AI Infrastructure Security Playbook

Azure Implementation Guide

Cloud-Specific Controls Mapped to ETSI EN 304 223 Risk Gradient L1–L8

V1.0 — February 2026

Risk-tiered · Capability-based · Exposure-driven

https://deepcyber.ai

Contents

1. Introduction

This guide provides Azure-specific implementation instructions for the AI Infrastructure Security Playbook. It maps each control from the playbook's seven dimensions (Governance, Identity, Data, Processing, Network, Supply Chain, Monitoring and Operations) to concrete Azure services, configurations, and recommended settings. All controls are cumulative: higher tiers inherit and extend lower-tier controls. Controls must be validated via automated policy enforcement where technically feasible.

This guide should be read alongside the main AI Infrastructure Security Playbook, which defines the Risk Gradient (L1–L8), cross-cutting risk modifiers, and the ETSI EN 304 223 alignment rationale. This document focuses on the how rather than the what and why.

This guide assumes an Azure environment with Management Groups, Azure Policy, and Microsoft Entra ID as the identity provider. Controls reference Azure-native services and Microsoft 365 security stack by default.

For Defender for Cloud, enable the Cloud Security Posture Management (CSPM) plan with Defender CSPM for advanced posture management. For Azure ML workloads, enable Azure Policy for Machine Learning and Defender for Cloud ML recommendations.

2. Getting Started

Before working through the detailed control checklists, follow these five steps to establish your baseline and create a prioritised action plan. This sequence moves from discovery through assessment to actionable remediation, ensuring you focus effort where risk is highest.

Step 1: Build Your AI Inventory

You cannot secure what you do not know exists. Begin by discovering and documenting every AI workload, tool, and service in use across your organisation. Shadow AI is often the largest unmanaged risk.

- ☐ Conduct a discovery exercise across all business units to identify every AI tool, service, and automation in use — including embedded features (Copilot, Gemini, Atlassian Intelligence), citizen developer automations, API integrations, and ML workloads
- ☐ Classify each AI workload against the Risk Gradient (L1–L8) using the tier definitions in the main playbook
- ☐ For each workload, assess the five cross-cutting risk modifiers (Exposure, Data Sensitivity, Autonomy, Integration Privilege, Physical-World Effect) to determine the effective risk level
- ☐ Document the data sensitivity classification for each workload: what data does it access, process, store, or generate? Include both designed access and potential access given current permissions
- ☐ Assign an owner to each AI workload who is accountable for its security posture
- ☐ Record all entries in a centralised AI Inventory (spreadsheet, CMDB, or dedicated AI governance tool) that is maintained as a living document

Step 2: Assess Your Current Security Baseline with CSPM

For existing AI workloads, your Cloud Security Posture Management (CSPM) tooling provides an immediate, evidence-based view of your current gaps. Enable Microsoft Defender for Cloud with Defender CSPM plan enabled. Enable Defender for Cloud across all subscriptions hosting AI workloads. Use the Regulatory Compliance dashboard to track posture against built-in standards and create a custom initiative mapping ETSI EN 304 223 provisions. Enable Defender for Containers, Defender for Key Vault, and Defender for Cloud Apps (CASB) for AI-specific coverage and review findings across all environments hosting AI workloads.

- ☐ Review findings and ensure the following baseline controls are in place for every AI workload:
  - (a) Encryption at rest and in transit is enforced on all AI data stores, model artifacts, and communication channels
  - (b) Identity and access policies follow least privilege — no overly permissive roles on AI resources
  - (c) Network exposure is minimised — no unintended public endpoints on AI services
  - (d) Logging and monitoring are enabled for all AI workloads
  - (e) Vulnerability scanning is active on all compute, containers, and dependencies
- ☐ Record your CSPM secure score or compliance percentage as your starting baseline for AI workloads
- ☐ Export the CSPM findings for your AI workloads for use in Step 3

Azure-specific CSPM setup:

- ☐ Enable Defender for Cloud on all subscriptions hosting AI workloads; activate Defender CSPM for advanced posture management
- ☐ Enable Microsoft Sentinel and connect Defender for Cloud as a data source for centralised finding aggregation
- ☐ Enable Purview compliance assessments and create a custom assessment mapping ETSI EN 304 223 provisions
- ☐ Enable Defender for Cloud Apps (CASB) to discover shadow AI SaaS usage across the organisation
- ☐ Review the Defender for Cloud Secure Score: record the baseline score for AI workload subscriptions
- ☐ Export Defender for Cloud recommendations and findings via Azure Resource Graph or Continuous Export to Log Analytics for ingestion by the DeepCyber ETSI Agent in Step 3

Step 3: Map Findings to ETSI EN 304 223 with the DeepCyber ETSI Agent

Raw CSPM findings tell you what is misconfigured, but not how it maps to AI-specific security requirements. The DeepCyber ETSI Agent bridges this gap by ingesting your CSPM findings, mapping

them against ETSI EN 304 223 provisions, and generating a tailored, prioritised remediation checklist aligned to your assessed risk gradient level.

- ☐ Feed your exported CSPM findings into the DeepCyber ETSI Agent along with your AI Inventory and risk gradient assessments from Step 1
- ☐ The agent maps each finding to the relevant ETSI EN 304 223 principle and provision, identifies which risk gradient levels are affected, and prioritises remediation by risk severity and blast radius
- ☐ Review the generated checklist: it will highlight quick wins (controls that close multiple ETSI provisions simultaneously) and critical gaps (controls missing at your highest risk gradient levels)
- ☐ Use the prioritised output to create your remediation backlog, assigning owners and target dates to each action item

Step 4: Remediate by Priority

Work through the prioritised remediation checklist, starting with critical findings at your highest effective risk gradient levels. Focus on controls that deliver the greatest risk reduction per unit of effort.

- ☐ Address critical and high-severity findings first, prioritising workloads at the highest effective risk gradient level
- ☐ Target quick wins that close multiple ETSI provisions simultaneously: enforcing encryption (P6), enabling MFA (P6), activating logging (P12), and applying least-privilege access (P5, P6)
- ☐ Use the detailed control checklists in this guide as the reference for implementing each control
- ☐ Track remediation progress against your CSPM secure score and ETSI provision coverage percentage

Step 5: Establish Continuous Governance

AI security is not a one-time exercise. New AI workloads are deployed continuously, cloud configurations drift, and new threats emerge. Establish recurring processes to maintain your security posture over time.

- ☐ Schedule recurring CSPM reviews (minimum monthly) across all AI workload environments
- ☐ Re-run the DeepCyber ETSI Agent periodically (quarterly recommended) to reassess posture against ETSI provisions as your environment evolves
- ☐ Update the AI Inventory whenever new AI workloads are deployed, existing workloads change scope, or workloads are decommissioned
- ☐ Integrate AI workload onboarding into existing change management processes: require risk gradient assessment and security review before any new AI workload enters production
- ☐ Report AI security posture to leadership quarterly using CSPM scores, ETSI provision coverage, and remediation velocity as key metrics

2A. How to Use This Playbook

This playbook is designed for multiple audiences. Use the guide below to find the fastest path to the content most relevant to your role.

Your Role	What to Do	Start Here
CISO / Security Leader	Review the service mapping table and minimum baseline to understand your cloud posture requirements.	Section 3 (Service Mapping), then Minimum Baseline table in Section 3A
Platform / Cloud Engineer	Use the tier checklists as implementation runlists. Each control names the specific cloud service and configuration.	Section 2 (Getting Started), then Section 4 (Implementation Checklists)
Security Architect	Map the service table to your existing cloud architecture and identify gaps per tier.	Section 3 (Service Mapping), then Section 4 (Checklists)
GRC / Compliance	Use the checklists as evidence of control implementation against ETSI EN 304 223 provisions.	Section 2 (Getting Started, Step 3: DeepCyber ETSI Agent), then Section 4 (Checklists)

Tip: All readers should begin with the Getting Started workflow (five steps) to build an AI inventory and establish a baseline before diving into detailed controls.

3. Azure Service Mapping

The following table maps each control dimension to the primary Azure services used to implement the playbook controls.

Control Dimension	Primary Services	Key Configuration
Governance	Azure Policy, Defender for Cloud, Purview Compliance Manager, Management Groups	Deploy Azure Policy initiatives for AI resources; use Defender for Cloud secure score; enable Purview compliance assessments against ETSI mapping
Identity	Entra ID, Managed Identity, Key Vault, PIM, Conditional Access	Enforce Managed Identity for all AI workloads; store secrets in Key Vault with auto-rotation; use PIM for JIT admin access; enforce Conditional Access with MFA
Data	Microsoft Purview (Information Protection, DLP, Data Map), Azure Storage encryption	Enable Purview sensitivity labels; deploy DLP policies on AI endpoints; enforce CMK encryption on all AI data stores
Processing	Azure OpenAI Service, AI Studio, App Service, AKS, API Management, Azure DevOps, GitHub Advanced Security	Deploy AI workloads on AKS/App Service; route all AI API calls through API Management; enforce DevOps pipeline gates
Network	NSGs, Private Link, Azure Firewall, WAF, Front Door, Virtual Networks	Isolate AI workloads in dedicated VNets; use Private Endpoints for Azure OpenAI and ML; deploy WAF on APIM; enable DDoS Protection
Supply Chain	Defender for DevOps, GitHub Dependabot, Azure Container Registry scanning, SBOM	Enable ACR vulnerability scanning; integrate Defender for DevOps in pipelines; enforce image signing with Notation
Monitoring & Ops	Microsoft Sentinel, Defender for Cloud, Azure Monitor, Log Analytics, Defender for Containers	Deploy Sentinel with AI-specific analytics rules; enable Defender for Cloud across all subscriptions; stream diagnostics to Log Analytics

3A. Minimum Baseline Controls by Tier

The following table summarises the minimum required infrastructure controls at each risk gradient level. Use this as a quick-reference before working through the detailed checklists. Controls are cumulative: each tier inherits all controls from lower tiers.

Tier	Pattern	Minimum Required Controls (cumulative)
L1	**Embedded AI**	MFA on all AI-enabled accounts, DLP integration with AI endpoints, AI acceptable use policy, CSPM enabled, prompt/response logging, shadow AI discovery, quarterly access reviews
L2	**AI-Assisted Dev**	+ Mandatory code review for AI-generated code, SAST/DAST in CI/CD, AI code provenance tracking, developer training on AI code risks, dependency scanning on AI-suggested packages
L3	**Citizen Agents**	+ Environment separation (dev/prod), connector permission reviews, DLP on low-code connectors, workflow approval gates, JIT access for agent service accounts, agent inventory register
L4	**API Orchestration**	+ API gateway with rate limiting, credential vault (no hardcoded keys), network segmentation for AI services, input validation on all API endpoints, egress filtering, API key rotation policy
L5	**Autonomous Agents**	+ Tool call allowlisting, agent sandboxing, memory TTL enforcement, human-in-the-loop for high-risk actions, permission boundaries preventing self-escalation, kill switches
L6	**ML Pipelines**	+ Data provenance and lineage tracking, artifact signing and integrity verification, training environment isolation, model registry access controls, pipeline audit logging, SBOM for model dependencies
L7	**Model Hosting**	+ Inference endpoint rate limiting, model extraction detection, adversarial input filtering, private endpoints (no public exposure), model versioning with rollback, A/B deployment gates
L8	**Multi-Agent**	+ mTLS between agents, capability-scoped identity per agent, trust boundary enforcement, circuit breakers, cascade failure detection, real-time agent behaviour monitoring, autonomous privilege escalation prevention

Note: "+" indicates controls added at this tier, in addition to all controls inherited from lower tiers. The detailed checklists in the following section provide the full implementation specification.

Control Dimensions at a Glance

Each dimension applies at every tier. The table shows what changes as you move up the risk gradient.

Dimension	At L1 (Foundational)	At L8 (High Assurance)	ETSI
Governance	AI acceptable use policy, shadow AI audits	AI safety board, autonomous system risk review, kill-switch governance	P1, P3, P4
Identity	MFA, Conditional Access on licences	mTLS, capability-scoped agent identity, JIT with session tokens	P4, P5, P6
Data	DLP on prompts, data classification	Provenance tracking, lineage audit, cross-agent data flow controls	P5, P8, P13
Processing	Tenant config review, feature toggles	Agent sandboxing, capability tokens, tool-call enforcement	P2, P6, P9
Network	Existing segmentation sufficient	Service mesh, private endpoints, per-agent egress rules, circuit breakers	P2, P6
Supply Chain	Vendor DPA review	Model SBOM, artifact signing, training data provenance, dependency pinning	P7
Monitoring	CSPM, basic alerting	Real-time agent behaviour monitoring, cascade detection, anomaly ML	P11, P12

Same seven dimensions at every tier – radically different infrastructure controls. The detailed checklists specify every control.

3C. Common AI Infrastructure Failure Patterns

AI infrastructure failures are rarely model failures – they are identity, network, and supply chain failures amplified by AI capability. The following patterns represent the most common infrastructure-level failures observed in AI deployments. Each maps directly to controls in the tier checklists that follow.

Failure Pattern	What Happens	Controls That Prevent It
Over-permissioned agent with production write access	A citizen-built Power Automate agent granted broad connector permissions writes directly to production HR or finance systems. No approval gate, no audit trail.	L3+: Governance, Identity
LLM API key committed to a public repository	An API key for a paid LLM service is hardcoded in source and pushed to GitHub. Automated scanners find it within minutes. The key has no spending cap or IP restriction.	L4+: Identity, Supply Chain
Fine-tuned model promoted without integrity verification	A model trained on sensitive data is promoted from staging to production with no cryptographic hash check. A tampered artifact enters the serving pipeline undetected.	L6+: Processing, Supply Chain
Inference endpoint exposed without rate limiting	A self-hosted LLM endpoint is deployed on a public subnet with no WAF, no rate limit, and no query pattern monitoring. Systematic extraction queries exfiltrate the model weights.	L7+: Network, Processing, Monitoring
Static service account shared across agent mesh	Multiple agents in a multi-agent system share a single service account with broad IAM permissions. One compromised agent inherits the access of all others.	L8: Identity, Governance
Cross-agent implicit trust enabling lateral compromise	Agents in a multi-agent system accept task delegations from any peer without verifying identity or capability. An injected prompt in one agent propagates through the mesh.	L8: Processing, Network, Identity
Shadow AI tool adopted without security review	A team enables an AI meeting transcription service and grants it access to calendar and email. No DPIA, no DLP, no vendor security review. Sensitive data flows to an unvetted third party.	L1+: Governance, Data
AI-generated code deployed without review	A developer uses AI code completion to generate database queries. The AI hallucinates a dependency and introduces an SQL injection vulnerability. No human review, no SAST scan.	L2+: Processing, Supply Chain

Every pattern above was preventable with controls already in this playbook. The tier checklists that follow provide the specific implementation steps.

4. Implementation Checklists by Risk Gradient Level

Each tier's controls are mapped to specific Azure services with actionable configuration items. Items marked with □ are implementation checkpoints.

4.1 L1: Embedded AI

Example: Microsoft 365 Copilot, Copilot for Security, Dynamics 365 Copilot, Viva AI features

Governance

- □ Use Microsoft 365 Admin Centre to control Copilot licence assignment by Entra ID group
- □ Deploy Azure Policy to restrict AI service creation in non-approved subscriptions
- □ Enable Purview Compliance Manager assessment for ETSI EN 304 223 awareness requirements
- □ Publish AI Acceptable Use Policy and link in Entra ID Terms of Use (enforce acceptance at sign-in)
- □ Use Defender for Cloud Apps (CASB) to discover shadow AI usage across SaaS applications
- □ Maintain Purview Data Map inventory of all data sources accessible to Copilot features

Identity

- □ Enforce Conditional Access policies requiring MFA and compliant device for Copilot access
- □ Apply Entra ID group-based Copilot licence assignment (no self-service enablement)
- □ Restrict Copilot admin settings to Global Admin and dedicated AI Admin roles via RBAC
- □ Review Entra ID access reviews quarterly for all users with AI feature licences
- □ Block Copilot access from non-compliant or unmanaged devices via Conditional Access device filters

Data

- □ Deploy Microsoft Purview DLP policies covering Copilot interactions (M365 DLP for Teams, SharePoint, Exchange)
- □ Apply Purview sensitivity labels to all SharePoint sites and documents accessible to Copilot
- □ Verify Microsoft 365 data residency settings (Multi-Geo if applicable) for Copilot data processing
- □ Enable Purview Audit (Premium) for Copilot interaction logging
- □ Configure Copilot to respect existing information barriers and ethical walls in Purview
- □ Review Microsoft Copilot data processing agreement and verify no training data retention

Processing (Apps, APIs, Integration)

- □ Review Microsoft 365 Admin Centre Copilot settings: disable in workloads where not approved
- □ Verify Copilot respects sensitivity labels on source documents (test with labelled test docs)
- □ Control Copilot plugin availability via Integrated Apps settings in M365 Admin Centre
- □ Disable third-party Copilot plugins unless explicitly approved and vetted

Network

- □ Verify Microsoft 365 Copilot traffic routes via standard M365 endpoints (no additional exposure)
- □ Block access to unapproved third-party AI services using Defender for Cloud Apps session policies
- □ Apply Conditional Access named locations to restrict Copilot access to corporate network or trusted IPs
- □ Review M365 endpoint connectivity requirements to ensure firewall rules permit only required Copilot traffic

Supply Chain

- □ Review Microsoft Trust Centre for Copilot security certifications (SOC 2, ISO 27001)
- □ Monitor Microsoft 365 Message Centre for Copilot security and compliance updates
- □ Verify Microsoft sub-processor list for Copilot data processing
- □ Review Microsoft AI responsible use commitments and data handling documentation

Monitoring and Operations

- ☐ Enable Defender for Cloud across all Azure subscriptions; review AI-related recommendations
- ☐ Deploy Microsoft Sentinel with M365 Defender connector for Copilot telemetry ingestion
- ☐ Create Sentinel analytics rules for unusual Copilot interaction patterns (bulk queries, sensitive data access)
- ☐ Monitor Copilot usage reports in M365 Admin Centre (Copilot Readiness and Usage dashboards)
- ☐ Use Purview Audit log search for Copilot-specific events
- ☐ Enable Defender for Cloud Apps to track Copilot-related cloud app activity
- ☐ Track Defender for Cloud secure score to measure AI security posture over time

4.2 L2: AI-Assisted Development

Example: GitHub Copilot, Copilot in Visual Studio, Azure DevOps with AI suggestions

Governance

- ☐ Control GitHub Copilot Business licences via Entra ID group membership
- ☐ Configure GitHub Copilot organisation settings: enable/disable code suggestions, block public code suggestions
- ☐ Publish AI code generation policy enforced via branch protection rules in GitHub/Azure DevOps
- ☐ Require AI-generated code flagging in PR descriptions (use PR template with AI-generated code checkbox)

Identity

- ☐ Federate GitHub Enterprise with Entra ID SSO and enforce Conditional Access with MFA
- ☐ Restrict Copilot admin settings to GitHub organisation owners via Entra ID role assignment
- ☐ Enforce SAML SSO for all developer access to GitHub with Copilot features

Data

- ☐ Configure GitHub Copilot to exclude sensitive repositories from context using content exclusion settings
- ☐ Enable GitHub Copilot code referencing to identify when suggestions match public code
- ☐ Review GitHub Copilot data handling: verify telemetry and suggestion data retention settings
- ☐ Deploy GitHub Advanced Security secret scanning to catch leaked credentials in AI-generated code

Processing (Apps, APIs, Integration)

- ☐ Enable GitHub Advanced Security: CodeQL (SAST), Dependabot (SCA), secret scanning across all repos
- ☐ Configure Azure DevOps pipeline gates requiring SAST pass before merge to main
- ☐ Enforce branch protection: require PR review with minimum 2 approvals, require status checks to pass
- ☐ Deploy GitHub Advanced Security code scanning with auto-fix suggestions for AI-generated vulnerabilities
- ☐ Use Dependabot to verify all AI-suggested dependencies exist and are version-pinned

Network

- ☐ Route GitHub Copilot traffic via corporate proxy if required by network policy
- ☐ Block AI coding tool endpoints from production Azure DevOps build agents that do not need them

Supply Chain

- ☐ Use GitHub Advisory Database and Dependabot to scan for vulnerable dependencies in AI-suggested packages
- ☐ Configure Azure Artifacts (or GitHub Packages) as approved package feed; block direct public registry access
- ☐ Generate SBOMs using GitHub dependency graph and submit to Dependency Review
- ☐ Pin all dependency versions; configure Dependabot to alert on new CVEs in pinned versions

Monitoring and Operations

- ☐ Monitor GitHub Advanced Security findings dashboard for vulnerability trends in AI-generated code
- ☐ Stream GitHub Audit Log to Sentinel for security event correlation
- ☐ Track Copilot usage metrics via GitHub Copilot Usage API and alert on anomalies
- ☐ Integrate Dependabot alerts into Sentinel via GitHub webhook connector
- ☐ Review code scanning findings weekly with focus on AI-generated code patterns

4.3 L3: Citizen Developer Agents

Example: Power Automate flows, Power Apps with AI Builder, Copilot Studio agents

Governance

- ☐ Deploy Power Platform DLP policies via Power Platform Admin Centre to classify connectors into Business/Non-Business/Blocked groups
- ☐ Use Power Platform environments: enforce sandbox for development, require admin approval for production environment access
- ☐ Maintain Power Platform Centre of Excellence (CoE) Starter Kit for inventory of all flows, apps, and agents
- ☐ Require Copilot Studio agent review via environment-level governance before publishing to production
- ☐ Apply Power Platform tenant isolation to prevent cross-tenant connector access
- ☐ Enforce mandatory approval steps (Power Automate approvals connector) for flows writing to production data
- ☐ Conduct quarterly access reviews of all citizen-built flows using CoE Kit analytics

Identity

- ☐ Enforce Conditional Access policies for Power Platform access with MFA and compliant device requirement
- ☐ Use service principals or Managed Identity for Power Automate connections where supported (not user delegated)
- ☐ Store connector credentials in Azure Key Vault referenced via custom connectors (not embedded in flows)
- ☐ Apply Power Platform security roles restricting who can create premium connectors and custom connectors
- ☐ Implement PIM for Power Platform Admin role: require JIT activation with approval
- ☐ Audit Power Platform connector credential usage in Entra ID sign-in logs

Data

- ☐ Apply Power Platform DLP policies blocking connectors that mix business-critical and personal data sources
- ☐ Enforce sensitivity labels on SharePoint, Dynamics 365, and Dataverse data accessed by Power Platform flows
- ☐ Use Azure Information Protection to classify data flowing through Power Automate connectors
- ☐ Encrypt Dataverse data at rest with customer-managed key (CMK) via Key Vault
- ☐ Enable Power Platform activity logging for audit trail of all flow executions and data access

Processing (Apps, APIs, Integration)

- ☐ Enforce Power Platform environment separation: dev, test, production with distinct DLP policies per environment
- ☐ Route custom connector traffic through Azure API Management with rate limiting and schema validation
- ☐ Apply Power Automate flow run limits and throttling per environment
- ☐ Block or restrict HTTP and custom connectors to admin-approved users only via DLP policy
- ☐ Deploy Copilot Studio agents with conversation topic guardrails and action confirmation prompts
- ☐ Test citizen developer flows against over-privilege scenarios before production promotion

Network

- ☐ Use Power Platform on-premises data gateway with restricted network access for hybrid connectivity
- ☐ Deploy Azure Virtual Network data gateway for Power Platform to access Azure resources via Private Link
- ☐ Apply Power Platform tenant isolation to prevent data flows to external tenants
- ☐ Restrict outbound traffic from Power Platform custom connectors via APIM network policies

Supply Chain

- ☐ Curate approved connector catalogue: disable non-approved connectors via DLP policy
- ☐ Vet Power Platform template gallery submissions before making available to citizen developers
- ☐ Monitor Power Platform connector security advisories from Microsoft
- ☐ Verify publisher identity for custom connectors before enabling in production environments

Monitoring and Operations

- ☐ Enable Defender for Cloud Apps for Power Platform: detect anomalous flow creation, mass data downloads
- ☐ Stream Power Platform activity logs to Sentinel via Management Activity API connector
- ☐ Create Sentinel analytics rules for: new premium connector usage, flow execution spikes, error rate increases
- ☐ Deploy Power Platform CoE Kit analytics dashboards for flow inventory and health monitoring
- ☐ Monitor Power Automate flow run history for failed executions indicating permission or security issues
- ☐ Conduct dormant flow audits quarterly: identify and disable unused flows with active connections

4.4 L4: Data Analytics & API Orchestration

Example: Python/C# calling Azure OpenAI API, Logic Apps with AI, Data Factory with LLM enrichment, RAG with Azure AI Search

Governance

- ☐ Require threat modelling for pipelines processing confidential data via Azure OpenAI or third-party AI APIs
- ☐ Document all AI API dependencies in Azure DevOps wiki or Architecture Decision Records
- ☐ Set Azure Cost Management budgets with alerts on Azure OpenAI consumption (50%, 80%, 100% thresholds)
- ☐ Use Azure Policy to enforce tagging compliance on all AI pipeline resources

Identity

- ☐ Use Managed Identity for all Azure services calling Azure OpenAI (no API keys where possible)
- ☐ Store third-party AI API keys in Key Vault with automatic rotation via Key Vault rotation policy
- ☐ Apply Azure OpenAI RBAC: scope Cognitive Services OpenAI User role to specific resource groups
- ☐ Apply Key Vault access policies using RBAC (not access policies) with least-privilege roles
- ☐ Never store API keys in App Settings, environment variables, or Azure DevOps pipeline variables in plaintext
- ☐ Enable Key Vault diagnostics: alert on unusual secret access patterns

Data

- ☐ Enforce CMK encryption on Azure OpenAI, Azure AI Search, and all storage accounts via Azure Policy
- ☐ Apply Azure OpenAI content filtering (built-in) and configure severity thresholds for harmful content
- ☐ Enable Azure OpenAI diagnostic logging for all API interactions (prompts, completions, token usage)
- ☐ Use Azure AI Search with role-based security trimming to enforce data access control in RAG pipelines
- ☐ Apply Azure Storage encryption with Key Vault CMK and enforce TLS 1.2 minimum
- ☐ Implement data minimisation: send only necessary context to Azure OpenAI (trim long documents)
- ☐ Enable Purview Data Map to track data lineage through AI enrichment pipelines

Processing (Apps, APIs, Integration)

- ☐ Deploy Azure API Management as gateway for all Azure OpenAI and third-party AI API calls with rate limiting, quotas, and subscription keys
- ☐ Implement circuit breaker patterns in Logic Apps / Functions using retry policies and dead-letter queues
- ☐ Separate dev, staging, and production Azure OpenAI deployments in distinct resource groups with distinct RBAC
- ☐ Version all pipeline code in Azure DevOps with branch protection and mandatory PR reviews
- ☐ Deploy Azure Functions with latest runtime; enable Azure DevOps pipeline SAST/DAST stages
- ☐ Apply APIM request validation policies: check content-type, max body size, and schema

Network

- ☐ Deploy Azure OpenAI with Private Endpoint in dedicated VNet; disable public network access
- ☐ Route all AI API traffic through APIM deployed in VNet with NSG rules
- ☐ Apply NSGs restricting pipeline compute (Functions, AKS, App Service) to approved destinations only
- ☐ Enable Azure Firewall or NVA for egress filtering on pipeline VNets
- ☐ Enable NSG Flow Logs and VNet Flow Logs for pipeline subnets

Supply Chain

- ☐ Scan all Python/Node.js dependencies with GitHub Dependabot or Snyk integrated in Azure DevOps pipelines

- ☐ Pin Azure OpenAI SDK and all AI client library versions; monitor for security advisories
- ☐ Use Azure Artifacts as approved package feed; block direct public registry access from build agents
- ☐ Generate SBOMs for all pipeline artifacts

Monitoring and Operations

- ☐ Monitor Azure OpenAI usage via Azure Monitor metrics: alert on token consumption spikes, error rates, and latency
- ☐ Track costs via Azure Cost Management with automated alerts
- ☐ Stream Azure OpenAI diagnostics, APIM logs, and Function logs to Log Analytics and Sentinel
- ☐ Enable Defender for Cloud for all pipeline subscriptions; review AI-specific recommendations
- ☐ Create Sentinel analytics rules for credential access anomalies from Key Vault diagnostics
- ☐ Deploy Application Insights for end-to-end request tracing across pipeline components

4.5 L5: Custom Autonomous Agents

Example: Semantic Kernel agents on Azure Functions, LangChain on AKS, Azure AI Agent Service, Copilot Studio custom agents with tools

Governance

- ☐ Require formal threat model (STRIDE for AI) before any autonomous agent is deployed to production
- ☐ Define graduated autonomy tiers: high-privilege actions require Durable Functions human approval activity
- ☐ Publish Agent Security Standard; enforce via Azure Policy on agent hosting resources
- ☐ Conduct agent red-team exercises in isolated dev/test subscriptions
- ☐ Define agent decommissioning runbook: Managed Identity deletion, Key Vault secret purge, Cosmos DB memory cleanup

Identity

- ☐ Assign each agent a unique Managed Identity; shared identities between agents are prohibited. Scope RBAC to approved tools/resources only at the narrowest scope (resource or resource group) and use deny assignments where applicable to prevent lateral privilege reuse
- ☐ Enforce tool-call permissions via Azure RBAC and resource-level access policies (not just agent prompt)
- ☐ Implement Durable Functions human approval activity for high-privilege agent actions
- ☐ Use Key Vault with RBAC and private endpoint for all agent secrets
- ☐ Apply PIM for any elevated agent identity access with time-limited activation
- ☐ Audit all agent Managed Identity usage in Entra ID sign-in and audit logs
- ☐ Prohibit client secrets or certificate-based app credentials for autonomous agents unless technically unavoidable. Where exceptions are required, enforce automatic rotation via Key Vault and Sentinel alerting on credential use

Data

- ☐ Encrypt agent memory stores (Cosmos DB, Azure AI Search, Redis) with CMK via Key Vault
- ☐ Apply Cosmos DB role-based access control: agents can only access their own partition key
- ☐ Implement Cosmos DB TTL on memory documents for automatic data expiry
- ☐ Deploy Azure OpenAI content filtering and custom blocklists for agent input/output guardrails
- ☐ Enable diagnostic logging on all agent data stores for audit trail
- ☐ Apply Purview sensitivity labels to data accessible to agents

Processing (Apps, APIs, Integration)

- ☐ Deploy agents on AKS with pod security standards (restricted profile) or Azure Container Apps with scale limits
- ☐ Enforce tool-call allowlists via Semantic Kernel plugin configuration validated at deployment time
- ☐ Apply Azure Functions or Container Apps scale limits to prevent runaway agent execution
- ☐ Version all agent configs, system prompts, and tool definitions in Azure DevOps repos with PR review
- ☐ Deploy Azure AI Content Safety as infrastructure-level guardrail (not just prompt-level)
- ☐ Implement health probes and circuit breakers using AKS liveness/readiness probes or Durable Functions patterns
- ☐ Test agents against prompt injection and tool abuse in isolated subscription environments

Network

- ☐ Deploy agents in dedicated VNet subnets with NSGs restricting egress to approved tool endpoints only
- ☐ Use Private Endpoints for Azure OpenAI, Cosmos DB, Key Vault, and Azure AI Search
- ☐ Block all outbound internet from agent subnets via Azure Firewall with explicit allowlist rules
- ☐ Implement NSG-based kill switch: Azure Function triggered by Sentinel incident that modifies NSG rules to isolate agent network segments. On activation, disable the agent's Entra Workload Identity or revoke its role assignments in addition to network isolation – network containment alone is

insufficient. Kill switches should be implemented as an automated containment runbook: revoke effective permissions, block network reachability, and quarantine workloads. A kill switch is not a single Azure feature – it is a coordinated set of actions executed reliably under incident conditions

- ☐ Enable NSG Flow Logs for agent subnets with Traffic Analytics

Supply Chain

- ☐ Pin Semantic Kernel / LangChain versions in container images; scan with Defender for Containers
- ☐ Use Azure Artifacts for approved packages; block public registry from build agents
- ☐ Audit external tool APIs consumed by agents: verify TLS, authentication, and security posture
- ☐ Enable ACR vulnerability scanning for all agent container images

Monitoring and Operations

- ☐ Log all agent tool calls in structured JSON to Application Insights custom events
- ☐ Create Sentinel analytics rules for agent behavioural anomalies: unusual tool calls, frequency spikes
- ☐ Monitor Cosmos DB / Redis memory store growth with Azure Monitor alerts
- ☐ Enable Defender for Cloud and Defender for Containers for all agent hosting subscriptions
- ☐ Deploy canary inputs via Logic Apps scheduled triggers to verify agent behaviour periodically
- ☐ Integrate agent Application Insights telemetry into Sentinel for SOC visibility

4.6 L6: Data Science & ML Pipelines

Example: Azure Machine Learning workspace, fine-tuning in Azure AI Studio, Databricks ML, Azure Synapse feature engineering

Governance

- ☐ Require formal threat model for all Azure ML training projects covering data poisoning and model supply chain risk
- ☐ Use Azure ML Model Registry with approval stages: require security sign-off before production deployment
- ☐ Enable Azure ML Responsible AI dashboard for bias, fairness, and explainability assessment
- ☐ Deploy Azure Policy restricting ML compute creation to approved VM sizes and regions
- ☐ Maintain Azure ML model cards for all production models
- ☐ Use Azure ML Data assets with versioning for training data provenance tracking

Identity

- ☐ Apply Azure ML workspace RBAC: separate roles for Data Scientist, ML Engineer, and ML Admin
- ☐ Use Managed Identity for all Azure ML compute (clusters, instances, endpoints)
- ☐ Restrict Model Registry write access to Azure DevOps pipeline service principal only
- ☐ Apply PIM for Azure ML workspace admin role with JIT activation
- ☐ Store all external model API keys in Key Vault with auto-rotation
- ☐ Enforce MFA on all Azure ML workspace users via Conditional Access
- ☐ Sign model artifacts: store the approved artifact hash (SHA-256) in Model Registry metadata or release manifest. Before deployment, compute the hash of the deployment artifact and compare approved vs deployed. Block promotion on mismatch

Data

- ☐ Encrypt Azure ML datastores with CMK via Key Vault (Storage, Blob, ADLS Gen2)
- ☐ Use Azure ML Data assets to version and track all training datasets
- ☐ Apply ADLS Gen2 ACLs or Synapse workspace managed private endpoints for training data isolation
- ☐ Enable Purview Data Map for end-to-end training data lineage tracking
- ☐ Use Macie equivalent (Purview Data Map sensitive data discovery) to scan training data for unintended PII
- ☐ Apply Storage Lifecycle Management policies for training data retention and deletion
- ☐ Enable Azure Storage diagnostic logging for all training data access events
- ☐ Apply Azure ML managed network to prevent training data exfiltration from compute

Processing (Apps, APIs, Integration)

- ☐ Deploy Azure ML managed compute in managed VNet with workspace-managed outbound rules
- ☐ Use Azure ML Pipelines (v2) for immutable, versioned pipeline definitions stored in Git
- ☐ Apply compute quotas and auto-shutdown on ML compute instances to prevent misuse
- ☐ Enable Azure ML Environment with curated images; scan custom Docker images with Defender for Containers
- ☐ Deploy Responsible AI components (fairness, explainability) as mandatory pipeline stages
- ☐ Implement model validation gates: accuracy, bias, and robustness tests before Registry approval
- ☐ Enforce Azure DevOps pipeline for model promotion: no manual model deployment to endpoints

Network

- ☐ Deploy Azure ML workspace in managed VNet mode (recommended) or customer-managed VNet
- ☐ Apply workspace managed outbound rules: allow only approved storage, ACR, and Key Vault endpoints
- ☐ Use Private Endpoints for Azure ML workspace, Storage, Key Vault, and ACR
- ☐ Block internet access from ML compute clusters; use workspace-managed outbound for approved packages only

- ☐ Enable NSG Flow Logs on ML workload subnets

Supply Chain

- ☐ Verify provenance of base models from Azure AI model catalog or HuggingFace Hub
- ☐ Enable ACR vulnerability scanning for all custom training and inference images
- ☐ Pin ML framework versions in Azure ML Environment definitions; scan for CVEs
- ☐ Use Azure Artifacts for Python packages consumed by training pipelines
- ☐ Generate SBOMs for all ML containers
- ☐ Store model checkpoints with storage versioning and hash in Model Registry metadata

Monitoring and Operations

- ☐ Enable Azure ML Model Monitor for data drift, prediction drift, and data quality
- ☐ Monitor ML compute utilisation with Azure Monitor alerts for cryptomining or unauthorised workloads
- ☐ Stream Azure ML workspace diagnostics to Log Analytics and Sentinel
- ☐ Enable Defender for Cloud for all ML subscriptions with Azure ML-specific Config policies
- ☐ Track Model Registry operations in Azure ML audit logs: alert on model promotion and deletion
- ☐ Run periodic model integrity checks (minimum daily): recompute deployed endpoint model artifact hash and compare against Registry-approved hash. Alert and block on mismatch to detect post-deployment drift or tampering. Automate via Azure Automation or Logic Apps on schedule
- ☐ Integrate Azure ML monitoring into centralised Sentinel dashboards

4.7 L7: Model Hosting & Serving

Example: Azure ML managed online endpoints, Azure ML serverless endpoints, Azure OpenAI PTU, self-hosted on AKS

Governance

- ☐ Publish Model Serving Security Standard; enforce via Azure Policy on endpoint resources
- ☐ Require security review gate in Azure DevOps pipeline before any Azure ML endpoint deployment
- ☐ Define rate-limiting policies on APIM to impede model extraction attempts
- ☐ Maintain endpoint inventory in Azure ML managed endpoints dashboard or Azure Resource Graph queries
- ☐ Define rollback SLA: use Azure ML endpoint traffic splitting for instant rollback

Identity

- ☐ Apply Azure ML endpoint authentication: key-based or Entra ID token (prefer Entra ID for internal consumers)
- ☐ Deploy APIM with subscription keys and OAuth 2.0 validation for per-consumer rate limiting
- ☐ Use Managed Identity for endpoint compute accessing Azure resources (storage, Key Vault)
- ☐ Audit endpoint invocation events in Azure ML workspace diagnostics and APIM analytics
- ☐ Apply Azure RBAC restricting who can invoke vs manage endpoints

Data

- ☐ Enforce TLS 1.2+ on all Azure ML endpoints (default) and AKS ingress controllers
- ☐ Deploy Azure AI Content Safety for input/output filtering on inference endpoints
- ☐ Enable Azure ML data collection (MDC) for production inference logging with CMK encryption
- ☐ Implement input validation via APIM request policies: schema, size limits, content type
- ☐ Apply Purview PII detection on model outputs where models may generate personal data

Processing (Apps, APIs, Integration)

- ☐ Verify model artifact integrity before deployment: compute the hash of the deployment artifact and compare against the approved hash stored in Azure ML Model Registry metadata. Block deployment on mismatch
- ☐ Implement canary deployments using Azure ML endpoint traffic mirroring and traffic splitting
- ☐ Deploy blue-green via Azure ML endpoint deployment slots with instant traffic switch
- ☐ Apply AKS pod security standards (restricted) for self-hosted inference containers
- ☐ Implement APIM policies: rate limiting, request throttling, IP filtering, circuit breaker
- ☐ Test endpoints against adversarial inputs and extraction probes in staging deployment

Network

- ☐ Deploy Azure ML managed endpoints with workspace managed VNet for network isolation
- ☐ Place APIM in internal VNet mode with Front Door and WAF for external consumer access
- ☐ Enable Azure DDoS Protection Standard on VNet hosting inference endpoints
- ☐ Apply NSGs restricting endpoint compute egress (no outbound internet)
- ☐ Use Private Endpoints for internal consumer access to Azure ML endpoints

Supply Chain

- ☐ Verify model artifact provenance from Azure ML Model Registry before deployment
- ☐ Scan inference container images with ACR scanning and Defender for Containers
- ☐ Pin serving framework versions and monitor for security advisories
- ☐ Implement ACR content trust (Docker Content Trust) for image signing

Monitoring and Operations

- ☐ Monitor Azure ML endpoint metrics (latency, error rates, request count) via Azure Monitor with auto-scaling rules

- ☐ Enable Azure ML Model Monitor for output distribution drift detection in production
- ☐ Deploy canary queries via Logic Apps and alert on output deviation from baseline
- ☐ Create Azure Monitor anomaly detection alerts on invocation patterns for extraction attempts
- ☐ Stream APIM analytics and Azure ML diagnostics to Sentinel
- ☐ Enable Defender for Cloud for inference hosting subscriptions
- ☐ Monitor endpoint deployment events and model version changes via Azure ML workspace audit logs

4.8 L8: Distributed / Multi-Agent

Example: Multi-agent Semantic Kernel on AKS, AutoGen agents, A2A/MCP implementations on Azure

Governance

- ☐ Require system-level threat modelling covering inter-agent trust, lateral movement, and cascading failure
- ☐ Publish Multi-Agent Security Architecture Standard with trust boundary definitions
- ☐ Mandate zero-trust between agents: use a unique Entra Workload ID for each agent identity. Shared identities between agents are prohibited. Agents must never be permitted to modify their own identity bindings, access policies, or trust relationships under any circumstance
- ☐ Implement infrastructure-level kill switches via Azure Automation runbooks modifying NSG rules
- ☐ Conduct multi-agent red team exercises in isolated dev/test subscriptions
- ☐ Use Management Group structure to define blast radius boundaries per agent trust domain

Identity

- ☐ Assign each agent a unique Entra Workload Identity (Federated Identity Credential on AKS)
- ☐ Implement mTLS between agents using Azure Key Vault certificates or Azure App Service Certificates
- ☐ Apply Entra ID application permissions restricting each agent's API access scope
- ☐ Prevent trust delegation: use Entra ID on-behalf-of flow with explicit consent, not implicit delegation
- ☐ Implement short-lived Entra ID tokens (max 1 hour) for inter-agent authentication
- ☐ Audit all inter-agent authentication events in Entra ID sign-in logs and send to Sentinel

Data

- ☐ Encrypt all inter-agent messages with mTLS and optional payload encryption via Key Vault
- ☐ Implement signed messages between agents using Key Vault asymmetric keys
- ☐ Apply VNet service endpoints and Private Link policies preventing data leakage across trust domains
- ☐ Log all inter-agent data exchanges to Log Analytics in structured format

Processing (Apps, APIs, Integration)

- ☐ Deploy centralised orchestration using Durable Functions with human approval gates
- ☐ Implement circuit breakers in Durable Functions or AKS using health checks and error thresholds
- ☐ Isolate each agent in its own AKS pod with Kubernetes Network Policies preventing cross-namespace access
- ☐ Deploy KEDA auto-scaling with rate limits on inter-agent Service Bus / Event Grid messaging
- ☐ Use Azure Chaos Studio for fault injection testing of multi-agent resilience
- ☐ Implement graceful degradation: agents must handle peer agent unavailability without cascading failure

Network

- ☐ Apply Kubernetes Network Policies preventing lateral movement between agent namespaces
- ☐ Enforce mTLS via AKS service mesh (Istio, Linkerd, or Open Service Mesh) for inter-agent traffic. Certificates must be bound to individual agent identities (Workload Identity or pod identity); shared certificates across agents are prohibited. Implement automated certificate rotation with certificate lifetime not exceeding 24 hours for agent identities
- ☐ Segment agent VNets from broader infrastructure using VNet peering with NSG restrictions
- ☐ Deploy Azure Firewall kill switches: Azure Automation modifies firewall rules to isolate agent groups. On activation, disable the agent's Entra Workload Identity or revoke its role assignments in addition to network isolation – network containment alone is insufficient. Kill switches should be implemented as an automated containment runbook: revoke effective permissions via Entra ID, block network reachability via NSG/Firewall, and quarantine workloads. A kill switch is not a single Azure feature – it is a coordinated set of actions executed reliably under incident conditions
- ☐ Enable NSG Flow Logs with Traffic Analytics for inter-agent network monitoring

Supply Chain

- ☐ Verify agent identity via Entra Workload ID before mesh admission
- ☐ Scan all agent container images with Defender for Containers and ACR scanning
- ☐ Implement ACR content trust requiring signed images for mesh agent deployment
- ☐ Monitor for rogue agents: alert on unregistered Workload Identity attempting API calls

Monitoring and Operations

- ☐ Deploy centralised Azure Monitor dashboards showing inter-agent message volumes, error rates, and latency
- ☐ Create Sentinel analytics rules for cascade detection: correlated failures across multiple agents
- ☐ Monitor Entra ID sign-in logs for inter-agent trust delegation anomalies
- ☐ Enable Defender for Cloud and Defender for Containers across all agent hosting subscriptions
- ☐ Integrate AKS monitoring (Container Insights) into Sentinel for multi-agent SOC visibility
- ☐ Use Azure Chaos Studio for periodic chaos engineering exercises validating resilience
- ☐ Alert on unexpected Workload Identity registration or deletion in Entra ID
- ☐ Retain full interaction history in Storage Account with immutability policies for forensics