



CLOUD BACKUP SYSTEM

Students: Krystal Lewin, Angel Jimenez, Nickelli Pierre, Buriel Noel

Instructor: Sadjadi S. Masoud

Faculty: Florida International University - College of Engineering & Computing



INTRODUCTION

System is a cloud-based backup and disaster recovery platform designed to maintain business continuity and quickly recover from disruptions. It integrates automated backups, data replication and failover strategies to minimize downtime and reduce data loss. Overall it provides a secure, reliable and cost effective approach to disaster recovery ensuring minimal disruption and long-term operational resilience.



PROBLEM

Small Businesses are at high risk of losing important data:

- They don't have reliable or automated backup systems
- When accidents occur they have no easy way of recovering lost data
- Not having a reliable backup system can cause many other issues.



SOLUTION

- A system where users can backup files, restore data, check connections
- Supports multiple cloud providers and selects the best & cost efficient one
- Built using Streamlit(UI) with python functions
- Backup files are saved and read using AWS 3 API
- Uses AWS 3

IMPLEMENTATION

- Implementation works by first setting up a secure cloud storage space, that includes risk analysis, recovery goals and design. The system is designed to replicate data, automate backups and store information for redundancy.
- The system encrypts and uploads data to distributed cloud environments improving resilience. Regular testing and monitoring is conducted to ensure security standard and validate recovery performance.

```
# ----- Dashboard -----
st.title("Dashboard")
st.write(f"Welcome, {(st.session_state.username)} {(st.session_state.role)}!")
st.markdown("<div class='muted'>Environment: local prototype - Files are stored on this machine only.</div>", unsafe_allow_html=True)

# Upload section
with st.container():
    st.markdown("### Upload a file")
    st.markdown("<div class='card'>", unsafe_allow_html=True)
    col1, col2 = st.columns([3,1])
    with col1:
        uploaded = st.file_uploader("Choose any file", accept_multiple_files=False)
    with col2:
        st.write("")
        st.write("")
        if st.button("Upload", use_container_width=True):
            if uploaded is None:
                st.warning("Select a file before uploading.")
            else:
                ok, msg = save_file(uploaded, st.session_state.username)
                if ok:
                    st.success(msg)
                else:
                    st.error(msg)
    st.markdown("</div>", unsafe_allow_html=True)

# Backups list
st.markdown("### Your backups")
files = list_files()
if not files:
    st.info("No files uploaded yet.")
else:
    for f in files:
        with st.container():
            st.markdown("<div class='card'>", unsafe_allow_html=True)
            c1, c2, c3 = st.columns([4,2,1])
            with c1:
                st.write(f"***{f}***")
```

CODE SNIPPET



RESULT

Deploy

Dashboard

Welcome, **demo** (user)!

Environment: local prototype • Files are stored on this machine only.

Upload a file

Choose any file

Drag and drop file here
Limit 200MB per file

Browse files

Upload

Your backups

No files uploaded yet.

Backup History

	file	user	action	time
0	assignment 3 - Restoring AdventureWorks2014.pdf	demo	delete	2026-04-07 21:54:44.003123

Share Info (Simulation)

Select user to share with

admin

Share



CONCLUSION

- Small Businesses have issues with important data and files
- Our system aims to better those issues and make it less expensive for those small businesses.