Project Charter

💡 Kickstart project planning and ensure stakeholder alignment using this project charter.

- Use AI to create and summarize content.
- Accelerate review cycles by assigning comments to reviewers directly from the Doc.
- Connect Docs with tasks and projects using links and relationships.

Section One

Project Title	Secure Cloud Governance - Least Privilege AWS Environment
Project Duration	5-7 Days
Project Manager	Armon Jackson
Project Sponsor	Self-Directed Portfolio Project

Purpose and Background

This project establishes a secure, low-cost AWS environment designed around least privilege and foundational governance. The purpose is to demonstrate practical cloud project management skills, reduce misconfiguration risk, and build a repeatable environment that aligns with security best practices. This setup will serve as the baseline for future automation and cloud portfolio work.

Objectives

- Deploy a private VPC with subnets and routing.
- Configure IAM roles/policies using least privilege.
- Set up S3 with all public access blocked.
- Enable billing alarms to prevent accidental charges.
- Document governance, risks, and lessons learned.
- Publish the project and diagram to GitHub for employers.

In-Scope

- AWS IAM, S3, VPC, Billing Alarms
- Architecture diagram creation (draw.io)
- Documentation (charter, risk register, lessons learned)

Out-of-Scope

- Advanced automation (CloudFormation, Terraform)
- Multi-account AWS Organizations
- CI/CD pipelines

Key Deliverables

- VPC deployed with subnets, routing, and security groups
- IAM least-privilege roles/policies
- S3 configured with Block Public Access
- Billing alarm with SNS notification
- Architecture diagram (PNG)
- Project documentation (Charter, Risk Register, Lessons Learned)
- GitHub repo containing screenshots and documentation

Success Criteria

- Environment deploys without errors
- No publicly accessible resources
- Billing alarm triggers correctly
- Documentation is complete, organized, and professional
- GitHub repo is clean and employer-ready

Stakeholders

- **Project Manager:** Armon
- **Project Reviewer:** Hiring managers / recruiters
- **End User:** Future cloud PM portfolio reviewers
- **Cloud Platform:** AWS Free Tier account

Timeline

Total duration: 5–7 days

- Setup & VPC: Day 1
- IAM & S3: Day 2
- Billing alarm: Day 3
- Diagram & documentation: Days 4–5
- GitHub upload: Day 6

Assumptions & Constraints

Assumptions:

- AWS Free Tier is available
- Internet connectivity stable
- No production workloads

Constraints:

- Free-tier cost limits
- Solo developer, limited time
- Must be simple enough to repeat

Risk Register—Secure Cloud Governance Project

Risk Description	Cause	Impact	Probability	Severity	Response Plan
Unexpected AWS charges	Misconfiguration or resources left running	Cost overages	Medium	High	Configure billing alarms; delete unused resources daily
IAM permissions too open	Overusing AdministratorAccess	Security exposure	Low	High	Use least-privilege, review every policy manually
IAM permissions too restrictive	Over-limiting policies	Tasks fail, deployment delays	Medium	Medium	Test each role; adjust based on required actions
Public S3 buckets	Default config mistakes	Data exposure	Low	High	Enable “Block Public Access” at both account and bucket level
Incorrect VPC routing	Wrong route table/subnet config	Services can’t communicate	Medium	Medium	Test with ping/curl; validate route tables
EC2 or other paid resources created accidentally	Mis-click or testing	Unwanted charges	Medium	High	Use IAM permissions that restrict EC2 creation unless needed

Forget to delete environment after project	Human error	Long-term charges	Medium	High	Create “Shutdown Checklist”; delete stack after repo upload
GitHub repo leaks info	Uploading AWS keys/screenshots with sensitive data	Account compromise	Low	High	Never screenshot keys; rotate credentials; use .gitignore

Section Two

Lessons Learned

During the setup of the Secure Cloud Governance environment, I learned the importance of slowing down and validating each service configuration before moving on to the next. IAM, S3, and VPC settings are highly interconnected, and small misconfigurations can break functionality or create security gaps. Creating a clear sequence of tasks in ClickUp helped prevent rework and kept the project aligned with least-privilege principles.

I also learned the importance of cost control when working in AWS. Even free-tier accounts can generate unexpected charges if resources are misconfigured or forgotten. Setting billing alerts early in the project proved to be one of the most valuable safeguards. Finally, documenting each step made the GitHub readme easier to build and showed how cloud PMs translate technical outcomes into consumable project artifacts.