

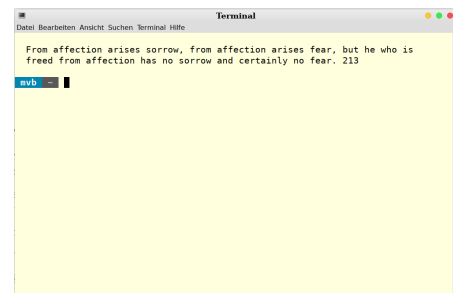
Configuration

<https://github.com/Martin-von-Boehlen/AzTfBeginnersCourse>

cygwin/Linux terminal configuration

No matter whether you use cygwin under Windows or a terminal emulator like xterm under Linux, please do the following simple steps.

1. Call up the options menu with cygwin by right-clicking on an empty part of the title bar. Call up your terminal setup with Linux
2. Choose a block cursor
3. Choose a monospace font with at least 14px
4. Choose high contrast background/foreground colours



bash configuration

Add a PATH to ~/bin, if you have not already done so.

1. Use nano or vi to edit ~/.bash_profile. Add a line `export PATH=$HOME/bin:$PATH` as **last** line
2. For cygwin: close terminal and open it again. For Linux: Log out and log in again
3. Enter `echo $PATH`. It should display `$HOME/bin:` (including colon) as the first part. Like in `/home/<username>/bin:`

az configuration

After installing Azure's shell `az` you don't need to configure anything. But please check whether you are able to login into Azure.

You **have** to be logged into Azure with your browser before you attempt `az login`! There must be a valid and active session. Command line login will not work otherwise. Then do

`az login --use-device-code --allow-no-subscriptions`

git/github configuration

First you must have a valid e-mail. With this you can sign up for a GitHub account by going to <https://github.com>



Then add your public key to your GitHub-account:

1. Generate your key pair. Leave passphrases empty. Then show your public key for github. Do **not** use `id_rsa_github` use `id_rsa_github.pub`

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/<username>/.ssh/id_rsa):id_rsa_github
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa_github
Your public key has been saved in id_rsa_github.pub
The key fingerprint is:
SHA256:VeoTq+scCQVrh8XgjG7tWL/Yd9Rr9LZsyqcpLIAbmH0 e-mail@somewhere
The key's randomart image is:
+---[RSA 3072]-----+
|      oo.   .      |
|      + =.  o      |
|      . * o +      |
|      . o o o o     |
|      * = S + .    |
|      + B E o .. o  |
|      . = * .. . o  |
|      . + =.oo =oo  |
|      ..*....*=+.  |
+-----[SHA256]-----+
$ cat ~/.ssh/id_rsa_github.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGDgZMKyeFmnTmACbRnAbvs9eeXcEV1FhVNu30SJ/
QdRxCHMlMmUSUdK96bA00WQ7maa3cawdp5TlDPVoJ3mFwpYXd/
mNS8oj0iX+lCP5Ckm6slTVSxN52ANGYLUTiaGscWR0CUp7NCw2hcyrbgTwt7rgWUPobS0p
kdZSqdDuGEQx+mgqeT7rdDmyQvPwEUxRupU3m3VT6LSjjz/
TI5shlKC9LgiB2GEX2xwnEhrrv+QVJZJQqYkld4oFGSt6tL2HRab4GN0TmenFzuA/
0e6nBKQY9g+fYFb+4loXjoe/7iGD/b7FZB6+8TTfoyn/
3BaB1FLruRdj6UqGkdHtT420BvaZs6bi9dD1TlaqmbuDt9ZRlBLg4PsCg+2zsNx2sIk9E8
Ta3iwZB2IB0VjFL7qlhn3rxRP1uIbanOgg0VbKhPIlUlvN+MmE6K/
NOM9xuQd5CUPHSGSBu9u5yYoeFqJLF9bUMItcBse1QGf3jv3eiYuuEi2qBmtN39DNtpkcQ
vSQLsgc= <your-e-mail@somewhere>
```

2. Copy everything up to and including the last '='. Do **not** copy blank and e-mail!
3. Go to your GitHub-account <https://github.com/settings/profile> and select "SSH and GPG keys"
4. Select "New SSH key". Sometimes it may read "Create SSH key". Chose a good name and paste your public key. Save it.

Notifications

Scheduled reminders

SSH and GPG keys

Repositories

Now, create your repository in the browser by selecting your profile and in the drop-down select "repositories". Click on "new"

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)



Owner *

Repository name *

 /

Great repository names are short and memorable. Need inspiration? How about **potential-robot**?

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

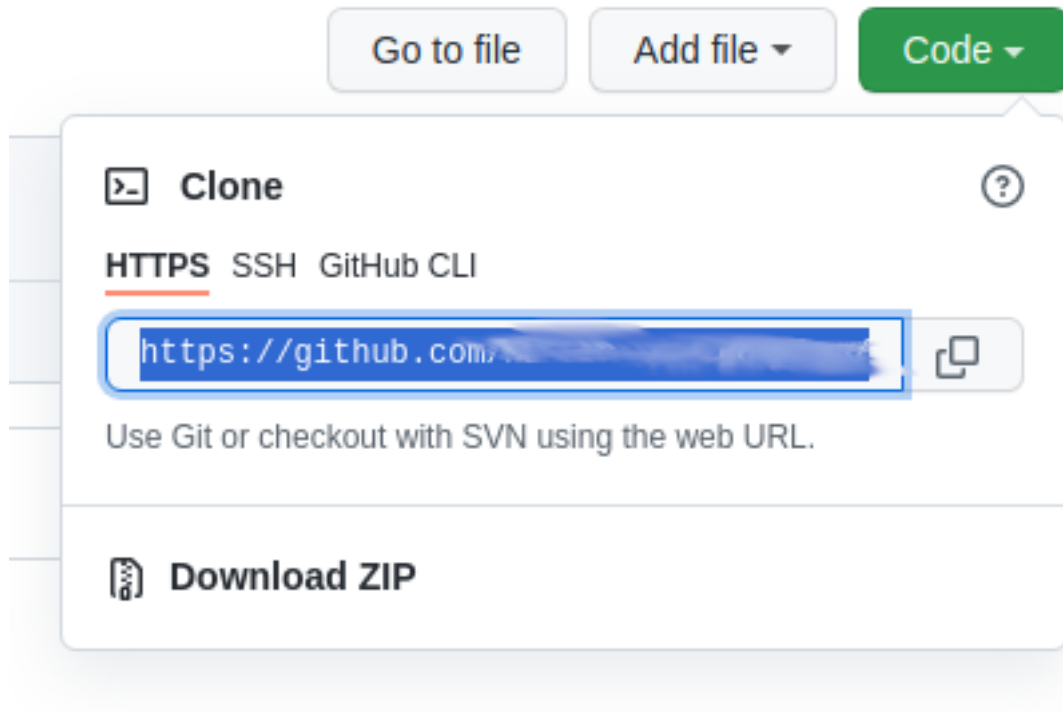
Skip this step if you're importing an existing repository.

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)
- ☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
- ☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Choose a **fitting** name and description and save it.

Store the path to your repository as a bookmark and store the git link where you can easily copy/paste from. In the front page of repository you click on “code”. In the pop-up there you’ll find your link. Ending on .git



After creating your repository and adding your public key to your GitHub-account, go to your bash and do **global** (per-user) config first, please. Note that the result of *git config --list* may have (many) more lines than shown.

```
git config --global user.name <your_github_username>
git config --global user.email <your_email@github>
git config --global core.sshcommand "ssh -i ~/.ssh/id_rsa_github -F /dev/null"
git config --list
ruser.name=<your_github_username>
user.email=<your_email@github>
core.sshcommand=ssh -i ~/.ssh/id_rsa_github -F /dev/null
```

For every repository you must then add a **local** origin like this:

```
git remote add origin git@github.com:<your_github_username>/<github_repos>.git
```

While you can set global parameters from anywhere you now must be inside the project directory!

terraform configuration

tbd

Some useful utilities in ~/bin

- For cygwin only: Put *terraform.exe* into ~/bin
- Change to ~/bin and enter *touch login.tf*. Edit *login.tf* to contain the following:

```
#!/bin/bash
az login --use-device-code --allow-no-subscriptions
az account list -o table --all --query "[].{TenantID: tenantId, Subscription: name, Default: isDefault}"
```

Save the file and enter *chmod 700 login.tf*.

- Change to ~/bin and enter *touch gitlogin.sh*. Edit *gitlogin.sh* to contain the following:

```
#!/bin/bash
ssh -i ~/.ssh/id_rsa_github -T git@github.com
```

Save the file and enter *chmod 700 gitlogin.sh*. Executing it should result in:

```
$ gitlogin.sh
Hi <username>! You've successfully authenticated, but GitHub does not provide shell access.
```