

# AWS Google Drive Clone - Chat Guide

User: I want to create a personal google drive clone to store my documents, images and videos, what architecture do I need to get that up and running using AWS?

Assistant: To create a personal Google Drive clone on AWS for storing documents, images, and videos, you can leverage AWS services and a multi-tier architecture that provides scalability, security, and cost-efficiency. Here's a high-level architecture breakdown:

1. **Frontend (User Interface)**: React (or Angular/Vue) for file management, hosted on AWS Amplify, S3, or Elastic Beanstalk.
2. **Backend (Application Layer)**: Node.js (Express) or Python (Flask/Django) deployed on AWS Lambda or EC2.
3. **Authentication and User Management**: AWS Cognito for sign-in, sign-up, MFA, and user management.
4. **Storage Layer**: AWS S3 for storing documents, images, and videos with versioning and lifecycle policies.
5. **Database Layer**: AWS DynamoDB or Amazon RDS (MySQL/PostgreSQL) for metadata storage.
6. **File Uploads and API Gateway**: Use API Gateway with Lambda for presigned URLs for file uploads and downloads.
7. **Media Processing (Optional)**: AWS Elastic Transcoder for video transcoding.
8. **Search Functionality**: AWS Elasticsearch or OpenSearch for searching file metadata.
9. **Logging and Monitoring**: AWS CloudWatch and CloudTrail for performance and security monitoring.

10. **Security and Permissions**: Use IAM and KMS for securing access to resources.

This architecture is scalable, secure, and cost-efficient for building a Google Drive-like application on AWS.

User: How long would you estimate someone with junior dev skills to get a prototype working?

Assistant: For a junior developer with basic AWS knowledge, it would take around **4-6 weeks** to get a working prototype. Here's a rough breakdown:

Week 1: Learn AWS basics and set up frontend (2-5 days).

Week 2: Implement S3 storage and backend for file uploads/downloads (4-6 days).

Week 3: Add user authentication with AWS Cognito and set up a database (5-6 days).

Week 4: Implement UI enhancements and CRUD file operations (5 days).

Week 5: Add search functionality and optimize performance/security (4-6 days).

Week 6: Test, debug, and deploy (3-5 days).

Complexity and learning curve may extend the timeline.

User: Where would you recommend I start?

Assistant: Here's a roadmap for where to begin:

1. **Learn AWS basics**: Start by familiarizing yourself with AWS S3, Lambda, API Gateway, Cognito, and DynamoDB.

2. **Set up S3 for file storage**: Create an S3 bucket and experiment with uploading and

downloading files.

3. **Build a Lambda backend**: Write Lambda functions to handle file uploads/downloads and integrate with API Gateway.
4. **Implement user authentication**: Use AWS Cognito for user sign-up, login, and secure file access.
5. **Set up a database**: Choose DynamoDB or RDS for storing file metadata.
6. **Build the frontend**: Use React or another framework to create a simple UI for file management.
7. **Add additional features**: File search, versioning, and media processing can be added later.

By following this progression, you'll gradually build your Google Drive clone while learning AWS.

User: Can you transcribe this chat into a well-organized PDF for future referral?

Assistant: Sure! Here's the PDF transcription of our chat.