

**Roadmap: Expanding HRS Cloud to Remediate Azure & GCP**

The goal is to keep AWS-hosted HRS Cloud as the single automation hub while **incrementally** adding the ability to detect and repair incidents in Microsoft Azure and Google Cloud Platform—without disrupting ongoing AWS or on-prem operations.

**1. Adopt a Cloud-Agnostic Integration Layer**

1. **Event Ingestion**
   * Deploy lightweight, stateless “collector” services inside Azure and GCP that forward platform alerts (Azure Monitor, Google Cloud Monitoring) to HRS Cloud via Amazon EventBridge’s **API destinations** or an **HTTPS ingress** exposed through Amazon API Gateway.
   * Normalize all incoming events to an internal JSON schema (e.g., CloudEvents 1.0) so rule-matching remains engine-agnostic.
2. **Action Gateway**
   * Instead of hard-coding AWS SDK calls in workflows, introduce an **Action Service** with a REST/JSON contract:

POST /actions/{provider}/{service}/{operation}  
{  
 "parameters": { … },  
 "target": { "accountId": “…”, "region": “…” }  
}

* + Back the service with provider-specific modules that call:
    - AWS SDK
    - Azure SDK / Azure CLI
    - Google Cloud Client Libraries

1. This keeps workflow definitions portable; adding a new cloud means adding a new backend adapter, not rewriting dozens of playbooks.

**2. Extend Workflow Connectors Incrementally**

|  |  |  |  |
| --- | --- | --- | --- |
| Priority | Target Cloud | First-Wave Connectors | Common BCP / Remediation Use Cases |
| 1 | **Azure** | VM restart, Scale-set resize, Storage snapshot, SQL DB failover, Key Vault secret rotation | App VM reboot, autoscaling mis-match, regional storage migration |
| 2 | **GCP** | Compute Engine instance reset, Instance group resize, Cloud SQL failover, GCS bucket ACL fix | Node crash recovery, capacity burst, DB primary promotion |
| 3 | Both | IAM policy patch, DNS failover (Azure DNS / Cloud DNS) | Broken access rights, cross-cloud DR cut-over |

Build each connector as a re-usable Workflow Task (Camunda 8) or Service Integration (Step Functions) and publish it to an internal registry so new playbooks simply reference the task.

**3. Shared Secrets & Identity**

* Store per-provider credentials in **AWS Secrets Manager**; retrieve at runtime via the Action Gateway using short-lived STS assume-role or workload-identity federation to avoid long-lived keys.
* Enforce least privilege with:  
  – AWS IAM roles for the gateway.  
  – Azure AD app registrations scoped to required resources.  
  – GCP service accounts with minimal IAM roles.

**4. Cross-Cloud Network Path**

* Use **AWS Direct Connect + Transit Gateway** (for Azure ExpressRoute, GCP Partner Interconnect) or fully encrypted site-to-site VPNs.
* Allow outbound “action” calls only; block unsolicited inbound traffic to HRS Cloud.

**5. Incremental Roll-Out Strategy**

1. **Pilot**: Integrate a single Azure subscription; automate two low-risk playbooks (e.g., VM reboot, storage snapshot).
2. **Measure**: Track MTTR, success rate, and cross-cloud latency; tune retries and timeouts.
3. **Scale**: Add remaining Azure resource types and multiple subscriptions / management groups.
4. **Repeat for GCP**, using lessons learned to shorten onboarding time.
5. **Unify Dashboards**: Consolidate execution logs from all providers into Amazon OpenSearch or an observability tool of choice for one-pane incident tracking.

**6. Governance & Compliance**

* Tag every workflow instance with provider, account, region, criticality for audit queries.
* Log all cross-cloud API calls centrally via AWS CloudTrail Data Events (gateway side) and provider-native logs (Azure Activity, Google Cloud Audit) shipped back to HRS for correlation.

**7. Talent & Process**

* Train SREs on cloud-agnostic incident patterns; maintain a **playbook matrix** (Provider × Service × Failure Mode × Remediation).
* Establish a **Connector Certification Process** so community-contributed tasks meet security and testing standards before production use.

By abstracting provider details behind a unified Action Gateway and rolling out connectors in prioritized waves, HRS Cloud gains multi-cloud remediation capability **without re-architecting existing AWS workflows** or compromising security.