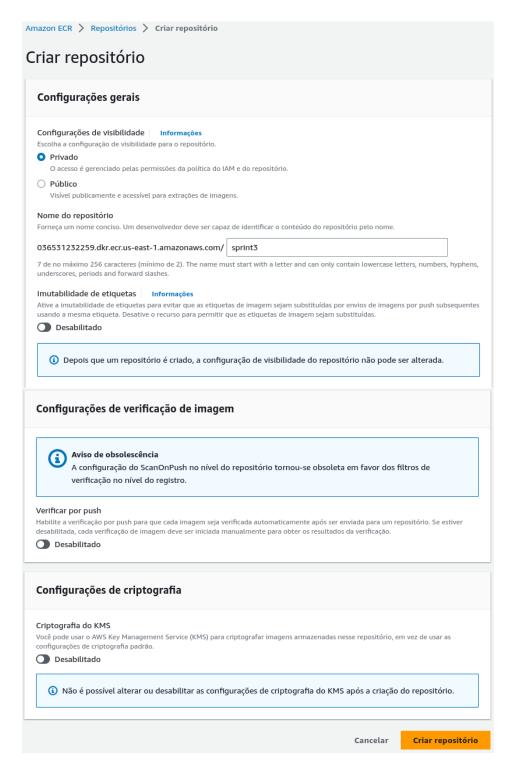
Configurando um *cluster* de instâncias EC2 que irão disponibilizar a API a partir de uma imagem Docker.

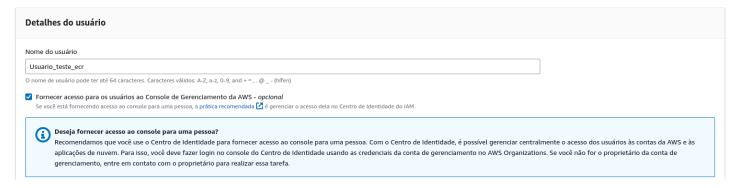
ECR (*Amazon Elastic Container Registry*) - A plataforma da AWS disponibiliza um serviço de repositório próprio para armazenamento de imagens, onde é possível hospedar a imagem Docker de uma aplicação.

• O primeiro passo é **criar o repositório**, escolhendo entre as opções privado ou público, e então nomeá-lo:

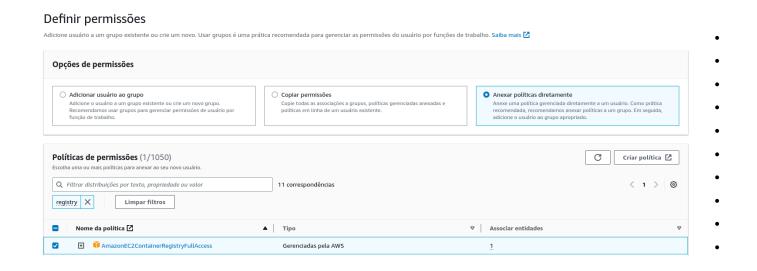


Para ter o acesso necesserário ao serviço do ECR, é preciso garantir que seu usário IAM esteja adequadamente configurado. Acesse o serviço IAM (Identity and Access Managent) e clique no menu Usuário e a seguir em Adicionar novo usuário; defina o nome do novo usuário e certifiquese de marcar a opção de "fonecer acesso para usuários ao console da AWS", como na imagem abaixo:

Especificar detalhes do usuário



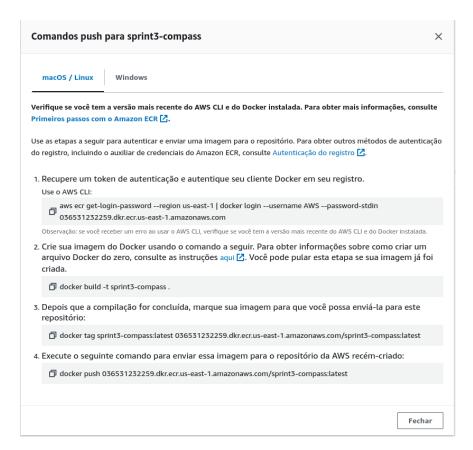
 A seguir, nas difinições de permissão, selecione "anexar políticas diretamente", busque e marque a opção "AmazonEC2ContainerRegistryFullAccess". Isto irá garantir que você tenha as permissões necessárias para a interação com o ECR:



Para configurar o acesso pelo terminal de sua máquina, proceda como mostrado neste <u>video</u>. As chaves de acesso garantem a segurança de sua conta! Tenha cuidado ao compartilhá-la, pois quem tiver suas chaves <u>terá livre acesso aos serviços AWS da sua conta</u>.

Observação: Caso já tenho um usuário IAM previamente registrado, certifique-se de que ele possui habilitada a política de acesso "AmazonEC2ContainerRegistryFullAccess".

 Voltando ao ECR, para hospedar a imagem no repositório, é necessário executar comandos push no terminal local. Após criado repositório, ao clicar em "comandos push para" são mostrados os comandos que devem ser executados localmente para que a imagem seja enviada para o repositório no ECR:



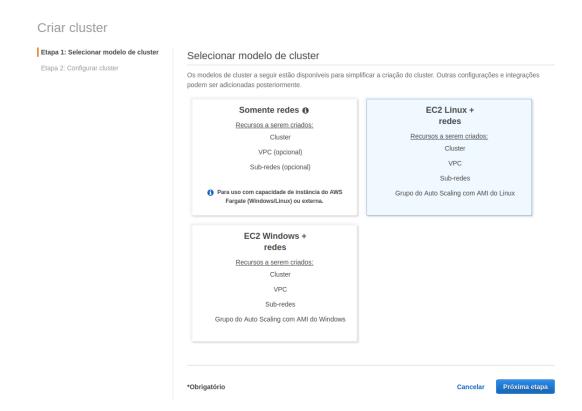
OBS: Para ações futuras é possível criar executáveis .bat ou .sh, para em caso de atualização dos códigos, a imagem no repositório atualiza automaticamente.

- Copiando os comando de push de seu repositório para um arquivo, é possível definir um script que irá facilitar o processo de envio e atualização da imagem no ECR:
 - No Windows, copie os comandos para o arquivo script.bat, na mesma pasta de seu Dockefile, e dê as permissões necessários para que seja executado.
 - No Linux, copie os comandos para um arquivo script.sh e execute: chmod +x script.sh

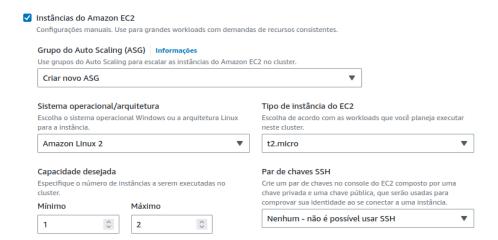
Executando o script, o resultado no terminal será:

```
| Date |
```

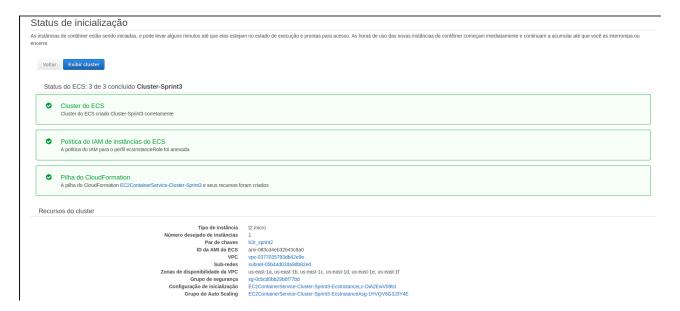
• Com a imagem hospedada no ECR, agora podemos criar o cluster usando o serviço ECS. No ECS, clique em *cluster*, e depois em *adicionar cluster*. Dê um nome ao cluster, selecione a VPC, subnets. Em seguida, selecione o modelo (tipo) de cluster:



• Selecione o tipo de imagem do S.O., o tipo de instância, e informe quantas máquinas o *cluster* deverá ter. Pela imagem, o cluster deverá ter no mínimo 1 e no máximo 2 instâncias, e o ECS irá gerenciar de forma a garantir que a quantidade desejada de instâncias estejam rodando:



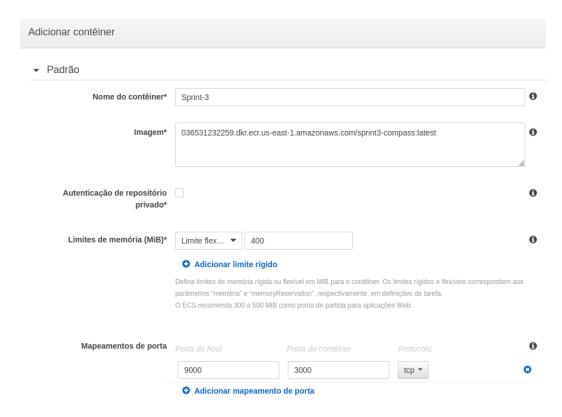
Cluster ECS inicializado:



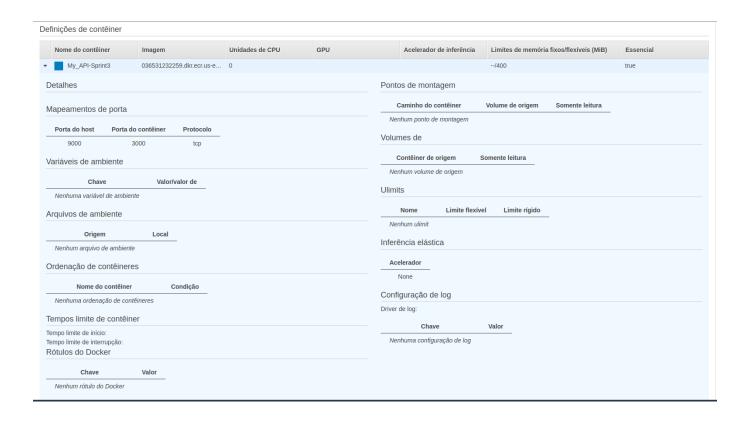
Criado o cluster, agora podemos criar a definição da tarefa (task definition), que irá derecionar o
ECS para qual cluster e imagem Docker utilizar. Na interface do ECS, clique em Definição de tarefas, depois em Criar nova tarefa e escolha o tipo de instância que seu cluster terá:



• Dê um nome e selecione a imagem Docker a ser utilizada na instância (copie o ID do repositório no ECR); indique também o mapeamento de portas (neste caso, o servido da API estará disponível na porta 9000, que foi associada à porta 3000 padrão de saída do NodeJS):

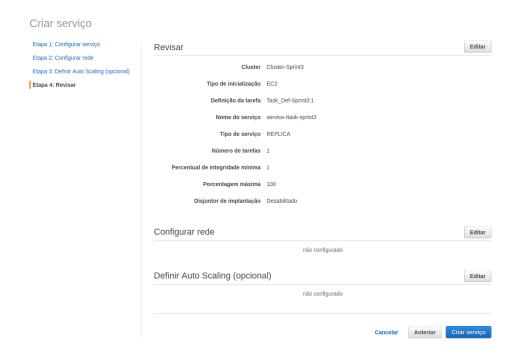


• Resumo da task:

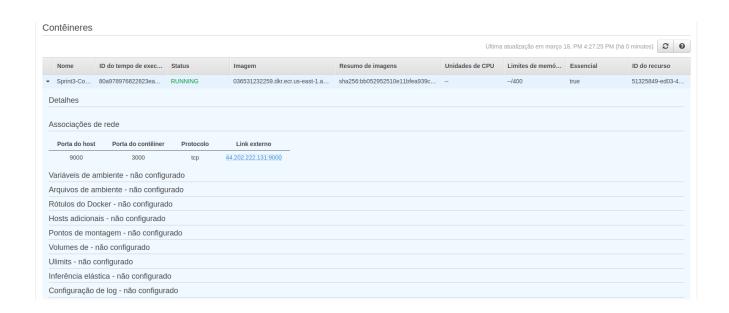


Agora falta configurar o Serviço (*Service*) que será o responsável por lançar a *Definição de tarefa* (*Task definition*) que, por sua vez, sabe qual cluster de instâncias ECS deve lançar e qual imagem Docker no ECR as instâncias irão rodar.

 Ainda na interface do ECS, selecione a Definição de tarefa que acabou de criar; em ações, selecione criar serviço:



• Resumo do service: cluster de 1 instância EC2 rodando a aplicação!



Adicionalmente, é possível configurar o **Load Balancer** (caso seu cluster tenha duas ou mais instâncias, redistribui o tráfego entre elas) e definir também configurações de **Auto Scaling** (a partir de determinada parâmetro de acesso, ou mesmo para data e hora definidas, o ECS pode aumentar a quantidade de instâncias fornecendo sua API – isto é muito útil para período de pico de acesso à sua aplicação instância).