

# WEBEX Amplified

---

Enabling Hybrid Work with Webex and ThousandEyes

Omer Ilyas

Omer Ilyas - Technical Marketing Engineer - oilyas@cisco.com

## Table of contents

---

1. Lab	3
1.1 Overview - Understanding AI and Its Integration with Webex	3
1.2 Pre-Requisites and Setup	4
1.3 Task 3: Log into the Lab Environment	17
1.4 Task 4: Install the ThousandEyes EPA	0
1.5 Task 5: Configure ThousandEyes Automated Session tests and EPA Monitoring	0
1.6 Analyzing ThousandEyes EPA Data	0
1.7 Understanding Fine-Tuning - Large Language Models	0
1.8 Deep Dive into LoRA and SFT	0
1.9 Fine Tuning - High Level Diagram	0
1.10 Logging Out and Ending the Lab Session	0
1.11 Conclusion	0

# 1. Lab

---

## 1.1 Overview - Understanding AI and Its Integration with Webex

---

Artificial Intelligence (AI) is transforming the way we work, enabling innovative solutions and enhancing productivity. This lab will guide you through the fundamental concepts of AI, various techniques, and how AI can be integrated with Webex to create efficient workflows.

### 1.1.1 Upon completion of this lab you will be able to

---

- Understand the basics of AI and its applications
- Learn about embedding techniques
- Explore vector databases
- Gain insights into Generative AI models
- Familiarize yourself with different AI frameworks
- Integrate AI with Webex to create seamless workflows
- Develop hands-on skills through practical exercises
- Understand Fine-tuning and Quantization
- Deploy Fine Tuning techniques

### 1.1.2 Prerequisites

---

- Basic understanding of AI concepts is helpful but not required.

### 1.1.3 Disclaimer

---

The lab design and configuration examples provided are for reference purposes only. This is a sample deployment, and not all recommended features are used or enabled optimally. For design-related questions, please contact your representative at Cisco or a Cisco partner or TME team.

### 1.1.4 Lab Overview - Enabling Hybrid Work with ThousandEyes

---

- Lab Login and Setup
- Quick AI Overview
- Configure and Access the Lab Systems
- Deploy AI Models and Techniques
- Set up and Configure AI Monitoring
- Review AI Integration with Webex
- Analyze Data and Optimize Workflows
- Wrap up and End the Lab

Let's get started! Click on Task 1 – Google Collab- Accessing Google Collab and creating account.

## 1.2 Pre-Requisites and Setup

---

### 1.2.1 Google Collab- Accessing Google Collab and creating account

---

Google Colab is a free, cloud-based platform that provides a convenient environment for running notebooks. If you want to create a machine learning model but don't have a computer that can handle the workload, Google Colab is the platform for you. In our lab, we will be using Google Colab to test and run our code. However, if you have your own Python environment and prefer to run the code on your local machine, please feel free to do so.

Here are some reasons why using Google Colab can be beneficial for this lab:

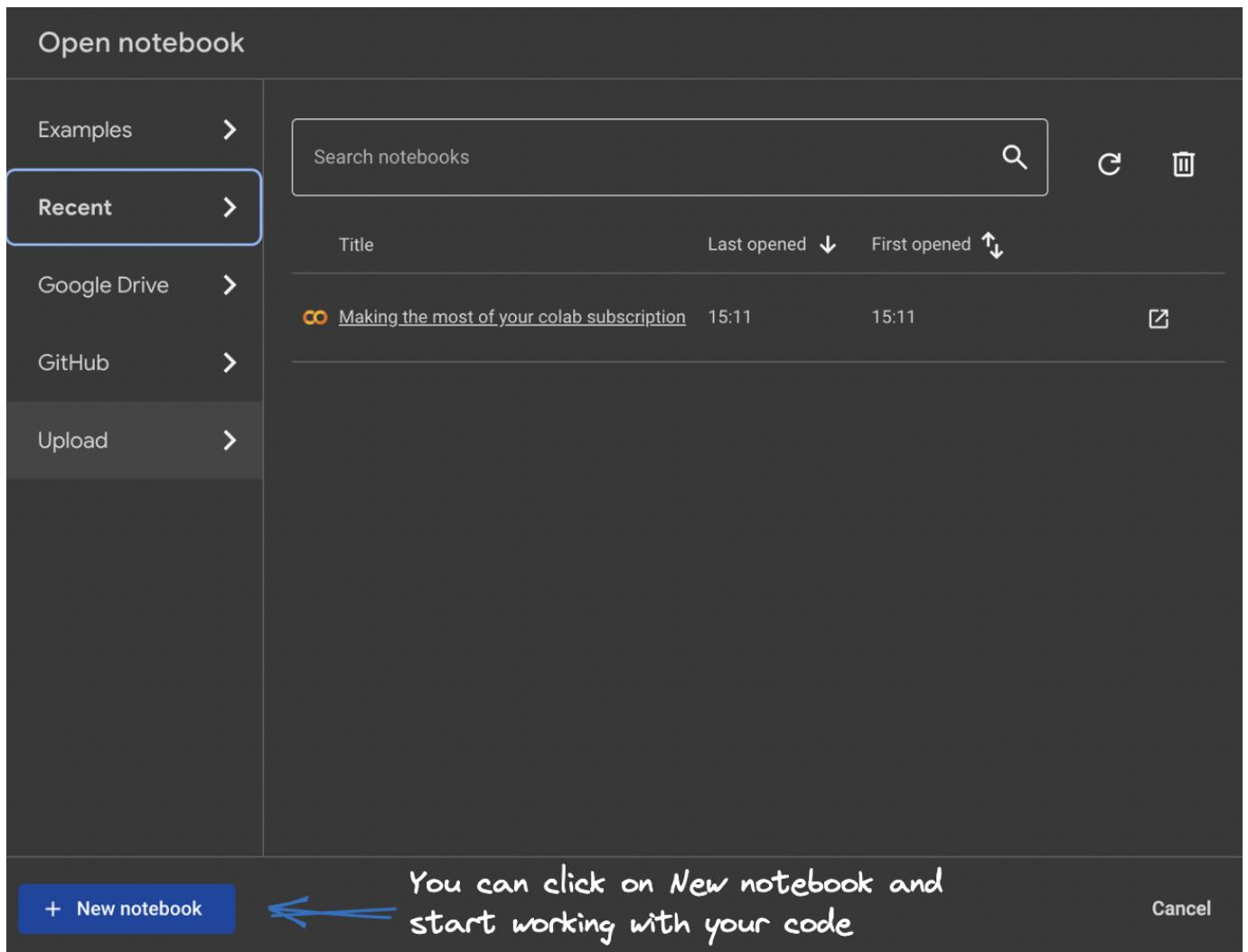
- Free Access to GPUs and TPUs: Google Colab offers free access to powerful GPUs and TPUs, which can significantly accelerate the training and fine-tuning of machine learning models.
- No Setup Required: With Colab, there is no need to set up your local environment. Everything runs in the cloud, which saves time and avoids configuration issues.
- Easy Collaboration: Colab notebooks can be easily shared and collaborated on with team members, making it an ideal tool for collaborative projects.
- Integration with Google Drive: Colab integrates seamlessly with Google Drive, allowing you to save and manage your work conveniently.
- Pre-installed Libraries: Many popular machine learning libraries, including TensorFlow and PyTorch, come pre-installed in Colab, making it easy to start working on your projects immediately.

### 1.2.2 Getting Started With Google Colab

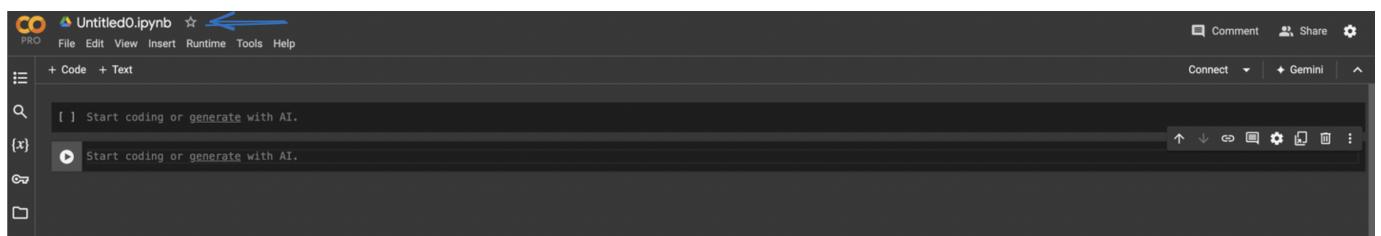
---

To start working with Google Collaboratory Notebook you first need to log in to your Google account, then go to this link [Google Colab](#)

- Create a new Jupyter Notebook



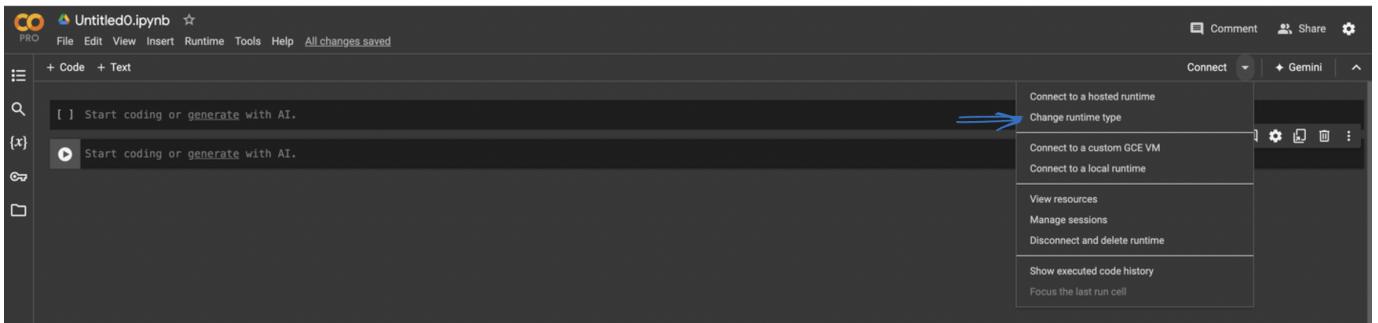
- On creating a new notebook, it will create a Jupyter notebook with Untitled0.ipynb and save it to your google drive in a folder named Colab Notebooks. Now as it is essentially a Jupyter Notebook, all commands of Jupyter Notebooks will work here.



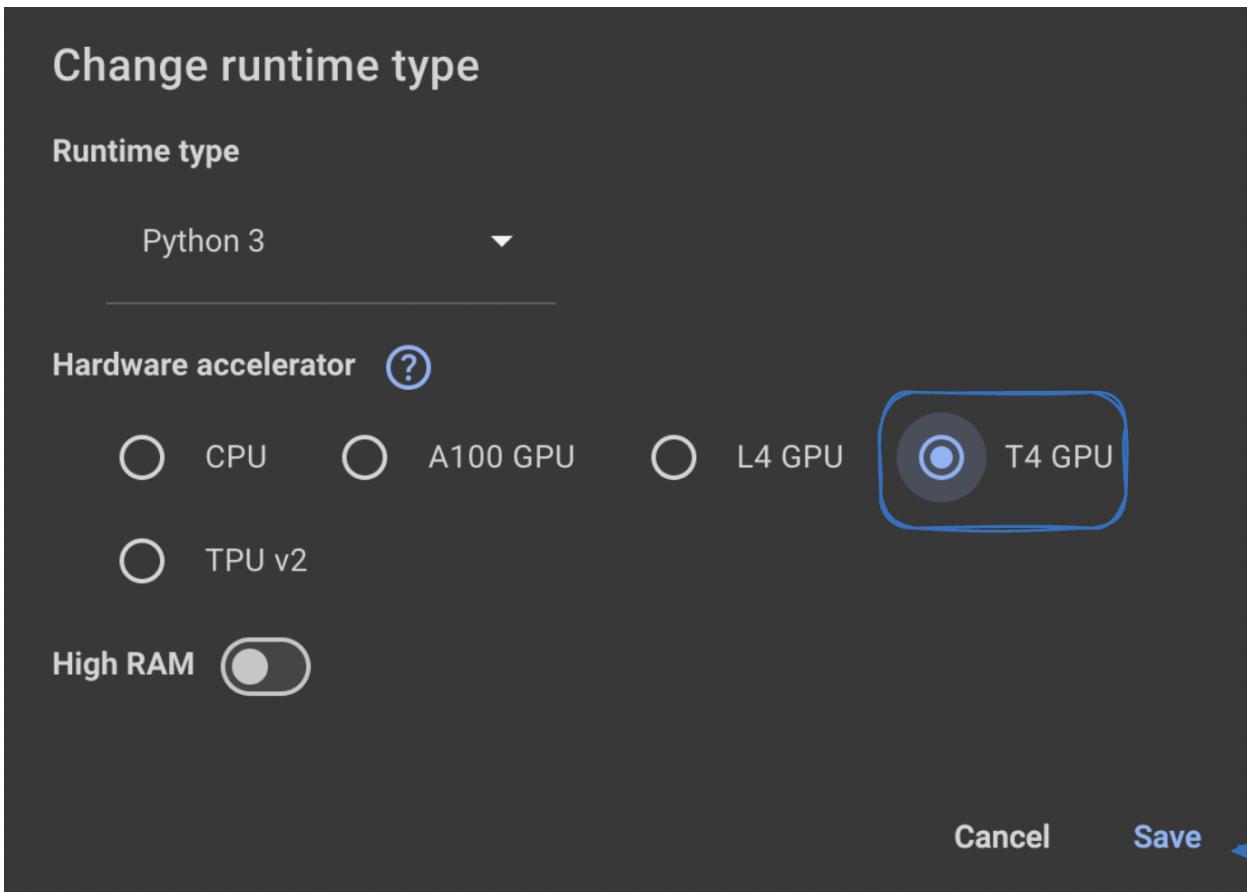
- There might be times when we need to fine-tune models or perform specific tasks that require changing the runtime environment in Colab. Google Colab offers different runtime environments that can be selected based on your requirements:
- Python Versions: You can select between different versions of Python (e.g., Python 2 or Python 3) depending on the compatibility of the code and libraries. We will be using Python3 for our lab.
- Hardware Accelerators: Colab provides access to hardware accelerators, which can be particularly useful for intensive computations. You can choose between:

```
None: No hardware acceleration, suitable for basic tasks.  
GPU: Accelerate your computations with a Graphics Processing Unit.  
TPU: Use a Tensor Processing Unit for even faster performance, especially beneficial for deep learning tasks.
```

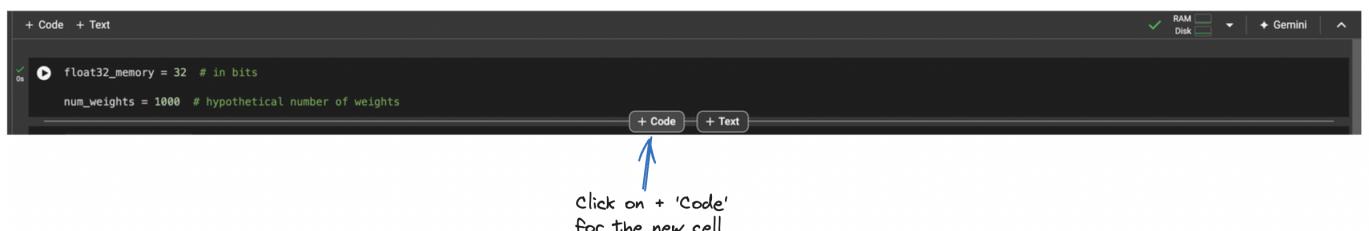
- Change Runtime Environment: Click the “Runtime” dropdown menu at the top of the Colab interface.



- Select “Change runtime type”: This will open a dialog box where you can configure the runtime environment.
- Select Python Version: Choose Python 3 from the “Runtime type” dropdown menu.
- Select Hardware Accelerator: From the “Hardware accelerator” dropdown menu, choose GPU, or TPU based on your needs.



- Save Settings: Click “Save” to apply the changes.
- New Cell: Whenever you want to copy the code in Google Colab and run it, be sure to click on + Code to add a new code cell.



- Execute Code: Click the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter" while the cell is selected.



### 1.2.3 Notes: On GPU and TPU Access:

While Google Colab offers free access to GPUs and TPUs, there are limitations. For more consistent access to high-performance GPUs and TPUs, you might need to subscribe to Colab Pro or Colab Pro+ accounts. These paid plans provide priority access to better hardware, longer runtimes, and more memory.

### 1.2.4 Using Huggingface Hub to share our Datasets

In this lab, we will be utilizing the Hugging Face Hub to load our custom datasets. Hugging Face provides an extensive repository of datasets that can be easily integrated into your machine learning workflows. For the purposes of this lab, we will demonstrate how to access/upload and use our custom datasets effectively.

However, when fine-tuning models in your own work environment, especially if you are using private data, there are important considerations to keep in mind:

- Private Datastores: If you are working with proprietary or sensitive data, it is crucial to use your organization's secure datastores. Ensure that all data handling complies with your organization's data privacy policies and regulations.
- Hugging Face Datasets: If you prefer to use Hugging Face for dataset storage and management, make sure to mark your datasets as private. This setting ensures that your data cannot be accessed by anyone outside your organization, maintaining the confidentiality and integrity of your information. Please refer to Huggingface documentation for more info.

Few more Consideration

- Upload Dataset: When uploading your dataset to Hugging Face, choose the appropriate privacy settings. You can set your dataset to private during the upload process.
- Check Permissions: Regularly review and manage the permissions of your datasets to ensure they remain private and secure.
- Collaborator Access: If you need to share the dataset with specific team members, use the Hugging Face interface to grant access to trusted collaborators only.

By following these guidelines, you can ensure that your data remains secure while leveraging the powerful tools and resources provided by Hugging Face. This approach not only enhances your workflow efficiency but also upholds the best practices in data security and privacy.

### 1.2.5 Accessing Hugging Face Hub and creating account

Hugging Face can be accessed by browsing to [huggingface.co](https://huggingface.co)

The screenshot shows the Hugging Face Hub homepage. At the top, there's a search bar with placeholder text "Search models, datasets, users...". Below the search bar, there are navigation links for "Models", "Datasets", "Spaces", "Posts", "Docs", "Solutions", "Pricing", "Log In", and "Sign Up". A banner at the top left says "NEW! AI Tools are now available in HuggingChat". The main content area features a large yellow emoji of a smiling face. Below it, the text "The AI community building the future." is displayed in a large, bold, white font. To the right, there's a sidebar with categories like "Tasks", "Libraries", "Datasets", "Languages", "Licenses", and "Other". Under "Tasks", there are sections for "Multimodal" (Text-to-Image, Image-to-Text, Text-to-Video, Visual Question Answering, Document Question Answering, Graph Machine Learning), "Computer Vision" (Depth Estimation, Image Classification, Object Detection, Image Segmentation, Image-to-Image, Unconditional Image Generation, Video Classification, Zero-Shot Image Classification), "Natural Language Processing" (Text Classification, Token Classification, Table Question Answering, Question Answering, Zero-Shot Classification, Translation, Summarization, Conversational, Text Generation, Text2Text Generation, Sentence Similarity), "Audio" (Text-to-Speech, Automatic Speech Recognition, Audio-to-Audio, Audio Classification, Voice Activity Detection), "Tabular" (Tabular Classification, Tabular Regression), and "Reinforcement Learning" (Reinforcement Learning, Robotics). On the right side, there's a list of models with their names, descriptions, and statistics. Some examples include "meta-llama/Llama-2-70b", "stabilityai/stable-diffusion-xl-base-0.9", "openchat/openchat", "Illyasvi1/ControlNet-v1-1", "cerspense/zeroscope\_v2\_XL", "meta-llama/Llama-2-13b", "tiiuae/falcon-40b-instruct", "WizardLM/WizardCoder-15B-V1.0", "CompVis/stable-diffusion-v1-4", "stabilityai/stable-diffusion-2-1", "Salesforce/xgen-7b-8k-inst", and "THUDM/llama-7b-hf".

### Signing up

- Browse to Hugging Face home page and click on Sign up. Follow the instructions as per below images

The screenshot shows the "Join Hugging Face" sign-up form. It has a yellow emoji at the top. The form asks "Join the community of machine learners!". It has fields for "Email Address" (containing "omer.lyas@boldbetz.com") and "Password" (containing "\*\*\*\*\*"). Below the password field are three validation rules: "Must contain at least 8 characters", "Must contain uppercase, lowercase letters, and numbers", and "If less than 12 characters, must contain a special character". At the bottom of the form are "Next" and "Already have an account? Log in" buttons. Blue arrows point from the text "Follow the instructions as per below images" to the "Email Address" field, the "Password" field, and the "Next" button.

# Complete your profile

One last step to join the community

Username

Full name

Avatar (optional)

Twitter username (optional)

Upload file

Twitter account

GitHub username (optional)

LinkedIn profile (optional)

GitHub username

LinkedIn profile

Homepage (optional)

Homepage

AI & ML interests (optional)

AI & ML interests

I have read and agree with the [Terms of Service](#) and the [Code of Conduct](#)

**Create Account**

- Please check your email address for a confirmation link and click to verify your account

Your email address has been verified successfully.

## Join an organization

Hugging Face is way more fun with friends and colleagues!

### Being part of an organization lets you:

- Show your university, lab or company work to the Hugging Face community
- Collaborate on private datasets, models and spaces
- Subscribe to a paid plan and use Hugging Face hosted training and inference services

Search for an organization

or [Create a new organization](#)

- Organization Creation (Optional): While you can upload datasets and fine-tune models directly on Hugging Face without creating an organization, you have the option to create an organization on Hugging Face. This can be particularly useful for team collaboration, as it allows you to upload all your datasets and models in one centralized location.

*Optional Step*

## New Organization

Complete your organization profile

Organization Username: WebexOne

Organization type: Classroom

Homepage (optional): Homepage

Logo (optional): Upload a file

Organization Full name: WebexOne

GitHub username (optional): GitHub alias

Twitter username (optional): Twitter account

AI & ML interests (optional): AI & ML interests

**Create Organization**

- At this stage, you will see no models or datasets created under your account.

The screenshot shows the Hugging Face Hub organization page for 'WebexOne'. At the top, there's a search bar and a navigation bar with links for Models, Datasets, Spaces, Posts, Docs, Solutions, Pricing, and a profile icon. Below the header, it says 'WebexOne Classroom' and 'Upgrade to Enterprise'. There are sections for 'AI & ML interests' (None defined yet), 'Team members' (1), and an 'Organization settings' card with a 'Create a Card' button. The main content area shows 'No organization card' with a message to customize the page. Below this, there are sections for 'Models' (None yet) and 'Datasets' (None yet), each with a blue arrow pointing to the text 'No models or Datasets created yet'. On the right side, a vertical sidebar is open, showing a profile section with 'Profile' and 'Webex', followed by 'Notifications' (Inbox (0)), and a list of actions: '+ New Model', '+ New Dataset', '+ New Space', '+ New Collection', 'Create organization', 'Settings', and 'Sign Out'. A blue arrow points from the top of the sidebar towards the profile picture.

- Access Your Models and Datasets: The same can be accessed by clicking your profile picture on the top right corner of the Hugging Face website. This will take you to your personal dashboard where you can view and manage your models and datasets.

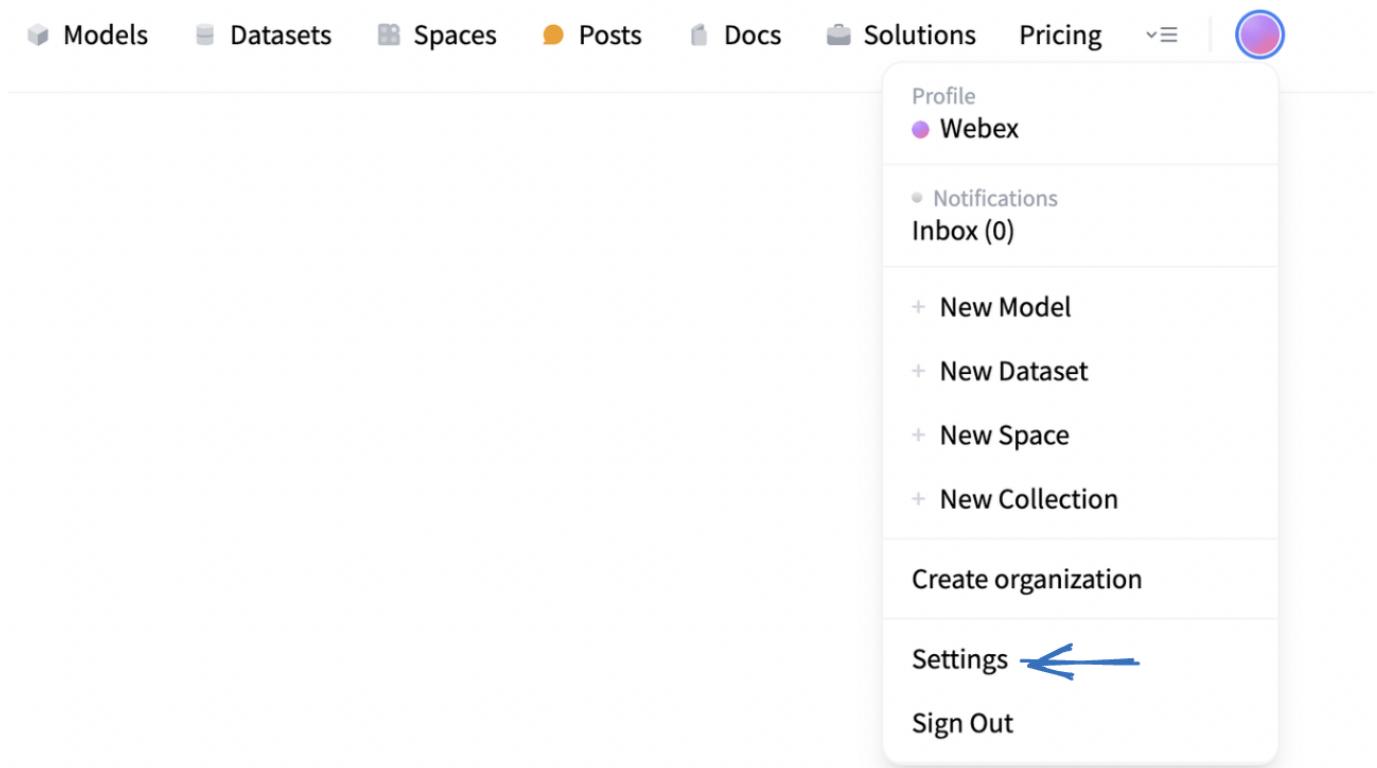
## Hugging Face API Keys

Create an API Key: As we will be uploading our datasets to the Hugging Face Hub, we need to create an API key for our account. This API key will be used to authenticate and interact with the Hugging Face services programmatically.

you can browse to [huggingface API Key](#)

or

Click on your profile picture > Settings > Access Tokens



The screenshot shows the 'Access Tokens' section within the Hugging Face Hub settings. On the left, there's a sidebar with links: Profile, Account, Authentication, Organizations, Billing, and 'Access Tokens' (which is highlighted with a blue arrow labeled '1'). The main area is titled 'User Access Tokens' and contains instructions about tokens. A button labeled '+ Create new token' is highlighted with a blue arrow labeled '2'.

Under the "Access Tokens" section, click on "Create new token." You will see options to select the token type and provide a token name. For example, you might name your token "Webexone" and select the appropriate permissions.

- fine-grained: tokens with this role can be used to provide fine-grained access to specific resources, such as a specific model or models in a specific organization. This type of token is useful in production environments, as you can use your own token without sharing access to all your resources.
- read: tokens with this role can only be used to provide read access to repositories you could read. That includes public and private repositories that you, or an organization you're a member of, own. Use this role if you only need to read content from the Hugging Face Hub (e.g. when downloading private models or doing inference).
- write: tokens with this role additionally grant write access to the repositories you have write access to. Use this token if you need to create or push content to a repository (e.g., when training a model or modifying a model card).

As we have a lab environment we will be using the "Write" permission. This token will have read and write access to all your resources and can make calls to inference API on your behalf, as shown in the image below.

## Create new Access Token

### Token type

Fine-grained   Read   Write ← 1

! This cannot be changed after token creation.

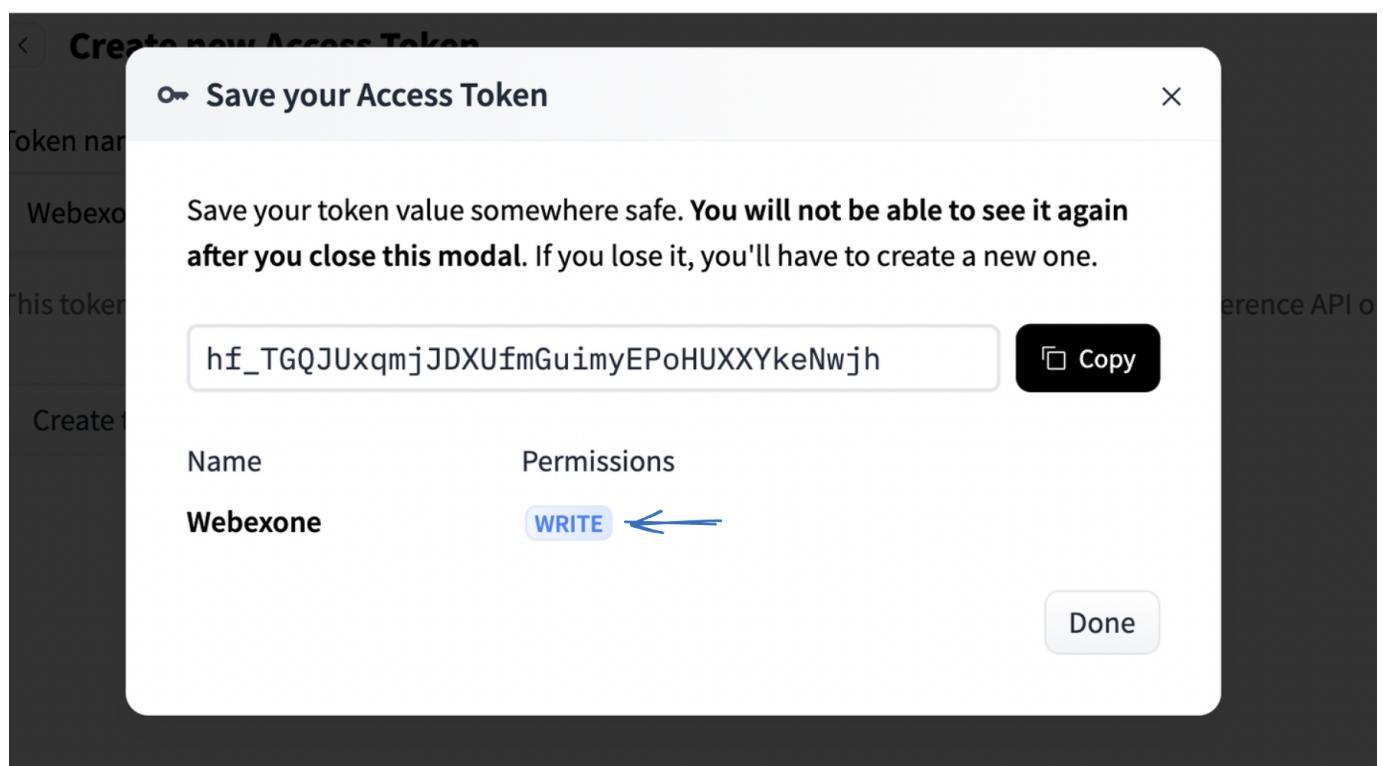
### Token name

Webexone

This token has read and write access to all your and your orgs resources and can make calls to inference API on your behalf.

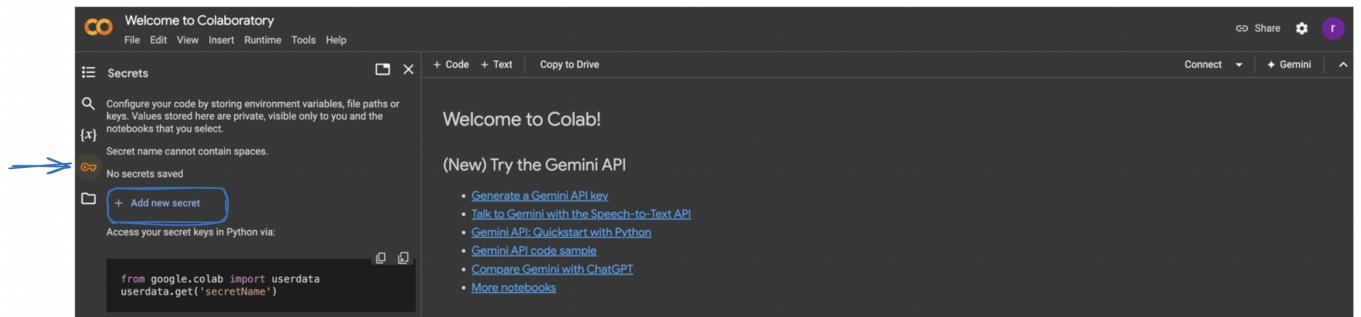
Create token ← 2

Save and Secure the Token: Once the token is generated, save it securely. This token will be required for accessing and managing your datasets via the API. hf\_TGQJUxqmjJDXUfmGuimyEPoHUXXYkeNwjh



## 1.2.6 Accessing Hugging Face API in Google Colab

- Open the Google Colab notebook and navigate to the new “Secrets” section in the sidebar.



- Click on “Add a new secret.” Enter the name example: HF\_TOKEN and value of the secret. Note: The name is permanent once set.
- The list of secrets is global across all your notebooks.
- Use the “Notebook access” toggle to grant or revoke access to a secret for each notebook.

The screenshot shows the 'Secrets' sidebar in Google Colab. It includes a search icon, a note about secret names, and a table for managing secrets. The table has columns for 'Notebook access', 'Name', 'Value', and 'Actions'. A specific row for 'HF\_TOKEN' is highlighted with a blue border. Handwritten annotations include arrows pointing to the 'Name' field (labeled 'Give it a meaningful name') and the 'Value' field (labeled 'The secret from Huggingface you created earlier').

Notebook access	Name	Value	Actions
<input checked="" type="checkbox"/>	HF_TOKEN	hf_TGQJUxq [REDACTED]	

Give it a  
meaningful  
name

The secret from Huggingface  
you created earlier

### Optional Steps below

Incorporating Secrets into Your Code - We will use it later in our lab

- To use a secret in your notebook, use the following code snippet

```
from google.colab import userdata  
my_secret_key = userdata.get('HF_TOKEN')
```

- Replace with your secret's name.

Using Secrets as Environment Variables - Optional Step , we will use it later in our lab

- For Python modules requiring API keys as environment variables, use the below snippet:

```
# Import Colab Secrets userdata module  
  
from google.colab import userdata  
import os  
  
# Set other API keys similarly  
os.environ["HF_TOKEN"] = userdata.get('HF_TOKEN')
```

## 1.3 Task 3: Log into the Lab Environment

Click My Demo Sessions select your ThousandEyes demo and click the green View button

The screenshot shows the 'Webex Demo Toolbox' interface. On the left, there's a sidebar with 'DASHBOARD' and 'DEMOS & TOOLS' sections. The 'DEMOS & TOOLS' section is expanded, showing categories like All, Calling, Devices, Feature VODs, Integrations, IT Admin, Meeting, Messaging, Security, and Tools. The 'My Demo Sessions' item in the main menu is highlighted with a red arrow pointing to it. The main content area is titled 'My Demo Sessions' and features a large orange eye icon. Below the icon, there's a title 'Troubleshooting Webex Meetings & Devices with ThousandEyes'. A detailed description follows: 'Demo the power of Webex and ThousandEyes working together to troubleshoot Webex meetings and devices using the Control Hub/ThousandEyes integration.' Below this, it shows 'Status: Active' and 'Ends: in 2 days'. Under 'Using:', there's a profile picture of a user named 'Omer Ilyas' with a red arrow pointing to it. At the bottom, there are three buttons: a green 'View' button, a blue 'Refresh' button, and a blue 'Actions' button with a dropdown arrow.

Lab Administration Information (New Read Only Admin User)

- Webex Control Hub/ThousandEyes Admin Access - A new read only admin user will be created. This is the new user you'll use to access the Webex ControlHub and ThousandEyes Platform (Note: Your user will most likely be different than this one as they are randomly assigned.)
- Email Address will be used for the SSO login access as well as the Password

## Control Hub/ThousandEyes Read Only Admin

Character for logging into Control Hub/ThousandEyes with to view troubleshooting.

The screenshot shows a user profile for Abbigail Macejkovic. It includes a circular profile picture, the name 'Abbigail Macejkovic', an email address 'Email: abbigail.macejkovic@cumulusorg.com' with a copy icon, a password field 'Password: [.....]' with a copy icon, and a 'PMR' status with a copy icon. Below these are the extension 'Extension: 700555' and a blue 'Refresh' button.

- Access information to the Webex ControlHub Portal - right click and use an Incognito Browser

## Control Hub URL

**TIP:** Right click link and choose open in incognito/private browser.

[Control Hub Administration](#)

- Access information to the ThousandEyes Portal be sure to right click and use an Incognito Browser

## ThousandEyes Portal

TIP: Right click link and choose open in incognito/private browser.

ThousandEyes Administration  

First thing will be to get your VM set up

- Click the Workstation 1 link. You should see a Windows Desktop show up in new browser tab.

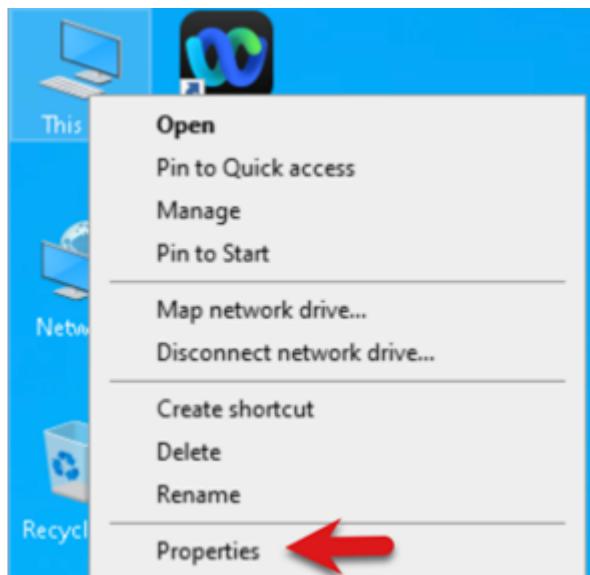
## Workstation

Your virtual workstation. Workstation should already be powered on and accessible by clicking the Workstation 1 link. Use the green and yellow buttons for troubleshooting purposes only. If you don't see a link, then you didn't schedule the demo with one.

Workstation 1



Right Click on the This PC desktop icon and click Properties so you can rename the VM.



Click Rename this PC

**About**

Your PC is monitored and protected.

See details in Windows Security

**Device specifications**

Device name	Workstation1
Processor	Intel(R) Xeon(R) CPU E7- 2830 @ 2.13GHz 2.70 GHz (2 processors)
Installed RAM	8.00 GB
Device ID	33B895B8-DA7E-4199-9608-A7835E1687C4
Product ID	00331-20300-00000-AA122
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

**Rename this PC**

Name the PC your-name-Roadshow (no spaces), click Next and click Restart Now

**Rename your PC**

You can use a combination of letters, hyphens, and numbers.

Current PC name: Workstation1

Omer-Roadshow

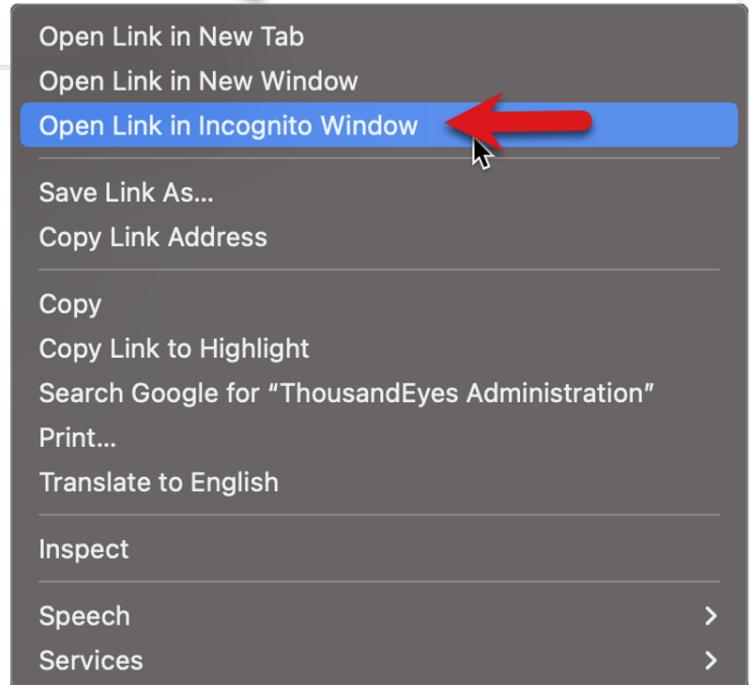
Next Cancel

While the VM reboots go back to your previous tab for demo . Right click the Control Hub URL and ThousandEyes Administration Portal link and open both in incognito browsers as we will be using them in the coming steps. The same creds can be used to login into Webex Control Hub and ThousandEyes

# ThousandEyes Portal

TIP: Right click link and choose open in incognito/private browser.

ThousandEyes Administration 



Use the credentials from the ControlHub/ThousandEyes Read Only Admin in the next steps.

## Control Hub/ThousandEyes Read Only Admin

Character for logging into Control Hub/ThousandEyes with to view troubleshooting.

The screenshot shows a user profile for 'Abbigail Macejkovic'. It includes a circular profile picture, the name 'Abbigail Macejkovic', an email address 'Email: abbigail.macejkovic@cumulusorg.com' with a copy icon, a password field 'Password: [.....]' with a copy icon, and a 'PMR' status with a copy icon. Below these are 'Extension: 700555' and a blue 'Refresh' button.

Your incognito browser will open to the ThousandEyes SSO login page as shown in the below image. Use the Username and Password from the ThousandEyes Read Only Admin to complete the login process. Click Don't show this again and Yes in the pop up window after you have completed the SSO login process.

The screenshot shows the ThousandEyes Single sign-on login page. It features the ThousandEyes logo, a 'Single sign-on' section with an input field containing the email 'abbigail.macejkovic@cumulusorg.com', and a large orange 'Log In' button. A red arrow points to the 'Log In' button. Below it is a link 'Log in with a password'.



## Sign in

abigail.macejkovic@cumulusorg.com

[Can't access your account?](#)



Back

Next



← abigail.macejkovic@cumulusorg.com

## Enter password

.....|

[Forgot my password](#)



Sign in