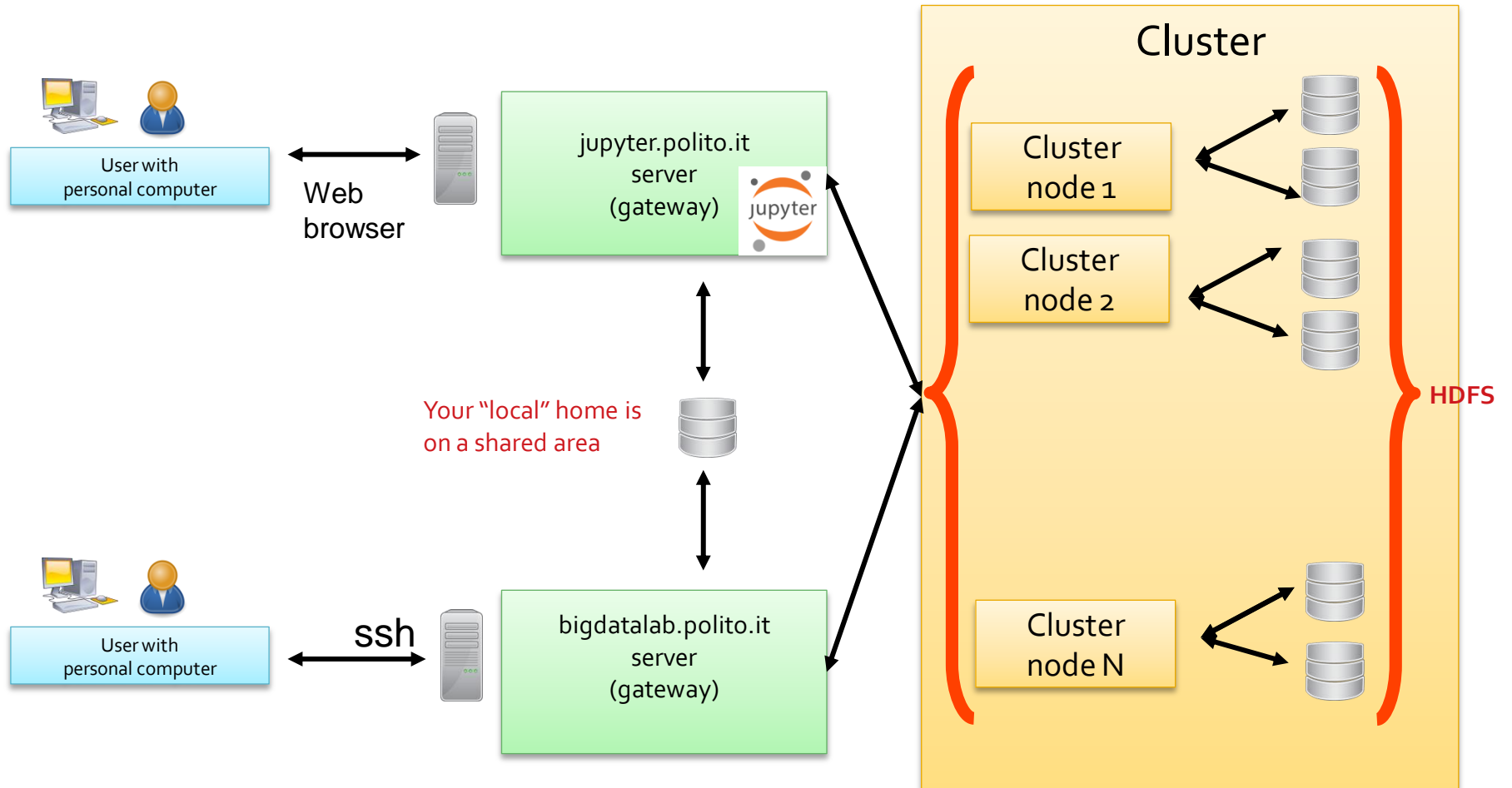


How to execute Spark applications by using Jupyter notebooks

The BigData@Polito environment



Jupyter notebooks



- Jupyter notebook
 - Browser-based interactive IDE
- Specific “notebooks” can be used to run Spark applications on the Spark cluster
 - PySpark (Local)
 - Run the application in a container (in a local instance of Spark)
 - PySpark (Yarn)
 - Run the application on the BigData@Polito cluster
 - Both notebooks read/write data from/in HDFS

Execute an application by using PySpark on a Jupyter notebook



- Copy the input data of your application from the local drive of your personal workstation on the HDFS file system of the cluster
- Open an interactive PySpark shell by using a Jupyter notebook
- Write the python/spark code you want to execute and execute it step-by-step by using the PySpark notebook
- The result is stored in the output HDFS folder specified in your application

Execute an application by using PySpark on a Jupyter notebook

A screenshot of the JupyterLab interface. The top menu bar includes 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', 'Settings', and 'Help'. The left sidebar shows a file explorer for the directory '/ 2019_course /' with various files and folders. The main area displays a '2019_course' notebook environment. Under the 'Notebook' section, there are three options: 'Python 3', 'PySpark (Local)', and 'PySpark (Yarn)'. The 'PySpark (Local)' and 'PySpark (Yarn)' options are circled in black. A callout box with a black border and white background points to these options, containing the text: 'Create a PySpark Jupyter notebook in the local file system of the machine hosting Jupyter (jupyter.polito.it)'. Below the 'Notebook' section, there is a 'Console' section with a 'Python 3' and a 'PySpark (Local)' option. At the bottom, there is an 'Other' section with icons for 'Terminal', 'Text File', 'Markdown File', and 'Contextual Help'.