

Request for Proposal: Kubernetes Backup and Data Protection

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 Kubernetes backup evaluation involves platform engineering, storage, security and the teams that own stateful applications. The proof of concept matters more than the written answers, because a restore either works on your clusters or it doesn't. The example schedule in this template runs about 10 weeks from kickoff to contract award; allow longer if you need to prove restores across several clouds or regions.

1. **Take inventory and set protection tiers (weeks 1-2).** List your clusters, distributions and Kubernetes versions, the namespaces and stateful applications to protect, the storage classes and CSI drivers behind them, and the volume of persistent data. Group applications into tiers with RPO, RTO and retention targets, and write down the recovery scenarios you must prove: a deleted namespace, a lost cluster, a region outage and a ransomware attack.
2. **Assemble the buying team.** Include platform engineering or SRE, the backup and storage team, security, the owners of your most important stateful applications (often the database team), and procurement and legal. Decide early who will own backup policies, a central team or the application teams themselves, because that shapes the RBAC and self-service requirements.
3. **Decide whether you need an RFI first (weeks 2-3).** If you haven't decided between extending your current backup platform and adopting a Kubernetes-native product, or your long list has more than five vendors, send a short RFI. Ask which distributions, Kubernetes versions and CSI drivers each vendor supports, whether the management plane is self-hosted or vendor-hosted, and how licensing is counted, then shortlist three or four vendors.

4. **Tailor and issue the RFP (weeks 3-6).** Delete requirements that don't apply, adjust priorities, and add your clusters, storage platforms and backup targets to Section 1. If you run several Kubernetes distributions or managed Kubernetes services, list each one with its version, so vendors confirm support for every one rather than in general. Give vendors three weeks to respond, with a written question-and-answer window in the first week.
5. **Score the responses (weeks 6-7).** 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 support matrix, and note whether each feature is generally available today or still on the roadmap.
6. **Run a proof of concept with two vendors (weeks 7-9).** Use a non-production cluster and your real stateful applications. Delete a namespace and restore it, restore an application into a new cluster with a different storage class, try to delete backups using cluster-admin credentials, and time backups and restores at realistic data volumes. Check the resource footprint of the in-cluster components, and call a reference customer of similar size.
7. **Negotiate and award (weeks 9-10).** Use the pricing table to compare like for like, including how licenses are counted as clusters autoscale and for standby disaster recovery clusters. Settle the renewal cap, and exit terms that keep your backups restorable, before you sign.

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 Kubernetes backup and data protection platform that captures application-consistent backups of Kubernetes resources and persistent volumes, keeps immutable copies off the cluster, and restores applications to any of our clusters or clouds. 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]
- [Number of Kubernetes clusters, distributions and versions in use, by environment and location]
- [Hosting model: on-premises, managed Kubernetes in public cloud, edge, or a mix]
- [Number of namespaces and stateful applications to protect, and the databases they run]
- [Persistent storage platforms, storage classes and CSI drivers in use, and total persistent data]
- [How applications are deployed: GitOps, CI/CD pipelines, Helm charts, operators]
- [Current backup tooling, retention policies and recovery objectives (RPO/RTO)]
- [Object storage, backup targets and disaster recovery sites available]

2. Project Objectives

- Protect every stateful Kubernetes application with consistent, policy-driven backups of its resources and persistent data
- Meet defined RPO and RTO targets for each protection tier, proven by regular restore tests
- Keep immutable, isolated backup copies that survive ransomware or the loss of a cluster, site or region
- Restore and migrate applications across clusters, distributions and clouds
- Let application teams back up and restore their own namespaces within guardrails, and manage protection policies as code

3. Scope of Work

The selected vendor will be responsible for:

- Deployment of the backup platform and its in-cluster components across [number] clusters in [locations]
- Configuration of protection policies, backup targets and immutable copies for [number] namespaces
- Integration with our identity provider, storage platforms and CSI drivers, key management, monitoring, SIEM and ticketing systems
- Migration from our current backup tooling, keeping existing backups restorable until they expire
- Disaster recovery plans and restore testing for [named critical applications]
- Training for platform administrators and application teams
- Ongoing support 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	Backup of each application's namespace-scoped resources (deployments, stateful sets, services, config maps, secrets, service accounts, persistent volume claims and custom resources) together with its persistent volume data, as one restorable unit.	Must		
F2	Capture of the cluster-scoped resources an application depends on, such as custom resource definitions, storage classes and cluster roles, so the application can be restored into a new, empty cluster.	Must		
F3	Backup scope defined by cluster, namespace, label selector or application grouping, with automatic protection of new namespaces and workloads that match a policy.	Must		
F4	Application-consistent backups using pre- and post-snapshot hooks or reusable templates that quiesce, flush or dump databases and other stateful services. List	Must		

ID	Requirement	Priority	Response	Vendor comments
	the databases that have prebuilt templates.			
F5	Persistent volume protection through the Kubernetes CSI snapshot API, with snapshot data exported to a separate backup target rather than kept only as snapshots on the primary storage.	Must		
F6	A file-system-level backup method for volumes whose CSI driver does not support snapshots. State the performance and privilege trade-offs.	Must		
F7	Crash-consistent snapshots of all of an application's volumes at the same moment, using Kubernetes volume group snapshots where the CSI driver supports them.	Nice		
F8	Backup schedules and retention per policy (hourly, daily, weekly, monthly and yearly), with several backup targets per policy and separate retention for each target.	Must		
F9	Restore of a whole namespace, a single application, individual Kubernetes resources or a single persistent volume, to the original namespace or a new one.	Must		
F10	File-level restore from a persistent volume backup without restoring the whole volume.	Nice		
F11	Restore to a different cluster, region, cloud or Kubernetes distribution, with built-in transformations during restore (for example, changing the namespace, storage class, image registry, labels or ingress hostnames).	Must		
F12	Migration of applications between clusters, distributions and clouds (for example, for Kubernetes upgrades or cloud moves) using the same backup and restore workflow.	Must		
F13	Disaster recovery plans that restore a defined set of applications to a standby cluster in a set order, with pre- and post-restore steps, and that can be run on demand as a test or as a failover.	Must		
F14	Scheduled restores into a standby or isolated test cluster, for warm-standby disaster recovery and automated recovery testing, with a pass or fail report for each run.	Nice		
F15	Self-service backup and restore for application teams within their own	Must		

ID	Requirement	Priority	Response	Vendor comments
	namespaces, inside limits set by platform administrators.			
F16	Protection of virtual machines that run on Kubernetes through a Kubernetes-native virtualization layer, if they are in scope.	Nice		

4.2 Technical and architecture

ID	Requirement	Priority	Response	Vendor comments
T1	Kubernetes-native deployment by Helm chart or operator, with configuration held as Kubernetes custom resources. List each in-cluster component and its CPU, memory and storage requirements.	Must		
T2	In-cluster components that run with the minimum permissions they need. Identify any component that needs privileged or host-level access, and explain why.	Must		
T3	Support for [list Kubernetes distributions and managed Kubernetes services in use], and for the current Kubernetes minor release plus at least two prior minor releases. State how soon new upstream releases are supported.	Must		
T4	One console and API to manage backup policies, jobs and restores across all our clusters (on-premises, cloud and edge). State whether the management plane is self-hosted, vendor-hosted, or available as both.	Must		
T5	Support for [number] clusters, [number] namespaces, [number] persistent volumes and [capacity] of protected data, with parallel backup and restore and documented throughput limits.	Must		
T6	Incremental backups after the first full copy, with deduplication and compression of exported data. State whether incremental backups use changed block tracking from the storage system or CSI driver.	Must		
T7	Recovery of the backup platform itself: if the cluster hosting the backup software is lost, we can install it in a new cluster and reconnect to existing backups and their catalog.	Must		
T8	Installation and operation in air-gapped environments, using our private container registry.	Nice		

4.3 Integration

ID	Requirement	Priority	Response	Vendor comments
I1	Single sign-on to the console and API via OIDC or SAML 2.0 with our identity provider, mapping identity-provider groups to roles.	Must		
I2	Validated snapshot and restore support for [list CSI drivers and storage platforms in use], with a published support matrix.	Must		
I3	Backup targets including S3-compatible object storage, public cloud object storage and NFS [list the targets you use].	Must		
I4	Backup policies, schedules and restore definitions expressed as Kubernetes custom resources or declarative files that we can keep in Git and apply through our GitOps and CI/CD tooling.	Must		
I5	A documented REST API and command-line interface covering every function available in the console.	Must		
I6	Metrics in a Prometheus-compatible format, and alerts for failed, missed or late jobs and unprotected workloads by email, webhook and our ticketing (ITSM) and incident tools.	Must		
I7	Streaming of job, audit and security event logs to our SIEM in a documented format.	Must		
I8	Infrastructure-as-code support (for example, a Terraform provider) for platform configuration and cluster onboarding.	Nice		

4.4 Security and compliance

ID	Requirement	Priority	Response	Vendor comments
S1	Role-based access aligned with Kubernetes namespaces and RBAC, so users can only see, back up and restore what they are authorized for. Include a read-only role for auditors.	Must		
S2	Immutable backup copies using object lock (WORM) on supported targets. Explain how the platform's retention settings interact with the storage-level lock.	Must		
S3	At least one isolated copy of each backup, held in a separate account or logically air-gapped target with separate credentials, so compromised cluster or Kubernetes administrator credentials cannot delete it.	Must		