

Streamlit Apps Dash: SnowflakeCortex
Share [Run](#)

```

10 def translate():
11     with st.container():
12         st.header("Translate With Snowflake Cortex")
13         col1, col2 = st.columns(2)
14         with col1:
15             from_language = st.selectbox("From", dict(sorted(supported_languages.items()))
16             with col2:
17                 to_language = st.selectbox("To", dict(sorted(supported_languages.items()))
18                 entered_text = st.text_area("Enter text", label_visibility="hidden", height=300, p
19                 if entered_text:
20                     entered_text = entered_text.replace("'", "\\'")
21                     cortex_response = session.sql("""select snowflake.cortex.translate(entered_t
22                     st.write(cortex_response)
23
24 def sentiment_analysis():
25     with st.container():
26         st.header("Sentiment Analysis With Snowflake Cortex")
27         entered_text = st.text_area("Enter text", label_visibility="hidden", height=400, p
28         if entered_text:
29             entered_text = entered_text.replace("'", "\\'")
30             cortex_response = session.sql("""select snowflake.cortex.sentiment(entered_t
31             st.caption("Score is between -1 and 1: -1 = Most negative, 1 = Positive, 0 =
32             st.write(cortex_response)
33
34 def summarize():
35     with st.container():
36         st.header("JSON Summary With Snowflake Cortex")
37         selected_llm = st.selectbox("Select Model", supported_llms)
38         entered_text = st.text_area("Enter text", label_visibility="hidden", height=400, p
39         if entered_text:
40             entered_text = entered_text.replace("'", "\\'")
41             prompt = f"Summarize this transcript in less than 200 words. Put the produc
42             cortex_prompt = "[INST] " + prompt + " [/INST]"
43             cortex_response = session.sql("""select snowflake.cortex.complete(
44             if selected_llm != 'gemma-7b':
45                 st.json(cortex_response)
46             else:
47                 st.write(cortex_response)
48
49 page_names_to_funcs = {
50     "Translate": translate,
51     "Sentiment Analysis": sentiment_analysis,
52     "JSON Summary": summarize
53 }
54
55 selected_page = st.sidebar.selectbox("Select", page_names_to_funcs.keys())
56 page_names_to_funcs[selected_page]()

```

Select

Translate

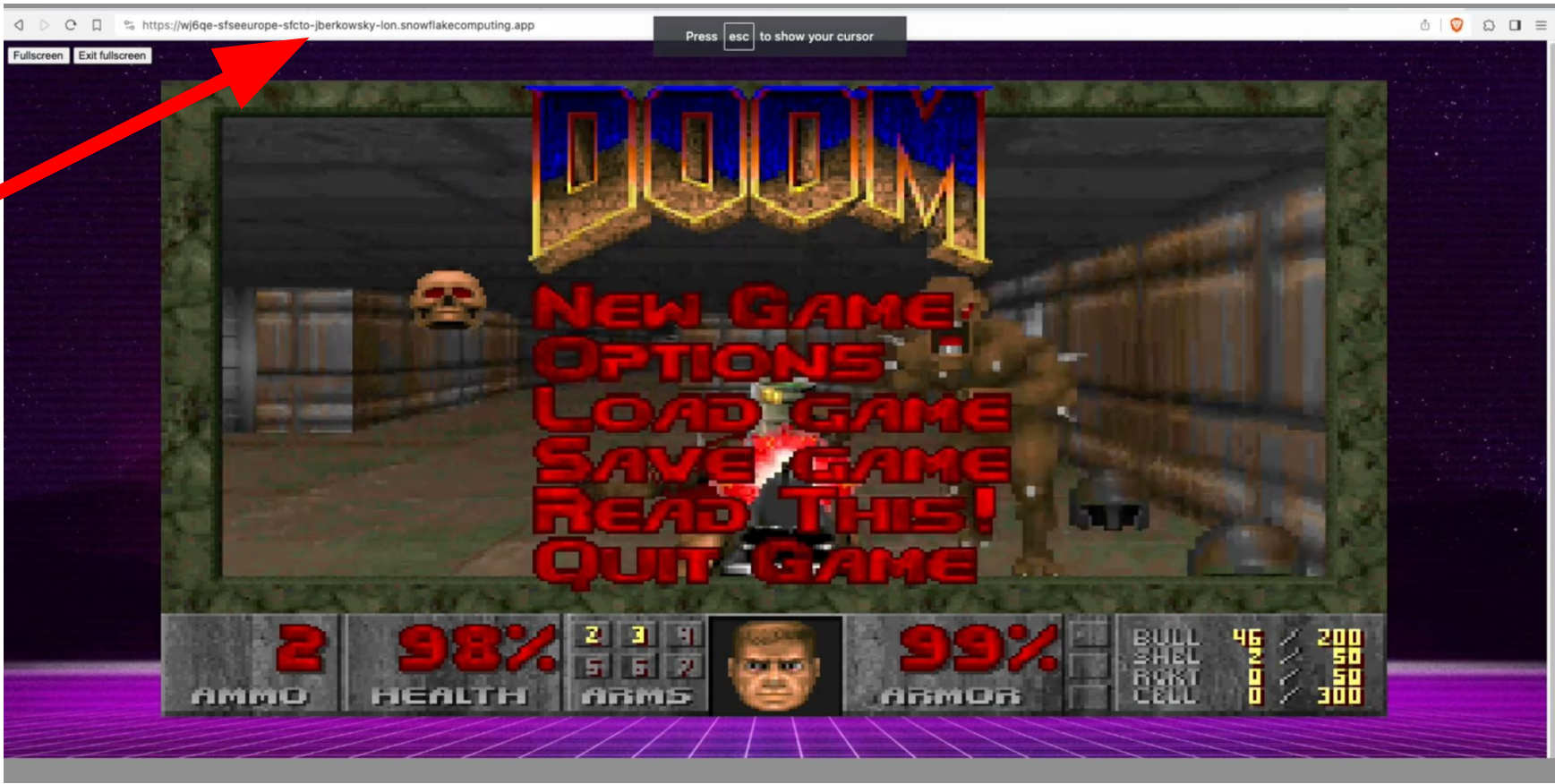
Translate With Snowflake Cortex

From to

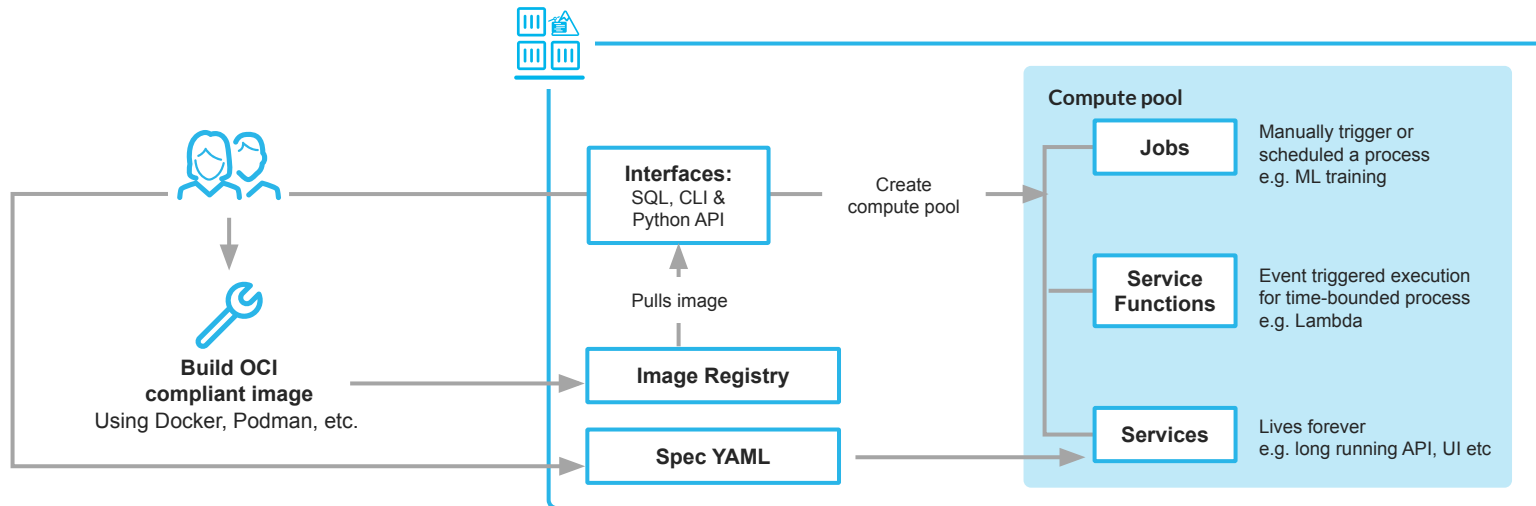
English German

For example: call transcript





Snowpark Container Services



Snowpark Container Services

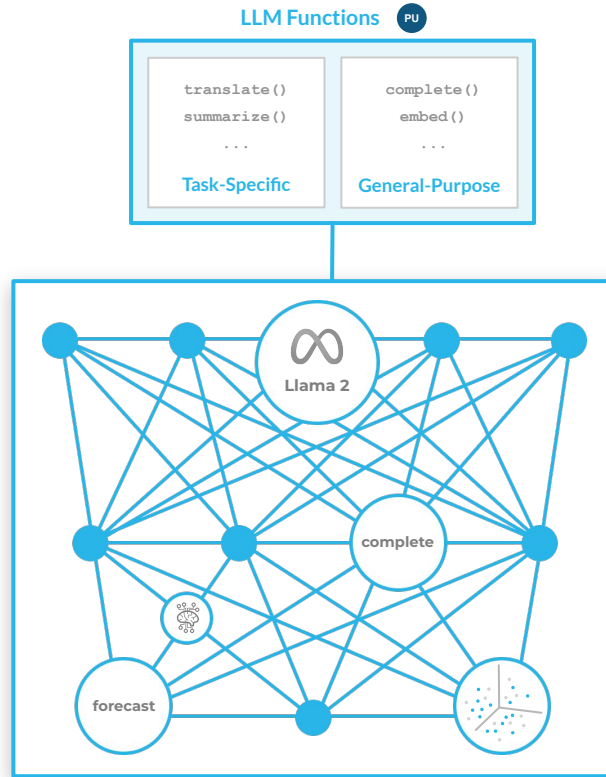
```
CREATE COMPUTE POOL my_cp1  
  MIN_NODES = 1  
  MAX_NODES = 3  
  INSTANCE_FAMILY = GPU  
  
CREATE SERVICE [ IF NOT EXISTS ] inference point  
  MIN_INSTANCES = 1  
  MAX_INSTANCES = 3  
  COMPUTE_POOL = my_cp1  
  SPEC = @my_service_stage/path/to/inference_point.yaml
```

Define compute pool
size and instance type

Choose and define
execution as service,
function or job

Point to spec file with
service definition

Snowflake Cortex



PU Public Preview

Cortex Functions Load call transcript data +

ACCOUNTADMIN GHDZ_S Share

Databases Worksheets

Pinned (0)

No pinned objects

Search objects ...

- > "DASH_DB"
- > ACCUTREND_UCC_SAMPLE
- > AGRICULTURE_DATA_ATLAS
- > AMANDA_TEST
- > AMANDA_TEST2
- > AMERICAN_COMMUNITY_S...
- > BRAZE
- > BUILDLOCAL
- > BUILDLOCAL_DAN
- > BUILD_ICEBERG_DB
- > CARTO
- > CARTO_AMERICAN_ATLAS
- > CARTO_FEATURES
- > CARTO_FEATURES_US
- > CC_QUICKSTART_CORTEX_...
- > CHAIRLIFT_CONSUMER_DATA
- > CHAIRLIFT_PKG
- > CHAIRLIFT_PROVIDER DATA

No Database selected Settings

Code Versions

```

1 select snowflake.cortex.translate('wie geht es dir heute?', 'de_DE', 'en_XX');
2

```

Ask Copilot

Run 3 useful LLM inference jobs in minutes with Snowflake Cortex



Dash Desai · Follow

Published in Snowflake · 5 min read · Feb 28, 2024



127



3



Overview

Getting started with AI on enterprise data can seem overwhelming, between getting familiar with LLMs, how to perform custom prompt engineering, and how to get a wide range of LLMs deployed/integrated to run multiple tests all while keeping that valuable enterprise data secure. Well, a lot of these complexities are being abstracted away for you in Snowflake Cortex — currently in Public Preview.



[DEV DAY]

Upskill in data & AI.
Meet fellow builders.
Have fun.

MOSCONE CENTER | SAN FRANCISCO | JUNE 6

FREE REGISTRATION

WHY ATTEND

SPEAKERS

AGENDA

REGISTER NOW

```
import streamlit as st
import altair as alt
import pandas as pd
from snowflake.snowpark.context import get_active_session
from snowflake.snowpark.functions import col

st.set_page_config(layout='wide')
session = get_active_session()

# Streamlit in Snowflake: Snowflake Cortex Demo by Dash

def sentiment_analysis():
    with st.container():
        st.header("Sentiment Analysis With Snowflake Cortex")
        entered_transcript = st.text_area("Enter call transcript", label_visibility="collapsed")
        entered_transcript = entered_transcript.replace("'", "\\'")
        if entered_transcript:
            cortex_response = session.sql(f"select snowflake.cortex.sentiment('{entered_transcript}')").collect()
            st.caption("Score between -1 and 1; -1 = Most negative, 1 = Most positive.")
            st.write(cortex_response)

def translate():
    with st.container():
        supported_languages = {
            "de": "German",
            "en": "English",
            "es": "Spanish",
            "fr": "French",
            "it": "Italian",
            "ja": "Japanese",
            "ko": "Korean",
            "pt": "Portuguese",
            "ru": "Russian",
            "zh": "Chinese"
        }
        st.header("Translate With Snowflake Cortex")
        col1, col2 = st.columns(2)
        with col1:
            from_language = st.selectbox("From Language", list(supported_languages.keys()))
        with col2:
            to_language = st.selectbox("To Language", list(supported_languages.keys()))
        entered_text = st.text_area("Enter text to translate")
        if entered_text:
            cortex_response = session.sql(f"select snowflake.cortex.translate('{entered_text}', '{from_language}', '{to_language}')").collect()
            st.write(cortex_response)

@st.cache_data()
def load_history():
    df_history = pd.DataFrame()
    return df_history

def main():
    st.sidebar.button("Home")
    st.sidebar.button("Sentiment Analysis")
    st.sidebar.button("Translate")
    st.sidebar.button("History")
    st.sidebar.button("Logout")
    sentiment_analysis()
    translate()
    load_history()
    df_history = load_history()
    st.write(df_history)
    st.write(cortex_response)
    st.write(cortex_response)

if __name__ == "__main__":
    main()
```

