

**MATERIAL DE APOIO**

## **Google Cloud Skill Boost**

### **Laboratório 12 do Intermediate: Implantar um aplicativo Streamlit integrado ao Gemini Pro no Cloud Run**

## Sumário

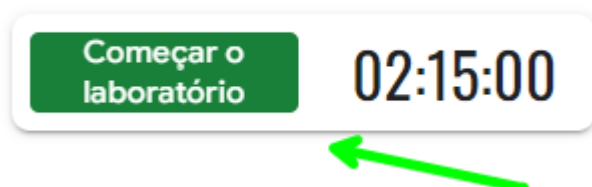
1.	Login no Console do Google Cloud .....	3
2.	Tarefa 1 .....	5

## 1. Login no Console do Google Cloud

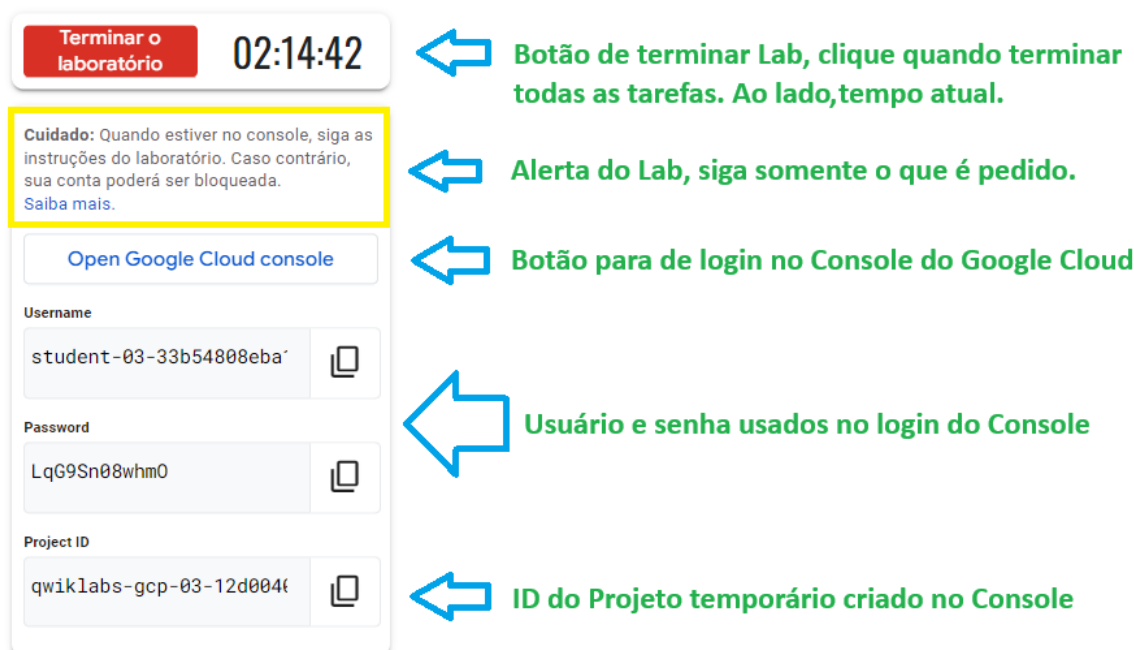
Link para o laboratório:

[https://www.cloudskillsboost.google/paths/236/course\\_templates/978/labs/488168?locale=pt\\_B  
R](https://www.cloudskillsboost.google/paths/236/course_templates/978/labs/488168?locale=pt_BR)

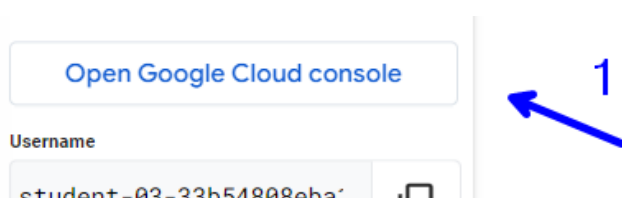
- 1) Primeiro passo é fazer login no Console do Google Cloud, clique no botão verde “Começar o laboratório”:

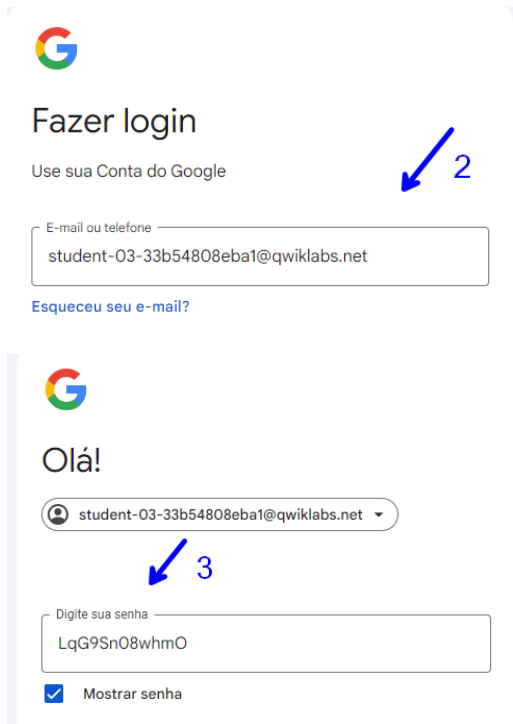


- 2) Após, irá abrir o Painel de login no Console:



- 3) Clique no botão “Open Google Cloud Console” (1) com o botão direito do mouse e escolha para abrir em uma aba Anônima/InPrivate, depois preencha o usuário (2) e senha (3) na página de login:





**Fazer login**

Use sua Conta do Google

E-mail ou telefone

student-03-33b54808eba1@qwiklabs.net

[Esqueceu seu e-mail?](#)

**Olá!**

student-03-33b54808eba1@qwiklabs.net

Digite sua senha

LqG9Sn08whmO

☒ Mostrar senha

4) Aceite todos os termos e condições do Google Cloud (1-2):

em myaccount.google.com.

Seu uso dos Serviços do Google com esta conta também é regido por políticas internas da sua organização.

**Entendi**

**Google Cloud**

**Welcome student fe2a879d!**

Create and manage your Google Cloud instances, disks, networks, and other resources in one place.



student fe2a879d

student-03-33b54808eba1@qwiklabs.net

[SWITCH ACCOUNT](#)

**Country**

Brazil

**Terms of Service**

☒ I agree to the [Google Cloud Platform Terms of Service](#), and the terms of service of [any applicable services and APIs](#).

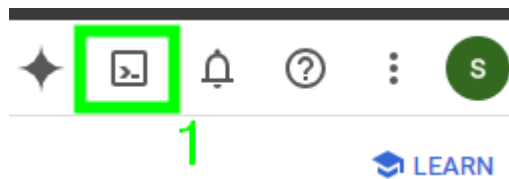
**Email updates**

☐ I would like to receive periodic emails on news, product updates and special offers from Google Cloud and Google Cloud Partners.

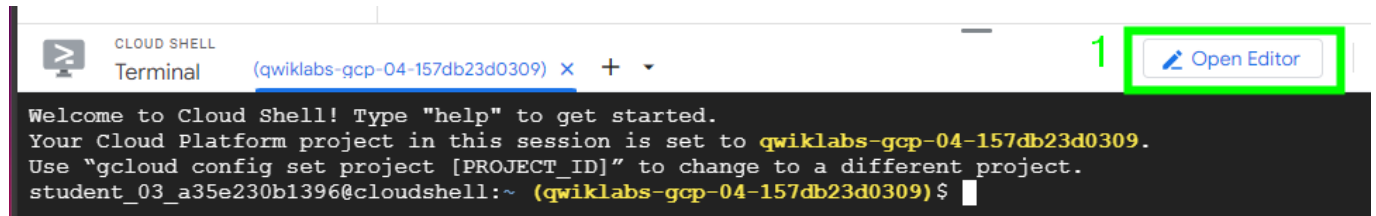
**AGREE AND CONTINUE**

## 2. Tarefa 1

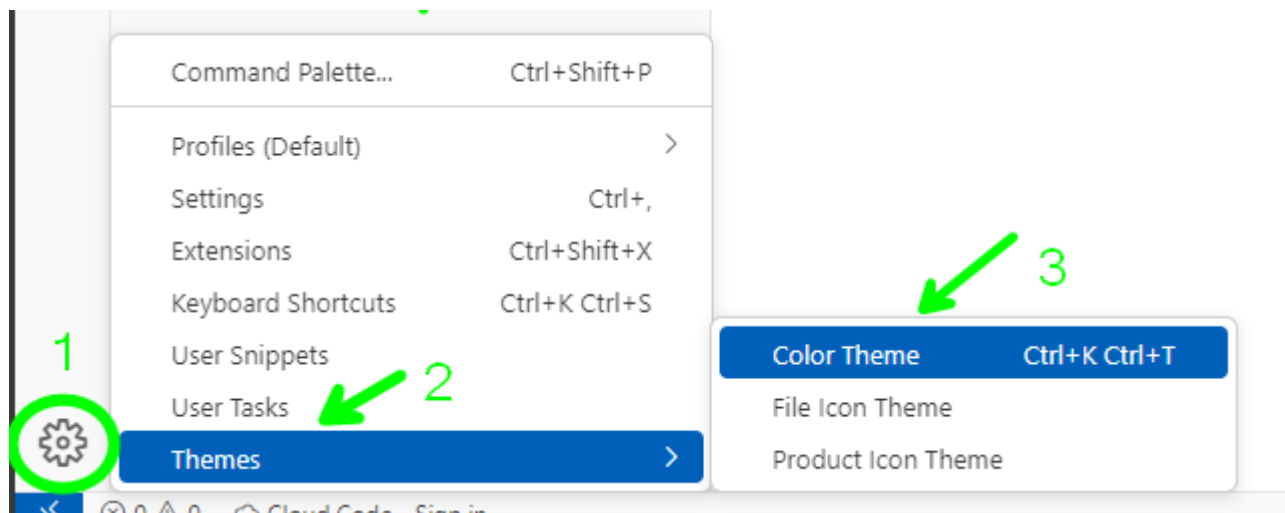
- 1) No canto superior direito clique em **Cloud Shell** (Ao lado do botão de perfil do estudante) **(1)**, e aperte Continue depois que abrir o terminal:



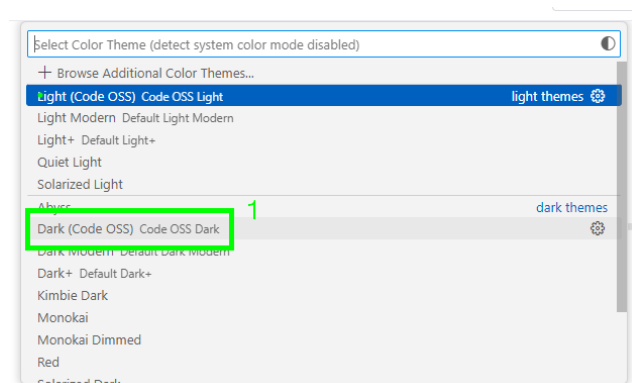
- 2) Depois que o **Terminal** carregar, clique em **Open Editor** **(1)**:



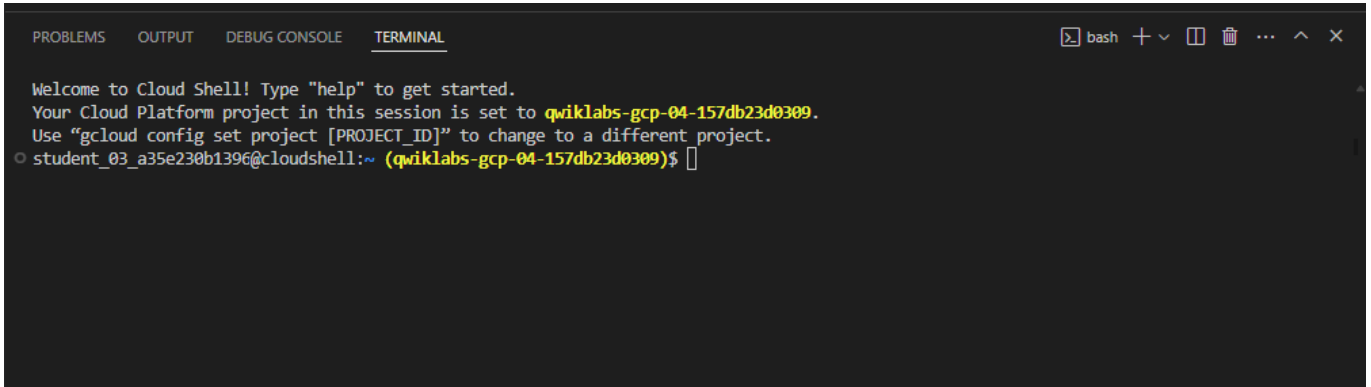
- 3) Assim que o Editor carregar, clique na engrenagem no canto inferior esquerdo **(1)**, depois em **Themes** **(2)**, e em seguida em **Color Themes** **(3)**:



- 4) Clique em **Dark (Code OSS)** **(1)**:



5) Aperte CTRL+J para abrir o Terminal (Confirme que está no projeto correto):



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-04-157db23d0309.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
○ student_03_a35e230b1396@cloudshell:~ (quiklabs-gcp-04-157db23d0309)$
  
```

6) Execute os seguintes comandos **em ordem** (mantenha-se nesse diretório até o final), clique em **Authorize** se precisar:

```
git clone https://github.com/GoogleCloudPlatform/generative-ai.git
```

```
cd generative-ai/gemini/sample-apps/gemini-streamlit-cloudrun
```

```
python3 -m venv gemini-streamlit
```

```
source gemini-streamlit/bin/activate
```

```
pip install -r requirements.txt
```

7) A página do laboratório irá passar os dados referente ao **PROJECT-ID e REGION**, passe ela como variáveis do ambiente (Se trocar de sessão execute novamente esses comandos). Troque o negrito pelas informações de seu laboratório:

```
GCP_PROJECT='PROJECT-ID'
```

```
GCP_REGION='REGION'
```

Ficando como o exemplo a seguir:



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
bash - gemini-streamlit-cloudrun
(gemini-streamlit) student_01_e7d42843bf3e@cloudshell:~/generative-ai/gemini/sample-apps/gemini-streamlit-cloudrun (quiklabs-gcp-03-33c07cd0fef1)$ GCP_PROJECT='quiklabs-gcp-03-33c07cd0fef1'
GCP_REGION='us-west1'
  
```

No Cloud Shell, execute os comandos a seguir:

```

GCP_PROJECT='quiklabs-gcp-03-33c07cd0fef1'
GCP_REGION='us-west1'
  
```

- 8) Execute o seguinte comando (Garanta que esteja no mesmo diretório anterior), volte ao **Cloud Shell** e clique em **Authorize** se necessário:

```
streamlit run app.py \

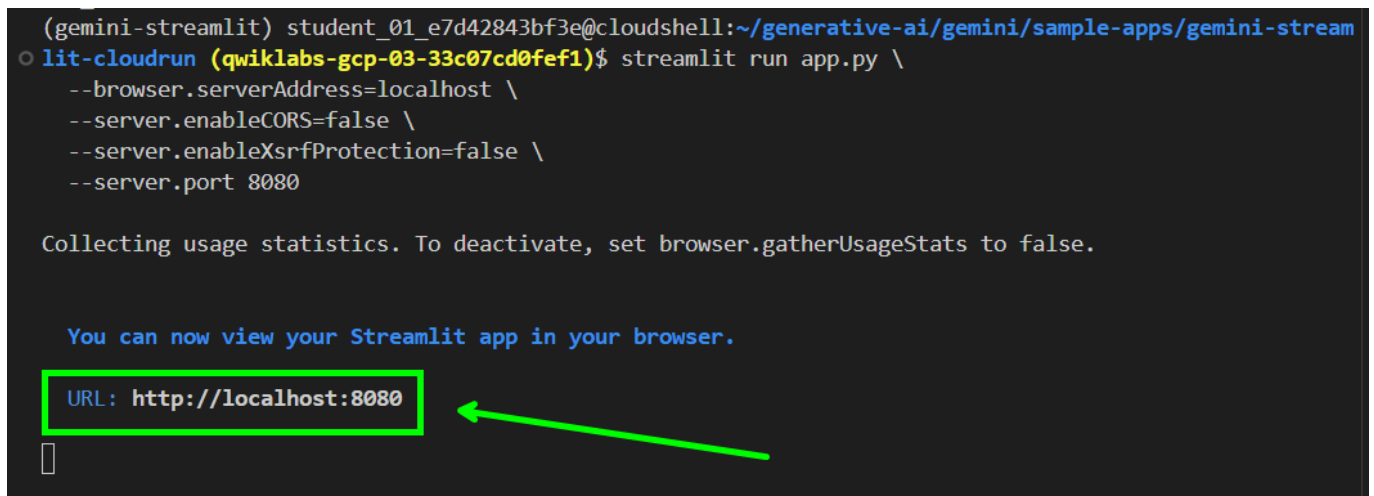
--browser.serverAddress=localhost \

--server.enableCORS=false \

--server.enableXsrfProtection=false \

--server.port 8080
```

- 9) Após implantado, clique no link da URL com **CTRL+Botão esquerdo do mouse**:



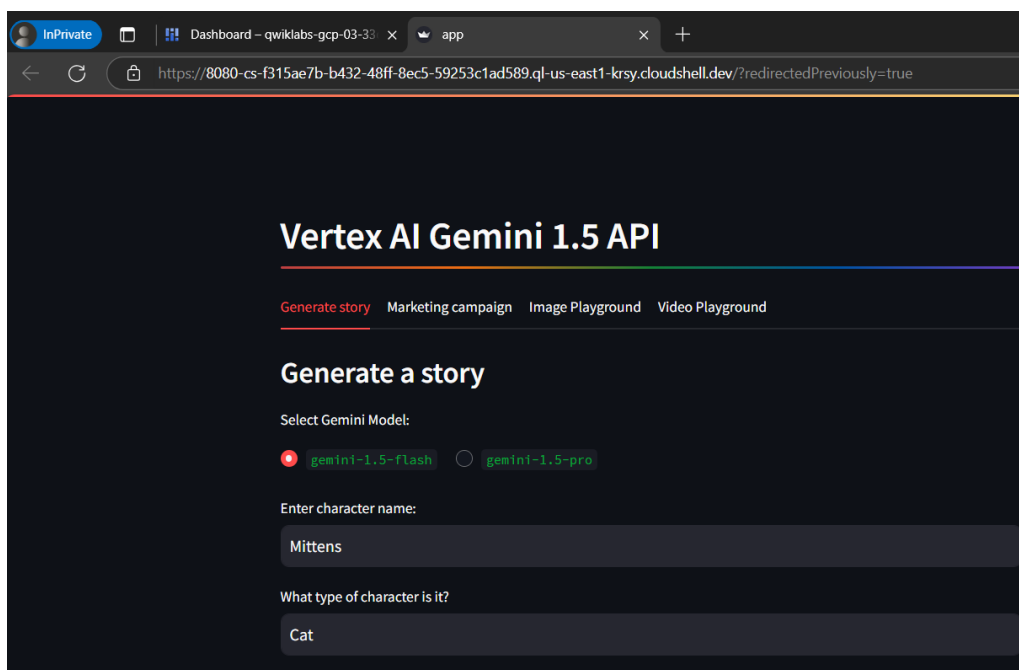
```
(gemini-streamlit) student_01_e7d42843bf3e@cloudshell:~/generative-ai/gemini/sample-apps/gemini-streamlit-cloudrun (qwiklabs-gcp-03-33c07cd0fef1)$ streamlit run app.py \
--browser.serverAddress=localhost \
--server.enableCORS=false \
--server.enableXsrfProtection=false \
--server.port 8080

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.

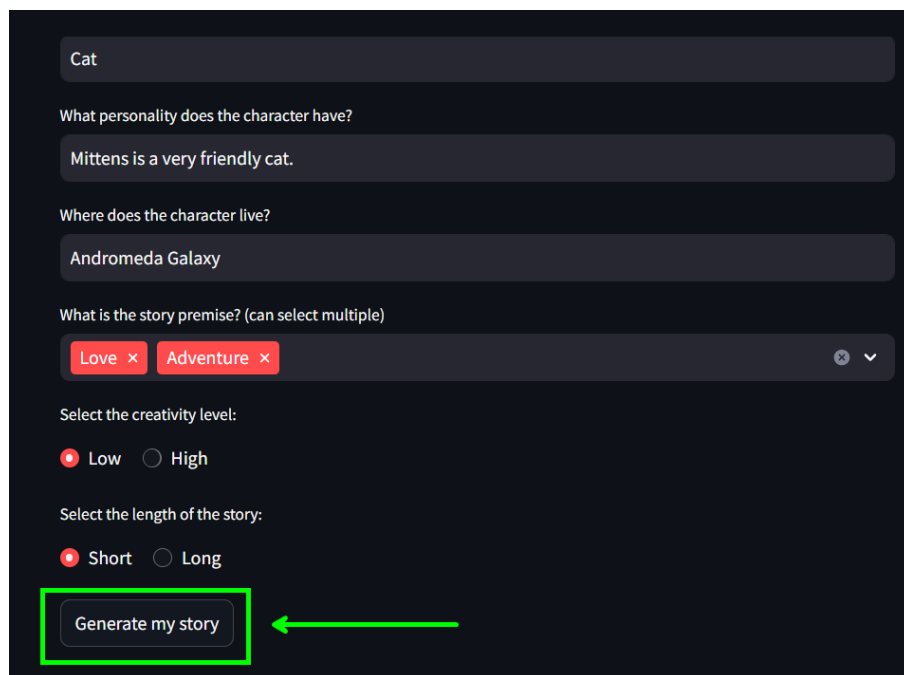
You can now view your Streamlit app in your browser.

URL: http://localhost:8080
```

- 10) A aplicação irá abrir em uma nova aba do navegador, verifique que está tudo ok:



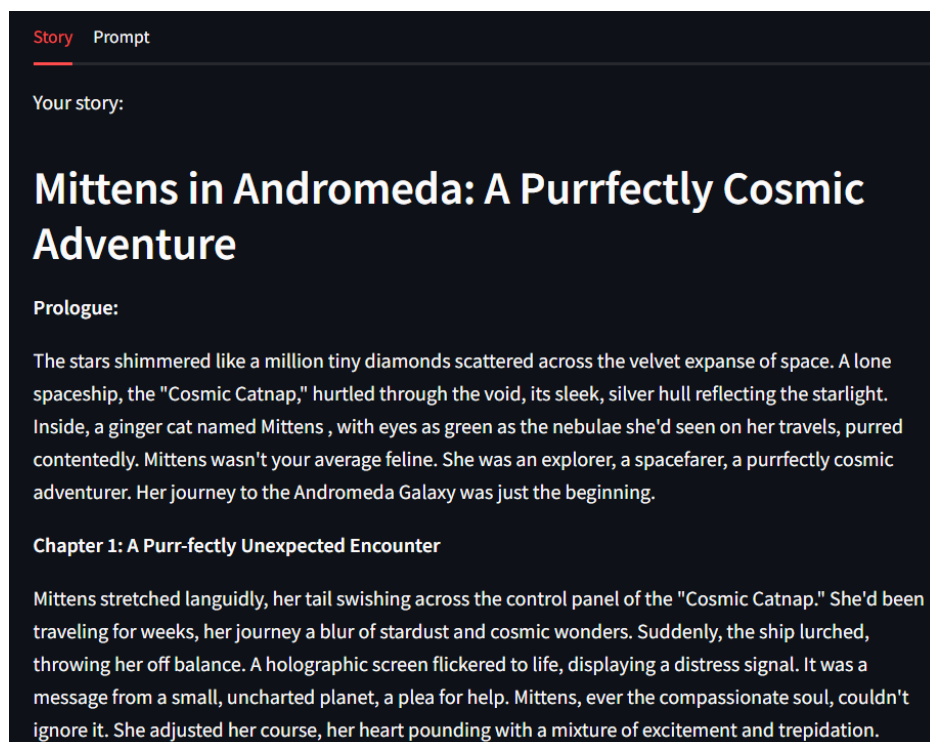
11) Teste a aplicação clicando no botão **Generate my story** abaixo, mude os parâmetros se desejar:



The screenshot shows a dark-themed form for generating a story. It includes several input fields and selection options:

- Character Name:** A text box containing "Cat".
- Personality:** A text box with the prompt "What personality does the character have?" and the input "Mittens is a very friendly cat."
- Location:** A text box with the prompt "Where does the character live?" and the input "Andromeda Galaxy".
- Premise:** A section titled "What is the story premise? (can select multiple)" with two red tags labeled "Love" and "Adventure", each with a close button (x). A dropdown arrow is visible on the right.
- Creativity Level:** Radio buttons for "Low" (selected) and "High".
- Story Length:** Radio buttons for "Short" (selected) and "Long".
- Action:** A button labeled "Generate my story" is highlighted with a green rectangular box. A green arrow points from the right towards this button.

12) Deve aparecer a resposta em formato de texto logo abaixo:



The screenshot displays the generated story in a dark-themed interface. At the top, there are two tabs: "Story" (active) and "Prompt". Below the tabs, the text "Your story:" is followed by the title "Mittens in Andromeda: A Purrfectly Cosmic Adventure".

**Prologue:**

The stars shimmered like a million tiny diamonds scattered across the velvet expanse of space. A lone spaceship, the "Cosmic Catnap," hurtled through the void, its sleek, silver hull reflecting the starlight. Inside, a ginger cat named Mittens, with eyes as green as the nebulae she'd seen on her travels, purred contentedly. Mittens wasn't your average feline. She was an explorer, a spacefarer, a purrfectly cosmic adventurer. Her journey to the Andromeda Galaxy was just the beginning.

**Chapter 1: A Purr-fectly Unexpected Encounter**

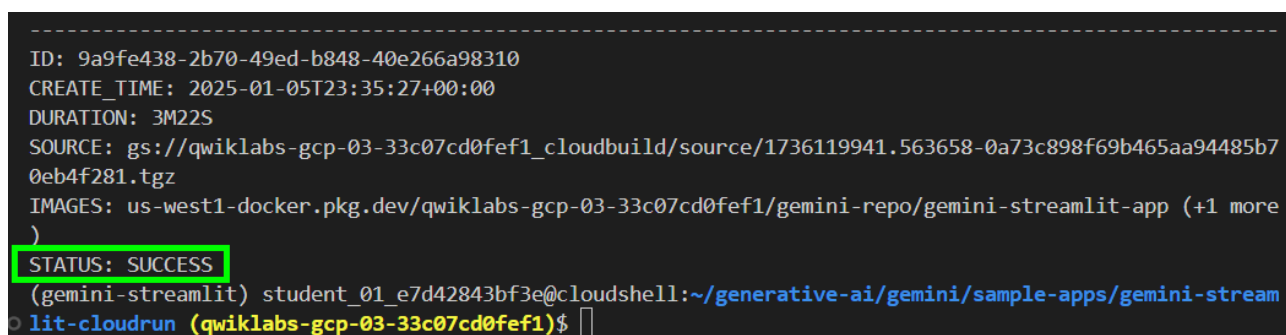
Mittens stretched languidly, her tail swishing across the control panel of the "Cosmic Catnap." She'd been traveling for weeks, her journey a blur of stardust and cosmic wonders. Suddenly, the ship lurched, throwing her off balance. A holographic screen flickered to life, displaying a distress signal. It was a message from a small, uncharted planet, a plea for help. Mittens, ever the compassionate soul, couldn't ignore it. She adjusted her course, her heart pounding with a mixture of excitement and trepidation.

13) Depois de concluir o teste do aplicativo, você pode interrompê-lo digitando Ctrl + C no **Cloud Shell**.



- 14) Execute os seguintes comandos no **Cloud Shell**, garanta que as variáveis de ambiente estejam setadas corretamente, realizadas anteriormente (Se tiver saído daquela sessão, execute os comandos de setar as variáveis de ambiente novamente, GCP\_PROJECT e GCP\_REGION):

```
AR_REPO='gemini-repo'
SERVICE_NAME='gemini-streamlit-app'
gcloud artifacts repositories create "$AR_REPO" --location="$GCP_REGION" --repository-
format=Docker
gcloud builds submit --tag "$GCP_REGION-
docker.pkg.dev/$GCP_PROJECT/$AR_REPO/$SERVICE_NAME"
```



```
-----
ID: 9a9fe438-2b70-49ed-b848-40e266a98310
CREATE_TIME: 2025-01-05T23:35:27+00:00
DURATION: 3M22S
SOURCE: gs://qwiklabs-gcp-03-33c07cd0fef1_cloudbuild/source/1736119941.563658-0a73c898f69b465aa94485b7
0eb4f281.tgz
IMAGES: us-west1-docker.pkg.dev/qwiklabs-gcp-03-33c07cd0fef1/gemini-repo/gemini-streamlit-app (+1 more
)
STATUS: SUCCESS
(gemini-streamlit) student_01_e7d42843bf3e@cloudshell:~/generative-ai/gemini/sample-apps/gemini-stream
lit-cloudrun (qwiklabs-gcp-03-33c07cd0fef1)$
```

- 15) Depois de implantar o Artifact Registry, execute os seguintes comandos:

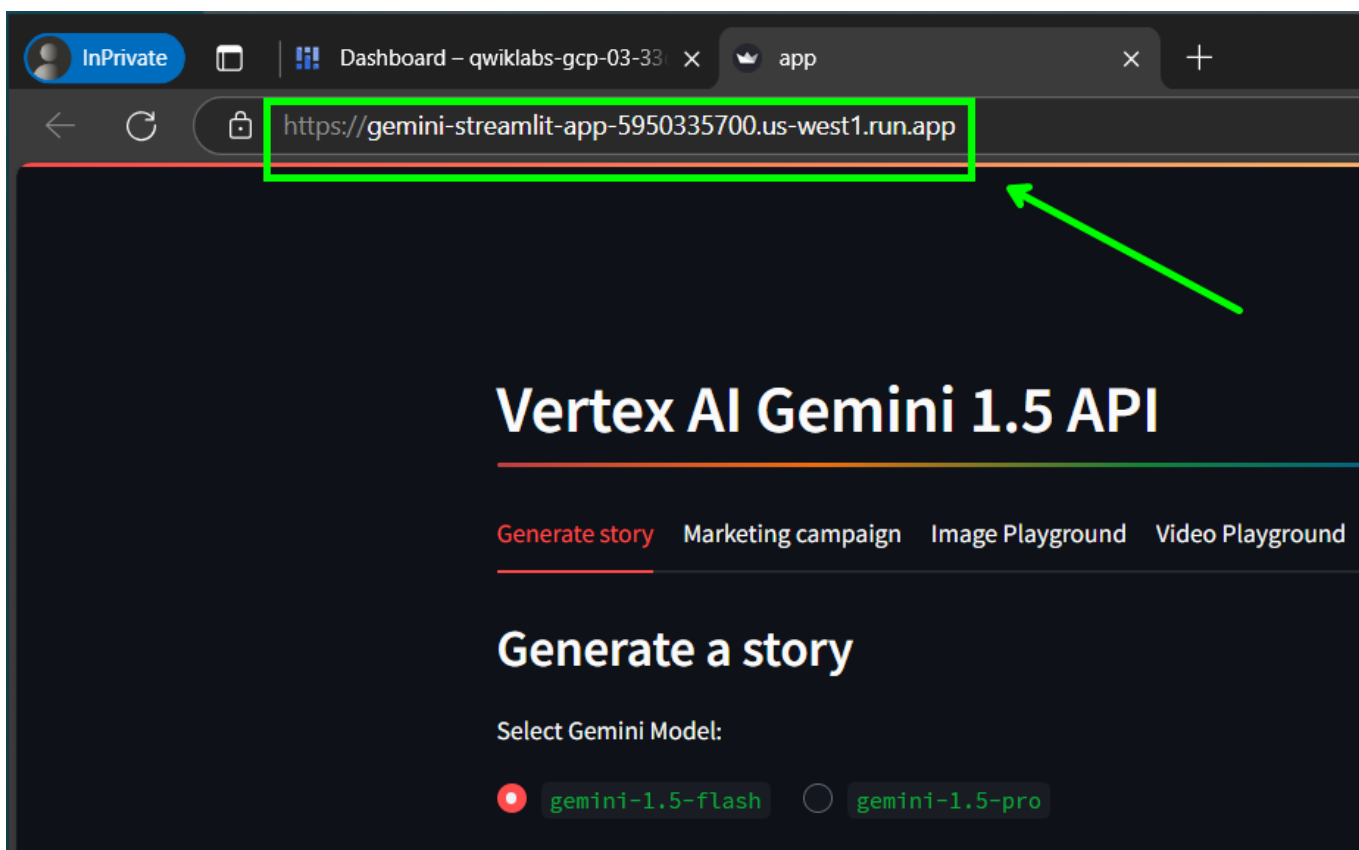
```
gcloud run deploy "$SERVICE_NAME" \
  --port=8080 \
  --image="$GCP_REGION-
docker.pkg.dev/$GCP_PROJECT/$AR_REPO/$SERVICE_NAME" \
  --allow-unauthenticated \
  --region=$GCP_REGION \
  --platform=managed \
  --project=$GCP_PROJECT \
  --set-env-vars=GCP_PROJECT=$GCP_PROJECT,GCP_REGION=$GCP_REGION
```

- 16) Quando a implantação for bem-sucedida, você vai receber um URL para o serviço Cloud Run. Você pode acessá-lo no navegador para visualizar o aplicativo Cloud Run que acabou de implantar.

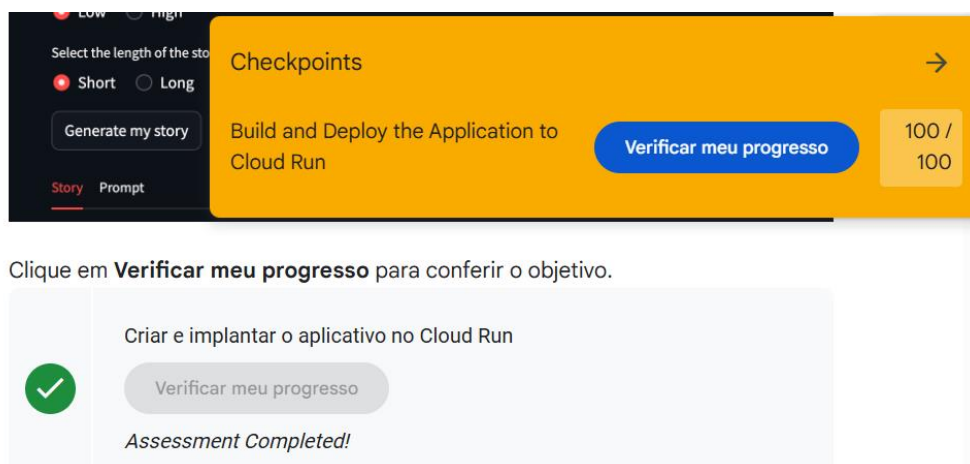
```

[us-west1]
OK Deploying new service... Done.
OK Creating Revision...
OK Routing traffic...
OK Setting IAM Policy...
Done.
Service [gemini-streamlit-app] revision [gemini-streamlit-app-00001-5qp] has been deployed and is serving 100 percent of traffic.
Service URL: https://gemini-streamlit-app-5950335700.us-west1.run.app
(gemini-streamlit) student_01_e/d42843b73e@cloudshell:~/generative-ai/gemini/sample-apps/gemini-streamlit-cloudrun (qwiklabs-gcp-03-33c07cd0fef1)$

```



17) Tique o progresso do laboratório e em seguida na aba amarela para a tarefa:



18) Clique no botão vermelho do painel para finalizar o laboratório:

