BIGDATA TEAM

MAIL-BD-2021-Q1 | User Guides

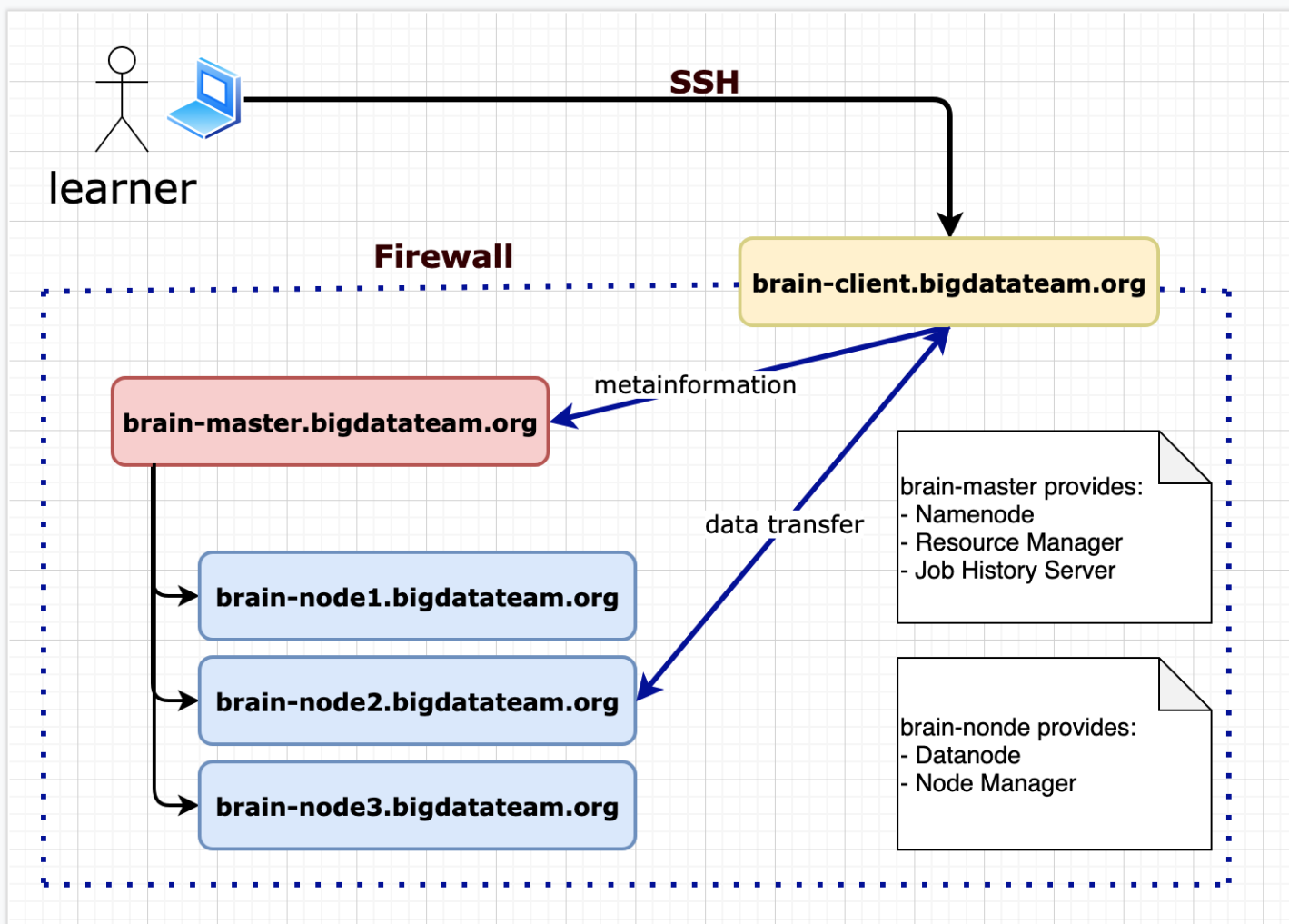
В рамках этого курса вас ожидает:

- Работа на Hadoop-кластере со сдачей заданий на программирование.
- Коммуникация в дружественной атмосфере с коллегами и преподавателями в Telegram-канале.

---

1. Описание кластера и сервисов Hadoop

В первом приближении Hadoop кластер выглядит следующим образом (клиентский узел - brain-client, мастер-сервер (где работает NameNode) - brain-master, рабочие узлы кластера - brain-node1, brain-node2, …):



Hadoop экосистема предоставляет следующие сервисы (и соответствующие порты):

Service	Port	Доступность извне[1]
HDFS Web UI	50070	НЕТ
Resource Manager	8088	НЕТ
YARN JobHistory	19888	НЕТ
Spark History server	18089	НЕТ

2. Доступ к кластеру и проброс портов (port forwarding)

К сожалению, бравые ребята из интернета любят взламывать Hadoop-кластер по этим портам, чтобы запускать майнинговые фермы. Поэтому наши администраторы закрывают ряд портов для доступа извне. Чтобы достучаться до Web-интерфейсов необходимо будет использовать ssh-туннели.

2.1. Unix, Linux, Mac OS

Например, чтобы увидеть HDFS Web UI, необходимо пробросить порт 50070 с мастера:

```
ssh your_login@brain-client.bigdatateam.org -L
50070:brain-master.bigdatateam.org:50070
```

*your_login - замените на свой логин

И пока открыта ssh-сессия, вы сможете заходить по адресу: http://localhost:50070/

Проброс дополнительно порта осуществляется дополнительным ключом -L и значением, пример (команда ниже должна быть записана в одну строку):

```
ssh your_login@brain-client.bigdatateam.org -L
50070:brain-master.bigdatateam.org:50070 -L 8088:brain-master.bigdatateam.org:8088
```

Для того, чтобы удобно копировать файлы с локального ноутбука, советуем пользоваться SCP.

[1] См. раздел про “port forwarding”

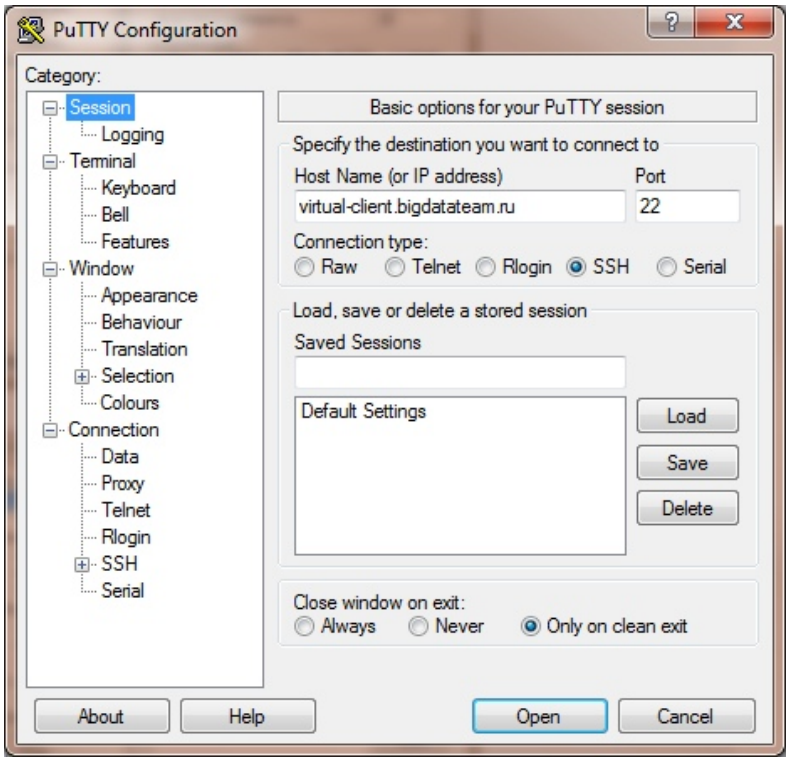
2.1. Windows (Putty)

Disclaimer: мы рекомендуем использовать git-bash на Windows и пользоваться инструкциями для Linux. Если с установкой git-bash есть сложности, то ниже доступны инструкции для Putty.

Если вы пользуетесь Windows, то жизнь у вас немного сложнее и вам необходимо правильно сконфигурировать Putty.

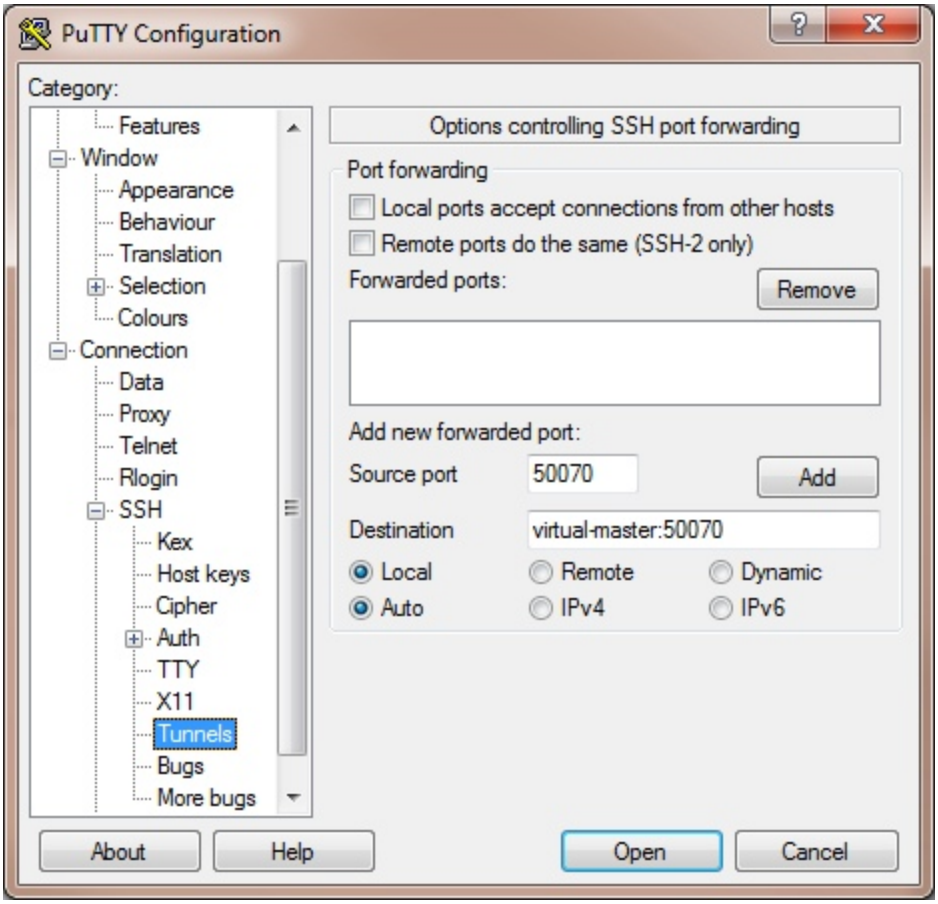
Необходимо настроить параметры в следующих двух категориях (выбор категории осуществляется с помощью двойного щелчка на имя категории в древовидной структуре слева):

1) Категория Session (открывается при запуске Putty), вводим “Host Name” (brain-client.bigdatateam.org[2]), “Port” оставляем значение по умолчанию (22).



[2] На скриншоте указан другой host (прошлого кластера), если кто-то будет пользоваться Putty в рамках текущего обучения - пришлите, пожалуйста, скриншоты с вашего компьютера, чтобы мы обновили инструкцию для ваших коллег. (вам - плюс в карму!)

2) Необходимо добавить проброс портов в категории SSH->Tunnels как показано на скриншоте ниже (только замените virtual-master на brain-master.bigdatateam.org):



Указывайте порт 50070 и/или 8088 в зависимости от того, какой UI нужен. После этого необходимо нажать кнопку “Add”.

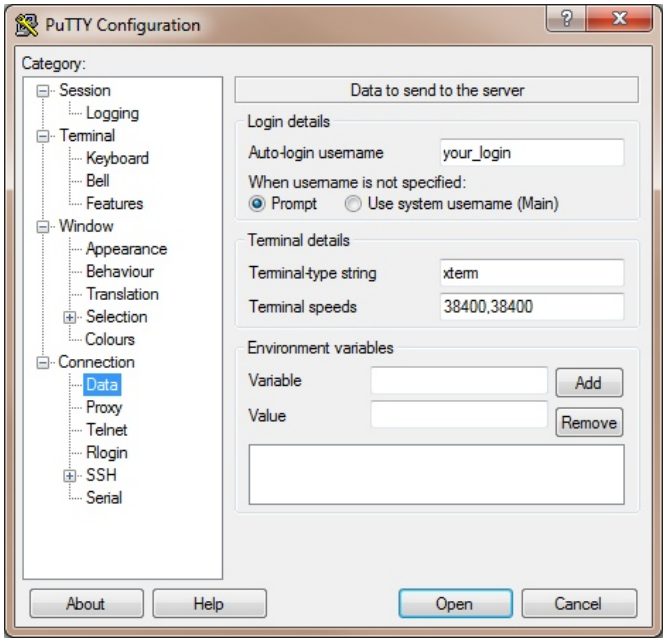
Для проброса нескольких портов еще раз укажите нужны source port и destination, а затем снова нажмите кнопку “Add”.

На данный момент вы уже можете (но прежде - прочтите следующую страницу) нажать кнопку “Open” и затем открыть выбранный UI через localhost в браузере:

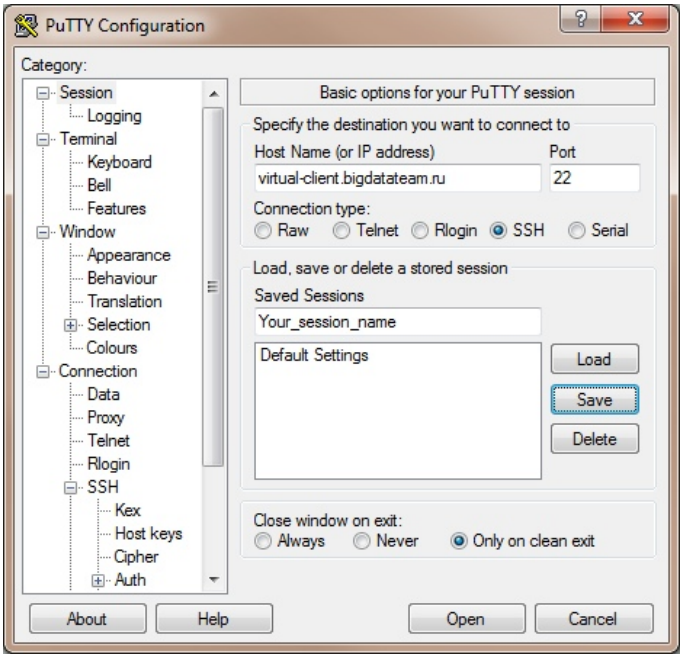
- http://localhost:50070/
- http://localhost:8088/

Но чтобы не настраивать все снова, можно проделать следующие шаги для сохранения настроек:

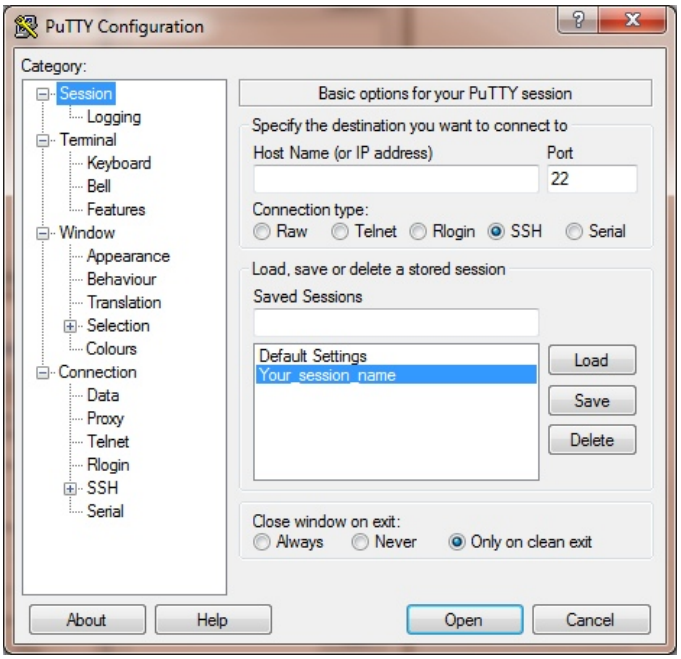
3) Позволяем Putty запомнить ваш логин, для этого вводим его в поле “Auto-login username”:



4) Чтобы непосредственно сохранить (перезаписать) настройки, необходимо ввести имя сессии (там, где на скриншоте написано “Your_session_name”) и нажать на “Save”:



5) Чтобы запустить сессию с сохраненным ранее параметрами, можно дважды нажать на имя сессии ИЛИ нажать на имя сессии так, чтобы имя стало выделено синим, и затем на кнопку “Open”. Чтобы сохранить отредактированные параметры, необходимо нажать на имя сессии и кнопку “Load”, а после редактирования перезаписать (“Save”).



Для того, чтобы удобно копировать файлы с локального ноутбука, советуем пользоваться PSCP.

3. Инструкция по работе с Apache Spark

Для запуска Spark через Jupyter, необходимо выполнить следующие шаги:

1. Усановить SSH-соединение к серверу brain-client.bigdatateam.org
2. В терминале выполнить следующую команду (**все в одной строке**, для удобства копирования - см. github.com/big-data-team/big-data-course)

```
PYSPARK_DRIVER_PYTHON=jupyter PYSPARK_PYTHON=python3.6
PYSPARK_DRIVER_PYTHON_OPTS='notebook --ip=0.0.0.0 --port=port_1'
pyspark --conf spark.ui.port=port_2 --driver-memory 512m --master yarn
--num-executors 2 --executor-cores 1
```

3. Открыть ssh-сессию с параметром `-L port_1:localhost:port_1`
4. Открыть в браузере `http://localhost:port_1`

3.1. Соответствие логина и портов.

Cluster Login	port_1	port_2

*будет обновлено после отбора на курс

3.2. Инструкция по запуску Spark Structured Streaming + Kafka

Для работы коннектора Spark к Kafka, требуется добавить в spark пакет spark-sql-kafka.
Это можно сделать добавив в конец CLI команды для запуска спарк (jupyter/spark-submit) ещё один аргумент:

```
--packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.0
```

Расширенную информацию по использованию этого коннектора можно найти в официальной документации:
https://spark.apache.org/docs/latest/structured-streaming-kafka-integration.html

3.3. Инструкция по запуску Spark + Cassandra

Для работы коннектора Spark Cassandra, необходимо добавить соответствующую библиотеку

```
--packages com.datastax.spark:spark-cassandra-connector_2.11:2.4.2
```

и параметр

```
--conf spark.cassandra.connection.host=brain-node1
```

в команду/скрипт запуска вашего ноутбука. Документация к коннектору доступна по ссылке:

https://github.com/datastax/spark-cassandra-connector#documentation

3.4. Документация по Spark

PySpark: https://spark.apache.org/docs/latest/api.html
Python API: https://spark.apache.org/docs/latest/api/python/pyspark.html
PySpark SQL API: https://spark.apache.org/docs/latest/api/python/pyspark.sql.html

4. Оконные функции

Про оконные функции в Hive можно:

- Послушать 5 минут видео на Coursera "Hive PTF (Window Functions)" (курс "Big Data Analysis: Hive, Spark SQL, DataFrames and GraphFrames", https://www.coursera.org/learn/big-data-analysis)
- Почитать официальную документацию Hive: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+WindowingAndAnalytics

По факту, синтаксис оконных функций в Hive не отличается от синтаксиса в реляционных БД, а материалов и документации в Hive на эту тему мало. Поэтому имеет смысл изучить материалы про оконные функции, доступные для реляционных баз данных. Например:

- Как посчитать всё на свете одним SQL-запросом. Оконные функции PostgreSQL. https://habr.com/ru/post/268983/

5. Unix Workbench, SSH tunneling and port forwarding

В рамках курса вам понадобятся следующие навыки:

- уметь зайти на сервер по ssh
- уметь пользоваться консольными утилитами bash (cat, head, tail, wc, sort, uniq, ...)

Для хорошего погружения рекомендуем “The Unix Workbench” в формате:

- книги: https://seankross.com/the-unix-workbench/
- или онлайн курса на Coursera: https://www.coursera.org/learn/unix

Что такое SSH tunneling и примеры проброса портов с подробным описанием доступны:

- Порт (компьютерные сети)
- https://en.wikipedia.org/wiki/Port_forwarding
- SSH tunnel
- SSH Port Forwarding Example

6. FAQ

Каким образом с помощью hdfs CLI узнать адрес Namenode (и других конфигурационных параметров)?

В hdfs CLI есть модуль getconf, с помощью которого можно узнать значения конфигурационных параметров HDFS:
`aadral@brain-client:~$ hdfs getconf -namenodes`
brain-master.bigdatateam.org
Значение любой переменной можно получить с помощью ключа -getconf, список параметров можно найти на сайте:
• https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml
Например получим стандартный размер блока HDFS:
`aadral@brain-client:~$ hdfs getconf -confKey dfs.blocksize`
134217728[3]

Хочу узнать больше про архитектуру HDFS и состояния реплики блока. Что почитать?
Очень понятное объяснение про состояния реплик и блоков, а также механизмы восстановления после сбоев доступно в следующей работе: Hairong Kuang, Konstantin Shvachko, Nicholas Sze, Sanjay Radia, Robert Chansler Yahoo! HDFS team 08/06/2009 http://files.cnblogs.com/files/inuyasha1027/appenddesign3.pdf Еще одна приятная работа про пределы масштабирования HDFS написана нашим соотечественников Константином Швачко (он же соавтор предыдущей работы): HDFS Scalability: The Limits to Growth Author(s): Konstantin V. Shvachko USENIX, Article Section: DISTRIBUTED SYSTEMS April 2010, Volume 35, Number 2 http://c59951.r51.cf2.rackcdn.com/5424-1908-shvachko.pdf

[3] 128 MB

Как посмотреть в Hive используемую базу данных?
`set hive.cli.print.current.db=true;`

Пишу в консоли скрипты MapReduce или Hive-запросы. Задача не выполняется и пишет странные ошибки, как убедиться, что у меня в коде нет “плохих” Unicode-символов, которые не видно глазом?
см. ресурс https://www.soscisurvey.de/tools/view-chars.php

Хочу узнать больше про регулярные выражения?
В рамках курса Big Data Analysis на Coursera есть опциональная лекция с ликбезом по регулярным выражениям (в Python): - Regular Expressions, Likbez (10 min) В презентацию даются ссылки на полезные ресурсы: Python “re”: • https://docs.python.org/2/library/re.html • https://docs.python.org/2/howto/regex.html#regex-howto Про регулярные выражения: • https://regexone.com/lesson/introduction_abcs • https://regex101.com/

Никогда раньше не использовал YAML, как проверить валидность файла?
Проверить валидность файла можно с помощью онлайн-ресурсов: • https://yaml-online-parser.appspot.com/ (показывает результат парсинга) • https://yamlchecker.com/ (есть встроенный syntax highlight) Посколько онлайн ловит не все проблемы, затем в обязательном порядке в Python: `>>> yaml.safe_load(open("/path/to/solution.yml"))`