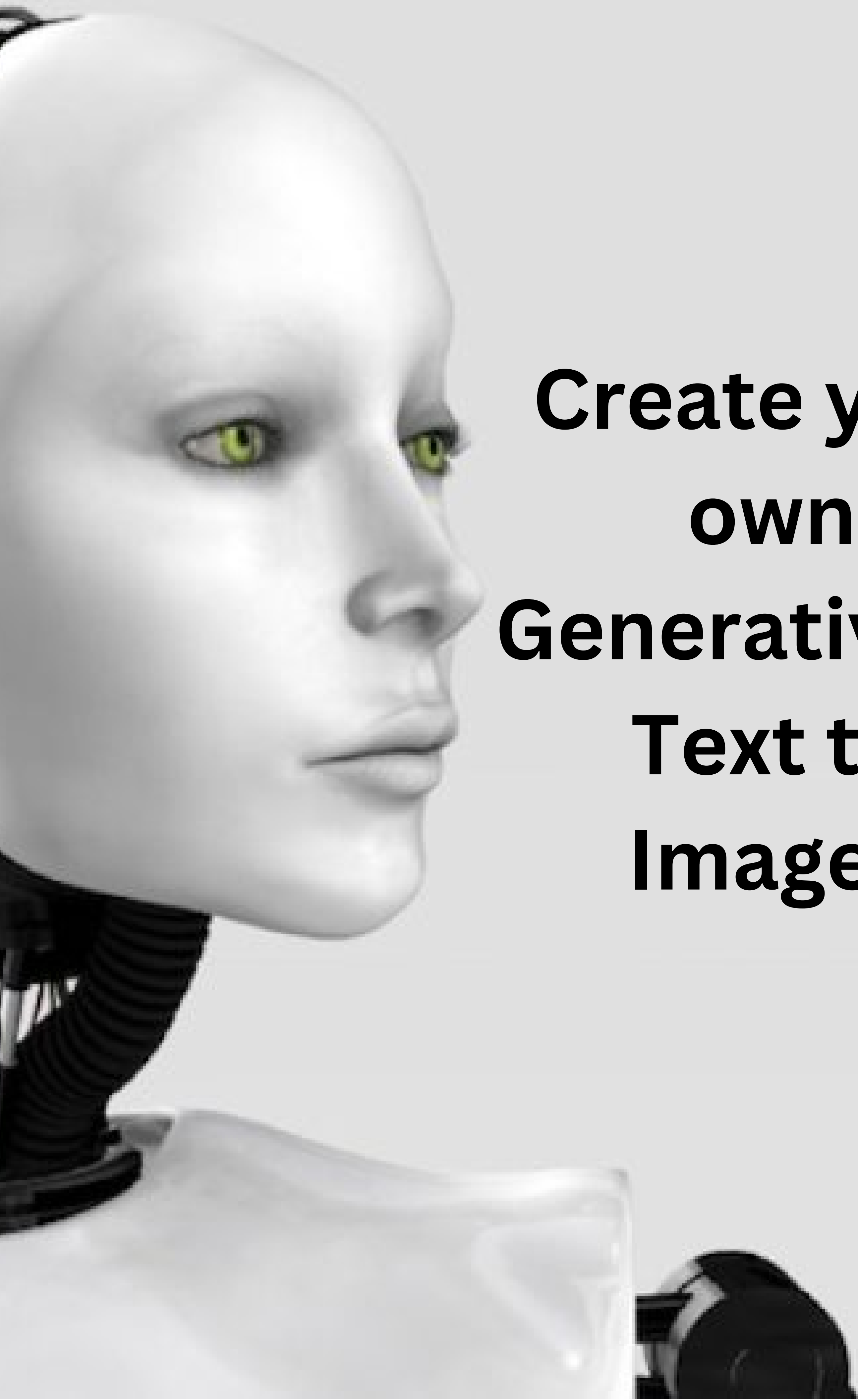
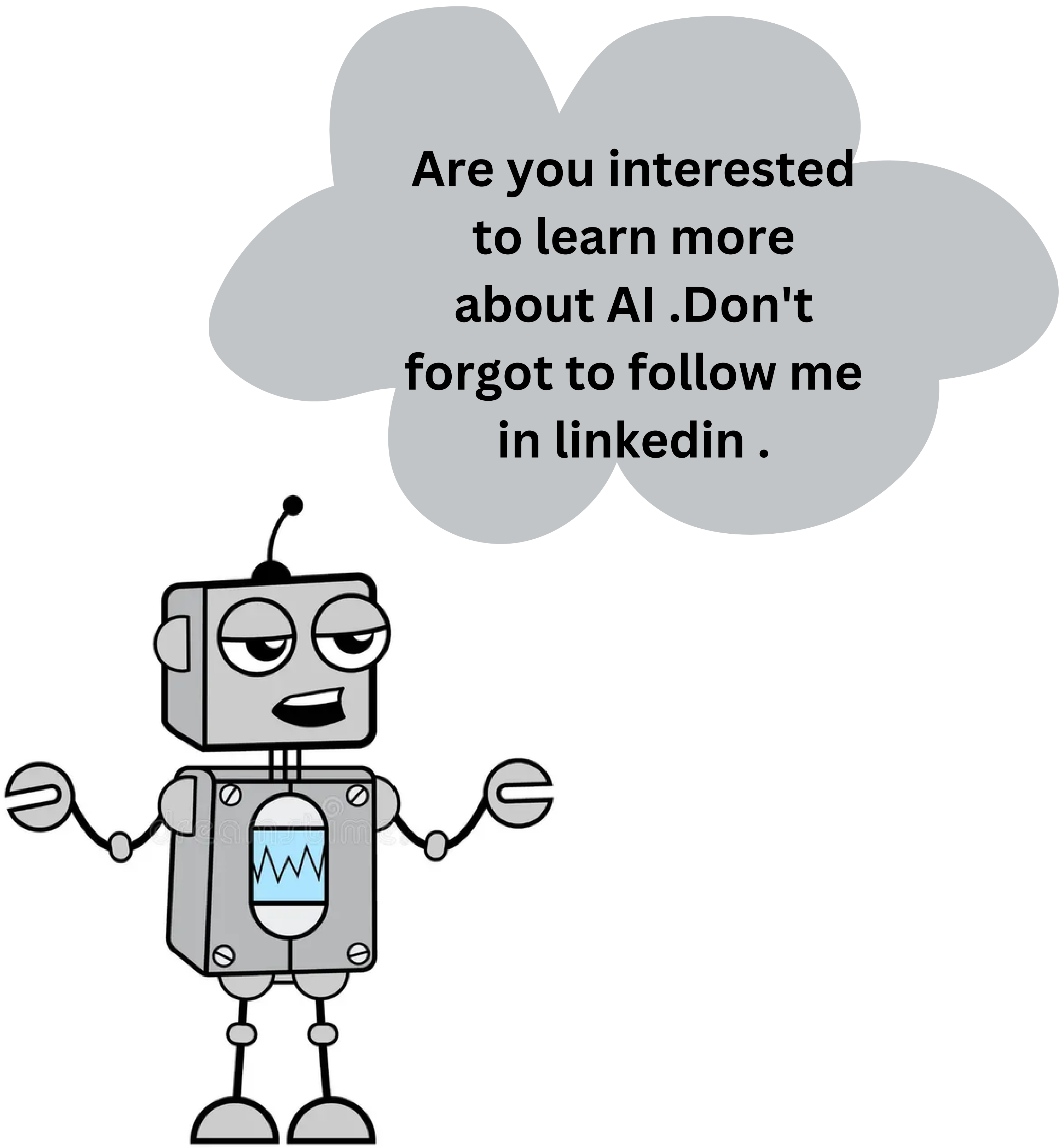


Create your own Generative AI Text to Images


Are you interested to learn more about AI .Don't forgot to follow me in linkedin .

What is stable diffusion ?

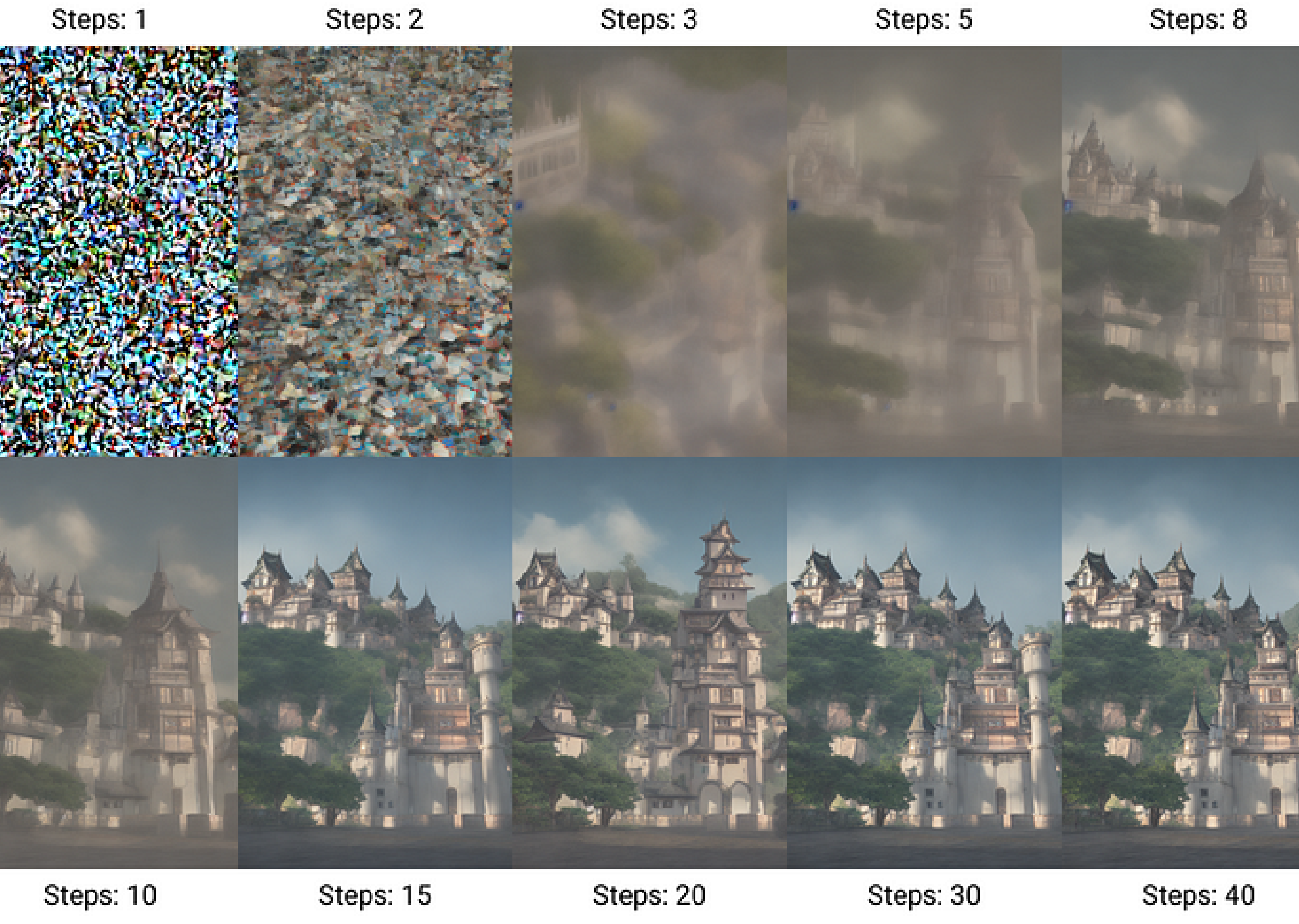
- Stable Diffusion is a method used in AI text-to-image generation that uses a diffusion model to create images from text descriptions.
- The diffusion model starts with a random image and then gradually adds noise to it, step by step.
- The noise is added in a controlled way, so that the image remains recognizable.

- At each diffusion step, the image becomes more refined, with the details becoming clearer.

- This process continues for several diffusion steps until the generated image is considered “stable”, meaning we’ve hit a point where further iterations aren’t likely to improve it.

- The process is shown below:

Process of Stable Diffusion

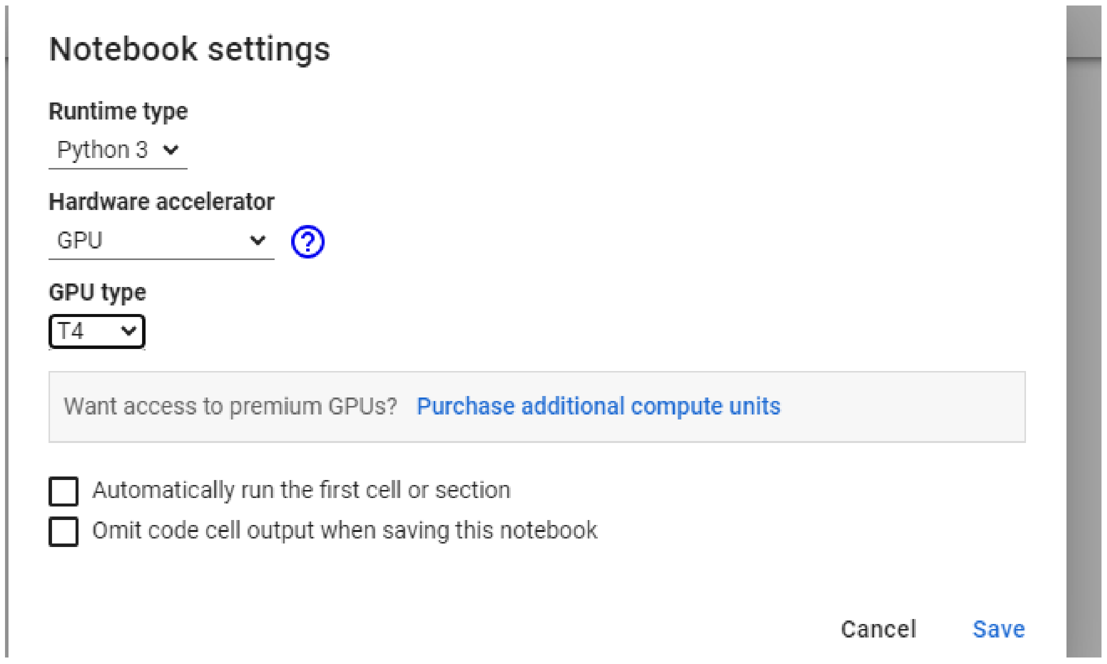


Let's move to Coding Part →

Prerequisites:

- Google account to sign into Colab and save notebooks
- Hugging Face account and an API token (free)

- Go to https://colab.research.google.com/
- From the file menu select Runtime -> Change Runtime Type, and select Hardware accelerator = GPU.



Notebook settings

Runtime type: Python 3

Hardware accelerator: GPU

GPU type: T4

Want access to premium GPUs? Purchase additional compute units

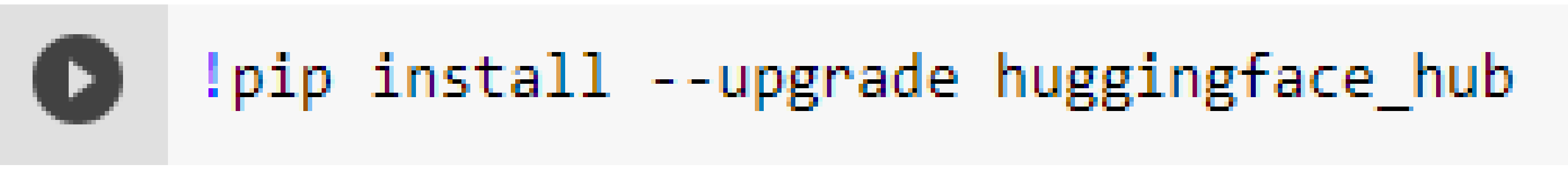
☐ Automatically run the first cell or section

☐ Omit code cell output when saving this notebook

- Click "save" .

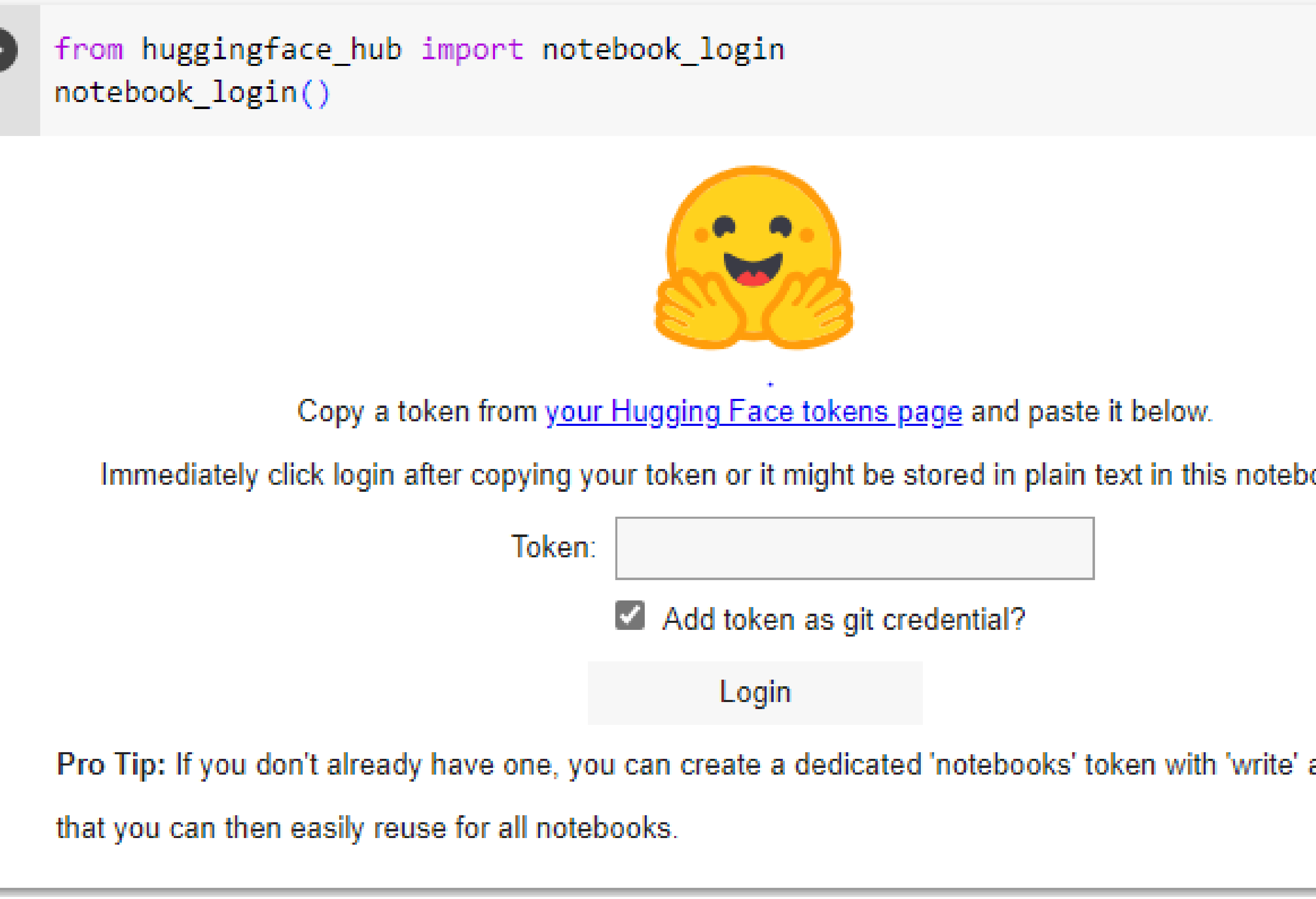
- Click on *Connect/Reconnect -> Connect to a hosted runtime*:

- Once you’ve connected to your runtime, we need to install some python libraries and authenticate using an API token:



```
!pip install --upgrade huggingface_hub
```

- Next we need to authenticate using a token from you HuggingFace account, when you execute the code below, you will be prompted in your notebook to paste in your HuggingFace API token:



```python
from huggingface_hub import notebook_login
notebook_login()
```

Copy a token from your Hugging Face tokens page and paste it below.

Immediately click login after copying your token or it might be stored in plain text in this notebo

Token:

☑ Add token as git credential?

Pro Tip: If you don't already have one, you can create a dedicated 'notebooks' token with 'write' a

that you can then easily reuse for all notebooks.

- Now click the "Your Hugging face token page "
- Generate "new token" .
- Copy the new token and paste it in token Field .
- Then click login

untitled1

July 1, 2023

we need to install some python libraries and authenticate using an API token:

```
[1]: !pip install --upgrade huggingface_hub
```

```
Collecting huggingface_hub
  Downloading huggingface_hub-0.15.1-py3-none-any.whl (236 kB)
                           236.8/236.8

kB 9.7 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from huggingface_hub) (3.12.2)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages
(from huggingface_hub) (2023.6.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from huggingface_hub) (2.27.1)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-
packages (from huggingface_hub) (4.65.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from huggingface_hub) (6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface_hub) (4.6.3)
Requirement already satisfied: packaging>=20.9 in
/usr/local/lib/python3.10/dist-packages (from huggingface_hub) (23.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub)
(1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub)
(2023.5.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub)
(2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->huggingface_hub) (3.4)
Installing collected packages: huggingface_hub
Successfully installed huggingface_hub-0.15.1
```

0.1 Next we need to authenticate using a token from you HuggingFace account, when you execute the code below, you will be prompted in your notebook to paste in your HuggingFace API token:

```
[2]: from huggingface_hub import notebook_login
     notebook_login()
```

```
VBox(children=(HTML(value='<center> <img\nsrc=https://huggingface.co/front/
 ↪assets/huggingface_logo-noborder.sv…
```

After successfully logging into your HuggingFace account, we're going download the diffusers and transformers python libraries:

```
[3]: !pip install -qq -U diffusers transformers
```

```
                                        1.1/1.1 MB
27.6 MB/s eta 0:00:00
                                        7.2/7.2 MB
83.8 MB/s eta 0:00:00
                                        7.8/7.8 MB
80.5 MB/s eta 0:00:00
                                        1.3/1.3 MB
70.8 MB/s eta 0:00:00
```

We need to create a StableDiffusion model pipeline so we can basically pass the model some text and have it generate an image based on that prompt. You might notice that one of the parameters we're passing is a path to a Stable Diffusion model hosted on HuggingFace. The examples in this post were tested using v1.5, you can try swapping for the latest (v2.1) at the time of writing:

```
[5]: from diffusers import StableDiffusionPipeline
     pipe = StableDiffusionPipeline.from_pretrained('runwayml/stable-diffusion-v1-5')
```

```
The cache for model files in Transformers v4.22.0 has been updated. Migrating
your old cache. This is a one-time only operation. You can interrupt this and
resume the migration later on by calling `transformers.utils.move_cache()`.

0it [00:00, ?it/s]

Downloading (…)ain/model_index.json:   0%|          | 0.00/541 [00:00<?, ?B/s]

Fetching 15 files:   0%|          | 0/15 [00:00<?, ?it/s]

Downloading model.safetensors:   0%|          | 0.00/492M [00:00<?, ?B/s]

Downloading (…)tokenizer/merges.txt:   0%|          | 0.00/525k [00:00<?, ?B/s]

Downloading (…)rocessor_config.json:   0%|          | 0.00/342 [00:00<?, ?B/s]

Downloading (…)_encoder/config.json:   0%|          | 0.00/617 [00:00<?, ?B/s]

Downloading (…)cial_tokens_map.json:   0%|          | 0.00/472 [00:00<?, ?B/s]

Downloading (…)_checker/config.json:   0%|          | 0.00/4.72k [00:00<?, ?B/s]
Downloading (…)cheduler_config.json:   0%|          | 0.00/308 [00:00<?, ?B/s]

Downloading model.safetensors:   0%|          | 0.00/1.22G [00:00<?, ?B/s]

Downloading (…)tokenizer/vocab.json:   0%|          | 0.00/1.06M [00:00<?, ?B/s]

Downloading (…)e6a/unet/config.json:   0%|          | 0.00/743 [00:00<?, ?B/s]

Downloading (…)8e6a/vae/config.json:   0%|          | 0.00/547 [00:00<?, ?B/s]

Downloading (…)okenizer_config.json:   0%|          | 0.00/806 [00:00<?, ?B/s]

Downloading (…)ch_model.safetensors:   0%|          | 0.00/3.44G [00:00<?, ?B/s]

Downloading (…)ch_model.safetensors:   0%|          | 0.00/335M [00:00<?, ?B/s]

Cannot initialize model with low cpu memory usage because `accelerate` was not
found in the environment. Defaulting to `low_cpu_mem_usage=False`. It is
strongly recommended to install `accelerate` for faster and less memory-intense
model loading. You can do so with:
```
pip install accelerate
```
.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["id2label"]` will be overriden.
```

[6]:

```python
import torch

# Initialize a prompt
prompt ="a peaceful beach at sunset"

# Pass the prompt in the pipeline
pipe(prompt).images[0]
```

```
  0%|          | 0/50 [00:00<?, ?it/s]
```

[6]: