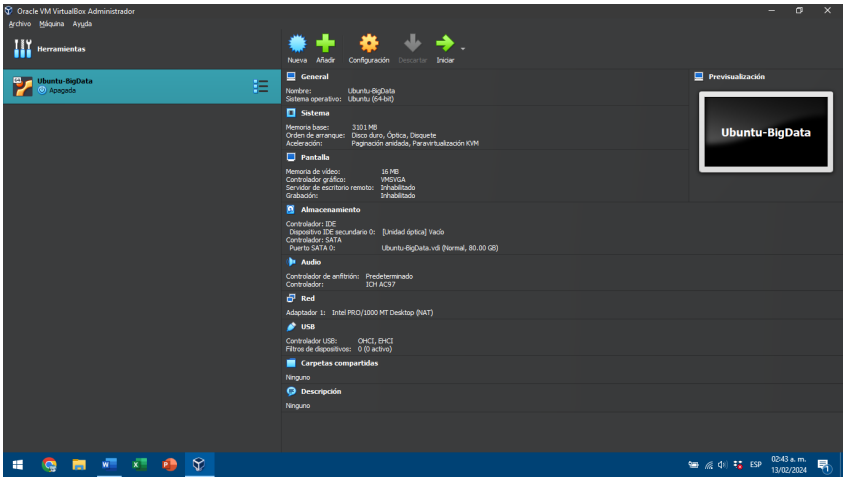
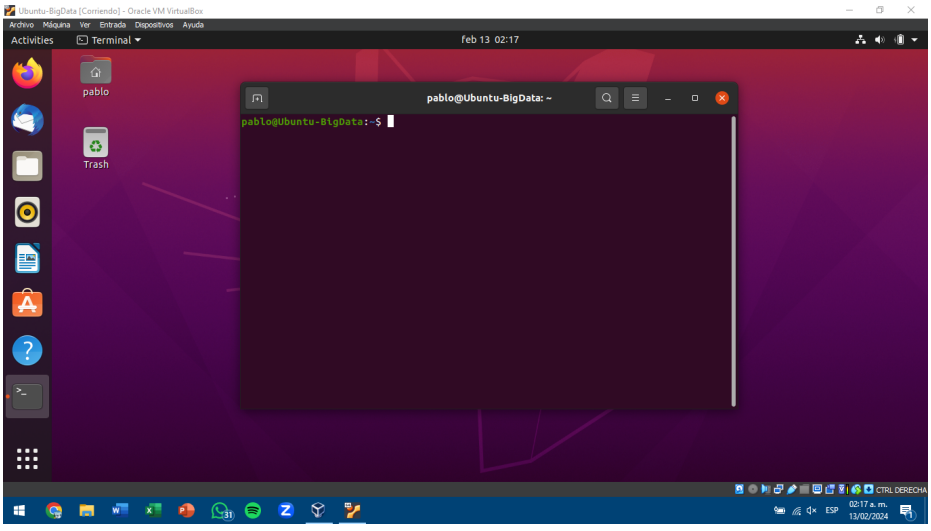
Nombre: Pablo Elías Ramírez Escalante	**Matrícula:** AL02883894
Infraestructura para Big Data	**Nombre del profesor:** Miguel de Jesús Martínez Felipe
Módulo 2	**Avance 2 Evidencia 1**
Fecha: 19/03/2024	
Bibliografia: *Ambari*. (s. f.). https://docs.cloudera.com/HDPDocuments/Ambari/Ambari-2.7.0.0/index.html	

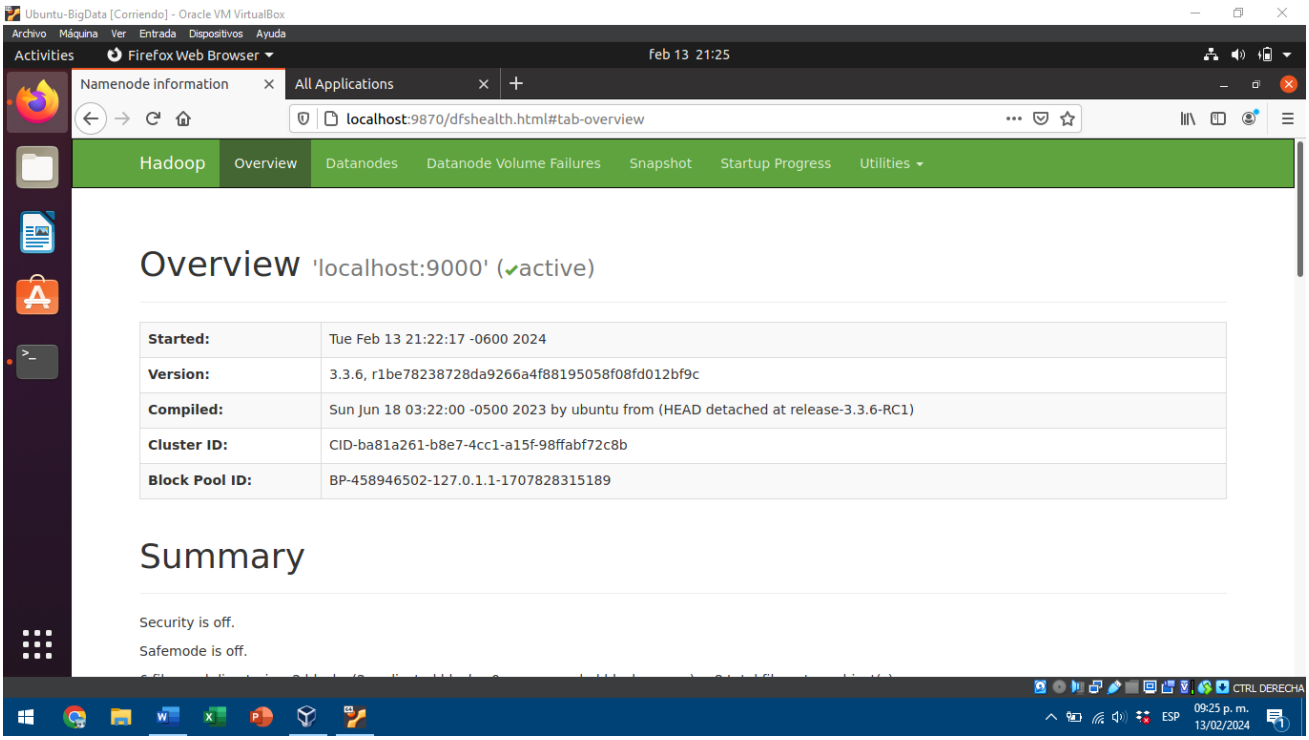
1. Entrega de captura de pantalla de su máquina en Azure (o local)

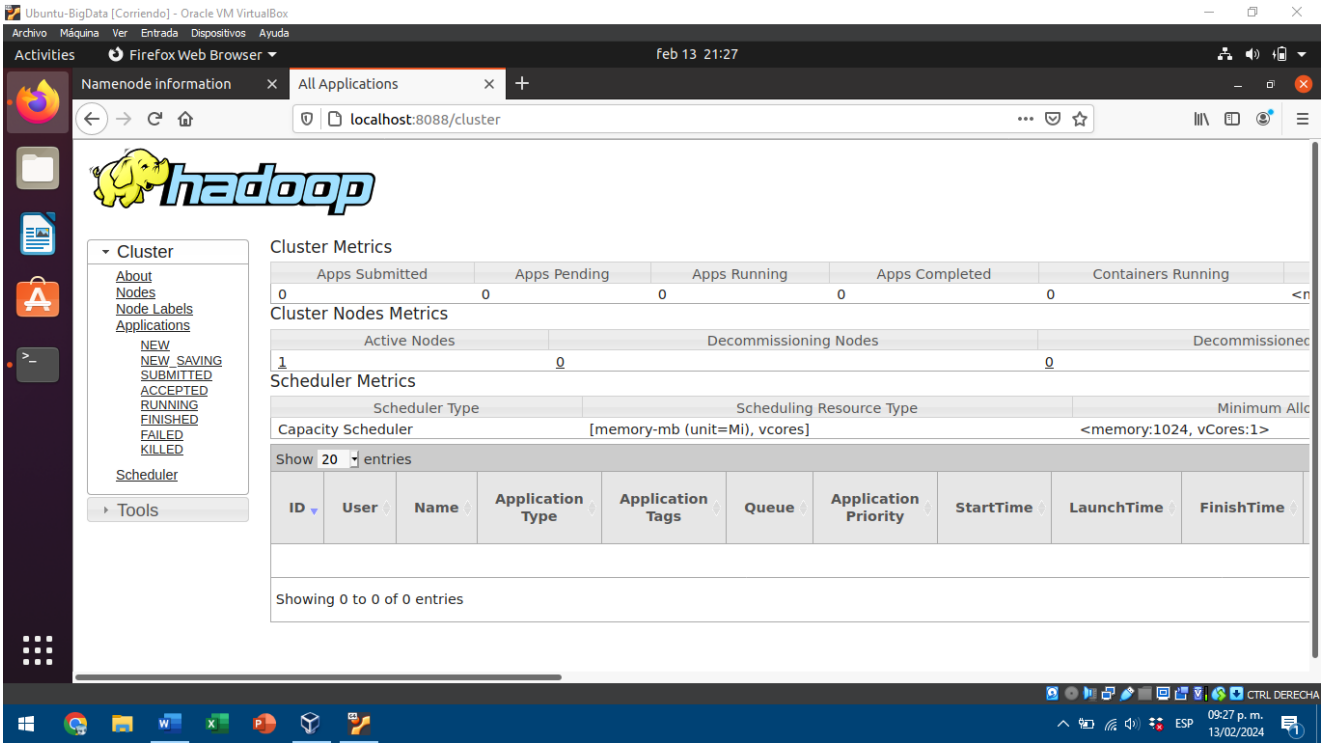


2. Entrega de captura de pantalla de ejecución de Ubuntu



3. Entrega de captura de pantalla de su clúster (Hadoop para quienes lo hicieron local)



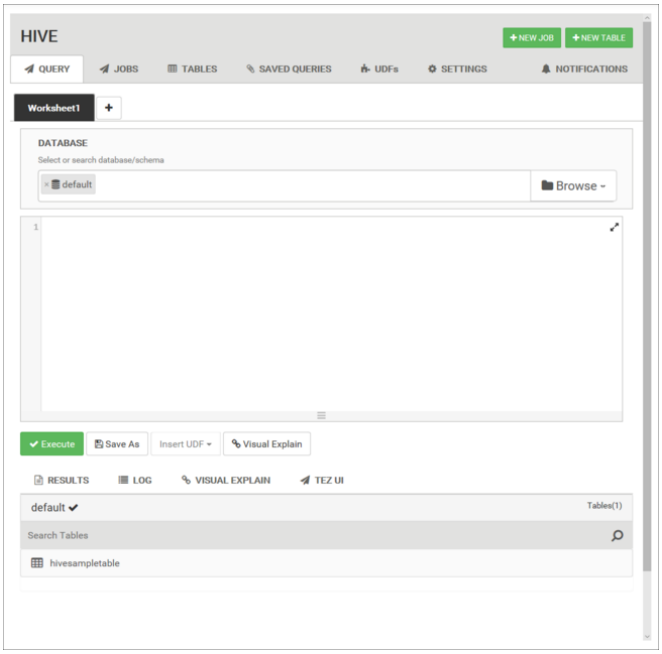


4. Investigación de ¿Qué es Ambari View? (Máximo una cuartilla, mínimo media cuartilla)

Ambari View es una aplicación con muchas funciones que se desarrolla para facilitar la gestión y el seguimiento del clúster de Hadoop. Para administradores y usuarios, Ambari View es una interfaz de usuario basada como una web fácil de usar e intuitiva para administrar y monitorear las instalaciones de Hadoop.

Está aplicación es capáz de proporcionar vistas personalizadas para diferentes roles de usuario y esa es una de las características más destacadas de Ambari View. Esto significa que los administradores pueden configurar Ambari View para que muestre solo la información relevante y las herramientas necesarias para cada usuario según sus responsabilidades dentro del clúster. Por ejemplo, un administrador de sistema puede acceder a vistas que incluyen herramientas de supervisión de recursos y diagnóstico de problemas, mientras que un desarrollador puede acceder a vistas centradas en la gestión de aplicaciones y recursos de computadora.

Además, ofrece capacidades de visualización de datos que permiten a los usuarios analizar el rendimiento y el estado de sus clústeres Hadoop. Esto contiene herramientas de visualización como gráficos interactivos y tablas dinámicas que facilitan la comprensión de la salud y el rendimiento del sistema.



5. Captura de pantalla del ejercicio en mongo código en Python e inserción en Mongo compass

Data to store/insert into MongoDB Database:

In [4]:
```
# One document:
doc_1 = {"_id": 1, "first_name": "Roldoro", "gender": "M", "shift": ["AM", "PM"], "ph_num": 456378903, "trains_for":
```

In [7]:
```
# Multiple documents:
docs = [
    {"first_name": "Miguel", "id": 1, "shift": "AM", "ph_num": 234564980, "trains_for": ["tennis", "badminton"], "ma
    {"first_name": "Victor", "gender": "M", "shift": "AM", "ph_num": 987654321, "trains_for": ["valleyball", "basket
    {"first_name": "Rosa", "gender": "M", "shift": "PM", "ph_num": 456234908, "trains_for": "badminton", "manager":
    {"first_name": "alberto", "gender": "M", "shift": "AM", "ph_num": 123456789, "trains_for": "tennis", "manager":
    {"first_name": "Emma", "gender": "F", "shift": "PM", "ph_num": 776655443, "trains_for": "indoor archery", "manag
    {"first_name": "Hira", "gender": "F", "shift": ["AM", "PM"], "ph_num": 246813579, "trains_for": ["valleyball", "
    {"first_name": "Isha", "gender": "F", "shift": "PM", "ph_num": 135792468, "trains_for": "indoor archery", "manag
    {"first_name": "Alice", "gender": "M", "shift": "AM", "ph_num": 2244668800, "trains_for": "badminton", "manager"
    {"first_name": "Deepali", "gender": "F", "shift": ["AM", "PM"], "ph_num": 113355779, "trains_for": ["tennis", "b
]
```

In [6]:
```
# Dummy documents:
test_docs = [
    {"first_name": "test_1", "gender": "F", "shift": "AM", "ph_num": 234564980, "trains_for": ["tennis", "badminton"
    {"first_name": "test_2", "gender": "M", "shift": "AM", "ph_num": 987654321, "trains_for": ["valleyball", "basket
    {"_id":14, "first_name": "test_3", "gender": "M", "shift": "AM", "ph_num": 987654321, "trains_for": ["valleyball
]
```

In [7]:
```
# Dummy Document:
test_doc = {"_id":14, "first_name": "test", "gender": "M", "shift": "AM", "ph_num": 987654321, "trains_for": ["valle
```

To insert documents to the above collection 'coach_details':

In [5]:
```
# To insert one document:
collection.insert_one(doc_1)
```

Out[5]:
```
InsertOneResult(1, acknowledged=True)
```

In [8]:
```
# To insert multiple documents:
collection.insert_many(docs)
```

Out[8]:
```
InsertManyResult([ObjectId('65f0d986720fe3c9f5130eb7'), ObjectId('65f0d986720fe3c9f5130eb8'), ObjectId('65f0d986720
fe3c9f5130eb9'), ObjectId('65f0d986720fe3c9f5130eba'), ObjectId('65f0d986720fe3c9f5130ebb'), ObjectId('65f0d986720f
e3c9f5130ebc'), ObjectId('65f0d986720fe3c9f5130ebd'), ObjectId('65f0d986720fe3c9f5130ebe'), ObjectId('65f0d986720fe
3c9f5130ebf')], acknowledged=True)
```