KrevoPilot Agent

Kubernetes RBAC, Data Collection and Customer Controls

A buyer-ready security reference for platform engineering, DevOps and information-security reviews.

Read-focused standard mode

The Helm chart's standard collector role contains only **get** and **list** permissions. Optional pod-log access adds **get** only. It has no exec, attach, port-forward or impersonation permission. Advanced telemetry, diagnostic pods and server-side dry-run validation are separately gated capabilities described in this document.

One outbound agent

A lightweight in-cluster agent sends privacy-filtered telemetry to the configured KrevoPilot API. The SaaS service does not require customer kubeconfig credentials.

Explicit collection

Collection is bounded by Helm feature flags, supported resource types, item limits and payload-size limits. Logs, advanced node telemetry and active diagnostics are off by default.

No autonomous changes

Standard operation collects evidence and produces review-first recommendations. Optional diagnostics are user-triggered; optional write validation uses Kubernetes server-side dry-run and does not persist a change.

Customer-controlled

Logs, manifests, Secret key metadata, readable object names, advanced metrics and diagnostics can be controlled independently. Enterprises can supply custom RBAC.

Default runtime posture

The chart runs one non-root pod with a read-only root filesystem, RuntimeDefault seccomp, no privilege escalation and all Linux capabilities dropped. Default resources are 50m CPU / 96Mi memory requested and 250m CPU / 256Mi memory limited.

Scope note

The default chart uses a ClusterRole because Kubernetes health, node capacity, cross-namespace application mapping and cluster-scoped storage/admission resources are part of the product. Customers requiring namespace-only access can disable automatic RBAC creation and bind a custom least-privilege role; cluster-wide features will then be reduced accordingly.

1. Core telemetry permissions

These permissions support the baseline health, inventory, metrics and ownership model. ReplicaSets are used to resolve pod ownership back to Deployments.

Kubernetes resource	Verbs	Information accessed	Why KrevoPilot uses it	Customer control
core/pods	get, list	Pod phase, readiness, restart state, owner references, resource requests/limits and sanitized runtime status.	Application health, pod readiness, ownership mapping and investigation targets.	Baseline. Remove only with custom RBAC; application visibility will be substantially reduced.
core/events	get, list	Kubernetes event reason, involved object, count and message according to event-message privacy mode.	Failure timelines and evidence such as FailedMount, BackOff and scheduling errors.	Set event message mode to scrubbed, summary or none. Fully remove with custom RBAC.
core/nodes	get, list	Node readiness, conditions, capacity/allocatable values and runtime/provider metadata.	Cluster capacity, node health and platform detection evidence.	Remove with custom RBAC; node and capacity views will be unavailable.
core/namespaces	get, list	Namespace names or aliases, depending on privacy settings.	Namespace filtering, tenancy context and application grouping.	Set preserveNamespaces=false to alias names; remove access only with custom RBAC.
metrics.k8s.io/pods	get, list	Current pod CPU and memory usage from Metrics Server.	Optimization history, utilization evidence and resource recommendations.	Remove with custom RBAC or omit Metrics Server; optimization metrics become unavailable.
metrics.k8s.io/nodes	get, list	Current node CPU and memory usage.	Cluster-capacity and pressure context.	Remove with custom RBAC; node usage views become unavailable.
apps/replicasets	get, list	ReplicaSet metadata and owner references.	Resolves pod -> ReplicaSet -> Deployment without collecting ReplicaSet manifests.	Remove with custom RBAC; Deployment ownership mapping may be incomplete.

Important log behavior

Regular cluster snapshots do not include pod logs. When logs are enabled, the conditional **pods/log get** rule supports bounded on-demand requests (500 lines and 60 minutes by default), followed by privacy filtering. A customer-owned Loki endpoint can be queried from inside the cluster instead of using the Kubernetes pod-log API.

2. Manifest and application-context permissions

These rules are included when manifests.enabled=true (the current chart default). They provide redacted topology and configuration context for YAML review, investigations and deployment guidance.

Kubernetes resource	Verbs	Information accessed	Why KrevoPilot uses it	Customer control
apps/deployments	get, list	Redacted workload spec, labels, selectors, pod template, probes, ports and resource configuration.	Exact application context, topology, YAML review and safe-fix guidance.	Set manifests.enabled=false or use custom RBAC.
apps/statefulsets	get, list	Redacted StatefulSet spec and pod template.	Stateful application ownership, storage and investigation context.	Set manifests.enabled=false or use custom RBAC.
apps/daemonsets	get, list	Redacted DaemonSet spec and pod template.	Node-level application mapping and investigation context.	Set manifests.enabled=false or use custom RBAC.
batch/jobs, cronjobs	get, list	Job/CronJob specs, schedules, owners and pod templates.	Batch workload inventory and failure diagnosis.	Set manifests.enabled=false or use custom RBAC.
core/services	get, list	Service selectors, ports, targetPorts and type.	Application topology and selector/port mismatch diagnosis.	Set manifests.enabled=false or remove this resource through custom RBAC.
core/endpoints	get, list	Service backend addresses and ports.	Legacy endpoint availability evidence.	Set manifests.enabled=false or use custom RBAC.
discovery.k8s.io/endpointslices	get, list	Endpoint readiness, addresses, ports and Service association.	Modern Service-backend mapping and connectivity diagnosis.	Set manifests.enabled=false or use custom RBAC.
networking.k8s.io/ingresses	get, list	Hosts, paths, backends and TLS reference names.	Ingress -> Service -> endpoint topology and routing diagnosis.	Set manifests.enabled=false or use custom RBAC.
networking.k8s.io/networkpolicies	get, list	Policy selectors, ingress/egress rules and namespace scope.	Connectivity and policy-block evidence.	Set manifests.enabled=false or use custom RBAC.
core/configmaps	get, list	Metadata and key names by default; values only when explicitly enabled.	Missing reference/key and runtime configuration diagnosis.	Set manifests.configMapValues=false (default), disable manifests, or remove through custom RBAC.
core/secrets	get, list	Metadata, type and key names only. Secret values are never collected.	Diagnoses missing Secret references and missing keys without exposing credentials.	Set manifests.secretMetadata=false, disable manifests, or remove through custom RBAC.

3. Storage, scaling, admission and platform permissions

Kubernetes resource	Verbs	Information accessed	Why KrevoPilot uses it	Customer control
core/persistentvolumeclaims	get, list	Claim phase, requested storage, access modes, StorageClass and bound volume reference.	Unbound PVC and application storage diagnosis.	Set manifests.enabled=false or use custom RBAC.
core/persistentvolumes	get, list	Volume phase, capacity, access modes, claim reference and CSI metadata.	Bound-volume and CSI mount investigation context.	Set manifests.enabled=false or remove this cluster-scoped resource with custom RBAC.
storage.k8s.io/storageclasses	get, list	Provisioner, binding mode, reclaim policy and parameters metadata.	Storage provisioning and scheduling context.	Set manifests.enabled=false or use custom RBAC.
storage.k8s.io/csidrivers	get, list	CSI driver capability metadata.	CSI mount/controller failure context.	Set manifests.enabled=false or use custom RBAC.
autoscaling/ horizontalpodautoscalers	get, list	Scale target, min/max replicas and metric targets.	Scaling context and optimization safety checks.	Set manifests.enabled=false or use custom RBAC.
admission/validating webhooks	get, list	Webhook rules, failure policy and Service/client configuration; no Secret values.	Diagnoses unreachable validating admission webhooks.	Set manifests.enabled=false or remove through custom RBAC.
admission/mutating webhooks	get, list	Webhook rules, failure policy and Service/client configuration; no Secret values.	Diagnoses mutating webhook failures and admission blockers.	Set manifests.enabled=false or remove through custom RBAC.
route.openshift.io/routes	get, list	OpenShift Route host, target, TLS and status metadata.	OpenShift platform detection, topology and route diagnosis.	Set manifests.enabled=false or remove for non-OpenShift/custom profiles.

Outbound network behavior

The agent requires outbound HTTPS to the configured KrevoPilot API. If a customer enables customer-owned Loki integration, it additionally contacts the configured in-cluster Loki endpoint. Optional active diagnostics contact only approved namespace-local targets. The optional chart NetworkPolicy can restrict egress to approved CIDRs and DNS; URL-level restriction is not possible with standard Kubernetes NetworkPolicy.

4. Optional capability envelopes

These capabilities are disabled unless a customer deliberately enables the feature and grants its separate permission envelope. Turning on a runtime flag does not itself grant Kubernetes access.

Kubernetes resource	Verbs	Information accessed	Why KrevoPilot uses it	Customer control
core/pods/log	get	Current and optional previous container logs for a selected pod/container; bounded by line and lookback limits.	User-requested investigation evidence. Normal snapshots never contain pod logs.	Keep logs.enabled=false (default), or remove the conditional rule. Customer-owned Loki can be used instead.
core/nodes/proxy	get	Bounded kubelet/cAdvisor and summary responses for a limited number of nodes.	Optional CPU-throttling, pod-network error/drop and PVC-filesystem fullness evidence.	Keep advancedMetrics.enabled=false (default). When enabled: 300s interval, 10 nodes, 5s timeout and 4 MB response cap by default.
core/pods (diagnostics)	create, get, delete	Only a short-lived, bounded diagnostic pod in the target namespace; no Secret values, volume mounts or Kubernetes API token inside that pod.	User-triggered namespace-local DNS, TCP, HTTP and TLS verification.	diagnostics.enabled=false by default. Requires a separate customer-approved role; revoke it to make diagnostics impossible.
apps workloads + core/configmaps	patch, update	A proposed patch sent with Kubernetes dryRun=All; the API validates it but does not persist it.	Optional review-time validation of a proposed fix against admission and schema rules.	Not part of the standard Helm collector role. Requires a separate role and WRITE_ACTIONS_ENABLED; no create/delete and never Secrets or RBAC.

Diagnostic-pod containment

The diagnostic pod is non-root, uses a read-only filesystem, has no volumes, receives no Kubernetes API token and is given tight resource and lifetime limits. It accepts no shell command. Kubernetes RBAC cannot restrict pod creation by generated name, so this capability is protected by both a separate role and strict agent-side command validation.

Advanced telemetry limits

Advanced node telemetry is off by default and adds zero node-proxy calls when disabled. When enabled, KrevoPilot reads metrics only; it does not access the kubelet exec, logs, run or port-forward endpoints. Response bytes, request timeout, refresh interval and node count are bounded.

5. Data handling and privacy controls

Sensitive values

Secret values are never collected. Environment values are redacted. ConfigMap values are excluded by default. Log and event text is scrubbed according to the configured privacy profile.

Object identity

Namespaces can remain readable or be aliased. Real Deployment, Service and other object names are enabled by default for actionable commands; customers can replace them with stable HMAC aliases.

Bounded payloads

The chart limits pods, events, manifests, YAML size, log lines and lookback time. The collector prioritizes investigation-relevant context within these limits.

No kubeconfig

The SaaS backend does not receive customer kubeconfig files or direct Kubernetes API credentials. The in-cluster ServiceAccount is the only Kubernetes identity used by the agent.

Information not collected by the standard agent

Kubernetes resource	Verbs	Information accessed	Why KrevoPilot uses it	Customer control
Secret values	not requested	Only Secret metadata/type/key names may be sent when enabled.	Prevents credential disclosure while preserving missing-key diagnosis.	Disable secret metadata entirely if even key names are restricted.
kubectl exec / attach / port-forward	not granted	No process execution, interactive session or tunnel capability.	The agent is an observer, not a remote administration channel.	No action required; permissions are absent.
Persistent changes in standard mode	not granted	The standard collector role has no create, update, patch or delete operations.	Recommendations remain review-first. Optional diagnostic pods and server-side dry-run validation require separate grants.	Keep optional capability roles absent/disabled for an entirely read-only installation.
RBAC identities and tokens	not granted	No ServiceAccount token, RoleBinding or user impersonation collection in the standard role.	Limits identity and credential exposure.	No action required; permissions are absent.
ConfigMap values	off by default	Only key names are sent unless explicitly enabled.	Reduces accidental configuration/data exposure.	Keep manifests.configMapValues=false.
Continuous pod logs	off by default	Logs are not included in normal snapshots.	Limits data volume and exposure.	Keep logs.enabled=false, or use customer-owned Loki for historical querying.

6. Recommended customer security profiles

The following profiles help a security reviewer choose the right balance between diagnosis depth and data minimization.

Standard observability

manifests.enabled=true; secretMetadata=true; configMapValues=false; realObjectNames=true; logs.enabled=false; diagnostics.enabled=false. Best default for actionable topology and investigations.

Privacy focused

Preserve namespaces off, real object names off, Secret metadata off, ConfigMap values off, logs off and diagnostics off. Useful where identifiers are considered sensitive.

Investigation enabled

Standard profile plus bounded logs and, if approved, the separate diagnostic-pod permission envelope. Review log storage, masks, CIDR allow-list and incident workflow first.

Strict custom RBAC

Set rbac.create=false and provide an organization-approved Role/ClusterRole. Remove unused resource families and accept that related UI and investigation evidence will show as unavailable.

What changes when a permission is removed?

KrevoPilot should degrade transparently rather than invent evidence. Missing permissions can reduce or remove the corresponding feature: no pod metrics means no utilization-based optimization; no Services/EndpointSlices means weaker traffic-path diagnosis; no workload manifests means no exact YAML context; no webhook resources means admission-webhook root cause cannot be fully traced. The agent itself remains operational for whatever resources are still permitted.

Security-review recommendation	Review the rendered ClusterRole before installation, start with the standard read-focused profile, and enable optional permission envelopes only after a separate review. Document removals because they intentionally limit evidence and product coverage. Do not add mutation verbs to the standard collector role.

Version basis

This document reflects KrevoPilot agent image 2.0.43 and Helm chart 0.1.44. Customers should review the ClusterRole generated by the exact chart version they plan to install because conditional permissions and optional features can evolve.

Contact

KreateRevo UG, Berlin, Germany | https://krevopilot.com