteTable, and Destination 0.0.0.0/0 Target Internet Gateway Status - Propagated No Route Origin CreateRoute. A green circle highlights the 'Save changes' button.

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	-	No	CreateRoute

Buttons: Cancel, Preview, Save changes.



Screenshot of the AWS VPC Route Tables page showing the creation of a new route table.

The top section shows the details for the newly created route table:

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0746fb5a5c074dbad	No	-	-
VPC	Owner ID		
vpc-0e2c8001ff2057a80 demo-vpc-ec2	147397866377		

The "Routes" tab is selected, showing two routes:

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-08201ab40ba882497	Active	No	Create Route
10.0.0.16	local	Active	No	Create Route Table

The bottom section shows the list of route tables:

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
-	rtb-6682e401	-	-	Yes	vpc-e
-	rtb-0f8937ff90025a737	-	-	Yes	vpc-0
<input checked="" type="checkbox"/> demo-public-rt	rtb-0746fb5a5c074dbad	-	-	No	vpc-0

A green box highlights the "Subnet associations" tab in the route table details view.



Screenshot of the AWS VPC Edit subnet associations page. A green oval highlights the 'Available subnets' table, showing two subnets: 'demo-public-a' and 'demo-private-b'. Another green oval highlights the 'Selected subnets' section, which contains 'subnet-094d562ea49c4f0d3 / demo-public-a'. The bottom right corner shows the 'Save associations' button.

Screenshot of the AWS VPC NAT gateways Info page. A green oval highlights the 'Actions' dropdown menu, specifically the 'Create NAT gateway' button.

Screenshot of the AWS VPC NAT gateways Info page, identical to the one above but with a different URL. A green oval highlights the 'Actions' dropdown menu, specifically the 'Create NAT gateway' button.



Screenshot of the AWS Management Console showing the process of creating a NAT gateway.

The screenshots show three steps in sequence:

- Step 1: NAT gateway settings**
 - Name**: demo-nat-gw
 - Availability mode**: Regional - new (selected)
 - VPC**: vpc-0e2c8001ff2057a80 (demo-vpc-vpc-ec2)
- Step 2: Availability mode**
 - Regional - new** (selected)
 - Zonal**
- Step 3: Additional settings**
 - Elastic IP allocation ID**: eipalloc-064df239dfd508602 (selected)
 - Tags**: Name: demo-nat-gw
 - Create NAT gateway** button highlighted with a green oval



Screenshot of the AWS VPC Route Tables page showing a list of existing route tables. A green circle highlights the 'Create route table' button at the top right of the table header.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-6682e401	-	-	Yes	vpc-e
-	rtb-0f8937ff90025a737	-	-	Yes	vpc-0
<input checked="" type="checkbox"/> demo-public-rt	rtb-0746fb5a5c074dbad	subnet-094d562ea49c4f...	-	No	vpc-0

Screenshot of the 'Create route table' wizard. The 'Route table settings' step is shown. A green circle highlights the 'Name' field where 'demo-private-rt' is entered. The 'VPC' dropdown shows 'vpc-0e2c8001ff2057a80 (demo-vpc-vpc-ec2)'. The 'Tags' step is also visible below.

Screenshot of the 'Create route table' wizard. The 'Route table settings' step is shown again. A green circle highlights the 'Name' field where 'demo-private-rt' is entered. The 'VPC' dropdown shows 'vpc-0e2c8001ff2057a80 (demo-vpc-vpc-ec2)'. The 'Tags' step is also visible below.

Screenshot of the 'Create route table' wizard. The 'Tags' step is shown. A green circle highlights the 'Value - optional' field where 'demo-private-rt' is entered. The 'Add new tag' button is visible. The 'Create route table' button at the bottom right is also highlighted with a green circle.



Screenshot of the AWS VPC Route Tables page showing a successful route creation:

Route table ID: rtb-05c66faa1967f1f5c | demo-private-rt was created successfully.

Details Info:

Route table ID: rtb-05c66faa1967f1f5c	Main: No	Explicit subnet associations: -	Edge associations: -
VPC: vpc-0e2c8001ff2057a80 demo-vpc-ec2	Owner ID: 147397866377		

Routes (1)

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable

Edit routes button highlighted with a green circle.

Screenshot of the "Edit routes" dialog for the route table:

Destination: 10.0.0.0/16
Target: local (Status: Active, Propagated: No, Route Origin: CreateRouteTable)

Target: NAT Gateway (Status: -) and nat- (Status: -)

Add route button and **Save changes** button highlighted with a green circle.

Screenshot of the AWS VPC Route Tables page showing updated routes:

Updated routes for rtb-05c66faa1967f1f5c / demo-private-rt successfully

rtb-05c66faa1967f1f5c / demo-private-rt

Details Info:

Route table ID: rtb-05c66faa1967f1f5c	Main: No	Explicit subnet associations: -	Edge associations: -
VPC: vpc-0e2c8001ff2057a80 demo-vpc-ec2	Owner ID: 147397866377		

Routes (2)

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	NAT Gateway	-	No	CreateRoute



Screenshot of the AWS Management Console showing the EC2 service interface across three pages:

- EC2 Home Page:** Shows the "Resources" section with a summary of running instances, auto scaling groups, capacity reservations, dedicated hosts, instances, load balancers, security groups, and volumes. A green circle highlights the "Instances (running)" button. The "Account attributes" and "Explore AWS" sections are also visible.
- Instances Page:** Shows the "Instances Info" section where no matching instances are found. A green circle highlights the "Launch instances" button. The left sidebar shows navigation options like Instances, Instance Types, Launch Templates, and Spot Requests.
- Launch an instance Page:** A detailed configuration page for launching a new instance. It includes fields for "Name and tags" (with "demo-ec2-public" entered), "Application and OS Images (Amazon Machine Image)" (with a search bar and "Quick Start" link), and "Summary" (with "Number of instances" set to 1, "Software Image (AMI)" set to "Amazon Linux 2023.9.2...read more", "Virtual server type (instance type)" set to "t3.micro", and "Firewall (security group)" set to "New security group"). Buttons for "Cancel", "Launch instance", and "Preview code" are at the bottom. A green circle highlights the "Launch instance" button.



Screenshot of the AWS Lambda console showing the function configuration page for the 'HelloWorld' function.

The Lambda function is configured with the following details:

- Function name:** HelloWorld
- Description:** A simple Hello World Lambda function
- Runtime:** Python 3.8
- Memory:** 128 MB
- Timeout:** 3 seconds
- Execution role:** Lambda@Edge (arn:aws:lambda:us-east-1:123456789012:role/lambdaBasicExecutionRole)
- Code:** A ZIP file containing the Lambda function code.

The function has a single alias named 'Production'.

Logs and Metrics sections are also visible on the right side of the page.



Screenshot of the AWS CloudFront console showing the creation of a new distribution. The distribution is named 'MyFirstCloudFrontDistribution' and is set to 'Public'. It has a single origin pointing to an Amazon S3 bucket named 'myfirstcloudfrontbucket'. The 'Default Cache Behavior' is configured with a 300-second TTL and a 'Compress' setting. The 'Behaviors' section shows a single behavior for the root path '/' with a 300-second TTL. The 'Logs' section indicates logs are being sent to an Amazon CloudWatch Logs log stream named 'CloudFront-logs-myfirstcloudfrontbucket'. The 'Metrics' section shows CloudFront metrics are enabled. The 'Access Logs' section shows CloudFront access logs are enabled and will be sent to an Amazon CloudWatch Logs log stream named 'CloudFront-logs-myfirstcloudfrontbucket'. The 'Origin Access Identity' section shows no OAI is currently assigned.



Screenshot of the AWS CloudShell interface showing the launch of an EC2 instance. The instance configuration includes:

- User data - optional**:
sudo yum update -y
sudo yum install -y nginx
sudo systemctl enable nginx
sudo systemctl start nginx
- Software Image (AMI)**: Amazon Linux 2023 AMI 2023.9.2...read more
ami-0fa3fe0fa7920f68e
- Virtual server type (instance type)**: t3.micro
- Firewall (security group)**: New security group

The "Launch instance" button is highlighted with a green oval.

Below the configuration, the instance status shows:

- Root volume, 3000 IOPS, Not encrypted
- Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage
- Add new volume
- Click refresh to view backup information
- 0 x File systems
- Advanced details

The "Launch instance" button is also highlighted with a green oval.

Finally, the success message indicates the instance has been successfully launched:

Success
Successfully initiated launch of instance (i-0dc45fbfa48a30bd2)

Next Steps:
Create billing and free tier usage alerts
Connect to your instance
Connect an RDS database
Create EBS snapshot policy

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1



AWS UG Triângulo Mineiro

The screenshot illustrates the AWS EC2 Instances interface, specifically the 'Launch instances' step. The top navigation bar shows the AWS logo, search bar, and account information (Account ID: 1473-9786-6377, Rogerio Fontes). The main content area displays the 'Instances (1)' section with one instance listed: 'demo-ec2-public' (i-014731e1d13834d3d) is running as a t3.micro instance. A green box highlights the 'Launch instances' button in the Actions dropdown menu. Below this, a 'Select an instance' dropdown is open, showing the same instance. The left sidebar contains navigation links for EC2 (Dashboard, Global View, Events), Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager), and Images.

EC2 > Instances > i-014731e1d13834d3d > Manage IP addresses

To assign additional public IPv4 addresses to this instance, you must [allocate](#) Elastic IP addresses and associate them with the instance or its network interfaces.

IPv4 addresses

Private IP address: 10.0.1.130 Public IP address:

[Unassign](#) [Assign new IP address](#)

[Auto-assign public IP](#) | [Info](#)

Allow secondary private IPv4 addresses to be reassigned

Allows you to reassign a private IPv4 address that is assigned to another network interface or instance to this instance.

Allow

[Cancel](#) [Save](#)

Instances (1) [Info](#)

Last updated 1 minute ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[Find Instance by attribute or tag \(case-sensitive\)](#) [Clear filters](#)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	demo-ec2-public	i-014731e1d13834d3d	Running	t3.micro	3/3 checks passed	View alarms +

Select an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name: demo-ec2-private [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

[Search our full catalog including 1000s of application and OS images](#)

[Recents](#) [My AMIs](#) [Quick Start](#)

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023.9.2... [read more](#)
ami-0fa3fe0fa7920f68e

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)

[Cancel](#) [Launch instance](#) [Preview code](#)



Screenshot of the AWS Cloud Console showing the 'Launch an instance' wizard. The first step, 'Application and OS Images (Amazon Machine Image)', is displayed. It shows a search bar, a 'Quick Start' tab selected, and a grid of AMI icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A 'Browse more AMIs' link is also present. The 'Amazon Machine Image (AMI)' section details the 'Amazon Linux 2023 kernel-6.1 AMI' with its ID and configuration. The 'Summary' panel on the right shows 1 instance, the selected 'Software Image (AMI)' (Amazon Linux 2023.9.2...), and the 'Virtual server type (instance type)' (t3.micro). Buttons for 'Cancel', 'Launch instance', and 'Preview code' are at the bottom.

Screenshot of the AWS Cloud Console showing the 'Launch an instance' wizard. The second step, 'Instance type', is displayed. It shows the selected 't3.micro' instance type with its details (Family: t3, 2 vCPU, 1 GiB Memory, Current generation: true) and pricing information. A 'Free tier eligible' badge is shown. The 'Summary' panel on the right remains the same, showing 1 instance, the selected AMI, and the instance type. Buttons for 'Cancel', 'Launch instance', and 'Preview code' are at the bottom.