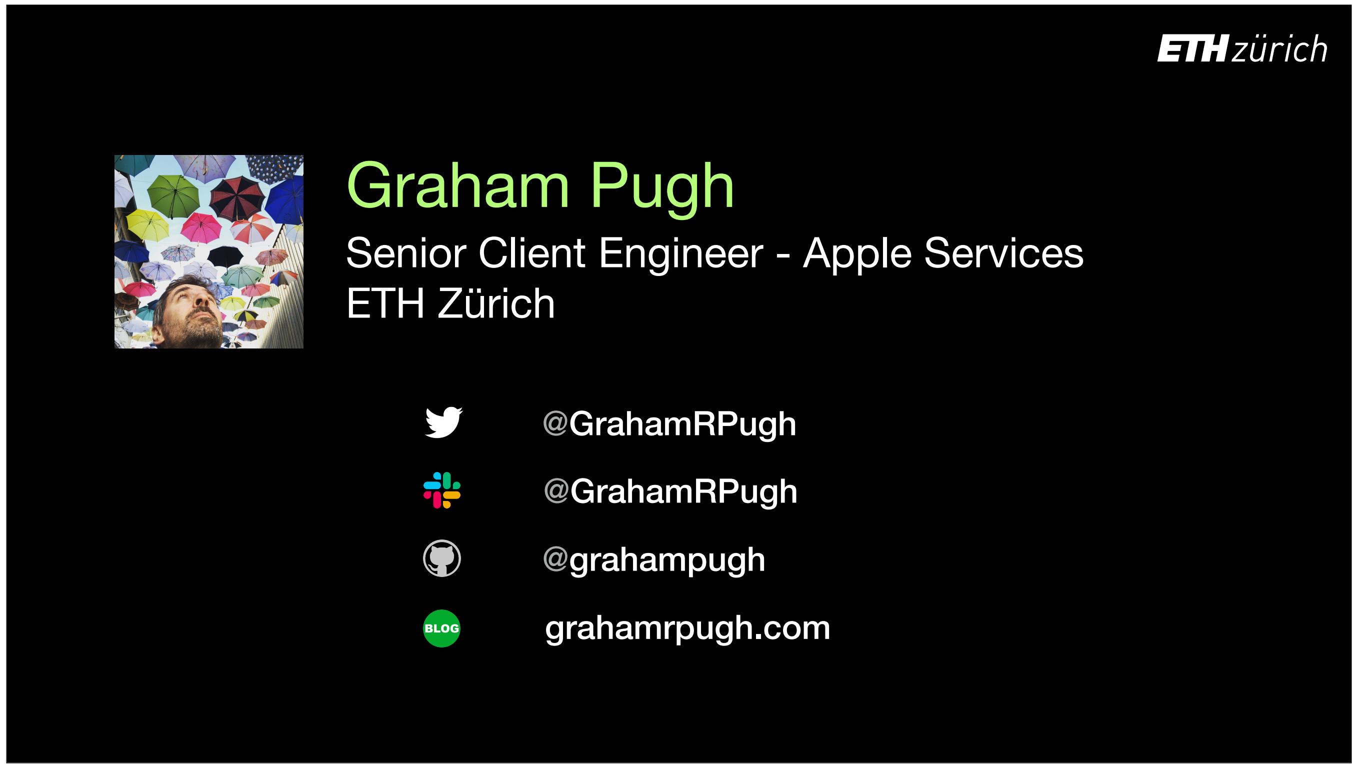
UP NEXT

erase-install: Download, Erase and/or Reinstall

GRAHAM PUGH



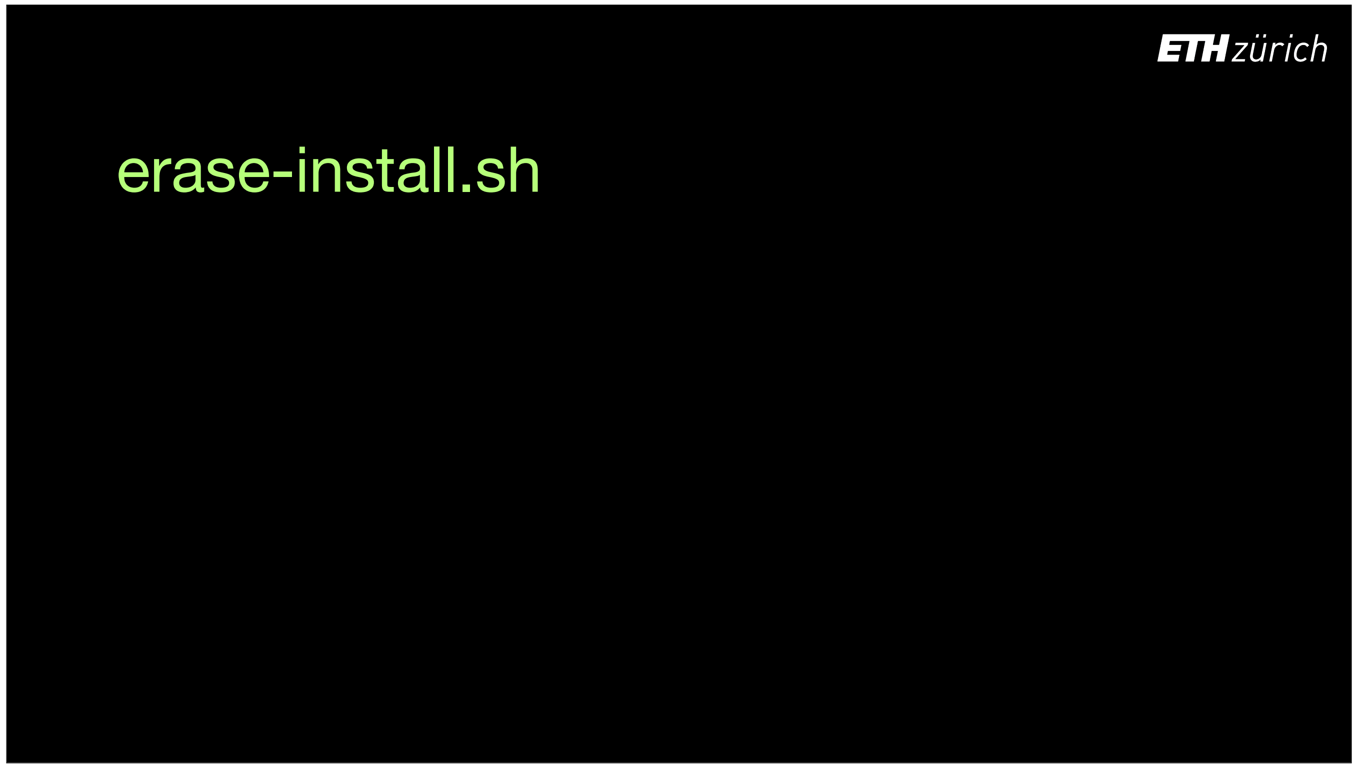
ETH zürich

Graham Pugh

Senior Client Engineer - Apple Services
ETH Zürich

	@GrahamRPugh
	@GrahamRPugh
	@grahampugh
	grahamrpugh.com

My name is Graham, I'm from the UK, and I work in Switzerland at a university called ETH Zürich.


erase-install.sh

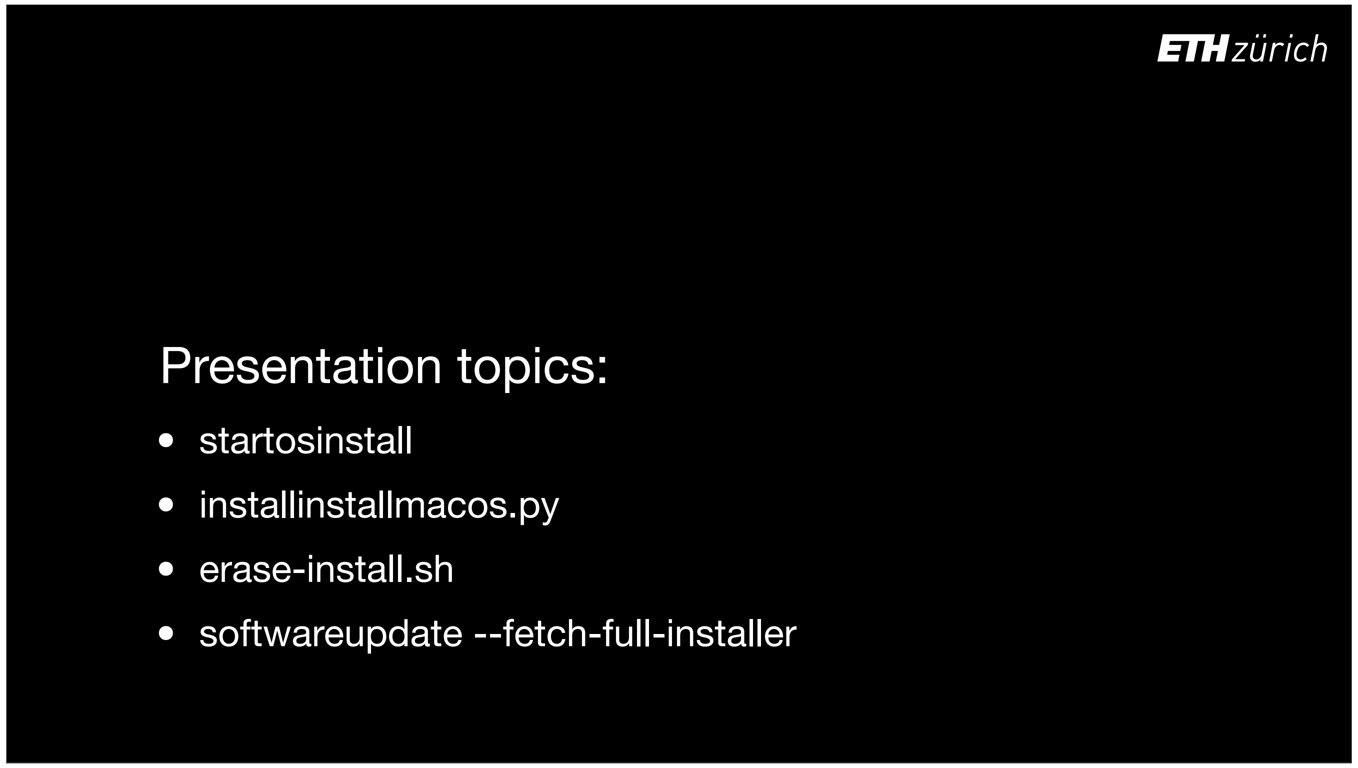
I'm going to talk about a project called erase-install, which you can use to automate the deployment and installation of macOS with a single click in your Self Service kiosk or from a single command.
It's been around for two years and to my surprise, got a lot of interest, some great additions through pull requests, and has 150 stars in GitHub.

They key thing I'd like to bring across today is that the name of this script is a bit deceptive. It doesn't just do erase-install. It should probably be called something like...
[click]

ETH zürich

download-and-or-erase-and-or-reinstall-and-or-upgrade-macos.sh

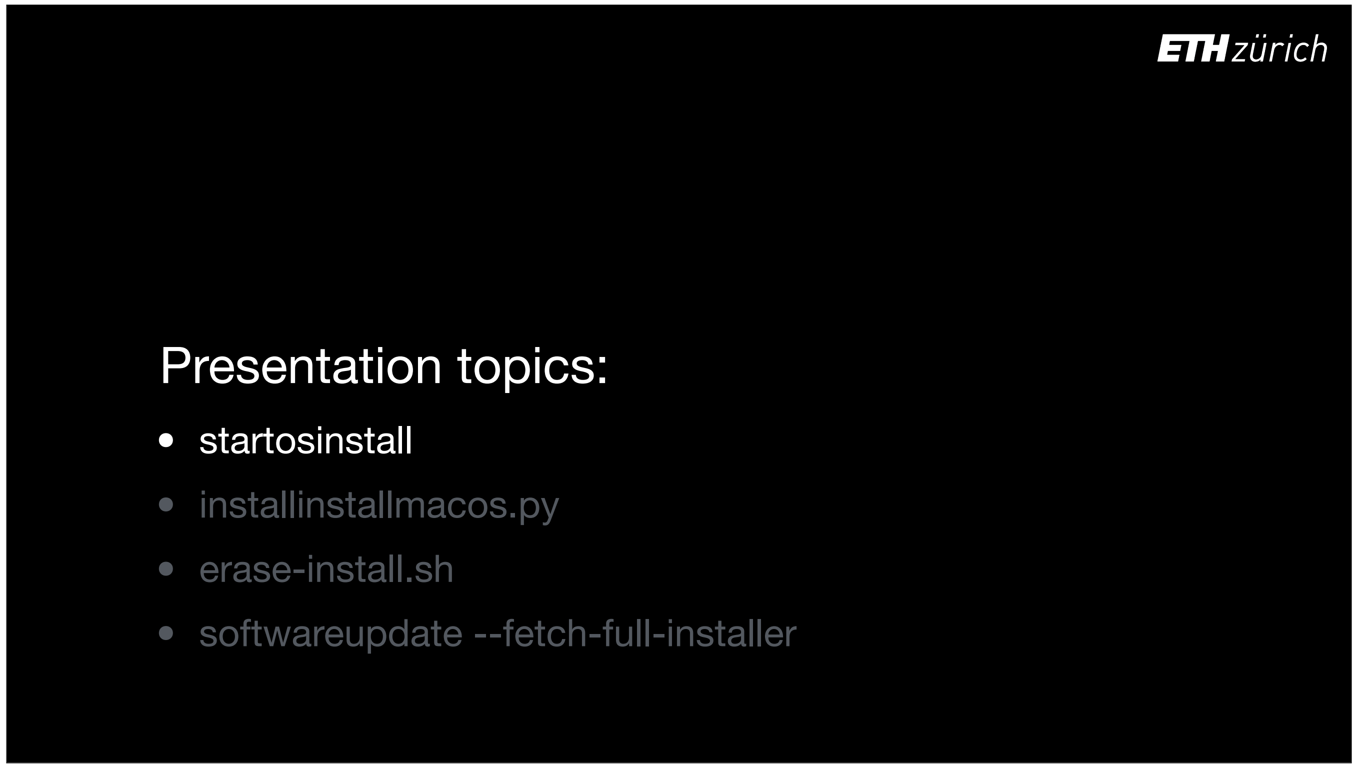
..like this. But that's a bit annoying to type.
It's a bit of a Swiss Army knife:
As well as a one-click "Erase all Contents and Settings" tool, it is also used for downloading specific OS builds for manual installation, and also for one-click in-place upgrades and emergency reinstalls.


ETH zürich

Presentation topics:

- startosinstall
- installinstallmacos.py
- erase-install.sh
- softwareupdate --fetch-full-installer

To explain how it works, I'll explain the component parts, namely startosinstall and installinstallmacos;
Then the wrapper script itself;
and we'll have a quick look at fetch-full-installer, which may become more important in future.



ETH zürich

Presentation topics:

- startosinstall
- installinstallmacos.py
- erase-install.sh
- softwareupdate --fetch-full-installer

Starting with startosinstall.

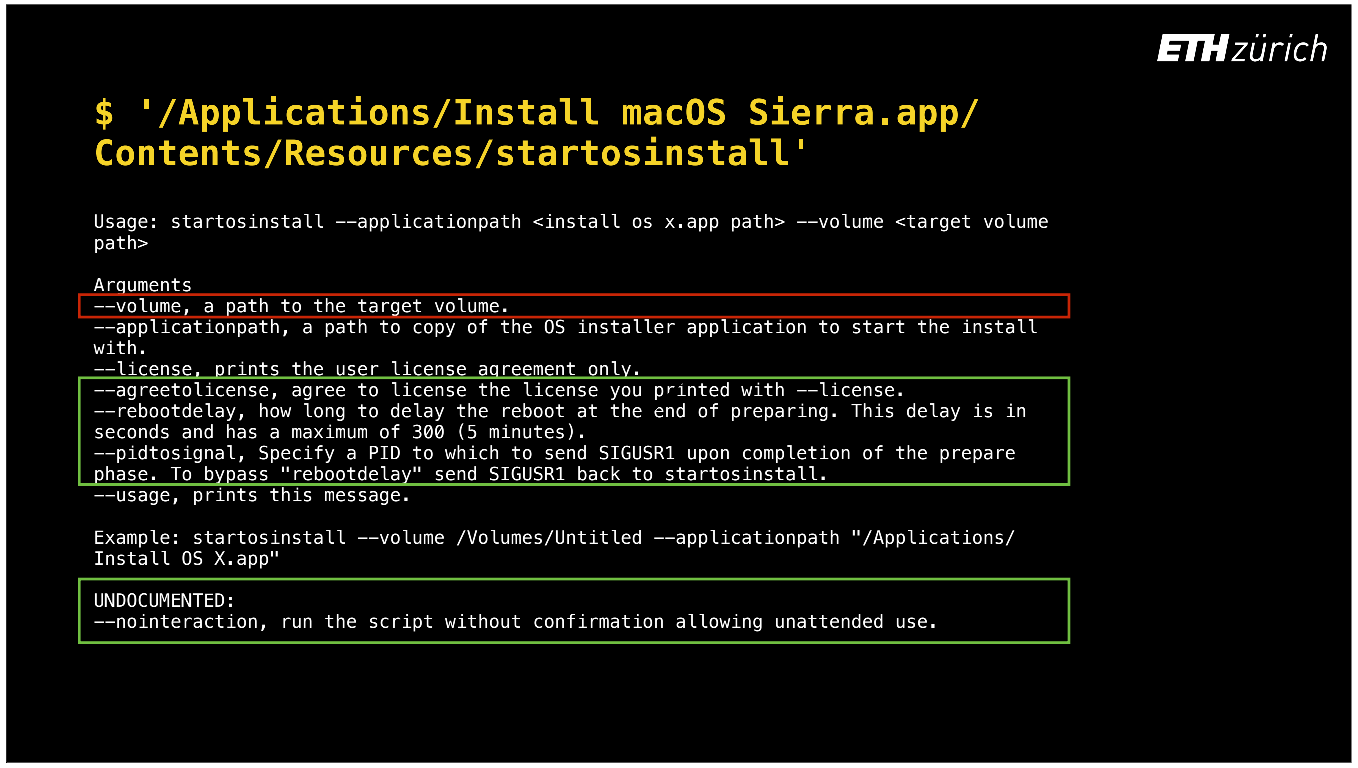
Startosinstall is a command line binary that is included with macOS installer apps since El Capitan.

It allows you to perform a macOS installation directly from the command line.

There have been improvements to the tool with every macOS release.

$ '/Applications/Install OS X El Capitan.app/Contents/Resources/startosinstall'

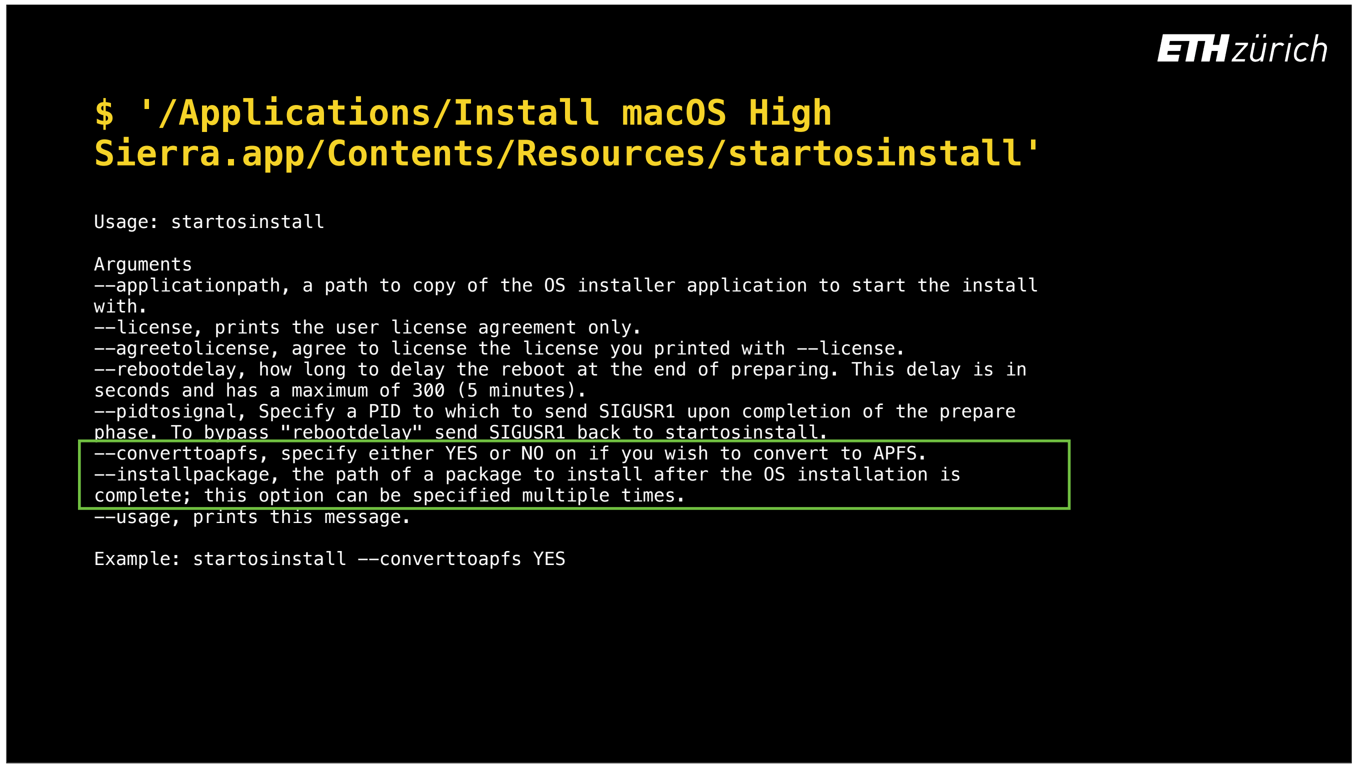
```
Usage: startosinstall --applicationpath <install os x.app path> --volume <target volume
path>

Arguments
--volume, a path to the target volume.
--applicationpath, a path to copy of the OS installer application to start the install
with.
--license, prints the user license agreement only.
--usage, prints this message.

Example: startosinstall --volume /Volumes/Untitled --applicationpath "/Applications/
Install OS X.app"
```



$ '/Applications/Install macOS Sierra.app/Contents/Resources/startosinstall'

```
Usage: startosinstall --applicationpath <install os x.app path> --volume <target volume
path>

Arguments
--volume, a path to the target volume.
--applicationpath, a path to copy of the OS installer application to start the install
with.
--license, prints the user license agreement only.
--agreetolicense, agree to license the license you printed with --license.
--rebootdelay, how long to delay the reboot at the end of preparing. This delay is in
seconds and has a maximum of 300 (5 minutes).
--pidtosignal, Specify a PID to which to send SIGUSR1 upon completion of the prepare
phase. To bypass "rebootdelay" send SIGUSR1 back to startosinstall.
--usage, prints this message.

Example: startosinstall --volume /Volumes/Untitled --applicationpath "/Applications/
Install OS X.app"

UNDOCUMENTED:
--nointeraction, run the script without confirmation allowing unattended use.
```

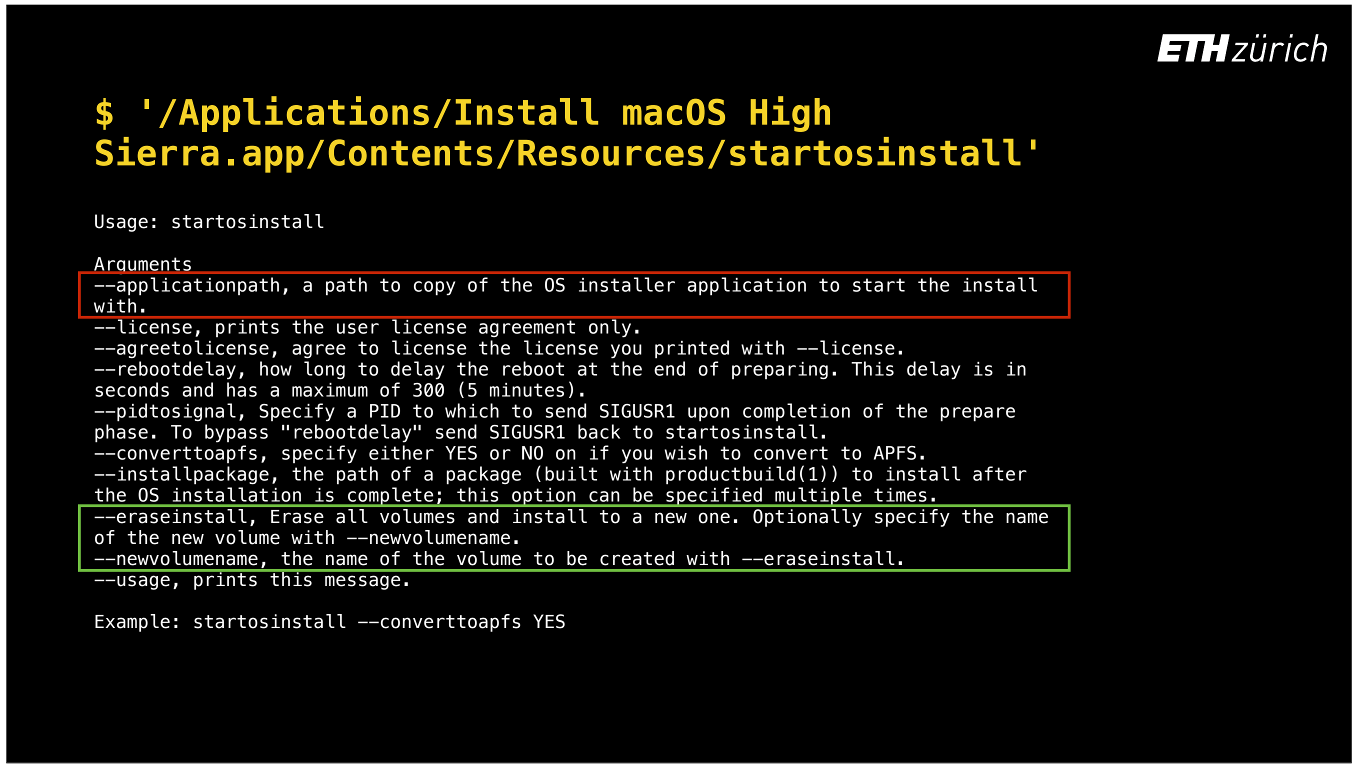
Most importantly, in Sierra we became able to use the command unattended, including a still undocumented flag called 'nointeraction'.



$ '/Applications/Install macOS High Sierra.app/Contents/Resources/startosinstall'

```
Usage: startosinstall

Arguments
--applicationpath, a path to copy of the OS installer application to start the install
with.
--license, prints the user license agreement only.
--agreetolicense, agree to license the license you printed with --license.
--rebootdelay, how long to delay the reboot at the end of preparing. This delay is in
seconds and has a maximum of 300 (5 minutes).
--pidtosignal, Specify a PID to which to send SIGUSR1 upon completion of the prepare
phase. To bypass "rebootdelay" send SIGUSR1 back to startosinstall.
--converttoapfs, specify either YES or NO on if you wish to convert to APFS.
--installpackage, the path of a package to install after the OS installation is
complete; this option can be specified multiple times.
--usage, prints this message.

Example: startosinstall --converttoapfs YES
```

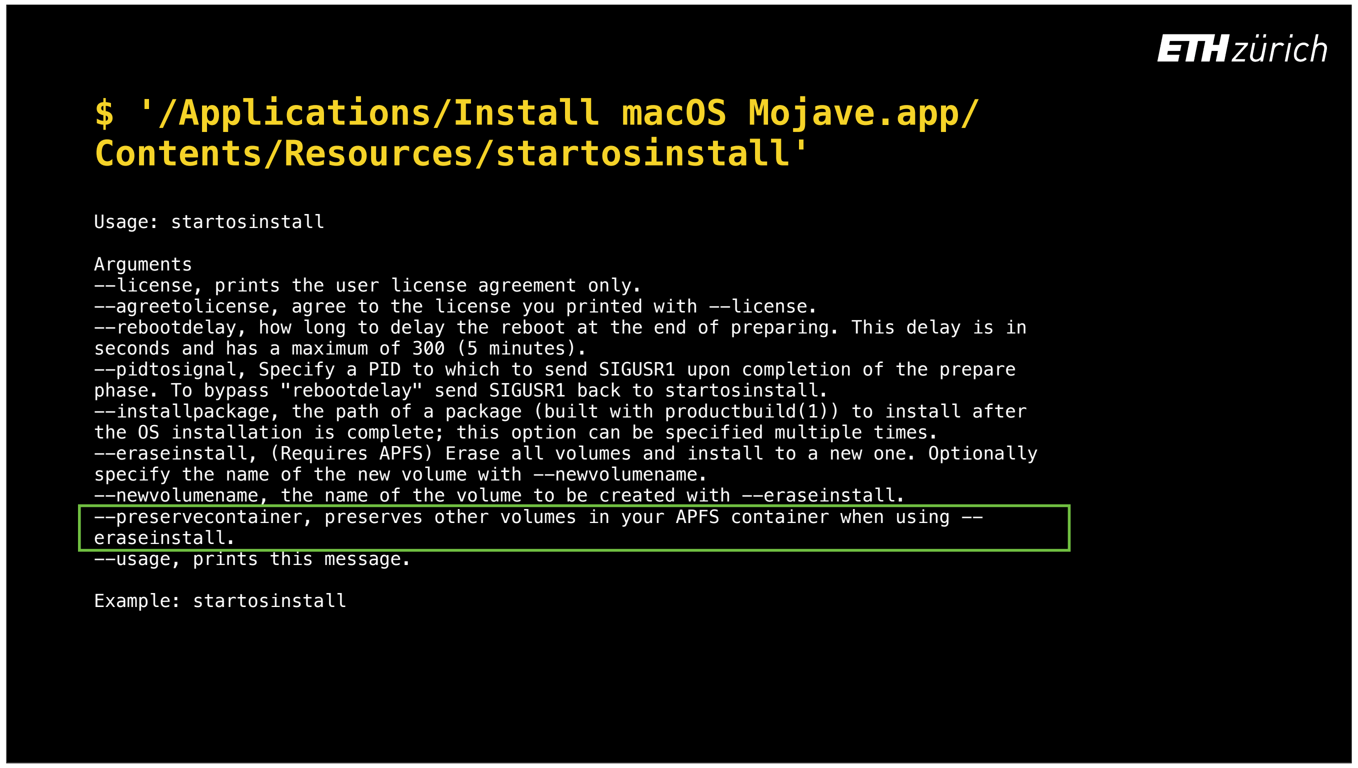
In High Sierra we got the `installpackage` flags, so we could install signed packages alongside macOS



$ '/Applications/Install macOS High Sierra.app/Contents/Resources/startosinstall'

```
Usage: startosinstall

Arguments
--applicationpath, a path to copy of the OS installer application to start the install
with.
--license, prints the user license agreement only.
--agreetolicense, agree to license the license you printed with --license.
--rebootdelay, how long to delay the reboot at the end of preparing. This delay is in
seconds and has a maximum of 300 (5 minutes).
--pidtosignal, Specify a PID to which to send SIGUSR1 upon completion of the prepare
phase. To bypass "rebootdelay" send SIGUSR1 back to startosinstall.
--converttoapfs, specify either YES or NO on if you wish to convert to APFS.
--installpackage, the path of a package (built with productbuild(1)) to install after
the OS installation is complete; this option can be specified multiple times.
--eraseinstall, Erase all volumes and install to a new one. Optionally specify the name
of the new volume with --newvolumename.
--newvolumename, the name of the volume to be created with --eraseinstall.
--usage, prints this message.

Example: startosinstall --converttoapfs YES
```

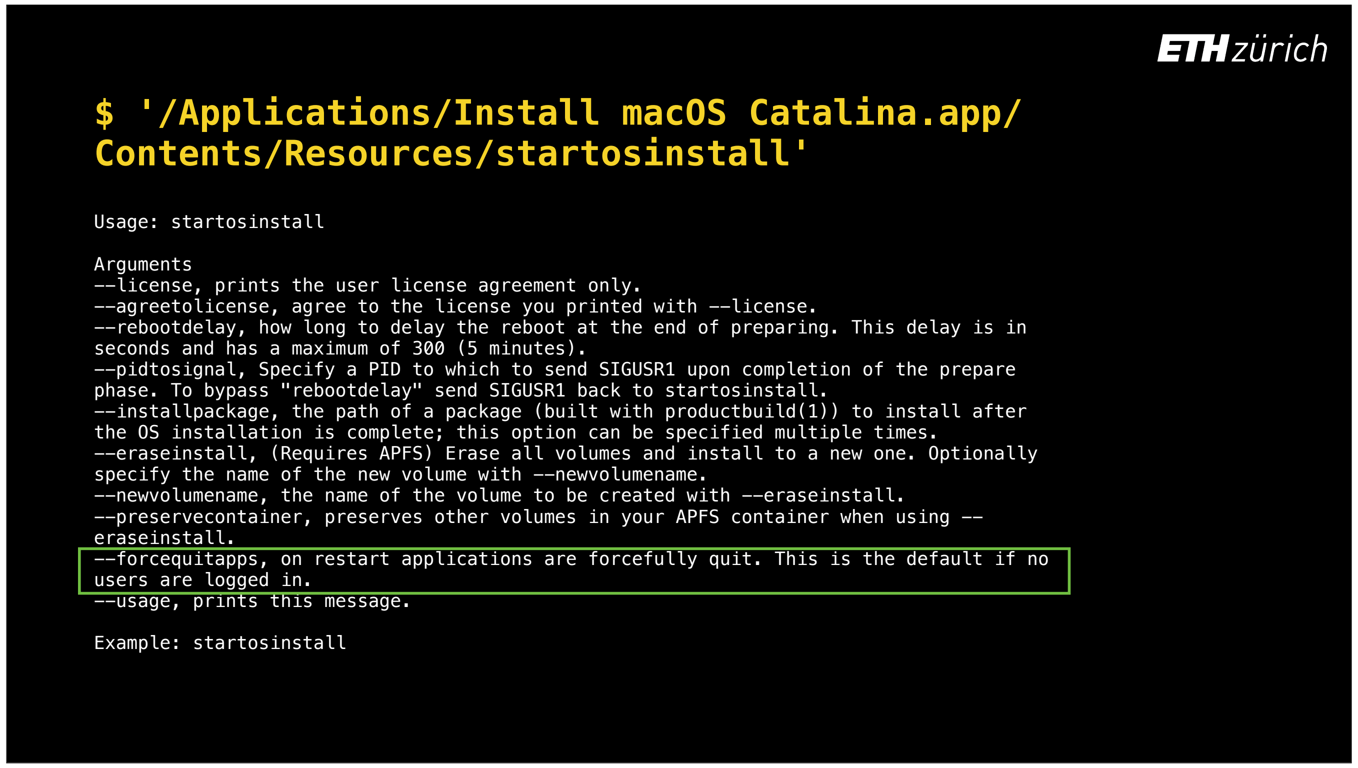
In 10.13.4 we got the 'eraseinstall' flag for the first time.



```
$ '/Applications/Install macOS Mojave.app/
Contents/Resources/startosinstall'
```

```
Usage: startosinstall

Arguments
--license, prints the user license agreement only.
--agreetolicense, agree to the license you printed with --license.
--rebootdelay, how long to delay the reboot at the end of preparing. This delay is in
seconds and has a maximum of 300 (5 minutes).
--pidtosignal, Specify a PID to which to send SIGUSR1 upon completion of the prepare
phase. To bypass "rebootdelay" send SIGUSR1 back to startosinstall.
--installpackage, the path of a package (built with productbuild(1)) to install after
the OS installation is complete; this option can be specified multiple times.
--eraseinstall, (Requires APFS) Erase all volumes and install to a new one. Optionally
specify the name of the new volume with --newvolumename.
--newvolumename, the name of the volume to be created with --eraseinstall.
--preservecontainer, preserves other volumes in your APFS container when using --
eraseinstall.
--usage, prints this message.

Example: startosinstall
```

In 10.14 we got the 'preservecontainer' flag which allows for using 'eraseinstall' without deleting all containers, so for example you can retain a separate data container.

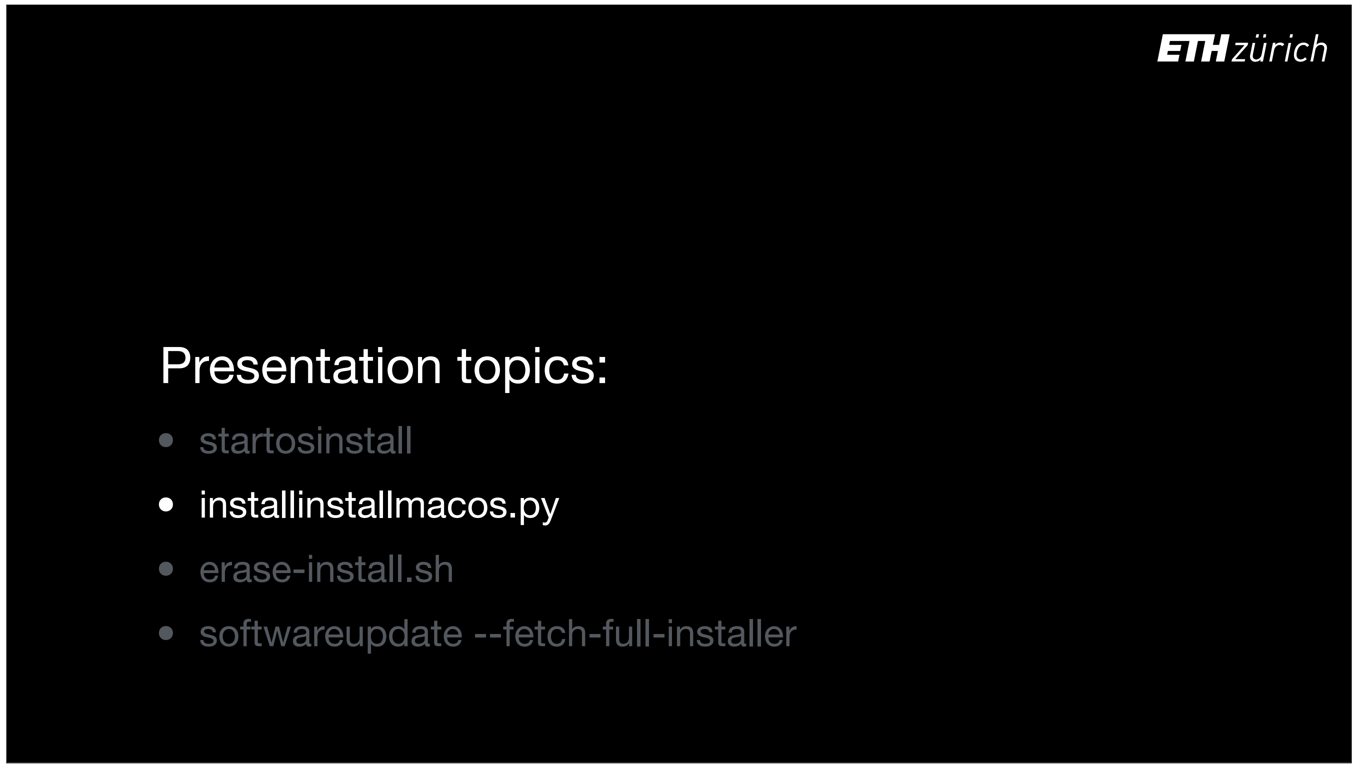


$ '/Applications/Install macOS Catalina.app/Contents/Resources/startosinstall'

```
Usage: startosinstall

Arguments
--license, prints the user license agreement only.
--agreetolicense, agree to the license you printed with --license.
--rebootdelay, how long to delay the reboot at the end of preparing. This delay is in
seconds and has a maximum of 300 (5 minutes).
--pidtosignal, Specify a PID to which to send SIGUSR1 upon completion of the prepare
phase. To bypass "rebootdelay" send SIGUSR1 back to startosinstall.
--installpackage, the path of a package (built with productbuild(1)) to install after
the OS installation is complete; this option can be specified multiple times.
--eraseinstall, (Requires APFS) Erase all volumes and install to a new one. Optionally
specify the name of the new volume with --newvolumename.
--newvolumename, the name of the volume to be created with --eraseinstall.
--preservecontainer, preserves other volumes in your APFS container when using --
eraseinstall.
--forcequitapps, on restart applications are forcefully quit. This is the default if no
users are logged in.
--usage, prints this message.

Example: startosinstall
```

And In 10.15 we got the 'forcequitapps' flag to make restarting the computer during reinstallation more reliable.

Together, these changes have made `startosinstall` increasingly reliable to use from a script.

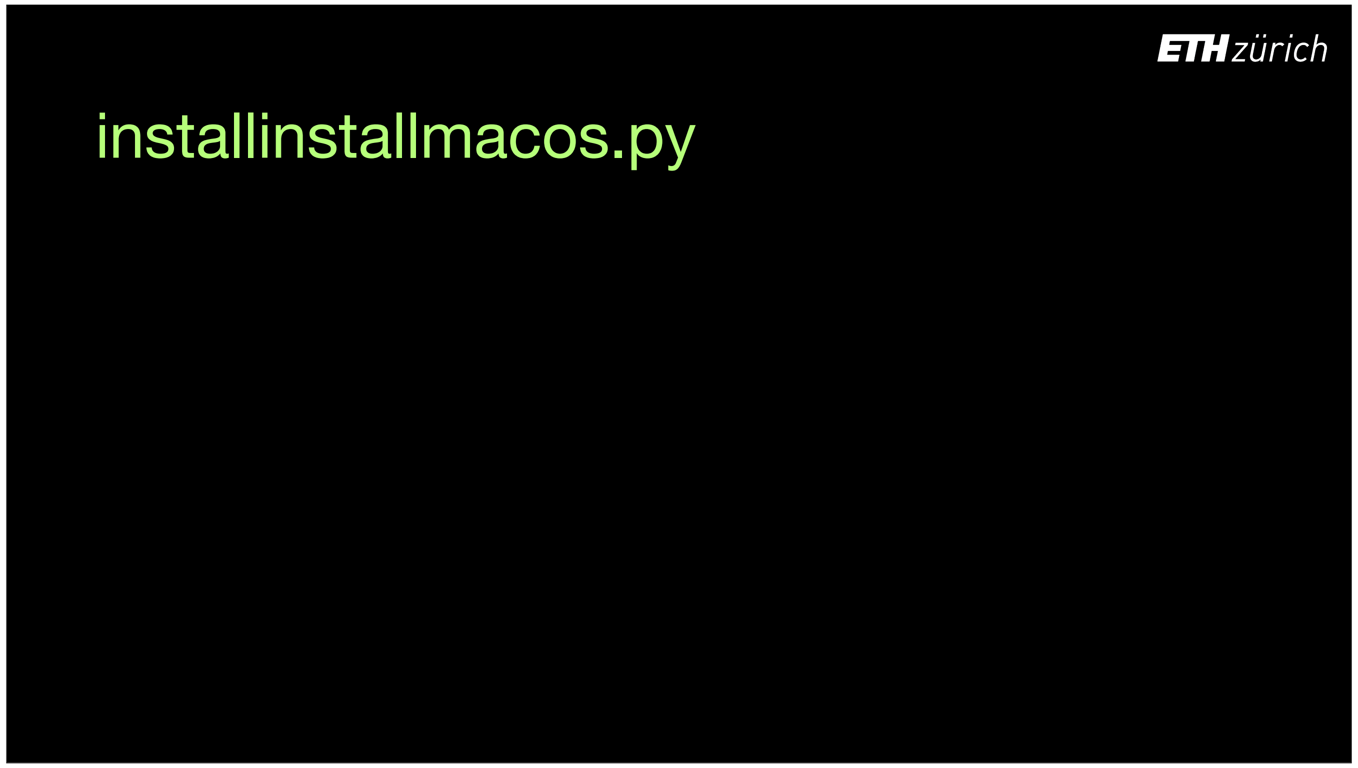

ETH zürich

Presentation topics:

- startosinstall
- installinstallmacos.py
- erase-install.sh
- softwareupdate --fetch-full-installer

Of course to use startosinstall we need to get the installer onto the system in the first place.

A good way to do that is using Greg Neagle's installinstallmacos.py script.



installinstallmacos.py

This is a python script, which you can get from Munki's GitHub.
[click] It checks Apple's softwareupdate catalogs and presents available macOS installers for download.
You can specify different seed programs such as the Developer or AppleSeed programs for obtaining betas.
It optionally places the downloaded updates inside a DMG.

installinstallmacos.py

github.com/munki/macadmin-scripts

ETH zürich

installinstallmacos.py

github.com/munki/macadmin-scripts

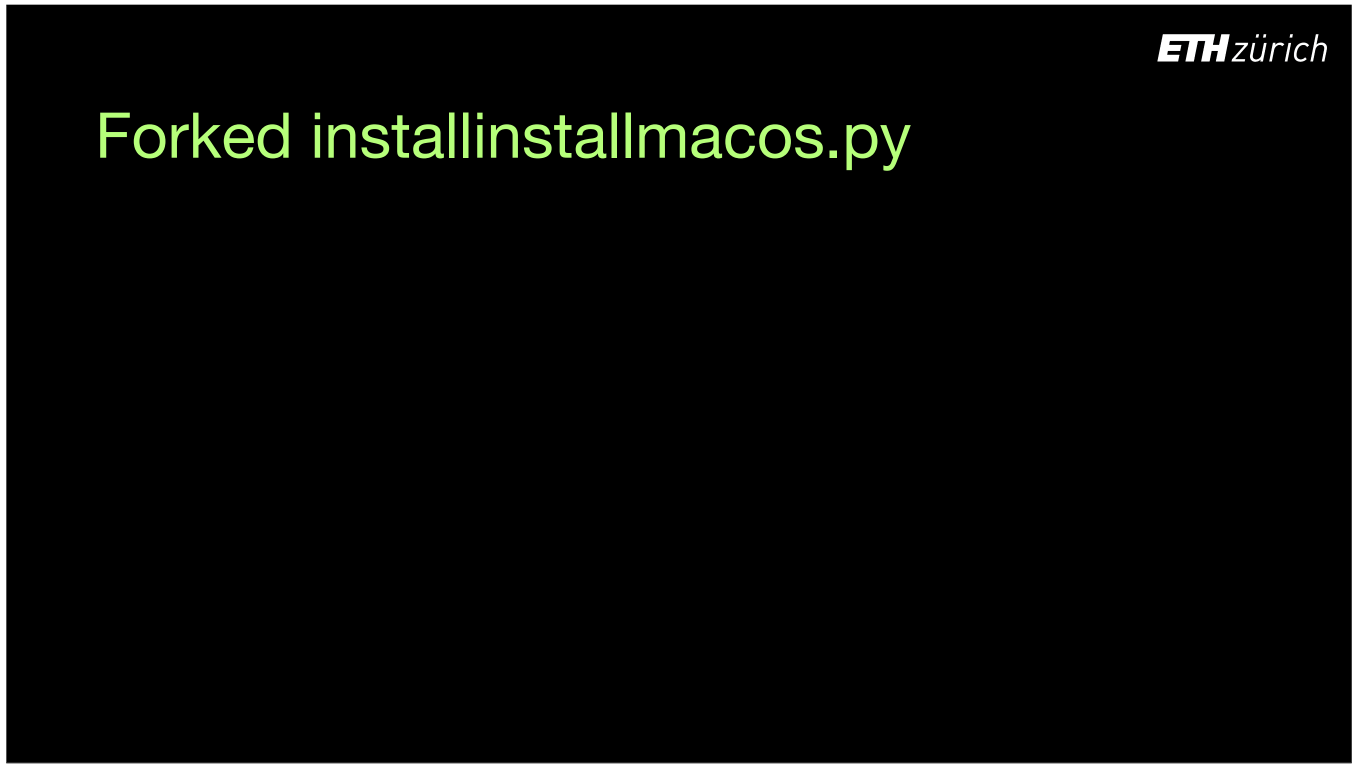
```
$ sudo python installinstallmacos.py --help
```

```
usage: installinstallmacos.py [-h] [--seedprogram SEEDPROGRAM]
                              [--catalogurl CATALOGURL]
                              [--workdir path_to_working_dir] [--compress]
                              [--raw] [--ignore-cache]

optional arguments:
  -h, --help            show this help message and exit
  --seedprogram SEEDPROGRAM
  --catalogurl CATALOGURL
  --workdir path_to_working_dir
  --compress
  --raw
  --ignore-cache
```

This is the output from installinstallmacos.
After presenting the available installers, you are prompted to choose one.
After choosing, it proceeds to download.
You cannot automate the choice.

```
$ sudo python installinstallmacos.py
```

```
Downloading https://swscan.apple.com/content/catalogs/others/
index-10.15-10.14-10.13-10.12-10.11-10.10-10.9-mountainlion-lion-snowleopard-
leopard.merged-1.sucatalog...
Downloading http://swcdn.apple.com/content/downloads/34/54/041-88800-A_HLMBDM42FL/
anrmoj880qkj0lbybqm0c3830p70nawjrv/InstallAssistantAuto.smd...
Downloading https://swdist.apple.com/content/downloads/34/54/041-88800-A_HLMBDM42FL/
anrmoj880qkj0lbybqm0c3830p70nawjrv/041-88800.English.dist...
Downloading http://swcdn.apple.com/content/downloads/17/32/061-26589-A_8GJTCGY9PC/
25fhcu905eta7wau7aoafu8rvdm7k1j4el/InstallAssistantAuto.smd...
Downloading https://swdist.apple.com/content/downloads/17/32/061-26589-A_8GJTCGY9PC/
25fhcu905eta7wau7aoafu8rvdm7k1j4el/061-26589.English.dist...
Downloading http://swcdn.apple.com/content/downloads/35/40/061-96006-A_3D42K49AN3/
r1513y9zscckrsmy018hbcryom8nq7gg1t/InstallAssistantAuto.smd...
Downloading https://swdist.apple.com/content/downloads/35/40/061-96006-A_3D42K49AN3/
r1513y9zscckrsmy018hbcryom8nq7gg1t/061-96006.English.dist...
 #    ProductID    Version    Build   Post Date  Title
 1    061-96006    10.15.4   19E287  2020-04-08  macOS Catalina
 2    041-91758    10.13.6    17G66  2019-10-19  macOS High Sierra
 3    061-86291    10.15.3  19D2064  2020-03-23  macOS Catalina
 4    061-26589    10.14.6   18G103  2019-10-14  macOS Mojave
 5    041-88800    10.14.4  18E2034  2019-10-23  macOS Mojave
 6    041-90855    10.13.5   17F66a  2019-10-23  Install macOS High Sierra Beta
 7    061-26578    10.14.5  18F2059  2019-10-14  macOS Mojave

Choose a product to download (1-7):
```


ETH zürich

Forked installinstallmacos.py

To be able to run this script from within another script, I had to expand the functionality, so that it didn't need that user interaction.
To achieve this, I made a fork, [click] added additional flags such as 'current', 'auto', 'version' and 'os' allowing us to automate the download of the build we want, and built in compatibility checks to validate the pre-made choice.

ETH zürich

Forked installinstallmacos.py

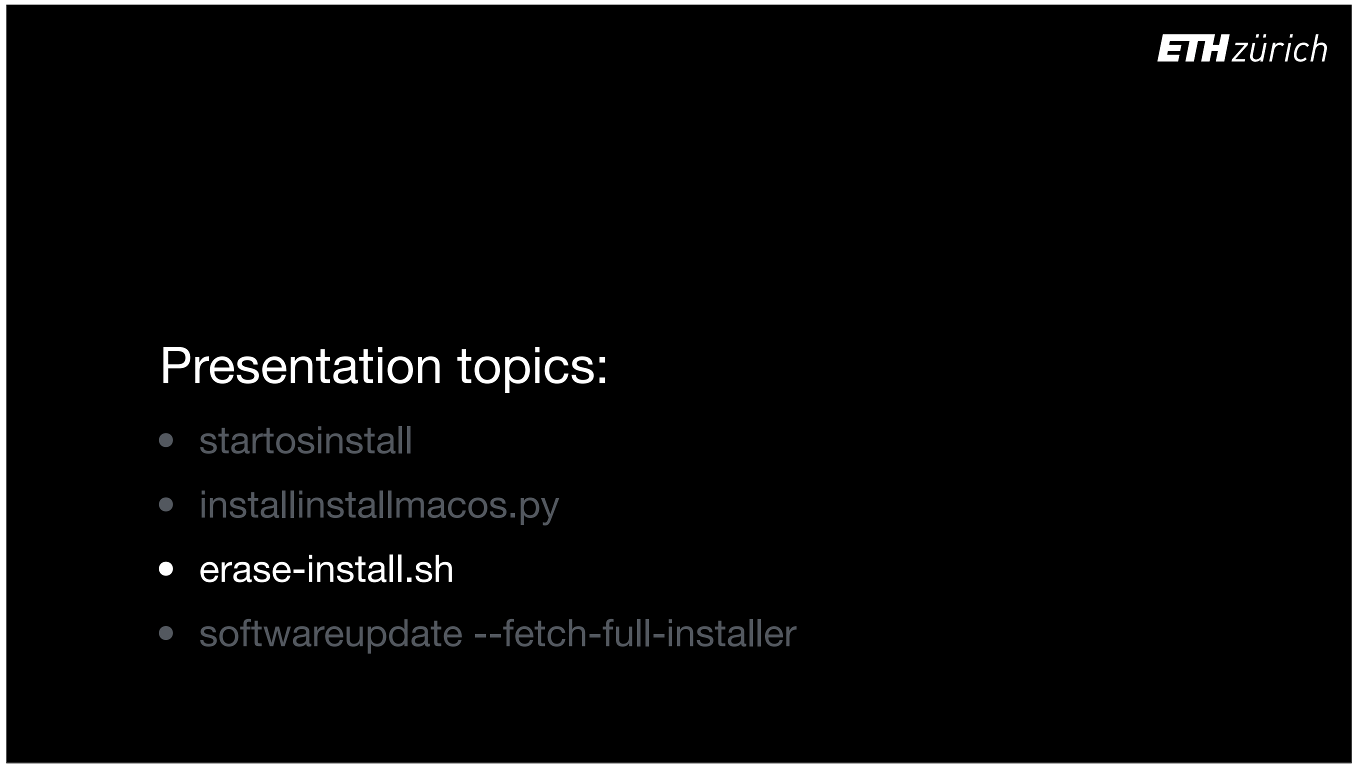
github.com/grahampugh/macadmin-scripts

Forked installinstallmacos.py

github.com/grahampugh/macadmin-scripts

```
$ sudo python installinstallmacos.py --help
```

```
usage: installinstallmacos.py [-h] [--seedprogram SEEDPROGRAM]
                              [--catalogurl CATALOGURL]
                              [--workdir path_to_working_dir] [--compress]
                              [--raw] [--ignore-cache] [--build build_version]
                              [--list] [--current] [--validate] [--auto]
                              [--beta] [--version match_version]
                              [--os match_os]
```


ETH zürich

Presentation topics:

- startosinstall
- installinstallmacos.py
- erase-install.sh
- softwareupdate --fetch-full-installer

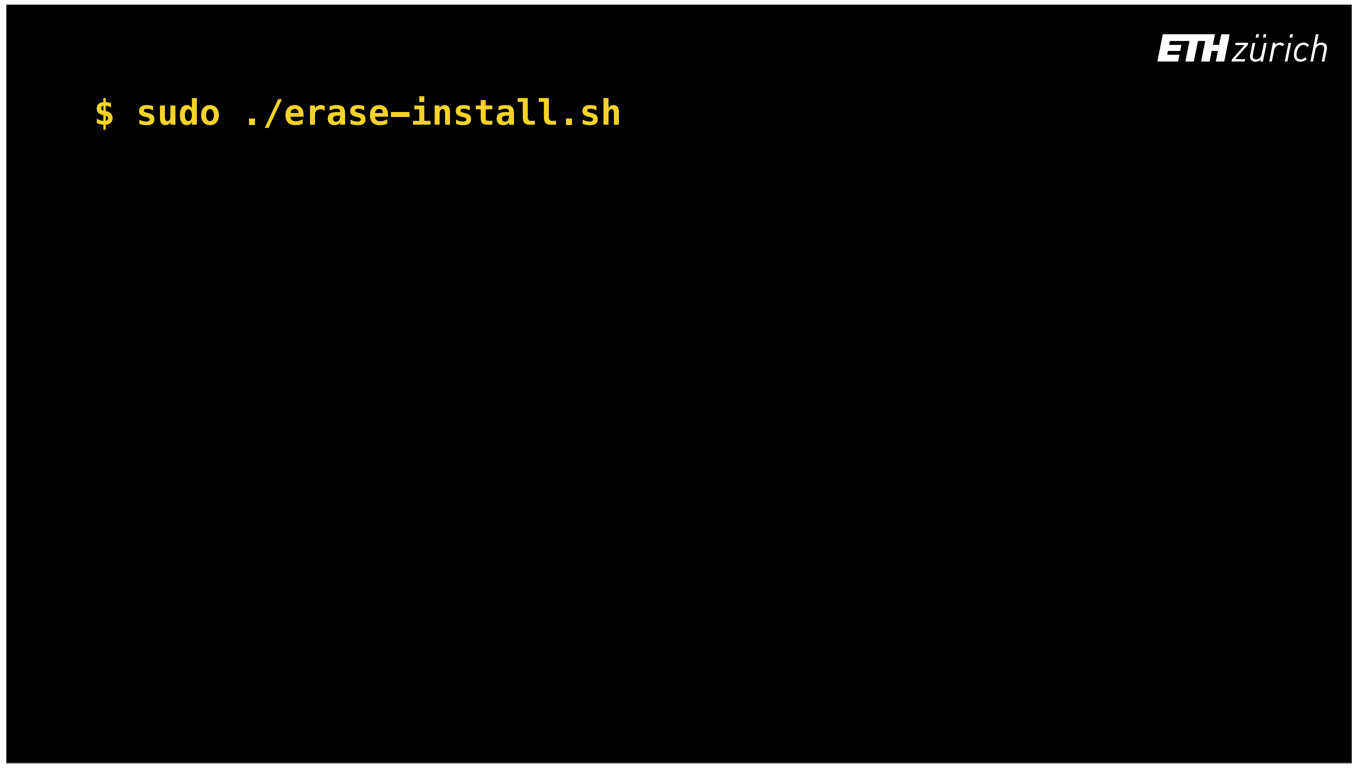
Now I'll show you how Erase-install brings installinstallmacos and startosinstall together, and the main options you're likely to want to use.

If I run erase-install with the `list` option, it downloads and runs the forked installinstallmacos but does not proceed to download anything. This is great for just seeing what's available.
It shows you your computer's Model, Board ID and current OS build, from which it identifies compatible upgrades.
On the right, you see which builds would or wouldn't install on the current system.

ETHzürich

$ sudo ./erase-install.sh --list

```
installinstallmacos.py - get macOS installers from the Apple software catalog

This Mac:
Model Identifier : MacBookPro14,2
Board ID         : Mac-CAD6701F7CEA0921
OS Version       : 10.15.4
Build ID         : 19E266

 #  ProductID       Version    Build    Post Date   Title                          Notes
 1  061-96006       10.15.4    19E287   2020-04-08  macOS Catalina
 2  041-91758       10.13.6    17G66    2019-10-19  macOS High Sierra              Unsupported macOS version
 3  061-86291       10.15.3    19D2064  2020-03-23  macOS Catalina                 Unsupported Board ID
 4  061-26589       10.14.6    18G103   2019-10-14  macOS Mojave                   Unsupported macOS version
 5  041-88800       10.14.4    18E2034  2019-10-23  macOS Mojave                   Unsupported Board ID
 6  041-90855       10.13.5    17F66a   2019-10-23  Install macOS High Sierra Beta Unsupported macOS version
 7  061-26578       10.14.5    18F2059  2019-10-14  macOS Mojave                   Unsupported Board ID

Valid seeding programs are: PublicSeed, CustomerSeed, DeveloperSeed
```



```
$ sudo ./erase-install.sh
```

If you simply want to download a macOS installer and do nothing else, run erase-install with no flags.

- This checks if there is an installer already on the system
- If not, it calls installinstallmacos to download the latest compatible version of macOS.
- The installer is compressed and stored as a DMG in this location

This is useful where you want to cache the installer ready for a later run of this same script with different arguments so it perform a reinstall much quicker.

```
$ sudo ./erase-install.sh
```

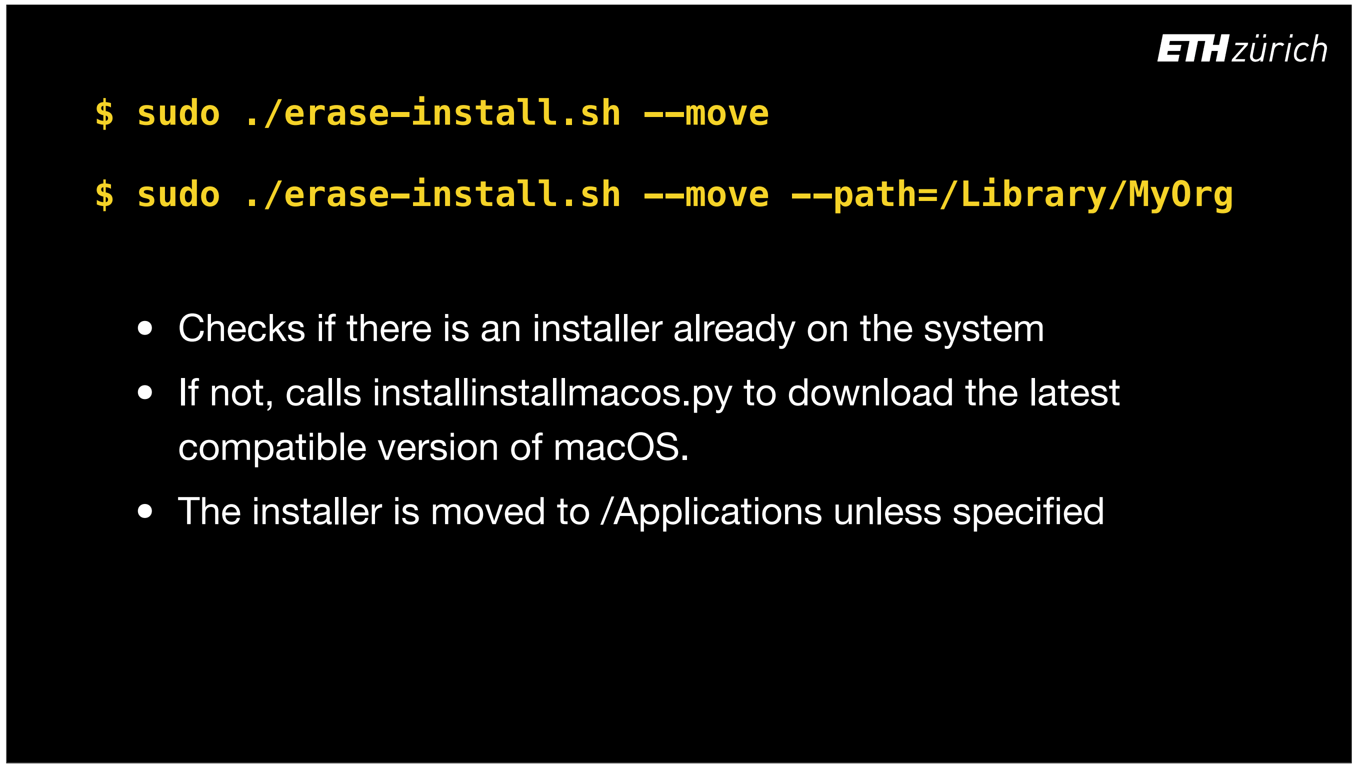
- Checks if there is an installer already on the system

```
$ sudo ./erase-install.sh
```

- Checks if there is an installer already on the system
- If not, calls installinstallmacos.py to download the latest compatible version of macOS.

```
$ sudo ./erase-install.sh
```

- Checks if there is an installer already on the system
- If not, calls installinstallmacos.py to download the latest compatible version of macOS.
- The installer is compressed and stored as a DMG in /Library/Management/erase-install

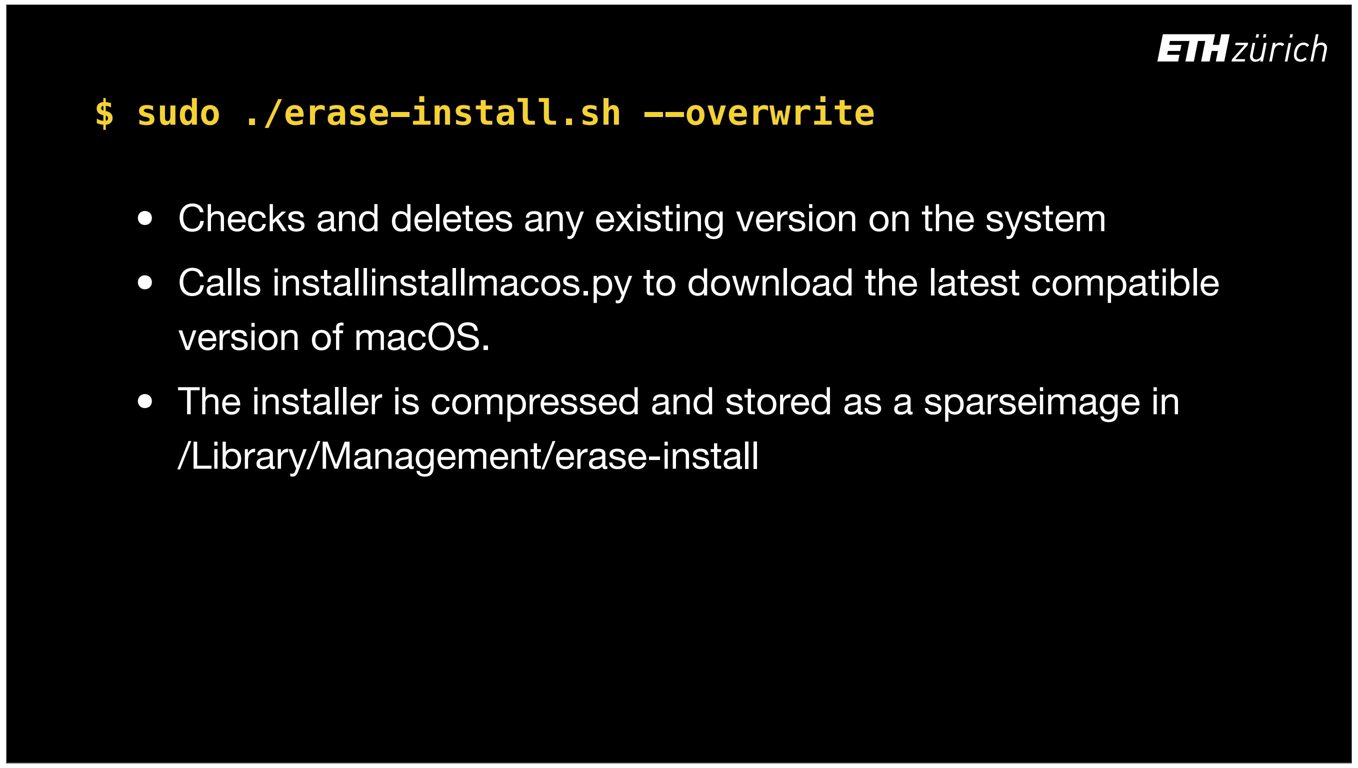


ETH zürich

```
$ sudo ./erase-install.sh --move

$ sudo ./erase-install.sh --move --path=/Library/MyOrg
```

- Checks if there is an installer already on the system
- If not, calls installinstallmacos.py to download the latest compatible version of macOS.
- The installer is moved to /Applications unless specified

You can use the `move` flag to move the installer app to /Applications, or somewhere else, so that the end user can run the installer manually from Finder.

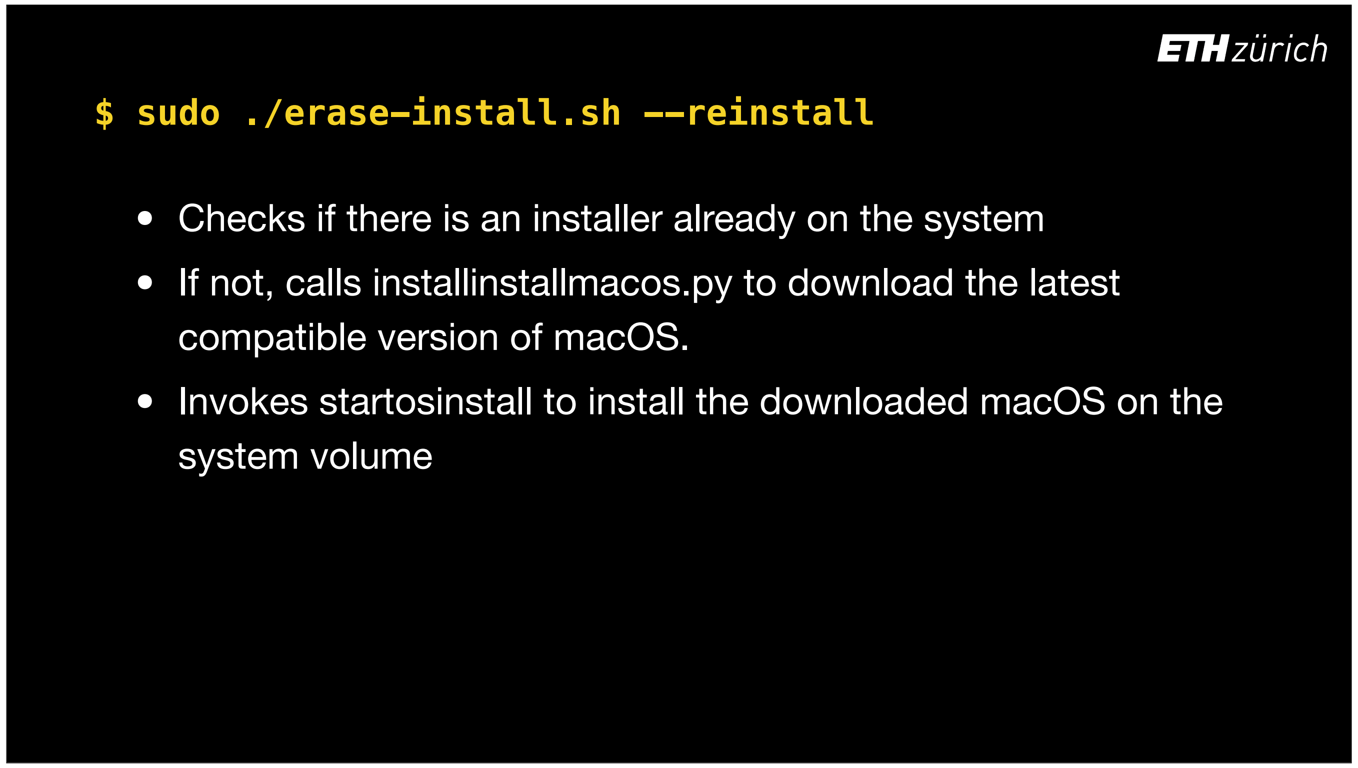


ETH zürich

```
$ sudo ./erase-install.sh --overwrite
```

- Checks and deletes any existing version on the system
- Calls installinstallmacos.py to download the latest compatible version of macOS.
- The installer is compressed and stored as a sparseimage in /Library/Management/erase-install

If you want to ignore any existing installers on the computer and make sure you get the latest build, use the --overwrite option, which deletes any existing installer found.

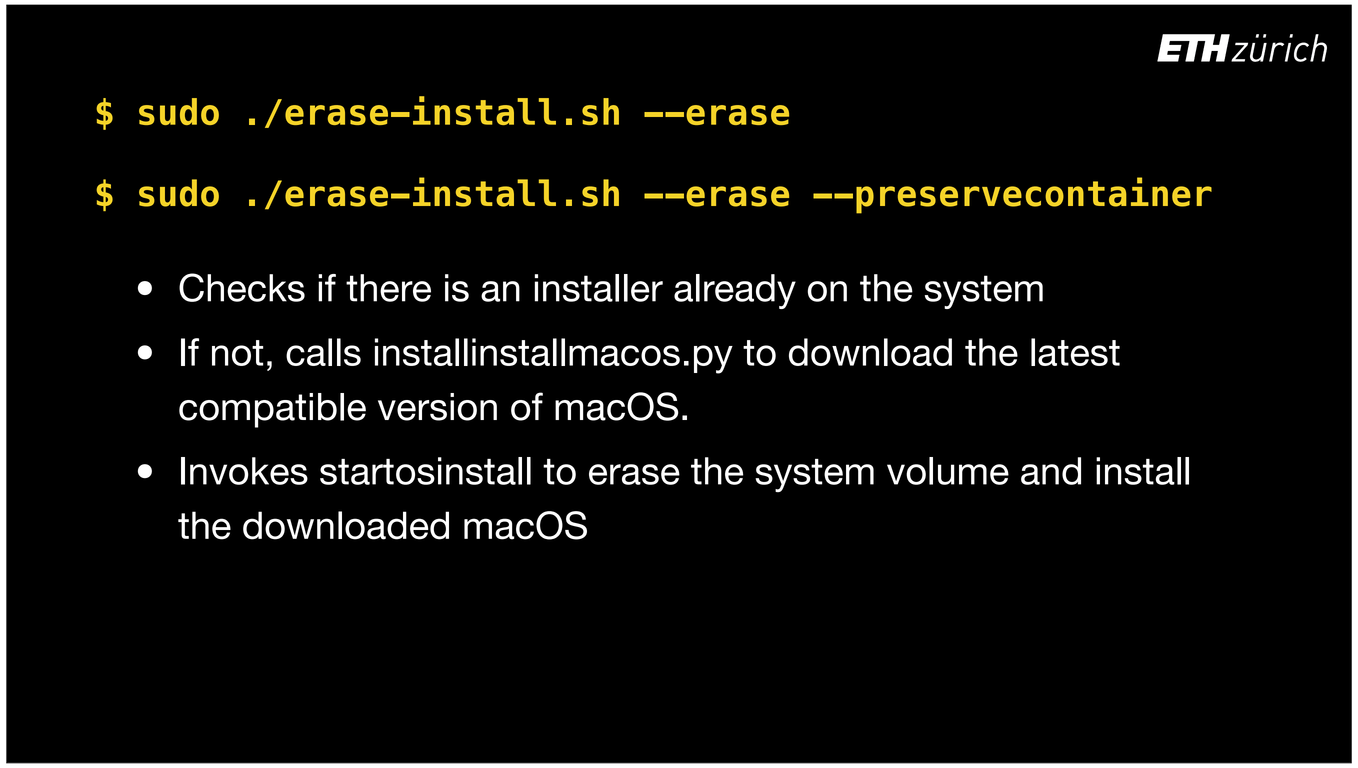


ETH zürich

```
$ sudo ./erase-install.sh --reinstall
```

- Checks if there is an installer already on the system
- If not, calls installinstallmacos.py to download the latest compatible version of macOS.
- Invokes startosinstall to install the downloaded macOS on the system volume

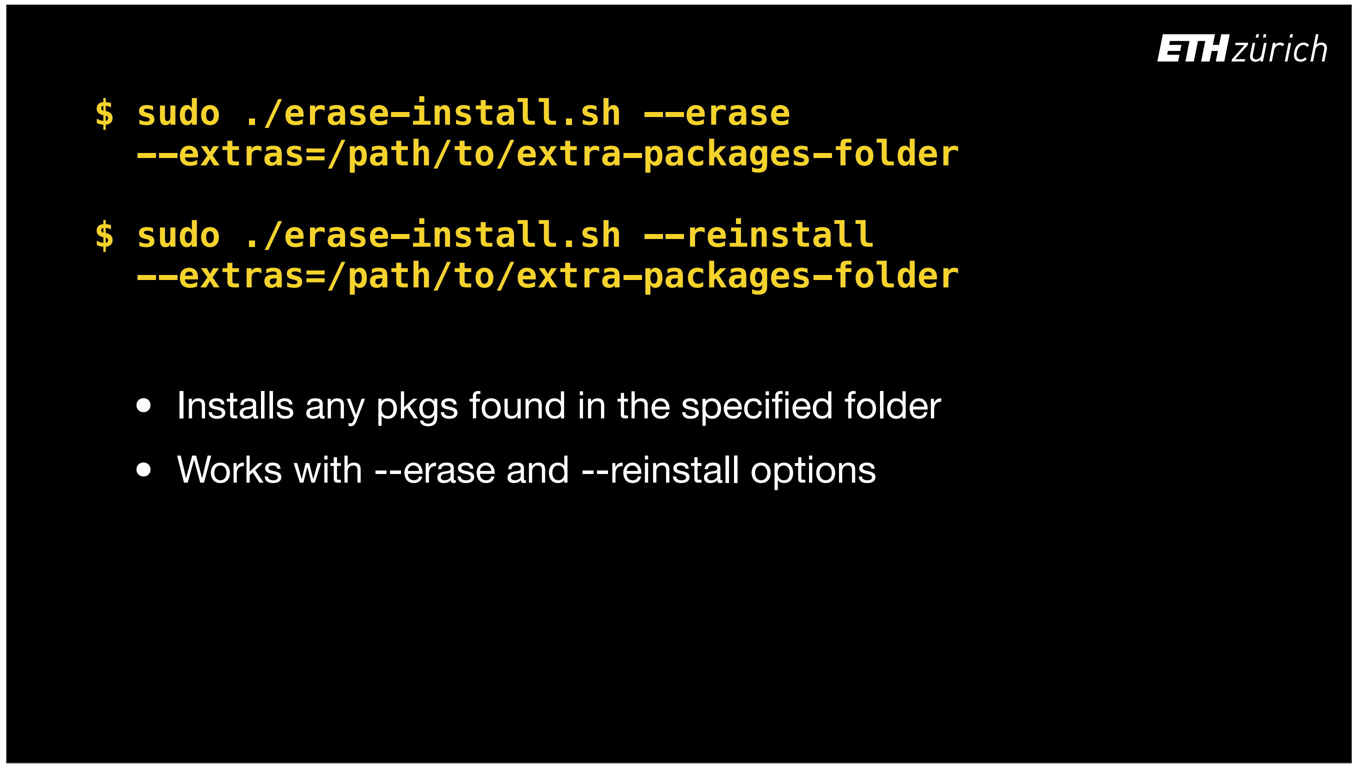
With the reinstall flag, the downloaded installer is used immediately to install over the top of the existing OS.
Typically this is used for upgrading macOS, but could also be used as an emergency reinstallation of the existing OS version.


ETH zürich

```
$ sudo ./erase-install.sh --erase

$ sudo ./erase-install.sh --erase --preservecontainer
```

- Checks if there is an installer already on the system
- If not, calls installinstallmacos.py to download the latest compatible version of macOS.
- Invokes startosinstall to erase the system volume and install the downloaded macOS

The erase option performs startosinstall with the eraseinstall argument. Therefore the system will be wiped. You can however specify the preserve container flag if you only wish to wipe the current container.



ETH zürich

```
$ sudo ./erase-install.sh --erase
  --extras=/path/to/extra-packages-folder

$ sudo ./erase-install.sh --reinstall
  --extras=/path/to/extra-packages-folder
```

- Installs any pkgs found in the specified folder
- Works with --erase and --reinstall options

The extras flag looks for pkgs in the specified folder and installs these packages using the installpackage option of startosinstall. This works with both the reinstall and erase options. This is useful for installing things on a vanilla system such as Munki, puppet or Ansible agents, NoMAD-Login, outset, Yo, or a Jamf QuickAdd package.

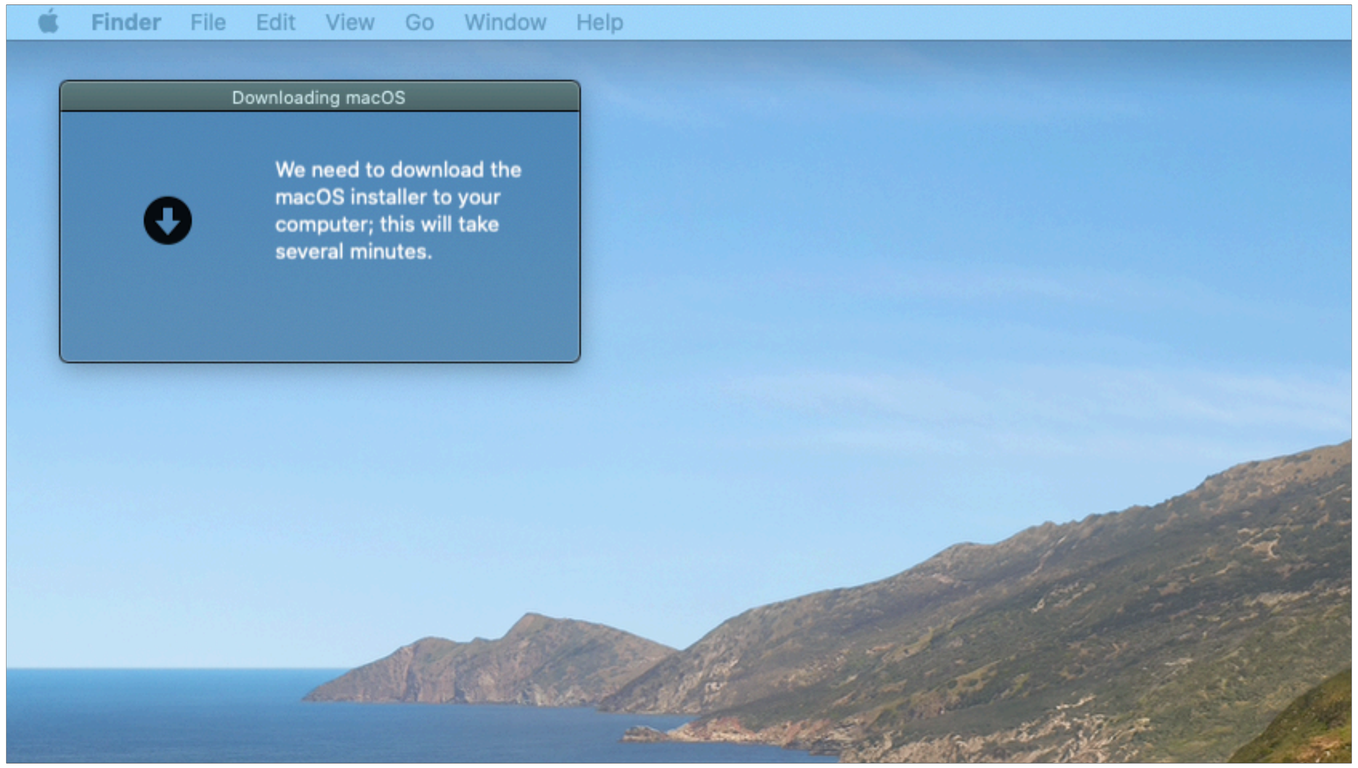
The default is to download the latest compatible OS, but there are many options for specifying which build you want to download.
Hopefully they are fairly self-explanatory
[click through them all]

All of these can be used together with the --move, --overwrite, --reinstall and --erase options.
You can also specify that you are looking for beta versions in the AppleSeed or DeveloperSeeds.
Note that these options will only work with what is available in Apple's software catalog!

```
$ sudo ./erase-install.sh --sameos
$ sudo ./erase-install.sh --sameos

$ sudo ./erase-install.sh --os=10.15
```

ETH zürich

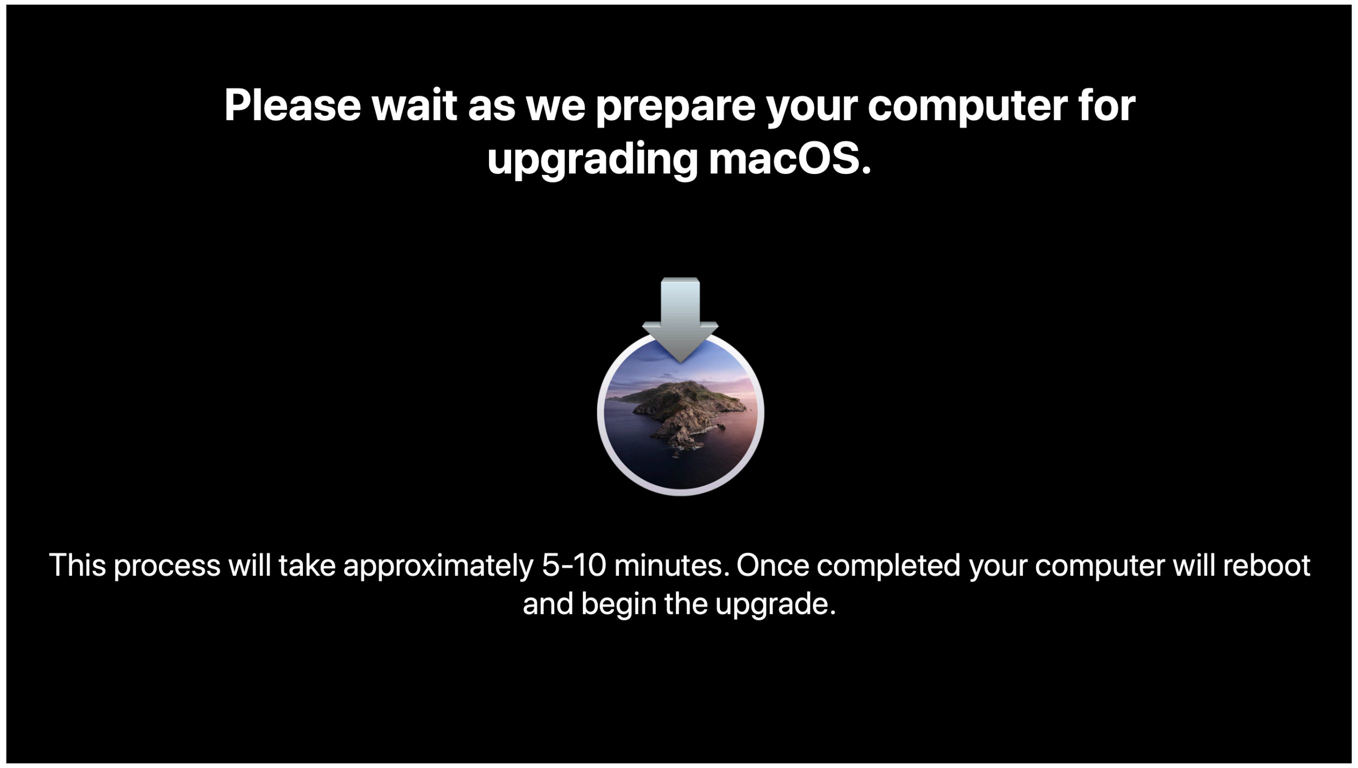
```
$ sudo ./erase-install.sh --sameos

$ sudo ./erase-install.sh --os=10.15

$ sudo ./erase-install.sh --version=10.15.4
$ sudo ./erase-install.sh --sameos

$ sudo ./erase-install.sh --os=10.15

$ sudo ./erase-install.sh --version=10.15.4

$ sudo ./erase-install.sh --samebuild
$ sudo ./erase-install.sh --sameos

$ sudo ./erase-install.sh --os=10.15

$ sudo ./erase-install.sh --version=10.15.4

$ sudo ./erase-install.sh --samebuild

$ sudo ./erase-install.sh --build=19E287
$ sudo ./erase-install.sh --sameos

$ sudo ./erase-install.sh --os=10.15

$ sudo ./erase-install.sh --version=10.15.4

$ sudo ./erase-install.sh --samebuild

$ sudo ./erase-install.sh --build=19E287

$ sudo ./erase-install.sh --beta
  --seedprogram=DeveloperSeed
```



Downloading macOS

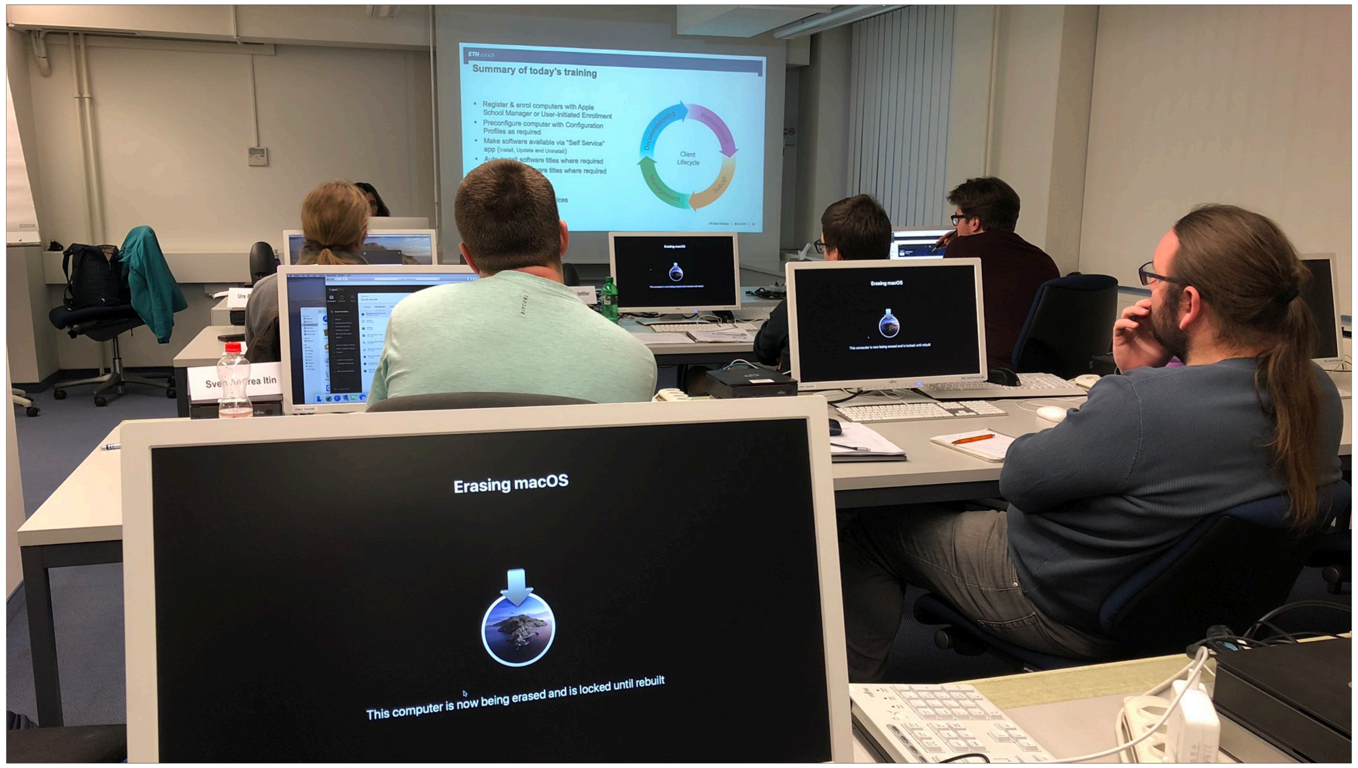
We need to download the macOS installer to your computer; this will take several minutes.

You don't need to run this script on computers enrolled into Jamf, but if you do, you get some nice user feedback, as I've built in calls to the jamfHelper tool. When downloading the installer, a window is shown stating that the download is occurring.

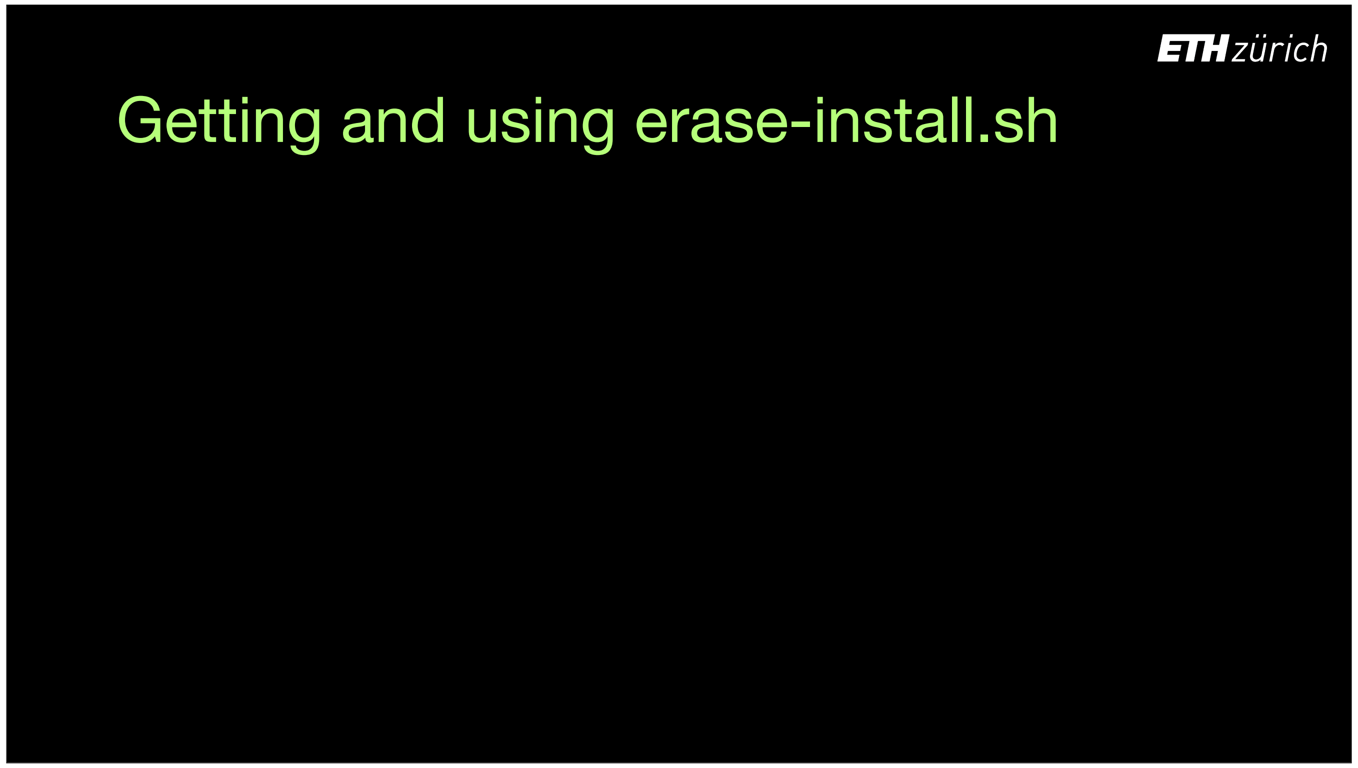

Please wait as we prepare your computer for upgrading macOS.

This process will take approximately 5-10 minutes. Once completed your computer will reboot and begin the upgrade.

When we get to the install stage, a full-screen display takes over, which stays until the computer does the restart. This is the screen that displays when running the reinstall option.


Summary of today's training
Erasing macOS
This computer is now being erased and is locked until rebuilt

The erase screen is similar – here it is in use at the end of one of our internal Jamf training sessions.



Getting and using erase-install.sh

To get hold of and use the script, you can copy it directly into Jamf from the GitHub page and supply the arguments into parameters in a Jamf policy,
Or, download it as a zip archive from the releases page.
Or using the AutoPkg download recipe, or get a packaged version. This installs the script in a folder at /Library/Management/erase-install
The JSS recipes take advantage of the packaged version, and put the correct command and arguments into the Self Service policy, so you don't need to engineer that yourself.

Getting and using erase-install.sh

github.com/grahampugh/erase-install

Getting and using erase-install.sh

github.com/grahampugh/erase-install

github.com/grahampugh/erase-install/releases

Getting and using erase-install.sh

github.com/grahampugh/erase-install

github.com/grahampugh/erase-install/releases

```
$ autopkg run erase-install.download
```

Getting and using erase-install.sh

github.com/grahampugh/erase-install

github.com/grahampugh/erase-install/releases

```
$ autopkg run erase-install.download
$ autopkg run erase-install.pkg
```

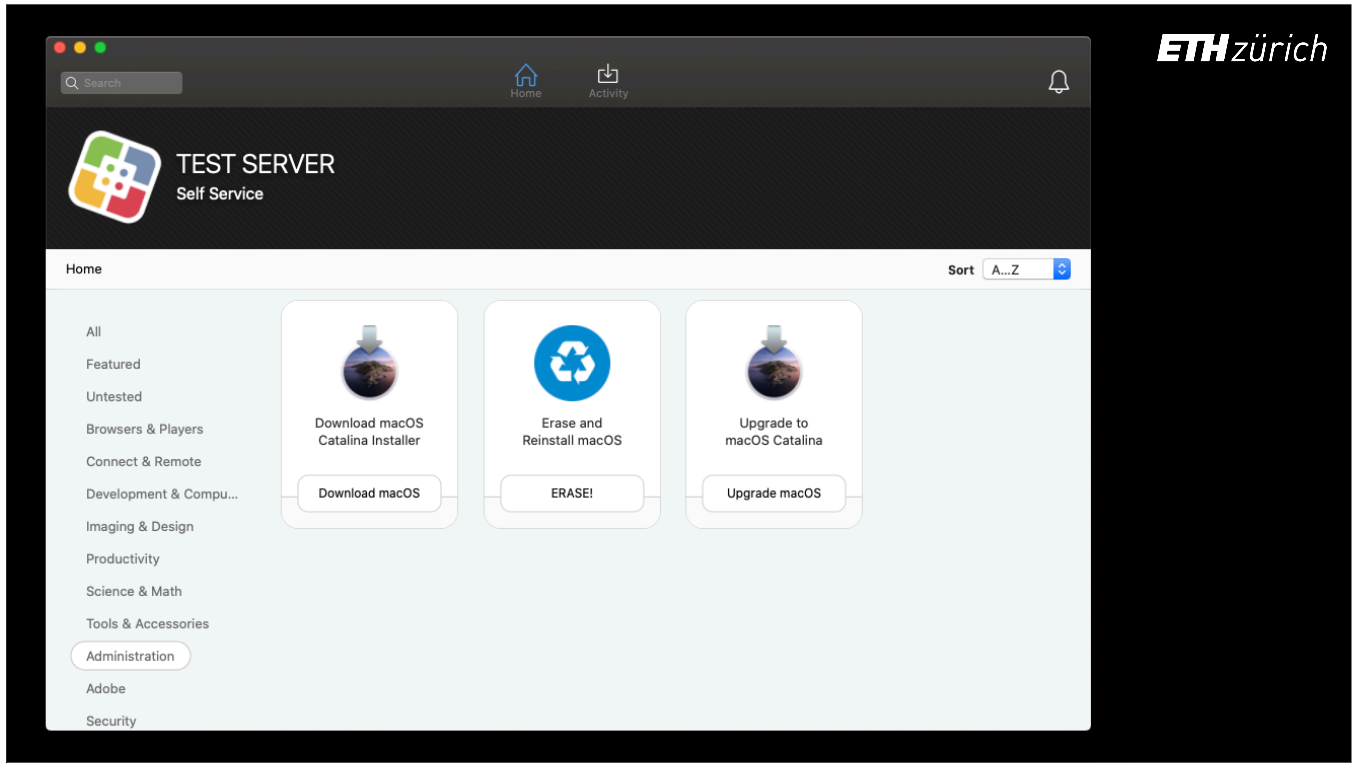
ETH zürich

Getting and using erase-install.sh

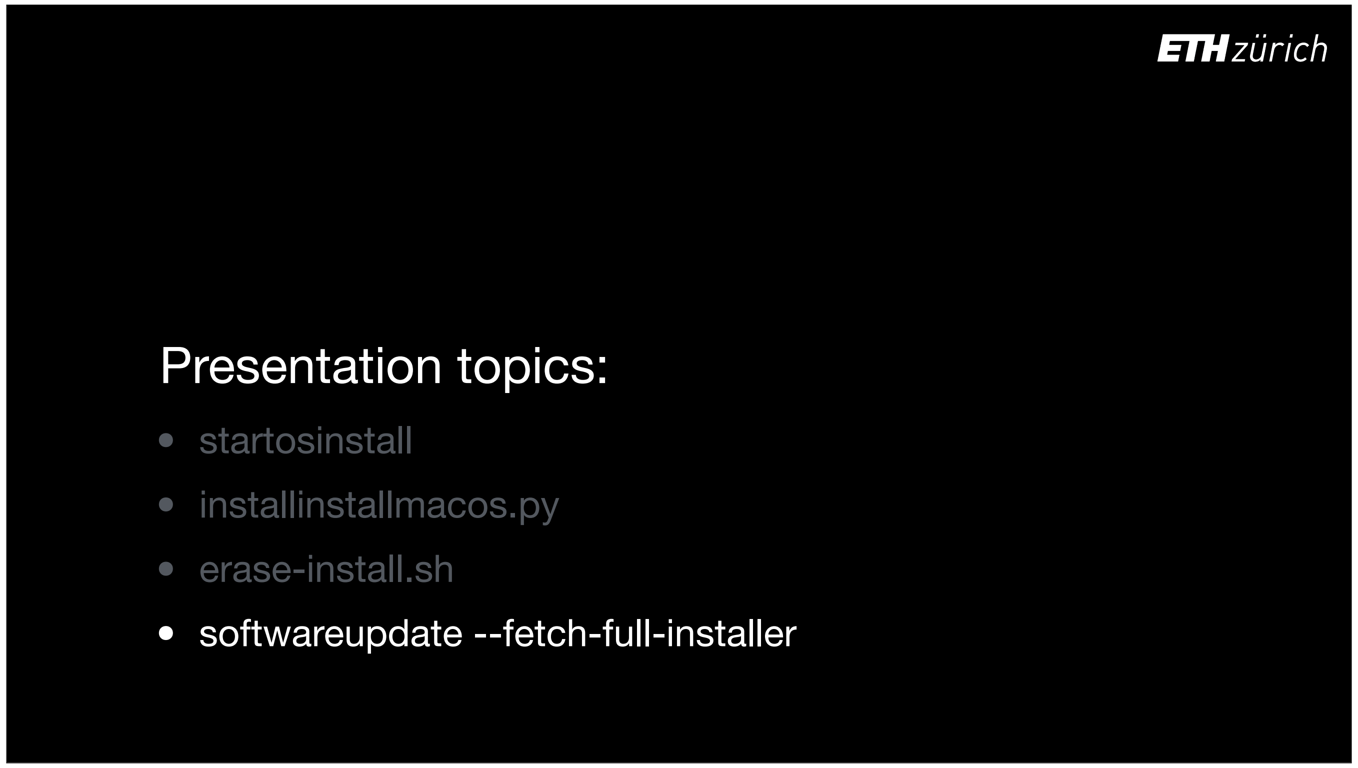
github.com/grahampugh/erase-install

github.com/grahampugh/erase-install/releases

```
$ autopkg run erase-install.download
$ autopkg run erase-install.pkg

$ autopkg run 'Erase and Reinstall macOS.jss'
$ autopkg run 'Download macOS Catalina.jss'
$ autopkg run 'Upgrade to macOS Catalina.jss'
```



The AutoPkg recipes produce self service policies that look like this. They all use the same package, they just specify different run arguments for the script, and have different scopes.


ETH zürich

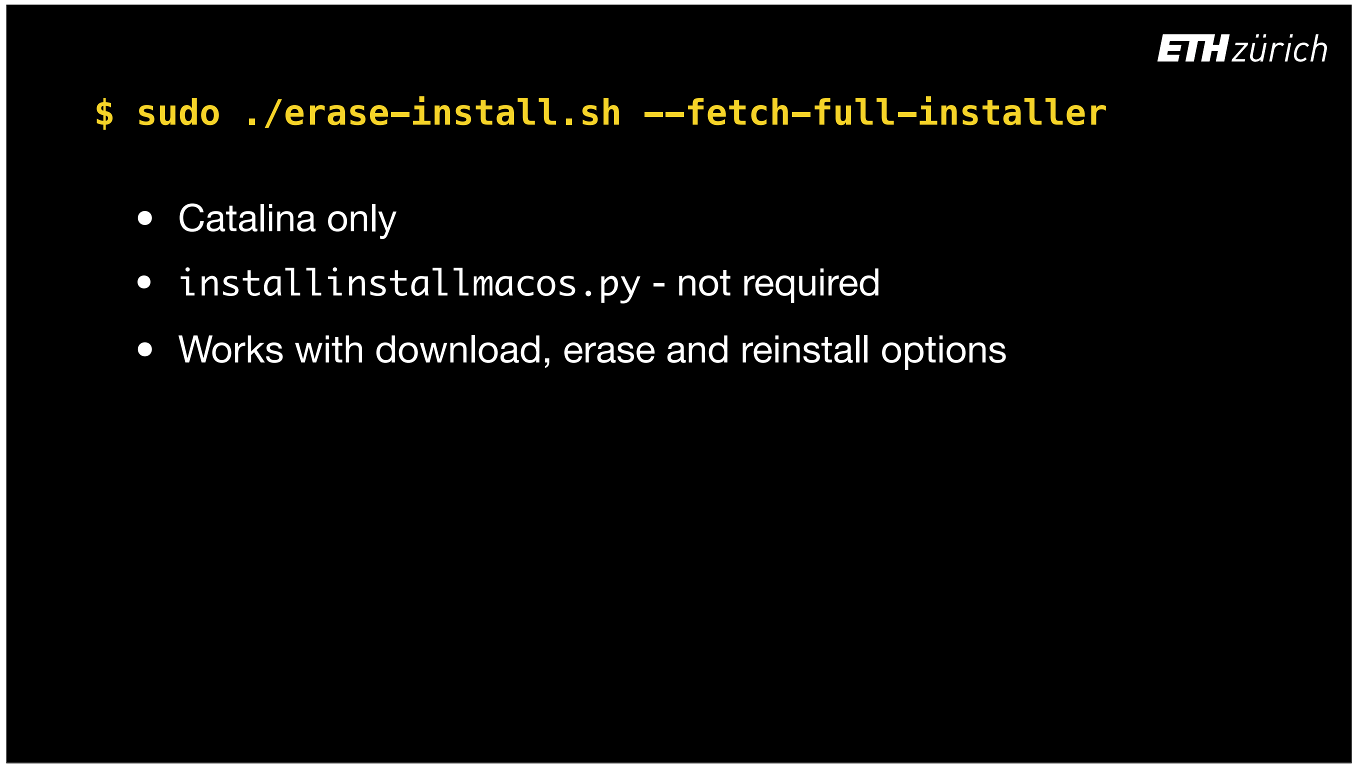
Presentation topics:

- startosinstall
- installinstallmacos.py
- erase-install.sh
- softwareupdate --fetch-full-installer

A quick note about fetch-full-installer

Fetch-full-installer was added to the softwareupdate command with the release of Catalina.
It operates somewhat like `installinstallmacos`, though with fewer options.
By default it will download the current recommended version for your system.
[click] You can specify a particular OS or version to try to download with the `full-installer-version` flag.
There is no way to show list of available builds, so whether a requests will work is trial and error.

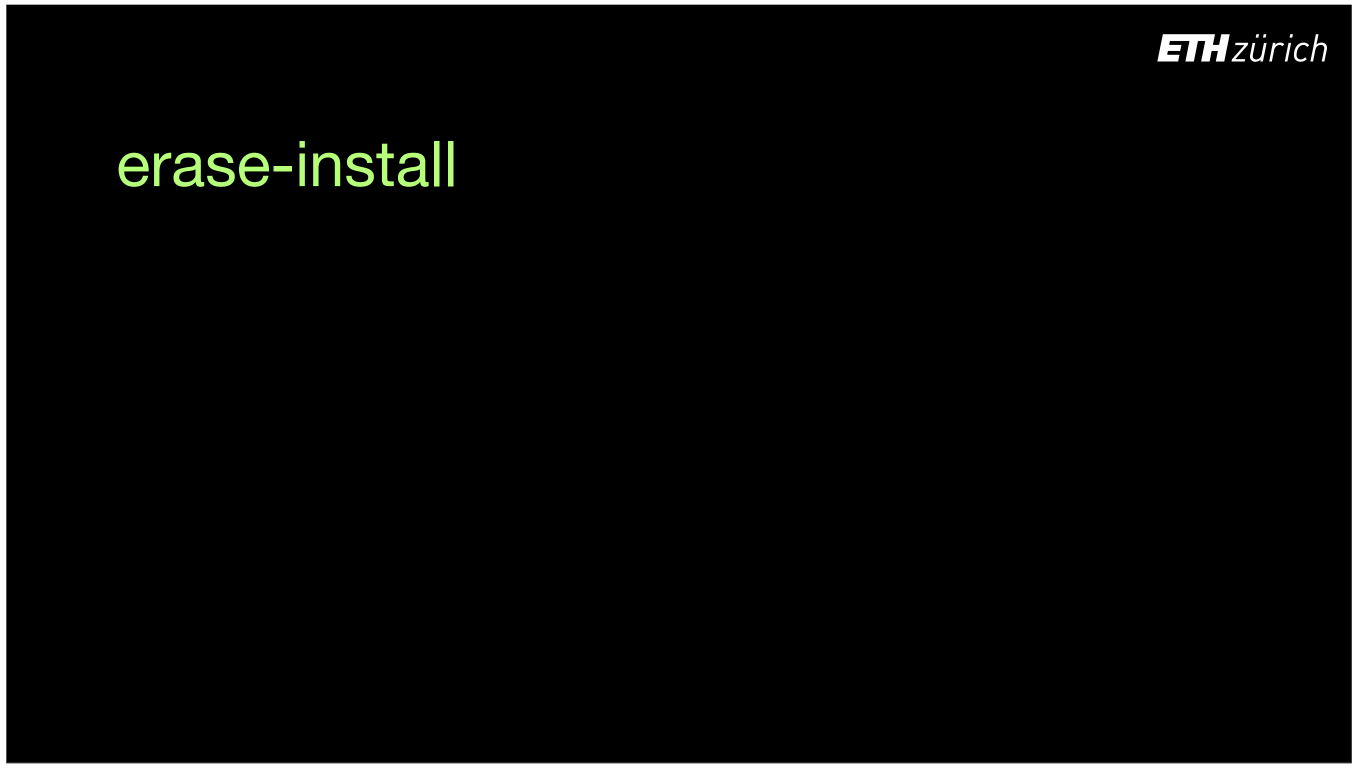
```
$ sudo softwareupdate --fetch-full-installer
$ sudo softwareupdate --fetch-full-installer

$ sudo softwareupdate --fetch-full-installer
  --full-installer-version 10.15
```



```
$ sudo ./erase-install.sh --fetch-full-installer
```

- Catalina only
- `installinstallmacos.py` - not required
- Works with download, erase and reinstall options

I have added experimental support for this in erase-install.
The advantage of this is that we don't need python on the system, something that could be useful going into macOS 10.16 which may not come with a python runtime by default.
The lack of feedback from the command means that it is less robust and needs more testing, however, so this remains an option rather than the default.



erase-install

- In conclusion
- We've wanted an Erase all Contents and Settings option for a long time, but it actually already exists.
- There is no need to manually download and package up macOS installers and upload them to your repo.
- Erase-install provides lots of flexibility for easily obtaining specific OS builds.
- You can also install signed packages at the same time as reinstalling the system.

ETH zürich

erase-install

Conclusions

- One-click 'Reset all Content and Settings' is already here

ETH zürich

erase-install

Conclusions

- One-click 'Reset all Content and Settings' is already here
- No need to package up macOS installers

erase-install

Conclusions

- One-click 'Reset all Content and Settings' is already here
- No need to package up macOS installers
- Lots of flexibility for obtaining the OS build you require

ETH zürich

erase-install

Conclusions

- One-click 'Reset all Content and Settings' is already here
- No need to package up macOS installers
- Lots of flexibility for obtaining the OS build you require
- Install packages at the time of reinstall

Thank You!

Photo: ETH Zürich / Gian Marco Castelberg

That's all from me, thank you for listening!