Lab 3.2 - Helm Installation

Obtaining the Helm Binary from GitHub Release Notes

We will now go over how to install the Helm CLI on your system.

Step 1

First, create a directory somewhere on your system called **`helm-install`**:

```
$ mkdir ${HOME}/helm-install
```

Next, open your browser to the Helm releases page at https://github.com/helm/helm/releases. Find the latest release starting with “v3” (for example “Helm v3.2.0”).

Note: You may see bug-fix releases for the old version, Helm 2. Do NOT use those releases for this course.

Step 2

You should see a section in the release titled “Installation and Upgrading”, followed by several download links. Find the links related to your operating system and architecture. On most modern systems, use **amd64**. If you are unsure about your system’s architecture, you can run the following command:

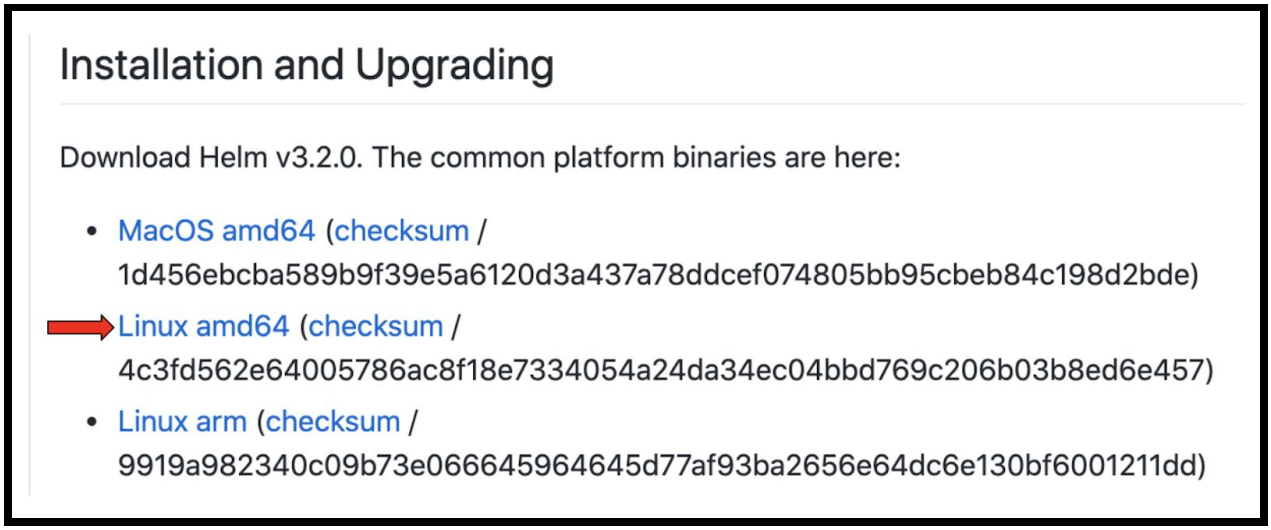
```
$ uname -m
```

If the result of that command is **`x86_64`**, use architecture **amd64**. If the result of that command

 © Copyright the Linux Foundation 2020. All rights reserved.

is `x86`, use architecture **i386**. Other results should be self-explanatory. It is important to use the correct architecture, otherwise Helm will not run on your system.

The following image shows an example of a download link for Linux amd64:



Step 3

Open a terminal and enter the install directory:

```
$ cd ${HOME}/helm-install
```

Step 4

Once you've determined the correct download link based on your system, right-click on the link and click "Copy Link Address". Then use wget to download the file:

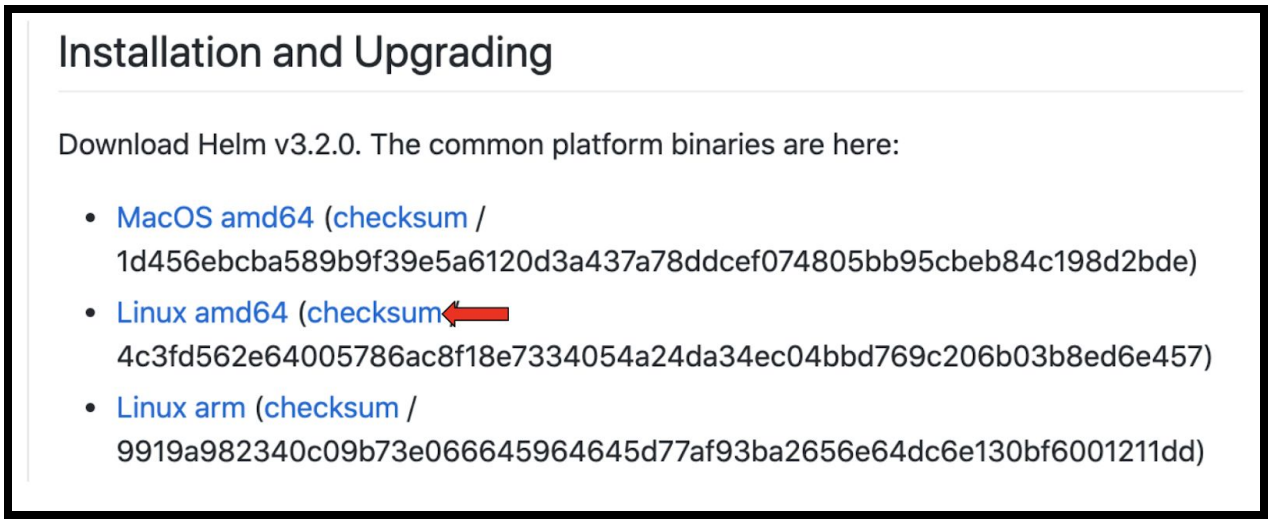
```
$ wget https://get.helm.sh/helm-v3.2.0-linux-amd64.tar.gz
```

Optional: Verify the Checksum

You are encouraged to validate the `.tar.gz` file against the SHA-256 checksum. This validates that the released artifact has not been modified since the time it was released.

The following image shows an example of a checksum link for Linux amd64:

 © Copyright the Linux Foundation 2020. All rights reserved.



Step 5

Once you've determined the correct checksum link, right-click on the link and click "Copy Link Address". Then use wget to download the file:

```
$ wget https://get.helm.sh/helm-v3.2.0-linux-amd64.tar.gz.sha256sum
```

Step 6

Run the following commands (*Note: The name of your* **`.tar.gz`** *and* **`.sha256sum`** *files will be different depending on the Helm version used as well as the download and checksum links you used for your system*):

```
$ cat helm-v3.2.0-linux-amd64.tar.gz.sha256sum
4c3fd562e64005786ac8f18e7334054a24da34ec04bbd769c206b03b8ed6e457
helm-v3.2.0-linux-amd64.tar.gz

$ shasum -a 256 helm-v3.2.0-linux-amd64.tar.gz
4c3fd562e64005786ac8f18e7334054a24da34ec04bbd769c206b03b8ed6e457
helm-v3.2.0-linux-amd64.tar.gz
```

If your system is missing the **`shasum`** command, try the **`sha256sum`** command instead:

```
$ sha256sum helm-v3.2.0-linux-amd64.tar.gz
4c3fd562e64005786ac8f18e7334054a24da34ec04bbd769c206b03b8ed6e457
helm-v3.2.0-linux-amd64.tar.gz
```

Notice in the example output above that the **`.tar.gz`** file's checksum matches the expected

 © Copyright the Linux Foundation 2020. All rights reserved.

checksum.

Optional: Validate the Signature

Note: This step requires that you have gpg (https://gnupg.org/) installed on your system.

If you want to be extra safe about what you're installing onto your system, you can validate the signature of both the `.tar.gz` and the `.sha256sum` files you have downloaded.

Each release artifact is signed by the maintainer who released it, and those signatures are then uploaded to the GitHub release. So even in the rare event that a hacker compromises the `.tar.gz` and `.sha256sum` files and replaces them with evil ones, those artifacts would not contain a valid signature.

Navigate your browser back to the release notes in GitHub. At the bottom of the release, you will find a section labeled "Assets". There will be listed numerous download links for files ending in `.asc`. Find the two `.asc` files related to the `.tar.gz` and `.sha256sum`. They should be fairly easy to locate, as they are the name of the original file suffixed with "`.asc`":

Step 7

Once you determined the correct links, save them into the `helm-install` directory. Once again, right-click on each link and click "Copy Link Address", then use wget to download the files:

 © Copyright the Linux Foundation 2020. All rights reserved.

```
$ wget
https://github.com/helm/helm/releases/download/v3.2.0/helm-v3.2.0-linu
x-amd64.tar.gz.asc

$ wget
https://github.com/helm/helm/releases/download/v3.2.0/helm-v3.2.0-linu
x-amd64.tar.gz.sha256sum.asc
```

Step 8

Next obtain the latest Helm **KEYS** file. This is a file which contains a collection of all of the maintainers' PGP keys which have been used to sign a Helm release:

```
$ wget https://raw.githubusercontent.com/helm/helm/master/KEYS
```

The **helm-install** directory should now contain 5 files total (**.tar.gz**, **.sha256sum**, two **.asc** files, and **KEYS**).

Step 9

Run the following commands to verify the signatures of both the **.tar.gz** package and the checksum using GNU Privacy Guard (gpg). This will create a temporary keyring based on the **KEYS** file, then use that to validate the signatures. Be sure to replace the **TARFILE** and **SUMFILE** with the **.asc** files you have downloaded:

```
$ TARFILE="helm-v3.2.0-linux-amd64.tar.gz.asc"
$ SUMFILE="helm-v3.2.0-linux-amd64.tar.gz.sha256sum.asc"

$ mkdir -p -m 0700 gnupgtemp
$ gpg --batch --quiet --homedir=gnupgtemp --import KEYS
$ gpg --batch --no-default-keyring --keyring "gnupgtemp/pubring.kbx"
--export > "tempkeyring.gpg"

$ COUNT=$(gpg --verify --keyring="${PWD}/tempkeyring.gpg"
--status-fd=1 "${TARFILE}" | grep -c -E '^\[GNUPG:\]
(GOODSIG|VALIDSIG)'); [[ ${COUNT} -ge 2 ]] && echo "Verified signature
of ${TARFILE}" || echo "FAILED to verify ${TARFILE}"

$ COUNT=$(gpg --verify --keyring="${PWD}/tempkeyring.gpg"
--status-fd=1 "${SUMFILE}" | grep -c -E '^\[GNUPG:\]
(GOODSIG|VALIDSIG)'); [[ ${COUNT} -ge 2 ]] && echo "Verified signature
```

 © Copyright the Linux Foundation 2020. All rights reserved.

```
of ${SUMFILE}" || echo "FAILED to verify ${SUMFILE}"
```

If the output for the last two commands end with "`Verified signature of...`", you are good to go. You can safely ignore any warnings. Move on to the next section to install Helm CLI.

If those last two commands end with "`FAILED to verify...`", please double-check, then discreetly send an email to cncf-helm-security@lists.cncf.io. This may be a security issue. More details on Helm security disclosures can be found here: https://github.com/helm/community/blob/master/SECURITY.md.

Installing the Helm Binary

The final step in the installation process is to extract the `.tar.gz` file which contains the Helm binary, and move it into a permanent location on your system.

Step 10

Inside the `helm-install` directory, extract the `.tar.gz` file:

```
$ tar -zxvf helm-v3.1.1-darwin-amd64.tar.gz
linux-amd64/
linux-amd64/LICENSE
linux-amd64/README.md
linux-amd64/helm
```

The output of that command should display the extracted contents. Look for the Helm binary, a file called `helm` (in this example, `linux-amd64/helm`).

Step 11

Lastly, move the Helm binary into a permanent location, such as `/usr/local/bin`:

```
$ sudo mv linux-amd64/helm /usr/local/bin/helm
```

Note: The last command may require some administrative privilege (hence the `sudo`). You can also move the Helm binary to another location in your PATH.

Step 12

Verify Helm is installed:

```
$ which helm
```

 © Copyright the Linux Foundation 2020. All rights reserved.

```
/usr/local/bin/helm
```

Step 13

Finally, make sure that Helm runs properly:

```
$ helm version
version.BuildInfo{Version:"v3.2.0",
GitCommit:"e11b7ce3b12db2941e90399e874513fbd24bcb71",
GitTreeState:"clean", GoVersion:"go1.13.10"}
```

Step 14

Congrats! You're ready to start using Helm. Feel free to discard the entire `helm-install` directory:

```
$ cd ../
$ rm -rf helm-install/
```

 © Copyright the Linux Foundation 2020. All rights reserved.