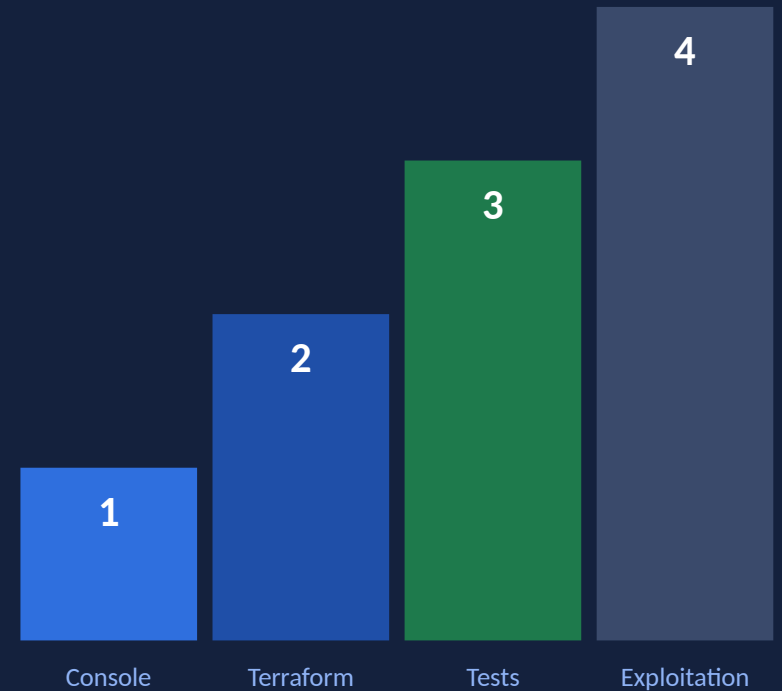


POUR L'ÉQUIPE PROJET ACME · DEV · DEVOPS · PM · SÉCURITÉ · DPO

Note de sécurité Acme : de la note à la preuve

Gemini Enterprise. Les 10 mesures, activées d'abord en natif dans Google Cloud, puis figées en Terraform, testées et suivies chaque jour.



En une page

Le socle (projet 06) couvrait le cœur de la note. Le projet 08 comble les écarts et rend chaque mesure prouvable.

1 Console Activer le natif Google, en observation quand c'est possible Protection immédiate, on mesure avant de bloquer Semaine 1	2 Terraform Reprendre chaque réglage en code relu, paliers A → D Traçable, reproductible par domaine, sans dérive Semaines 2-3	3 Tests 22 tests, chacun avec son journal de preuve Une mesure non prouvée n'existe pas Semaines 3-4	4 Exploitation Journal quotidien, revues, réponses aux alertes Détecter, réagir, prouver à l'audit En continu
--	---	---	--

Ajouts du projet 08 : agents protégés, Shadow AI, SIEM, Access Approval, analyse avant indexation, contrôle de sortie, VPC-SC élargi, Agent Registry, journal quotidien, contrôle de conformité.

Qui fait quoi

Rôle	Ce qu'il fait	Ce qu'il ne peut pas faire
Chef de projet / PM	Planifie les phases, obtient les décisions, suit le PV de recette et les indicateurs	—
Sécurité (RSSI, SOC)	Possède Model Armor, clés, journaux ; valide chaque PR garde-fou ; traite les alertes	Déployer seule : tout passe par la CI
DevOps / plateforme	Phase 1 en console, phase 2 en Terraform via la CI, import de la console	Modifier un garde-fou sans la sécurité (CODEOWNERS)
Développeurs d'agents	Agent Runtime + Model Armor + confirmation humaine ; Agent Registry	Exposer un service à Internet, déployer hors CI
Admins Gemini Enterprise	App, connecteurs, ouverture aux groupes	Toucher aux filtres, aux clés, lire les journaux d'usage
DPO	Journalisation du contenu, durées de conservation	—

Règle d'or : après la phase 2, plus aucune modification de sécurité en console. Toute exception déclenche une alerte et doit correspondre à une PR.

Les 10 mesures et notre réponse

#	Mesure de la note	Natif (phase 1)	Terraform / script (phase 2)	Statut
1	Agents sur Agent Runtime	Plancher Model Armor sur Agent Platform	90_agents_registry.tf, 30_ai_guardrails.tf	Couvert
2	Shadow AI	Politiques d'org, Asset Inventory, SCC	75_shadow_ai.tf · inventaire	Couvert (Wiz : hors app)
3	Model Armor	Templates, réglage de l'app, plancher	30_ai_guardrails.tf (DLP France)	Plus strict
4	Journaux, AxT, Access Approval, SIEM	Audit Logs, AxT, Access Approval	70, 80, 85_siem_access.tf	Couvert
5	Filtrage de sortie	Agent Gateway, IP fixes	60_advanced.tf, 95_secure_web_proxy.tf	Couvert (correction)
6	RAG : droits, documents	Connecteurs fédérés	scan_before_ingest.py	Couvert
7	VPC Service Controls	Dry-run, niveau d'accès	60_advanced.tf (+ aiplatform)	Dry-run → appliqué
8	Identité, domaines, rôles	IdP, groupes	10_identity_access.tf, tfvars par domaine	Décision IdP
9	Validation humaine	Confirmations GE	require_confirmation, destructiveHint	Couvert
10	Agent Registry, partage	Registre, partage désactivé	90_agents_registry.tf, 50_app.tf	Couvert

Pourquoi d'abord la console, puis Terraform ?



La protection n'attend pas le code

Journaux d'audit et Model Armor : dix minutes en console.
Attendre deux semaines de code, c'est deux semaines sans protection.



Observer avant de bloquer

Plancher des agents en inspection, VPC-SC en dry-run : on voit ce qui serait bloqué, on règle, sans casser un service.



Le code ensuite, pour durer

Relu (PR), traçable (Git), reproductible pour chaque domaine, vérifiable : le plan montre toute dérive.



Rien n'est perdu entre les deux

Les ressources créées en console sont importées : Terraform les adopte sans les recréer.

Le calendrier

Semaine	Phase	Livrable	Critère de sortie
S1	1 • Console	Journal de bord des activations	Audit actif ; Model Armor sur l'app ; plancher agents en inspection
S2	2 • Terraform A	PR socle + palier A, import de la console	Plan sans écart ; verify_controls.sh au vert
S3	2 • Terraform B, C	SIEM ; politiques d'org ; VPC-SC dry-run	Flux reçu par le SOC ; aucune violation VPC-SC légitime
S3-S4	3 • Tests	PV de recette des 22 tests	Tests bloquants OK ; risques résiduels signés
S5	Bascule	Plancher agents en blocage ; VPC-SC appliqué	Deux semaines sans faux positif
Ensuite	4 • Exploitation	Rapports quotidiens et mensuels	Indicateurs au comité

Activer le natif dans la console (1/2)

Qui DevOps + sécurité

Durée semaine 1

Guide docs/01-phase1-console.md

#	Action	Rechercher dans la console	Réglage clé	§
1	Journaux d'audit	Audit Logs	Data Read + Write : Discovery Engine, Agent Platform, Model Armor, Secret Manager, Agent Registry	4
2	Templates Model Armor	Model Armor	Région eu ; injection, URL, IA responsable ; données sensibles avancées (DLP France) ; Inspect & Block	3
3	Model Armor dans l'app	Gemini Enterprise	App → Configurations : deux templates ; mode fermé ; rôle Model Armor User	3
4	Plancher + Agent Platform	Model Armor	Paramètres de plancher → Services → Agent Platform : Inspecter uniquement ; Cloud Logging	1
5	Agent Registry, partage	Agent Registry	Agents et MCP enregistrés ; partage d'agents désactivé ; skills sur approbation	10

Les journaux d'audit en premier : par défaut, qui interroge quoi n'est pas journalisé. Sans eux, aucune enquête n'est possible.

Activer le natif dans la console (2/2)

Qui DevOps + sécurité + équipe Cloud centrale Durée semaine 1 Guide docs/01-phase1-console.md

#	Action	Rechercher dans la console	Réglage clé	§
6	Inventaire Shadow AI	Asset Inventory	Ou scripts/shadow_ai_inventory.sh sur l'organisation	2
7	Politiques d'organisation	Règles d'administration	run.allowedIngress, iam.allowedPolicyMemberDomains, gcp.resourceLocations	2
8	Accès de Google	Access Transparency / Approval	AxT sur l'organisation (support Enhanced/Premium) ; Access Approval : tous les services	4
9	Journaux et SIEM	Routeur de journaux	Buckets eu ge-security et ge-usage ; sink Pub/Sub (peut être laissé à Terraform)	4
10	VPC Service Controls	VPC Service Controls	Périmètre dry-run, avec aiplatform ; niveau d'accès IP Acme	7
11	AI Protection	Security Command Center	Si SCC Enterprise/Premium ; sinon Wiz si c'est le standard	2

Chaque activation est notée dans un journal de bord : il sert à l'import Terraform et de preuve pour l'audit.

Les deux réglages qui font la différence

Plancher Model Armor sur Agent Platform

Dans Gemini Enterprise, Model Armor ne filtre pas les agents personnalisés. Le plancher, avec Agent Platform, filtre chaque appel Gemini du projet : agents, notebooks, scripts. Même si un développeur oublie le plugin.

2 semaines Inspecter uniquement → journaux

Puis Inspecter et bloquer (une ligne Terraform)

```
armor_agent_platform_mode = "inspect"  
# après 2 semaines sans faux positif :  
armor_agent_platform_mode = "block"
```

VPC Service Controls en dry-run

Anti-exfiltration : même avec des identifiants volés, rien ne sort du projet.

Piège 1 l'interface GE passe par discoveryengine : sans niveau d'accès, on bloque les utilisateurs.

Piège 2 créer les connecteurs avant d'appliquer.

Ajout aiplatform et agentregistry dans le périmètre, comme l'exige la note.

```
protoPayload.metadata.@type=  
"...VpcServiceControlAuditMetadata"  
protoPayload.metadata.dryRun=true
```

Figer en Terraform : quatre paliers

Qui DevOps (exécution), sécurité (revue)

Durée semaines 2-3

Guide docs/phase2-terraform/

Palier	Contenu	Prérequis
A • Projet	Socle 06 ; audit aiplatform + agentregistry ; plancher Agent Platform ; droits Agent Registry ; alerte Shadow AI	Droits sur le projet
B • SOC	Export Pub/Sub vers le SIEM, même filtre que le bucket sécurité	Identité du connecteur SIEM
C • Organisation	Politiques d'org ; VPC-SC + niveau d'accès ; Access Approval	Rôles org ; AxT ; ID client Workspace
D • Réseau	Secure Web Proxy, liste blanche de domaines	VPC, sous-réseaux, liste validée

Reprendre la console sans rien recréer

```
import {
  to = google_model_armor_template.shield["ge-prompt-shield"]      # adresse dans le code
  id = "projects/PROJECT_ID/locations/eu/templates/ge-prompt-shield" # ressource existante
} # terraform plan -> "1 to import" ; tout "~ update" = un écart à trancher en revue
```

Ce qui change par rapport au socle 06

Fichier	Ajout	§
30_ai_guardrails.tf	Plancher étendu : integrated_services = AI_PLATFORM, inspect ou block	1, 3
60_advanced.tf	aiplatform et agentregistry dans le périmètre ; niveau d'accès IP Acme	7
70_detection.tf	Journaux d'accès aux données pour aiplatform et agentregistry	4
75_shadow_ai.tf (nouveau)	3 politiques d'organisation ; alerte service public / agent créé	2
85_siem_access.tf (nouveau)	Pub/Sub, sink, droit du writer ; Access Approval	4
90_agents_registry.tf (nouveau)	Droits du registre ; agent sur Agent Runtime, identité dédiée, Agent Gateway	1, 9, 10
95_secure_web_proxy.tf (nouveau)	Proxy, politique, liste blanche, sous-réseau dédié	5

Circuit : PR → plan publié → validation sécurité obligatoire (CODEOWNERS) → apply par la CI → alerte « garde-fou modifié » → verify_controls.sh. Un KO fait échouer le déploiement.

Agents, Shadow AI, Model Armor

1

Agents sur Agent Runtime

Natif Plancher Model Armor sur Agent Platform.

Code Identité dédiée, ModelArmorPlugin en mode fermé, refus de démarrer sans Model Armor, sortie Agent Gateway.

Pourquoi Héberger ne filtre rien. Plugin + plancher : deux protections indépendantes. Exception encadrée : MCP sur Cloud Run.

2

Shadow AI

Natif Asset Inventory, politiques d'org, SCC AI Protection.

Code Cloud Run non public, partage au domaine, UE ; alerte « service public / agent créé » ; inventaire mensuel.

Pourquoi Empêcher coûte moins cher que détecter ; l'inventaire prouve l'état. Quarantaine réseau et Wiz : organisation.

3

Model Armor

Natif Templates, réglage de l'app, plancher.

Code Inspect & Block en entrée et en sortie ; DLP France (NIR, CNI, passeport, IBAN) ; mode fermé.

Pourquoi Plus strict que la note. Le filtre « de base » ne voit ni IBAN ni NIR.

Journaux, sortie, documents

4

Journaux, AxT, Access Approval, SIEM

Natif Audit Logs, Access Transparency, Access Approval.

Code Audit étendu à aiplatform et agentregistry ; buckets UE ; Pub/Sub vers le SIEM ; Access Approval sur tous les services.

Pourquoi Access Approval couvre Gemini Enterprise, Agent Runtime, Model Armor et KMS : pas d'accès Google sans accord.

5

Filtrage de sortie

Natif Agent Gateway, IP fixes des connecteurs.

Code Agent Gateway ; agent relié à la passerelle ; Secure Web Proxy optionnel, liste blanche.

Pourquoi GE est un SaaS : il ne sort pas par notre VPC. Agent Gateway pour GE et les agents, SWP pour nos charges.

6

RAG : droits et documents

Natif Connecteurs fédérés : droits de la source à chaque requête.

Code scan_before_ingest.py : Model Armor analyse PDF, Office, CSV ; quarantaine ; fail closed.

Pourquoi Model Armor ne voit pas les documents récupérés : on les analyse avant de les publier.

Périmètre, identité, humain, registre

7

VPC Service Controls

Natif Dry-run, niveau d'accès.

Code aiplatform, agentregistry ajoutés ; IP Acme.

Pourquoi Le dry-run évite la coupure.

8

Identité et domaines

Natif IdP de l'app, groupes.

Code Rôles séparés ; un tfvars et un état par domaine.

Pourquoi Un domaine compromis n'expose pas les autres.

9

Validation humaine

Natif Confirmations GE, outils destructifs.

Code require_confirmation ; destructiveHint.

Pourquoi Dernier rempart contre l'injection indirecte.

10

Agent Registry

Natif Registre ; partage désactivé.

Code Lecture utilisateurs, publication développeurs.

Pourquoi Hors registre = non gouverné, non joignable.

Le modèle d'agent sécurisé

```
# §9 : toute écriture passe par un humain
tools=[get_order,
        FunctionTool(refund_order,
                      require_confirmation=needs_human)]

# §3 : Model Armor, mode fermé
ModelArmorPlugin(config=ModelArmorConfig(
    prompt_template_name=os.environ["PROMPT_TEMPLATE"],
    response_template_name=os.environ["RESPONSE_TEMPLATE"],
    block_on_screening_failure=True))

# Jamais d'agent déployé sans Model Armor
raise RuntimeError("PROMPT_TEMPLATE manquant")
```

- ✓ Déployé sur Agent Runtime, par la CI
- ✓ Identité dédiée par agent
- ✓ ModelArmorPlugin, mode fermé
- ✓ Refus de démarrer sans Model Armor
- ✓ require_confirmation sur les écritures
- ✓ MCP : readOnlyHint / destructiveHint
- ✓ Sortie via Agent Gateway

Essai local : LOCAL_DEV=1 adk web . → « Rembourse 10 € sur CMD-1002 » → l'interface demande une confirmation.

Prouver : 22 tests, un journal par test

Qui sécurité (exécution), PM (suivi)

Durée semaines 3-4

Guide docs/03-plan-de-tests.md

#	Test	Attendu	Preuve
T01-T07	Jailbreak, IBAN, carte, NIR, URL ; affaiblir un template	Bloqués ; plancher refuse	Script ; audit
T08-T10	Jailbreak vers l'agent ; appel Gemini direct	Bloqué ; journalisé puis bloqué	SanitizeOperationLogEntry
T11-T12	Remboursement ; outil MCP destructif	Confirmation demandée	Interface, traces
T13-T15	Lien externe inconnu ; fichier de B ; PDF piégé	Non chargé ; invisible ; quarantaine	Navigateur ; CSV
T16-T19	Hors groupe ; admin GE ; allUsers ; agent hors CI	Refusés ou alerte	PERMISSION_DENIED ; e-mail
T20-T22	verify_controls ; PR template ; hors réseau	Tout OK ; alerte + SIEM ; violation	Sortie ; SIEM ; VPC-SC

Passage en production : T01-T12, T14, T16, T17, T20, T21 OK. Les autres OK, ou risque accepté par écrit par le RSSI.

Exploiter : le journal quotidien

Qui sécurité / SOC

Durée 10 min par jour

Guide docs/04-exploitation-journaliere.md

Fréquence	Quoi	Outil
Chaque jour	Changements, blocages, Shadow AI, accès refusés, VPC-SC ; alertes	daily_report.sh, Alerting
Chaque semaine	Faux positifs Model Armor ; nouveaux agents au registre	Log Analytics, Agent Registry
Chaque mois	Conformité et inventaire Shadow AI archivés	verify_controls.sh, shadow_ai_inventory.sh
Chaque trimestre	Groupes, liste blanche, conservation, exceptions	IAM, terraform.tfvars

Journal	Où	Durée
Audit admin et accès aux données, verdicts Model Armor, VPC-SC	Bucket ge-security	365 j
Usage Gemini Enterprise (contient des questions)	ge-usage, sécurité seule	90 j
Copie temps réel pour le SOC	Pub/Sub → SIEM	Selon SOC

Quand une alerte sonne

Alerte	Première question	Si la réponse est non
Garde-fou modifié	Une PR Terraform correspond ?	Incident : terraform apply remet la configuration validée ; retirer les droits ; RSSI
Pic de blocages	Un seul utilisateur, ou beaucoup ?	Un seul : abus (charte). Beaucoup : faux positif, ajuster par PR
Service public / agent créé	PR et inscription au registre ?	Contacter l'auteur ; retirer allUsers, arrêter ou régulariser
Demande Access Approval	Le ticket support est-il connu ?	Refuser ; sinon approuver pour la durée minimale
Fuite suspectée	—	Retirer l'accès ; désactiver la clé KMS si besoin ; exporter les journaux ; DPO (72 h)

Sept points de la note à corriger

#	La note dit	La réalité	Ce que nous faisons
1	Exemple JSON Model Armor	Champs invalides : refusé par l'API	Syntaxe correcte dans le code
2	Config de base = données Acme	Orientée US : ni IBAN, ni NIR, ni CNI	Config avancée, DLP France
3	« Detect and redact »	Dans GE, une détection bloque	« Détecter et bloquer »
4	Filtre d'URL = Echo Leak couvert	Seulement les URL connues	Test T13, sortie, humain
5	Model Armor protège tous les modèles	Pas les agents personnalisés dans GE	Plugin + plancher Agent Platform
6	Le SWP filtre la sortie de l'IA	GE ne sort pas par notre VPC	Agent Gateway ; SWP pour nos charges
7	WIF Entra / Okta	Pas de connecteurs Workspace ; changer efface l'historique	Décision avant la production

Pourquoi cette approche est la bonne

- 1 **Protéger dès la semaine 1** avec le natif Google, sans attendre le code
- 2 **Ne rien casser** ce qui peut bloquer passe d'abord par l'observation
- 3 **Durer** tout finit en code relu, reproductible par domaine
- 4 **Se prouver** 22 tests, une commande de conformité, un journal quotidien
- 5 **Corriger la note** là où elle promettait ce qui n'existe pas

Prochaine étape

Atelier de deux heures (sécurité, plateforme, PM, DPO) pour acter les quatre décisions : IdP, SIEM, Access Transparency, exception MCP. Puis phase 1 dans le projet de recette.