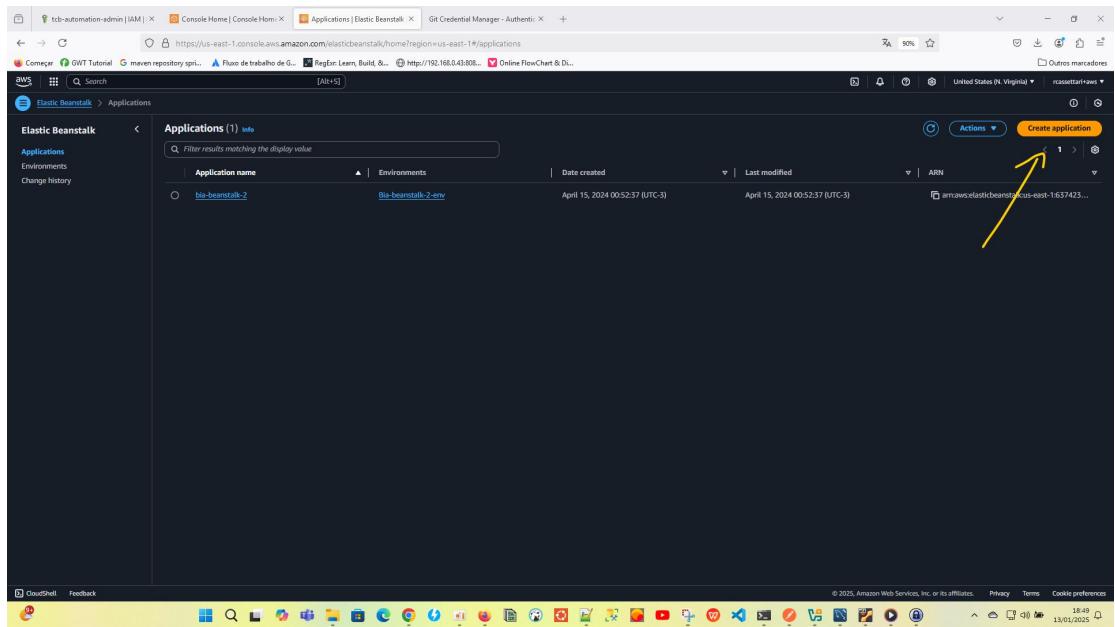
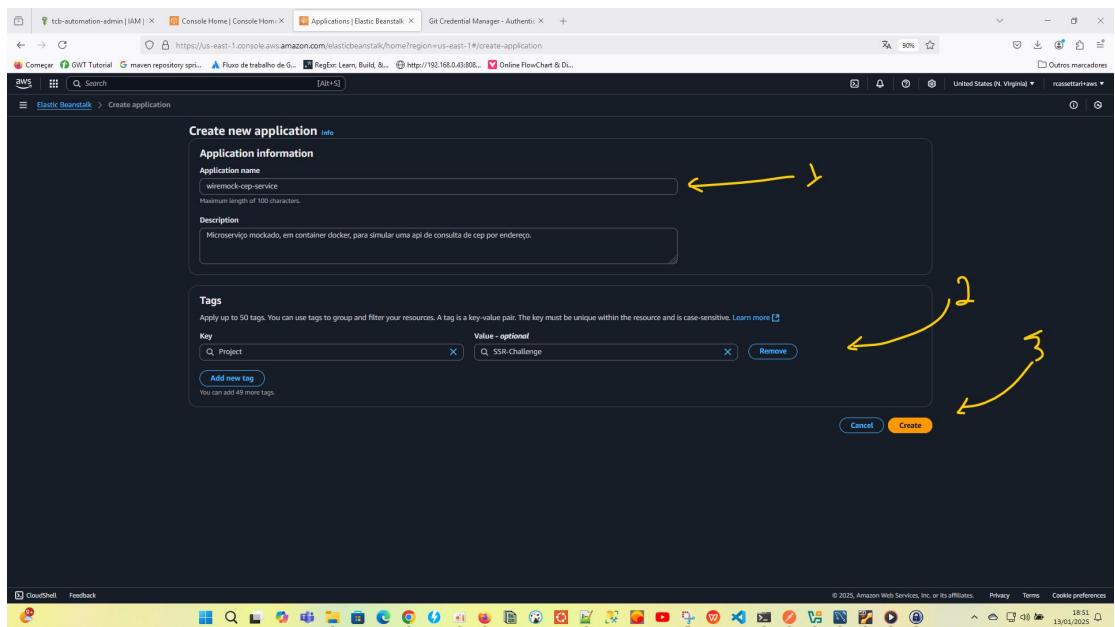


## 1 - Início criação de uma aplicação nova no AWS Elastic BeansTalk:



## 2 - Defino nome da aplicação para o Elastic Beans Talk, descrição, crio um tag para ela e clico em "create":



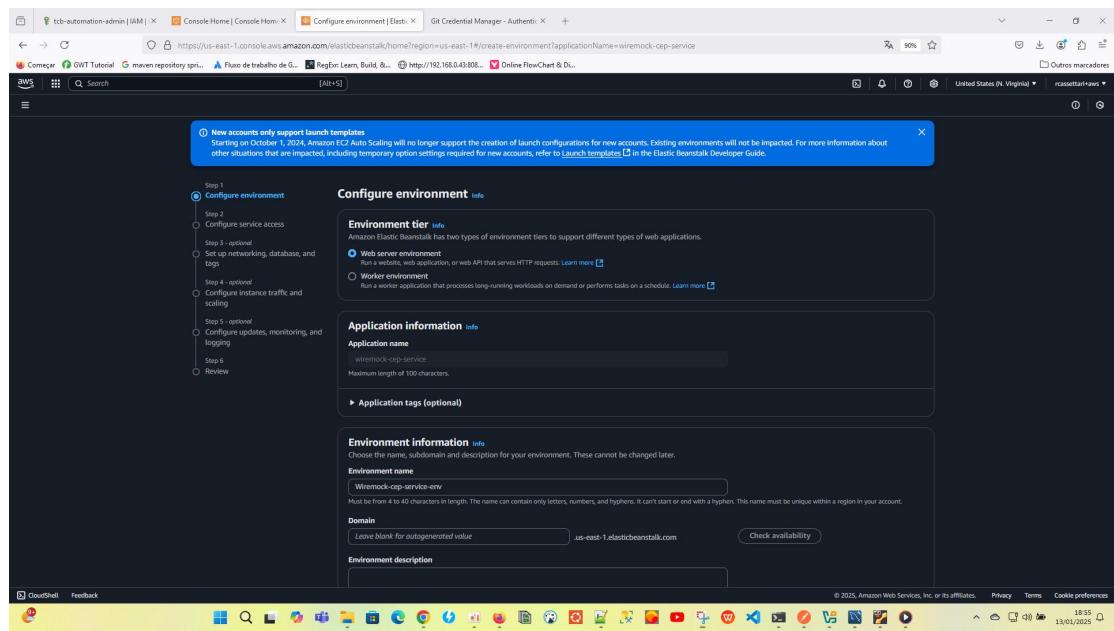
3 - Aplicação criada, mas ainda não temos nenhum environment para essa aplicação no beanstalk:

The screenshot shows the AWS Elastic Beanstalk console. In the left sidebar, under 'Application: wiremock-cep-service', there are 'Application versions' and 'Saved configurations'. The main area is titled 'Application wiremock-cep-service environments (0)' and contains a message: 'No environments. No environments currently exist for this application.' Below this message is a blue button labeled 'Create environment'. The top navigation bar includes tabs for 'Console Home', 'AWS', 'Actions', and 'Create new environment'. The status bar at the bottom right shows the date and time: '13/01/2025 18:52'.

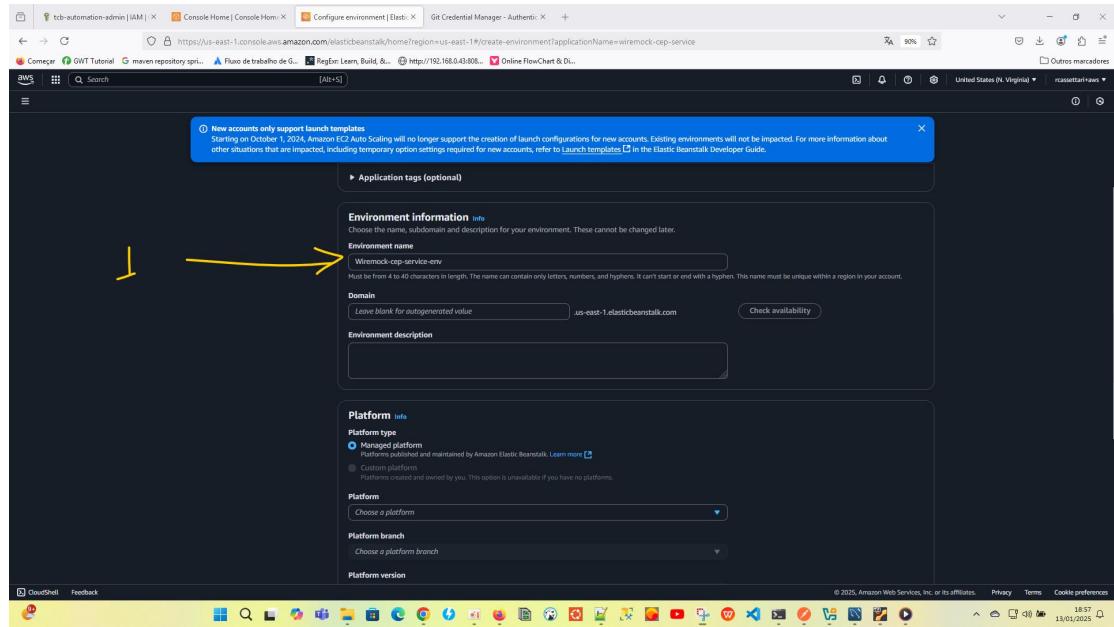
4 - Clico em “Create Environment” para iniciar a criação do ambiente:

This screenshot is identical to the previous one, showing the 'wiremock-cep-service' application page with no environments. However, a yellow arrow points to the 'Create environment' button, indicating the user's intended action. The rest of the interface and status bar are the same as in the first screenshot.

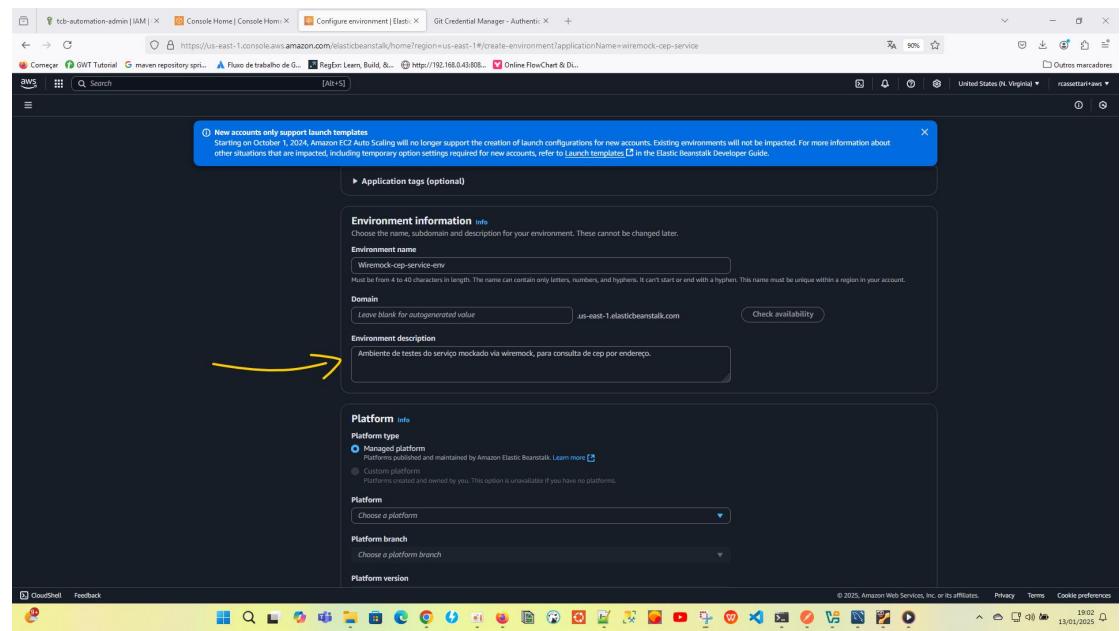
5 - A tela inicial do create environment é essa aqui:



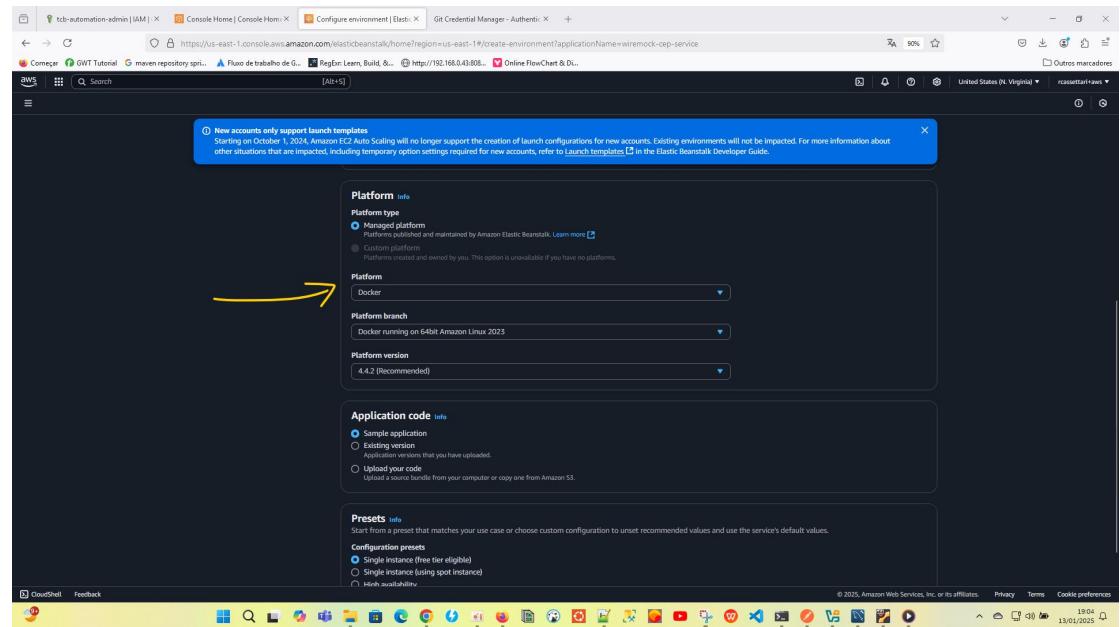
6 - Mantendo o nome do environment que ele sugeriu, derivado do nome da aplicação:



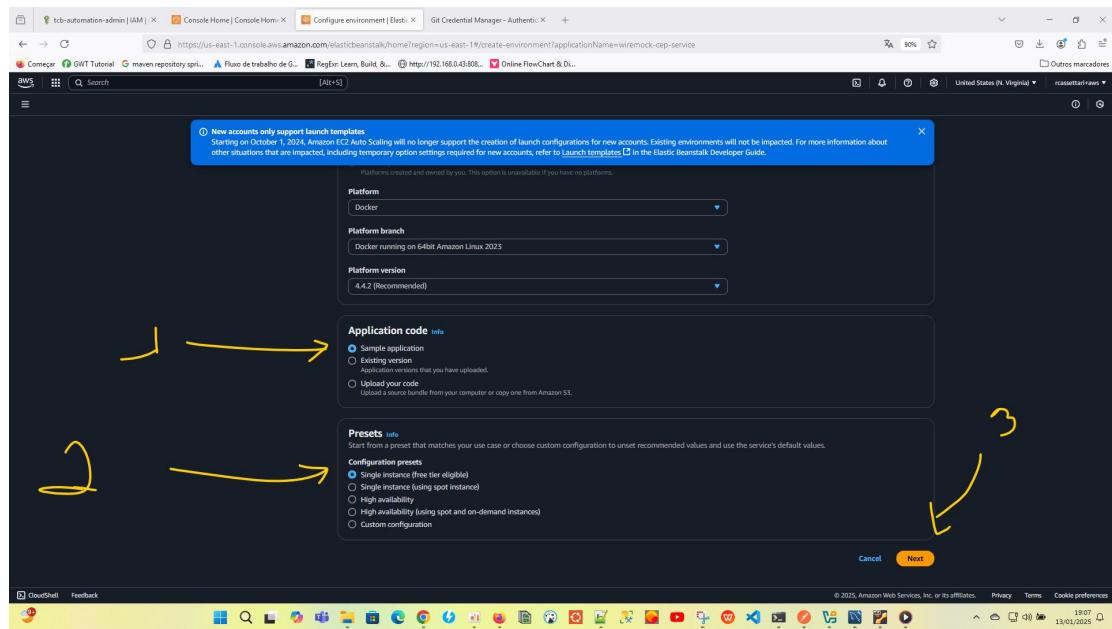
## 7 - Defino uma descrição para o environment:



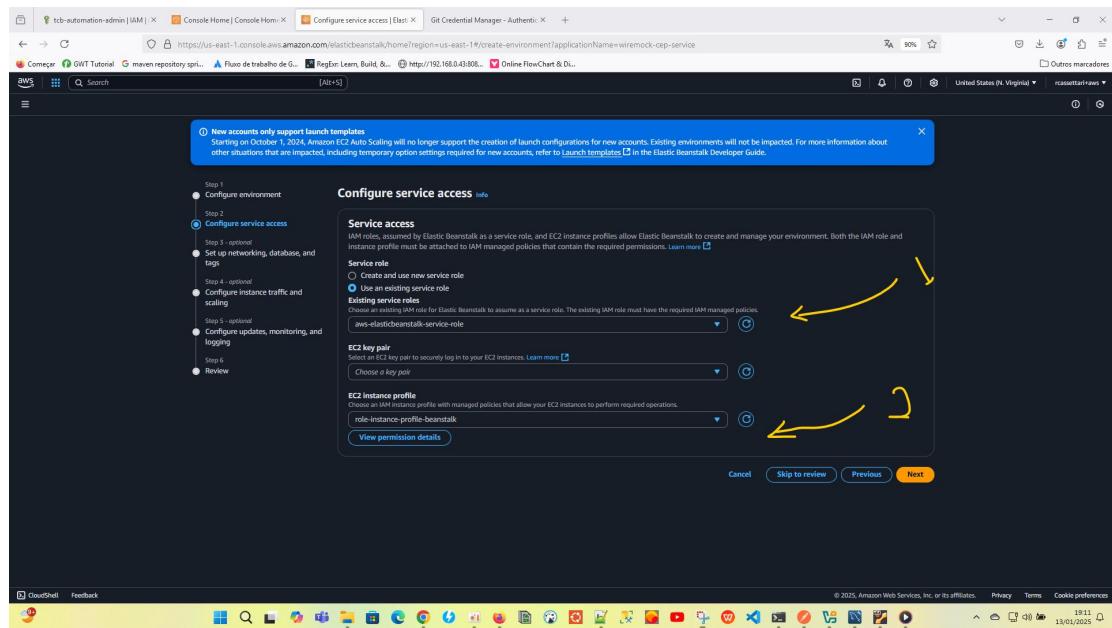
## 8 - Em platform, escolho “Docker”:



9 - O application e presets, deixa tudo default por enquanto, carregando a “sample application” mesmo, com single instance no presets e clico em “next”:



10 - Na configuração do service access, escolho alguns roles que eu já tinha anteriormente criado, para service role (“aws-elasticbeanstalk-service-role”) e, em “EC2-instance-profile”, escolho o role “role-instance-profile-beanstalk”:



## 11 - Revejo os permissions em uso no “role-instance-profile-beanstalk”:

The screenshot shows the AWS IAM 'Roles' section. A specific role named 'role-instance-profile-beanstalk' is selected. The 'Permissions' tab is active, displaying three managed policies attached to the role:

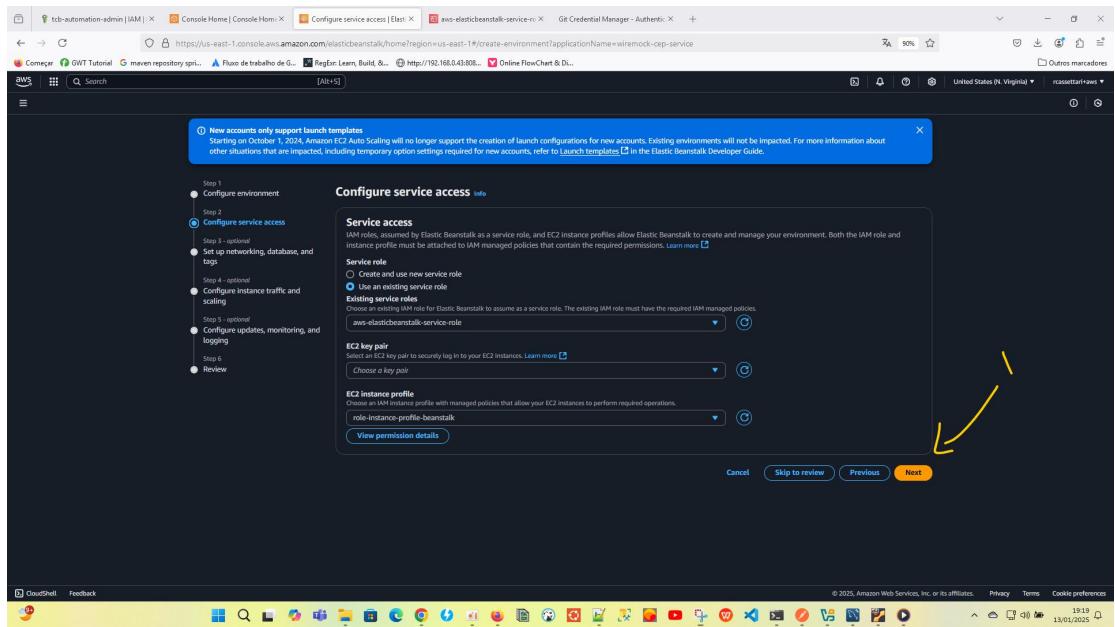
Policy name	Type	Attached entities
AmazonEC2ContainerRegistryPowerUser	AWS managed	1
AmazonSSMManagedInstanceCore	AWS managed	2
AWSLambdaBasicExecutionRole	AWS managed	1

## 12 - Revejo as permissões em uso em “aws-elasticbeanstalk-service-role”:

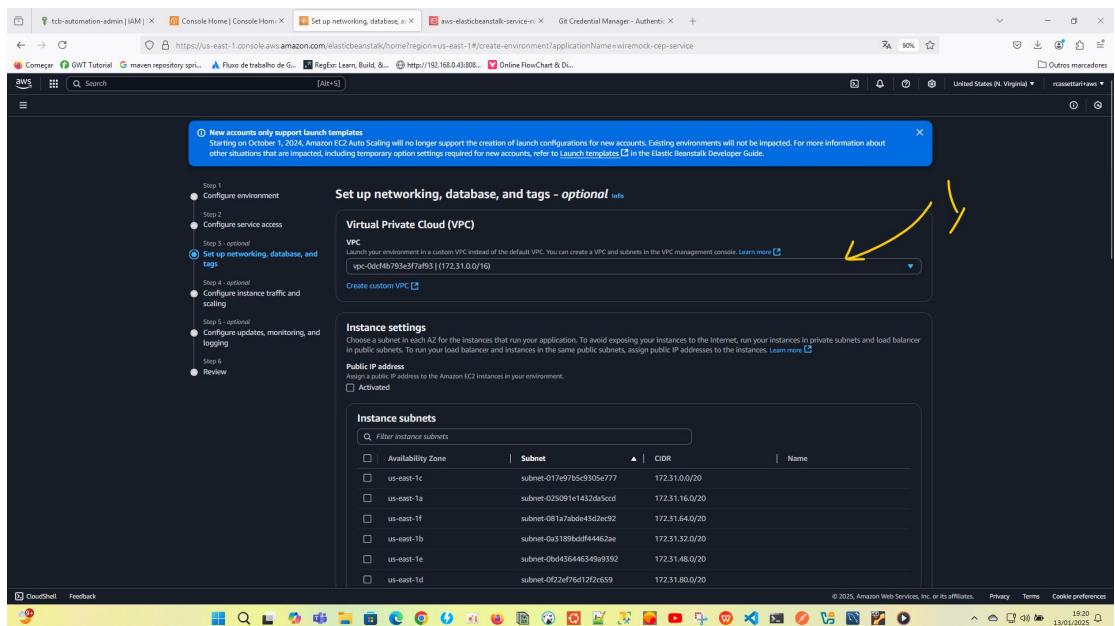
The screenshot shows the AWS IAM 'Roles' section. A specific role named 'aws-elasticbeanstalk-service-role' is selected. The 'Permissions' tab is active, displaying two managed policies attached to the role:

Policy name	Type	Attached entities
AWSLambdaBasicExecutionRole	AWS managed	1
AWSLambdaVPCExecutionRole	AWS managed	1

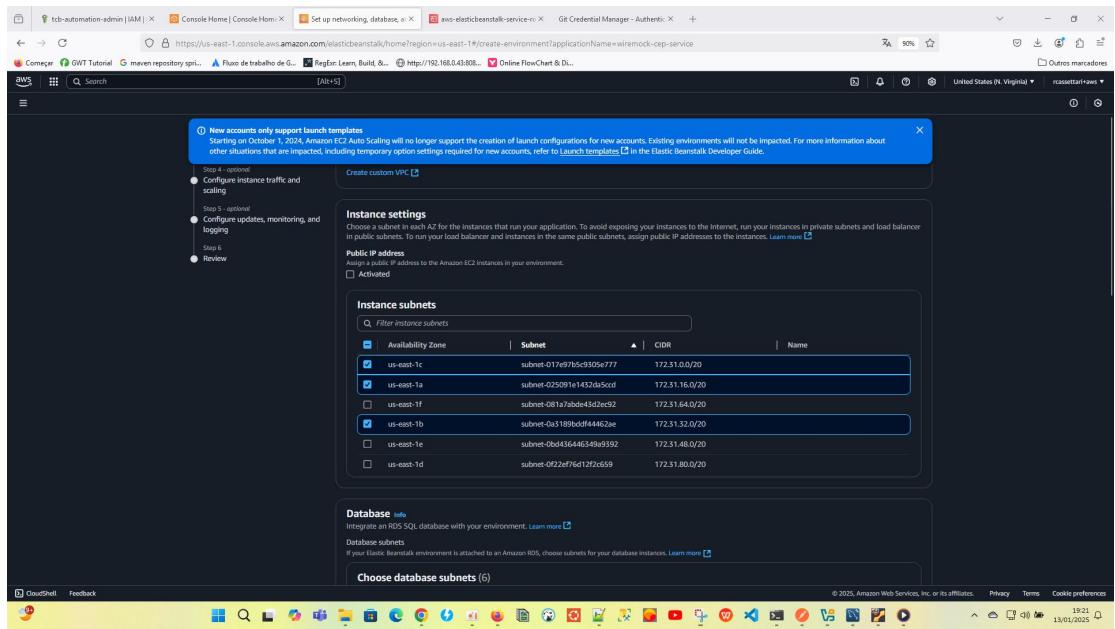
13 - Volto para a aba do navegador que estava parada no “Configure service access” e clico em “next”:



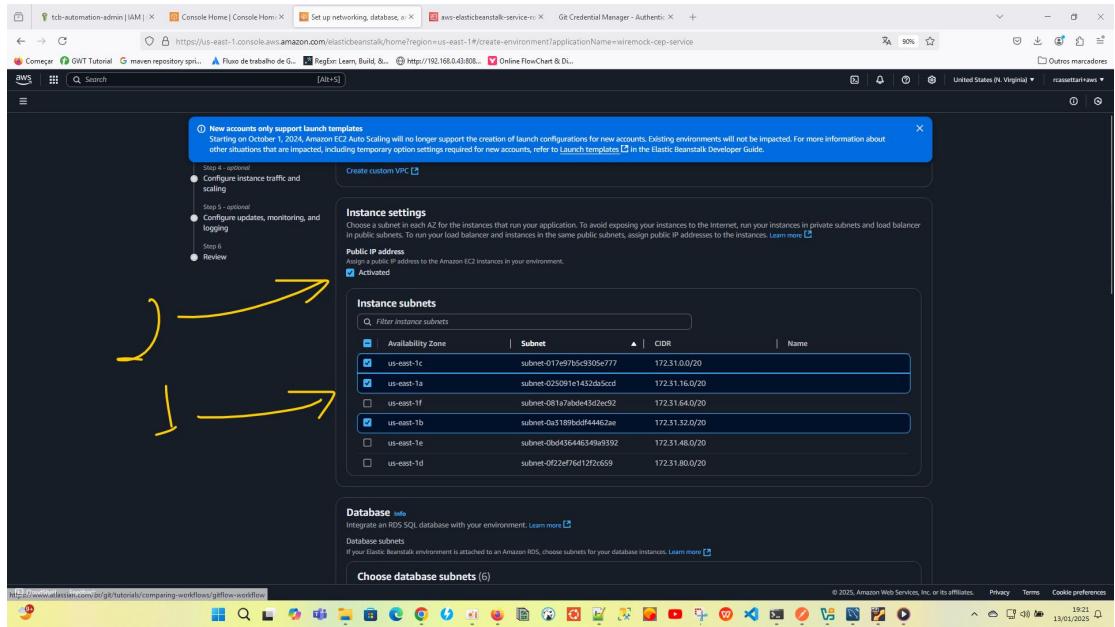
14 - Escolho a VPC default:



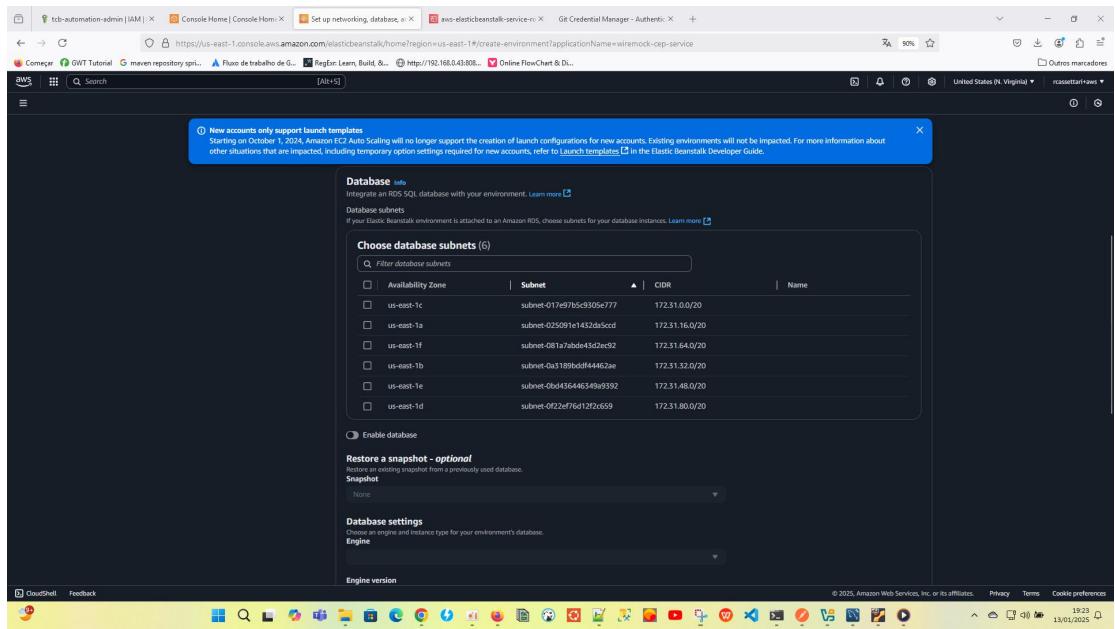
## 15 - Escolho 3 subnets, 1a, 1b e 1c:



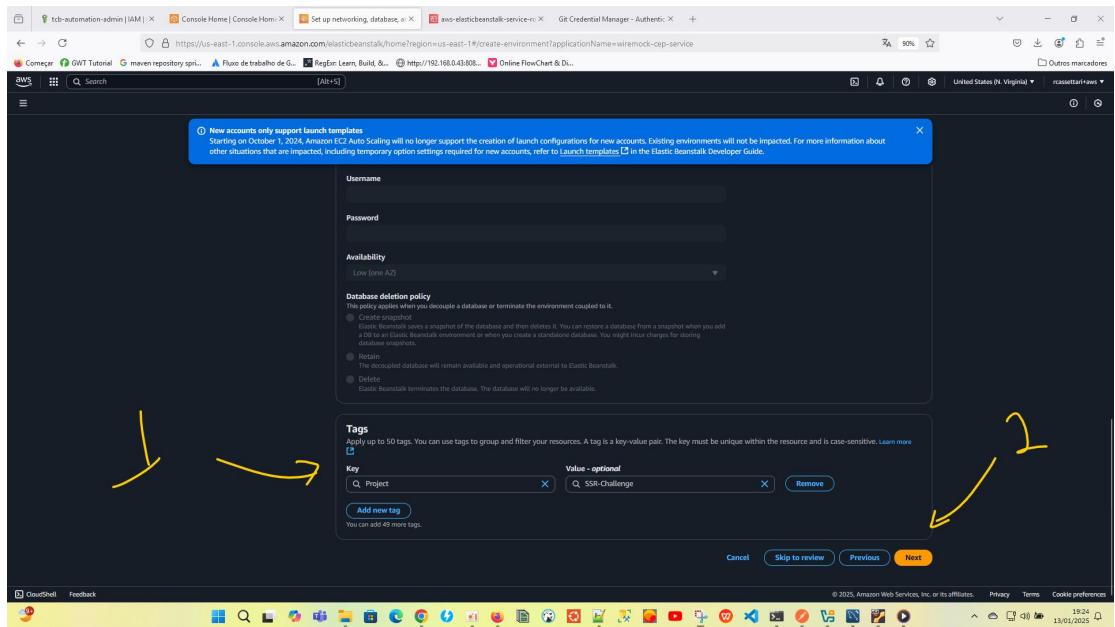
## 16 - Além de selecionar as 3 subnets, eu ativo o “Public IP address”:



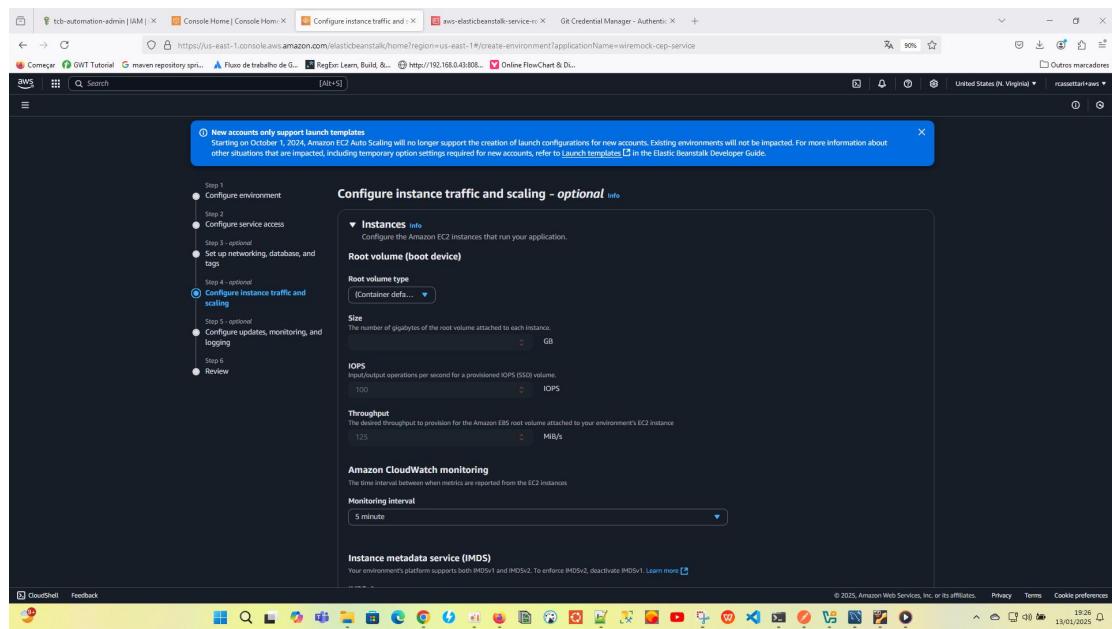
17 - Não habilito database me “Enable database” ( o wiremock do container não usa database):



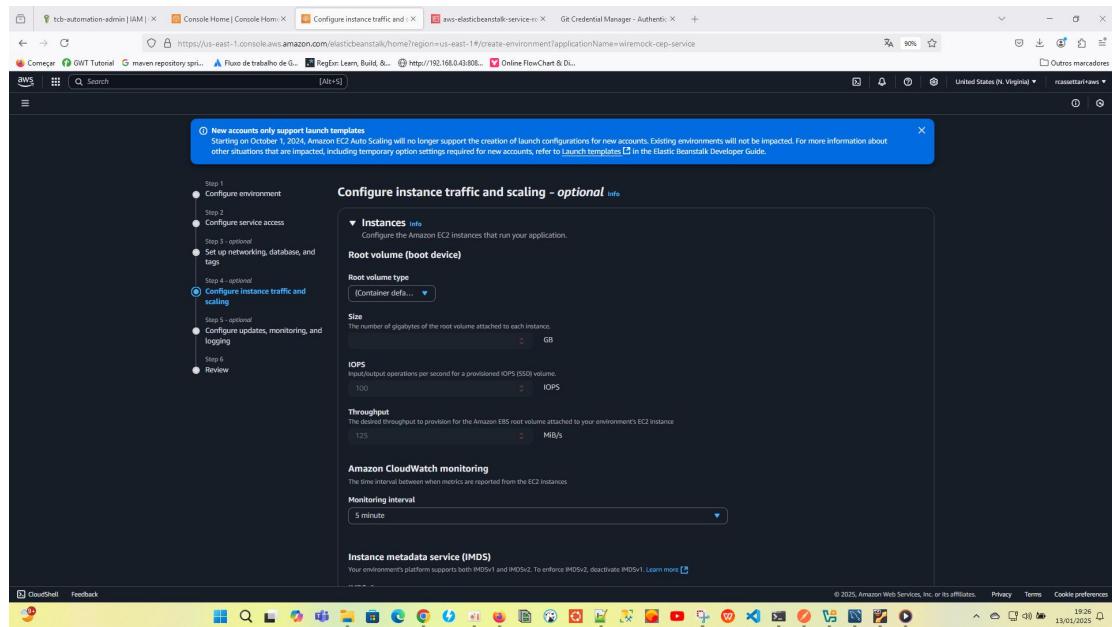
18 - Coloco uma tag, identificando o projeto do desafio “SSR-Challenge” e clico em “Next”:



19 - Entro na tela do “Configure instance traffic”, que é assim:



20 - Mantinho tudo padrão nessa parte inicial:



21 - Mantendo o checkbox IMD5v1, como desativado ( evita ativar o endpoint que mostra detalhes das instances e usa o v2, mais seguro):

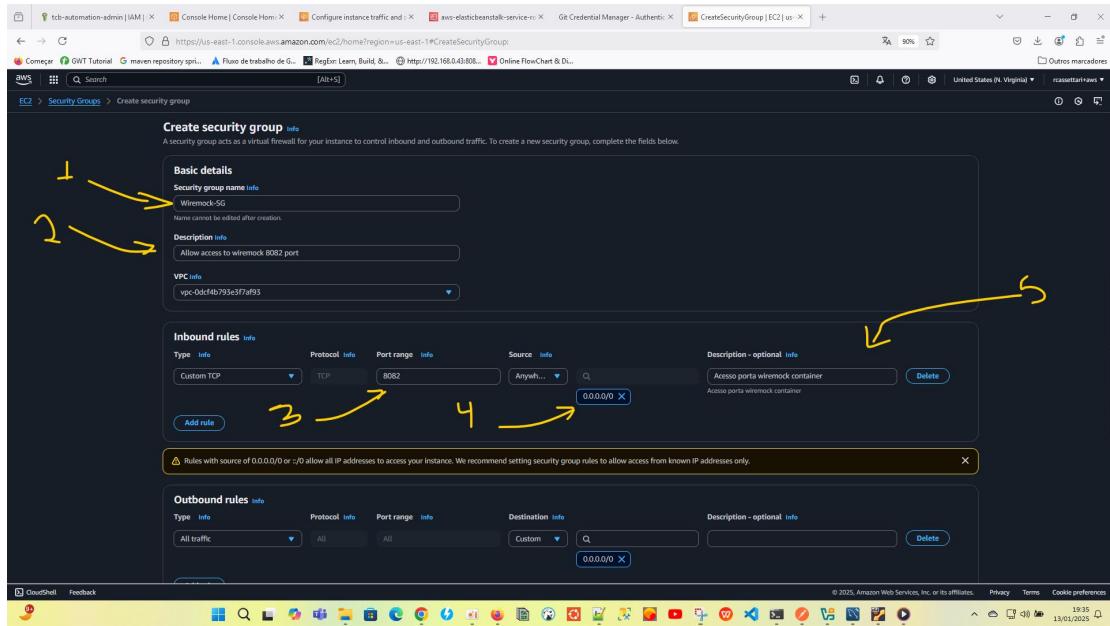
The screenshot shows the AWS Elastic Beanstalk 'Configure instance traffic and...' page. A modal window titled 'Instance metadata service (IMDS)' is open, containing a message about IMDSv1 being deprecated. Below the message, there is a checkbox labeled 'Desactivated'. A yellow arrow points to this checkbox.

22 - Crio novo security group:

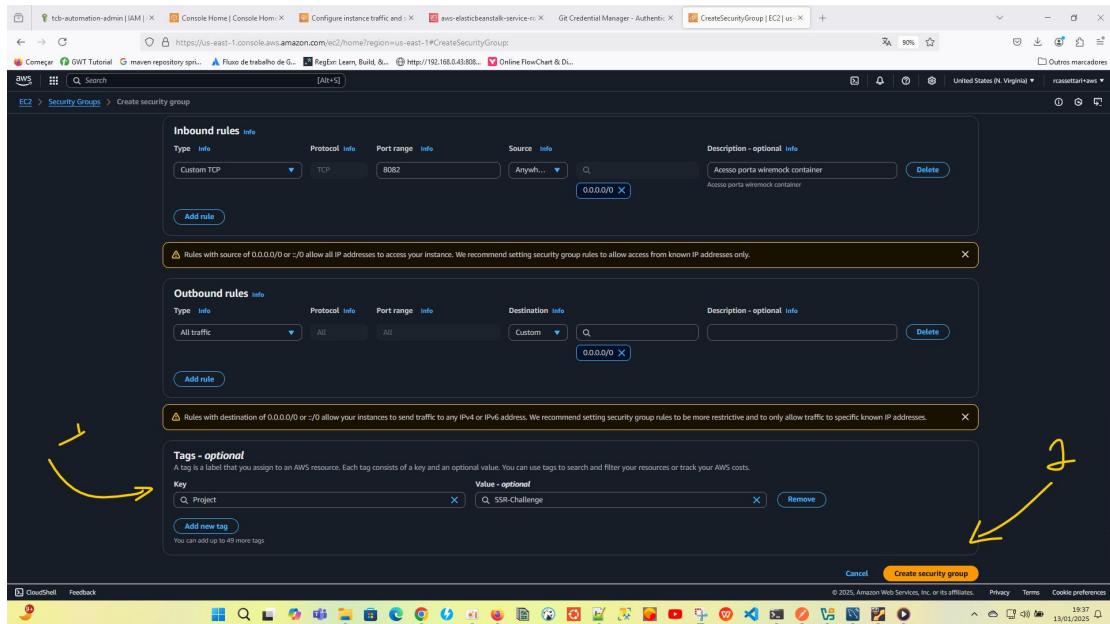
The screenshot shows the AWS EC2 'Security groups' page. A new security group named 'bla-alb' has been created and is listed in the table. A yellow arrow points to the 'Create security group' button at the top right of the table.

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count
-	sg-053f798d9f3bb439	bla-alb	vpc-0cd4fb793e3f7a931	acesso do bla alb	637423554202	2 Permission entries
-	sg-01470063b0733e1e	EC2toRDS-sg	vpc-0cd4fb6d1995c5ed1	Sg para permitir acesso ao MySQL atrav...	637423554202	1 Permission entry
-	sg-083bc04a4c891457	bla-beanstalk	vpc-0cd4fb793e3f7a931	acesso na porta 80	637423554202	1 Permission entry
-	sg-07220cff0b65729ff	default	vpc-0cd4fb793e3f7a931	default VPC security group	637423554202	1 Permission entry
-	sg-098ba24fd9e547cd	vs-code-ssh	vpc-0cd4fb793e3f7a931	vs-code-ssh	637423554202	1 Permission entry
-	sg-0a754812d95d9191	default	vpc-0cd4fb793e3f7a931	default VPC security group	637423554202	1 Permission entry
-	sg-03976d82bd5d56e84	bla-db	vpc-0cd4fb793e3f7a931	acesso do bla db	637423554202	4 Permission entries

23 - Dou nome no security group, coloco descrição, defino port range “8082”, abro para qualquer IP “0.0.0.0” e defino uma descrição para o inbound rule:



24 - Defino um tag para o projeto e clico em “Create Security Group”:



25 - Vejo a mensagem indicando que o novo security group foi criado:

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroupList>. The page displays a success message: "Security group [sg-059585e9d021115b8 | Wiremock-SG] was created successfully". Below this, the "Details" section shows the security group name "Wiremock-SG", ID "sg-059585e9d021115b8", owner "637423554202", and a single inbound rule allowing port 8082. The "Inbound rules" table lists one rule: "sg-04fcfcf995c40fc0e9" (IPv4) on TCP port 8082 from "0.0.0.0/0" with the source "Acesso porta wiremock". The left sidebar shows various AWS services like IAM, AMI Catalog, and Network & Security.

26 - Na tela do beans talk, que estava parada na outra aba, dou um refresh na lista de security groups:

The screenshot shows the AWS Elastic Beanstalk Security Groups page with the URL <https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#SecurityGroupList>. A blue banner at the top states: "New accounts only support launch templates" and "Starting on October 1, 2024, Amazon EC2 Auto Scaling will no longer support the creation of launch configurations for new accounts. Existing environments will not be impacted. For more information about other account types that are impacted, including temporary option settings required for new accounts, refer to Launch templates in the Elastic Beanstalk Developer Guide." Below this, the "EC2 security groups" section lists nine security groups: awseb-e-irnw4tvb9-stack-AWSEBSecurityGroup, bia-alb, bia-beanstalk, bia-db, bia-dev, bia-ec2, bia-web, default, and vs-code-ssh. A yellow arrow points to the "Create New" button in the top right corner of the list. The bottom section shows "Capacity info" and "Auto scaling group" configuration options.

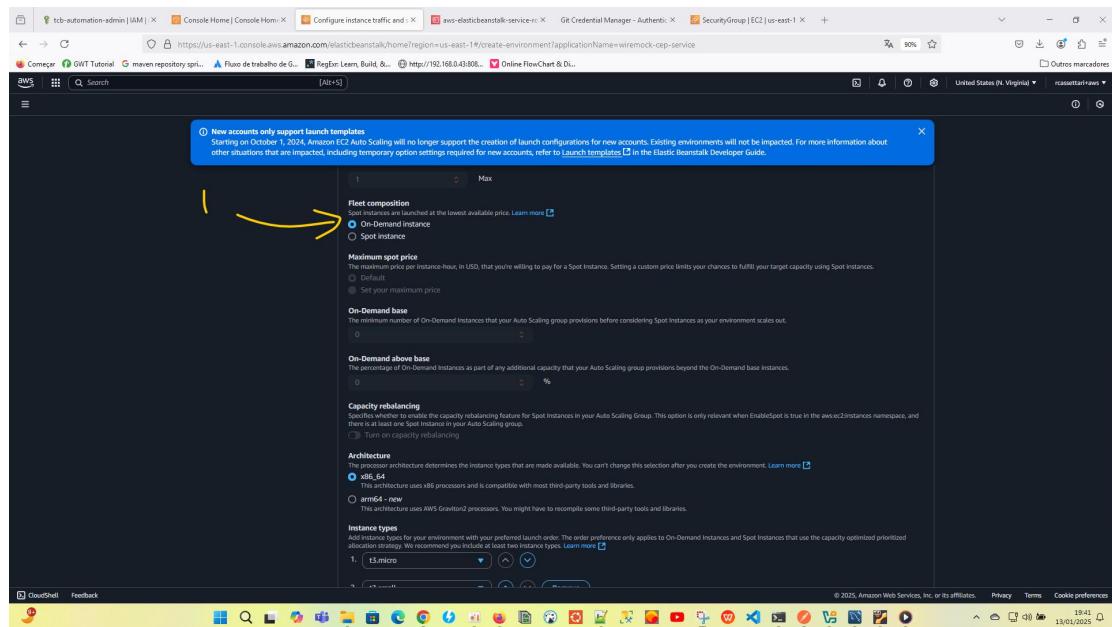
27 - Seleciono meu security group previamente criado ( mas para o perfil web-tier, o beanstalk usa um security group que ele cria para porta 80 também ):

The screenshot shows the AWS Elastic Beanstalk 'Create New Environment' wizard. In the 'Configure instance traffic and...' step, under the 'EC2 security groups' section, a list of existing security groups is displayed. One group, 'Wiremock-SG', is selected and highlighted with a blue border. A yellow arrow points from the left towards this selection. Another yellow arrow points from the bottom towards the 'Wiremock-SG' entry in the list.

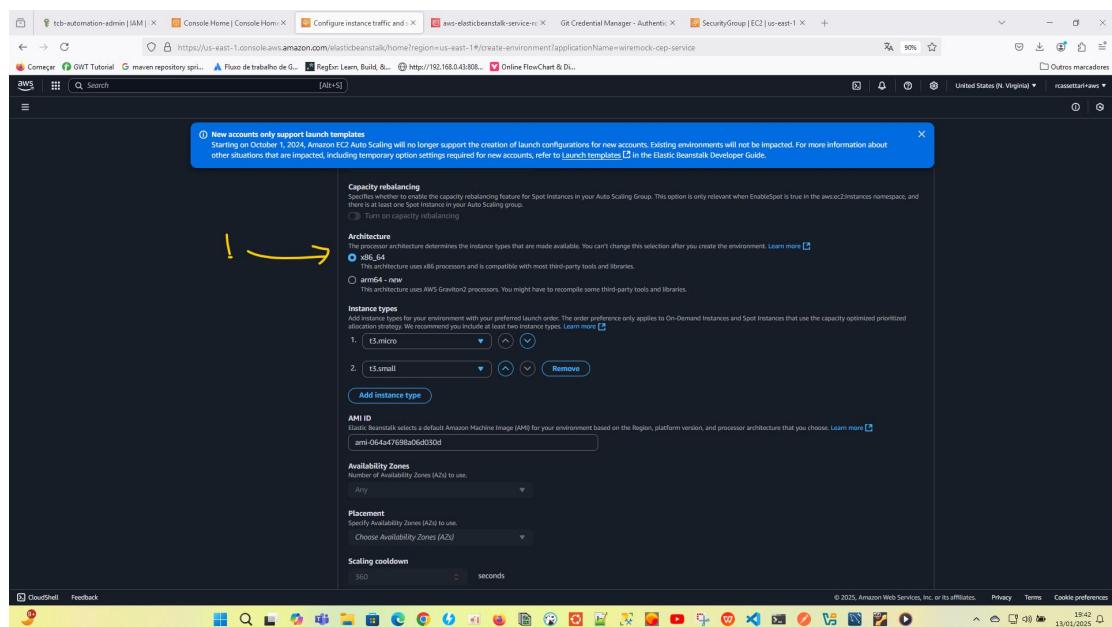
28 - Em capacity, como só quero fazer teste, mantenho em “Single Instance” e mantenho o número de instances como “1” Min e “1” no Max.

The screenshot shows the 'Capacity' configuration step of the Elastic Beanstalk wizard. Under the 'Auto scaling group' section, the 'Environment type' dropdown is set to 'Single instance'. The 'Instances' section shows '1' for both 'Min' and 'Max' values. A yellow arrow points from the left towards the 'Single instance' dropdown. Another yellow arrow points from the bottom towards the 'Min' and 'Max' input fields.

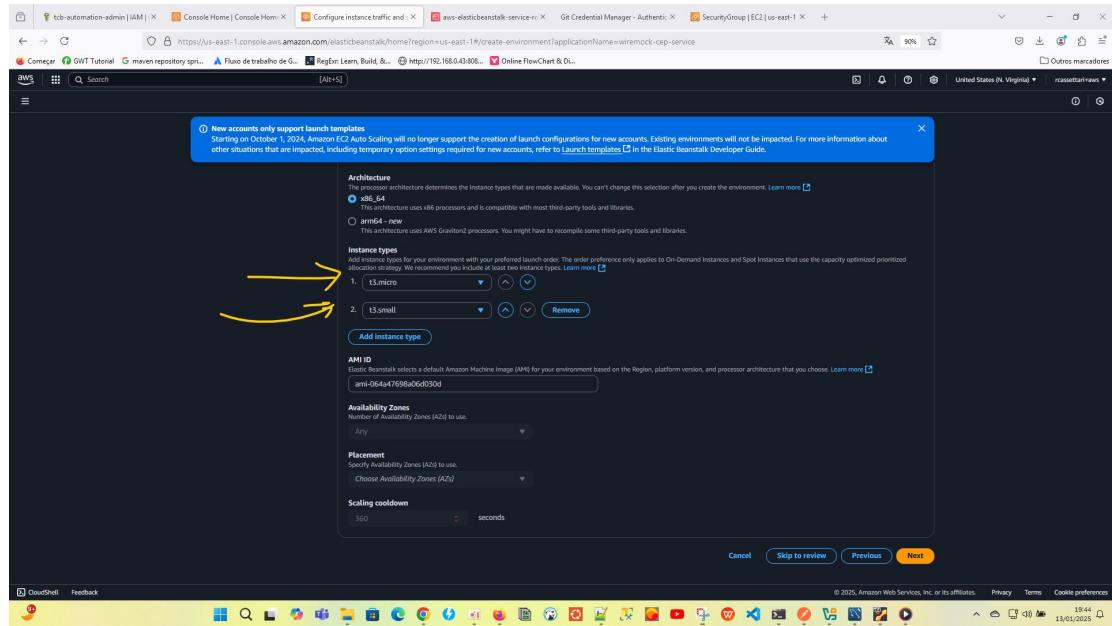
29 - Mantenho selecionado “On-Demand-instance” nesse radio-button ( se por spot, ele pode requisitar a instance):



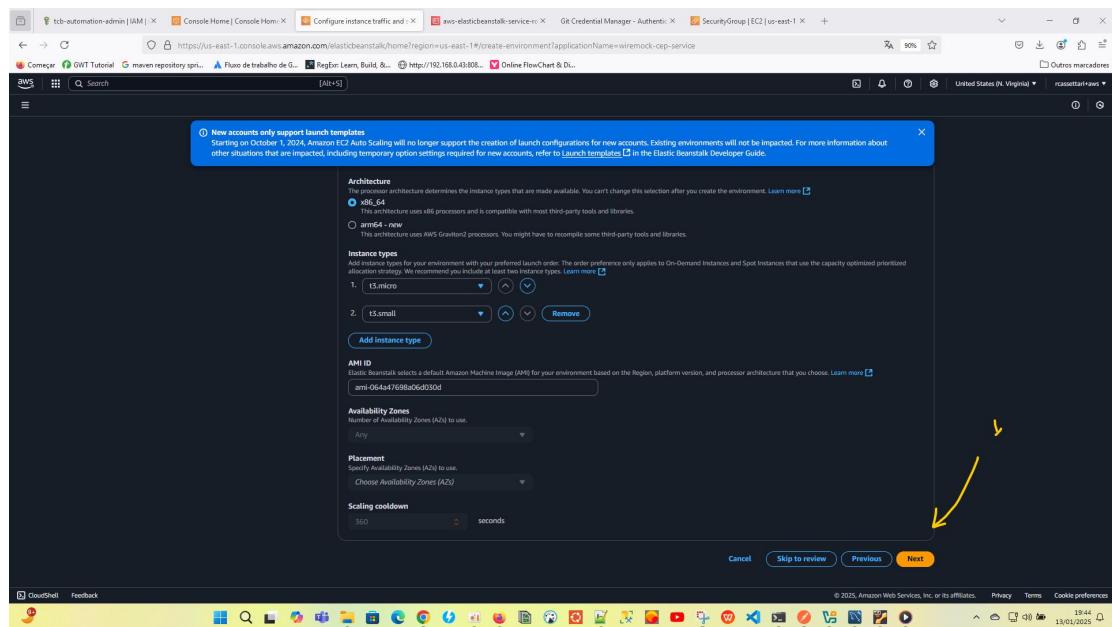
30 - Mantenho arquitetura “x86\_64”:



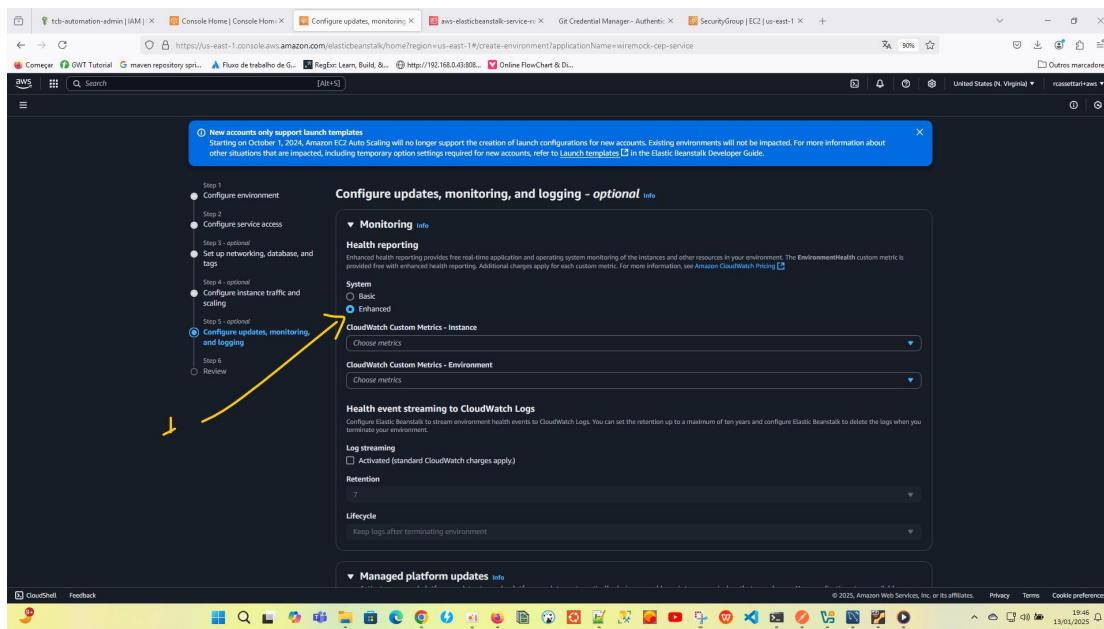
31 - Mantendo os instance types padrão que vieram “t3.micro” e “t3.small” na segunda opção ( a ordem de prioridades ):



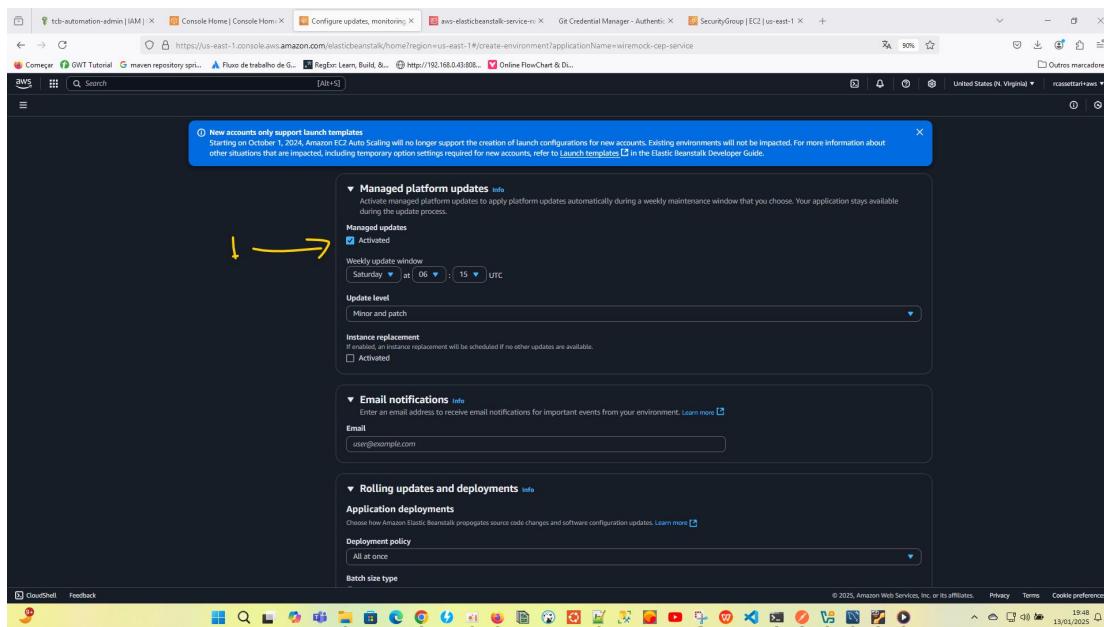
32 - Clico no botão “Next”:



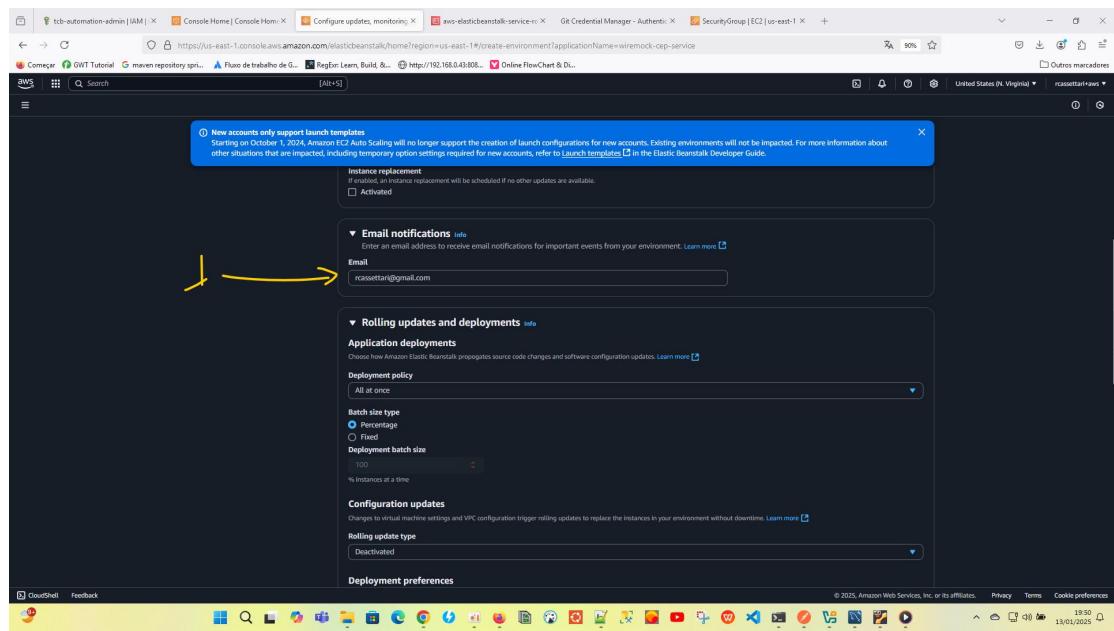
### 33 - Mantendo monitoramento “Enhanced”:



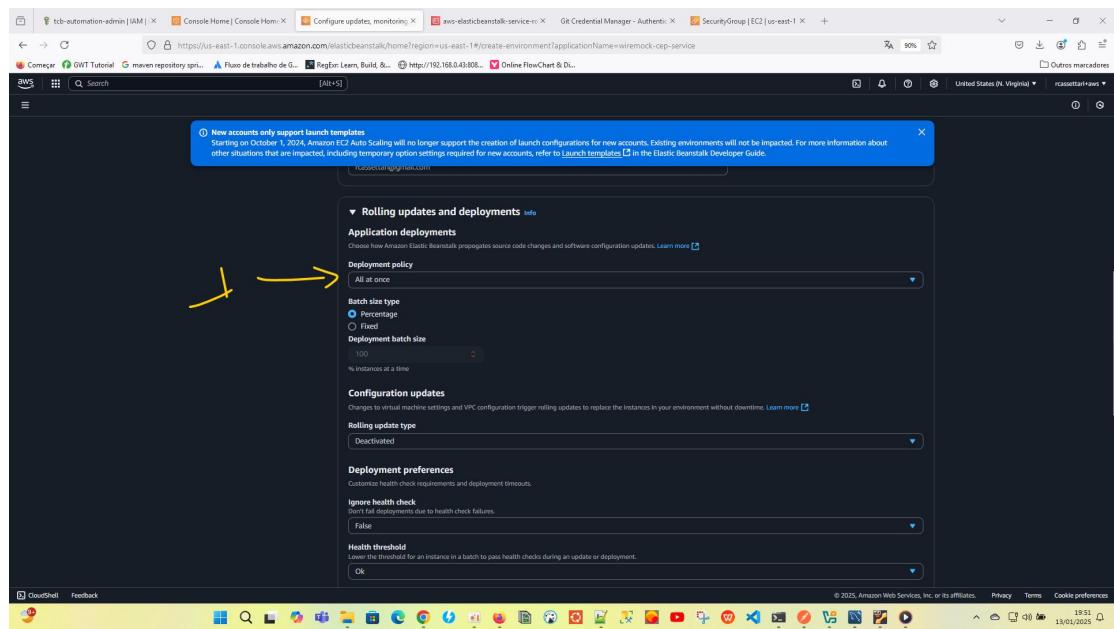
### 34 - Mantendo “Managed Updates” ativo, o horário de sábado, para um ambiente de testes, está bom:



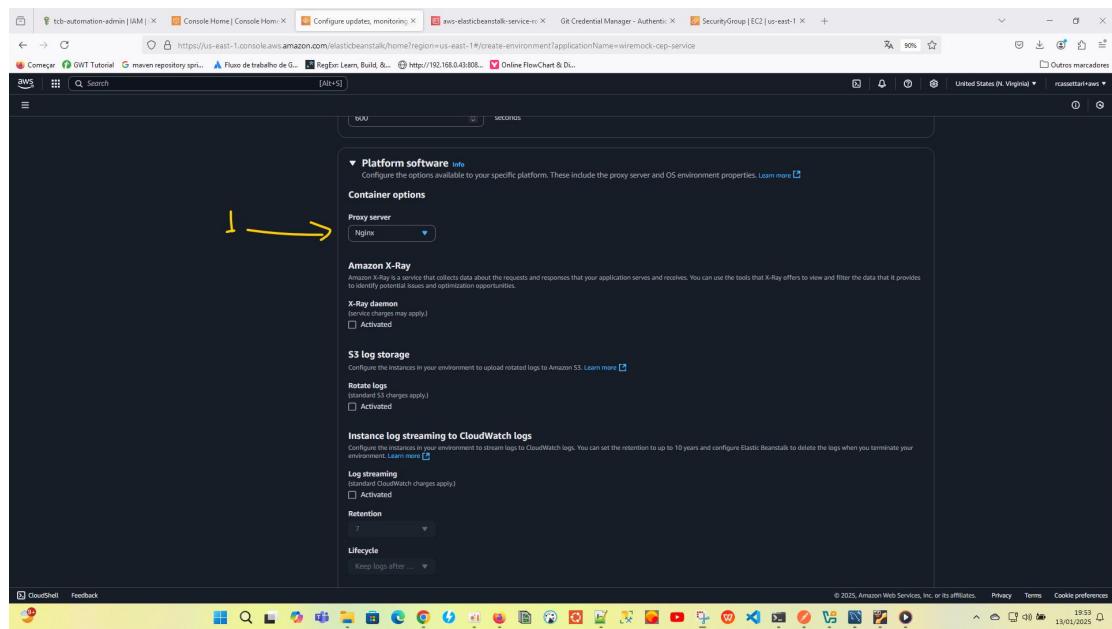
### 35 - Defino um e-mail para receber as notificações do ambiente:



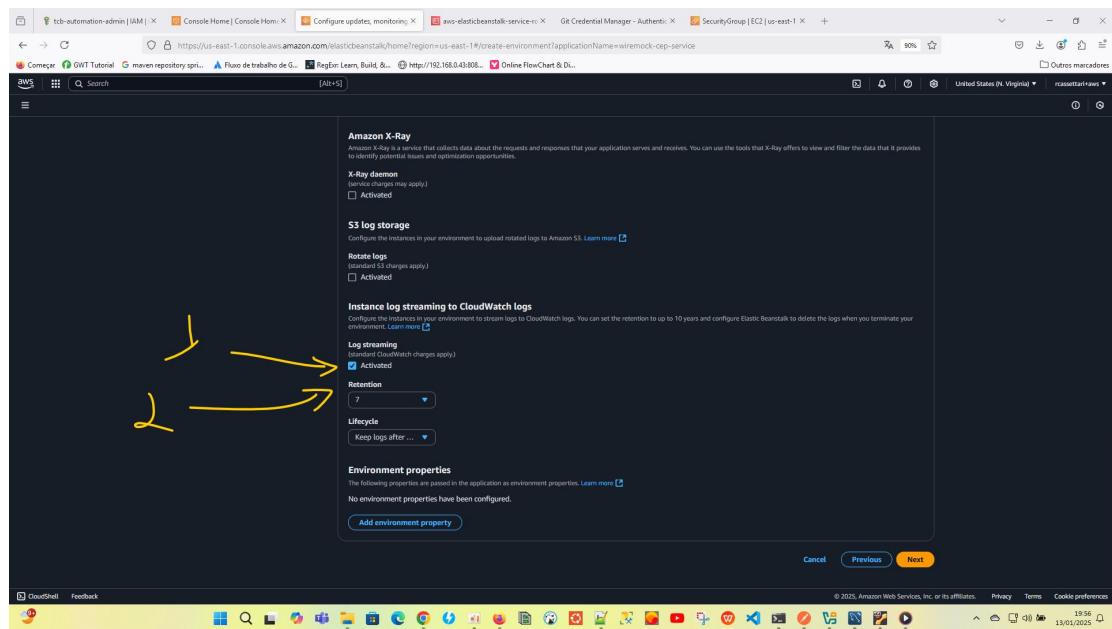
36 - Mantendo o deployment policy em "All at once", só vou ter 1 instance mesmo (se fosse ambiente de negócios sério, teria mais instances e não poderia manter isso assim):



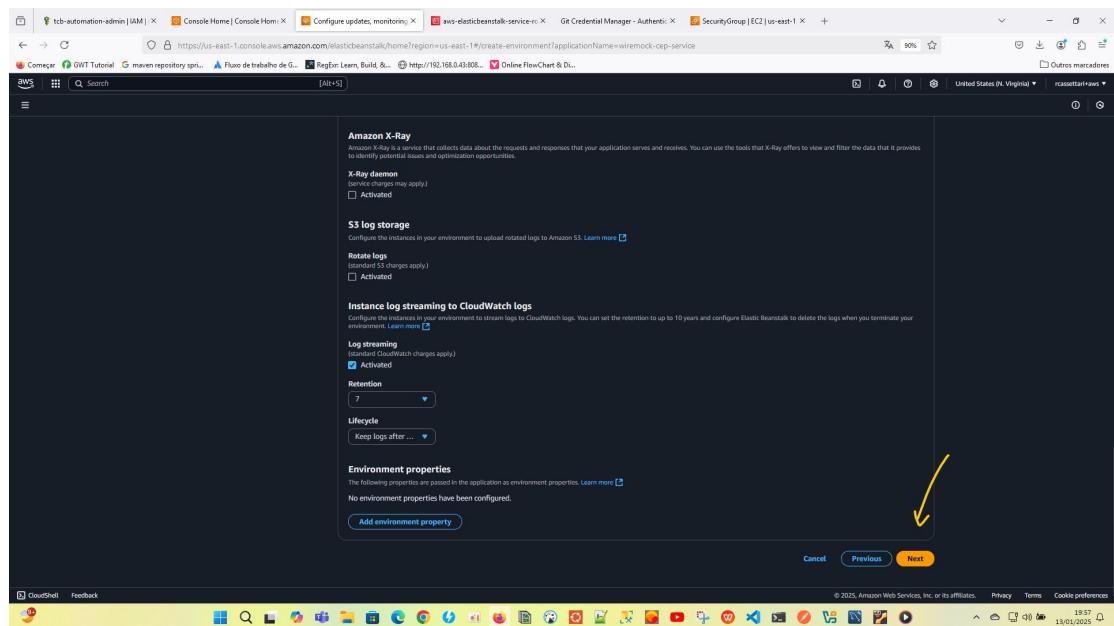
### 37 - Mantendo “Nginx” proxy selecionado na lista:



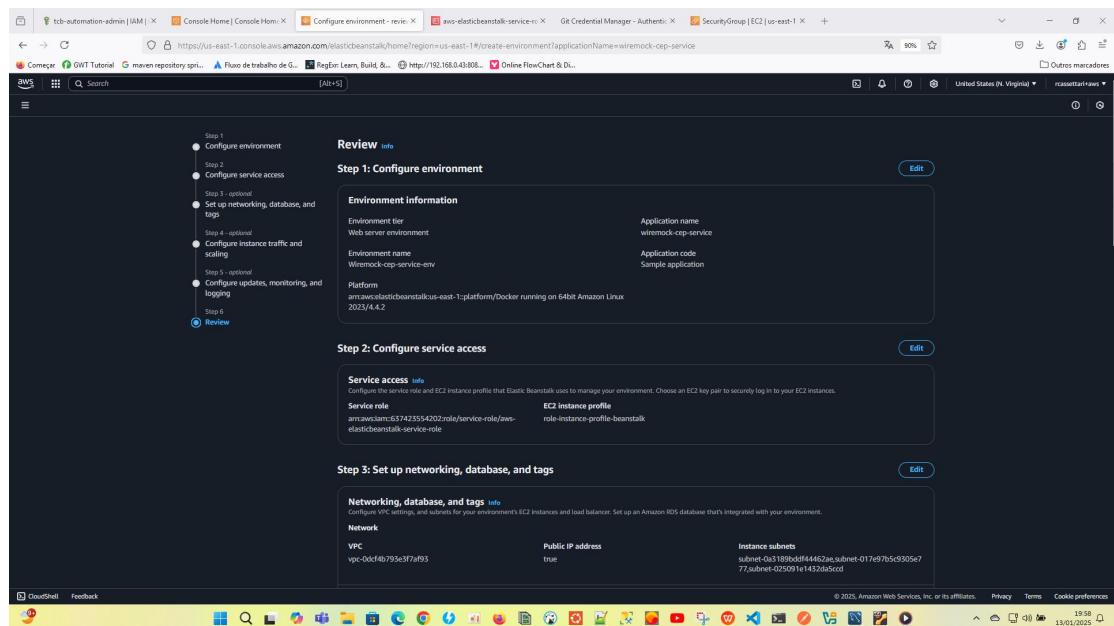
### 38 - Ativo o “Log streaming” para o cloudwatch e selecione o período para 7 dias:



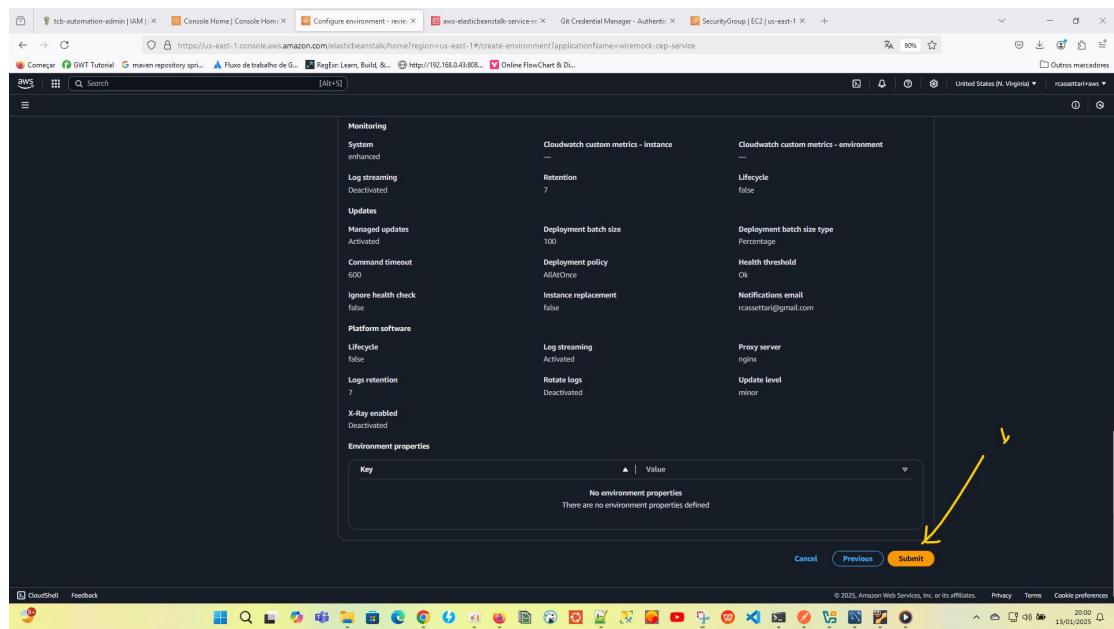
### 39 - Clico em “Next”:



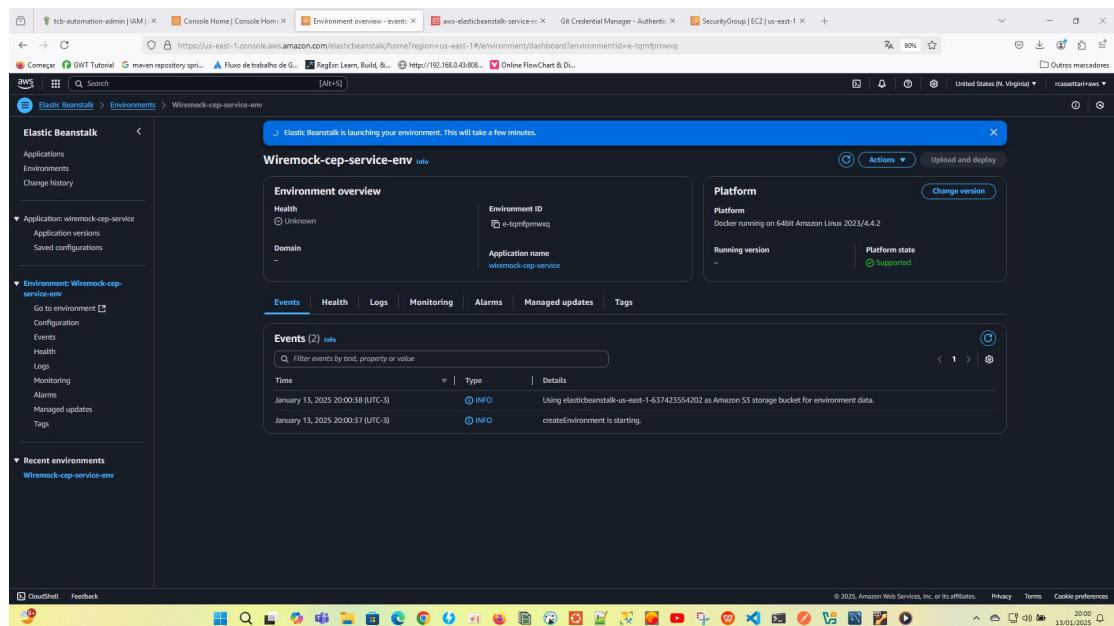
### 40 - Aparece a tela inicial do review:



41 - Faço scroll até o fim da tela do review e clico em “Submit”:



42 - Aparece a mensagem que o novo environment está sendo iniciado:



43 - Pelo EC2 instances, vejo que ele subiu uma instance para meu environment novo (está Running):

The screenshot shows the AWS CloudWatch Metrics Insights interface. A single log entry is displayed:

```
2025-01-13T03:47:00Z INFO Environment health has transitioned from Pending to Ok. Initialization completed 8 seconds ago and took 5 minutes.
```

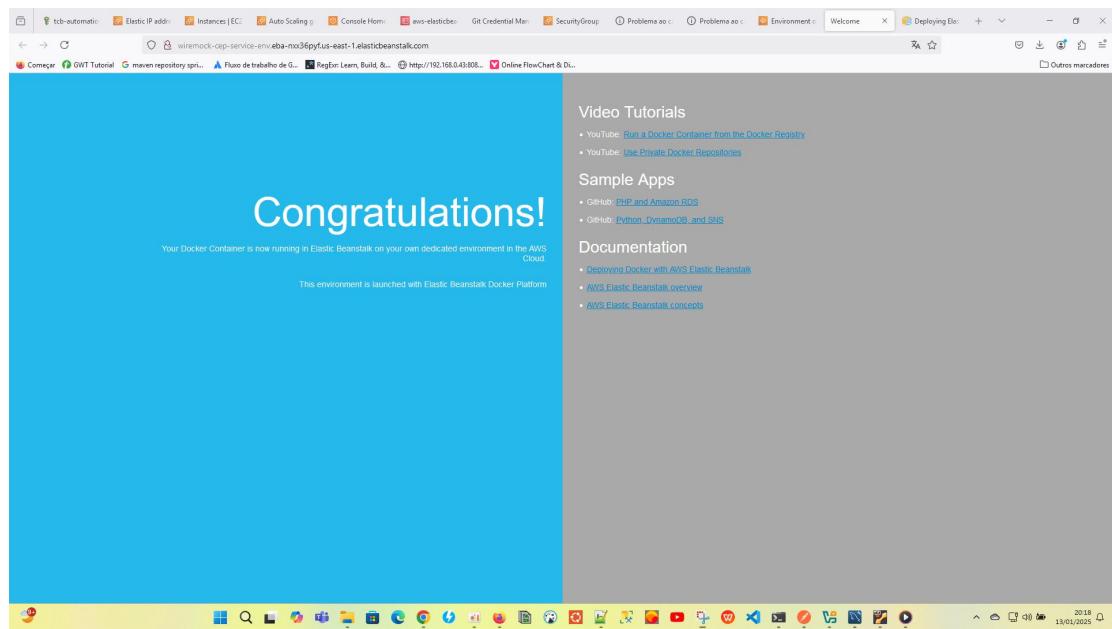
44 - Volto no meu environment e clico em “Domain”:

The screenshot shows the AWS Elastic Beanstalk Environment Overview page for the 'Wiremock-cep-service-env' environment. The 'Domain' field is highlighted with a yellow arrow:

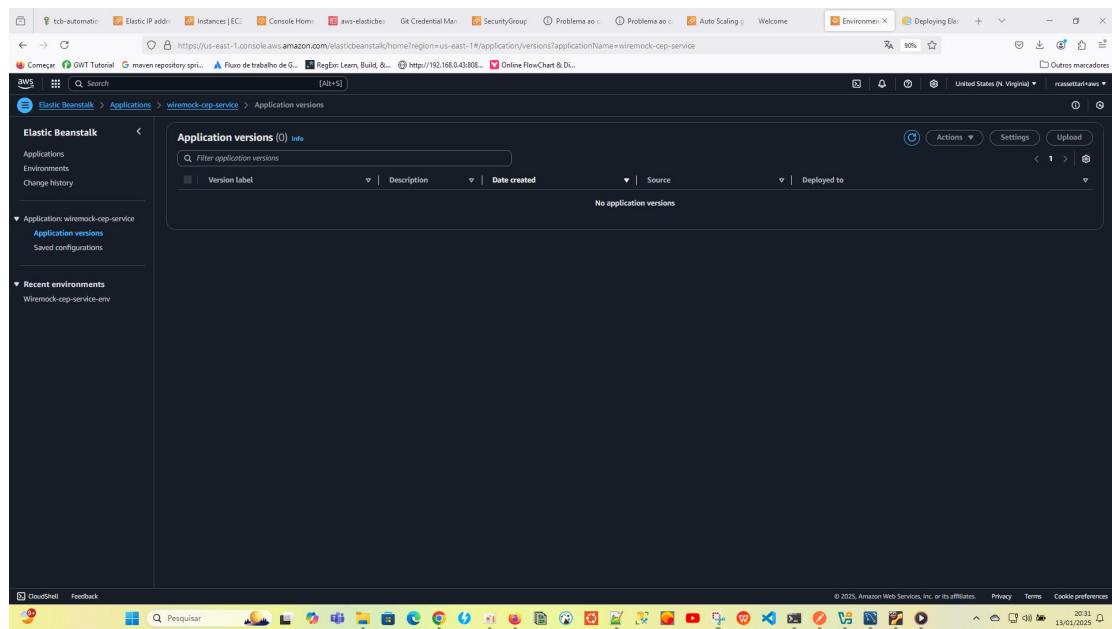
Wiremock-cep-service-env

Domain: Wiremock-cep-service-env.eba-inx3lipyf.us-east-1.elasticbeanstalk.com

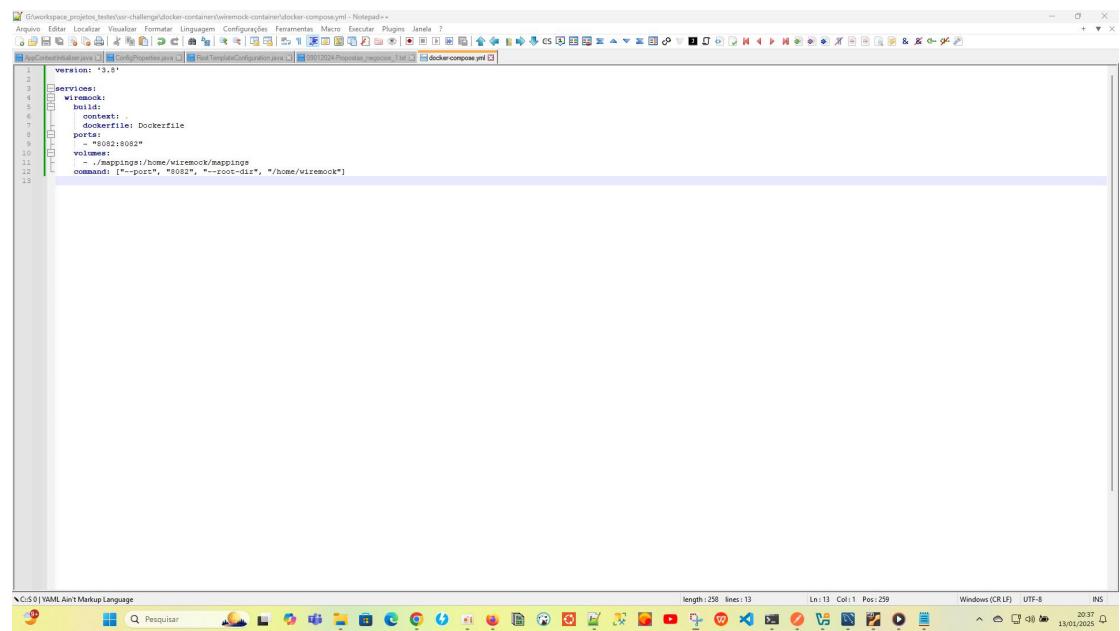
45 - Aparece o “sample application” que eu tinha selecionado para executar:



46 - Vou no meu application e vejo que ainda não tenho application versions:

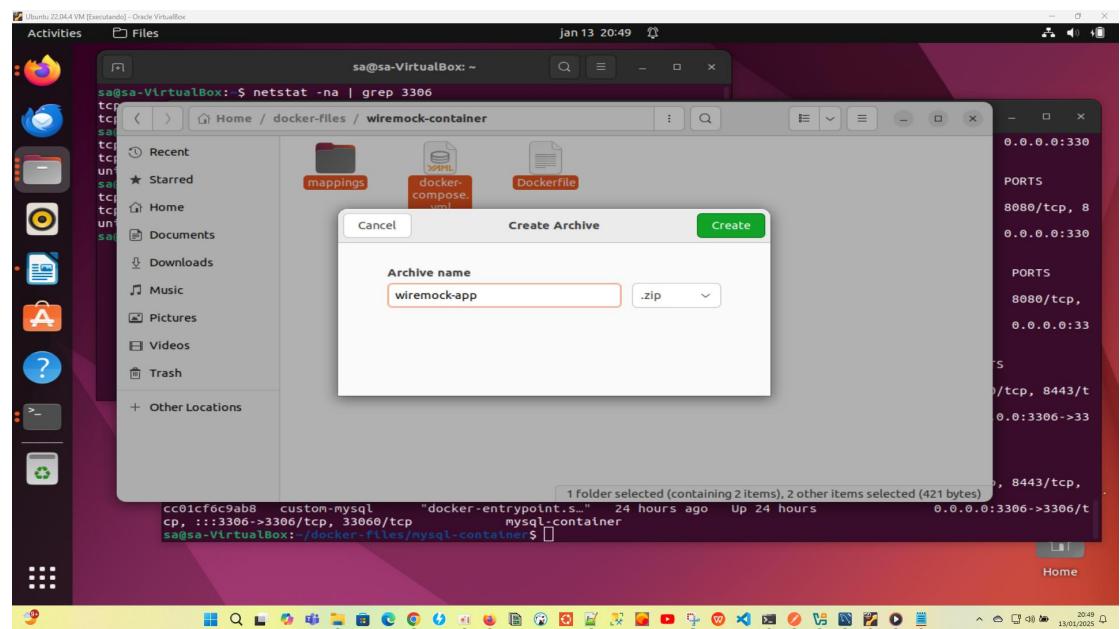


47 - Crio um docker-compose.yml para subir o wiremock sozinho (podia ser mais coisas):

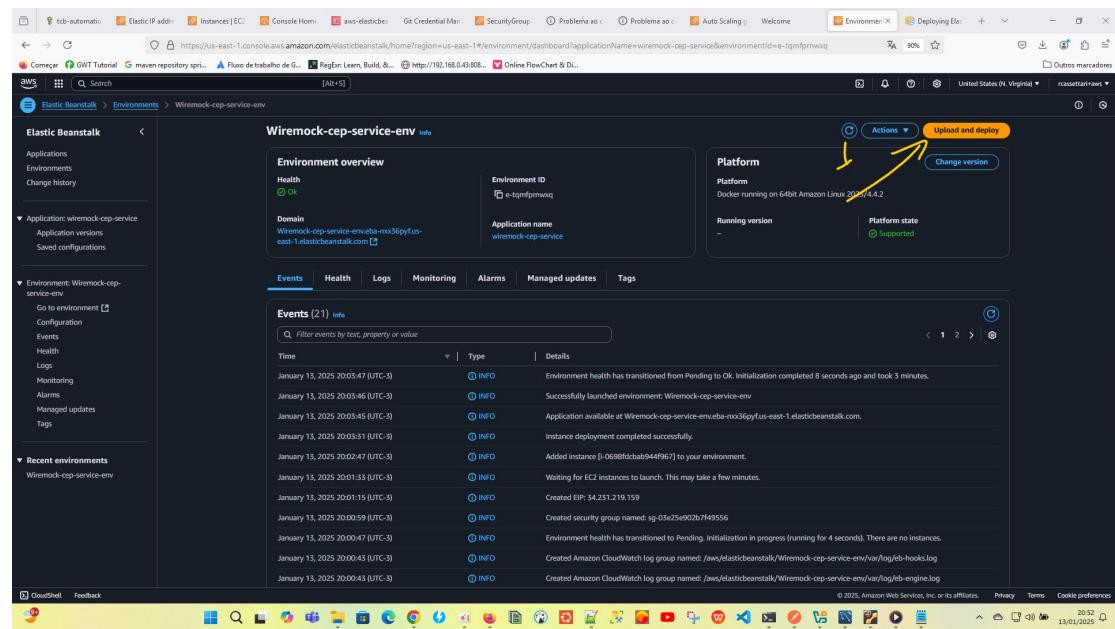


```
version: '3.8'
services:
  wiremock:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "8082:8082"
    volumes:
      - ./mappings:/home/wiremock/mappings
    command: ["--port", "8082", "--root-dir", "/home/wiremock"]
```

48 - Coloco o diretório de mappings, Dockerfile e docker-compose.yml na raiz compacto tudo para um arquivo .zio que chamei de “wiremock-app.zip”:

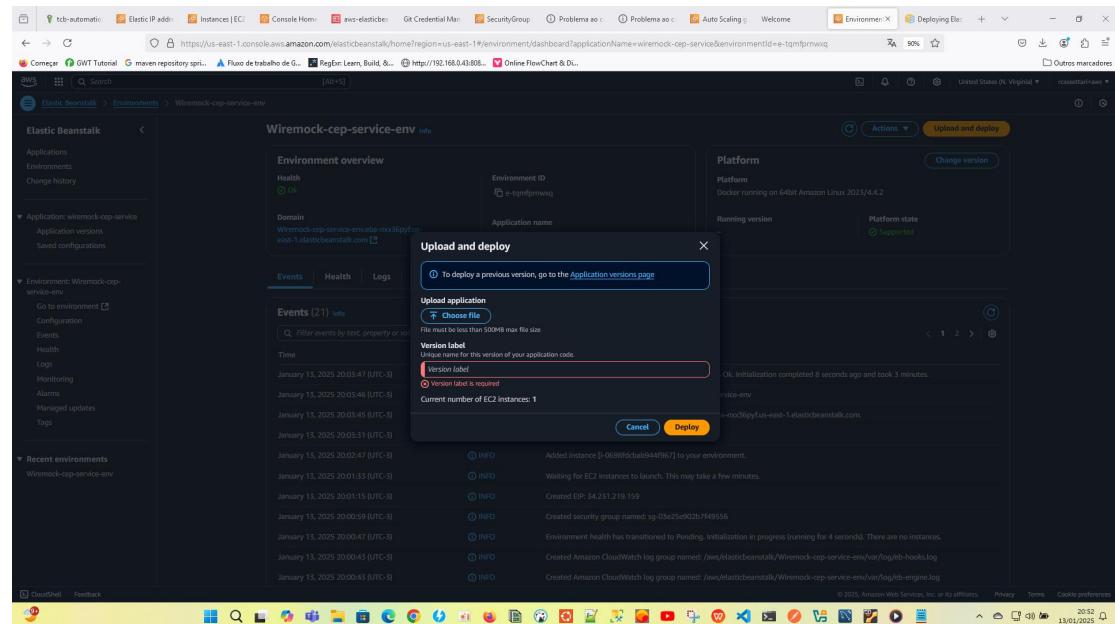


49 - No meu ambiente, clico em “Upload and deploy”:



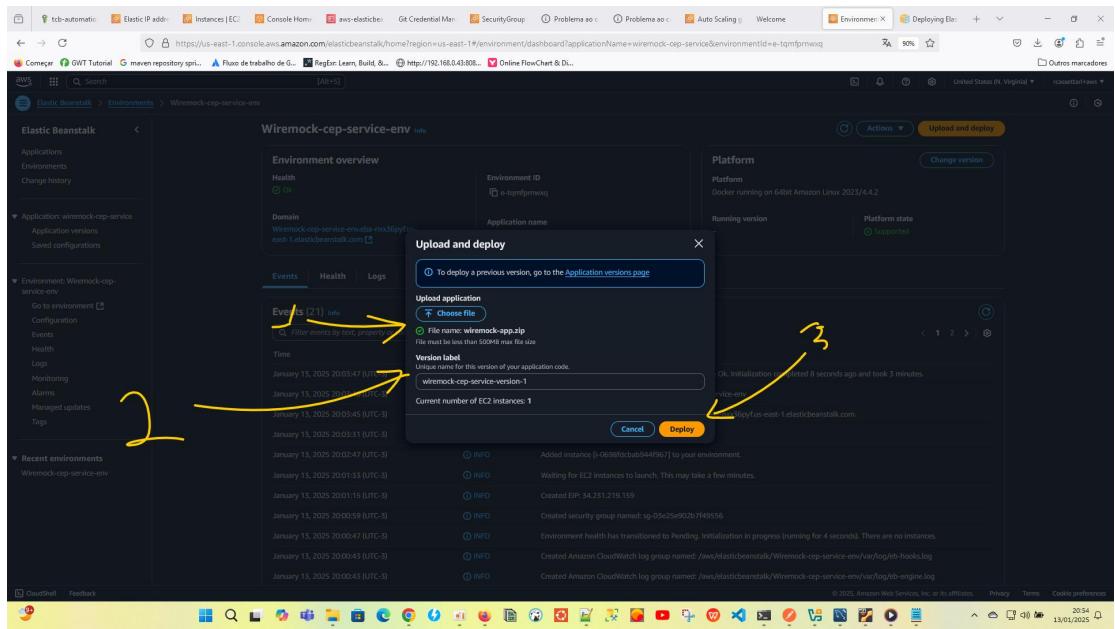
The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a navigation sidebar with options like Applications, Environments, and Configuration. The main area is titled 'Wiremock-cep-service-env' and shows an 'Environment overview' with a green 'Health' status. It includes sections for Platform (running on 64bit Amazon Linux 2023/4.4.2), Events (with 21 items listed), and Logs. At the top right, there are three buttons: 'Actions', 'Upload and deploy' (which has a yellow arrow pointing to it), and 'Change version'.

50 - Aparecerá a tela para carregar o arquivo e definir a versão:

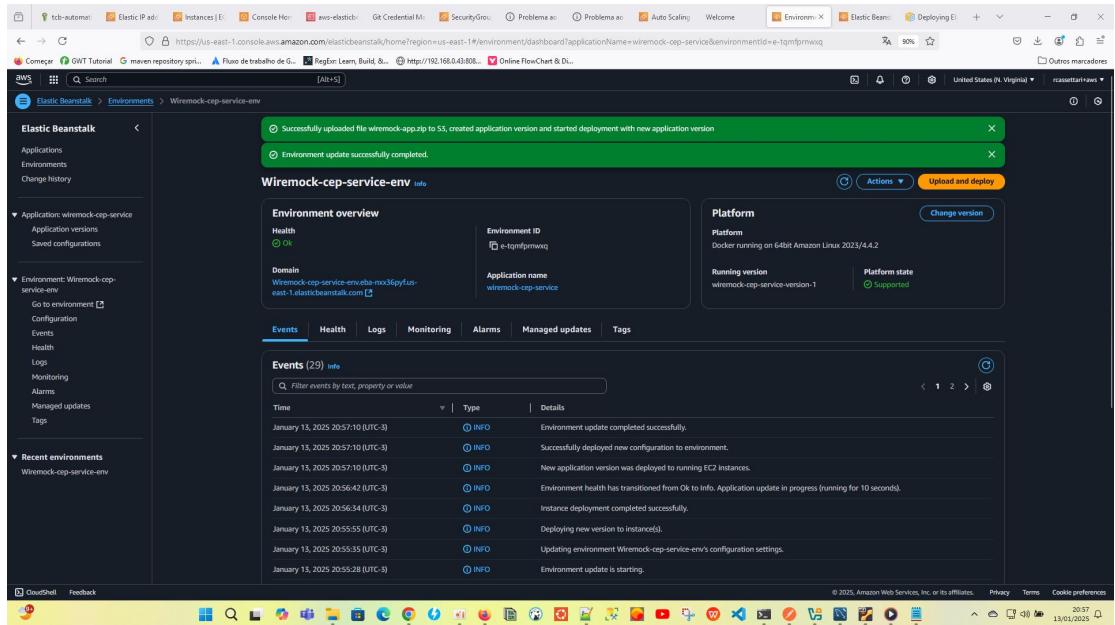


This screenshot shows the 'Upload and deploy' dialog box. It has a message at the top: 'To deploy a previous version, go to the Application versions page'. Below that is a 'Choose file' button with a note: 'File must be less than 500MB max file size'. There's also a 'Version label' field with a red border and the error message 'Version label is required'. At the bottom, there's a 'Deploy' button and a 'Cancel' button.

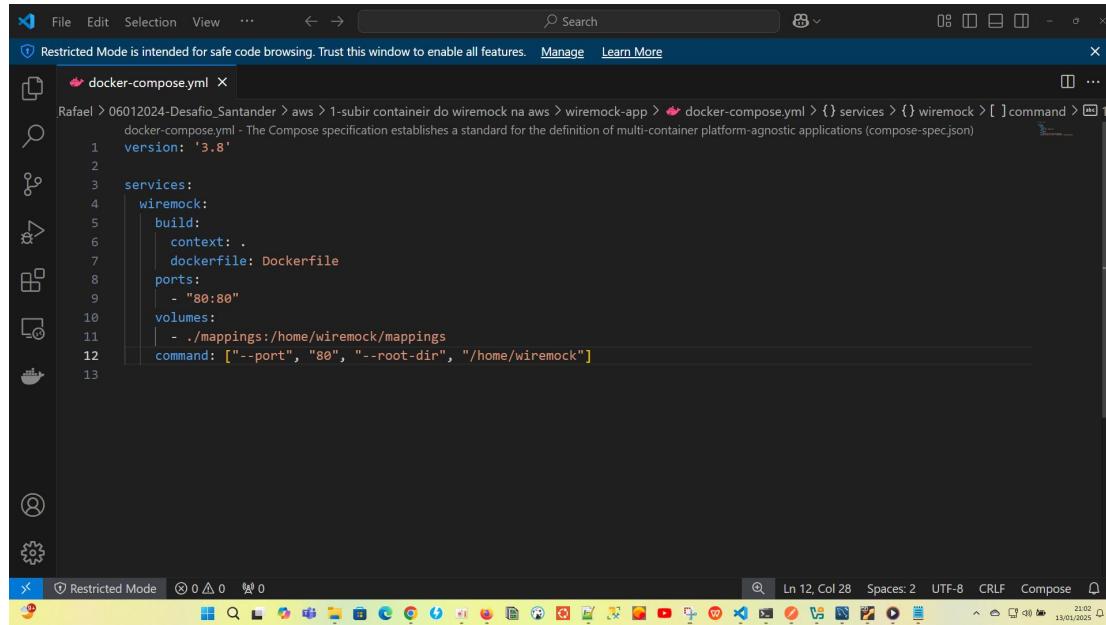
51 - Escolho meu arquivo .zip, defino label para a versão e clico em “deploy”:



52 - Vejo a mensagem que o deploy foi executado:



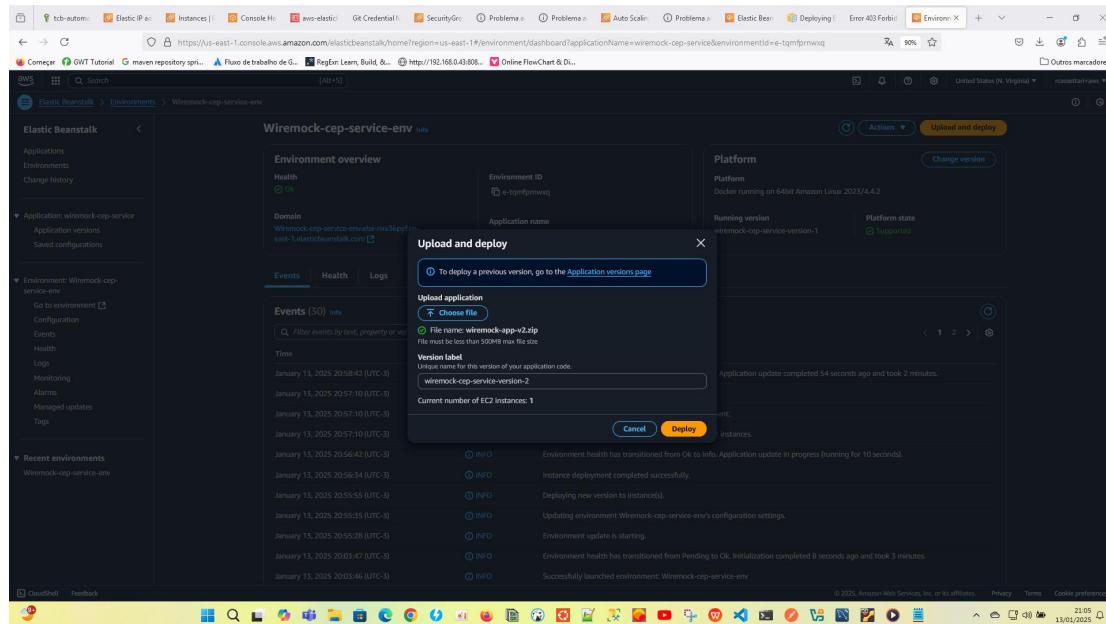
53 - Mudo tudo para porta 80, tanto no docker-compose.yml, como no dockerfile ( eu creio que tinha funcionado, mas eu me esqueci e tava testando no url errado, via método GET e era POST - coisas do sono...) :



```
version: '3'

services:
  wiremock:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "80:80"
    volumes:
      - ./mappings:/home/wiremock/mappings
    command: ["-port", "80", "-root-dir", "/home/wiremock"]
```

54 - Faço novo upload and deploy, escolhendo novo arquivo v2, com label também com versão diferente:



55 - Vejo a mensagem do deploy v2 carregado para o s3 bucket:

The screenshot shows the AWS Elastic Beanstalk console with the URL <https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/dashboard?applicationName=wiremock-cep-service&environmentId=e-tqmfpfmwq>. The main pane displays the environment overview for 'Wiremock-cep-service-env'. A prominent green banner at the top states: "Successfully uploaded file wiremock-app-v2.zip to S3, created application version and started deployment with new application version". Below this, the 'Events' tab is selected, showing a list of 30 events. The most recent event is dated January 13, 2025, at 20:58:42 UTC-3, with the status 'INFO' and the message: "Environment health has transitioned from Info to Ok. Application update completed 54 seconds ago and took 2 minutes." Other events show the deployment process starting and progressing.

56 - Clico em “Request logs”:

The screenshot shows the same AWS Elastic Beanstalk console as above, but with the 'Logs' tab selected in the navigation bar. A yellow arrow points to the 'Request logs' button in the top right corner of the logs panel. The logs panel displays a table with columns for Log file, Time, EC2 instance, and Type. One log entry is visible: 'Download' at 'January 13, 2025 21:09:35:304 (UTC-3)' with EC2 instance 'i-069ffdcab944f967' and type 'tail'.

57 - No log aberto, vejo que havia começado a aparecer os logs típicos do wiremock, indicando quando um request não bate com os arquivos de mappings:

```
Conectar | GWT Tutorial | Maven repository spm... | Fluxo de trabalho do G... | RegEx Learn, Build, ... | http://192.168.0.43:8080/ | Online FlowChart & D... | Outros m... https://elasticbeanstalk-us-east-1-637423554202.s3.amazonaws.com/resources/environments/logs/tail-tel-tqmpfpmwql-0696fbdb944f967/fail-logs/173611337529.txt?X-Amz-Security-Token=IQoJ... elasticsearch +
```

wiremock-1 disable-banners: false  
wiremock-1 disable-journal: false  
wiremock-1 verbose: false  
wiremock-1 extensions:  
wiremock-1 response-template\_webhook  
2025-01-14 00:00:55.796  
| Closer stub | Request |  
| POST [path]/api/v1/lookup/cap | GET /  
| {?(@.streetType == 'Rua' & @.street == 'São Geraldo' &  
| .city == 'São Bernardo do Campo' & @.state == 'SP')} |  
Request was not matched
2025-01-14 00:00:24.625
Closer stub
POST [path]/api/v1/lookup/cap
{?(@.streetType == 'Rua' & @.street == 'São Geraldo' &
.city == 'São Bernardo do Campo' & @.state == 'SP')}
Request was not matched
-----
2025-01-14 00:09:24.643
Closer stub
POST [path]/api/v1/lookup/cap
{?(@.streetType == 'Rua' & @.street == 'São Geraldo' &
.city == 'São Bernardo do Campo' & @.state == 'SP')}
Request was not matched
-----
/var/log/docker-events.log  
2025-01-14T09:55:50.728Z container exec\_start: /bin/sh -c curl -f http://localhost:8080/\_admin/health || exit 1 ee0bdc40d7672b4cdebf9b526c132ab4ecbc1e72b304e4d15c139425986 (com.docker.compose.config  
hash=00000000000000000000000000000000) com.docker.compose.number=1, com.docker.compose.image=hash=25cfe1edff0dadd29972e14ef244e6fb76125e083955, com.docker.compose.oneoff=false  
com.docker.compose.project=currently, com.docker.compose.project.config\_file=/var/app/current/docker-compose.yml, com.docker.compose.project.working\_dir=/var/app/current, com.docker.compose.servic... com.docker.compose.version=2.32.1, 21:11 13/01/2025

58 - Colocando o endereço do domain do application no postman, com o método http correto (POST), o wiremock responde a partir da AWS:

The screenshot shows the Postman interface with a collection named 'Testes Desafio Santander'. A POST request is selected, targeting the endpoint `http://wiremock-cep-service-env.etsa-mxx36pyf.us-east-1.elasticbeanstalk.com/api/v1/lookup/cep`. The request body is a JSON object with a single key 'cep':

```
1 {  
2   "cep": "099820220"  
3 }
```

The response body shows the value '099820220'. A yellow arrow points from the 'cep' key in the request body to the response body.

59 - Faço uma segunda consulta no wiremock service deployado na AWS:

The screenshot shows the Postman interface with a collection named "Testes Desafio Santander". A specific API endpoint, "POST Query Cep by Address - Container AWS", is selected. The request URL is "http://wiremock-cep-service-env.eba-nrx36pyf.us-east-1.elasticbeanstalk.com/api/v1/lookup/cep". The request body contains the following JSON:

```
1 "streetType": "Avenida",
2 "street": "Paulista",
3 "city": "Sao Paulo",
4 "state": "SP"
```

Yellow arrows highlight several parts of the interface: one arrow points from the "Body" tab to the JSON code; another points from the "Send" button to the response area; a third points from the response status bar to the JSON results; and a fourth points from the JSON results back to the "Body" tab.

60 - O arquivo “wiremock-app-v2.zip”, último que foi feito deploy, tinha tudo na raiz do .zip :

The screenshot shows a Windows File Explorer window displaying the contents of a zip file named "wiremock-app-v2.zip". The file structure is as follows:

- Nome
- Tipo
- Tamanho Compact...
- Protegido ...
- Tamanho
- Razão
- Data de modificação

Nome	Tipo	Tamanho Compact...	Protegido ...	Tamanho	Razão	Data de modificação
<b>mappings</b>	Pasta de arquivos					13/01/2025 21:01
<b>docker-compose.yml</b>	Arquivo Fonte Yaml	1 KB	Não	1 KB	39%	13/01/2025 21:01
<b>Dockerfile</b>	Arquivo	1 KB	Não	1 KB	31%	13/01/2025 21:02

61 - O beans talk tinha criado um security group a mais na porta 80 ( o default dele ), além do que eu criei na porta 8082 (como eu tinha testado inicialmente no método errado e sem url correto, não vi funcionar):

The screenshot shows the AWS CloudWatch Metrics interface. A metric named "trb-autos" is displayed with a value of 1. A yellow arrow points from the text "funcionar)" to the value "1".

Pronto ! Coloquei um application do beanstalk para rodar um environment na AWS, com um container contendo um dos serviços do desafio, que é o microserviço mockado que simula a aplicação de consulta de cep por endereço nos correios.