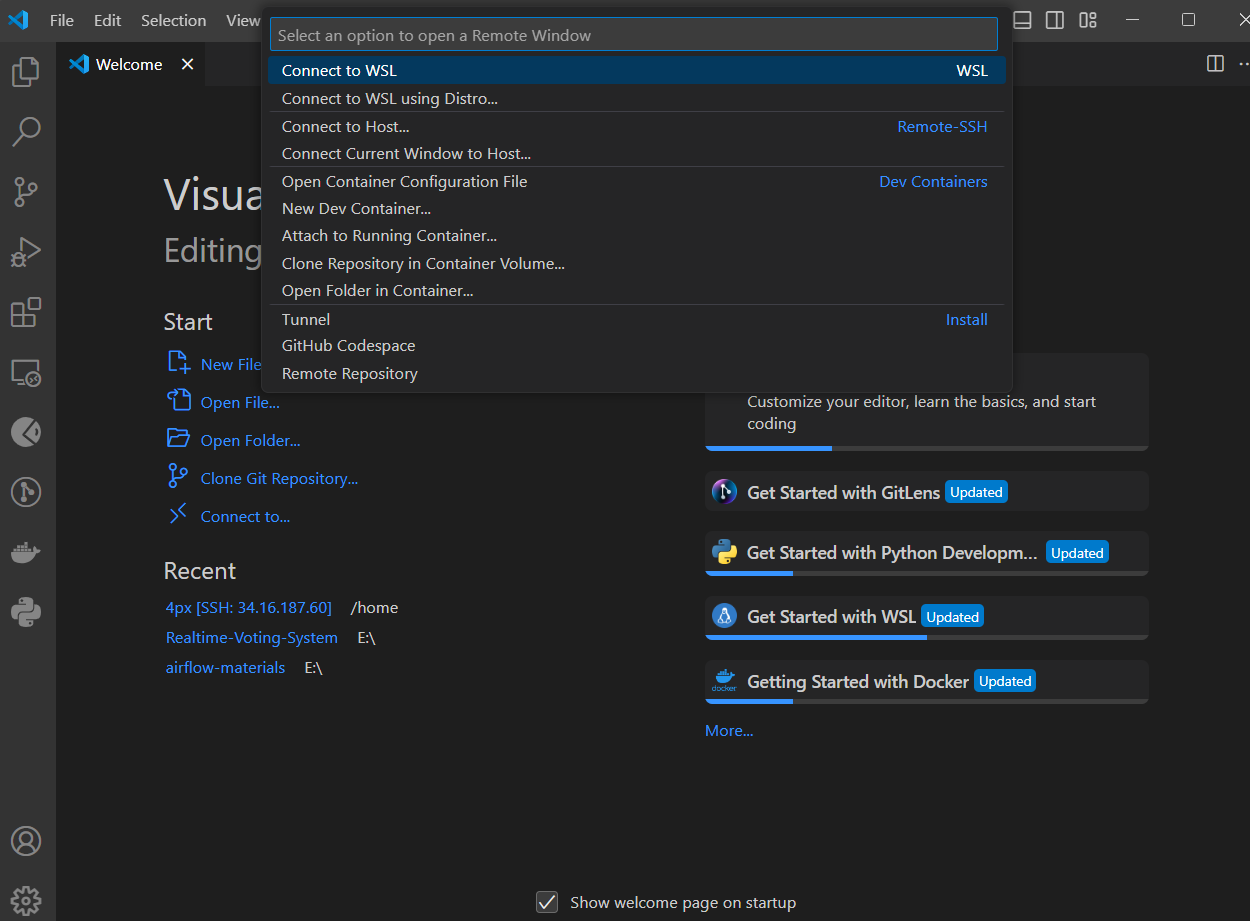
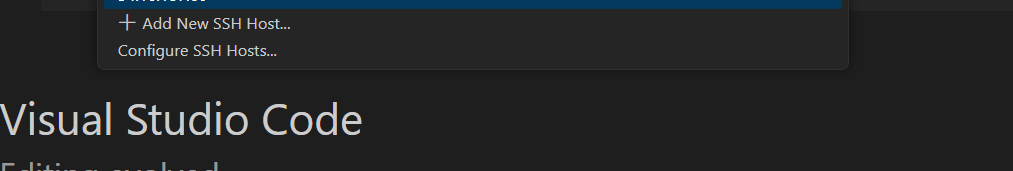
**HOW TO ACCESS Airflow on RND-1 VM?**

* To access Airflow, we need to have Visual Studio Code (VS Code) installed. From VS Code, we'll SSH into the rnd-1 VM to connect to Airflow.

### **Steps to SSH into the RND-1 VM**

1. **Install the Remote - SSH Extension in VSCode**.
2. **Open a new window and click on "Connect to"**.
3. **Select "Connect to Host"**.
4. 

4. Click on add new host 

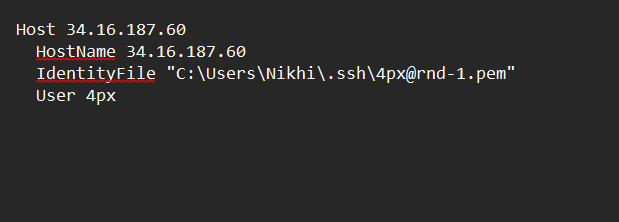
5. Type SSH command - ssh -i 4px@rnd-1.pem 4px@34.16.187.60 and When prompted, choose the SSH configuration file to save your new host entry. The default is usually ~/.ssh/config.

6. Download the pem file for rnd-1 vm from confluence - [GCP VMs - TCL Analytics Cloud - Confluence](https://everlytics2.atlassian.net/wiki/spaces/TAC/pages/3833860/GCP+VMs)

7 add this file to the .ssh folder, where the config file for this host is located. The .ssh folder is typically found at C:\Users\USERNAME\.ssh.

8 . Attempt to connect to the added host.

9 . If you encounter an error, check the logs. If it indicates that a key is missing, copy the path of the key and add it to the config file.



**Sometimes the VM may not be running, so ensure that the VM is up and operational**.

After successfully establishing an SSH connection, we can use the terminal to start Airflow.

### **Airflow**

* **Airflow Host**: 34.16.187.60
* **Airflow Port**: 8181

#### **Commands to Start Airflow**

1. **Start the Webserver**:  
   nohup airflow webserver --port 8181 > airflow.log &
2. **Run DAGs in the Background (Start the Scheduler)**:  
   nohup airflow scheduler &

#### 

**Access Airflow**:  
 Open your browser and go to<http://34.16.187.60:8181> to access the Airflow UI.  
 **Link**:[34.16.187.60](https://34.16.187.60:8181/home)

#### 

#### **Commands to Stop Airflow**

1. **Check Running Airflow Instances (e.g., worker, scheduler, etc.)**:  
   ps aux | grep airflow
2. **Kill the Process**:  
   kill <ID>

**Accessing Ambari**

**Host: hadoop.ddns.net**

**Port:8080**

Ambari agents can be started directly from the UI by navigating to the specified host and port. The UI also displays the host and port details for available services such as HDFS, Kafka, and others.

### **Accessing HDFS**

* **Host**: hadoop.ddns.net
* **Port**: 50075 (Verify this in the Ambari UI)

To access HDFS storage, follow these steps:

1. Go to the **Ambari UI** and select **HDFS** from the list of services.
2. Click on **NameNode UI** under **Quick Links**.
3. In the NameNode UI, select **Browse the Filesystem** to explore the HDFS storage.

### **Integrating HDFS and Airflow**

There are two ways to integrate HDFS with Airflow:

1. Using the **Airflow Provider Package**
2. Using **WebHDFS**

#### **Steps for Using the Airflow Provider Package**

1. First, install the Airflow provider package for HDFS that is compatible with the current Airflow version. To do this, locate the constraint file in the Airflow VM, select the appropriate HDFS version, and install the package using pip in the terminal.
2. In the Airflow UI, go to **Admin** > **Connections**  and add a new connection

Fill in the following details for your HDFS connection:

* **Connection ID**: Set a unique name
* **Connection Type**: Select HDFS from the dropdown menu. (if you don’t find it that means the package is not installed )
* **Host**: Enter the hostname or IP address of your HDFS cluster.
* **Port**: 50070

1. Save the connection
2. Use the same connection ID in airflow dag while using hdfs

You can now refer to the Airflow documentation for instructions on using HDFS operators and hooks.

Additionally, if you choose to use WebHDFS, you can skip the previous steps and simply utilize the Bash operator to execute your actions.

**Accessing KAFKA**  In the Ambari UI, select **Kafka** and navigate to the **Configurations** tab to view the host and port details.

**HOST**: same as Ambari host

**Port**: 6667

The Kafka console for debugging, logs, and executing commands can only be accessed by SSHing into the Ambari VM.

### **Steps for SSH Access**

1. Request the SSH key from the person responsible for managing Ambari.
2. Create a new file in your Linux environment using your preferred text editor.
3. Copy the SSH key into the file and save it with a .pem extension.
4. Change the permissions of the file using the command: chmod 400 <filename>
5. Verify the permissions with ls -l <filename>
6. Connect to the host using the following command:  
   ssh -i <filename> root@host

**Note:** You can use the same steps to access Vertica.

Location for kafka commands = The location for Kafka commands is usr/odp/current/kafka-broker.

**Integrating Kafka and Airflow**

1. To integrate Kafka with Airflow, install the Kafka provider package using pip, ensuring you use the exact version specified in the constraints file.
2. In the Airflow UI, go to **Admin** > **Connections**  and add a new connection
3. Add a unique connection ID and select Kafka as the connection type; if Kafka is unavailable, the package is not installed correctly.
4. Specify the Kafka configuration in the extras field in JSON format, including parameters like bootstrap.servers, consumer group.id etc.
5. Use the same connection ID in hooks and operators; refer to the Airflow documentation for more details.

**Accessing Vertica**

**Host: 34.56.159.191**

**Port: ask sai for port**

To access Vertica, first SSH into the Vertica VM using the steps outlined in the Accessing Kafka section. Once you have successfully SSHed into the Vertica VM, follow these steps:

1. Check the running pods in the Vertica namespace by running the command: kubectl get all -n vertica.
2. Enter the Vertica pod with the command: kubectl exec -it verticadb-sc1-0 -n vertica -- /bin/bash.
3. Now start the vertica using - vsql -U dbadmin -d vertdb
4. Type /d to check existing tables in Vertica.

**Integrating Vertica with Airflow**

First, install the Airflow provider package for Vertica that is compatible with the current Airflow version. To do this, locate the constraint file in the Airflow VM, select the appropriate Vertica version, and install the package using pip in the terminal.

In the Airflow UI, navigate to **Admin > Connections** and add a new connection. Fill in the following details for your Vertica connection:

* **Connection ID**: Set a unique name.
* **Connection Type**: Select Vertica from the dropdown menu (if it’s not available, the package may not be installed).
* **Host**: Enter the hostname or IP address of your Vertica cluster.
* **Port**: Specify the port number .
* User - dbadmin

Save the connection, and use the same connection ID in your Airflow DAG when interacting with Vertica.