

## 01. [15p] Initial setup

### [5p] Google Cloud account

While there are many cloud providers (e.g.: AWS, Microsoft Azure, DigitalOcean, etc.), today we are going to use Google Cloud. By signing up here [<https://gcp.secure.force.com/GCPEDU?cid=7pKiq86GWNFWQ%2BCpTFn9pLUtjlfhChsYLM9v2VmPRz1Zuc9546iEOA1e97Br2Did/>] with your university email (`${YOUR_ID}@stud.acs.upb.ro` most likely), you will get \$50 in credits to play around with their infrastructure up until the 10<sup>th</sup> of April. In case you aren't forwarding your UPB emails to/from your main account via IMAP/POP [<https://support.google.com/mail/answer/7104828?hl=en>], you can access them in outlook [<https://outlook.office365.com/>].

### [5p] Google Cloud SDK

Normally, cloud providers give you the option to access a web dashboard [<https://console.cloud.google.com/>]. Using this is fine for starting up one virtual machine, or checking your billing settings. However, if you want to do some automation work, you will want to utilize the gcloud SDK [<https://cloud.google.com/sdk/docs/install>]. By following the instructions here, you should be able to install the **gcloud** CLI application. If the steps for Ubuntu don't work (very likely on WSL), try the more generic approach for Linux [<https://cloud.google.com/sdk/docs/install#linux>] (i.e.: download a `.tar.gz` and run the install script). Remember: Ubuntu is a Linux distribution! Finally, you will be asked to run:

```
# create default configuration for usage with gcloud
$ gcloud init
```

In addition to logging in with your Google account, you will also be asked to create a new project. Because of how Google Cloud is designed, this project must have a *globally unique* ID. So don't go for something obvious like `"lab-ii"`, but instead derive something from your university account name. That should be unique enough.

### [5p] Billing

CLI tools like **ip** and **gcloud** work based on a more modern paradigm. If older programs mainly use flags (i.e.: `- -this`, `-t`) to specify what functionality should be invoked at runtime, these use commands and subcommands [<https://clig.dev/#subcommands>] for a more intuitive classification. For example:

```
# <tool_name> <subject> <action>
$ gcloud projects list
PROJECT_ID      NAME                PROJECT_NUMBER
lab-radumantu   lab-radumantu       1234567890
```

If you're unsure what a command does, add `- -help` anywhere to get a manual page. Otherwise, check the web reference [<https://cloud.google.com/sdk/gcloud/reference>].

If you're still using **zsh**, first of all congrats. Second, try tab completion to get a list of possible commands based on what you've already written:

```
$ gcloud compute networks <TAB>
create          get-effective-firewalls  subnets
delete          list                     update
describe        peerings                 vpc-access
```

Note that sometimes, the feature that you want to access is in fact hidden behind the **alpha** or **beta** commands. These commands indicate that the development version of regular commands should be used in stead.

```
# try to list your configured billing accounts
$ gcloud billing accounts list
ERROR: (gcloud.billing) Invalid choice: 'accounts'.
This command is available in one or more alternate release tracks. Try:
  gcloud alpha billing accounts
  gcloud beta billing accounts

# now try the same thing with the alpha variant of the command
$ gcloud alpha billing accounts list
ACCOUNT_ID      NAME                                OPEN  MASTER_ACCOUNT_ID
XXXXXX-XXXXXX-XXXXXX Billing Account for Education  True
```

Now knowing your *project ID* and your *billing account ID*, it's time to link the two. This way, when you request resource allocation from the **gcloud compute** engine, Google will know to use the \$50 credit account (charges are usually made at the end of the month). While there's also an alpha version [<https://cloud.google.com/sdk/gcloud/reference/alpha/billing/accounts/projects/link>] of the following command, we'll be using the *recommended beta* variant:

```
# link billing account to gcloud project
$ gcloud beta billing projects link ${PROJECT_ID} --billing-account ${BILLING_ACC_ID}
```

Before we proceed to actually starting instances in different locations, here's one final config that we should do.

```
# set default project id
$ gcloud config set core/project ${PROJECT_ID}

# check that new value was saved
$ gcloud config list
[core]
...
project = ${PROJECT_ID}
```

Setting the *core/project* propriety is optional. However, not setting it would have meant that every time you invoked a **gcloud compute** command, you would have been required to also pass a `- -project=${PROJECT_ID}` flag. Which is annoying...