

Record of Installation M1 Mac mini Monterey

22 Oct 2021

Press and hold Power until Options, Select Options continue

Authenticate known admin user

Disk Utility Continue

Macintosh HD Erase Format APFS

Macintosh HD-Data Erase Format APFS

Automatically connected to known wifi

Authenticate Apple ID

Exit to Recovery Utilities

Reinstall macOS Big Sur Continue Agree Agree

Select Macintosh HD Continue

Country- United Kingdom

Accessibility - Not Now

Join wifi network

Data & Privacy - Continue

Migration Assistant - Not Now

Apple ID - Sign In

Terms and Conditions - Tick I have read, Agree

Create a Computer Account - Details, Continue

Find My - Continue

Enable Location Services - Continue

Express Set Up - Continue

Analytics - Continue

Screen Time - Continue

Siri (enabled) - Continue

Improve Siri & Dictation - Share Audio Recordings, Continue

FileVault Disk Encryption - both ticked, Continue

Choose Your Look - Dark, Continue

Calendar would like to use your current location - OK

About this mac - 11.6 Serial No XXXX, Software Update

Update Now Device Support update, Safari 15.0

Settings - Energy Saver Prevent from sleeping display off, start up after power failure

Settings iCloud unticked Photos, contacts, News, Stocks

Connected to eduroam

Settings - Sharing tick Screen Sharing, Remote login. Added user

Installed macOS Big Sur 11.6.1

Installed macOS Monterey 12.0.1 from Settings Software Update.

Continue Agree Agree Continue

All your files in iCloud, Store files ticked Continue

Downloaded Xcode 13.1 from App Store

Requires Password for additional purchases - Always

Open Xcode Agree, To run Intel-based Install Rosetta, Open test project

Git login details

In terminal xcode-select - -install

From code.visualstudio.com downloaded VSCode 1.60.0 Apple Silicon to Applications

My extensions added to VSCode

From Desktop.github.com installed Github Desktop for macOS Apple Silicon

From julialang.org downloaded Julia 1.6.2 (x86) and 1.7.0-rc2

From nodejs.org other downloads nodejs .pkg for 64-bit-ARM64 17.0.1, Run package install from Downloads

Downloaded Wolfram Engine for Developers macOS ARM

In Downloads executed dmg WolframEngine_12.3.1_MAC-ARM_DLM.dmg

2 steps drag Wolfram Engine to Applications and Install script from package

In Applications run Wolframscript - enter wolfram login credentials

In /Applications/Utilities copy Terminal and Paste Terminal, Rename one of them

Terminal_x86, File get info and tick Open with Rosetta for Terminal_x86

In Terminal (not _x86):

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

```
echo 'eval "$(/opt/homebrew/bin/brew shellenv)'" >> /Users/YOURUSER/.zprofile
```

```
eval "$(/opt/homebrew/bin/brew shellenv)"
```

In Github Desktop login clone URL <https://github.com/WolframResearch/WolframLanguageForJupyter.git>

In Terminal_x86:

```
/usr/bin/python3 -m venv x389
```

```
source x389/bin/activate
```

```
pip install --upgrade pip setuptools wheel
```

```
pip install jupyterlab
```

```
pip install numpy sympy matplotlib pandas plotly bokeh scipy seaborn
```

```
statsmodels octave-kernel ipympl pyvista ipyvtklink vispy jupyter_rfb ipywidgets
```

```
watermark scilab-kernel gnuplot-kernel mdanalysis stochastic mdanalysis-tests
```

```
nglview
```

```
jupyter-nbextension enable nglview --py --sys-prefix
```

```
python -m gnuplot_kernel install - -user
```

```
In -s /Applications/Julia-1.6.app/Contents/Resources/julia/bin/julia /Users/YOURUSER/x389/bin/julia
```

```
julia
```

```
using Pkg
```

```
Pkg.add("IJulia")
```

```
Pkg.add("Plots")
```

```
Pkg.add("DifferentialEquations")
```

```
Pkg.add("Plotly")
```

```
Pkg.add("Pluto")
```

```
Pkg.add("Makie")
```

```
Pkg.add("AbstractPlotting")
```

```
Pkg.add("Molly")
```

```
Pkg.add("DynamicalSystems")
```

```
Pkg.add("GLMakie")
```

```
exit()
```

Open /Applications/WolframScript and complete credentials

```
cd Documents/GitHub/WolframLanguageForJupyter
export WOLFRAMSCRIPT_KERNELPATH=/Applications/Wolfram\ Engine.app/
Contents/MacOS/WolframKernel
./configure-jupyter.wls add
```

In Terminal:

```
brew install qt
brew install octave
Run with octave - -gui
brew install wxmaxima
brew install r
brew install ffmpeg
brew install cmake
brew install code-server
mkdir ~/.config/code-server
Edit ~/.config/code-server/config.yaml
bind-addr:192.168.1.16:8081
auth:password
password:mightbe
cert:false
code-server &
```

From scilab.org downloaded Scilab 6.1.1-MacOS 64bits dmg

Installed from dmg with install JRE option

Scilab warning on Monterey selected Try anyway

```
In -s /Applications/scilab-6.1.1/Contents/bin/scilab-cli /Users/YOURUSER/x389/
bin/scilab-cli
```

From xquartz.org downloaded and installed XQuartz-2.8.1.dmg

Ensure iCloud Drive , options has Desktop and Documents folder ticked

```
In -s /Users/YOURUSER/Documents/Developer /Users/YOURUSER/Developer
```

In Terminal_x86 In x389 environment:

```
export JUPYTER_CONFIG_DIR=~/.x389
export WOLFRAMSCRIPT_KERNELPATH=/Applications/Wolfram\ Engine.app/
Contents/MacOS/WolframKernel
export SCILAB_EXECUTABLE=/Applications/scilab-6.1.1.app/Contents/bin/scilab-
adv-cli
jupyter notebook - -generate-config
jupyter notebook password
jupyter-lab --ip=XXXX - -port=8888 --no-browser --notebook-dir=~/.Developer/
solutions
```

From <https://github.com/conda-forge/miniforge> downloaded Miniforge3-MacOSX-arm64

In Terminal (not x86) cd Downloads

```
bash Miniforge3-MacOSX-arm64.sh yes Enter yes
```

Quit Terminal and restart

If forget to get script to init then do:

```
/Users/YOURUSER/miniforge3/bin/conda init zsh
```

```
conda config - -set auto_activate_base false
Close terminal and re enter
/Users/YOURUSER/miniforge3/bin/conda create -n a396 python=3.9.6
/Users/YOURUSER/miniforge3/bin/conda activate a396
which python shows /Users/YOURUSERminiforge3/envs/a396/bin/python
python -V shows Python 3.9.6
which pip shows /Users/YOURUSER/miniforge3/envs/a396/bin/pip
conda install -c conda-forge jupyterlab
conda install -c conda-forge numpy
conda install -c conda-forge sympy
conda install -c conda-forge matplotlib
conda install -c conda-forge pandas
conda install -c conda-forge plotly
conda install -c conda-forge bokeh
conda install -c conda-forge scipy
conda install -c conda-forge seaborn
conda install -c conda-forge ipympl
conda install -c conda-forge pyvista
conda install -c conda-forge ipyvtklink
conda install -c conda-forge vispy
conda install -c conda-forge watermark
conda install -c conda-forge lfortran
conda install -c conda-forge stochastic
conda install -c conda-forge nglview
conda install -c conda-forge astropy plasmapy
jupyter-nbextension enable nglview --py --sys-prefix
pip install octave-kernel scilab-kernel gnuplot-kernel mdanalysis mdanalysis-tests
ln -s /Applications/Julia-1.7.app/Contents/Resources/julia/bin/julia /Users/
YOURUSER/miniforge3/envs/a396/bin/julia
ln -s /Applications/scilab-6.1.1/Contents/bin/scilab-cli /Users/YOURUSER/
miniforge3/envs/a396/bin/scilab-cli
R (capital)
install.packages('IRkernel')
75
install.packages('plotly')
IRkernel::installspec()
quit()
jupyter labextension install @techrah/text-shortcuts
jupyter labextension list
jupyter kernelspec list
jupyter nbextension list
mv .jupyter miniforge3/envs/a396
export JUPYTER_CONFIG_DIR=~/.miniforge3/envs/a396
export WOLFRAMSCRIPT_KERNELPATH=/Applications/Wolfram\ Engine.app/
Contents/MacOS/WolframKernel
export SCILAB_EXECUTABLE=/Applications/scilab-6.1.1.app/Contents/bin/scilab-
adv-cli
```

```
In -s /Applications/Julia-1.7.app/Contents/Resources/julia/bin/julia /opt/  
homebrew/bin/julia17  
julia17  
using Pkg  
Pkg.add("IJulia")  
Pkg.add("Plots")  
Pkg.add("DifferentialEquations")  
Pkg.add("Plotly")  
Pkg.add("Pluto")  
Pkg.add("Makie")  
Pkg.add("AbstractPlotting")  
Pkg.add("Molly") *  
Pkg.add("DynamicalSystems")  
Pkg.add("GLMakie")  
A star * indicates package errors in 1.7 rc2 kernel (not ARM compatible yet?) Use  
1.6.2 kernel which runs with Rosetta  
using Pluto  
Pluto.run(host="0.0.0.0")  
Pluto errors Monterey x86 and arm versions  
exit()  
jupyter notebook - -generate-config  
jupyter notebook password  
jupyter-lab build  
jupyter-lab - -no-browser - -ip=192.168.1.16 - -port=8889 - -notebook-dir=~/  
Developer/solutions  
Verify working with test notebook Then shutdown
```

In VScode Terminal conda activate base pip install fortran-language-server

```
mkdir minlammps  
git clone -b unstable https://github.com/lammps/lammps.git minlammps  
cd minlammps  
git checkout  
Standard cmake build and pylammps build  
mkdir build  
cd build  
cmake ../cmake  
make  
cd ..  
mkdir build-shared  
cd build-shared  
cmake ../cmake -DPKG_MOLECULE=yes -DLAMMPS_EXCEPTIONS=yes  
-DBUILD_LIB=yes -DBUILD_SHARED_LIBS=yes  
make
```