Aplicação em Container na Nuvem - Webscraper de GPUs na AWS

Nome do Responsável:

- Aparecido Valdemir de Freitas

Nome dos Integrantes:

- Alexandre Moreno Ciosani
- Jorge Orselli Mutran
- Kaick Kenithi Nishiya
- Lucas Quinto Roli
- Matheus Pestilli Rodrigues João
- Wendel Pereira de Amorim

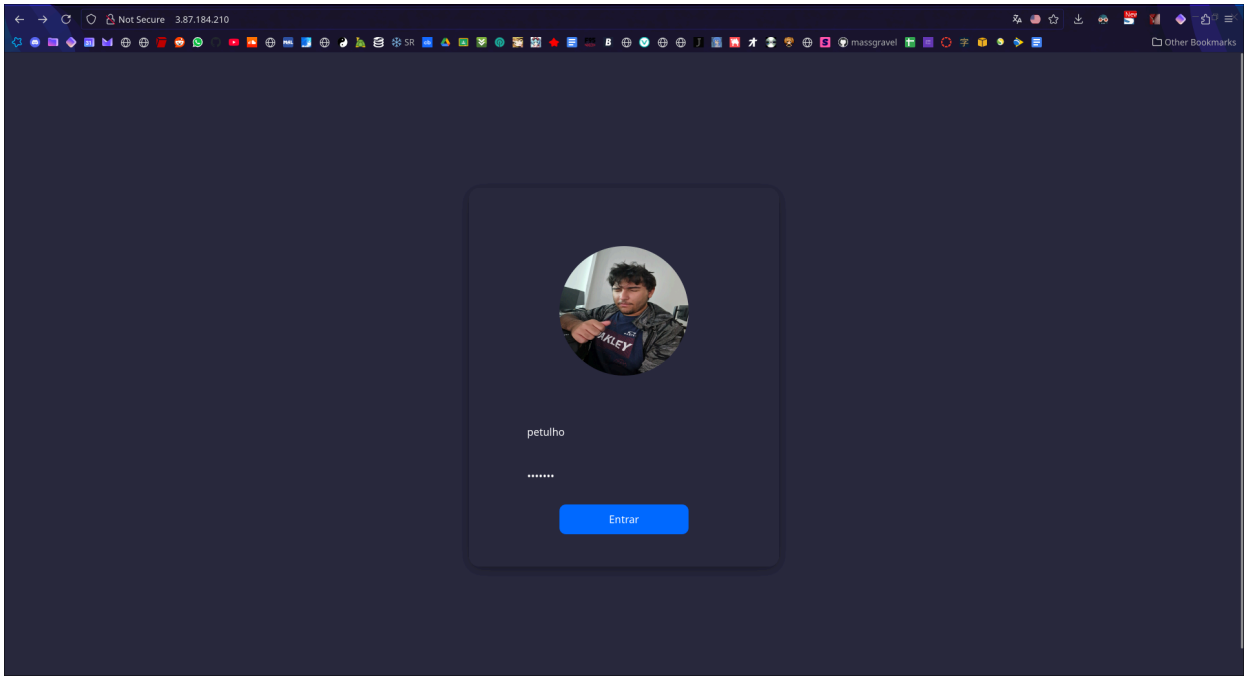
Introdução

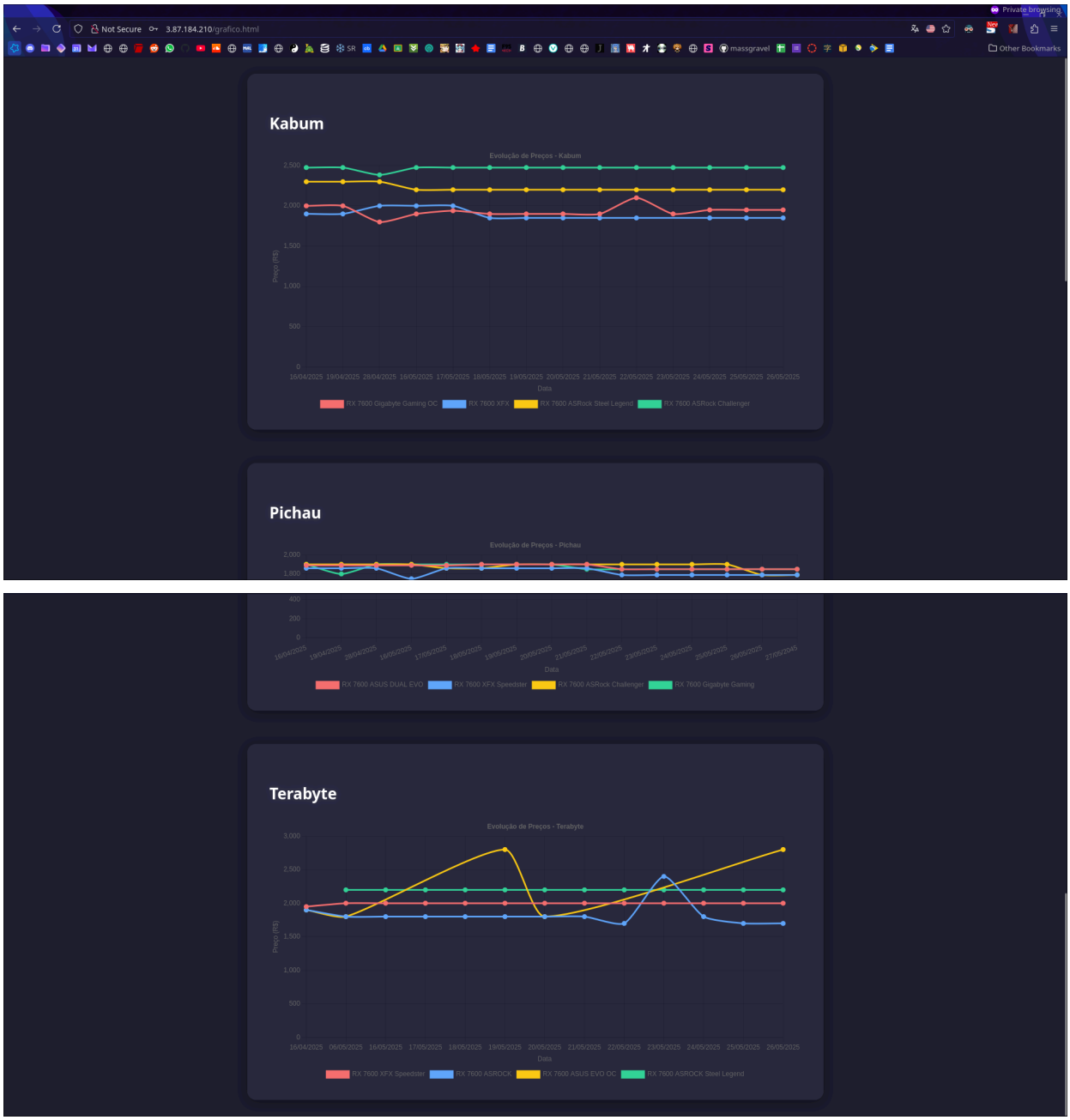
O projeto tem como objetivo demonstrar o funcionamento de uma aplicação que coleta o preço de determinadas placas de vídeo à venda nos websites de comércio eletrônico e analisar a evolução dos preços como também rastrear a mudança de preços no mercado de GPUs. A aplicação é conteinerizada usando o Docker e é alocada em nuvem como requerido pelo responsável pelo projeto.

Visão Geral

O website pode ser acessado através do seguinte link: http://3.87.184.210/.
O site possui uma simples página de login (apenas um if else para verificar o usuário e senha).
As credenciais são:
Usuário: petulho
Senha: sodorme

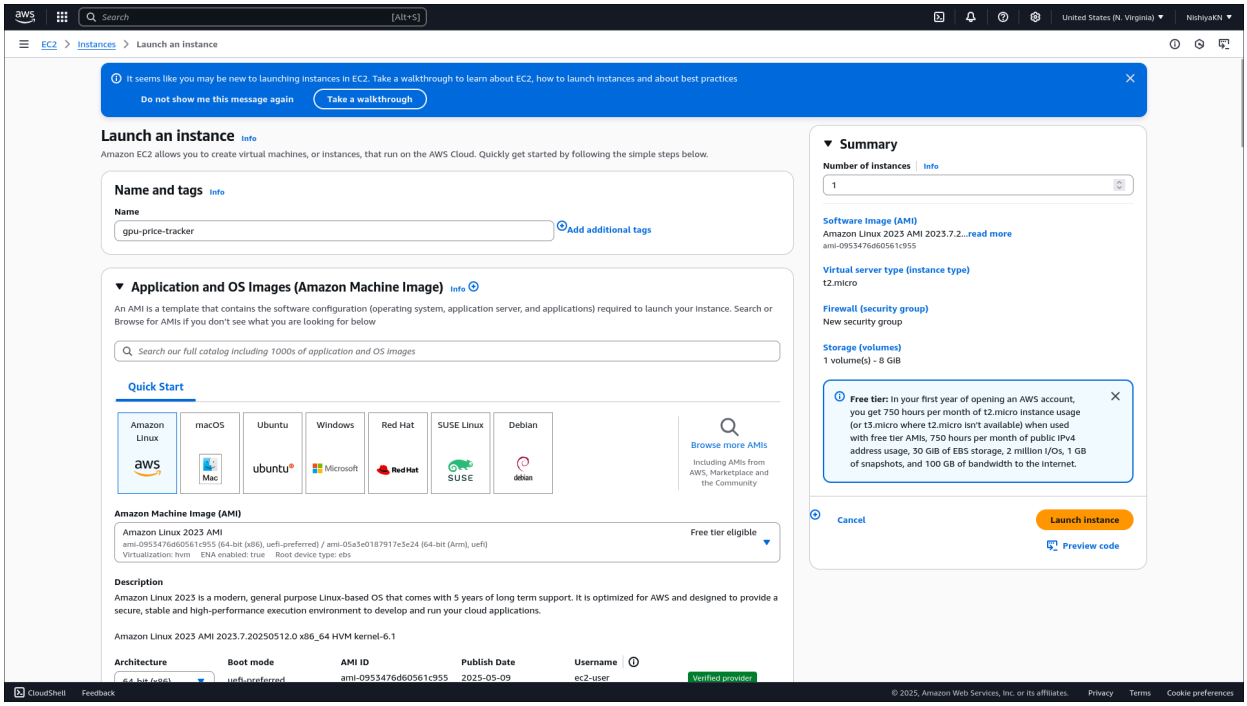
Mas é possível acessar o website diretamente através do link: http://3.87.184.210/grafico.html

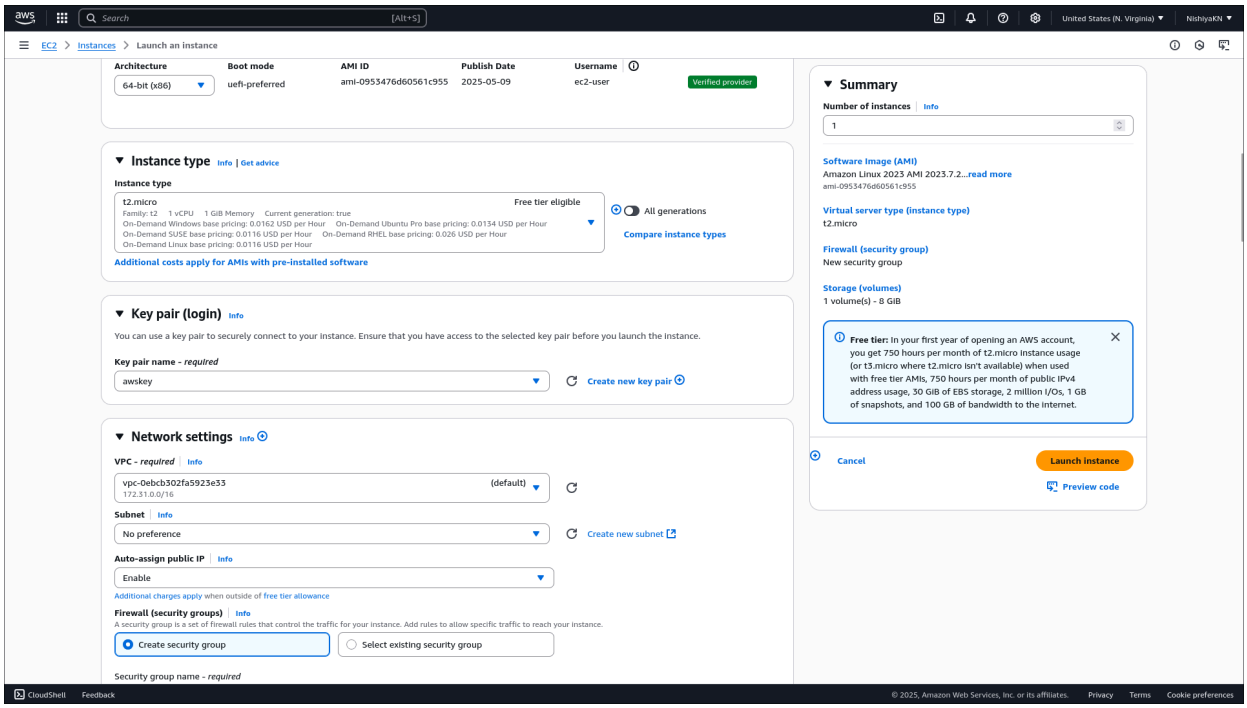


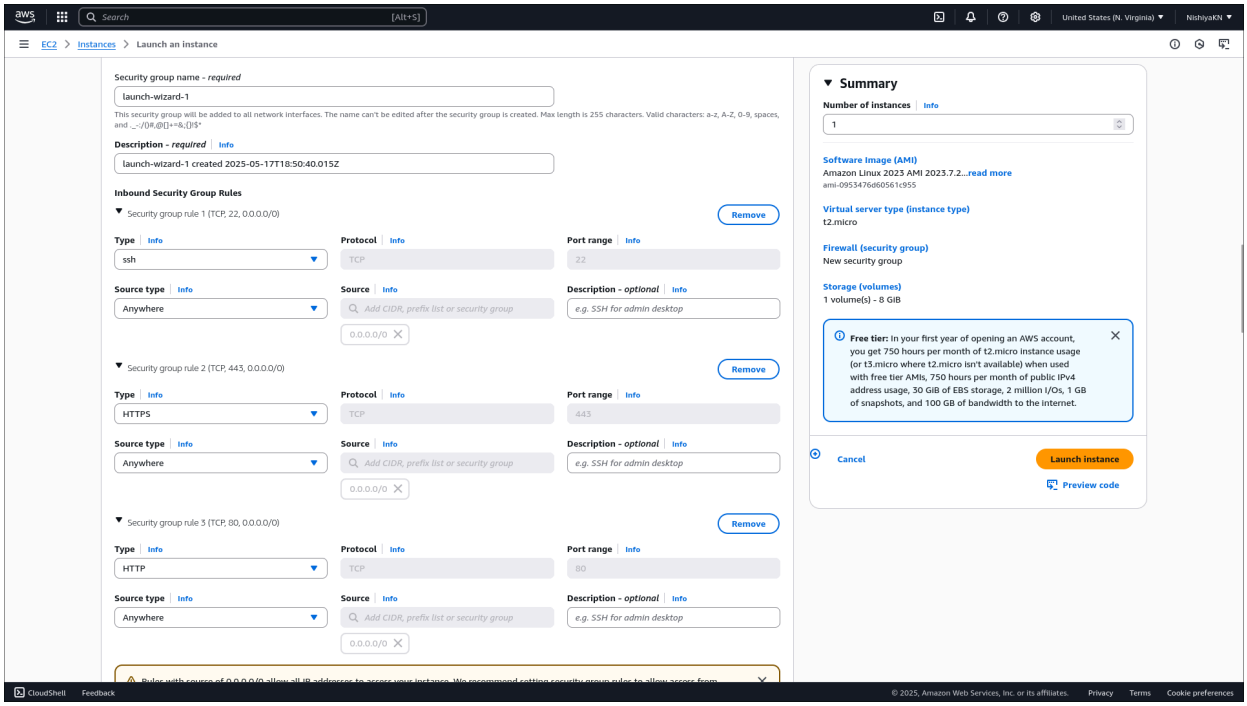


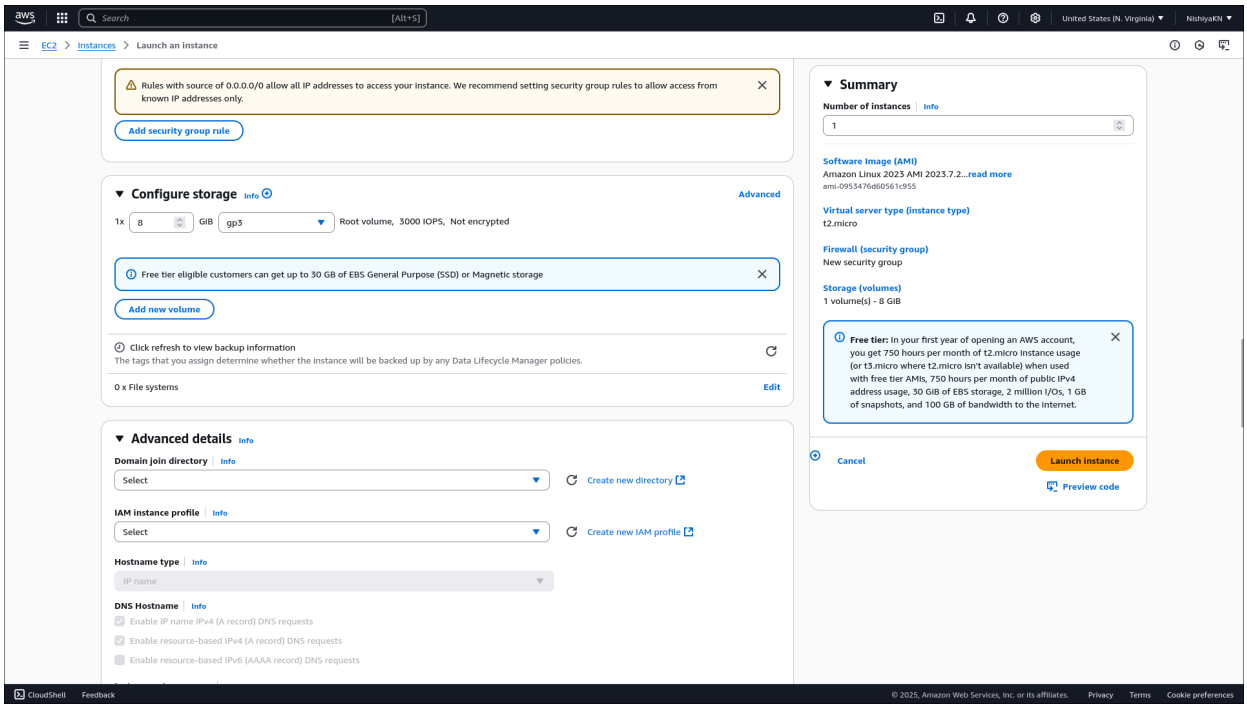
Configuração da AWS

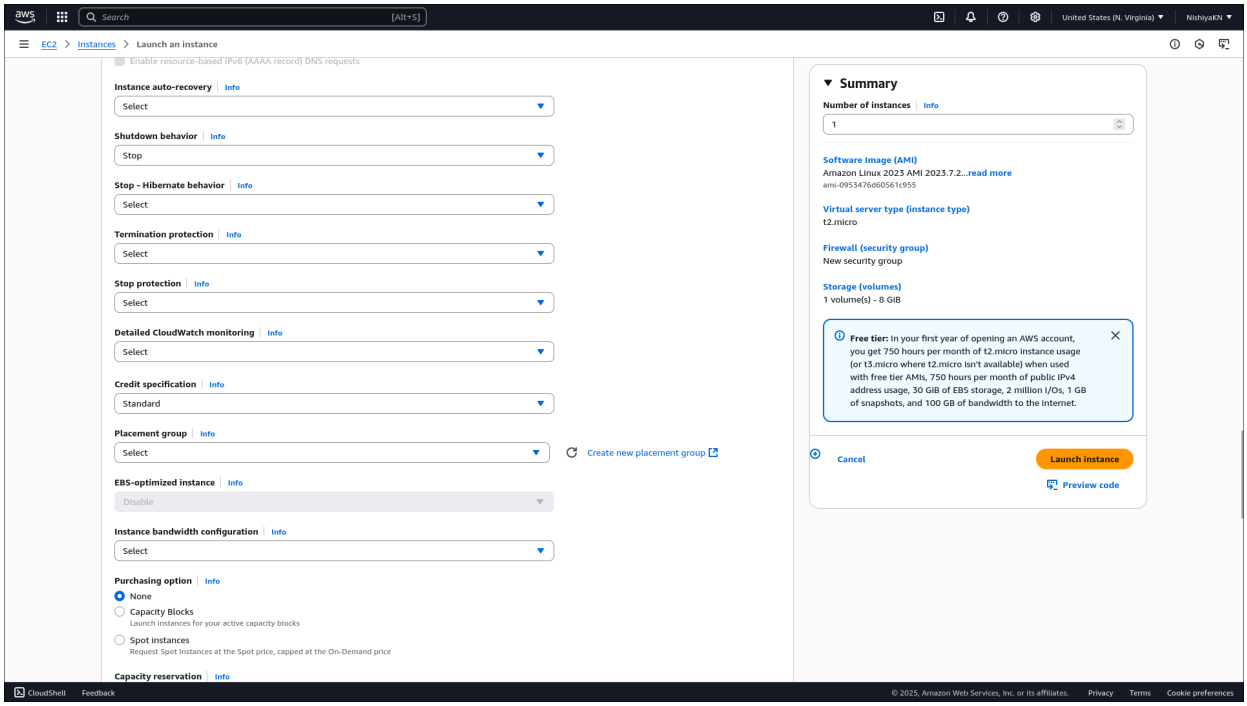
O projeto utiliza uma instância t2.micro (EC2) na AWS, onde possui dois containers Docker: price-tracker e cido-web.

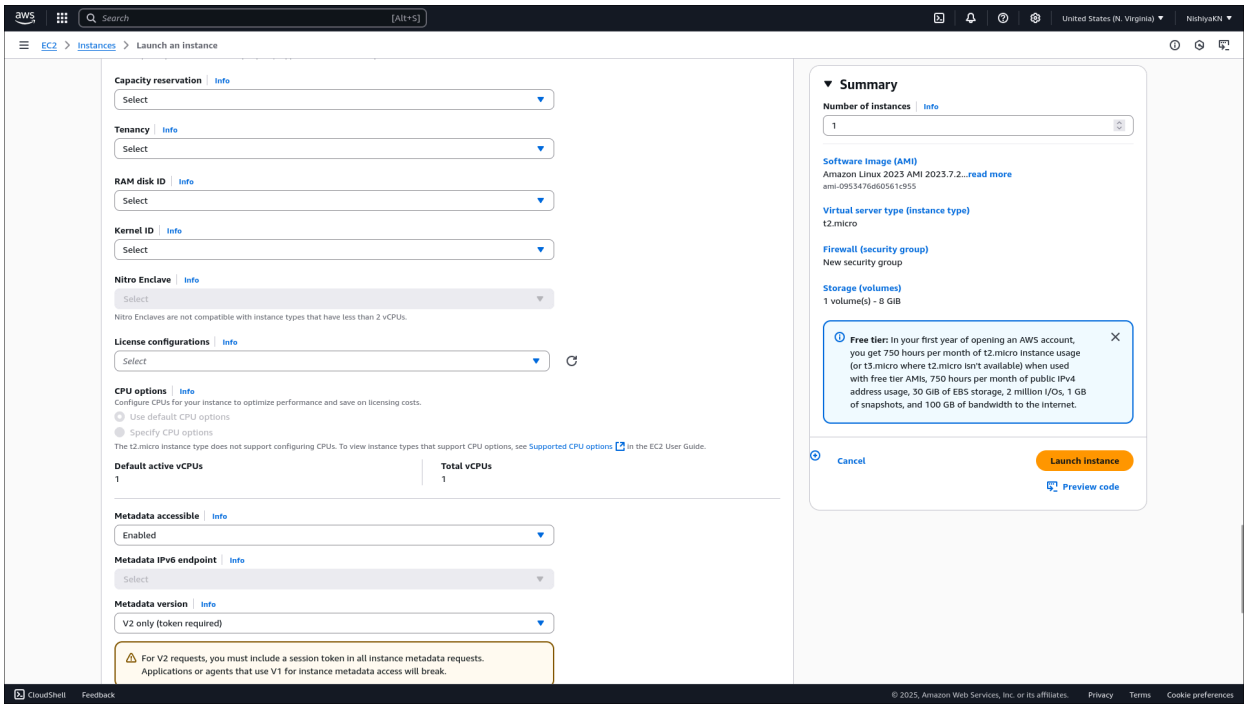

Search [Alt+S]
United States (N. Virginia)
NishiyaKN
EC2 > Instances > Launch an instance
It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices
Do not show me this message again
Take a walkthrough
Launch an 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
gpu-price-tracker
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 AMI
ami-0953476d60561c955 (64-bit (x86), uefi-preferred) / ami-05a3e0187917e3e24 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible
Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.
Amazon Linux 2023 AMI 2023.7.20250512.0 x86_64 HVM kernel-6.1
Architecture
Boot mode
AMI ID
Publish Date
Username
64-bit (x86)
uefi-preferred
ami-0953476d60561c955
2025-05-09
ec2-user
Verified provider
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0953476d60561c955
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

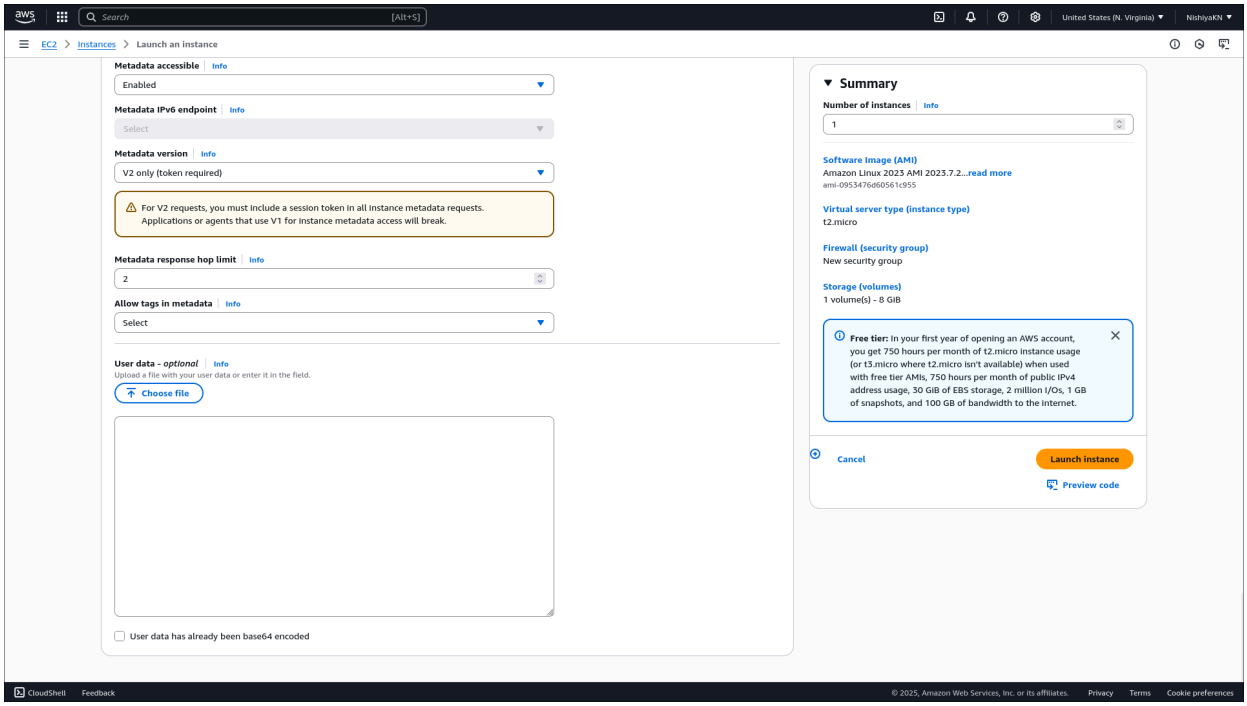

Search [Alt+S]
United States (N. Virginia)
NishiyaKN
EC2 > Instances > Launch an instance
Architecture
Boot mode
AMI ID
Publish Date
Username
64-bit (x86)
uefi-preferred
ami-0953476d60561c955
2025-05-09
ec2-user
Verified provider
Instance type Info | Get advice
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
awskey
Create new key pair
Network settings Info
VPC - required Info
vpc-0ebcb302fa5923e33
172.31.0.0/16
(default)
Subnet Info
No preference
Create new subnet
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Security group name - required
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0953476d60561c955
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

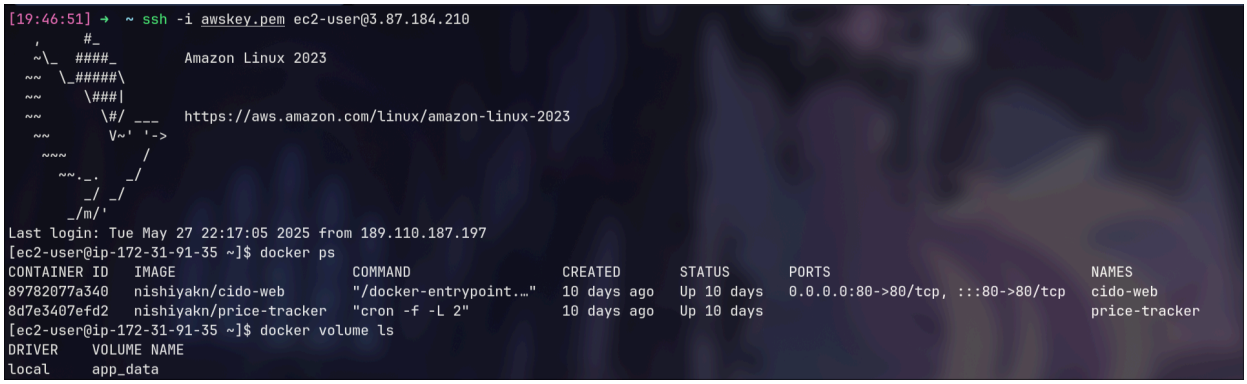

Search [Alt+S]
United States (N. Virginia)
NishiyaKN
EC2 > Instances > Launch an instance
Security group name - required
launch-wizard-1
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*
Description - required Info
launch-wizard-1 created 2025-05-17T18:50:40.015Z
Inbound Security Group Rules
Security group rule 1 (TCP, 22, 0.0.0.0/0)
Remove
Type Info
ssh
Protocol Info
TCP
Port range Info
22
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or security group
Description - optional Info
e.g. SSH for admin desktop
0.0.0.0/0
Security group rule 2 (TCP, 443, 0.0.0.0/0)
Remove
Type Info
HTTPS
Protocol Info
TCP
Port range Info
443
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or security group
Description - optional Info
e.g. SSH for admin desktop
0.0.0.0/0
Security group rule 3 (TCP, 80, 0.0.0.0/0)
Remove
Type Info
HTTP
Protocol Info
TCP
Port range Info
80
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or security group
Description - optional Info
e.g. SSH for admin desktop
0.0.0.0/0
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0953476d60561c955
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
United States (N. Virginia)
NishiyaKN
EC2 > Instances > Launch an instance
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Add security group rule
Configure storage Info
Advanced
1x 8 GiB gp3 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
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.
0 x File systems
Edit
Advanced details Info
Domain join directory Info
Select
Create new directory
IAM instance profile Info
Select
Create new IAM profile
Hostname type Info
IP name
DNS Hostname Info
Enable IP name IPv4 (A record) DNS requests
Enable resource-based IPv4 (A record) DNS requests
Enable resource-based IPv6 (AAAA record) DNS requests
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0953476d60561c955
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
United States (N. Virginia)
NishiyaKN
EC2 > Instances > Launch an instance
Enable resource-based IPv6 (AAAA record) DNS requests
Instance auto-recovery Info
Select
Shutdown behavior Info
Stop
Stop - Hibernate behavior Info
Select
Termination protection Info
Select
Stop protection Info
Select
Detailed CloudWatch monitoring Info
Select
Credit specification Info
Standard
Placement group Info
Select
Create new placement group
EBS-optimized instance Info
Disable
Instance bandwidth configuration Info
Select
Purchasing option Info
None
Capacity Blocks
Launch instances for your active capacity blocks
Spot instances
Request Spot Instances at the Spot price, capped at the On-Demand price
Capacity reservation Info
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0953476d60561c955
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
United States (N. Virginia)
NishiyaKN
EC2 > Instances > Launch an instance
Capacity reservation Info
Select
Tenancy Info
Select
RAM disk ID Info
Select
Kernel ID Info
Select
Nitro Enclave Info
Select
Nitro Enclaves are not compatible with instance types that have less than 2 vCPUs.
License configurations Info
Select
CPU options Info
Configure CPUs for your instance to optimize performance and save on licensing costs.
Use default CPU options
Specify CPU options
The t2.micro instance type does not support configuring CPUs. To view instance types that support CPU options, see Supported CPU options in the EC2 User Guide.
Default active vCPUs
1
Total vCPUs
1
Metadata accessible Info
Enabled
Metadata IPv6 endpoint Info
Select
Metadata version Info
V2 only (token required)
For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0953476d60561c955
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences





Price-Tracker

Possui diversos scripts escritos em python que periodicamente são ativados pelo crond. Os scripts acessam os websites da Kabum, Terabyte e Pichau para verificar os valores de certas GPUs. Quando é possível obter os valores, eles são armazenados em arquivos JSON, que estão em um volume compartilhado.
Usamos o seguinte comando para rodá-lo:
docker run -d --name price-tracker -v app_data:/app nishiyakn/price-tracker

Cido-web

Possui o nginx rodando para fornecer o website. Ele pega os arquivos JSON do volume compartilhado e cria gráficos para os valores de cada item conforme o tempo.
Usamos o seguinte comando para rodá-lo:
docker run -d --name cido-web --restart always -p 80:80 -v app_data:/usr/share/nginx/html/price_history nishiyakn/cido-web

Trechos de Código

Login:

O trecho de código de login é usado para registrar o usuário a usar as dependências do container, utilizando-se de uma função para definir um “nome de usuário” e “senha”.

Dados:

O trecho de código de dados é usado para comparar preços e datas de certos produtos, sendo eles placas de vídeo, com o intuito de analisar a evolução dos preços do produto com o preço da categoria e/ou do produto correspondente. A organização dos dados é feita definindo o nome do produto, criando assim um objeto, e dando características a ele como data, que utiliza apenas números ex.: `"date": "2025-04-17"`
, e o preço que é definido utilizando uma string para referenciarmos o mesmo futuramente, ex.: `"price": 1889.99`

Grafico.js:

Demonstra uma tabela feita para armazenar as informações recebidas pelos dados, dando uma forma mais amigável para o usuário ler os dados disponíveis de forma organizada.
Para realizarmos isso, precisamos primeiro pegar as informações da tabela dados(1,2,3) com o comando FETCH, após isso fazemos com que o gráfico leia os arquivos em json utilizando a função “.then (res => res.json())”, nós então coletamos os dados utilizando a mesma função e modificamos os atributos dos itens para aparecerem de maneira correta e alterando a cor dos textos para não confundirmos o que cada um é.
Após a modificação, prosseguimos para a criação dos conteiner dos dados onde ficariam visíveis ao usuário, mas não modificáveis. Lá contém a aparência dos containers, qual seu tamanho, posição, se será responsivo ao click do mouse, cor do texto, cor do contêiner, quais dados ele irá armazenar e de qual repositório de dados ele irá adquirir informações.

Utilização do Docker:

Dockerfile:

O Comando **FROM** do arquivo é usado para puxar a imagem original (Nginx) do Docker Hub e a trazendo para o modelo web.
O comando **COPY** é utilizado para copiar todos os arquivos estáticos do website com exceção dos arquivos JSON sendo eles o index, o gráfico(html e javascript), login(CSS e javascript), as configurações do nginx, os parâmetros de estilo do site(CSS) e uma imagem.
O comando **EXPOSE** para expor a porta 80 utilizada pelo programa.
E por fim o comando **CMD** para iniciar o Nginx.

Docker Compose:

Os dados do docker compose são as configurações da aplicação docker sendo utilizada, que incluem: sua versão atual, serviços em utilização, nome do arquivo docker, porta configurada, nome do container em utilização e arquivos/dados carregados.

Dockerfile no Scraper:

Aqui é configurado alguns arquivos essenciais para que a aplicação funcione corretamente.
Como necessitamos de uma imagem usamos o comando FROM python:3.11-slim para usar o sistema de imagem do python chamado python slim image.
Também precisamos de um diretório que iremos usar, então usamos o comando WORKDIR /app para pegá-lo.
Alguns deles requerem que instalemos dependências como as dependências de sistemas e a do python. Para isso usamos o comando **RUN** para fazer com que uma linha de comando específica faça com que o computador busque o arquivo na internet ou localmente e instale o necessário.
Precisamos definir a data e hora, para isso precisamos pegar a timezone da região utilizando o comando ENV TZ=America/Sao_Paulo.
Usamos esse comando com as linhas de código para instalar dependências que o sistema precisa para

```
apt-get update && \
apt-get install -y --no-install-recommends gcc python3-dev cron vim less tzdata && \
rm -rf /var/lib/apt/lists/*
```

Também pode ser visto em RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone pagar definirmos a data correta e atual e horário; RUN pip install --no-cache-dir -r requirements.txt para instalar dependências do python; RUN chmod 0777 /app para rodar o aplicativos; RUN chmod 0644 /etc/cron.d/price-tracker é usado para rodar a função que irá coletar os dados das lojas; RUN crontab /etc/cron.d/price-tracker é usado para que a função de coleta de dados funcione a todo o tempo, mesmo com o arquivo desligado.
É necessário também copiar alguns arquivos localizados em outra pasta/diretório para que os conteúdos sejam atrelados ao arquivo atual e lidos pelo programa. Para esse fim utilizamos o comando **COPY** requirements.txt para copiar o cachê do docker; COPY src /app/ copia os arquivos do app; COPY cronjob /etc/cron.d/price-tracker Copia uma função que adquire os preços das lojas.
Por fim utilizamos o comando CMD ["cron", "-f", "-L", "2"] para iniciar o crond.