

Request for Proposal: Virtualization Platform (Hypervisor and Management)

Template from RFPhub.com. Replace everything in square brackets, adjust the Must-have/Nice-to-have priorities, and delete this note and the instructions section before you issue the RFP.

Instructions for the buyer (delete before issuing)

A virtualization platform or hypervisor RFP involves infrastructure, storage, network, security, backup and application teams at once. Migration is the part that written answers tell you least about, so the proof of concept should move your own VMs and fail your own hosts. The example schedule in this template runs about 12 weeks from kickoff to contract award; allow longer if you need to test several hardware platforms or sites.

1. **Take inventory and set goals (weeks 1-2).** List your hosts, clusters and sites, server models with sockets and cores per host, VMs by guest operating system and version, your storage and network design, and the backup, monitoring and automation tools connected to the current hypervisor. Flag VMs that need special handling, such as clustered applications with shared disks, GPU workloads, virtual appliances and latency-sensitive systems. Note the date your current support or subscription term ends, because it sets the deadline for the migration.
2. **Assemble the buying team.** Include the virtualization and infrastructure team, storage, network, security and backup, the owners of your most important applications, the service desk, and procurement, finance and legal. Ask application owners to check whether their software vendors support each candidate hypervisor, because an unsupported database or virtual appliance can rule a platform out.
3. **Settle the architecture, and decide whether you need an RFI (weeks 2-4).** Decide whether you'll reuse your current servers or buy new ones, and whether storage will be hyperconverged, external SAN or NAS, or a mix. If those choices are still open, or your long list has more than five vendors,

send a short RFI asking for supported hardware, guest operating systems and storage options, migration tooling for your current hypervisor, and the licensing metric, then shortlist three or four vendors.

4. **Tailor and issue the RFP (weeks 4-7).** Delete requirements that don't apply, adjust priorities, and fill in the placeholders: your current hypervisor and version, server models, guest operating system list, VM counts and storage protocols. Give vendors three weeks to respond, with a written question-and-answer window in the first week.
5. **Score the responses (weeks 7-8).** Have each evaluator score independently against the rubric before comparing notes, and set aside any vendor that misses a Must-have. Check every 'yes' against the vendor's published hardware compatibility list, guest operating system support matrix and configuration maximums, and note whether each feature is generally available today or still on the roadmap.
6. **Run a proof of concept with two vendors (weeks 8-11).** Use your own hardware or a matching configuration. Migrate a representative set of VMs (Windows and Linux guests, a database, a VM with several disks and one with a fixed IP address) and record the downtime for each. Power off one host and cut another host's management network to watch high availability respond, back up and restore a VM with your backup software, apply a patch through a rolling update, and call a reference customer that migrated from the same hypervisor you run today.
7. **Negotiate and award (weeks 11-12).** Use the pricing table to compare total cost on your own hardware over three to five years. Settle the license metric, the renewal cap, what happens if you don't renew, migration services and exit terms before you sign, and align the start of the new subscription with your migration waves.

Contents

1. Introduction and Background
2. Project Objectives
3. Scope of Work
4. Requirements
5. Vendor Questions
6. Evaluation Criteria and Scoring

7. Pricing Response
8. Submission Guidelines
9. Timeline

1. Introduction and Background

[Company Name] is seeking proposals for a virtualization platform (hypervisor, cluster management, storage and networking) to run our virtual machines, together with the migration of our existing workloads from our current hypervisor. This RFP describes our requirements, the questions we need answered, and how proposals will be evaluated.

Organization background

- [Describe your organization: industry, size, locations]
- [Regulatory or contractual requirements that apply]
- [Current hypervisor, version and management tools, and the date the current support or subscription term ends]
- [Number of hosts, clusters and sites, with server models and CPU sockets and cores per host]
- [Number of VMs by guest operating system and version, and total vCPUs, memory and storage allocated]
- [Storage in use: hyperconverged, SAN or NAS, protocols, capacity and performance needs]
- [Network design: physical switching, VLANs, and any software-defined networking or micro-segmentation in use]
- [Backup, disaster recovery, monitoring and automation tools connected to the current platform]
- [Workloads needing special handling: clustered applications, shared disks, GPUs, virtual appliances, latency-sensitive systems]

2. Project Objectives

- Run our virtual machines on a supported platform that meets our availability, performance and security requirements
- Migrate [number] VMs from [current hypervisor] in planned waves, with defined downtime per VM and a rollback path

- Keep or improve current capabilities for high availability, live migration, storage, networking and backup
- Automate provisioning and operations through APIs and infrastructure as code
- Move to a predictable licensing and support model with a published product lifecycle

3. Scope of Work

The selected vendor will be responsible for:

- Delivery of the virtualization platform software and management tools for [number] hosts in [locations]
- Design and deployment of clusters, storage and networking on [our existing hardware / new hardware]
- Migration of [number] VMs from [current hypervisor] in waves agreed with [Company Name], including validation and rollback
- Integration with our identity provider, backup software, monitoring, SIEM and automation tools
- Training for administrators and operations staff
- Ongoing support, updates and security patches for the contract term

4. Requirements

For each requirement, enter a response code and a short explanation. Y = available today in the proposed edition; P = partially met or needs configuration/services (explain); R = on the roadmap (give a date); N = not supported. Must-have requirements are pass/fail; Nice-to-have requirements add points.

4.1 Functional

ID	Requirement	Priority	Response	Vendor comments
F1	A bare-metal (type 1) hypervisor for x86-64 servers that runs our guest operating systems [list each operating system and version], with a published guest operating system support matrix and the paravirtualized drivers and guest tools each one needs.	Must		
F2	VM lifecycle management from the console and API: create, clone, convert to and deploy from templates, snapshot, resize,	Must		

ID	Requirement	Priority	Response	Vendor comments
	and hot-add CPU, memory, disks and network adapters where the guest operating system supports it.			
F3	Live migration of running VMs between hosts in a cluster, and of their disks between storage locations, without downtime. Include live migration between hosts with different CPU generations through a CPU compatibility setting, state the limits on concurrent migrations, and state whether live migration between clusters is supported.	Must		
F4	High availability that automatically restarts VMs on surviving hosts after a host failure, with restart priority and order, admission control that reserves capacity for [number] host failures, and VM-to-VM and VM-to-host affinity and anti-affinity rules that high availability respects.	Must		
F5	Documented, configurable high availability behavior for host isolation, network partition and loss of storage access, including how the cluster detects failures (heartbeats, quorum or a witness) and how it prevents the same VM from running on two hosts at once. State the expected time from a host failure to VM restart.	Must		
F6	Maintenance mode that evacuates a host through live migration, so it can be patched, upgraded or replaced without VM downtime.	Must		
F7	Automatic load balancing that moves VMs between hosts based on CPU and memory use, with settings for how often and how aggressively it acts.	Nice		
F8	Import tooling that migrates VMs from our current hypervisor [name the platform and version]: it connects to the source platform, converts disks and virtual hardware to the target format, and maps source networks and storage to target networks and storage. List the source platforms and versions supported, and the documented limits of the tooling.	Must		
F9	Driver and guest tools handling during conversion: installing the paravirtualized storage and network drivers the target platform needs (for example, VirtIO drivers) before or during conversion, removing the source platform's guest tools, and keeping each VM's boot mode (BIOS or	Must		

ID	Requirement	Priority	Response	Vendor comments
	UEFI), disk order, MAC addresses and IP settings where we choose.			
F10	Bulk migration in scheduled waves, with both cold migration and warm migration (copying data while the source VM keeps running, then a final sync and a short cutover). State the expected downtime per VM for each method and what it depends on.	Must		
F11	A rollback path for every migrated VM: the source VM stays intact and can be powered back on if the migrated copy fails validation, plus a per-VM migration report showing what was converted, changed or skipped.	Must		
F12	Performance and health monitoring for VMs, hosts, clusters and storage (CPU, memory, storage latency and throughput, and network use), with configurable alerts and at least [period] of history.	Must		
F13	Capacity planning: forecasts of when CPU, memory and storage will run out, right-sizing recommendations for over- and under-sized VMs, and what-if modeling for new workloads or hosts.	Nice		
F14	GPU passthrough and shared virtual GPU support for [VDI, AI or graphics workloads], on GPUs whose vendor supports the platform. State the supported GPU families and any separate GPU software licensing.	Nice		

4.2 Technical and architecture

ID	Requirement	Priority	Response	Vendor comments
T1	One console and API to manage every cluster and site in scope. State whether the management plane runs as a VM or appliance, is distributed across the hosts, or is a vendor-hosted service; how it is made highly available; and what keeps working (running VMs, high availability restarts, live migration) while it is unavailable.	Must		
T2	Support for our server hardware [list server models, CPUs, network adapters and storage controllers], with a published hardware compatibility list. State whether servers of different generations or manufacturers can be mixed in one cluster, and any limits that creates.	Must		

ID	Requirement	Priority	Response	Vendor comments
T3	Documented configuration maximums (hosts per cluster, VMs per host and per cluster, and vCPUs, memory and disk size per VM) that meet our needs: [numbers]. State the minimum cluster size, and whether two-node clusters with a witness are supported for small sites.	Must		
T4	External SAN and NAS storage over the protocols we use [iSCSI, Fibre Channel, NVMe over Fabrics including NVMe over TCP, NFS], with multipathing and a published storage compatibility list. Confirm support for [list the storage arrays in use].	Must		
T5	Hyperconverged storage that pools the hosts' local drives, with replication or erasure coding across hosts, configurable failure domains, deduplication and compression, and documented behavior during drive failure, host failure and rebuild. Make this Must-have if you plan a hyperconverged design.	Nice		
T6	Thin provisioning, VM snapshots and space-efficient clones on every supported storage type. State snapshot limits and their performance impact, and whether storage quality-of-service limits can be set per VM or per disk.	Must		
T7	Virtual networking with VLAN-tagged networks (IEEE 802.1Q), virtual switch configuration defined once and applied to every host in a cluster, NIC teaming and link aggregation (LACP), jumbo frames, and IPv4 and IPv6 for VM and management traffic.	Must		
T8	Software-defined networking: overlay networks (for example, VXLAN or Geneve) that span hosts and clusters, virtual routing, IP address management, and optional BGP EVPN peering with our physical network.	Nice		
T9	VM-level firewalling: rules enforced at each VM's virtual network interface, managed centrally, and kept with the VM when it moves to another host. State whether rules can be based on tags or groups for micro-segmentation, and whether traffic flow visibility is included to help build rules.	Must		

4.3 Integration

ID	Requirement	Priority	Response	Vendor comments
I1	A documented backup API that supports image-level, incremental VM backups (snapshot-based, with changed block tracking or an equivalent) and application-consistent snapshots through a guest agent. Confirm support by our backup software [name and version], and list the backup products that officially support the platform today.	Must		
I2	Built-in VM backup to S3-compatible object storage, with support for object lock so copies can't be changed or deleted, for sites or workloads without a separate backup product.	Nice		
I3	A documented REST API (for example, with an OpenAPI description) covering every function available in the console, plus event notifications or webhooks for VM, host and alarm events.	Must		
I4	A Terraform provider and a command-line interface for managing VMs, networks, storage and cluster settings as code. State whether the Terraform provider is maintained and supported by the vendor.	Must		
I5	VM templates with first-boot customization through cloud-init for Linux guests and an equivalent mechanism for Windows guests (hostname, network settings, users, SSH keys and scripts).	Must		
I6	Single sign-on to the console and API through SAML 2.0 or OIDC with our identity provider, mapping identity-provider groups to platform roles.	Must		
I7	Streaming of audit, event and host logs to our SIEM over syslog, export of metrics to our monitoring tools (for example, in a Prometheus-compatible format or through the API), and alerts by email, webhook and our ticketing (ITSM) platform.	Must		
I8	Kubernetes alongside VMs: a supported way to run Kubernetes clusters on the platform, including a CSI driver for persistent volumes, or to run containers and VMs side by side under one management plane.	Nice		

4.4 Security and compliance