**Setup Lean Env on Linux VM**

1. Clone Lean source code from <https://github.com/mochiliu3000/Lean>, note that Redis related code is in “redis” branch:

*git clone https://github.com/mochiliu3000/Lean.git*

*git checkout -b redis origin/redis*

1. Follow Lean’s README to install Mono, Nuget and Python:

<https://github.com/mochiliu3000/Lean/blob/master/readme.md>

1. Follow Lean’s README to restore NuGet packages:

*nuget restore QuantConnect.Lean.sln*

1. Install Docker and use it to start RabbitMQ and Redis:
   1. <https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/>
   2. <https://hub.docker.com/_/redis/>

*docker pull redis && docker run -itd -p 6379:6379 redis*

* 1. <https://hub.docker.com/_/rabbitmq/>

*docker pull rabbitmq && docker run -itd -p 5672:5672 rabbitmq*

1. Process RU data and Put it in right folder under Lean
   1. RU raw data file is: RU.1min.2011.08.08.14.46.00.2017.02.10.14.59.00.txt
   2. Clone <https://github.com/mochiliu3000/Optimizer_Lean> and use the python script in it to process the raw data
   3. Edit the directories variables under Optimizer\_Lean/scripts:

*vim Optimizer\_Lean/scripts/txt\_to\_csv.py*

txt\_file\_dir = "./RU.1min.2011.08.08.14.46.00.2017.02.10.14.59.00.txt"

csv\_file\_dir = "./csv"

zip\_file\_dir = "./zip"

***Line 26, please change to "\_spy\_minute\_trade.csv"***

Hence need to put the txt file under Optimizer\_Lean/scripts, and need to create 2 folders: csv and zip

* 1. Make sure python 2.7 is installed then Run the script. All the zip files will be generated under zip folder:

*python txt\_to\_csv.py*

* 1. Rename and Copy the zip folder to Lean data folder:

*mv Lean/Data/equity/usa/minute/spy Lean/Data/equity/usa/minute/spy\_bak*

*mkdir Lean/Data/equity/usa/minute/spy*

*cp Optimizer\_Lean/scripts/zip/\* Lean/Data/equity/usa/minute/spy*

1. Config Lean: (Make sure the following lines are not in comment)

*vim Launcher/config.json*

"algorithm-type-name": "ShortStrategyAlgorithm" or "algorithm-type-name": "ParameterAlgorithm",

"log-handler": "QuantConnect.Logging.ConsoleLogHandler",

"data-provider": "QuantConnect.Lean.Engine.DataFeeds.RedisDataProvider",

1. Compile and Run Lean:

*xbuild QuantConnect.Lean.sln (Build debug version)*

*xbuild /p:Configuration=Release QuantConnect.Lean.sln (Build release version)*

*cd Launcher/bin/Debug && mono ./QuantConnect.Lean.Launcher.exe*

1. To check Redis data:

*docker exec -it <Redis Container ID> bash*

*redis-cli*

*keys \**

1. To run Rabbitmq commands:

*docker exec -it <Rabbitmq Container ID> bash*

*rabbitmqctl list\_queues*

*rabbitmqctl stop\_app*

*rabbitmqctl reset*

*rabbitmqctl start\_app*

*FLUSHALL*

1. Lean Trading Visulization
   1. Change config.json to use "backtesting-desktop"
   2. Under "backtesting-desktop" section in config.json, it triggers: (Change the folder from Debug to Release)

"desktop-exe": "../../../UserInterface/bin/Release/QuantConnect.Views.exe"

* 1. Lean requires a x-server to show the visualization

*sudo apt-get install xorg openbox*

*export DISPLAY=:0.0* ***(This need double check)***

* 1. And the source code is under: Lean\userinterface\winforms\leanwinform.cs. You can directly run it or use run it from Launcher

*cd Lean/UserInterface/bin/Release && mono ./QuantConnect.Views.exe 1234*

*cd Lean/Launcher/bin/Debug && mono ./QuantConnect.Lean.Launcher.exe*