How to prevent nvm from slowing down your shell

NOTE: This also works on bash.

Load nvm when you use it, instead of at shell init
This is literally the best optimization I've done for my zsh configuration. If you're using nvm chances are that it slows down your shell startup significantly (in a scale of half a second or so). This happens when These lines are executed (~/.zshrc):

```
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"
```

You put these into your shell configuration file to load nvm (which is slow for some reason). Now you don't necessarily use nvm on every shell startup, so it makes more sense to load it up when you actually need it. So a very simple shell function can take care of that. Here goes:

```
n () {
  if ! command -v nvm &> /dev/null; then
    export NVM_DIR="$HOME/.nvm"
    [ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
    [ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"
  fi
  nvm "$@"
}
```

In other words, just wrap the loading stuff in an `if` statement which checks to see if `nvm` is available or not. Only when it's not available, it loads `nvm` and then proceeds with the rest of the function. The rest of the function is this tiny line `nvm "$@"` which simply passes all the arguments to `nvm`.

So now the first time you use this `n` function, you're gonna get a bit more delay (which is fine since `nvm` is slow already) but then, every other time (in that same shell) it'll skip over that `if` statement.

Note: This has one tiny flaw, you can't use completion with the `n` command, but you can add it if you want. For my case, the `~/.nvm/bash_completion` is responsible for adding those completion items to my shell. So at the very end of this file, you can add this line:

```
complete -o default -F __nvm n
```