

Big Data Analytics

732A54

Technical Introduction

(Introduction of relevant tools and how to use them for the labs)

Mina Abd Nikooie Pour

Based on slides by Maximilian Pfundstein and Erik Rosendal

2024, April

Deadline for lab groups today!

Do not forget to sign up to lab groups in WebReg.



732A54:

<https://www.ida.liu.se/webreg3/732A54-2024-1/LAB/>

TDDE31:

<https://www.ida.liu.se/webreg3/TDDE31-2024-1/LAB>

Objectives for Today's Session

This presentation aims to give you some hints how to use the NSC Sigma cluster along with some theoretical and practical information.

The aim of the labs is not only to learn PySpark, but also to learn how to connect to a cluster and give you an opportunity to broaden your technical knowledge.

This introduction does not cover the programming part of PySpark.

Outline

- ✓ git
 - Introduction
 - Submission rule for lab reports
- ✓ Theoretical Introduction
 - Linux Systems
 - Shells
 - Virtual Environments and Modules
 - Apache Spark and PySpark
- ✓ Practical Introduction
 - Secure Shell & Keys
 - Connecting
 - Developing
 - Submit a job

Git

Introduction

git

- git is a **distributed source version-control system**
 - Distributed
 - Decentralized
 - GitHub, **GitLab** etc are "always running" clients
- git is already installed on Unix systems
- Windows: Must install it manually
 - <https://git-scm.com/download/win>



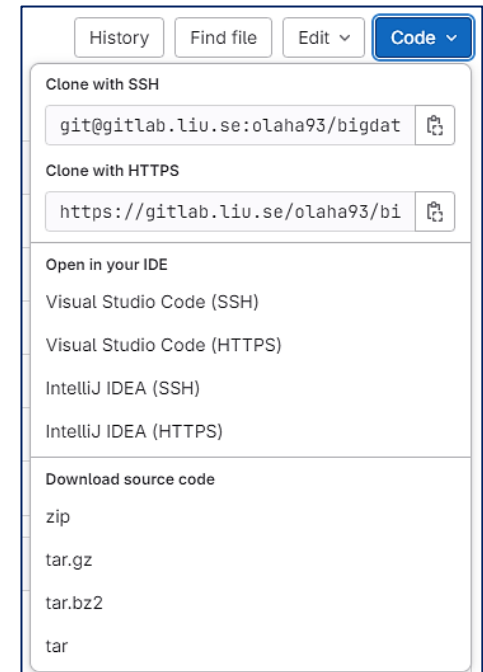
git

- For submitting your lab reports
 - Fork the repository to your repository
 - "Forking" is copying a repository on a hosted git-instance from one user to another
 - To log into GitLab, you can use your LiU ID.
 - Make **private** then grant access rights to:
 - Lab Partner
 - Lab Assistants
 - Teachers
 - Read the readme and Lab Compendiums!

git

- The lab is hosted on a self-hosted GitLab instance
 - <https://gitlab.liu.se/olaha93/bigdata>
- Fork it to your repository
- Bring a copy to your local machine
 - SSH
 - `git clone git@gitlab.liu.se: liuID/bigdata.git`
 - HTTPS
 - `git clone https://gitlab.liu.se/liuID/bigdata`
 - Download as zip file (or another format)

For example:



← ↻ 🔒 https://www.ida.liu.se/~732A54/lab/assignments.en.shtml

IDA - Department of Computer and Information Science

LIU ▶ IDA ▶ Undergraduate ▶ Courses ▶ 732A54 ▶ Lab ▶ Lab Assignments

732A54 (2023)

Course Literature

Examination

Help for written exam

Timetable/Slides

Lab Sessions

Lab Assignments

Sign up for labs (only 732A54)

Contact

INTERNAL

IDA internal

732A54 and TDDE31 Big Data Analytics

Lab Assignments

Deadlines

The final deadlines are the same dates as the dates of the written exam as possible during the course. If you have received comments latest 2 weeks after the exam date.

IMPORTANT: After July, it is not guaranteed that the account [olaha93/bigdata](#) will be available. Please see the repository [olaha93/bigdata](#) to log into GitLab.

Submission Rule: For each lab, the report and code should be submitted by the deadline. For the time being, it is not permitted to use AI-based assistance.

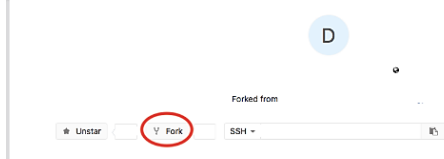
Lab exercises

[732A54 > Lab Assignments \(liu.se\)](#)

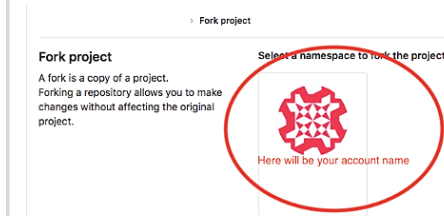
Getting started

- Log into gitlab.liu.se with your LiU-ID
- Fork the repository [olaha93/bigdata](#)

Press the "Fork" button on the top of this page to copy this repository to your account.

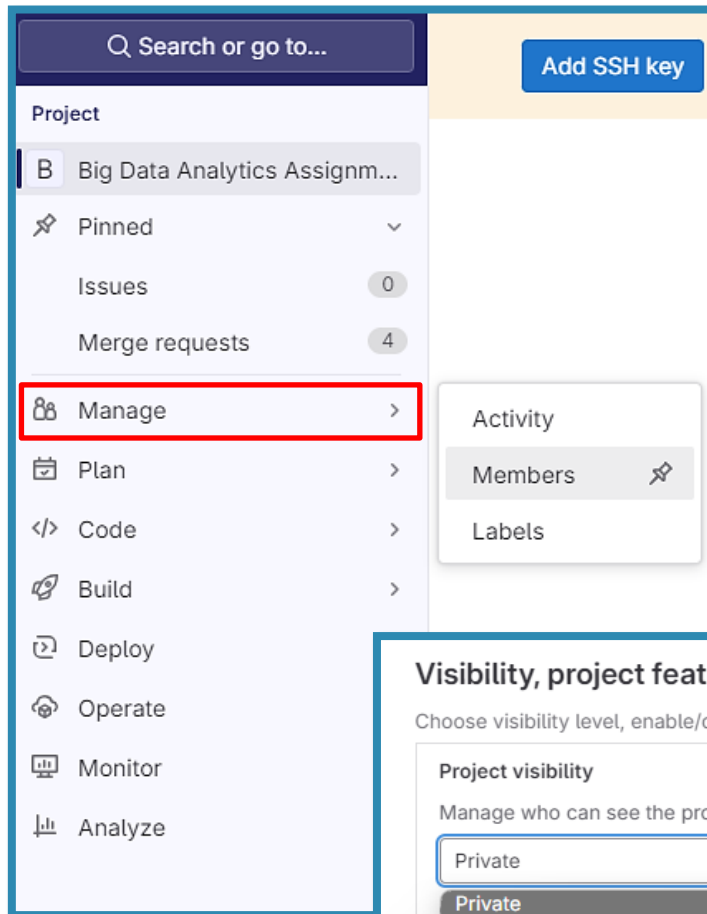


Then, on the next page that pops up, choose your account:



After successfully forking the repository, you will see a message such as the following:

The project was successfully forked.



Search or go to...

Add SSH key

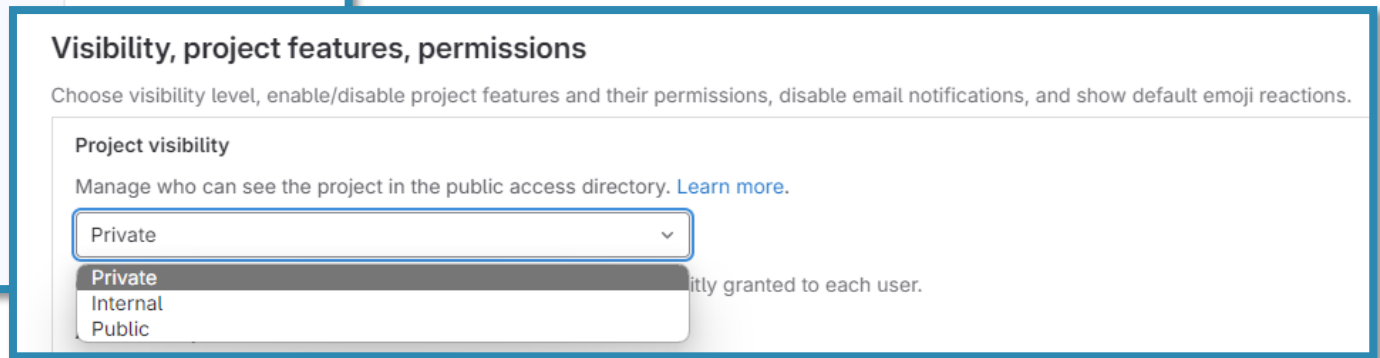
Project

- B Big Data Analytics Assignm...
- Pinned
- Issues 0
- Merge requests 4
- Manage**
- Plan
- Code
- Build
- Deploy
- Operate
- Monitor
- Analyze

Activity

Members

Labels



Visibility, project features, permissions

Choose visibility level, enable/disable project features and their permissions, disable email notifications, and show default emoji reactions.

Project visibility

Manage who can see the project in the public access directory. [Learn more.](#)

Private

Private
Internal
Public

ity granted to each user.

git

- `git pull origin master`
- `git add {file}`
- `git commit -m "Some informative comment"`
- `git push origin master`

git

- Merge conflicts happen and are normal!
 - You can prevent them by not working on the same file, for example by pair programming
- If it happens: Open the conflicted files, search for the conflict, solve it
 - `git mergetool`
- Then stage, commit and eventually push the file

git

- There are GUI clients for git:
 - GitKraken
 - SourceTree
 - Sublime Merge
 - and many more...

git

- For submission, please ensure that you include the link to your lab repository each time you submit or resubmit your report
 - *email a link of your repository* to your lab assistant
 - (BDA3 must be sent to jose.m.pena@liu.se)
 - the subject for that email *must* adhere to the following pattern:
CourseCode_Year_labCode_userName1_userName2

Linux Systems

Theoretical Introduction

Linux Systems

- Prefer using the CLI rather than GUIs, simplifies the "how-to" long-term
- ThinLinc is available for the most parts of your labs
- All relevant information can also be found here:
 - <https://www.nsc.liu.se/systems/sigma>

Shells

Theoretical Introduction

Shells

- The Terminal is the application, the shell the actual interactor
- Command line shells:
 - sh
 - bash (default on most Linux systems)
 - cmd.exe (default on Windows)
 - zsh (default on macOS since Catalina)

Virtual Environments and Modules


Theoretical Introduction

Virtual Environments and Modules

- There exist programs, that set up environments (venv) or modules for you
 - module: <http://modules.sourceforge.net/>
 - conda: <https://www.anaconda.com/>
- Modules are actually doing a bit more, but this will not be part of this introduction
- <https://www.nsc.liu.se/software/modules/>

```
Usage
module --help      General help with module commands
module avail       List the available modules and recommendations
module load ...    Load the selected modules into your session (can also write: module add)
module list        List your currently loaded modules (will be flushed at logout)
```

- If you, for example, launch a python script, your OS needs to know where python executable (the interpreter) is.

```
Open ▾  run_local_with_historyserver.q [Read-Only]  
/software/sse2/tetralith_el9/manual/spark/course-examples/BDA_demo  
1 #!/bin/bash  
2 #SBATCH --time=10:00  
3 #SBATCH --nodes=2  
4 #SBATCH --exclusive  
5  
6 echo "START AT: $(date)"  
7  
8 module load spark/3.5.1-hadoop-3.3.6-hpc1-bdist  
9  
10 # Cleanup and start from scratch  
11 rm -rf spark  
12  
13 # Startup hadoop filesystem and yarn  
14 hadoop_setup  
15  
16 echo "Prepare output and input directories and files..."  
17 # The following command will make folders on your home folder on HDFS, the input and output folders should  
   and saveAsTextFile functions in the code  
18 hadoop fs -mkdir -p "BDA" "BDA/input"  
19 hadoop fs -test -d "BDA/output"  
20 if [ "$?" == "0" ]; then  
21     hadoop fs -rm -r "BDA/output"  
22 fi  
23  
24 hadoop fs -copyFromLocal ./input_data/temperature-readings-small.csv "BDA/input/"  
25 # Remove the comment when you need specific file below  
26 #hadoop fs -copyFromLocal ./input_data/temperature-readings.csv "BDA/input/"  
27 #hadoop fs -copyFromLocal ./input_data/precipitation-readings.csv "BDA/input/"  
28 #hadoop fs -copyFromLocal ./input_data/stations.csv "BDA/input/"  
29 #hadoop fs -copyFromLocal ./input_data/stations-Ostergotland.csv "BDA/input/"
```

Apache Spark and PySpark

Theoretical Introduction

Apache Spark and PySpark

- Apache Spark is written in Java and thus needs the Java JVM to run
- APIs are available for Scala, Java, SQL, Python, R
- This course uses Python and therefore the PySpark API
- Stand-alone and cluster mode
- <https://spark.apache.org/docs/2.4.3/>
- <https://spark.apache.org/docs/2.4.3/api/python/index.html>

← → ↻ 🏠 spark.apache.org/docs/2.4.3/api/python/pyspark.html

🔍 New Tab

PySpark master documentation »



Table of Contents

- pyspark package
 - Subpackages
 - Contents
 - SparkConf
 - SparkContext
 - SparkFiles
 - RDD
 - StorageLevel
 - Broadcast
 - Accumulator
 - AccumulatorParam
 - MarshalSerializer
 - PickleSerializer
 - StatusTracker
 - SparkJobInfo
 - SparkStageInfo

pyspark package

Subpackages

- pyspark.sql module
- pyspark.streaming module
- pyspark.ml package
- pyspark.mllib package

Contents

PySpark is the Python API for Spark.

Public classes:

- **SparkContext:**
 - Main entry point for Spark functionality.
- **RDD:**
 - A Resilient Distributed Dataset (RDD), the basic abstraction in Spark.
- **Broadcast:**

[pyspark package — PySpark master documentation \(apache.org\)](https://spark.apache.org/docs/2.4.3/api/python/pyspark.html)

Secure Shell & Keys

Practical Introduction

Secure Shell & Keys

- Enables creating a remote secure shell, a tunnel
- Can do forward and backwards forwarding
- As well as x-forwarding
- Uses a keypair of a public and a private key, default location is `.ssh`. Unix systems have a default key pair which you can use.
- If not: `ssh-keygen`
- On Windows (e.g. PuTTY) you must create them on your own or use WSL

Secure Shell & Keys

- git can use https or ssh as the underlying protocol
- ssh uses key pairs instead of username and password
- If you log into any git system (GitHub, GitLab) the first time, they usually want you to upload your **public** key for authentication

 You won't be able to pull or push project code via SSH until you add an SSH key to your profile

Add SSH key

Don't show again

```
(base) → .ssh cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC5o2fgA3WMD0IsadxA07xcm/PyCdfqddRm8xC/D
E6jWZjYdRcf2UbrckBx78VJcSpxf8PiWxQBw0rsXgZa6Qp7z6GOYja03E0Hux8m2ERX0D+0+UVnKR
LepiHtwLmjbWCSHck1hrrzRA5BQ/MSYW41hTZ78+IP08aeYogkH97RAscD2HiX/oMP1kRxJA17taj
+GEAnK0eAG1lloqREs3D8K20REl0ng1iw1Mz70r3uz10nTK+ABA0AuRoEXorRrb0EwN71wMh9cRElY
```

Add an SSH key

To add an SSH key you need to [generate one](#) or use an [existing key](#).

Key

Paste your public SSH key, which is usually contained in the file '~/.ssh/id_ed25519.pub' or '~/.ssh/id_rsa.pub' and begins with 'ssh-ed25519' or 'ssh-rsa'. Don't use your private SSH key.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQBAQC5o2fgA3WMD0IsadxA07xcm/PyCdfqddRm8xC
/DE6iWZIYdRcf2UbrckBx78VJcSpxf8PiWxQBwOrsXaZa6Qp7z6GOYia03EOHux8m2ERX0D+0
```

Title

Expires at

Give your individual key a title

Add key

Connecting to Sigma

Practical Introduction

Connecting

- Request Project Membership at SNIC/NSC
 - Project is "liu-compute-2024-2"
 - Follow the instruction
 - [732A54 > NSC Account Application Procedure \(liu.se\)](#)
- Request a login account for Sigma
 - <https://supr.snic.se/> login with SWAMID
 - Choose Linköping University, use liuID to log in
- General info about Sigma:
<https://www.nsc.liu.se/systems/sigma>



Connecting

- [732A54 > NSC Account Application Procedure \(liu.se\)](#)

732A54 (2024)

Course Literature

Examination

Help for written exam

Timetable/Slides

Lab Sessions

Lab Assignments

Sign up for labs
(only 732A54)

Contact

INTERNAL

IDA internal

Student Pages

732A54 and TDDE31 Big Data Analytics

NSC Account Application Procedure

The course project

For this course we use **SNIC**-provided supercomputing resources at the Secure Shell & Keys **National Supercomputer Centre (NSC)**. SNIC/NSC has allocated a special project for our course; the project number is: **LiU-compute-2024-2**

We will have a reserved partition of the Sigma resource for prioritized usage *during scheduled course lab hours*. Outside lab hours you might submit jobs to Sigma but these jobs might wait in the queue for days.

The course project at NSC will at the end of July, so make sure that your labs are completed by the exam date directly after the course. Supervision will only be given during scheduled lab hours.

Note that the teachers in the course may require access to a special directory in your account for grading your exercises.

Student accounts at NSC

All account handling is now done via the national-level SNIC portal **SUPR**. Depending on if you have been registered before or not, the process is different:

Connecting

- CLI (SSH)
- ThinLinc
- GUI (SSH, X-Forwarding)
- More Information for GUI:
<https://www.nsc.liu.se/support/graphics/>

Connecting

- If using Windows, need to enable OpenSSH Client
 - Windows 10: “Add an optional feature”
- If using Windows, you need a terminal:
 - Git Bash: <https://git-scm.com/>
 - WSL: <https://docs.microsoft.com/en-us/windows/wsl/install-win10>
 - <https://cmdr.net/>

Connecting

- Connect
 - `ssh -X ${account}@sigma.nsc.liu.se`
 - `${account}` = NSC account name, e.g. `x_user`
 - Asked for password, password chosen when requesting account for Sigma
- Close connection
 - `exit`

Connecting

- Want to be lazy? Upload your public key!
 - `ssh-copy-id ${account}@sigma.nsc.liu.se`
 - Issue that command in your **local** terminal!

Connecting

- Some useful Linux commands
 - Connect: `ssh`
 - List directory: `ls`
 - Create directory: `mkdir`
 - Change directory: `cd`
 - move one directory back: `cd ..`
 - Secure copy (run on local machine): `scp (-r)`
- Word editors
 - `emacs`
 - `vim`

Connecting

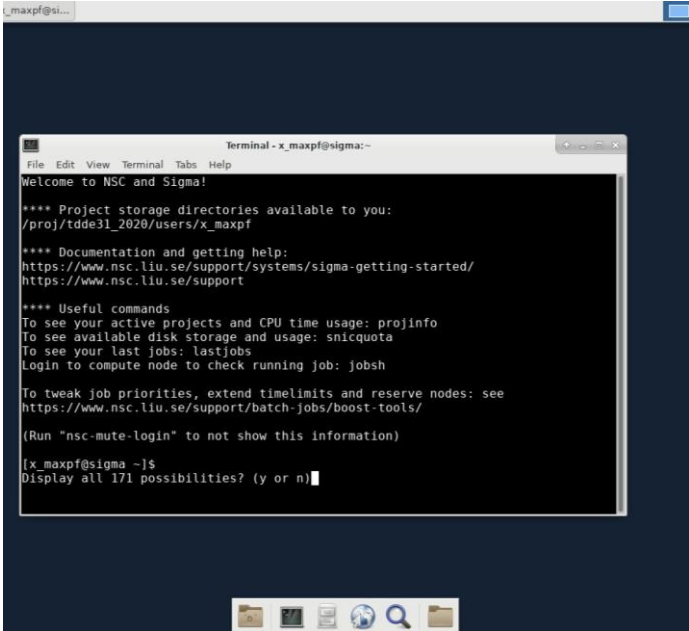
- SSH can do X-forwarding, meaning that you can display a remote GUI applications locally
- When you ssh into a machine, add the option **-X**
- You need a X Window system
 - Linux: xauth
 - macOS: <https://www.xquartz.org/>
 - Windows: PuTTY & Xming
- For details on setting up for your system: Google

Connecting - ThinLinc

Practical Introduction

Connecting - ThinLinc

- Directly use ThinLinc to connect to the cluster
 - `sigma.nsc.liu.se`
 - `${account}`
 - Password
- **Max one login per lab group!**



```

Terminal - x_maxpf@sigma:~
Welcome to NSC and Sigma!

**** Project storage directories available to you:
/proj/tdde31_2020/users/x_maxpf

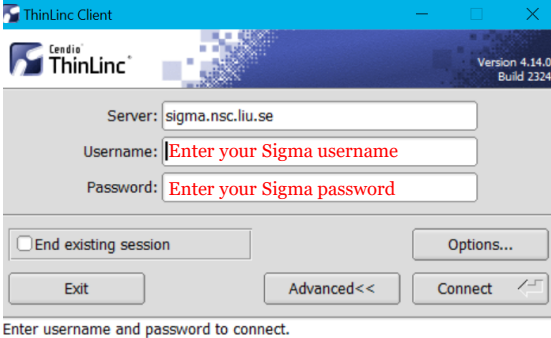
**** Documentation and getting help:
https://www.nsc.liu.se/support/systems/sigma-getting-started/
https://www.nsc.liu.se/support

**** Useful commands
To see your active projects and CPU time usage: projinfo
To see available disk storage and usage: snicquota
To see your last jobs: lastjobs
Login to compute node to check running job: jobsh

To tweak job priorities, extend timelimits and reserve nodes: see
https://www.nsc.liu.se/support/batch-jobs/boost-tools/

(Run "nsc-mute-login" to not show this information)

[x_maxpf@sigma ~]$
Display all 171 possibilities? (y or n)
  
```



ThinLinc Client

Cendio ThinLinc[™] Version 4.14.0 Build 2324

Server:

Username:

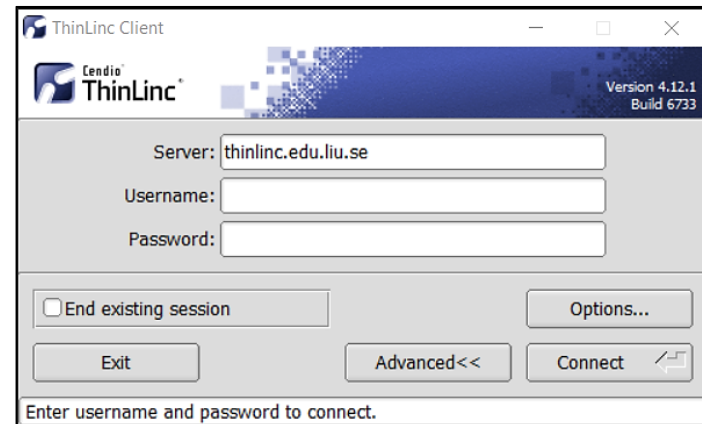
Password:

End existing session

Enter username and password to connect.

Connecting - ThinLinc

- Connect to LiU via ThinLinc (Linux Mint) and do everything from there
 - `thinlinc.edu.liu.se`
 - `{liuid}@student.liu.se`
 - password
- More info about ThinLinc:
 - [Remote login \(sharepoint.com\)](#)



Developing

Practical Introduction

Developing

- “Disconnect” between coding and execution
- Code in separate IDE and execute on cluster
 - VS Code with python Plugin
 - PyCharm
 - JupyterLab
 - GitLab Web IDE
- Develop directly on the cluster
 - vim/emacs

Submit a Job

Practical Introduction

Submit a Job

1. Copy files
2. Submit Job
3. Monitor Job
4. Retrieve Results

Submit a Job | Copy files

- To copy entire folder on Sigma use `cp -R`
– `cp -R {FROM} {TO}`
- To copy to or from local computer use `scp -r`
– `scp -r {FROM} {TO}`
- For script files you can use git (through GitLab)

Submit a Job | Submit Job

- Add job to queue
 - `sbatch -A ... --reservation ... run.q`
- Reservation: Check compendium or
 - `listreservations`
- Look at queue
 - `squeue`
 - `squeue -A liu-compute-2024-2`
 - `squeue -u ${account}`

Submit a Job | Retrieve results

- Look at last entries in file
 - `tail -f ${file}`

Submit a Job | Copy files

- Copy results
 - `scp -r`
`${account}@sigma.nsc.liu.se:/home/${account}/.../output ./`
 - `scp`
`${account}@sigma.nsc.liu.se:/home/${account}/.../output/* ./`

In Summary

- Step 1: Login Sigma with 'ssh -X' connection or Thinlinc.
- Step 2: Copy the demo to your home folder on Sigma.
 - /software/sse2/tetralith_el9/manual/spark/course-examples/BDA_demo/
- Create Q1.py file by renaming demo.py
- Modify Q1.py
- Modify run_yarn_with_historyserver.q: directory of input data
- Change data path to match data in Documents
- Use sbatch to submit the job

```
**** Useful commands
To see your active projects and CPU time usage: projinfo
To see available disk storage and usage: snicquota
To see your last jobs: lastjobs
Login to compute node to check running job: jobsh

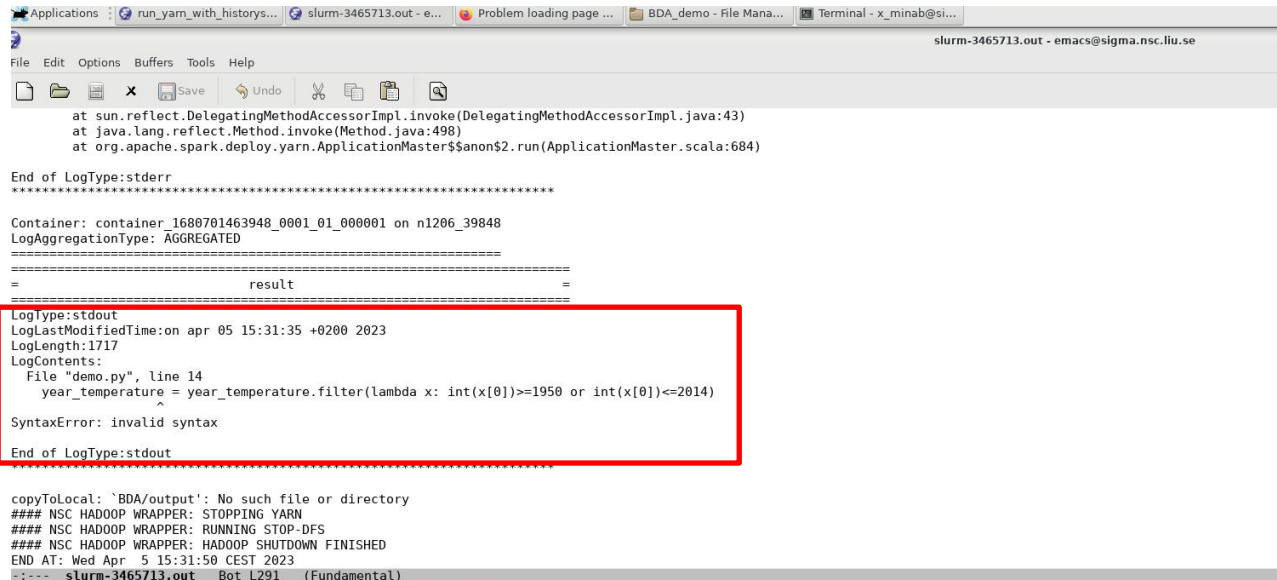
To tweak job priorities, extend timelimits and reserve nodes: see
https://www.nsc.liu.se/support/batch-jobs/boost-tools/

(Run "nsc-mute-login" to not show this information)

[x_mir @sigma BDA_demo]$ sbatch -A liu-compute-2024-2 --reservation devel run_yarn_with_historyserver.q
Submitted batch job 3761678
[x_mir @sigma BDA_demo]$ squeue -u x_mir
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

In Summary

- Check the history log from slurm-ID.out or run spark_browse_job, or check the output folder.



```
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.apache.spark.deploy.yarn.ApplicationMaster$$anon$2.run(ApplicationMaster.scala:684)

End of LogType:stderr
*****

Container: container_1680701463948_0001_01_000001 on n1206_39848
LogAggregationType: AGGREGATED
=====
=                                     result                                     =
=====

LogType:stdout
LogLastModifiedTime: on apr 05 15:31:35 +0200 2023
LogLength: 1717
LogContents:
  File "demo.py", line 14
    year_temperature = year_temperature.filter(lambda x: int(x[0])>=1950 or int(x[0])<=2014)
    ^
SyntaxError: invalid syntax

End of LogType:stdout
-----

copyToLocal: `BDA/output': No such file or directory
#### NSC HADOOP WRAPPER: STOPPING YARN
#### NSC HADOOP WRAPPER: RUNNING STOP-DFS
#### NSC HADOOP WRAPPER: HADOOP SHUTDOWN FINISHED
END AT: Wed Apr 5 15:31:50 CEST 2023
----- slurm-3465713.out Bot L291 (Fundamental)
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial      Learn basic keystroke commands (Emacs användarhandledning)
Emacs Guided Tour   Overview of Emacs features at gnu.org
View Emacs Manual   View the Emacs manual using Info
Absence of Warranty GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions  Conditions for redistributing and changing Emacs
Ordering Manuals    Purchasing printed copies of manuals
To quit a partially entered command, type Control-g.

This is GNU Emacs 24.3.1 (x86_64-redhat-linux-gnu, GTK+ Version 3.22.30)
of 2020-04-03 on x86-01.bs.sys.centos.org
Copyright (C) 2013 Free Software Foundation, Inc.

If an Emacs session crashed recently, type Meta-x recover-session RET
to recover the files you were editing.
Dismiss this startup screen  Never show it again.
```

Things that might be a problem but usually are not

ReqNodeNotAvail

If you see this Reason for a queued job that you expected to start normally: don't worry, it will most likely start normally.

This Reason will temporarily be shown for jobs while we are doing a "rolling upgrade" (an automatic reboot of all compute nodes when their current job finishes).

However, it can also be shown if you request an impossible combination of hardware so that there is no node in the cluster that the job could run on.

If you are unsure, you can run `scontrol hold JOBID; scontrol release JOBID`. If the job after that is still shown as `ReqNodeNotAvail` and you don't understand why, contact [NSC Support](#) and ask for an explanation.

Table 4: Time and Reservation Name

RESERVATION_NAME	Time
bigdata-2024-04-09	04-09, 17:00 to 19:15
bigdata-2024-04-11	04-11, 08:00 to 10:15
bigdata-2024-04-12	04-12, 15:00 to 17:15
bigdata-2024-04-16	04-16, 13:00 to 17:15
bigdata-2024-04-18	04-18, 08:00 to 10:15
bigdata-2024-04-19	04-19, 15:00 to 17:15
bigdata-2024-04-23	04-23, 13:00 to 17:15
bigdata-2024-04-25	04-25, 08:00 to 10:15
bigdata-2024-04-26	04-26, 15:00 to 17:15
bigdata-2024-04-30	04-30, 13:00 to 17:15
bigdata-2024-05-02	05-02, 08:00 to 10:15
bigdata-2024-05-03	Timetable/Slides
bigdata-2024-05-14	Lab Sessions
bigdata-2024-05-16	Lab Assignments
bigdata-2024-05-17	Sign up for labs (only 732A54)
bigdata-2024-05-21	Contact
devel	INTERNAL
	IDA Internal
	Student Pages
	Emergency
	<p>Deadlines</p> <p>The final deadlines are the same dates as the dates of the written exams, although it is highly recommended to do them as possible during the course. If you have received comments that require you to improve your solutions, hand in as late as possible during the course. If you have received comments that require you to improve your solutions, hand in as late as possible during the course. If you have received comments that require you to improve your solutions, hand in as late as possible during the course.</p> <p>IMPORTANT: After July, it is not guaranteed that the accounts on NSC are still available.</p> <p>Submission Rule: For each lab, the report and code should be handed in via a repository in LiU's GitLab. For please see the repository olaha93/bigdata. To log into GitLab, you can use your LIU ID.</p> <p>For the time being, it is not permitted to use AI-based assistants such as ChatGPT for solving any of the assignments.</p> <p>Lab exercises</p> <p>Lab RDB - Relational databases - ONLY 732A54</p> <ul style="list-style-type: none"> Exercise <p>Lab BDA1 - Spark</p> <p>Make sure you read the Lab Compendium before you start the following three labs. (This lab compendium is up to date therefore please refer to this latest version.)</p> <ul style="list-style-type: none"> Exercises

4 Hand In

You are supposed to use GitLab⁷ to submit your report and code. For each lab, please submit the code and a report that contains your results (a snippet of the results is enough if the results contain many rows) and answers to the questions. In cases where a plot of your results is asked, you can include the figure directly in the report. You can use a tool of your preference to produce the plots (R, Excel, matplotlib in Python, etc.). Comment each step in your code to provide a clear picture of your reasoning when solving the problem.

Please ensure to include the link to your repository when submitting or resubmitting your lab report.

The END
Thank you for your attention.