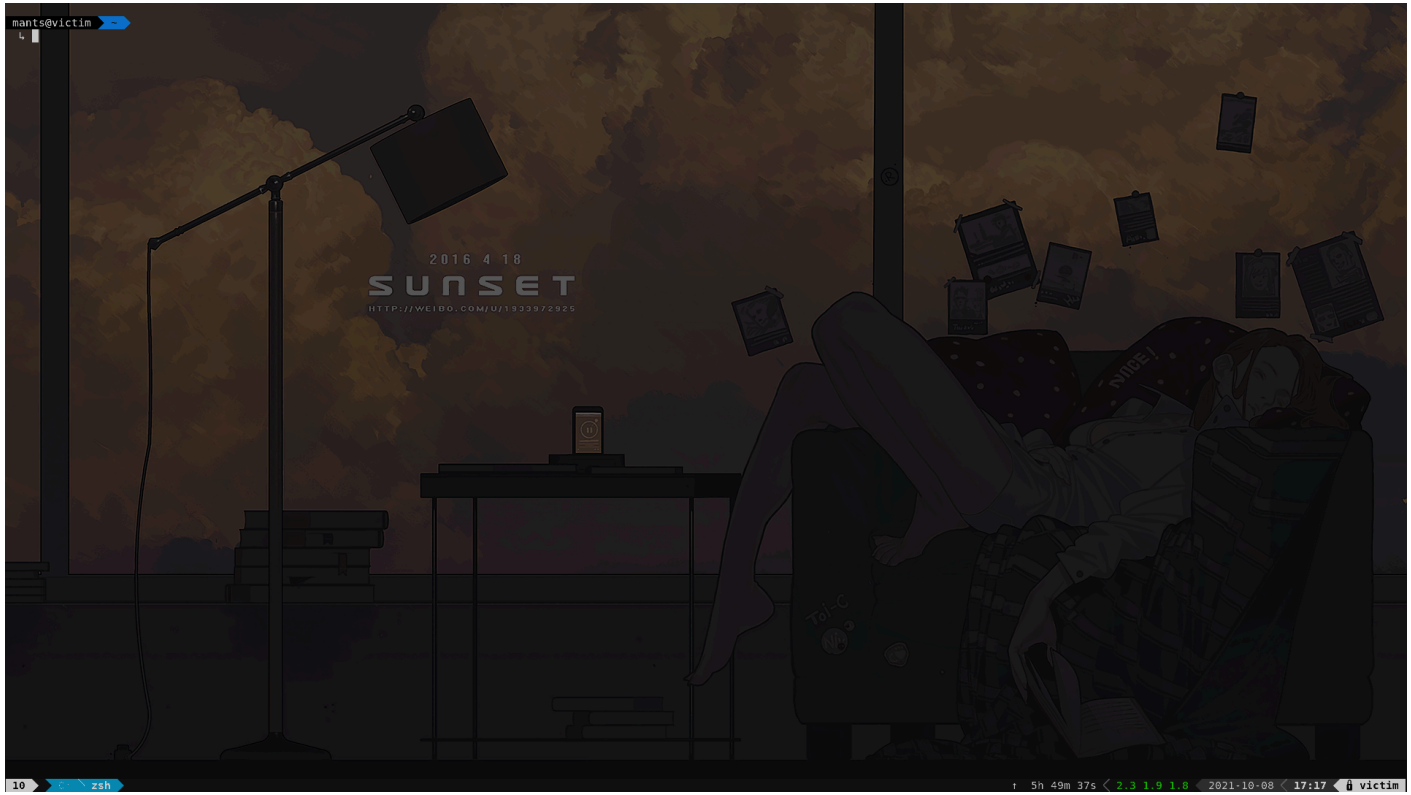
01. [40p] zsh

bash is a wonderful shell but there are better alternatives for day-to-day use. One of these alternatives is **zsh** (pronounced “Z-shell”). **zsh** provides a more robust tab completion, easier navigation through directories and a larger selection of plugins, among many other. At the same time, it retains compatibility with **bash** features such as Word Designators [https://www.gnu.org/software/bash/manual/html_node/Word-Designators.html] or Modifiers [https://www.gnu.org/software/bash/manual/html_node/Modifiers.html].



Click GIF to maximize.

In this exercise, you will install **zsh** and configure it with predefined themes and plugins.

[10p] Task A - Dependency installation

Pretty much everything you will install in Ubuntu (at least while you're still new to this) will be done via **apt**. **apt** is a CommandLine Interface (CLI) for the package management system (think Windows Store, but better). First thing you must do after a fresh install is to **update** your information about the packages available for download. Once you do, you'll be able to do a term search and install whatever you please.

```
$ sudo apt update
```

```
$ apt search zsh
```

Note a few things:

1. The first command is prefixed by **sudo** while the second isn't. **sudo** basically tells your shell to execute the remainder of the command (in this case `apt update`) with *"administrator privileges"*. Updating your local package information is a sensitive operation that not every user should be able to do. But just searching for a packet (without installing it) is a harmless operation, so no need for **sudo** there.
2. The `apt search zsh` command returns a rather long list. This is because **apt** performs an approximate search in both the available package names and descriptions. If you want an *exact* match, try using a regular expression [https://www.computerhope.com/jargon/r/regex.htm] such as *^zsh$* which should be interpreted as: "beginning of the line, followed by the string 'zsh', followed by the end of the line".

Anyway, moving forward we will want to use a popular collection of **zsh** configurations called oh-my-zsh [https://github.com/ohmyzsh/ohmyzsh]. As such, we will need to install a few dependencies first:

```
$ sudo apt install curl wget git zsh fonts-powerline
```

These are pretty well known tools and you are *very* likely to use them again; here is a short description of each one:

- **curl** : tool for transferring data to/from servers (e.g.: sending an HTTP request and printing the page's HTML code).
- **wget** : non-interactive web downloader; fetches resources from URLs and saves them locally.
- **git** : a CLI for project versioning, enabling you to collaborate with other people; right now we'll use it only to download **oh-my-zsh** but we'll return to it in a future lab.
- **zsh** : basically, an improved **bash** and what we want to install / customize.
- **fonts-powerline** : this is not a tool, but a font set that is required to render the pointy bits of the *agnoster* theme. Will be useful for **tmux** in the next exercise as well.

[10p] Task B - oh-my-zsh

With this out of the way, let's download **oh-my-zsh** and copy their template configuration file in your *HOME* directory. Then change your user's default shell to **zsh**:

```
$ git clone https://github.com/ohmyzsh/ohmyzsh.git ~/.oh-my-zsh
$ cp ~/.oh-my-zsh/templates/zshrc.zsh-template ~/.zshrc
$ chsh -s $(which zsh)
```

Things to note here:

- Now you have a *.oh-my-zsh* directory in your home. The '.' before the name marks it as an invisible directory, so you will need to specify the `-a` flag when you run **ls**. This directory contains all the sample themes, plugins and auto-complete scripts that **zsh** will use, so don't delete it.
- *.zshrc*, like *.bashrc*, is the default configuration file that the shell will try to access on startup. If you want to change the theme or add plugins, this is the place to look.
- The default shell change that resulted because of **chsh** will not take effect until you relog to your current user. However, you can simply run **zsh** in your terminal and **bash** will be replaced by **zsh**.
- in case you are confused by `$(which zsh)`: this is called Command Substitution [https://www.gnu.org/software/bash/manual/html_node/Command-Substitution.html]. Before the *main* command (i.e.: **chsh**) is executed, **bash** will interpret `which zsh` separately. It's output to *stdout* (in this case, */usr/bin/zsh*, the location of **zsh**) will

replace `$(...)`. So the final form of the command you gave, after substitutions, will be `chsh -s /usr/bin/zsh`.

[10p] Task C - Theme customization

Notice how your **zsh** theme is different from the one in the GIF above? That's because your default is *robbyrussell* and that is *agnoster* (with a few tweaks). Take a look here [https://github.com/larryhynes/oh-my-zsh-themes] and choose a theme that suits you. Then, edit *.zshrc*. When you're done editing the config file and you've saved the changes, source [https://bash.cyberciti.biz/guide/Source_command] it for them to take effect:

```
$ source ~/.zshrc
```

[10p] Task D - Plugin customization

Finally, let's add a few plugins to our shell by editing the *plugins* variable in *.zshrc*. Note that *plugins* is a bash array and the elements are space-separated, not comma-separated.

- dirhistory [https://github.com/ohmyzsh/ohmyzsh/tree/master/plugins/dirhistory] : keeps track of your directory changes and lets you jump back and forth. Note that this is already available in *~/.oh-my-zsh/plugins/dirhistory/* .
- zsh-autosuggestions [https://github.com/zsh-users/zsh-autosuggestions] : gives you command completion suggestions based on your past commands. This plugin is not readily available in the *plugins/* directory, so you will have to download it there first (HINT: scroll down in the repo's page and read the README).

dirhistory shortcuts (last 30 directory changes are buffered):

- *Alt* + ← : go to previous directory
- *Alt* + → : undo previous directory back-jump
- *Alt* + ↑ : same as `cd ..`

Some **bash** / **zsh** shortcuts that can work well with **zsh-autosuggestions**:

- *Ctrl* + ← : move cursor one word left
- *Ctrl* + → : move cursor one word right
- *Ctrl* + *A* : move cursor at start of line
- *Ctrl* + *E* : move cursor at end of line
- *Ctrl* + *W* : delete from start of word and until cursor
- *Ctrl* + *R* : reverse search for command in history