1   How to get LSV clusters running - Step by Step Guide
2
3   Prerequisites:
4   - OS: any Linux distribution, or alternatively WSL (Windows Subsystem for Linux) if you're currently on Windows (how to WSL: https://learn.microsoft.com/en-us/windows/wsl/install)
5   (ssh should also be working out of the box on the Windows command line, but I don't know if there are any intricacies with that and if it will work just as good as on Linux)
6
7   - very basic knowledge of how to use the Linux command line (necessary to connect to (using ssh) and work on the cluster nodes, which run on Ubuntu)
8   (I will walk you through the basics as good as I can)
9
10   - a copy of the project files for the cluster that I put here: [LINK HERE]
11
12   - LSV account, working and password changed from master to your own password
13
14   - ssh installed (should come preinstalled on pretty much any Linux disto, but if not, install via the command 'apt install ssh' (e.g. for Ubuntu)
15   or use any other package manager of your choice)
16
17   - note: wherever I have put the [yourUsername] placeholder, replace it with just your username, drop the square brackets (and don't put any whitespaces in or something like that)
18
19
20 Disclaimer:
21
22   I have explicitly written this guide assuming you don't know how to operate a Linux terminal, to make sure everyone understands what we are doing, even without having any Linux experience.
23   I tried to go into great detail on each step, explaining the commands you need and how to use them, even for the most basic commands.
24   If you already are using Linux and have some experience working with the Linux terminal, much of this guide will feel redundant and overly detailed.
25   I made this decision because I wanted to make sure that everyone can not just reproduce but also understand what they are doing, even if they have no Linux experience.
26   (And because I remember having asked about operating systems and being told that we have 2 or 3 persons on Windows in the group.)
27
28   Please also check out the lsv_cluster_nodes_structure.png infographic I made (in the sopro25 folder), which visualizes the remote LSV nodes we will be using and connecting to.
29
30   This guide is the compiled documentation of my work setting up the project files for the cluster and HTCondor infrastructure, as straightforward as possible, as detailed as needed.
31   The LSV Wiki's guide on HTCondor and the README files in /sopro25 as well as /sopro25/submit-files, which were written by Marius Mosbach, provide some more insight in how to use HTCondor.
32   Make sure to check that out if you need more information.

```
33
34          In case you have any questions or need help setting up HTCondor yourself, do
    not hesitate to contact me, I will help you with anything you need as well as I can.
35
36
37  Some tips in advance:
38
39          - While working with the terminal / command line, make extensive use of the
    autocomplete function, by pressing Tab whenever possible (unsure if Windows terminal
    also has this).
40                  It will autocomplete fully or partially (if in doubt), when you start
    typing the name of a file or directory you want to interact with and then press Tab.
41                  It will make your life much easier.
42
43          - If at any point you need to edit a file once it is already on a cluster node,
    you can use the command 'nano [filename]' to edit it from the terminal (must be in the
    same directory).
44
45
46
47  Some essential Linux console commands you will need:
48
49
50  pwd                 -       prints out the current working directory and path
    (where you are)
51
52  cd [directory]      -       go to the specified directory (you navigate, but
    forward only)
53
54  cd ..               -       navigate to the parent folder ("Go back 1 step". You
    can also go back more steps by using ' cd ../.. ' or ' cd ../../.. ' etc.)
55
56  cd ~                -       navigate to your home directory (' /nethome/
    [yourUsername] '), from anywhere.
57
58  ls                  -       list all contents (subdirectories and files) of the
    directory you are currently in
59
60  mkdir [X]           -       create a new directory X in the directory you are
    currently inside of. ("Make new directory [X] here".)
61
62  rm [X]              -       delete a file or directory. Add flag ' -r ' to also
    recursively delete the contents of a directory. ("Remove [X]".)
63
64  nano [file]         -       edit a file inside the terminal using the nano editor,
    when you need to edit files that are already on the remote node. (nano > vim)
65
66  ssh [destination]   -       connect to some remote node via SecureShell, the LSV
    cluster, for example: ' ssh [yourUsername]@contact.lsv.uni-saarland.de '
67
68  man [command]       -       displays a manual page for the specified command, where
    you can read more on what it does and how to use it
69
```

```
70   scp -r [sourcepath] [destination:path]           -         securely copy directories /
     files via SecureShell (ssh).
71
72           A practical example of me copying a folder (directory) from my device to the
     LSV cluster node would look like this:
73
74                   scp -r ~/Desktop/sopro_cluster_setup/projects/sopro25
     awehner@contact.lsv.uni-saarland.de:/nethome/awehner/projects
75
     |                                   |                                          |
76   syntax:     [command] [path on my machine to the file I want to copy] [my LSV user
     account, '@', hostname of the target, ':', destination path]
77
78
79
80   Outline of the file registry structure on the LSV cluster:
81
82   (Your account and data in the registries outlined below are available on every node.
     There are far more directories on each node, but only the following matter to us):
83
84
85   /
86   |_____ /data
87                 |_____ /users
88                               |_____ /[yourUsername]
89                                             |_____ /logs
90                                                           |_____ /sopro25
91
92   |_____ /nethome
93                 |_____ /[yourUsername]
94                               |_____ /projects
95                                             |_____ /sopro25 (we will copy this
     directory here ourselves, details below)
96
97
98   This is what the file registry structure should look like. In case something does not
     work right away, please make sure all paths outlined here exist (if not, please create
     the relevant missing registries, details on how to do that below).
99
100  Your home registry (' /~ ') will be /nethome/[yourUsername], so we will put all your
     project files in the /projects directory there (create the directory if it's not there,
     details below). In this guide, I will go into detail about how to get code running and
     submit jobs to the cluster using HTCondor (the system that manages, queues and executes
     jobs on the cluster).
101
102
103
104  1. Download the sopro25 project folder from here: LINK HERE
105
106
107
108  2. Open the following three files in an editor of your choice:
109
```

```
110          - run.sh        (in folder 'scripts')
111          - setup.sh      (in folder 'scripts')
112          - run.sub       (in folder 'submit-files')
113
114
115
116  3. In all three files, you will need to edit any occurence of 'awehner' (my LSV
     username) with your own LSV username, to correctly set up the paths to work on your LSV
     account on the cluster.
117          here's where to edit in detail:
118
119          - run.sh        line 4
120          - setup.sh      lines 4,7,8
121          - run.sub       lines 5,6,9,10,11
122
123
124
125  4. Connect via ssh to contact.lsv.uni-saarland.de, using the following command:
126
127          ssh [yourUsername]@contact.lsv.uni-saarland.de
128
129          It will prompt you for your password upon connecting, just enter the password
     and press enter.
130
131
132
133  5. Check if the file registry on the remote node matches the outline in the graph, if
     not, create all missing directories and see that each path exists.
134
135          You will start out in /nethome/[yourUsername] , so to get to /data you have to
     navigate back to / first, using:
136
137                  cd ../.. (go back 2 directories)
138
139          Then use this to get to /data/users/[yourUsername]:
140
141                  cd data/users/[yourUsername]
142
143          To go back to /nethome/[yourUsername] , navigate to / using cd ../../.. (to go
     back 3 directories) again and then use:
144
145                  cd nethome/[yourUsername]
146
147          or you can use
148
149                  cd ~ (the "Go back home" - command)
150
151          from anywhere to get directly back to /nethome/[yourUsername] (your home
     directory).
152
153
154
155          First navigate to /nethome[yourUsername] and check if /projects exists with
     the
```

```
      ls command, if not, create it using the command:
156
157                  mkdir projects
158
159        Then, use ls again to confirm you have created it there.
160
161        Next navigate to /data/users/[yourUsername] and check if /logs exists, if not,
      create /logs with:
162
163                  mkdir logs
164
165        You can again check with the ls command if it's now there.
166
167
168
169   6. Copy over the data from your machine to the cluster node.
170
171   Open a fresh terminal (that's not logged in to the cluster node via ssh) and copy over
      the sopro25 project folder from your machine to /nethome/[yourUsername]/projects on the
      remote node, using the scp -r command:
172
173        For example, from my machine it would be this (first argument: source, second
      argument: destination, -r flag to recursively copy the folder's content):
174
175                  scp -r ~/Desktop/sopro_cluster_setup/projects/sopro25
      awehner@contact.lsv.uni-saarland.de:/nethome/awehner/projects
176
177        You will be prompted to enter your password again, so just enter it and confirm
      with Enter.
178
179        Once you have finished copying the files, switch back to the other terminal
      that's connected to the node and navigate to /nethome/[yourUsername]/projects and use
      the ls command to check if the directory is there.
180
181
182   7. Build and push the docker image.
183
184        Now you need to connect (from the contact.lsv.uni-saarland.de node) to
      workstation 71 of the LSV cluster, which we can use to build docker images and push
      them to the LSV docker registry.
185
186        Use the command:
187
188                  ssh [yourUsername]@ws71lx.lsv.uni-saarland.de
189
190        to connect to the workstation, enter your password when prompted and confirm
      with enter.
191
192        Here, you can also access your data like you would on the contact node.
193
194        Navigate to /projects/sopro25/docker and build the docker image using the
      following command:
195
```

196   IMPORTANT NOTES:
197   - You have to edit the command to include your username towards the end where I
     put the [yourUsername] placeholder.
198   - When executing the command, include the whitespace and dot at the very end,
     it is part of the command.
199
200   docker build -f Dockerfile --build-arg USER_UID=$UID --build-arg
     USER_NAME=$(id -un) -t docker.lsv.uni-saarland.de/[yourUsername]/
     dockerimage_sopro25:v1 .
201
202   Lastly, after you finished building the image, push it to the LSV docker
     registry with this command (again, you will need to replace the placeholder with your
     username towards the end):
203
204   docker push docker.lsv.uni-saarland.de/[yourUsername]/
     dockerimage_sopro25:v1
205
206   After successfully pushing the image to the registry, log out of the
     workstation node by simply typing the command 'logout' into the terminal.
207
208   You should now be back to the contact node.
209
210   8. Submit a job
211
212   From the contact node, connect to the submit node by using the following
     command:
213
214   ssh [yourUsername]@submit.lsv.uni-saarland.de
215
216   Enter your password when prompted, confirm with Enter.
217
218   I have prepared a HTCondor submitfile (' run.sub ') that should work out of the
     box with the paths and struture we have created so far.
219
220   Submit the job by using the following command (remember to put in your username
     instead of the placeholder):
221
222   condor_submit /nethome/[yourUsername]/projects/sopro25/submit-files/
     run.sub
223
224   If it does not immediately throw you any errors, it will first tell you which
     cluster it has submitted your job to (a five digit number).
225   Remember this number, as it tells you which of the logfiles correspond to that
     exact job submission.
226
227   You can then monitor your job with the command:
228
229   watch condor_q [yourUsername]
230
231   There you can see the status of all your jobs and check if they are still
     running, on hold, rejected or finished.
232
233   Wait to see when the job finishes. It usually takes up to a minute to
     complete,

depending on availability of the compute node we are running on.

234

235   Once the job finishes, exit this monitor by pressing CTRL + C once, to
interrupt the monitor process.

236

237

238

239 9. Check if logfiles were produced

240

241   When the job is finished running, navigate back to / and from there to data/
users/[yourUsername]/logs/sopro25/logfiles and check if there is any files in that
directory.

242   If everything worked correctly, there should be three files (one .log file,
one .out file and one .err file), each named something like this:

243

244   run.sh.(XXXXX).2025_month_day_somemorenumbers.log/.out/.err

245

246   The first number in parentheses (XXXXX) will be the cluster/job number from
before, the rest will be today's date and then some more numbers.

247

248   You can check these out using the cat command, like this for example:

249

250   cat run.sh.39139.2025_07_28_1753713314.out

251

252   (This was one of my output files, 39139 is the job number, the rest is the date
and probably timestamps.)

253

254   (Also, remember to make use of the autocomplete function of your terminal by
pressing Tab a lot.)

255

256   If you start typing 'cat run.sh' and then press Tab, it will autocomplete using
whats in the folder until a point at which it is unsure of what file exactly you mean.

257   The first time will be until the file extension, as you now have three files
with the same name but different file extensions in your directory.

258   Press Tab and you just have to type the extension, to access either file you
want to see.

259   If you do this again later, when you have some older and more recent logfiles
in the directory, it will autocomplete up to some point in the job number.

260   You will then have to type in the job number to specify, and you can hit Tab
again so that you don't have to type the whole thing out.

261

262

263

264 10. Read all three logfiles

265

266   If all paths are working correctly as intended, each job submission will put
out three files:

267   - a log file, containing information about the job on the cluster

268   - an error file, which in case something goes wrong during the job will
contain info about why it failed (file empty means the job finished as intended)

269   - an output file, which will contain the output of your actual project,
the code you are trying to run

270

271   I have put a small example task in the project files (you can check out the
    code in /sopro25/src/mnist.py), to make the cluster actually run something.
272
273   In the output file, you should see some output at the very bottom (way below
    the long list of packages and CUDA info).
274   It should be some loss and accuracy metrics of two tiny neural networks I have
    trained and evaluated in the code.
275
276   If you can see that, the cluster setup works.
277
278
279
280 Troubleshooting in case of errors:
281
282   If something goes wrong at any point, it is most likely a path issue.
283
284   Make sure all relevant paths used in the files exist in your LSV account
    registry, or create them if necessary.
285
286   If you encounter any errors you don't know how to fix or need help in general,
    send me a message and I will do my best to help you get things running.
287
288
289
290 Whats next?
291
292
293   - Managing dependencies
294
295   Installing new dependencies requires building and pushing a new dockerimage.
296   In the Dockerfile, lines 32 and below contain the dependencies that are
    currently in the image we're using.
297   Just add whatever you need here in the same format as the ones already there,
    and docker will install and build them into the image.
298   In the case that you choose to use a new image name, the submission file
    'run.sub' needs to be updated with the correct docker image name (line 5, parameter
    'docker_image')
299
300
301   - Getting the actual project code into the cluster
302
303   Whatever code we want to run on the cluster needs to be put in the /sopro25/src
    directory .
304   Following that, the file 'run.sh' (in /sopro25/scripts) will need to be updated
    with the new path and file name of the code to specify what code to run.
305   (line 14, under #run code; currently this specifies my example task 'mnist.py')
306
307
308   - Adapting the submitfile to request code execution on similar hardware as the
    one outlined in the Averitec shared task documents
309
310   At the moment, the submitfile is set up in a way that requests a specific
    cluster compute node and some specific hardware specs.

311   We will need to change this to more closely match the hardware outlined in the shared task (or something comparable).

312

313

314

315   Lastly, depending on your willingness to deal with this system or not, I can always take on the job of running code and managing dependencies for the project, as I have spent some time getting into it now and have set up everything on my machine and on my cluster account.

316

317   This guide should just serve as a springboard into the workflow of HTCondor and the LSV cluster for you, if you want to get into it yourself.

318

319   If you don't want the hassle, just tell me when to pull your code from the repo and run it on the cluster, if you want me to test it for you.

320

321   I have uploaded the files for this on a separate GitHub repository for now (link above), so that you can clone/pull or download it more easily to test it, but I will integrate it into the GitLab repo soon(ish).

322

323   Also, if you have the time and nerves to spare, you should check out the other README files in the sopro25 project folder. They come from the LSV HTCondor documentation, written by Marius Mosbach, which goes a bit more into detail regarding HTCondor itself, what commands to use and how to handle it in use.

324

325   If you find any errors in this document or something does not work as expected, please let me know as soon as possible.

326

327   Alex

328

329   PS: Also please do me the favour of keeping a separate list of all dependencies you use, so that I can keep the docker image up to date more easily. Thanks!

330