Composio BYOC on AWS

Operated by Composio, in your AWS account

Reference architecture

This whitepaper describes Composio BYOC, also called a Composio-managed VPC, in which Composio operates its application services in a customer-owned AWS account. Applications call a local endpoint, provider requests originate from the customer environment, and credentials and operational data use customer-side storage.

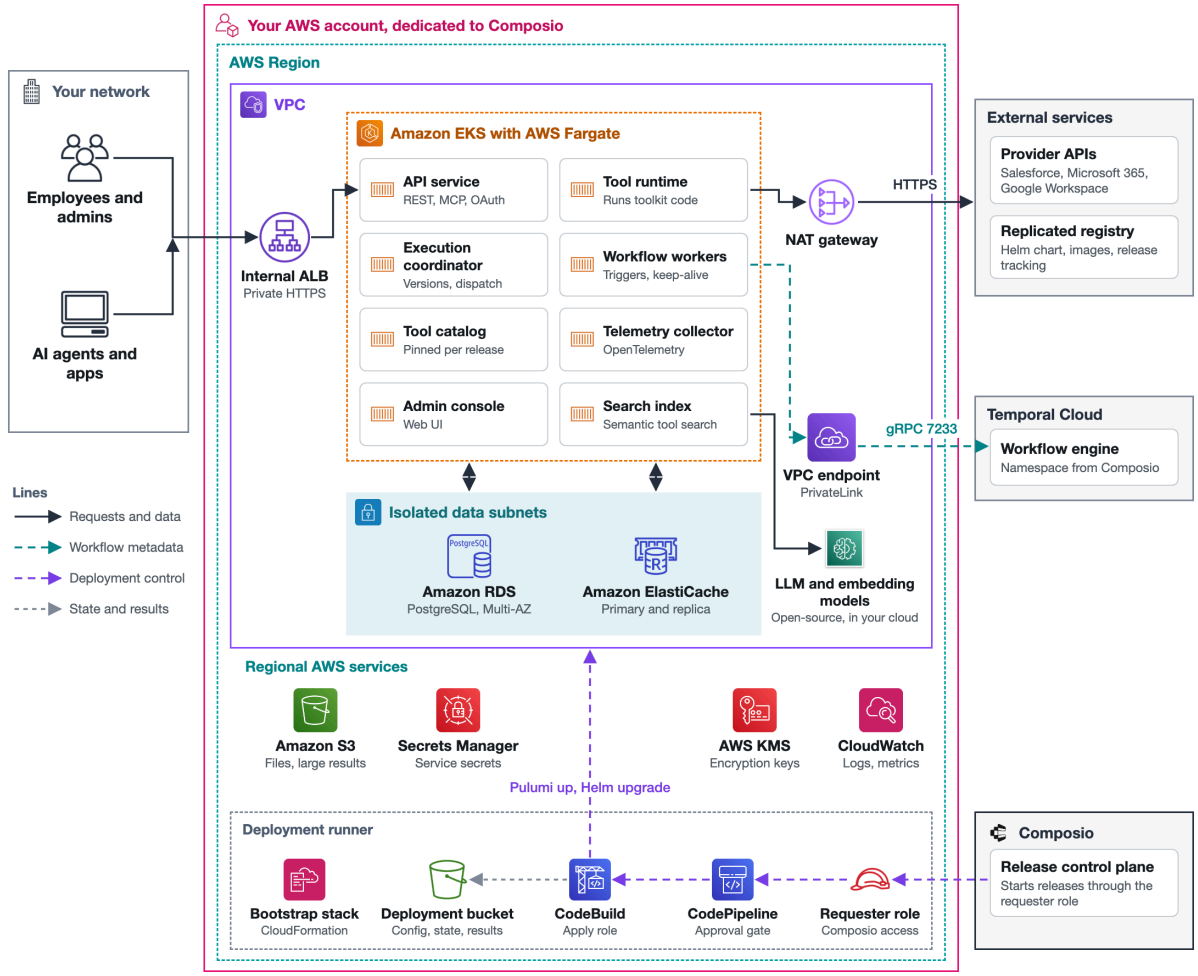

Figure 1. Composio manages the deployment resources in the customer's AWS account through approved IAM roles. Other AWS services are accessed by the deployment but do not run inside its VPC.

The reference compute platform is Amazon EKS with AWS Fargate, so there are no worker nodes to provision, patch, or log in to, and each pod runs in its own isolated compute boundary. Composio coordinates releases outside the account, and Temporal Cloud orchestrates background work such as Triggers. This reference targets commercial AWS regions. Composio and the customer define which external services the deployment can contact and which features it supports. They also specify where credentials are stored, who can access them, and Composio's permissions to operate and support the deployment.

AWS services and network boundaries

In the reference architecture, application containers can be updated or replaced independently of the databases, caches, and object storage that hold deployment state. Figure 2 shows where these components sit in the network. Resource sizes, Availability Zone coverage, inbound access, and data retention are configured to meet the customer's workload, availability, and security requirements.

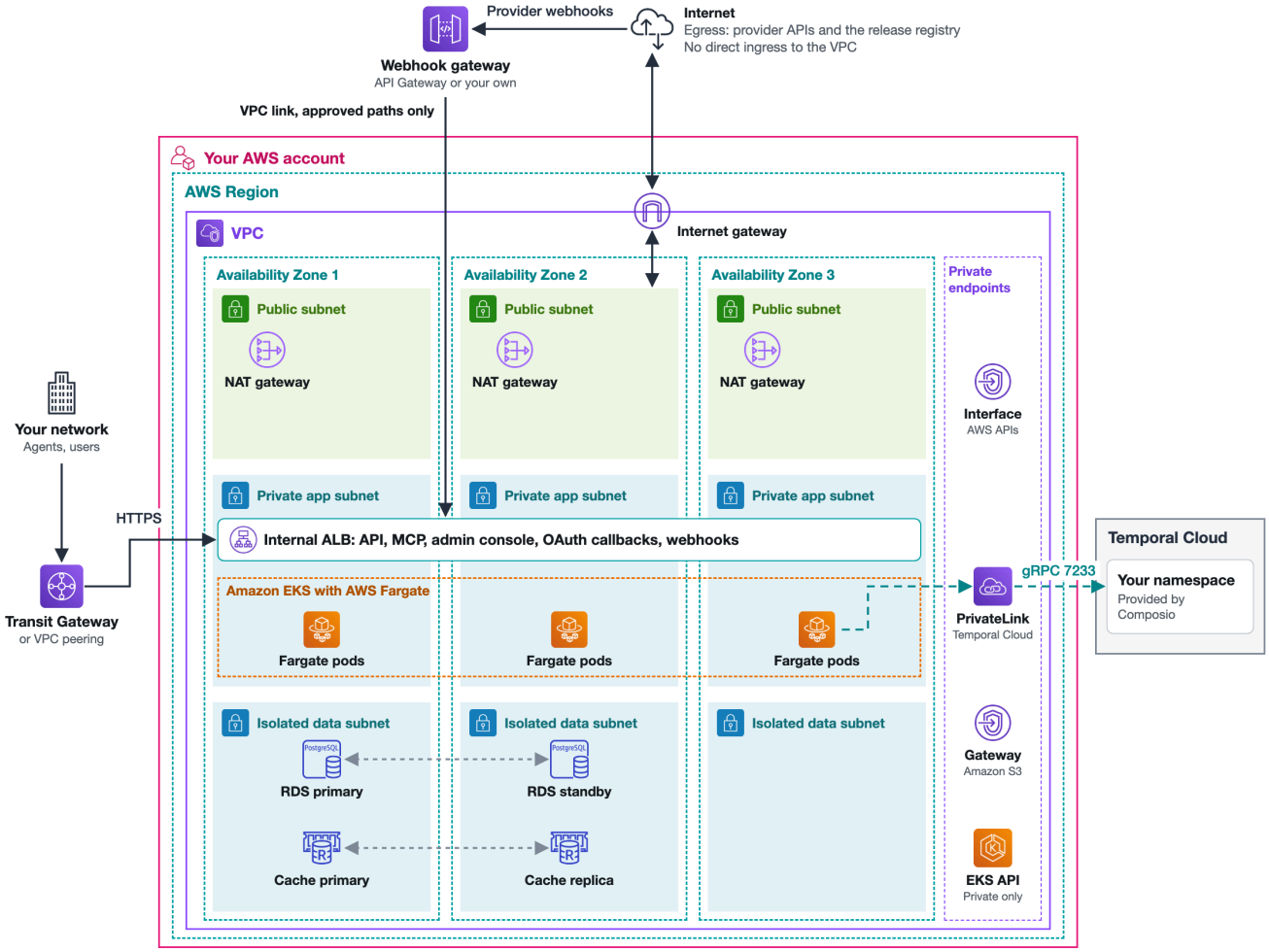

Figure 2. Reference network. Fargate pods run in private subnets and data stores in isolated subnets across Availability Zones. Applications reach the platform through an internal load balancer, and provider webhooks enter only through the customer's gateway.

Infrastructure	What it provides
Amazon EKS with AWS Fargate	Run API/authentication, coordinator, runtime, workers, and the local catalog as Fargate pods in private subnets. Connectors execute through local HTTP.
Amazon RDS PostgreSQL	Store projects, connections, encrypted provider credentials, logs, and durable worker state. Logical databases can share an RDS instance, which runs Multi-AZ.
Amazon ElastiCache Redis	Coordinate caches, sessions, and OAuth flows with a primary and a replica. Provider tokens are never stored in the cache in plaintext.

Infrastructure	What it provides
Amazon S3	Store files, large results, polling artifacts, and worklists. Separate application, pipeline, and infrastructure-state access and retention.
Internal Application Load Balancer	Provide private HTTPS ingress for applications, agents, administrators, and OAuth callbacks. TLS terminates at the load balancer, which targets Fargate pods by IP address. Application APIs are never exposed to the internet.
Webhook gateway	If webhook Triggers are enabled, a public HTTPS endpoint that the customer controls, such as Amazon API Gateway, routes only approved webhook paths to the internal load balancer through a VPC link.
IAM, STS, KMS, and Secrets Manager	Separate runtime and deployment permissions. Store registry credentials in Secrets Manager. Provider tokens remain encrypted application records.
CloudFormation, CodePipeline, and CodeBuild	The managed deployment workflow includes initial setup, change previews, customer approval, deployment, and health checks. Each environment, such as staging and production, has its own pipeline.
Monitoring and audit services	Send logs and OpenTelemetry to an approved sink. Configure required CloudTrail event categories and VPC Flow Logs. Diagnostic exports leave the account only with the customer's approval.
Search index	Finds the right tools for an agent's task using a vector store on EKS. It needs an LLM and an embedding model, which can be open-source models deployed in the customer's cloud.

Private routes, security groups, and network ACLs limit service access, and the data stores sit in isolated subnets with no route to the internet. Services reach regional AWS APIs through VPC endpoints. The deployment runner works inside the VPC and manages the cluster through the private EKS API endpoint, so the Kubernetes API is never exposed to the internet. Outbound, the deployment needs only provider APIs over HTTPS on port 443, Temporal Cloud over gRPC with TLS on port 7233, and the Composio registry and license endpoints.

Tool execution and provider authentication

The application configures the SDK or MCP client to use the customer endpoint. Its backend authenticates the user and creates a session scoped to that user's identifier, allowed tools, and connected accounts. Tool calls run through the local API, coordinator, and execution runtime, and connection setup, revocation, and request-time token refresh run on the same local services.

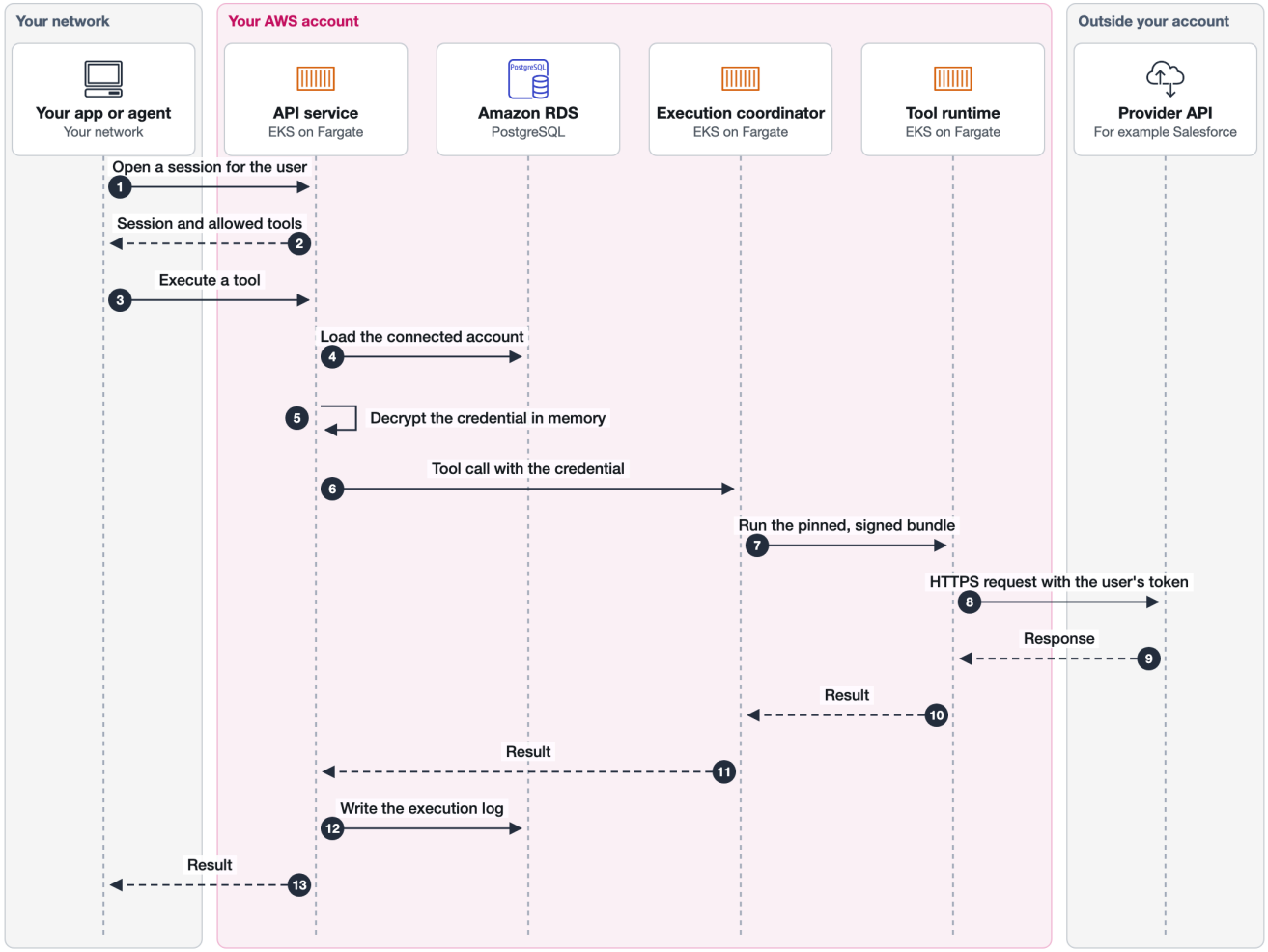

Figure 3. Credentials are resolved inside the customer deployment, and the provider response returns along the same local service path.

Provider authentication. Customers create their own OAuth applications and register callback URLs. The provider redirects the user's browser to the local callback. A private callback works when that browser can resolve and reach it through the customer network or VPN, subject to provider rules. The local service exchanges the code and stores encrypted tokens. Administrator consent still applies.

Application isolation. The backend derives project and user context from the signed-in user. Project credentials stay on trusted servers. Connection lookup and execution enforce project scope, user ownership, and any explicit sharing policy. Separate Composio projects can isolate business units or downstream customers within the dedicated installation.

Runtime secrets and logs. Composio manages the stack credentials and application keys, and the customer manages the OAuth client credentials and certificates it issues. Both are rotated on a schedule set with the customer. Runtime and support permissions determine who can access plaintext. Each project can suppress tool request and response bodies in logs while retaining operational metadata.

Telemetry. Logs, metrics, and traces go only to the sink the customer chooses. Replicated, which Composio uses to distribute releases, reports license and instance status to Composio, such as the installed version and component health. The customer can disable this reporting. The vector store's own usage reporting to its vendor is turned off.

Catalog and search. The release includes a local tool catalog and signed toolkit bundles, and the runtime verifies each bundle's signature before loading its code. Catalog updates arrive through releases. Tool search uses an LLM and an embedding model, and both can be open-source models deployed in the customer's cloud. The customer's agent or model receives any tool results the application supplies.

Custom connectors. Composio shares the code of every custom connector it builds for a customer before deployment and with each new version.

Background work for Triggers

Certain features, such as Triggers, require long-running workflows. These include checking a mailbox every few minutes, handling a webhook, keeping credentials alive for providers whose tokens expire while idle, and revoking connections in bulk. All of them run as Temporal workflows. Temporal Cloud remembers where each workflow is and when it should run next. The code runs on workers in your cluster, and the workers always call out to Temporal, so Temporal never needs a path into your VPC.

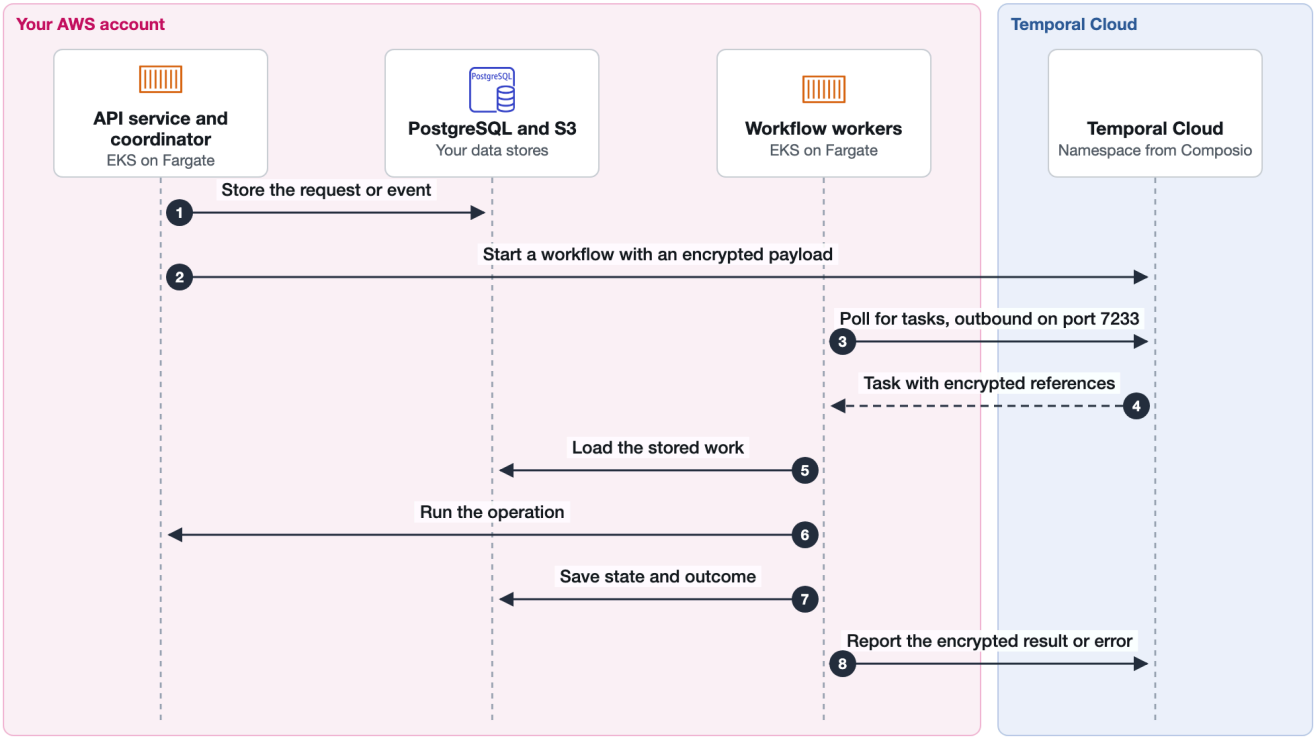

Figure 4. Workers load business records from customer storage and report orchestration results to Temporal. Returned tasks are replies to outbound polling.

Background work	What runs in your account	What Temporal tracks
Polling triggers	Workers load the trigger's state, call the provider through the tool runtime, save the new cursor and items, and deliver events.	Due times, trigger IDs, retries, and outcomes

Background work	What runs in your account	What Temporal tracks
Webhook triggers	The API service verifies each event, and the coordinator stores it encrypted. Workers transform it and deliver it to your endpoint.	Processing tasks, counts, and failures
Credential keep-alive	Workers load the scheduled refresh, and the API service fetches new tokens and saves them.	Callback IDs, due times, retries, and completion
Bulk revocation	Workers load the list of connections, and the API service revokes access at each provider.	Job progress, results, and failures

Temporal receives the information needed to schedule and track background work. In the illustrated flows, credentials, webhook bodies, polling data, and refresh callback bodies remain on local service and storage paths. Workflow inputs, activity results, error messages, and stack traces are encrypted before leaving the customer environment, using keys retained there. Temporal cannot decrypt these payloads. Searchable metadata remains readable by Temporal and is limited to non-sensitive identifiers and operational values. For a customer security review, Composio provides the data flows, the metadata Temporal can read, encryption and key custody details, namespace access controls, and history retention settings.

Deployment operations and recovery

The CloudFormation bootstrap creates the pipeline, IAM roles, versioned S3 configuration and state, and registry credential secret. Composio registers the outputs and requests a pinned release. The release identifies the chart, service and catalog images, migration images, and infrastructure revision. Apply uses the artifacts and configuration versions prepared for the approved preview. CodeBuild runs independently of EKS.

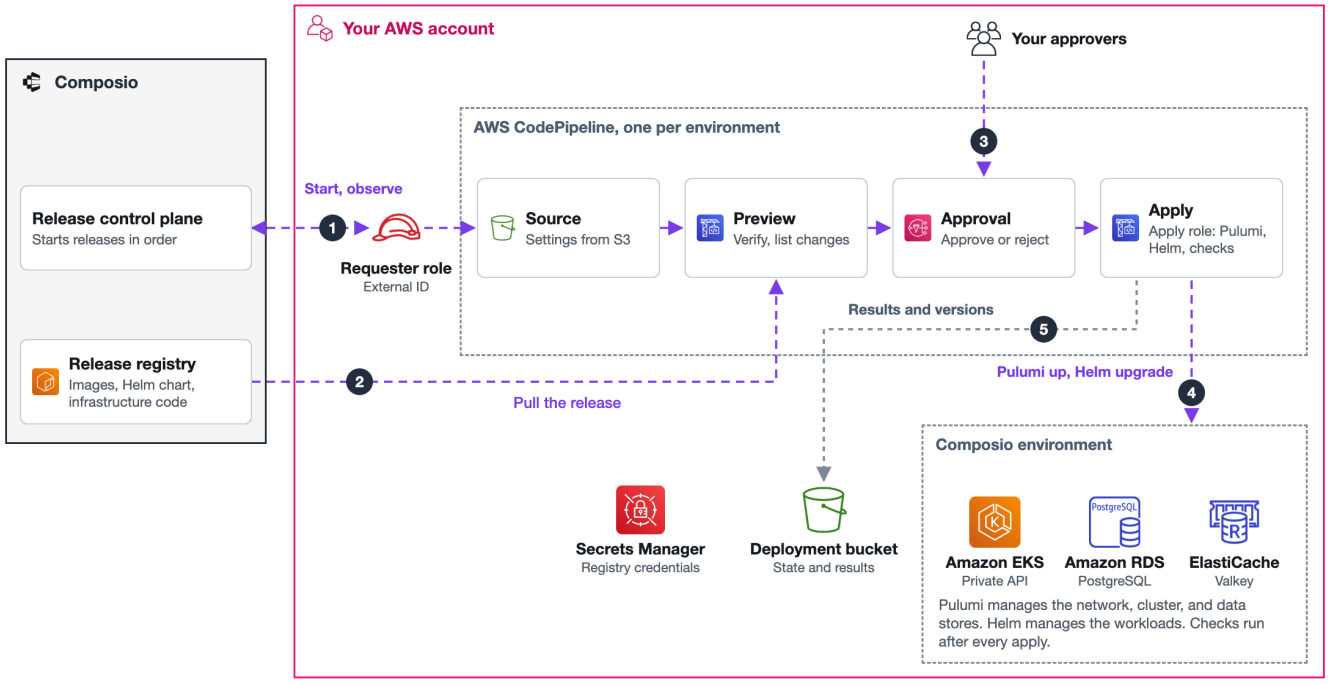

Figure 5. Preview and customer approval precede infrastructure changes, migrations, and Helm rollout. Verification determines whether a release is marked installed.

Access and release control. The requester role starts and observes the named pipeline and maintains its scoped registry credential. A separate apply role changes approved resources. STS supplies temporary credentials under a trust policy naming Composio and the installation's external ID. Application releases cannot widen bootstrap permissions or remove the approval gate. Those controls require a separate reviewed update. Support access and diagnostic exports remain separately approved. The customer controls account policies, network, and approvals.

Release commitments. If the customer declines a release, the pipeline stops and the installed release keeps running. Before retiring a release, Composio gives written notice and makes the replacement available for testing. Planned feature removals and breaking API changes come with advance notice and migration steps. Critical security fixes use an expedited approval, and Composio never upgrades, shuts down, or uninstalls a deployment without the customer's approval.

Failure or change	Expected behavior and recovery boundary
Preview fails or approval is rejected	The pipeline stops before apply. Existing application resources remain on the installed release.
Apply, migration, or rollout fails	Stop the affected release, retain diagnostics, and investigate. A compatible application rollback is a new deployment. Database migrations and infrastructure changes need separate compatibility review.
Worker or runtime stops	Replacement pods start on Fargate from approved images. In-flight requests may fail and need a safe retry. A provider call can repeat after an unclear failure, so agent workflows should tolerate a repeated action. Temporal workflows resume from recorded progress.
Temporal is unreachable	Background work is delayed. Synchronous tool calls can continue while local dependencies and required credentials remain available. Scheduled refresh delays can eventually affect those calls.
Database or Availability Zone is lost	Recovery depends on the selected replication, backups, retention, and tested restore procedures. Composio and the customer set recovery objectives before production.

Deployment prerequisites and feature scope

AWS environment. The customer provides a dedicated AWS sub-account in the chosen region. Composio works with the customer to configure network connectivity, DNS, certificates, and required service quotas. The customer confirms that account policies allow the required AWS services and deployment operations. The customer also approves deployment permissions, names release approvers, and authorizes support access. Composio configures redundancy across Availability Zones, maintenance safeguards, database and cache failover, backups, and restore testing to meet the customer's availability and recovery requirements.

Provider access. The customer provides access to private systems and creates its own OAuth applications. The customer registers the deployment's callback URLs and required scopes with each provider and obtains administrator consent where required. Users' browsers must be able to reach the OAuth callback URLs. If webhooks are enabled, the provider's servers must be able to reach the customer's webhook gateway, independently of browser access.

Models. Tool search needs an LLM and an embedding model. Both can be open-source models that the customer deploys in its own cloud. Running them there keeps search queries inside the customer's environment.

Temporal Cloud. Composio provides background orchestration through its existing Temporal Cloud account. The customer does not need a separate Temporal account or contract. Composio manages the deployment's dedicated namespace, region, history retention, authentication, billing, and support. Workers in the customer environment connect outbound over TLS using namespace-restricted credentials. Composio coordinates network configuration with the customer, including AWS PrivateLink where required. Workflow payloads and failure details are encrypted using keys retained in the customer environment. Readable metadata is limited to non-sensitive operational values.

Capability	Availability
API, SDK, and core execution	Included. The base request path, connections, revocation, storage, and local logs run without Temporal.
Sessions, MCP, and semantic search	Included. Semantic search needs an LLM and an embedding model, which can be open-source models deployed in the customer's cloud.
Triggers and background workflows	Included with Temporal Cloud, covering polling and webhook Triggers, credential keep-alive, and bulk revocation.
Enterprise MCP Gateway and SSO	Deployed separately. Core session MCP transport does not include the gateway's identity layer, SSO, or SCIM.
Sandbox and custom integration tooling	Workbench execution and self-service custom toolkit or OpenAPI tooling are not included. Composio can build custom connectors under the customer's agreement.

Further reading

- Composio token custody. https://docs.composio.dev/docs/security/token-custody
- Composio session configuration. https://docs.composio.dev/docs/configuring-sessions
- AWS Fargate on Amazon EKS. https://docs.aws.amazon.com/eks/latest/userguide/fargate.html
- Temporal payload encryption. https://docs.temporal.io/production-deployment/data-encryption