

Navodila za ustvarjanje virtualke na GCloud-u

[Link -> GCloud](#)

"Try for free" in se prijavimo z google računom. Potrebno je podatki bančni račun.

Accelerate your digital transformation with high-performance virtual machines. New customers get \$300 in free credits to spend on Google Cloud during the first 90 days. All customers get a general purpose machine (f1-micro instance) per month for free, not charged against your credits.

-> Bančna kartica je potrebna zaradi preverjanja identitete. Po poteku promocijskega obdobja, je potrebno posebej aktivirati če želite plačevati za virtualke.

Compute products [Contact](#)

Compute Engine

Accelerate your digital transformation with high-performance virtual machines.

New customers get \$300 in free credits to spend on Google Cloud during the first 90 days. All customers get a general purpose machine (f1-micro instance) per month for free, not charged against your credits.

[Try it free](#)

Benefits

- ✓ General purpose (E2, N1, N2, N2D) machines provide a good balance of price and performance
- ✓ Compute optimized (C2) machines offer high-end vCPU performance for compute-intensive workloads
- ✓ Memory optimized (M2) machines offer the highest memory and are great for in-memory databases
- ✓ Accelerator optimized (A2) machines are based on the A100 GPU, for very demanding applications

Use cases

- VM migration to Compute Engine
- Genomics data processing
- BYOL or use license-included images

Documentation

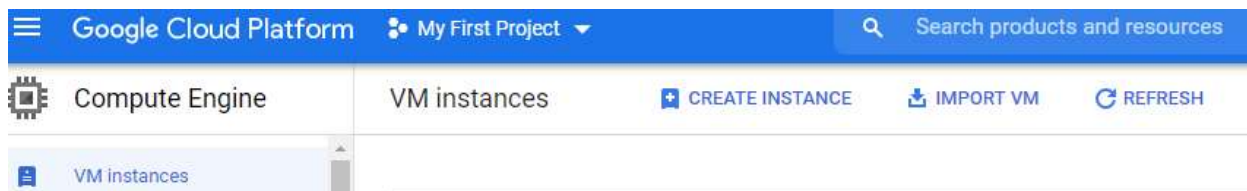
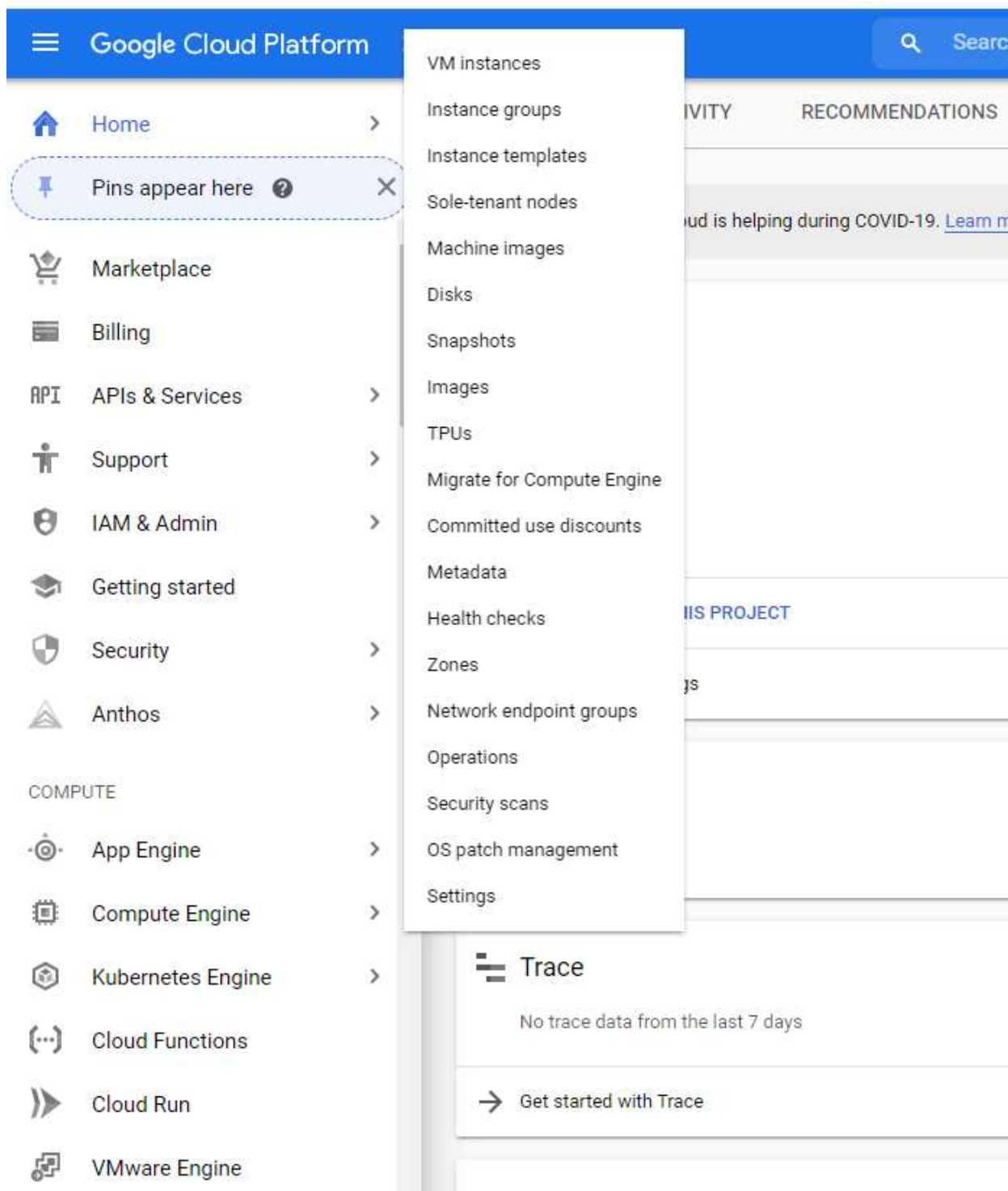
Take the next step

Gartner

Google is named a Leader Magic Quadrant for Cloud Infrastructure and Platform

[Get a copy of the report](#)

Usvarimo virtualno mašino. -> Izberemo VM instances



Name: python-tecaj (oziroma poljubno)

Region: Evropska regija (optimalno: europe-west-3)

Machine configuration: e2-standard-4 (oziroma poljubno)

My First Project

Search products and resources

ons:

atch

plate

hine

Name ?
Name is permanent

python-tecaj

Labels ? (Optional)

+ Add label

Region ?
Region is permanent

Zone ?
Zone is permanent

europa-west3 (Frankfurt)

europa-west3-c

Machine configuration

Machine family

General-purposeMemory-optimised

Machine types for common workloads, optimised for cost and flexibility


Series

E2

CPU platform selection based on availability

Machine type

e2-standard-4 (4 vCPU, 16 GB memory)

vCPU

4

Memory

16 GB

GPUs

-

CPU platform and GPU


Confidential VM service ?

☐ Enable the Confidential Computing service on this VM instance.

Container ?

☐ Deploy a container image to this VM instance. [Learn more](#)

Boot disk ?



New 10 GB standard persistent disk

Image

Debian GNU/Linux 10 (buster)

Change

Boot disk: Spremenimo v Ubuntu 20.04 LTS

file:///C:/Users/18ICTA2/Desktop/Iskraemeco/gcloud/G-Cloud virtualka namestitev.html

3/12

Size (GB): 40GB

Ostale nastavitve pustimo privzete.

Google Cloud Platform

Create an instance

To create a VM instance, select one of the options:

- New VM Instance**
Create a single VM instance from scratch
- New VM Instance from template**
Create a single VM instance from an existing template

Boot disk

Select an image or snapshot to create a boot disk, or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#).

Public images | Custom images | Snapshots | Existing disks

Operating system
Ubuntu

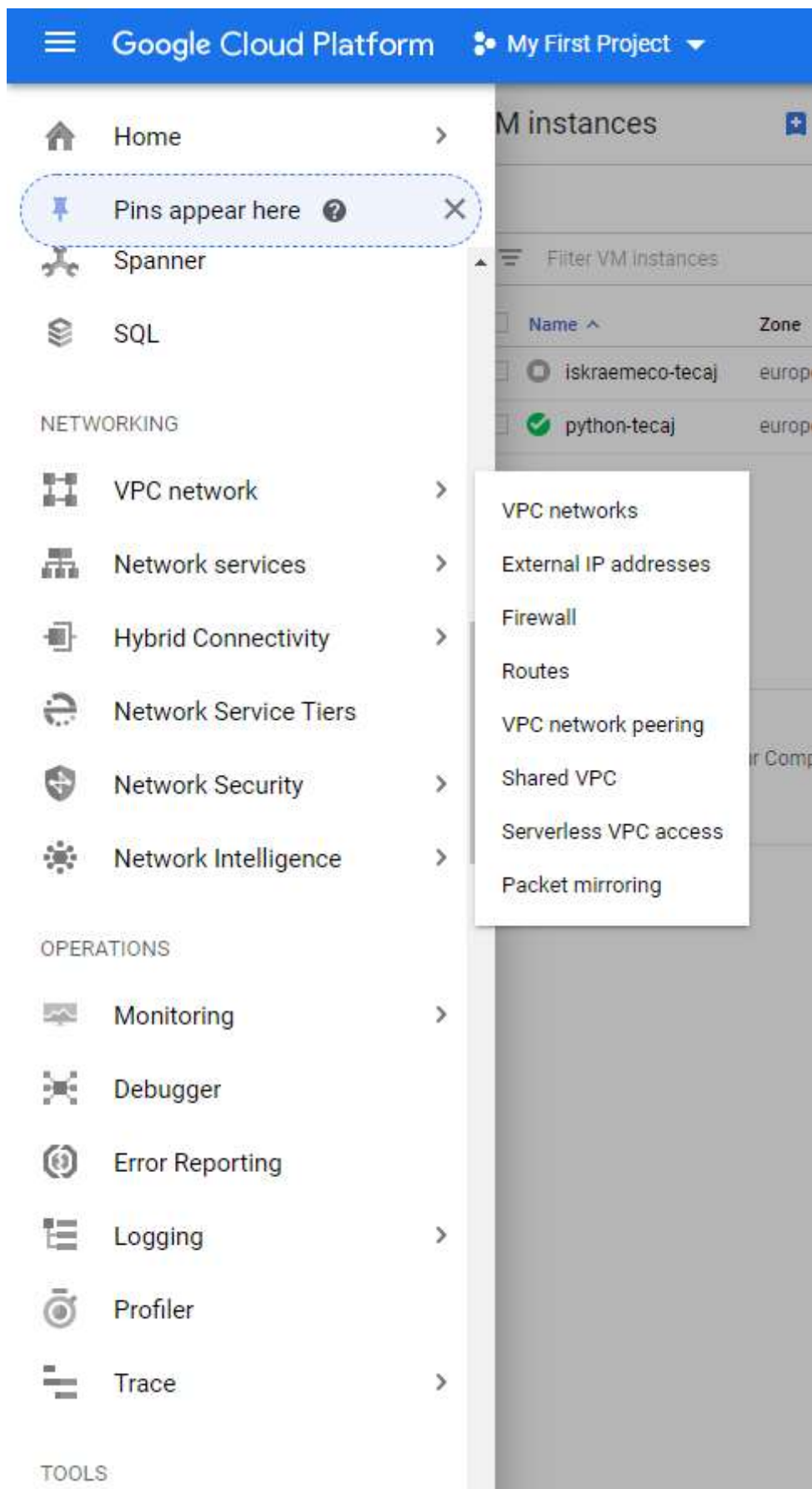
Version
Ubuntu 20.04 LTS

amd64 focal image built on 2020-10-28, supports Shielded VM features

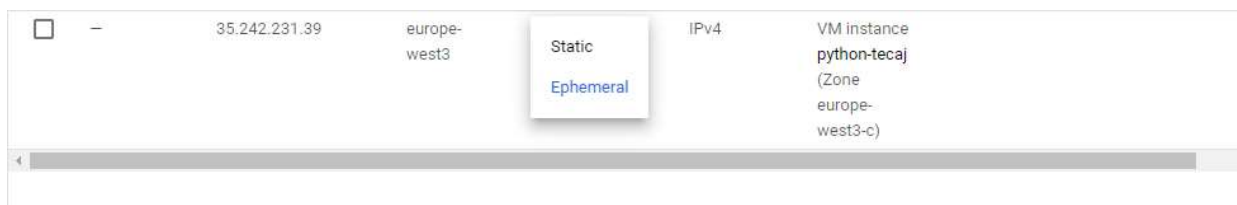
Boot disk type | Size (GB)

Standard persistent disk | 40

Odpremo Networking -> VPC network -> External IP addresses.



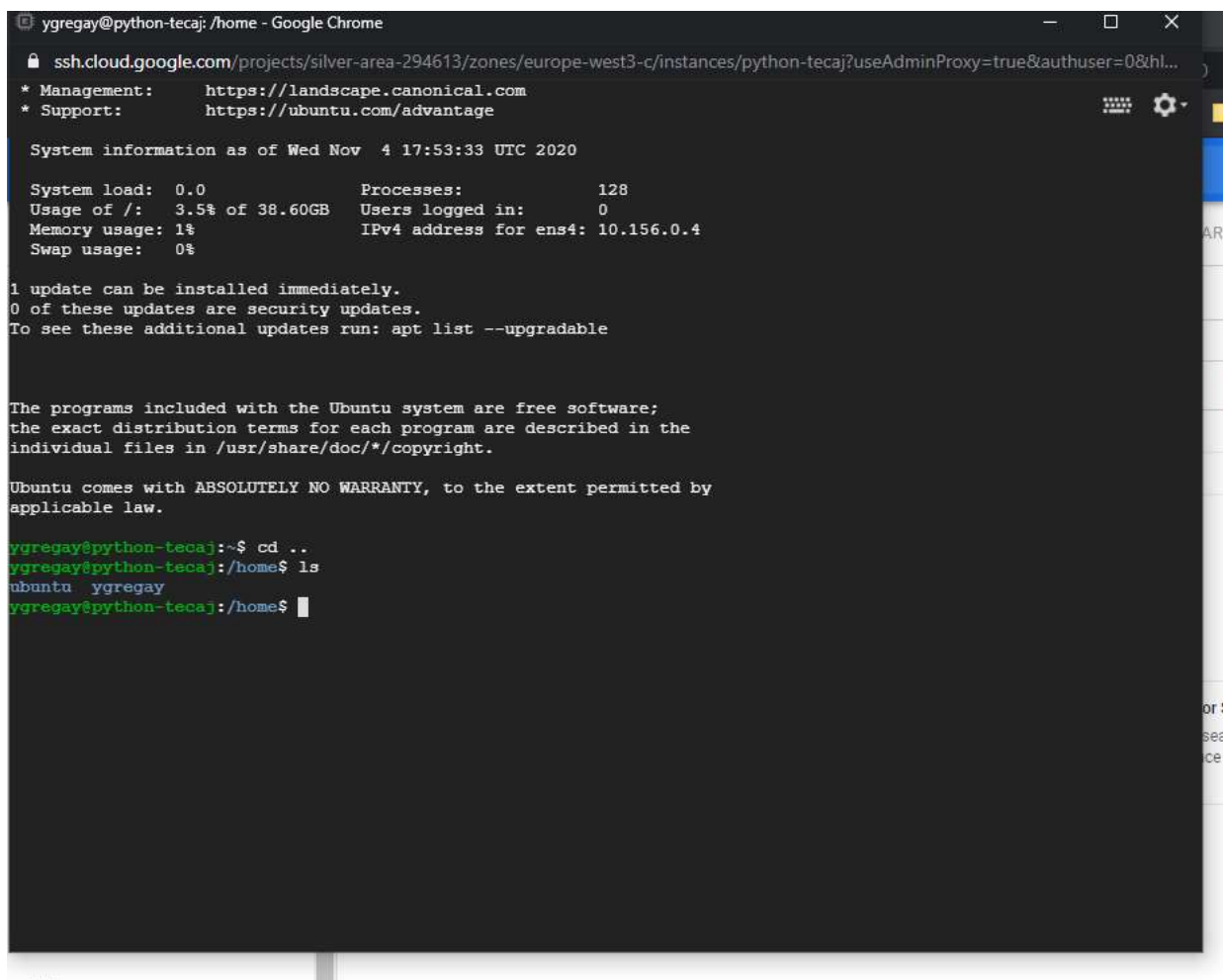
Tip naše virtualke spremenito iz *Ephemeral* v *Static*. Ime je lahko poljubno.



Sprva vzpostavimo SSH povezavo tako, da kliknemo gumb SSH, ki nam odpre konzolo v novem browser oknu.



Premaknemo se v *home* direktorij.



Ustvarimo novega uporabnika `sudo adduser <new_user>`


```

ygregay@python-tecaj:~$ cd ..
ygregay@python-tecaj:/home$ ls
ubuntu  ygregay
ygregay@python-tecaj:/home$ sudo adduser gregor
Adding user `gregor' ...
Adding new group `gregor' (1003) ...
Adding new user `gregor' (1002) with group `gregor' ...
Creating home directory `/home/gregor' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for gregor
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y

```

In ga premaknemo v sudo skupino. `sudo usermod -aG sudo <new_user>`

```

ygregay@python-tecaj:~$ sudo usermod -aG sudo gregor
ygregay@python-tecaj:~$

```

Zamenjamo na novonarejenega uporabnika. `su <new_user>`

In se premaknemo v njegovo mapo. `cd`

Ustvarimo mapo `.ssh`. `mkdir .ssh`

In se premaknemo v to mapo. `cd .ssh`

```

ygregay@python-tecaj:/home$ su gregor
Password:
gregor@python-tecaj:/home$ cd
gregor@python-tecaj:~$ ls
gregor@python-tecaj:~$ ls -la
total 20
drwxr-xr-x 2 gregor gregor 4096 Nov  4 17:56 .
drwxr-xr-x 5 root   root   4096 Nov  4 17:56 ..
-rw-r--r-- 1 gregor gregor  220 Nov  4 17:56 .bash_logout
-rw-r--r-- 1 gregor gregor 3771 Nov  4 17:56 .bashrc
-rw-r--r-- 1 gregor gregor  807 Nov  4 17:56 .profile
gregor@python-tecaj:~$ mkdir .ssh
gregor@python-tecaj:~$ cd .ssh
gregor@python-tecaj:~/.ssh$ ls -la
total 8
drwxrwxr-x 2 gregor gregor 4096 Nov  4 18:00 .
drwxr-xr-x 3 gregor gregor 4096 Nov  4 18:00 ..
gregor@python-tecaj:~/.ssh$

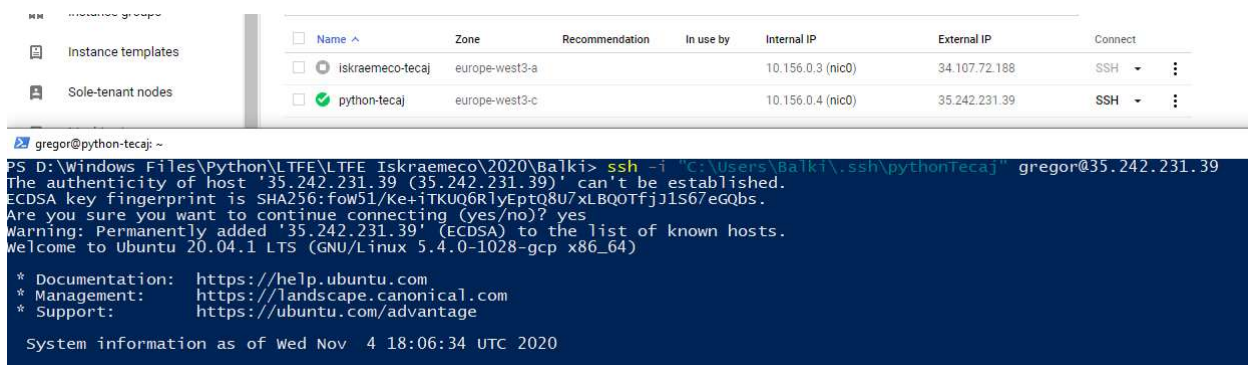
```

Ustvarimo nov file `authorized_keys`. `nano authorized_keys`

Vanj prilepimo public key (to je public key našaga računalnika s katerim se želimo povezati na virtualko).



Sedaj se lahko preko powershella povežemo. `ssh -i "<pot\do\nasega\kljuca>" <new_user>@<external_IP>` (pot do ključa je potrebna če ni v privzeti mapi)



Zaženemo sledeče ukaze

```
sudo apt-get update;
```



```
sudo apt-get install -y make build-essential libssl-dev zlib1g-dev libbz2-dev  
libreadline-dev libsqlite3-dev wget curl llvm libncurses5-dev libncursesw5-dev  
xz-utils tk-dev libffi-dev liblzma-dev python-openssl;
```

```
curl https://pyenv.run | bash;
```

```
nano ~/.bashrc;
```

Na konec datoteke prilepimo:

```
export PATH="$HOME/.pyenv/bin:$PATH"  
eval "$(pyenv init -)"  
eval "$(pyenv virtualenv-init -)"
```

```

gregor@python-tecaj: ~
GNU nano 4.8 /home/gre
*)
;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminati

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export PATH="$HOME/.pyenv/bin:$PATH"
eval "$(pyenv init -)"
eval "$(pyenv virtualenv-init -)"

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text
^X Exit          ^R Read File    ^_ Replace      ^U Paste Text

```

Zaženemo:

```
exec "$SHELL"
```

Zaženemo komande:

```
pyenv install -v 3.9.0;
```

```
pyenv global 3.9.0;
```

Ustvarimo novo mapo, kjer bomo imeli gradiva.

```
mkdir tecaj;
```

In se premaknemo v to mapo.

```
cd tecaj;
```

V mapo kloniramo Git. `git clone https://github.com/leon11s/python-napredni-public.git`;

Kopiramo kloniran git repositorij v našo mapo. `cp -r python-napredni-public <nova_mapa>`

```
gregor@python-tecaj:~$ mkdir tecaj
gregor@python-tecaj:~$ cd tecaj
gregor@python-tecaj:~/tecaj$ ls
gregor@python-tecaj:~/tecaj$ git clone https://github.com/leon11s/python-napredni-public.git
Cloning into 'python-napredni-public'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 40 (delta 11), reused 37 (delta 8), pack-reused 0
Unpacking objects: 100% (40/40), 7.28 KiB | 677.00 KiB/s, done.
gregor@python-tecaj:~/tecaj$ ls -la
total 12
drwxrwxr-x 3 gregor gregor 4096 Nov  4 18:27 .
drwxr-xr-x 6 gregor gregor 4096 Nov  4 18:27 ..
drwxrwxr-x 6 gregor gregor 4096 Nov  4 18:27 python-napredni-public
gregor@python-tecaj:~/tecaj$ cp -r python-napredni-public moja_mapa
gregor@python-tecaj:~/tecaj$ ls -la
total 16
drwxrwxr-x 4 gregor gregor 4096 Nov  4 18:28 .
drwxr-xr-x 6 gregor gregor 4096 Nov  4 18:27 ..
drwxrwxr-x 6 gregor gregor 4096 Nov  4 18:28 moja_mapa
drwxrwxr-x 6 gregor gregor 4096 Nov  4 18:27 python-napredni-public
gregor@python-tecaj:~/tecaj$
```

Premaknemo se v novo mapo. `cd <nova_mapa>`

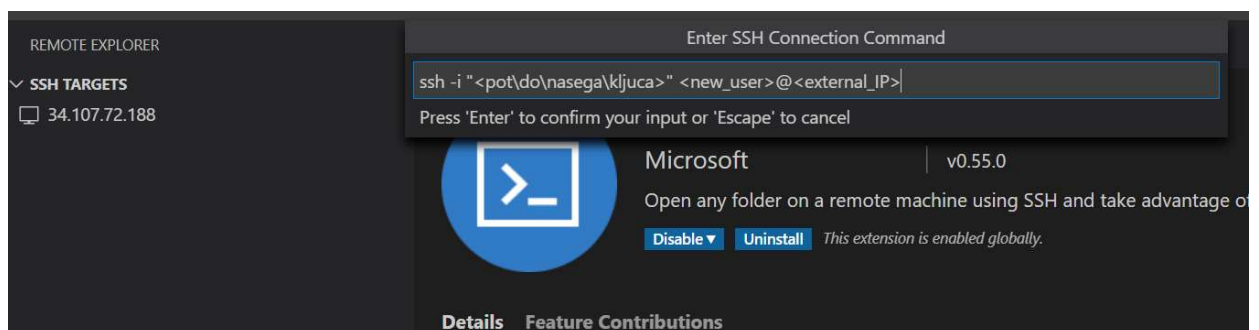
Namestimo venv: `python -m pip install virtualenv`

Ustvarimo virtualno okolje: `python -m venv .venv`

In ga zaženemo: `source .venv/bin/activate`

Upgradamo pip: `python -m pip install --upgrade pip`

Odpremo visual studio code in instaliramo Remote-SSH extension



Za ustvariti ssh ključ na windows naredimo:

ssh-keygen

"C:\Users\Gregor\.ssh\imeKljuca"