**Alunos:** Ruan Carlos Binder da Silva e Yasmin Cristina Leite

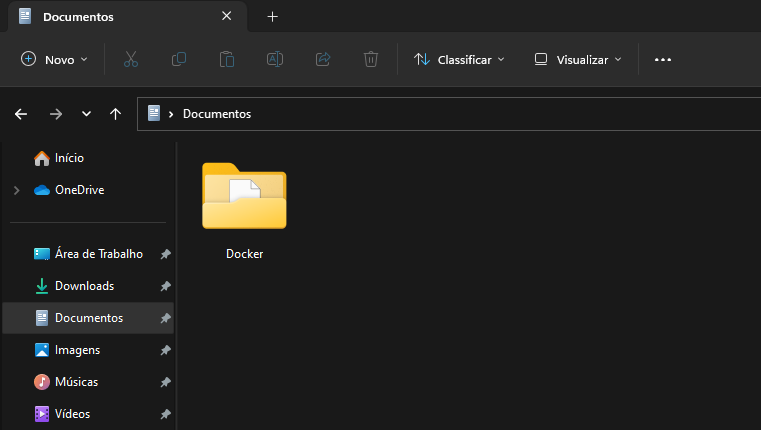
**Tarefa:** Implantação e Execução de um Cluster Apache Hadoop

**Realizando preparação do ambiente**

1. Realizar instalação do [Docker Desktop no Windows](https://docs.docker.com/desktop/install/windows-install/)
2. Após instalar o Docker, é necessário instalar o WSL (Windows Subsystem for Linux), abra um terminal de linha de comando em modo administrativo e digite o comando: **wsl --install**

**Cluster Apache Hadoop**

1. Criar um diretório chamado Docker



1. Dentro do diretório criado, crie um arquivo como o nome de **docker-compose.yaml** com o conteúdo:

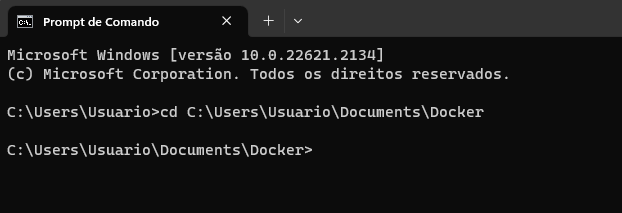
| version: "2"  services:  namenode:  image: apache/hadoop:3  hostname: namenode  command: ["hdfs", "namenode"]  ports:  - 9870:9870  env\_file:  - ./config  environment:  ENSURE\_NAMENODE\_DIR: "/tmp/hadoop-root/dfs/name"  datanode1:  image: apache/hadoop:3  command: ["hdfs", "datanode"]  env\_file:  - ./config  datanode2:  image: apache/hadoop:3  command: ["hdfs", "datanode"]  env\_file:  - ./config  datanode3:  image: apache/hadoop:3  command: ["hdfs", "datanode"]  env\_file:  - ./config  datanode4:  image: apache/hadoop:3  command: ["hdfs", "datanode"]  env\_file:  - ./config  resourcemanager:  image: apache/hadoop:3  hostname: resourcemanager  command: ["yarn", "resourcemanager"]  ports:  - 8088:8088  env\_file:  - ./config  volumes:  - ./test.sh:/opt/test.sh  nodemanager:  image: apache/hadoop:3  command: ["yarn", "nodemanager"]  env\_file:  - ./config |
| --- |

1. Crie o arquivo config

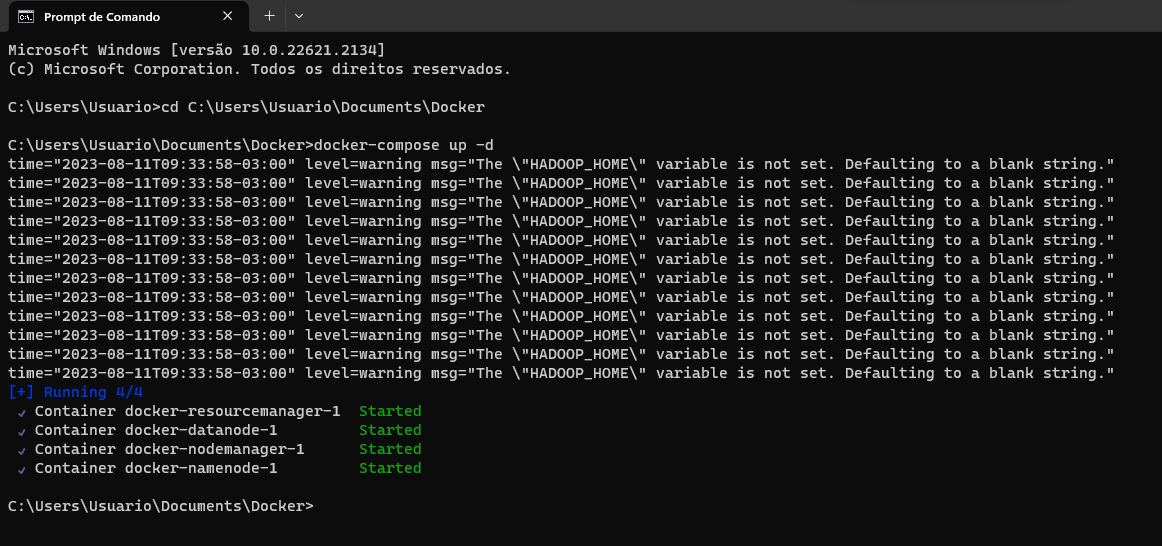
| CORE-SITE.XML\_fs.default.name=hdfs://namenode  CORE-SITE.XML\_fs.defaultFS=hdfs://namenode  HDFS-SITE.XML\_dfs.namenode.rpc-address=namenode:8020  HDFS-SITE.XML\_dfs.replication=1  MAPRED-SITE.XML\_mapreduce.framework.name=yarn  MAPRED-SITE.XML\_yarn.app.mapreduce.am.env=HADOOP\_MAPRED\_HOME=$HADOOP\_HOME  MAPRED-SITE.XML\_mapreduce.map.env=HADOOP\_MAPRED\_HOME=$HADOOP\_HOME  MAPRED-SITE.XML\_mapreduce.reduce.env=HADOOP\_MAPRED\_HOME=$HADOOP\_HOME  YARN-SITE.XML\_yarn.resourcemanager.hostname=resourcemanager  YARN-SITE.XML\_yarn.nodemanager.pmem-check-enabled=false  YARN-SITE.XML\_yarn.nodemanager.delete.debug-delay-sec=600  YARN-SITE.XML\_yarn.nodemanager.vmem-check-enabled=false  YARN-SITE.XML\_yarn.nodemanager.aux-services=mapreduce\_shuffle  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.maximum-applications=10000  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.maximum-am-resource-percent=0.1  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.resource-calculator=org.apache.hadoop.yarn.util.resource.DefaultResourceCalculator  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.queues=default  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.capacity=100  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.user-limit-factor=1  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.maximum-capacity=100  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.state=RUNNING  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.acl\_submit\_applications=\*  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.acl\_administer\_queue=\*  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.node-locality-delay=40  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.queue-mappings=  CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.queue-mappings-override.enable=false |
| --- |

1. Abra o prompt de comando(cmd) e insira o comando para acessar a pasta:

cd C:\Users\Usuario\Documents\Docker



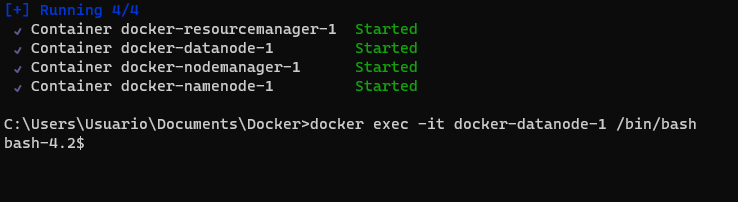
1. Execute os contêineres do docker usando docker-compose utilizando o comando: docker-compose up -d



**Acesso ao Cluster**

1. Realize o login em um nó criado, especificando o contêiner:

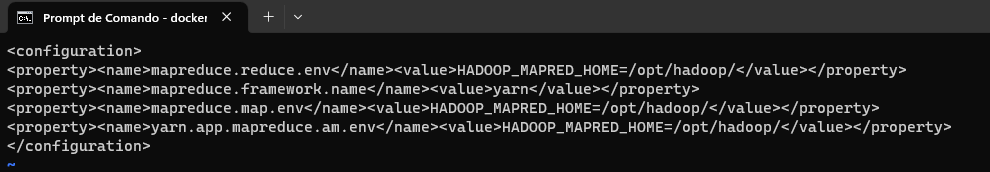
docker exec -it <nome\_do\_container> /bin/bash



1. Antes de executar o programa teste, configure o arquivo mapred-site.xml acessando-o com o comando:

vi /opt/hadoop/etc/hadoop/mapred-site.xml

1. Ao acessar o arquivo mapred-site.xml, insira o caminho /opt/hadoop/ na primeira, terceira e quarta linha linha após HADOOP\_MAPRED\_HOME=, o arquivo deverá ficar da seguinte maneira:

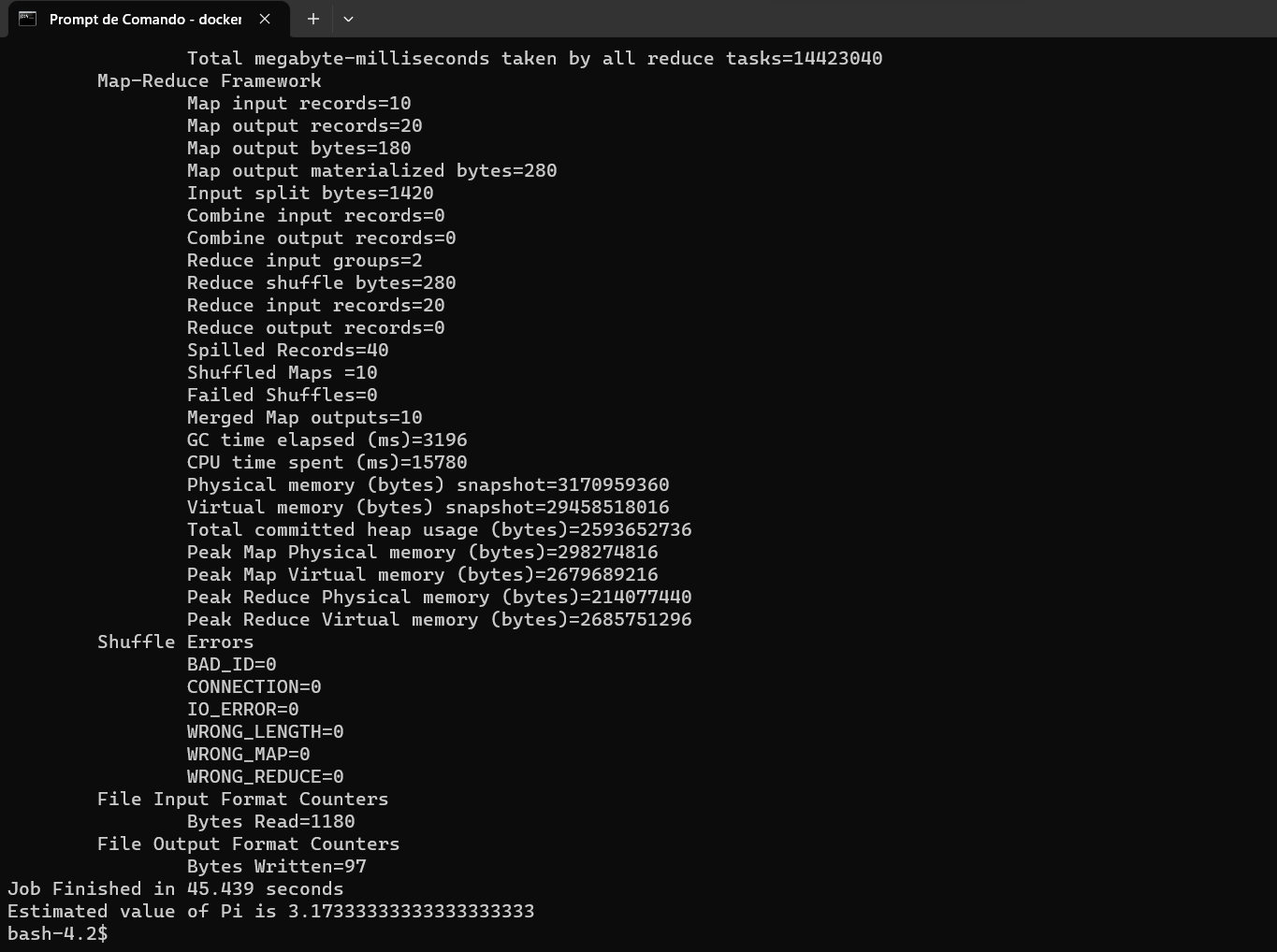


**Executando programas testes**

Após ter acessado o nó e configurado-o corretamente, insira os seguintes comandos para realizar os testes:

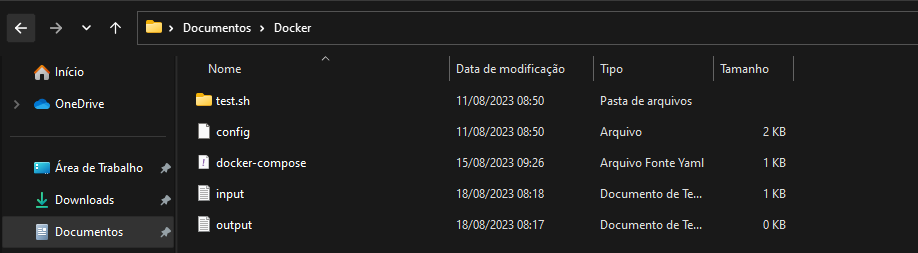
**Para para estimar o valor de Pi:**

* yarn jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar pi 10 15

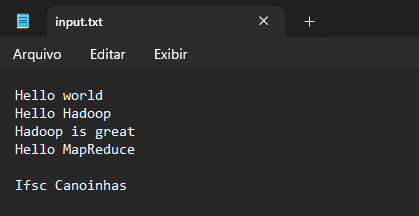


**Contagem de palavras:**

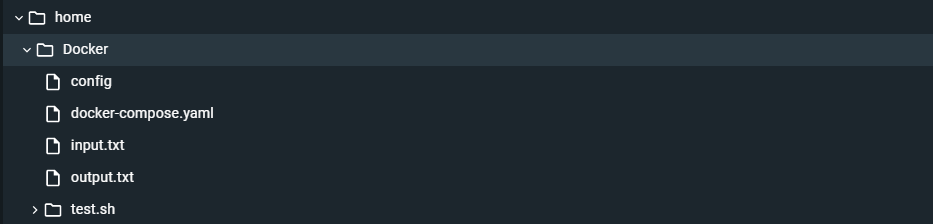
1. No diretório Docker, crie os arquivos input.txt e output.txt.



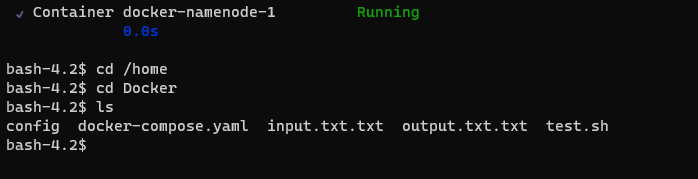
1. Abra o arquivo input.txt em um editor de texto e insira o texto que será utilizado para a contagem de palavras.



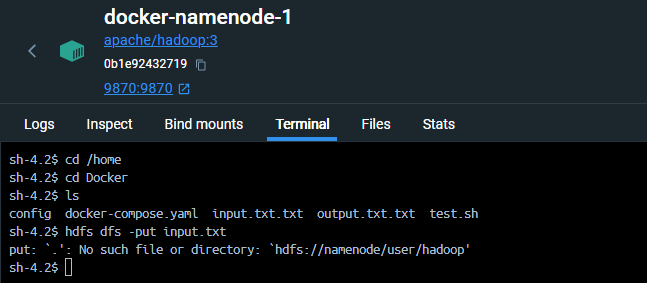
1. No container docker-namenode-1, importe a pasta Docker para o diretório home.



1. No terminal, verifique se a pasta Docker foi importada e se os arquivos estão presentes usando os comandos:



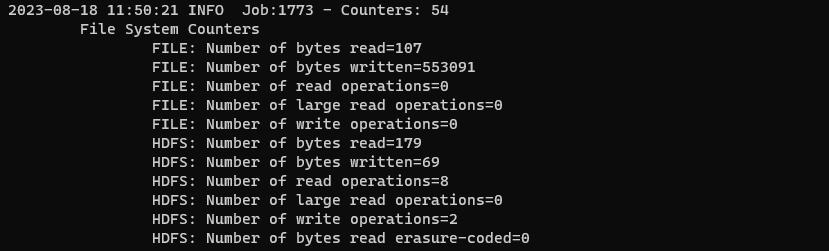
1. Copie o arquivo local para o sistema de arquivos distribuído HDFS, com o comando “hdfs dfs -put input.txt”



Caso apareça o erro (No such file or directory: ‘hdfd://namenode/user/hadoop’), execute o seguinte comando para criar o diretório: hdfs dfs -mkdir -p /user/hadoop.

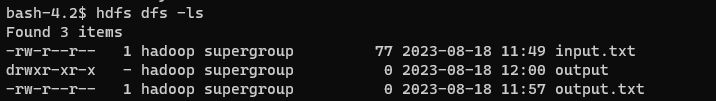
1. Execute o comando:

yarn jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount input.txt output



1. Agora execute os comandos:

**hdfs dfs -ls**



**hdfs dfs -ls output.txt**

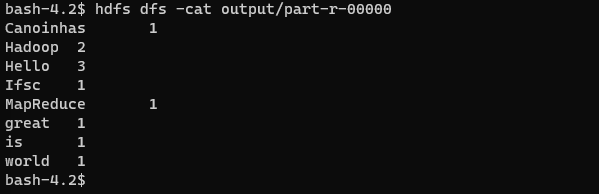
****

**hdfs dfs -get output.txt/part-r-00000**

****

**hdfs dfs -cat output.txt/part-r-00000**

Resultado da contagem das palavras:

****