STREAM SERIES

desvendando a ferramenta

NodeJS e ElasticSearch

Persistindo mensagens do console.log

SERGIO CABRAL

sergiocabral.com

Índice

Este conteúdo é gratuito. Foi preparado como esboço para uma aula do Experts Club, da Rocketseat.

Se tudo estiver certo, você consegue a versão atualizada desse *e-book* nos links:

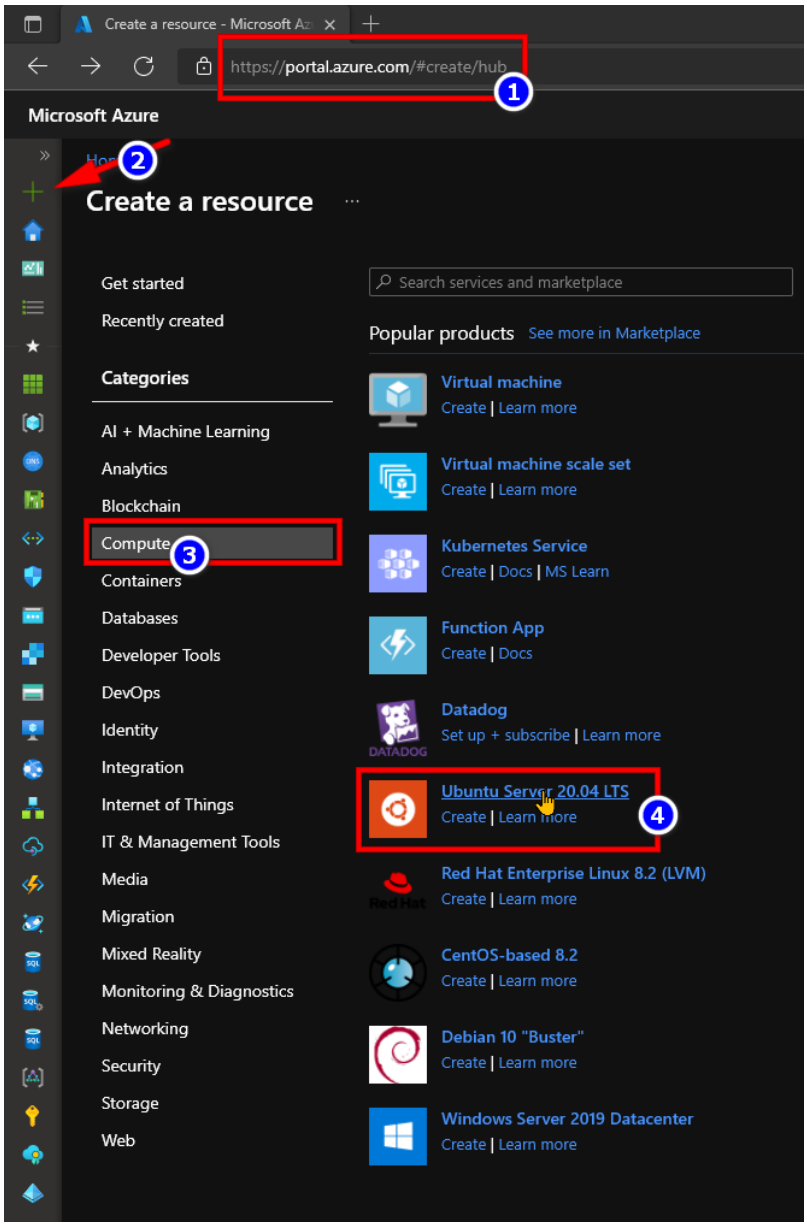
- ebook.sergiocabral.com/nodejs-logging-to-elasticsearch (versão on-line)
- ebook.sergiocabral.com/nodejs-logging-to-elasticsearch.pdf (versão para download em PDF)

1. ElasticSearch: Configurando ambiente

Será criada uma infraestrutura bem simples apenas para servir ao propósito da aula.

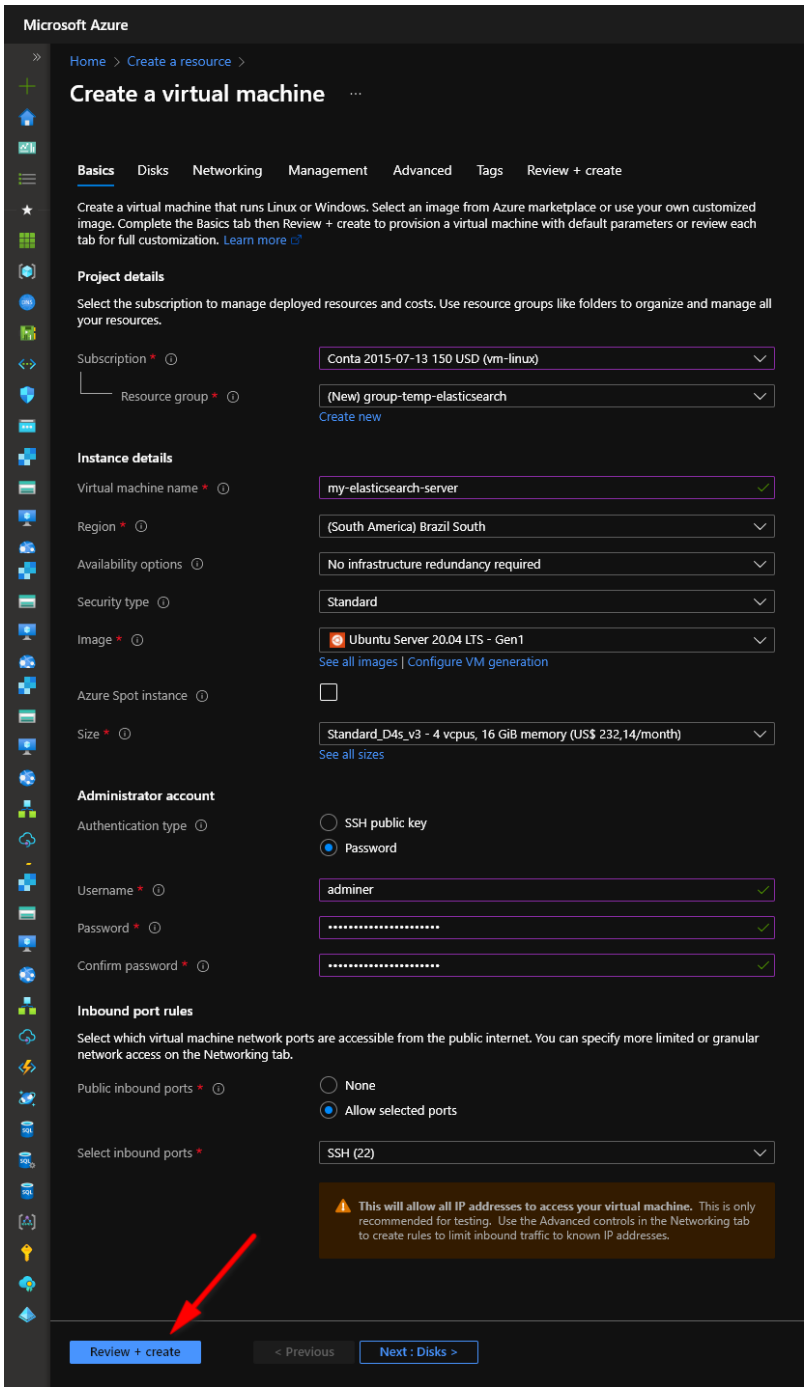
1.1. Criando o servidor

Vou começar criando uma Máquina Virtual (VM). Vou usar o Microsoft Azure como serviço de nuvem. Para mim é sempre mais fácil partir de uma imagem Linux Ubuntu.



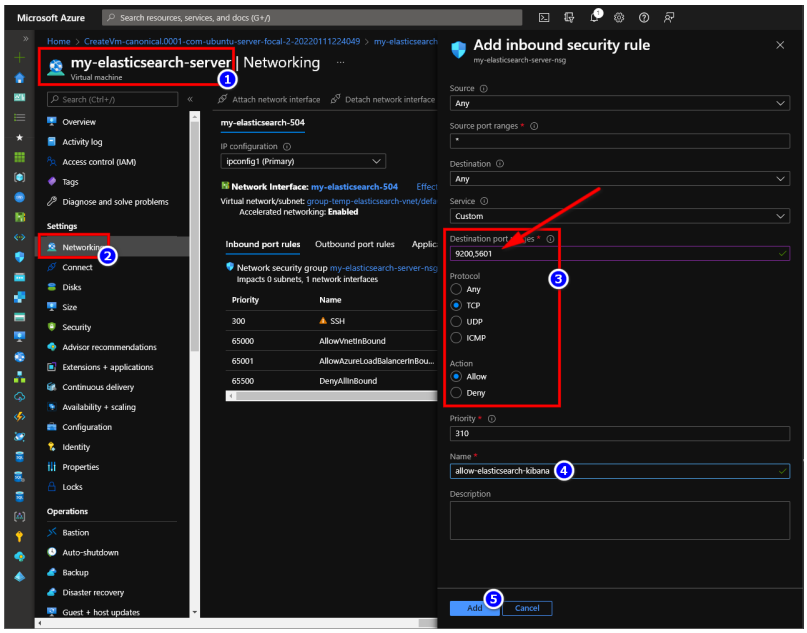
Como qualquer banco de dados, o ElasticSearch consome armazenamento e memória. A memória é especialmente necessária ao realizar consultas e seu consumo é maior conforme mais dados em disco são gravados.

Mas para o propósito dessa aula pouca coisa é necessária. O motivo de eu escolher uma VM com 16GB de memória e 4 CPUs é porque 1) ela será temporária e 2) ao instalar as aplicações as coisas andam mais rápido.



Então, simplesmente defini esses parâmetros e cliquei avançar até concluir.

Depois de criada a VM, precisamos acessá-la no painel do Azure para poder abrir as portas de rede (firewall) que vão permitir as conexões de entrada. As portas permitidas foram 9200 e 5601, que respectivamente são do Elasticsearch (o banco de dados) e Kibana (a interface web de visualização dos dados).



Como já disse, por ser uma VM temporária eu não fiz questão de levar em conta nenhuma boa prática de segurança ou seja lá o que for. Eu só quero um servidor Elasticsearch.

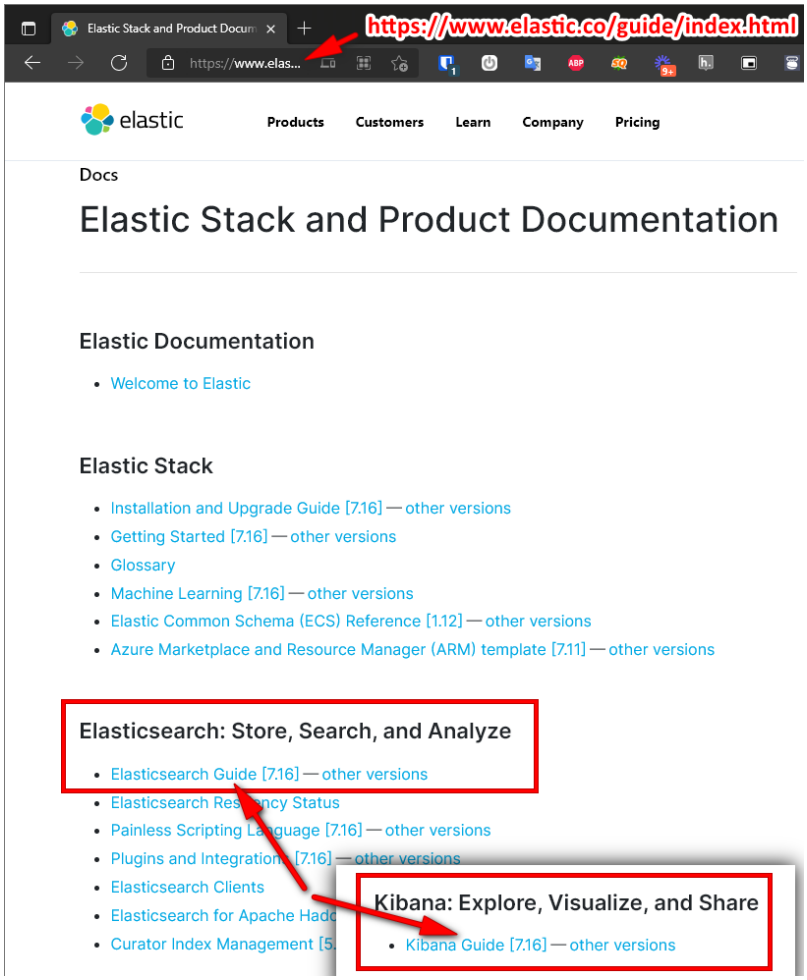
1.2. Configurando o servidor

Após a criação da VM executei os seguintes comandos para obter um ambiente do ElasticSearch e Kibana.

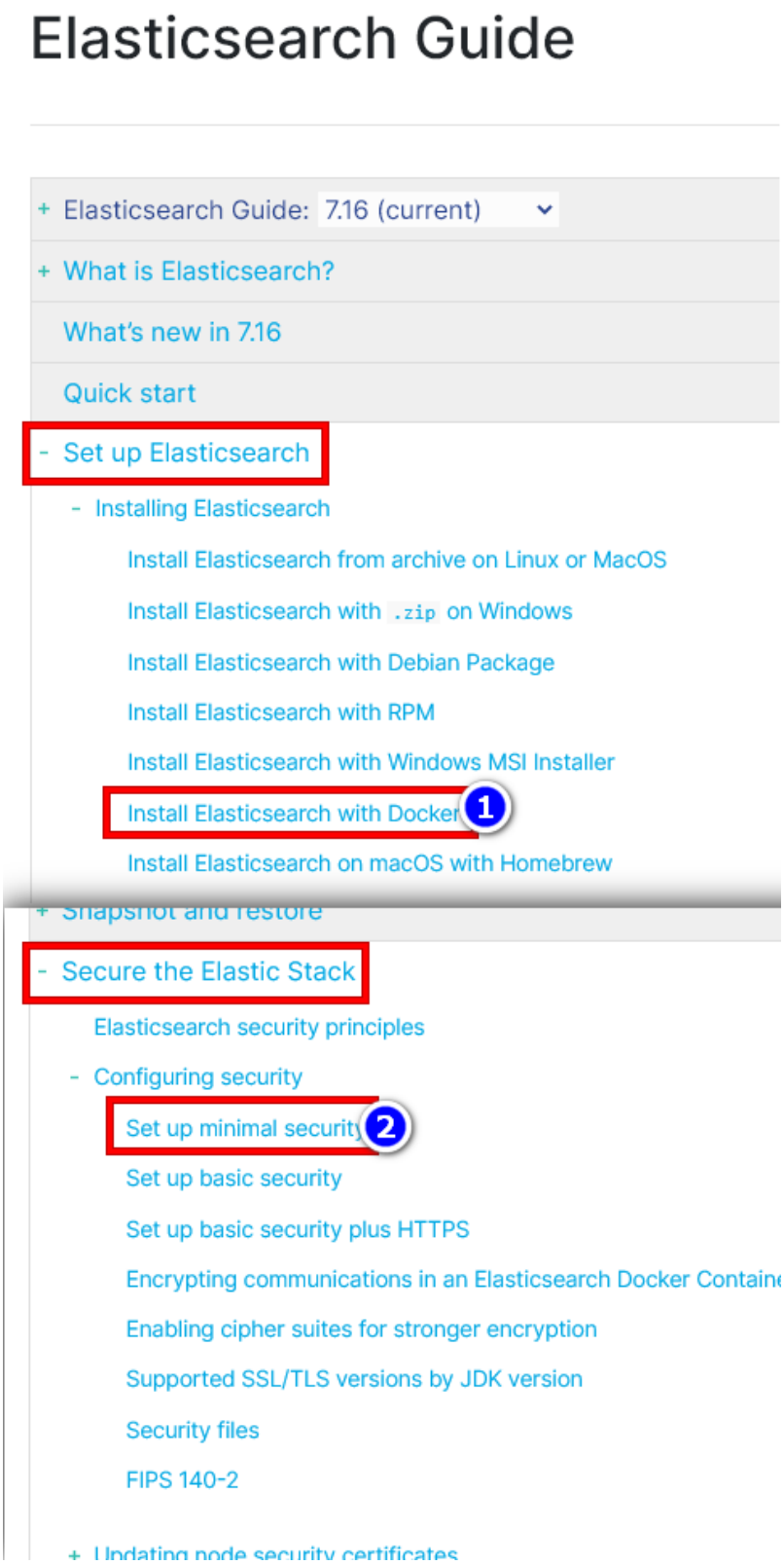
```
# Conectando na VM conforme o IP que o Azure atribuiu.
# Usando o usuario e senha que escolhi.
ssh adminer@20.206.112.142

# Faz download da lista aplicacoes disponiveis para instalacao.
sudo apt update

# Instala o docker.
sudo apt install docker.io -y
```

1.3. Instalando Elasticsearch e Kibana via Docker

No site oficial do Elasticsearch, em elastic.co, opção de menu "Learn" e, então, "Docs", você acha os tutoriais de instalação.



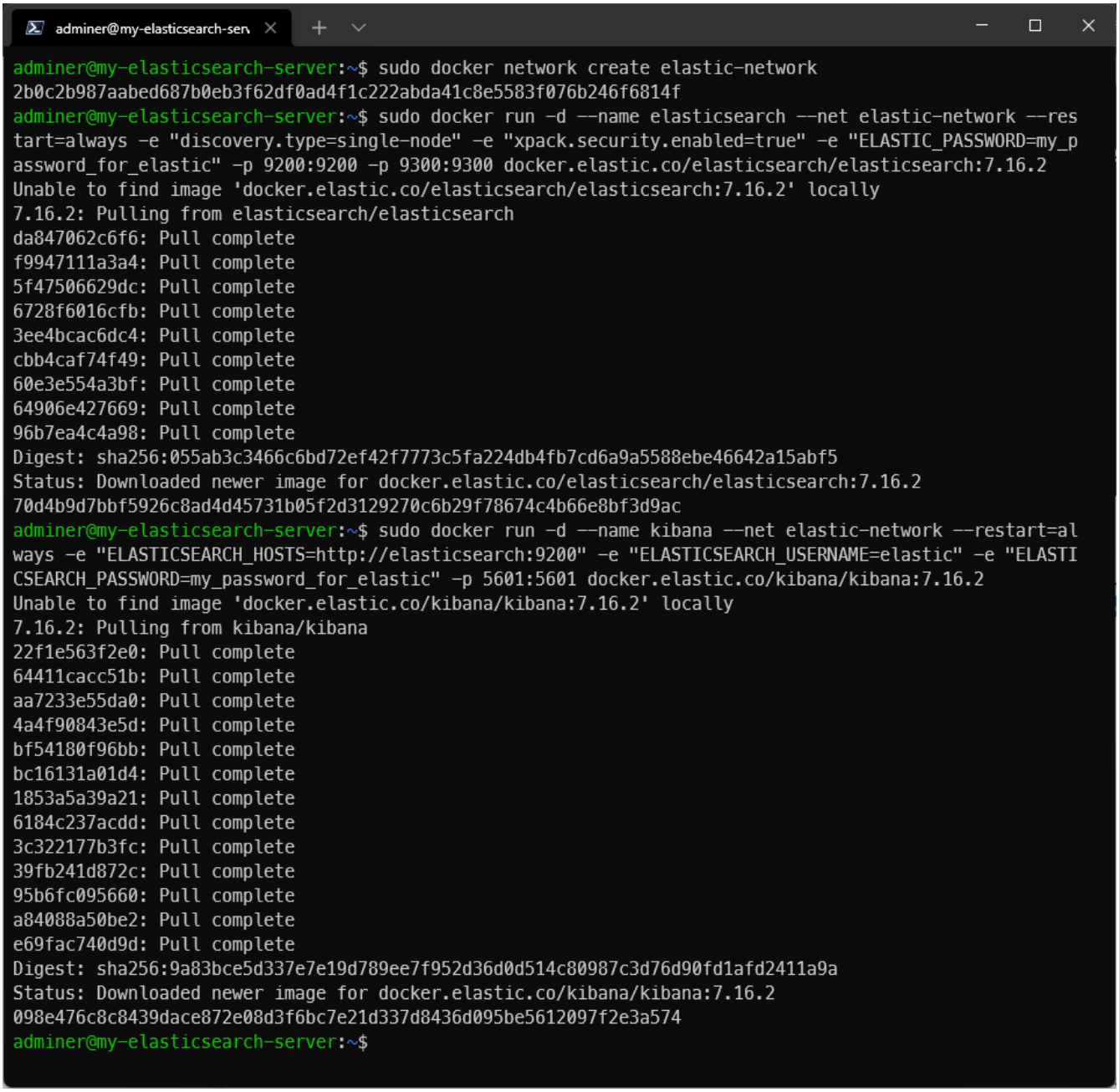
Seguindo o tutorial para instalar via Docker, em "Set up Elasticsearch", "Installing Elasticsearch" e "Install Elasticsearch with Docker". E consultando também o tutorial para definir senha de autenticação em "Secure the Elastic Stack", "Configuring security" e "Set up minimal security"...



Os comandos necessários são:

```
# Cria uma rede que sera usada para o Kibana se comunicar com o Elasticsearch.
sudo docker network create elastic-network

# Executar uma imagem:         sudo docker run
# Em segundo plano:            -d
# Nome do container:           --name elasticsearch
# Nome da rede:                --net elastic-network
# Sempre reinicia o container: --restart=always
# Servidor unico:              -e "discovery.type=single-node"
# Habilita autenticacao:       -e "xpack.security.enabled=true"
# Define a senha tipo admin    -e "ELASTIC_PASSWORD=my_password_for_elastic"
# Porta para dados via API:    -p 9200:9200
# Porta para dados binarios:   -p 9300:9300
# Imagem Docker:               docker.elastic.co/elasticsearch/elasticsearch:7.16.2
#
sudo docker run -d --name elasticsearch --net elastic-network --restart=always -e
"discovery.type=single-node" -e "xpack.security.enabled=true" -e
"ELASTIC_PASSWORD=my_password_for_elastic" -p 9200:9200 -p 9300:9300
docker.elastic.co/elasticsearch/elasticsearch:7.16.2

# Executar uma imagem:         sudo docker run
# Em segundo plano:            -d
# Nome do container:           --name kibana
# Nome da rede:                --net elastic-network
# Sempre reinicia o container: --restart=always
# Endereco do servidor         -e "ELASTICSEARCH_HOSTS=http://elasticsearch:9200"
# Nome do usuario              -e "ELASTICSEARCH_USERNAME=elastic"
# Senha do usuario             -e "ELASTICSEARCH_PASSWORD=my_password_for_elastic"
# Porta para dados via API:    -p 5601:5601
# Imagem Docker:               docker.elastic.co/kibana/kibana:7.16.2
#
sudo docker run -d --name kibana --net elastic-network --restart=always -e
"ELASTICSEARCH_HOSTS=http://elasticsearch:9200" -e "ELASTICSEARCH_USERNAME=elastic" -e
"ELASTICSEARCH_PASSWORD=my_password_for_elastic" -p 5601:5601
docker.elastic.co/kibana/kibana:7.16.2
```

Tendo por saída no terminal:

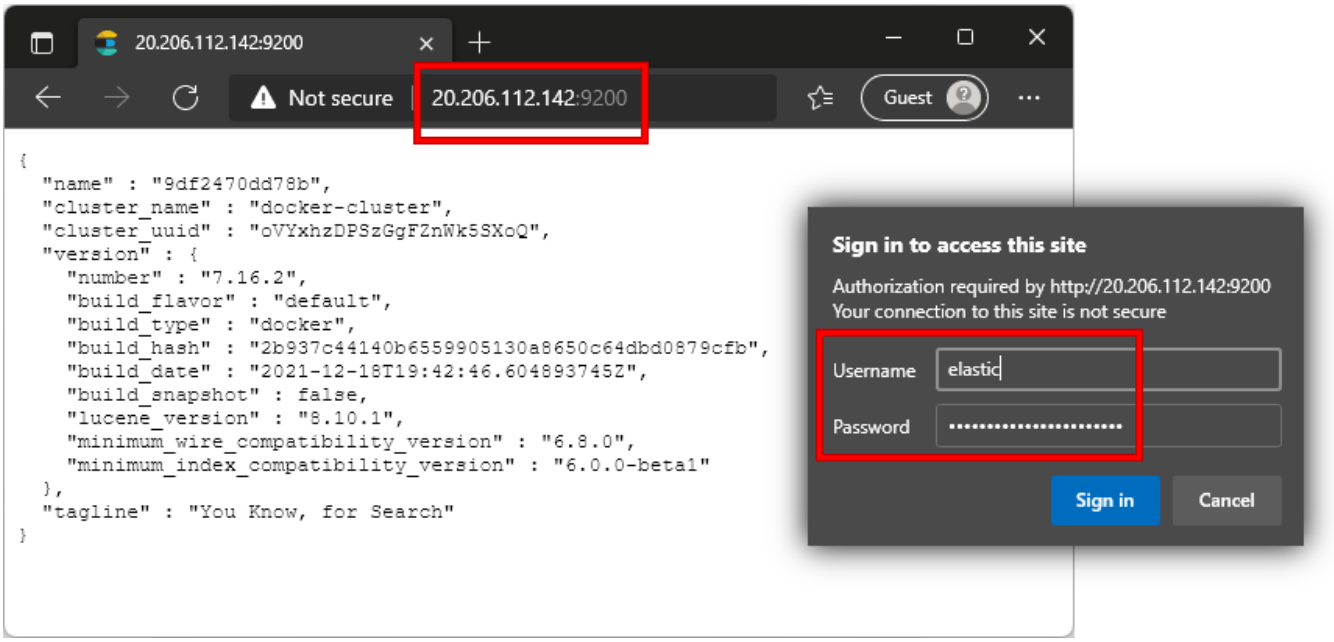


1.4. Acessando as aplicações

As duas aplicações são servidas via protocolo `HTTP`. Então basta acessar o IP especificando a porta de cada aplicação.

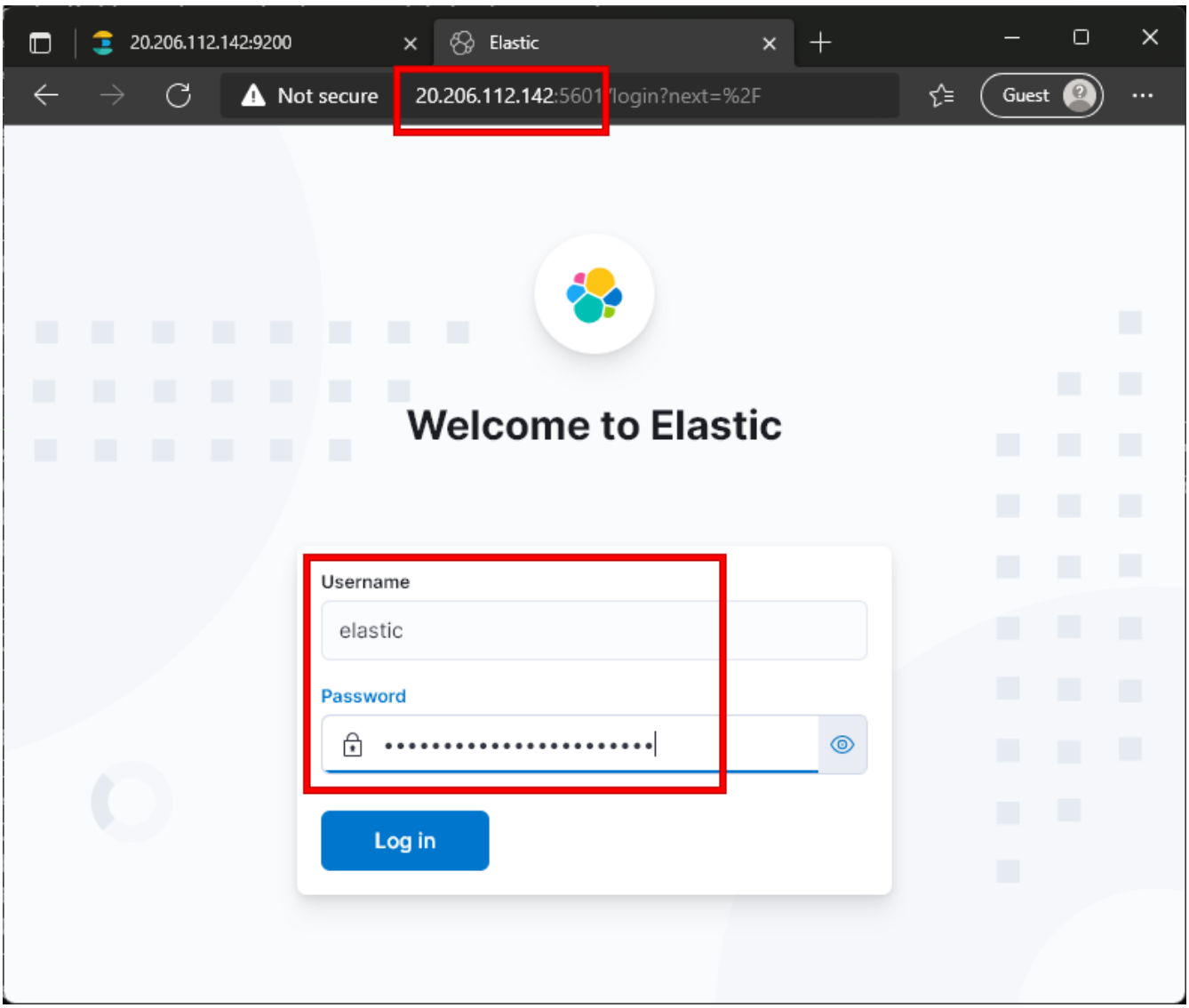
1.4.1. Elasticsearch

O Elasticsearch é servido por padrão na porta `9200`. Será solicitado usuário e senha, sendo o usuário tipo administrador `elastic` e a senha a que foi definida no passo anterior. Não existe interface porque é apenas um endpoint para receber requisições tipo API.



1.4.2. Kibana

O Kibana é de fato uma aplicação web, servida por padrão na porta `5601`. Use o mesmo usuário e senha mencionado no tópico anterior.



2. NodeJS: Criando uma aplicação base

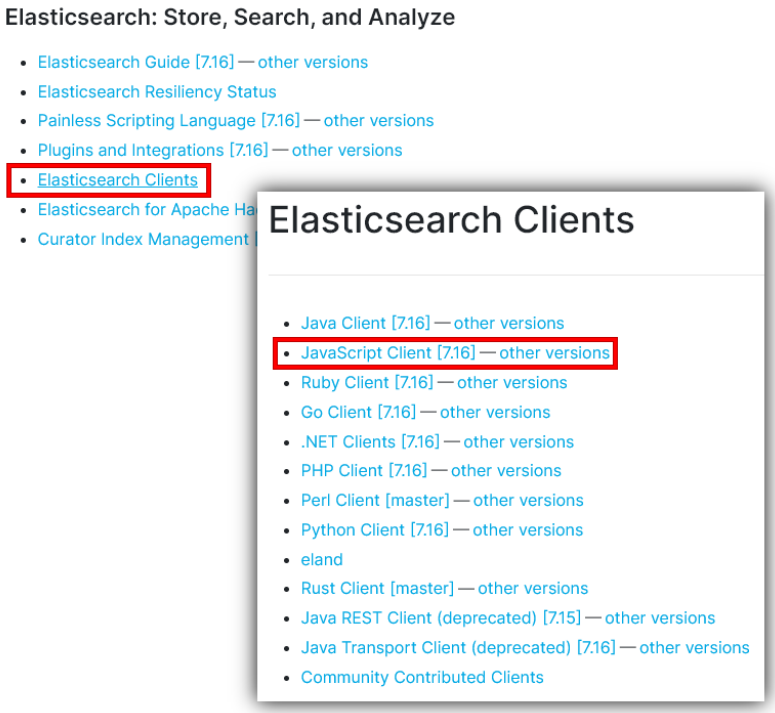
O NodeJS é um ambiente para execução de JavaScript. Por outro lado, o NPM é um gerenciador de aplicações e bibliotecas feitas em NodeJS.

2.1. Criando uma aplicação tipo NPM

Para fazer pronto uso de um cliente capaz de se conectar ao banco de dados do Elasticsearch, começamos criando nossa aplicação como um pacote NPM e em seguida adicionamos a biblioteca `@elastic/elasticsearch`, conforme orienta a documentação oficial.

```
# Cria um projeto tipo NPM com par□metros padr□o
npm init -y

# Instala a biblioteca do cliente Elasticsearch
npm install @elastic/elasticsearch
```

Podemos ver mais sobre esse cliente acima e outros no site do Elasticsearch, em elastic.co, opção de menu "Learn" e, então, "Docs".



2.2. Se comunicando com o Elasticsearch

Crie um arquivo chamado `index.js` e adicione o seguinte código:

```
const { Client } = require('@elastic/elasticsearch');

async function main() {
    // Conexao com o Elasticsearch
    const client = new Client({
        node: 'http://20.206.112.142:9200',
        auth: { username: 'elastic', password: 'my_password_for_elastic' }
    });

    // Inserir um documento
    await client.index({ index: 'my-index-name', id: '1234',
    body: { name: 'Sergio Cabral', website: 'sergiocabral.com' } });

    // Atualizar o documento anterior
    await client.update({ index: 'my-index-name', id: '1234',
    body: { doc: { website: 'sergiocabral.dev' } } });

    // Consulta o documento pelo id
    let response = await client.get({ index: 'my-index-name', id: '1234' });
    console.log(response.body);

    response = await client.search({ index: 'my-index-name', body: { query: {
wildcard: { website: '*sergio*' } } } });
    console.log(response.body.hits.hits[0]);

    // Exclui o documento anterior pelo id
    await client.delete({ index: 'my-index-name', id: '1234' });
}

main();
```

Para mais possibilidades consulte a documentação completa.