OVHcloud
Professional Services



# Well-Architected Framework

## *Deep Dive Across All Six Pillars*

| | |
|---|---|
| Project | OVHcloud Well-Architected Framework |
| Company | OVHcloud |
| Date | August 2026 |
| Version | 2.0 |
| Author | OVHcloud Professional Services |

| Version | Date | Author | Description of changes |
|---|---|---|---|
| 1.0 | June 2026 | Professional Services | Initial release |
| 2.0 | August 2026 | Professional Services | Updated based on internal feedback |

# Table des matières

*Introduction* ..... 2
*Well-Architected Review Process for OVHcloud* ..... 2
*Pillar 1 Operational Excellence* ..... 4
*Pillar 2 Security* ..... 6
*Pillar 3 Reliability* ..... 8
*Pillar 4 Performance Efficiency* ..... 10
*Pillar 5 FinOps/Cost Optimisation* ..... 12
*Pillar 6 Sustainability* ..... 14

# Introduction

This document provides a deep-dive reference for architecting workloads on OVHcloud Public Cloud based on the best practices for 6 pillars. Each pillar is elaborated with OVHcloud-specific services, design patterns, anti-patterns, and concrete implementation guidance.

The framework is not prescriptive — it is a structured lens for evaluating trade-offs. Not every recommendation applies to every workload. Use this document as a reference during architecture design, peer review, and quarterly well-architected reviews.

### Framework Overview

| Pillar | Core objective |
|---|---|
| **1 · Operational Excellence** | Automate operations, observe workloads, and continuously improve |
| **2 · Security** | Protect data, systems, and assets under EU jurisdiction |
| **3 · Reliability** | Recover quickly from failure, meet availability targets |
| **4 · Performance Efficiency** | Use resources efficiently at any scale |
| **5 · Cost Optimisation** | Deliver value at the lowest viable cost |
| **6 · Sustainability** | Minimise environmental impact of cloud workloads |

# Well-Architected Review Process for OVHcloud

A Well-Architected Review is a structured and prioritized evaluation of a workload against the six pillars. It should be conducted: before the production launch of a new workload, after a significant architecture change, after a production incident, and quarterly for all Tier-1 workloads.

**Review Steps**

- Scope definition — identify the workload boundary, the teams responsible, and review participants.
- Per-pillar assessment — for each of the six pillars, work through framework questions. Capture findings as: no risk (green), medium risk (amber), or high risk (red).
- Prioritization — rank findings by business impact × implementation effort. Security and reliability high-risk findings are always priority 1.
- Improvement plan — for each high/medium risk finding, define a specific remediation action, an owner, and a target date with verifiable deliverables.
- Implementation — deliver improvements via IaC pull requests, CI/CD deployment, and staging validation before production.
- Re-review — schedule the next review before closing the current one. Tier-1: quarterly. Tier-2: semi-annual. Tier-3: annual.

## OVHcloud-Specific Review Checklist

| Pillar | Key questions to answer in every review |
|---|---|
| **Operational Excellence** | Is all infrastructure managed via infrastructure as code ? Are Logs Data Platform dashboards and alerts active ? Was the last runbook tested within 90 days? |
| **Security** | Are all instances in vRack ? As no direct internet interfaces is recommended, use of floating IP if instance has to be accessed for internet ? Is MFA enabled for all Control Panel users? Are Security Groups default-deny? Is a SecNumCloud region required for this workload? Is HDS required for this workload? |
| **Reliability** | Is the workload deployed across at least 2 AZs? Is there a tested DR plan with measured RTO/RPO? Was the last backup restoration test successful? Are managed services (database, Kubernetes, private registry, S3, ...) multi-AZ? |
| **Performance Efficiency** | Is the correct instance flavour family in use? Is CPU utilisation > 30% on average? Are managed services used in preference to self-managed equivalents? Are load tests run before every release? |
| **Cost Optimisation** | Are all public cloud project tagged? Are Savings Plans in place for baseline compute? Are there any instances with < 10% CPU for > 2 weeks? Are Object Storage lifecycle rules configured? |
| **Sustainability** | What is the current Scope 1+2 footprint from the Environmental Impact Tracker? Are non-production instances stopped outside working hours? Is the workload in the lowest-carbon available region? |

# Pillar 1 Operational Excellence

*Automate, observe, and continuously improve every layer of your OVHcloud workload.*

**OVHcloud services covered**

OVHcloud API · OpenStack API · Terraform/ OpenTofu OVH provider · · Horizon dashboard · Logs Data Platform · OVHcloud CLI - OVHcloud Cloud Shell

**Overview**

The Operational Excellence pillar focuses on running and monitoring systems to deliver business value and continually improving supporting processes and procedures. On OVHcloud, this means treating infrastructure as code, building observability from day one, and creating a culture of continuous improvement through post-mortems and toil reduction.

**Design Principles**

- Perform operations as code — use the Terraform OVH provider or OpenStack provider for all resource definitions; avoid manual clicks in the Control Panel.
- Treat cloud infrastructure as software — every change flows through a pull request, peer review, and CI/CD pipeline before reaching production.
- Build observability before the first production workload — instrument metrics, logs, and traces from day one using OVHcloud Logs Data Platform.
- Define the operating model before deployment — runbooks, on-call escalation paths, and post-mortem processes must exist before production traffic arrives.
- Measure and reduce toil quarterly — any manual task performed more than twice per sprint is a candidate for automation.

**Key Guidance**

| Area | Guidance |
|---|---|
| **Infrastructure as Code** | Use the Terraform OVH provider or OpenStack provider. Store all states in OVHcloud Object Storage (S3-compatible). |
| **GitOps & Change Management** | Enforce branch protection on main. Run terraform plan on every PR and terraform apply on merge via GitHub Actions or GitLab CI. Use separate OpenStack projects for dev, staging, and production. |
| **Observability Stack** | Deploy Prometheus + Grafana for metrics, Logs Data Platform (OpenSearch/Graylog) . Define SLOs and error budgets per service. |
| **Runbooks & Incidents** | Every recurring task (instance resize, snapshot rotation, |

| | |
|---|---|
| | certificate renewal, incident response) must have a documented runbook to roll back if required in order to minimise P1 incidents resolution time |
| **Operational Maturity** | Support tier: production workloads should be on OVHcloud Business or Enterprise support. Use OVHcloud public product roadmap on GitHub to track upcoming changes and deprecations. |
| **Anti-pattern to avoid** | Clicking "Create instance" in Horizon or the Control Panel without a corresponding Terraform resource creates infrastructure drift that becomes invisible to the state file. |

# Pillar 2 Security

*Protect data, systems, and assets. Leverage OVHcloud European sovereignty to eliminate extraterritorial legal risk.*

**OVHcloud services covered**

OVHcloud IAM · OpenStack Keystone · Security Groups · vRack private network · Anti-DDoS (VAC) · SecNumCloud zones · Key Management Service · Object Storage SSE · Block Volume encryption - OVHcloud Connect - Managed Private Registry

**Overview**

The Security pillar encompasses protecting data, systems, and assets while delivering business value through risk assessments and mitigation strategies. OVHcloud European ownership and SecNumCloud qualification eliminate CLOUD Act exposure. Security must be applied at every layer — network, compute, application, and data — and assumed-breach thinking must underpin every design decision.

**Design Principles**

- Implement a strong identity foundation — never share root/NIC credentials; assign least-privilege policies per team, service, and pipeline using OVHcloud IAM.
- Enable traceability — forward all OVHcloud managed services logs to Logs Data Platform;
- Apply security at all layers — no instance handling sensitive data should expose a network interface directly to the internet.
- Protect data in transit and at rest — enable SSE-OMK on all Object Storage buckets, encrypt Block Storage volumes at creation time, and store certificates in OVHcloud KMS and secrets in OVHcloud secret manager. Use managed private registry to store your application artifacts.
- Prepare for security events — define a cloud-specific incident response runbook: how to revoke a compromised application credential, isolate an instance, and engage OVHcloud Enterprise Support.

**Key Guidance**

| Area | Guidance |
|---|---|
| **Identity & Access Management** | Use OVHcloud IAM for resource-level policies. Enforce MFA for all Control Panel users. Use OpenStack application credentials with limited scope and 90-day rotation for service-to-service auth. Never use the NIC root account for API calls. |
| **Network Security & Isolation** | Enforce default-deny on all Security Groups — restrict |

| | |
|---|---|
| | SSH to bastion IPs only, never 0.0.0.0/0. Use separate VLANs for control plane, data plane, management, and DMZ. |
| **Data Protection** | Enable SSE-OMK on all Object Storage buckets. Encrypt Block Storage volumes at creation. . Enable Object Storage versioning and  for backup/compliance buckets. Configure lifecycle rules to transition old versions to low cost or long term storage (Infrequent Access, Cold Archive...). |
| **Sovereignty & Compliance** | OVHcloud is not subject to the US CLOUD Act. For highest-sensitivity workloads, use the SecNumCloud Trusted Zone (Gravelines, Roubaix, Strasbourg). Additional certifications are also available: , HDS, PCI-SNC |
| **Detection & Response** | Run quarterly security drills: rotate all API tokens, review IAM policies, scan for unpatched CVEs, verify audit log forwarding. |
| **Anti-DDoS** | OVHcloud VAC anti-DDoS is automatically enabled for all public cloud instances and public IP at no cost, mitigating volumetric attacks at the network edge before reaching your instances. |

# Pillar 3 Reliability

*Recover quickly from failure, meet availability targets, and verify reliability through testing — not declaration.*

**OVHcloud services covered**

3-AZ regions (Paris, Milan and Berlin in 2027) · OVHcloud Load Balancer (Octavia) · Managed Databases (multi-AZ) · Managed Kubernetes · Object Storage (11-nines durability) · OVHcloud DNS failover

**Overview**

The Reliability pillar focuses on the ability of a workload to perform its intended function correctly and consistently when expected. On OVHcloud, reliability is achieved by designing for failure across the 3-AZ region architecture, preferring managed services with built-in HA, automating recovery procedures, and verifying all assumptions through regular testing. An untested DR plan is not a DR plan.

**Design Principles**

- Automatically recover from failure — design systems that self-heal through health checks, automatic instance replacement, and pod restarts.
- Test recovery procedures regularly — tabletop DR exercises, annual DR failover tests, and monthly backup restoration tests are mandatory, not optional.
- Scale horizontally to increase aggregate availability — avoid single points of failure at every layer.
- Define RTO and RPO before selecting backup mechanisms — not after an incident.
- Prefer OVHcloud managed services — Managed Databases, Managed Kubernetes, Managed Private Registry and Object Storage provide built-in HA and SLAs that would require significant engineering to match on self-managed deployments.

**Key Guidance**

| Area | Guidance |
|---|---|
| **Multi-AZ Architecture** | Deploy workloads across at least two OVHcloud Availability Zones within a region. Use anti-affinity rules for Managed Kubernetes node pools. Use OVHcloud Load Balancer (Octavia) with health checks and automatic instance switchover for IPLB. |
| **Backup & Restore** | OVHcloud Object Storage provides 11-nines (99.999999999%) durability — it is the most cost-effective and durable backup target on the platform. Validate restore procedures monthly. Enable automated |

| | |
|---|---|
| | snapshots for Managed Databases and block volumes. |
| **Workload Tiers & RTO/RPO** | Define and document RTO/RPO per workload before selecting DR mechanisms. For instance: Critical (Tier 1): RTO < 1 hour. Important (Tier 2): RTO < 4 hours. Standard (Tier 3): RTO < 24 hours. Development: best effort. |
| **Disaster Recovery** | Implement multi-region failover. Execute a full DR failover test annually, measuring actual RTO and RPO. Update runbooks based on findings. |
| **Managed Services Preference** | Use Managed Databases (PostgreSQL, MySQL, MongoDB, OpenSearch…) and Managed Kubernetes with Managed Private Registry to benefit from built-in HA, automated failover, and OVHcloud SLAs. Use self-managed when a managed service does not meet a specific compliance requirement. |

# Pillar 4 Performance Efficiency

*Select the right OVHcloud services and flavours for each workload, and continuously validate that performance meets requirements.*

**OVHcloud services covered**

B-series (balanced) · C-series (CPU optimised) · R-series (RAM optimised) · T-series (GPU) · NVMe Block Storage · High-perf Object Storage · Managed Kubernetes · AI Training / AI Deploy · Managed Databases · CDN integration

**Overview**

The Performance Efficiency pillar focuses on using computing resources efficiently to meet system requirements, and maintaining that efficiency as demand changes and technologies evolve. OVHcloud offers a structured flavour hierarchy, NVMe storage options, and fully managed AI/GPU infrastructure that eliminates capacity management overhead for machine learning workloads.

**Design Principles**

- Right-size before committing — start with the smallest flavor that comfortably fits the workload at P95 load; monitor for 2 weeks; downsize if average CPU < 15% or RAM < 30%.
- Use flex option on public cloud instance to allow flavor downsizing.
- Match storage tier to access pattern — NVMe Block Storage for databases and IOPS-sensitive workloads, Standard Object Storage for backups and data lakes, High Performance Object Storage for AI/ML training, Cold Storage for archives.
- Use managed services to eliminate performance tuning overhead — Managed Databases handle connection pooling, replication, and failover; Managed Kubernetes manages node OS and container runtime.
- Benchmark before every major release — define performance regression thresholds and fail the CI pipeline if P99 latency exceeds the threshold.
- Place AI training compute in the same region as training data — cross-region data transfer introduces latency and throughput constraints that dramatically extend training job duration.

**Key Guidance**

| Area | Guidance |
|---|---|
| **Instance Flavour Selection** | B-series for general web workloads. C-series for compute-intensive batch. R-series for memory-intensive batch. I-series (local NVMe) for I/O-intensive stateful workloads. T-series (GPU) for AI/ML inference and training. Right-size monthly; downsize if CPU < 15% sustained. |
| **Storage Tier Selection** | • NVMe Block Storage (up to 20,000 IOPS with High |

| | |
|---|---|
| | Speed Gen2): for databases and transaction logs.<br>• Standard Object Storage: backups, archives, data lake landing zones (11-nines durability, no IOPS guarantee — not suitable as a database backend).<br>• High Performance Object Storage: AI/ML datasets requiring multi-GB/s throughput.<br>• Cold Storage : regulatory archives accessed less than once per quarter. |
| **Auto-Scaling & Elasticity** | Configure Kubernetes Horizontal Pod Autoscaler (HPA) tied to CPU/memory thresholds. Use Cluster Autoscaler to scale node pools down when underutilised. |
| **AI & GPU Workloads** | AI Training jobs: billed per GPU-minute, terminate on completion — no idle GPU cost. AI Deploy: auto-scales for inference, scales to zero when idle. |
| **Network Performance** | Deploy in the OVHcloud region closest to the majority of end-users. Use the OVHcloud private backbone (via vRack multi-region) for cross-region replication — not the public internet. |

# Pillar 5 FinOps/Cost Optimisation

*Deliver maximum business value at the lowest viable cost. OVHcloud pricing model eliminates several hyperscaler cost traps by design.*

**OVHcloud services covered**

Hourly pay-as-you-go · No internet egress fees (EU) · Savings Plans (1month to 3yr) · Cold Storage archiving · Resource tagging · Project budget alerts · Free Managed Kubernetes control plane

**Overview**

The Cost Optimisation pillar focuses on running systems to deliver business value at the lowest price point. OVHcloud structural advantages — no internet egress fees between European regions, no inter-AZ data transfer fees, and a free Managed Kubernetes control plane — change the cost architecture of workloads in fundamental ways. Workloads that require large volumes of data to end-users can be dramatically cheaper on OVHcloud than on hyperscalers through architecture freedom alone.

**Design Principles**

- Implement cloud financial management from day one — enforce mandatory tagging (team, environment, application, cost-center) on every project.
- Use separate OpenStack projects per team or application — this provides natural cost attribution without relying entirely on tags.
- Set budget alerts at 80% and 100% of monthly budget per OpenStack project — alerts must trigger before overspending, not after.
- Schedule a 30-minute monthly cost review — identify top 10 cost drivers and any instances running at < 10% CPU utilisation for more than 2 weeks.
- Commit to Savings Plans for predictable baseline compute — 3-year Savings Plans provide up to 54% discount; apply on-demand pricing only to variable or experimental workloads.

**Key Guidance**

| Area | Guidance |
|---|---|
| **Resource Tagging & Cost Allocation** | Enforce mandatory tags: team, environment, application, cost-centre. Use the OVHcloud Billing API to export consumption data to OVHcloud data platform for creating FinOps dashboards. Publish monthly cost reports per team — teams that receive bills for their infrastructure make better resource decisions. Leverage on the fact OVHcloud provides FOCUS 1.0-compliant |

| | |
|---|---|
| | data for all its Public Cloud services, enabling FinOps teams to collect, process, and analyse cloud financial data in a standardized way. |
| **Compute Optimisation** | Downsize instances where CPU < 15% or RAM < 30% sustained (highest-ROI optimisation available). Stop non-production instances outside working hours via OVHcloud API — can reduce non-production compute bills by 60–70%. Use Kubernetes resource requests and limits to prevent over-consumption. use Kubernetes Cluster Autoscaler to scale down node pools. |
| **Savings Plans & Commitments** | Commit 1 or 3-year Savings Plans for stable baseline compute (web servers, databases, always-on services) for up to 54% discount. Apply on-demand hourly pricing only to variable or experimental workloads. Free Managed Kubernetes control plane: no additional charge for the Kubernetes control plane — pay only for worker node instances. |
| **Storage Cost Management** | Configure Object Storage lifecycle rules on every bucket — automatically transition objects older than 90 days to Cold Storage (80–90% cheaper per GB). Delete snapshots for terminated/test instances monthly. Quarterly audit: identify unattached Block Storage volumes and volumes with consistently low I/O utilisation. |
| **Networking Costs** | No egress fees between OVHcloud EU regions or between services within a project. Audit Floating IPs monthly; release any that are not actively in use. |

# Pillar 6 Sustainability

*Minimise the environmental impact of your OVHcloud workloads. OVHcloud water-cooling technology and vertically integrated model provide industry-leading energy efficiency.*

**OVHcloud services covered**

PUE 1.09–1.30 · Water cooling (no AC) · Environmental Impact Tracker · Renewable energy PPAs · ISO 50001 certified · ECO refurbished servers · Cold Storage (energy-efficient archiving) · Low-carbon regions (FR, CA)

**Overview**

The Sustainability pillar focuses on minimising the environmental impacts of running cloud workloads. OVHcloud proprietary water-cooled data centres achieve a PUE as low as 1.09 — meaning only 0.09 watts of facility overhead per watt of IT load, versus an industry average of 0.55 watts. Choosing OVHcloud over a less efficient provider is itself the single largest sustainability decision a workload owner can make.

**Design Principles**

- Measure before you optimise — use the OVHcloud Environmental Impact Tracker (certified against GHG Protocol and Bilan Carbone methodology) to establish baseline Scope 1, 2, and 3 emissions.
- Select low-carbon regions for latency-insensitive workloads — France (nuclear ~70% + hydro) and Canada (hydroelectricity ~95%) have the lowest grid carbon intensity on the OVHcloud platform.
- Eliminate idle compute — use Kubernetes Cluster Autoscaler to scale node pools to zero; stop non-production instances outside working hours.
- Right-size aggressively for sustainability, not just cost — a smaller instance at 60% CPU utilization is more sustainable than a larger instance at 15% CPU running the same workload.
- Include cloud emissions in CSRD reporting — OVHcloud tracker provides the Scope 3 data needed to fulfil CSRD obligations for cloud-related emissions.

**Key Guidance**

| Area | Guidance |
|---|---|
| **OVHcloud Sustainability Baseline** | PUE 1.09–1.30 (measured annually across 88% of datacentres). Proprietary water cooling — no air conditioning required. ECO range: up to 17% of deployed components are refurbished; average server lifespan 4.5 years (up to 9). Net-zero target Scope 1+2: |

| | |
|---|---|
| | FY2025. All Scopes: FY2030. ISO 50001 certified, signatory of Climate Neutral Data Centre Pact. |
| **Carbon Footprint Measurement** | Use the OVHcloud Environmental Impact Tracker — certified by IJO, covering Scope 1 (direct), Scope 2 (electricity), and Scope 3 (value chain). Calculations use per-datacentre PUE and location-based energy mix data from Electricity Maps. Export via API for ESG/CSRD reporting. |
| **Region Carbon Intensity** | Lowest carbon: France (Gravelines, Roubaix, Strasbourg — nuclear + hydro) and Canada (Beauharnois — hydroelectricity ~95%). Higher carbon: Germany (mixed), UK (mixed), Poland (coal-heavy), Singapore (gas-heavy). Run latency-insensitive batch jobs (training, analytics) in France or Canada. |
| **Workload-Level Practices** | Eliminate idle compute with Cluster Autoscaler. Use Managed Kubernetes and container patterns for higher average utilisation. Move archival data to Cold Storage (low-power storage hardware). Schedule batch jobs during off-peak hours when grid carbon intensity is lowest. |
| **Software & Code Efficiency** | A 2× improvement in algorithmic efficiency reduces compute resource requirements by ~50% and energy consumption by a similar margin. Treat code efficiency as a sustainability lever, especially for high-throughput services. Minimise container image sizes to reduce registry storage and pull energy. |
| **Governance & Reporting** | Set annual reduction targets as a KPI alongside SLOs. Add a sustainability section to every architecture review: estimated carbon footprint, region selection rationale, and potential design changes (instance sizing, scheduling, storage tier). Cite OVHcloud net-zero commitments in supplier sustainability disclosures. |