









### PROVIDER CREDENTIALING

Verifies provider qualifications with insurers to ensure claims are processed smoothly.

Credentialing begins with the group, followed by individual providers within the group.



### VOB / PRIOR AUTHORIZATION

Confirms patient insurance coverage and benefit eligibility.

Secures prior approvals for procedures to prevent claim denials.

Documents total treatment hours per day/week to align with payor requirements.



### MEDICAL CODING

Converts diagnoses, treatments, and procedures into standardized billing codes.

Covers both ABA & Non-ABA services to ensure compliance.



### BILLING & REJECTIONS

Prepares, validates, and submits claims to insurers for reimbursement.

ABA Claims: Pre-filled based on EMR documentation by providers.

Non-ABA Claims: Requires superbill notes & manual data entry by agents to EMR.



### AR, DENIALM

Tracks unpaid claims, follows up with insurers, and reduces outstanding balances.

Analyzes denial reasons, appeals rejections, and ensures maximum reimbursement.

Identifies claim errors, missing info, and payer denials to correct and resubmit claims.



### PAYMENT POSTING

Records payments from insurers & patients, reconciles accounts, and resolves underpayments.

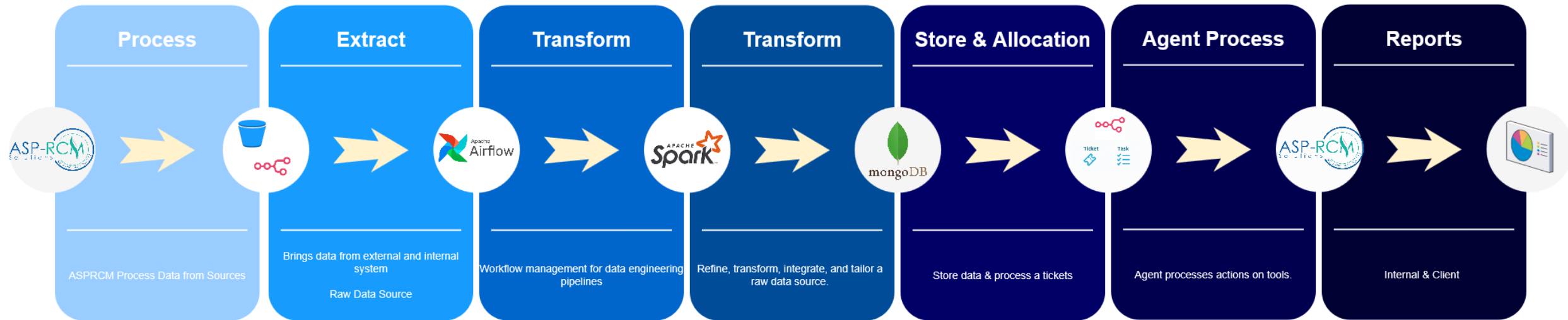
Matches payments to claims for accurate financial tracking.

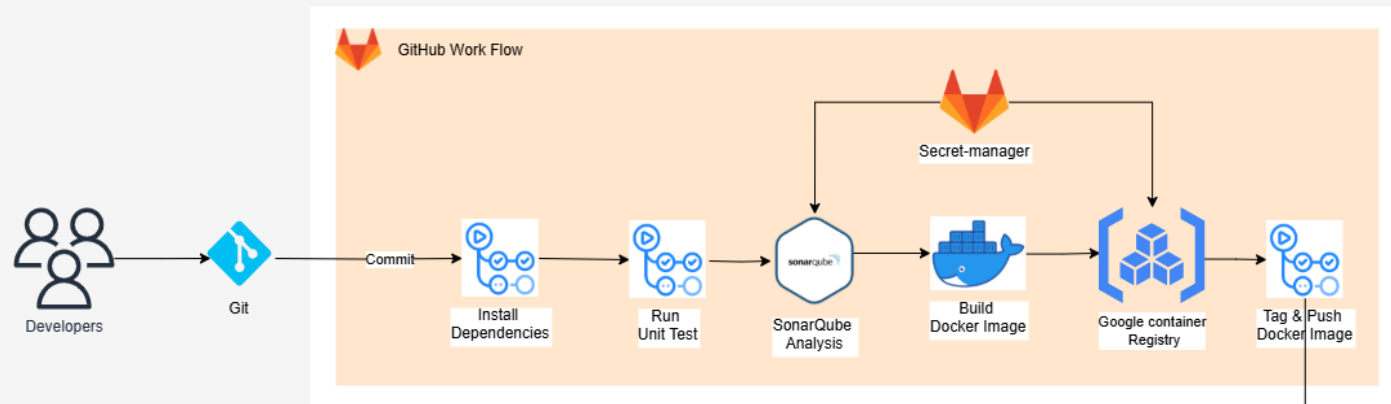


### PATIENT INVOICING

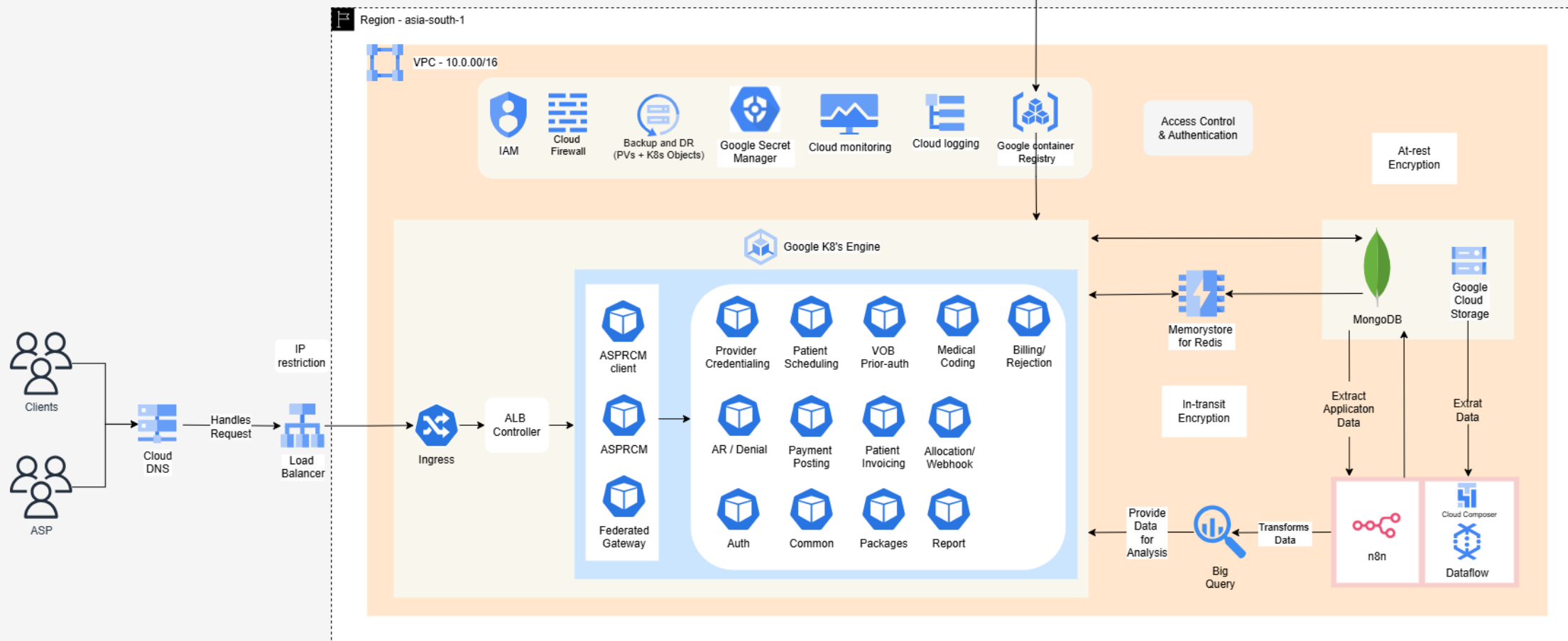
Generates patient statements, follows up on balances, and manages collections.

Ensures patients are notified on outstanding balances to improve payment turnaround.





CI



CD

# n8n vs Make vs Zapier vs Airflow vs GCP

Category	Feature	n8n (Best for Advanced Use Cases)	Make (Best for No-Code ETL)	Zapier (Best for Simple Automation)	Apache Airflow + Spark (Best for Complex Data Pipelines)	GCP Dataflow + Cloud Composer (Best for Scalable Data Orchestration)
Hosting & Deployment	Cloud-Based	✔ Yes	✔ Yes	✔ Yes	✔ Yes (Managed on AWS, GCP, etc.)	✔ Yes (Fully Managed)
	Self-Hosted	✔ Yes (Docker, Kubernetes)	✘ No	✘ No	✔ Yes (Open-source)	✘ No (Managed only)
	On-Premises Deployment	✔ Yes	✘ No	✘ No	✔ Yes	✘ No
AI & Machine Learning	Supports AI APIs (OpenAI, Hugging Face, etc.)	✔ Yes	✔ Yes	✔ Yes	✘ No (Requires external scripts)	✔ Yes (Vertex AI Integration)
	Custom AI Model Integration	✔ Yes (Self-hosted possible)	✘ No	✘ No	✔ Yes (Python-based workflows)	✔ Yes (TensorFlow, BigQuery ML)
	AI-based Data Processing (ML workflows)	✔ Advanced (JS/Python)	⚠ Limited	✘ No	✔ Yes (Custom ML Pipelines)	✔ Yes (Apache Beam + ML APIs)
Workflow Complexity	Multi-Step Automation	✔ Yes (Highly customizable)	✔ Yes	✔ Yes	✔ Yes (DAG-based)	✔ Yes (Cloud Composer DAGs)
	Conditional Logic, Loops, and Branching	✔ Yes	✔ Yes	✔ Yes	✔ Yes	✔ Yes
	Advanced Error Handling	✔ Yes	✔ Yes	✘ No	✔ Yes	✔ Yes

# n8n vs Make vs Zapier vs Airflow vs GCP

Category	Feature	n8n (Best for Advanced Use Cases)	Make (Best for No-Code ETL)	Zapier (Best for Simple Automation)	Apache Airflow + Spark (Best for Complex Data Pipelines)	GCP Dataflow + Cloud Composer (Best for Scalable Data Orchestration)
ETL (Extract, Transform, Load)	Extract Data from APIs, DBs, Files	✔ Yes	✔ Yes	✔ Yes	✔ Yes	✔ Yes
	Data Transformation (Regex, Scripting)	✔ Yes	⚠ Limited	✘ No	✔ Yes (Python, SQL, Pandas)	✔ Yes (Apache Beam, SQL)
	Load Data to Multiple Destinations	✔ Yes	✔ Yes	✘ No	✔ Yes	✔ Yes
Code Flexibility & Custom Logic	Custom Scripting (JS, Python)	✔ Yes	✘ No	✘ No	✔ Yes (Python-based)	✔ Yes (Python, Java, Beam)
	API/Webhook Support	✔ Yes	✔ Yes	✔ Yes	✔ Yes	✔ Yes
	Webhook Triggers	✔ Yes	✔ Yes	✔ Yes	✔ Yes	✔ Yes
Load Handling & Scalability	Handles High Workloads	✔ Yes (Parallel Execution)	⚠ Limited	✘ No	✔ Yes (Optimized for Large Data)	✔ Yes (Serverless Auto-Scaling)
	Parallel Execution	✔ Yes	✔ Yes	✘ No	✔ Yes (Task dependencies)	✔ Yes (Autoscaling Workers)
	Bulk Processing & Queues	✔ Yes	✔ Yes	✘ No	✔ Yes	✔ Yes



# n8n vs Make vs Zapier vs Airflow vs GCP

Category	Feature	n8n (Best for Advanced Use Cases)	Make (Best for No-Code ETL)	Zapier (Best for Simple Automation)	Apache Airflow + Spark (Best for Complex Data Pipelines)	GCP Dataflow + Cloud Composer (Best for Scalable Data Orchestration)
Excel & File Processing	Extract Data from Excel/CSV	✔ Yes	✔ Yes	✔ Yes	✔ Yes (Using Pandas, Apache POI)	✔ Yes (Cloud Storage + BigQuery)
	Handle Multiple Excel Files	✔ Yes	✔ Yes	⚠ Limited	✔ Yes	✔ Yes
	Transform Excel Data (Merge, Filter, Edit)	✔ Yes	✔ Yes	✖ No	✔ Yes	✔ Yes (SQL, Python, Beam)
	Write Back to Excel/Google Sheets	✔ Yes	✔ Yes	✔ Yes	✔ Yes	✔ Yes
Report Generation	Automated Report Generation	✔ Yes	✔ Yes	✔ Yes	✔ Yes	✔ Yes
	Generate PDFs, Excel Reports	✔ Yes	✔ Yes	⚠ Limited	✔ Yes (Jupyter, Superset, Custom)	✔ Yes (Looker, Data Studio)
	Merge Data from Multiple Sources	✔ Yes	✔ Yes	✖ No	✔ Yes	✔ Yes
Pricing & Free Tier	Free Plan Availability	✔ Yes	✔ Yes	✔ Yes	✔ Yes (Open-source)	✔ Yes (Pay-as-you-go)
	Paid Plan Starting Price	💰 Lower	💰 Medium	💰 Expensive	✔ Free (Self-hosted) / Paid (Managed)	💰 Pay-per-use (GCP Billing)

# n8n vs Make vs Zapier vs Airflow vs GCP

Category	Feature	n8n (Best for Advanced Use Cases)	Make (Best for No-Code ETL)	Zapier (Best for Simple Automation)	Apache Airflow + Spark (Best for Complex Data Pipelines)	GCP Dataflow + Cloud Composer (Best for Scalable Data Orchestration)
API & Developer Friendliness	API Rate Limits	✔ High	✔ High	⚠ Medium	✔ High	✔ High (GCP Quotas)
	Custom Connectors	✔ Yes	✔ Yes	✔ Yes	✔ Yes	✔ Yes (GCP APIs, Cloud Functions)
	Marketplace for Prebuilt Workflows	✔ Yes	✔ Yes	✔ Yes	✖ No (Custom Python Workflows)	✖ No (Custom DAGs)
Ease of Use	Beginner-Friendly UI	✔ Yes	✔ Yes	✔ Yes	✖ Code-based Interface	✖ Requires Python Skills
	Drag-and-Drop Workflow Builder	✔ Yes	✔ Yes	✔ Yes	✖ No (Requires Coding)	✖ No (DAGs in Python)
	Best for Non-Developers	✔ Yes	✔ Yes	✔ Yes	✖ No (Developer-Focused)	✖ No (GCP Experience Needed)
Integrations	Number of Prebuilt Integrations	✔ 200+	✔ 1000+	✔ 5000+	🔧 Custom via Python	🔧 Custom via GCP Services
	Support for SaaS Apps	✔ Yes	✔ Yes	✔ Yes	✔ Yes	✔ Yes
	Support for On-Prem Apps	✔ Yes	✔ Yes	✖ No	✔ Yes	✔ Yes

# Dataprep vs Data Fusion vs Dataproc vs Dataflow vs Airflow Vs n8n

Service	Purpose	Pricing Model	100GB/day (Monthly Cost)	1TB/day (Monthly Cost)	Best For	Visual Interface	Handling Large Workflows (e.g., A140 + B140 join → AB140 transform)
<b>Dataprep</b>	Data cleaning, transformation	Per-GB processed	\$3,000 – \$6,000	\$30,000+ 🚀	Data cleaning (expensive at scale)	✅ No-code UI (Trifacta, drag & drop for filters/joins)	⚠️ Good for small/medium joins but very costly for large workflows
<b>Data Fusion</b>	ETL, data integration	Per-instance + vCPU-based	\$1,296 – \$3,000	\$1,500 – \$4,000	No-code ETL with hybrid cloud	✅ Drag & drop UI (CDAP)	✅ Good for large-scale joins & transformations
<b>Dataproc + Cloud Composer (Airflow)</b>	Big data processing (Hadoop, Spark) with workflow orchestration	Per-cluster (vCPU & memory-based) + Cloud Composer orchestration	\$864 – \$1,200 + \$300 – \$600	\$1,000 – \$1,500 + \$600 – \$1,000	Spark/Hadoop batch processing at scale	⚠️ Limited UI (DAG-based, CLI for Spark jobs)	✅ Best for massive joins on structured data (Hadoop/Spark optimizations)
<b>Dataflow (Spark) + Cloud Composer (Airflow)</b>	Real-time & batch ETL with workflow orchestration	Per-job + vCPU usage + Cloud Composer orchestration	\$500 – \$2,000 + \$300 – \$600	\$1,200 – \$5,000 + \$600 – \$1,000	Event-driven real-time ETL	⚠️ Limited UI (Apache Beam Visualizer)	✅ Best for streaming & event-driven transformations
<b>Apache Airflow + Spark</b>	Workflow orchestration & Spark-based ETL	Self-hosted (VM cost + Spark cluster)	\$500 – \$1,500	\$1,000 – \$3,000	Complex workflow orchestration	⚠️ Limited UI (DAG-based, Python scripts required)	✅ Best for scheduling massive ETL pipelines but requires coding
<b>n8n</b>	Low-code automation & API workflows	Self-hosted (or Cloud per-execution pricing)	\$100 – \$500 (self-hosted) / \$500+ (Cloud)	\$500 – \$2,500 (Cloud pricing varies)	Low-code automation, API integrations	✅ Drag & drop UI	❌ Not optimized for huge joins & transformations (better for automation)

# Dataflow vs Airflow

Factor	Apache Airflow + Spark	Dataflow + Cloud Composer
Ease of Setup	✗ Requires manual deployment, scaling, and monitoring	✓ Fully managed by GCP
Scalability	⚠ Needs cluster autoscaling setup	✓ Auto-scales with demand
Maintenance	✗ Requires infra management (VMs, Kubernetes, etc.)	✓ No maintenance overhead
Streaming Processing	⚠ Needs structured Spark Streaming setup	✓ Beam-native, optimized for real-time
Integration with GCP Services	✗ Requires manual connectors	✓ Built-in GCS, BigQuery, Pub/Sub support
Monitoring & Debugging	⚠ Requires external tools (Grafana, Prometheus)	✓ Built-in monitoring, logging
Total Cost	✓ Lower long-term if optimized	✗ Higher due to managed pricing

# Dataflow vs Airflow

Setup	Airflow Cost	Compute Cost (Spark/Dataflow)	Storage/Networking	Total Monthly Cost
Self-Hosted Airflow + Spark (Medium - 12 hrs/day)	~\$100 – \$200	~\$600	~\$140	\$900/month
Self-Hosted Airflow + Spark (Large - 24 hrs/day)	~\$500	~\$1,500	~\$200	\$2,200/month
Self-Hosted Airflow + Spark (Heavy - 24 hrs/day, High Throughput)	~\$800	~\$3,000	~\$300	\$4,100/month
GCP Dataflow + Cloud Composer (Batch Medium - 12 hrs/day)	~\$300 – \$600	~\$500 – \$1,000	~\$100	\$1,000 – \$1,500/month
GCP Dataflow + Cloud Composer (Batch Large - 24 hrs/day)	~\$600 – \$800	~\$2,000 – \$3,000	~\$300	\$2,900 – \$4,100/month
GCP Dataflow + Cloud Composer (Batch Heavy - 24 hrs/day, High Throughput)	~\$1,000	~\$5,000	~\$500	\$6,500/month
GCP Dataflow + Cloud Composer (Batch Heavy + Low Streaming - Mixed Workloads)	~\$1,200	~\$6,000	~\$700	\$7,900/month

Requirement	Explanation & Detail / Sub-checklist	Required for HIPAA?	Status
HIPAA-eligible region	Use asia-south1 or any other HIPAA-compliant GCP region	✅ Yes	🟢 Planned
Composer V2 with Private IP	Use Composer <b>V2</b> with <b>private IPs</b> to isolate environment inside VPC	✅ Yes	🟢 Planned
VPC-native environment	Place Composer in a <b>VPC-native subnet</b> (no public IPs)	✅ Yes	🟢 Planned
Shielded VMs enabled	Use Shielded VMs to ensure boot-time verification and hardened OS	✅ Yes	🟢 Planned
CMEK encryption for buckets	Use <b>Customer-Managed Encryption Keys</b> for Composer DAG bucket and logs bucket	✅ Yes	🟢 Planned
Secret Manager integration	Store secrets (tokens, DB creds) in <b>Secret Manager</b> ; never use Airflow Variables or env vars for secrets	✅ Yes	🟢 Planned
Least privilege IAM for Airflow UI	Assign Composer roles (Admin, Viewer) only as needed via IAM; never use Editor or Owner broadly	✅ Yes	🟢 Planned
Composer SA isolation	Create a dedicated <b>Service Account</b> with least privilege for Composer environment	✅ Yes	🟢 Planned
Audit logs for DAGs & API calls	Enable <b>Cloud Audit Logs</b> for Composer, Airflow UI, DAG/API actions	✅ Yes	🟢 Planned
No PHI in logs or XComs	Avoid logging PHI or passing PHI via Airflow <b>XComs</b> , logs, or metadata	✅ Yes	🟢 Planned
Application-level PHI access logs	Log metadata for all DAGs accessing PHI: DAG ID, trigger source, SA/user, job result	✅ Yes	🟢 Planned
Version control for DAGs	Use Git-based CI/CD (GitLab, GitHub Actions) to deploy DAGs	✅ Yes	🟢 Planned
Monitoring for DAG/task failures	Track changes, rollbacks, approvals	🚩 Recommended	🟢 Planned
Dead-letter handling for failures	Set alerts in <b>Cloud Monitoring</b> for failed or delayed DAGs/tasks	✅ Yes	🟢 Planned
Limit external network egress	Catch and store failed tasks/records in a <b>DLQ</b> bucket (CMEK-encrypted)	✅ Yes	🟢 Planned
Time-based or label-based IAM	Minimize/disable internet access	✅ Yes	🟢 Planned
Set environment variable whitelist	Use <b>IAM Conditions</b> to allow DAG triggers only at certain times or by network origin	🚩 Recommended	🟢 Planned
Airflow Web UI hardened access	Avoid injecting secrets via env vars in Composer	✅ Yes	🟢 Planned
DAG-level RBAC in Airflow UI	Allow only non-sensitive configs	✅ Yes	🟢 Planned
Restrict Airflow REST API	Secure Web UI using <b>IAP</b> , VPN, or private IP; avoid exposing Composer's Web UI publicly	✅ Yes	🟢 Planned
Isolate PHI and non-PHI workflows	Enable <b>Airflow UI RBAC mode</b> to control DAG visibility and editing by user role	✅ Yes	🟢 Planned
Use DAG tags or metadata	Disable or protect Airflow's REST API with strict token auth	✅ Yes	🟢 Planned
Limit Airflow Plugin usage	Deploy sensitive DAGs in separate Composer environments or use DAG tagging and RBAC	🚩 Recommended	🟢 Planned
Secure scheduler → external calls	Tag DAGs with labels like sensitive=true, data_type=PHI for audit filtering	✅ Yes	🟢 Planned
Log rotation and retention policy	Only install <b>vetted plugins</b> ; disable user install rights	✅ Yes	🟢 Planned
Secure DAG retries and idempotency	Scheduler should only call trusted internal services; use private DNS, internal load balancers	✅ Yes	🟢 Planned
Use CI/CD for DAG deploys	Set log expiration in <b>GCS</b> (e.g., delete after 6 years); CMEK-enabled & versioned	✅ Yes	🟢 Planned
Composer environment labeling	Ensure DAGs can retry without duplicating PHI writes or leaking state	✅ Yes	🟢 Planned
	Avoid UI uploads; use CI/CD for deployments with approvals & audit trail	✅ Yes	🟢 Planned
	Tag Composer env with labels (hipaa=true, env=prod) to scope access, filter logs	🚩 Recommended	🟢 Planned

## GCP CloudComposer

## HIPAA Compliance Checklist

GCP Dataflow

HIPAA Compliance Checklist

Requirement	Explanation & Detail / Sub-checklist	Required for HIPAA?	Status
HIPAA-eligible region	Use regions like asia-south1 that are part of GCP's HIPAA-compliant list	✔ Yes	● Planned
Private IP workers	Set WORKER_IP_PRIVATE to ensure workers use private IPs	✔ Yes	● Planned
	Avoid public egress via NAT		
VPC-native subnetwork	Workers should be deployed in a VPC-native subnet scoped to HIPAA workloads only	✔ Yes	● Planned
CMEK encryption (input/output)	Use <b>Customer-Managed Encryption Keys</b> for all input/output storage (GCS and BigQuery)	✔ Yes	● Planned
No public bucket access	GCS input/output/staging buckets must have publicAccessPrevention = enforced and no IAM allUsers/allAuthenticatedUsers	✔ Yes	● Planned
Cloud Audit Logs	Enable Admin Activity and Data Access logs for Dataflow, GCS, BigQuery	✔ Yes	● Planned
App-level PHI logging control	Never log raw PHI	✔ Yes	● Planned
	Log only record metadata (record_id, job_id, timestamps)		
Monitoring and alerting	Use centralized audit log format	⚠ Recommended	● Planned
	Set up Cloud Monitoring to track job health, failures, retries		
Data validation / lineage tracking	Send alerts to PagerDuty/Slack/email	✔ Yes	● Planned
	Validate inputs to reject malformed PHI		
VPC Service Controls	Log source/destination of PHI	✔ Yes	● Planned
	Tag outputs with lineage labels or metadata		
Dead-letter bucket (secure)	Use <b>VPC-SC perimeter</b> around Dataflow, GCS, BigQuery	✔ Yes	● Planned
	Prevents data exfiltration outside org		
Job parameter audit logging	Monitor breach logs	✔ Yes	● Planned
	Use CMEK-encrypted bucket for failed records		
CMEK for temp/shuffle storage	Log job + timestamp	✔ Yes	● Planned
	Delete after retention period		
Service Account key management	Log job config (source paths, parameters, runtime) without PHI	✔ Yes	● Planned
	Tag job runs with owner, purpose, dataset		
Data retention policies (GCS/BQ)	Ensure temp and staging locations are encrypted with CMEK	✔ Yes	● Planned
	Set auto-delete lifecycle on temp files		
Job ownership / access review	Use short-lived or federated SA credentials	✔ Yes	● Planned
	Rotate keys regularly		
Data retention policies (GCS/BQ)	Disable unused keys	✔ Yes	● Planned
	Configure GCS lifecycle rules to delete after X days		
Job ownership / access review	Set BQ table expiration	✔ Yes	● Planned
	Log all deletions		
Job ownership / access review	Assign job owner via label/metadata	✔ Yes	● Planned
	Review IAM permissions quarterly		
Job ownership / access review	Track user access to PHI data pipelines	✔ Yes	● Planned

# Database










Feature	MongoDB	PostgreSQL	Cassandra	BigQuery
Transaction Load (OLTP)	✔ High transaction volume, handles high throughput	✔ Full ACID compliance, great for transactional systems	✗ Limited support for transactions, eventually consistent	✗ Not suitable for OLTP workloads, analytics-focused
Dynamic Schema Support	✔ Highly flexible schema, dynamic collections	✔ JSONB support for semi-structured data	✔ Schema-less (Wide Column Store)	✔ Flexible schema (tables can change structure, but schema is less dynamic compared to NoSQL)
Data Ingestion	✔ Fast for high-frequency, real-time data	✔ Good support, but can become slow with huge datasets	✔ Optimized for high-volume writes, distributed	✔ Ideal for batch processing, not designed for real-time writes
Integration with Apache Airflow	✔ Good integration using operators and connectors	✔ Native support for PostgreSQL operators	✔ Good integration using connectors for distributed systems	✔ Native integration with Airflow using operators and connectors
Integration with n8n	✔ Native connectors for MongoDB	✔ Native connectors for PostgreSQL	✔ Can be integrated using n8n connectors	✔ Native connectors for BigQuery
Support for Reporting Tools (Power BI, Looker, etc.)	✔ Native support for reporting tools (Power BI, Tableau) via connectors	✔ Full support for reporting tools like Power BI, Looker, and Tableau	✗ Limited reporting support, best for operational queries	✔ Excellent integration with reporting tools like Looker, Power BI, Tableau for analytics
Read Performance (Querying)	✔ Good for high read/write throughput on unstructured data	✔ Strong performance for complex queries and joins	✔ Optimized for high-throughput reads but not for complex queries	✔ Excellent for fast analytical queries on large datasets, not ideal for OLTP
Write Performance	✔ High write throughput, supports high-volume operations	✔ Good, but can slow down with high-frequency writes	✔ Very high write throughput, optimized for write-heavy workloads	✗ Not optimized for frequent writes, designed for analytics
Scalability	✔ Horizontal scaling with sharding	✔ Vertical scaling (good for moderate scaling needs)	✔ Horizontal scaling (distributed architecture)	✔ Massively scalable, horizontal scaling on Google Cloud
Cost	? Cost depends on deployment and infrastructure	? Can be expensive with high-volume, transactional data	? Low operational costs for distributed systems	? Pay-per-query model, cost-effective for large queries but can be expensive for frequent queries
Consistency Model	✔ Tunable consistency (eventual or strong consistency)	✔ Strong consistency (ACID compliant)	✗ Eventual consistency, not suitable for strict consistency requirements	✗ Eventual consistency, optimized for analytics
Backup and Recovery	✔ Backup through replication, easy recovery	✔ Strong backup support with WAL (Write-Ahead Logging)	✔ Good for distributed backup solutions	✔ Snapshot and backup via Google Cloud, great for large datasets
Maintenance Overhead	✔ Moderate, needs sharding and scaling management	✔ Moderate, but requires more management for scaling	✔ Low overhead for large write loads, but can be complex to manage	✔ Low maintenance overhead, fully managed by Google Cloud
Use Case Suitability	Good for real-time applications, flexible data models, and high-frequency writes	Best for traditional relational databases, complex queries, and OLTP	Excellent for write-heavy, distributed systems like IoT or log data	Best for large-scale analytics, reporting, and data warehousing



# MongoDB EnterpriseEdition vs CommunityEdition

Feature/Capability	Community Edition	Enterprise Edition
License	Open-source (SSPL)	Commercial (requires subscription)
Basic DB Functionality	✔ Yes	✔ Yes
Authentication & Authorization	✔ Basic	✔ Advanced (LDAP, Kerberos, etc.)
<b>Encryption at Rest (TDE)</b>	✘ No	✔ Yes
FIPS 140-2 Compliance	✘ No	✔ Yes
<b>Auditing / Audit Logs</b>	✘ No	✔ Yes
Client-Side Field Level Encryption (FLE)	✔ Yes (≥4.2)	✔ Yes (with support)
Ops/Monitoring (Ops Manager)	✘ No	✔ Yes (GUI, backup, automation)
In-memory Storage Engine	✘ No	✔ Yes
Hot Backups (No downtime)	✘ No	✔ Yes
Support from MongoDB Inc.	✘ Community only	✔ Enterprise Support
BI Connector & Analytics	✘ No	✔ Yes
MongoDB Atlas	✘ Not included	✔ Integrated

# MongoDB Enterprise Self-Hosted vs Atlas

Feature / Cost Area	Enterprise Self-Hosted	MongoDB Atlas (Managed)
Base Cost	 ~\$10K–\$100K+/year (license only)	Starts at ~\$60/month, pay-as-you-go
Infrastructure (servers, VMs)	You pay (e.g., EC2, GCP VMs, on-prem hardware)	Included in the cost (MongoDB manages everything)
Ops/Backups/Monitoring	You build + maintain	Fully managed (automated)
TDE (Encryption at Rest)	 Included (with setup)	 Included
LDAP / Kerberos Auth	 Included	 Included in Atlas Enterprise tier
Audit Logs	 Yes	 Yes (Enterprise tier)
Support	Paid support via license	Tiered support (Basic, Standard, Premium)
Scaling	Manual or scripted (Terraform, Ansible, etc.)	Auto-scaling built-in
Backups	You manage	Included (with point-in-time recovery)
High Availability	You configure (replica sets, sharding)	Built-in
Setup Time	Hours to days (manual config)	Minutes (GUI or API)
Compliance (HIPAA, etc.)	Possible with effort	Built-in options (SOC2, ISO, HIPAA, etc.)
BI Connector / Analytics	 Yes	 Yes (Enterprise tier)

Resource		Value
Tier		M20 (shared infra)
vCPUs		1 vCPU
RAM		~3.75 GB
Storage		20 GB
Price		\$0.19/hr (~\$138.70/month)
Extra storage		10GB × \$0.25
Total		\$138.70 + \$2.50 = ~\$141.20/month
Backup Retention		✔ 30 Days Continuous Snapshot = 30 × \$0.20 = \$6.00/month
Oplog Backup (5 GB est.)		5 × \$1.00 = \$5.00/month
Compliance		✔ HIPAA-ready
Total		\$174.10/month

## MongoDB Atlaspricing on GCP

Requirement	Details / Notes	Required for HIPAA?	Status
Dedicated Cluster	M10 or higher	✔ Yes	✔ Enabled (M20)
Signed BAA	Via MongoDB Sales	✔ Yes	● Will Have
Encryption at Rest	Enabled by default, FIPS 140-2 compliant	✔ Yes	✔ Enabled
Encryption in Transit (TLS)	Enabled by default	✔ Yes	✔ Enabled
IP Whitelisting / VPC Peering	Network-level access restriction	✔ Yes	● Will Have
Access Control (RBAC)	Least-privilege roles, secure app/database user handling	✔ Yes	● Will Have
30-Day Continuous Backups	Upgrade from 15d to 30d PITR	✔ Yes	● Will Have
Audit Logging	Tracks access, role changes (requires Atlas Enterprise)	⚠ Recommended	● Will Have
Monitoring & Alerts	Set thresholds + notify on unusual activity	⚠ Recommended	● Will Have
Data Retention & Deletion Policy	Define, automate PHI lifecycle	✔ Yes	● Will Have
Internal Docs & Training	SOPs, staff awareness, logs	✔ Yes	● Will Have

## MongoDB Atlas

## HIPAA Compliance Checklist

Requirement	Explanation & Detail / Sub-checklist	Required for HIPAA?	Status
HIPAA-eligible region	Use HIPAA-approved region (e.g., asia-south1, us-central1) for dataset storage	✅ Yes	🟢 Planned
CMEK encryption for datasets	Use <b>Customer-Managed Encryption Keys (CMEK)</b> for all datasets containing PHI	✅ Yes	🟢 Planned
Dataset-level IAM	Use granular IAM roles (dataViewer, dataEditor, etc.) at the <b>dataset level</b> , not project-wide	✅ Yes	🟢 Planned
Fine-grained table access control	Use table-level or <b>column-level ACLs</b> to restrict PHI access to specific users	✅ Yes	🟢 Planned
Row-level security (RLS)	Implement <b>row-level security policies</b> to restrict PHI visibility within a shared table	⚠️ Recommended	🟢 Planned
Column-level encryption/masking	Use application-side or UDF-based field masking, or encrypt sensitive columns (e.g., SSN, DOB)	✅ Yes	🟢 Planned
Disable public dataset access	Ensure <b>no dataset or table</b> is shared with allUsers or allAuthenticatedUsers	✅ Yes	🟢 Planned
BigQuery audit logs	Enable <b>Data Access Logs</b> for all read/write/query operations	✅ Yes	🟢 Planned
CMEK-encrypted audit log sinks	Export logs to <b>CMEK-secured GCS or BigQuery sink</b>	✅ Yes	🟢 Planned
Logging PHI query metadata	Log query metadata (user, timestamp, resource, query ID), but <b>never include raw PHI in logs</b>	✅ Yes	🟢 Planned
PHI tagging via metadata/labels	Tag PHI-related tables with labels: contains_phi=true, retention_days=1800	✅ Yes	🟢 Planned
Data retention & expiration	Set <b>table expiration policies</b> to auto-delete stale PHI Use BQ TTL settings	✅ Yes	🟢 Planned
Restrict EXPORT DATA statements	Monitor and log use of EXPORT DATA or bq extract; apply approval gates if needed	✅ Yes	🟢 Planned
Use of authorized views	Use <b>authorized views</b> to expose only safe columns/rows to downstream users	✅ Yes	🟢 Planned
BQ reservation & workload isolation	Use <b>separate slots/reservations</b> for PHI vs analytics workflows to isolate job runtime environments	⚠️ Recommended	🟢 Planned
Monitoring query behavior	Set up alerts for abnormal query frequency, size, cost, or long-running jobs	⚠️ Recommended	🟢 Planned
Prevent ad hoc PHI exports	Use IAM Conditions or tagging + alerting to block or flag PHI tables from ad hoc querying/export	✅ Yes	🟢 Planned
Audit all data sharing & transfers	Monitor and log all <b>cross-project or cross-org dataset shares</b> , especially if involving PHI	✅ Yes	🟢 Planned
Limit access to storage integrations	Restrict who can create <b>external tables</b> to GCS or Cloud SQL	✅ Yes	🟢 Planned
Use labels for billing + governance	Apply labels like env=prod, compliance=hipaa, team=ops to datasets and jobs for cost tracking + access control	⚠️ Recommended	🟢 Planned

GCP BigQuery

## HIPAA Compliance Checklist

Requirement	Explanation & Detail / Sub-checklist	Required for HIPAA?	Status
Self-hosted on GCP	Use GCE or GKE — <b>do not use n8n Cloud</b> (no BAA)	✓ Yes	● Planned
HIPAA-eligible GCP region	Deploy in asia-south1 or other HIPAA-compliant GCP region	✓ Yes	● Planned
Private IP instance / GKE pod	Deploy with no public IP — only behind internal load balancer or VPN	✓ Yes	● Planned
TLS/HTTPS enabled	Enforce HTTPS via secure ingress (e.g., NGINX or GCP Load Balancer with SSL cert)	✓ Yes	● Planned
Use CMEK-encrypted persistent storage	CMEK-protected persistent disk volumes, logs, uploads, and credentials	✓ Yes	● Planned
<b>MongoDB &amp; BigQuery as backends</b>	Connect only to <b>secure, encrypted</b> MongoDB (CMEK or EBS-encrypted) and BigQuery (with CMEK); no PHI in connection strings or logs	✓ Yes	● Planned
Store secrets in Secret Manager	Store n8n tokens, credentials, encryption keys in <b>GCP Secret Manager</b> , not in .env or Airflow Variables	✓ Yes	● Planned
Webhooks require auth / IP filter	Secure webhooks with auth headers, tokens, or IP allowlists — <b>do not expose PHI on open endpoints</b>	✓ Yes	● Planned
No PHI in logs	Avoid logging patient names, NPI, dates of service, etc. from workflows/webhooks	✓ Yes	● Planned
Application-level PHI audit logs	Track access to PHI by logging workflow name, record ID, timestamp, actor (user/agent), and action taken	✓ Yes	● Planned
Admin UI restricted	Lock n8n UI behind <b>IAP, VPN, or internal-only access</b> ; enforce SSO or 2FA	✓ Yes	● Planned
RBAC for internal users	Use owner, editor, viewer roles for team; don't share admin rights	✓ Yes	● Planned
<b>Air-gapped staging zones for PHI</b>	Use <b>separate staging or processing workflows</b> when PHI is being sent to or received from 3rd-party apps like email	✓ Yes	● Planned
Email tracking with Outlook (PHI-safe)	When reading Outlook inbox to create tickets: <b>redact PHI from logs</b> ; store allocations in secure backend (Mongo/BQ); avoid putting PHI in email subjects	✓ Yes	● Planned
Ticket allocation to agents (PHI-safe)	Allocation workflow reads <b>PHI like patient name, date of service</b> — must have logging, RBAC, encryption, and access review for compliance	✓ Yes	● Planned
Workflow audit tagging	Tag workflows that touch PHI with phi=true for better visibility and compliance review	✓ Yes	● Planned
Centralized workflow logging	Stream workflow runs + errors to <b>Cloud Logging or BigQuery</b> , encrypted with CMEK	✓ Yes	● Planned
Disable unused integrations	Turn off or restrict access to integrations <b>not approved for PHI handling</b>	✓ Yes	● Planned
No hardcoded credentials in workflows	All credentials (API keys, passwords) must be stored in n8n's <b>encrypted credential vault</b>	✓ Yes	● Planned
Secure email sending of allocations	Ensure <b>PHI is only sent in encrypted attachments or redacted summaries</b> if part of outbound emails to agents	✓ Yes	● Planned
External API access through proxy/gateway	Optional: Use GCP API Gateway or NGINX proxy for outbound calls to apply filtering or transformation	⚠ Recommended	● Planned

**n8n**  
(Self-hosted on GCP)

## HIPAA Compliance Checklist

Requirement	Explanation & Detail / Sub-checklist	Required for HIPAA?	Status
HIPAA-eligible region	Create buckets in asia-south1 or other HIPAA-compliant GCP regions	✔ Yes	🟢 Planned
CMEK encryption	Use Customer-Managed Encryption Keys (CMEK) instead of Google-managed keys	✔ Yes	🟢 Planned
Uniform bucket-level access	Enforce uniform ACLs (no object-level IAM); restrict access to service accounts only	✔ Yes	🟢 Planned
Public access prevention	Set publicAccessPrevention = enforced to block any anonymous or public access	✔ Yes	🟢 Planned
Access logs enabled	Enable bucket access logging to another CMEK-protected bucket to track who accessed or modified PHI	✔ Yes	🟢 Planned
Versioning enabled	Turn on object versioning to prevent silent overwrite or loss of PHI	✔ Yes	🟢 Planned
Lifecycle rule – 7-year retention (7 years)	Create a GCS lifecycle policy to automatically delete PHI after 2555 days	✔ Yes	🟢 Planned
Audit log export (via sinks)	Create log sinks to store DATA_ACCESS and ADMIN_ACTIVITY logs to CMEK-encrypted buckets or BigQuery	✔ Yes	🟢 Planned
Bucket labels for PHI tracking	Tag PHI buckets with labels like contains_phi=true, retention_policy=7yrs, env=prod	✔ Yes	🟢 Planned
IAM least privilege	Grant roles like storage.objectViewer, storage.objectCreator only to required service accounts and users	✔ Yes	🟢 Planned
Cross-project access restrictions	Do not share buckets across projects/orgs unless explicitly allowed and audited	✔ Yes	🟢 Planned
Bucket data classification	Maintain a registry or Data Catalog of buckets that store PHI, PII, or other sensitive categories	✔ Yes	🟢 Planned
Integrity validation for uploads	Use hash validation (MD5, CRC32C) on critical uploads (e.g., audit logs, clinical reports) to detect tampering	✔ Yes	🟢 Planned
Access notification alerts	Set up alerts when PHI buckets are accessed, modified, or when new objects are created	⚠ Recommended	🟢 Planned
GCS retention lock (optional)	Use Bucket Lock (WORM — Write Once Read Many) for immutable audit data	⚠ Recommended	🟢 Planned

## Google CloudStorage (GCS)

## HIPAA Compliance Checklist

Service Area	Requirement	Explanation / Sub-checklist	Required for HIPAA?	Status
GKE / GCE (Infrastructure)	Private cluster / Private IPs	Use private clusters or VMs inside VPC without public IPs	✔ Yes	🟢 Planned
	Shielded VMs	Use secure boot, kernel integrity, and hardened VM images	✔ Yes	🟢 Planned
	CMEK-encrypted persistent disks	Attach CMEK-encrypted disks to all stateful workloads (MongoDB, n8n, etc.)	✔ Yes	🟢 Planned
	Kubernetes RBAC	Apply namespace- or workload-specific RBAC policies	✔ Yes	🟢 Planned
	GKE Workload Identity	Use Workload Identity instead of SA key files	✔ Yes	🟢 Planned
IAM + VPC & Network Controls	Least privilege IAM	Assign only the minimum necessary IAM roles per service	✔ Yes	🟢 Planned
	IAM Conditions	Apply time- or location-based constraints to sensitive roles	⚠ Recommended	🟢 Planned
	VPC Service Controls	Apply VPC-SC to protect GCS, BigQuery, Dataflow from data exfiltration	✔ Yes	🟢 Planned
	IAM audit log export	Log all IAM role changes and review quarterly	✔ Yes	🟢 Planned
Secret Manager	Use for all secrets	Store all service tokens, DB creds, API keys in Secret Manager	✔ Yes	🟢 Planned
	Per-secret IAM policies	Use fine-grained IAM to restrict access to each secret	✔ Yes	🟢 Planned
	Secret versioning + access logs	Enable audit logging and version tracking on all secrets	✔ Yes	🟢 Planned
	Secret rotation policy	Implement rotation schedule for long-lived credentials	✔ Yes	🟢 Planned
Cloud Logging & Monitoring	Enable Cloud Audit Logs	Turn on Admin Activity, Data Access, and System logs for all services	✔ Yes	🟢 Planned
	CMEK-encrypted log sinks	Export logs to GCS/BQ with CMEK for long-term retention	✔ Yes	🟢 Planned
	Retain logs for 6+ years	Configure GCS/BQ lifecycle to retain logs as per HIPAA	✔ Yes	🟢 Planned
	Anomaly alerting	Alert on suspicious activity (IAM change, traffic spike, export attempt)	⚠ Recommended	🟢 Planned
GitLab CI/CD (Pipelines)	No secrets in code or YAML	Load all sensitive values via CI environment variables from Secret Manager	✔ Yes	🟢 Planned
	Controlled branch deployment	Deploy from protected branches (main, release) only	✔ Yes	🟢 Planned
	Log redaction / no PHI in logs	Ensure logs during build/deploy pipelines don't expose PHI	✔ Yes	🟢 Planned
	Image signing / scanning	Sign images or scan for vulnerabilities before deploying to GKE	⚠ Recommended	🟢 Planned
Outlook / Email Handling	PHI-safe email ingestion via n8n	When parsing emails into tickets, redact PHI in logs and webhook traces	✔ Yes	🟢 Planned
	Emailing PHI securely to agents	Use redacted emails or secure attachments for allocations sent via Outlook	✔ Yes	🟢 Planned
	TLS-encrypted email APIs	Ensure IMAP/SMTP/Graph API are accessed over TLS	✔ Yes	🟢 Planned
	No PHI in email subjects/logs	Avoid including patient names, dates of service, NPI in subject lines or logs	✔ Yes	🟢 Planned

Other GCP Services

HIPAA Compliance Checklist