

# Kubernetes Icons

v0.1

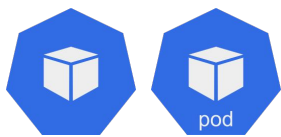


- Set of icons to create diagrams
- Use them as SDK, feel free to create
- Non official library
- Kubernetes official blue color
  - #326ce5
  - RGB(50,108,229)
- Maintainers:
  - Arnaud Mazin <[amazin@octo.com](mailto:amazin@octo.com)>
  - Etienne Coutaud <[ecoutaud@octo.com](mailto:ecoutaud@octo.com)>

# Table of contents

- |                               |                                |
|-------------------------------|--------------------------------|
| 1...Compute                   | 10...Group and links           |
| 2...Storage                   | 11...Kubernetes ressources map |
| 3...Network                   | 12...Diagrams examples         |
| 4...RBAC Model                |                                |
| 5...Pods Configuration        |                                |
| 6...Cluster Configuration     |                                |
| 7...Others                    |                                |
| 8...Infrastructure components |                                |
| 9...Control Plane components  |                                |

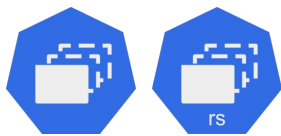
# Compute



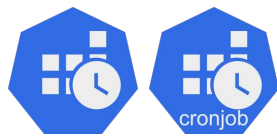
**Pod:** Pod is a collection of containers that can run on a host. This resource is created by clients and scheduled onto hosts.



**Job:** Job represents the configuration of a single job.



**ReplicaSet:** ReplicaSet ensures that a specified number of pod replicas are running at any given time.



**CronJob:** A CronJob manages time based Job, namely:

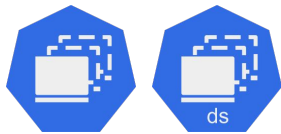
- once at a specified point in time
- repeatedly at a specified point in time



**Deployment:** Deployment enables declarative updates for Pods and ReplicaSets.

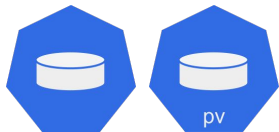


**StatefulSet:** StatefulSet represents a set of pods with consistent identities. Identities are defined as: network, storage.



**DaemonSet:** DaemonSet represents the configuration of a daemon set.

# Storage



**PersistentVolume:** is a storage resource provisioned by an administrator.

---



**PersistentVolumeClaim:** PersistentVolumeClaim is a user's request for and claim to a persistent volume.

---



**StorageClass:** StorageClass describes the parameters for a class of storage for which PersistentVolumes can be dynamically provisioned.

# Network



**Ingress:** Ingress is a collection of rules that allow inbound connections to reach the endpoints defined by a backend. An Ingress can be configured to give services externally-reachable urls, load balance traffic, terminate SSL, offer name based virtual hosting etc.

---



**Service:** Service is a named abstraction of software service (for example, mysql) consisting of local port (for example 3306) that the proxy listens on, and the selector that determines which pods will answer requests sent through the proxy.

---



**Endpoint:** Endpoints is a collection of endpoints that implement the actual service.

---



**NetworkPolicy:** NetworkPolicy describes what network traffic is allowed for a set of Pods.

# RBAC model



**ServiceAccount:** binds together: a name, a principal that can be authenticated and authorized \* a set of secrets.



**ClusterRole:** ClusterRole is a cluster level, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding or ClusterRoleBinding.



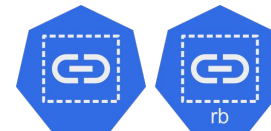
**User:** Human user of Kubernetes cluster.



**ClusterRoleBinding:** A cluster role binding grants the permissions defined in a role/clusterrole to a user or set of users. Permissions are granted cluster-wide.



**Group:** Set of Service Accounts or Users.



**RoleBinding:** A role binding grants the permissions defined in a role/clusterrole to a user or set of users. Permissions are granted within a namespace.



**Role:** Role is a namespaced, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding.

# Pods Configuration

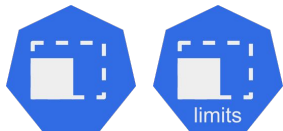


**ConfigMap:** ConfigMap holds configuration data for pods to consume.



**Secret:** Secret holds secret data of a certain type.

# Cluster configuration



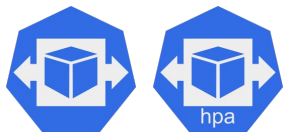
**LimitRange:** LimitRange sets resource usage limits for each kind of resource in a Namespace.

---



**Quota:** ResourceQuota sets aggregate quota restrictions enforced per namespace.

---



**HorizontalPodAutoscaler:** configuration of a horizontal pod autoscaler.



# Others



**CustomResourceDefinition:** Extension of Kubernetes API.



**PodSecurityPolicy:** governs the ability to make requests that affect the Security Context that will be applied to a pod and container.

# Infrastructure components



**Cluster:** Kubernetes cluster.



**Master:** Kubernetes Control Plane.



**Node:** Worker machine in Kubernetes cluster.



**ETCD:** Kubernetes's backing store.

# Control Plane components



**K8s API Server:** Kubernetes API.



**Kubelet:** The kubelet is the primary “node agent” that runs on each node.



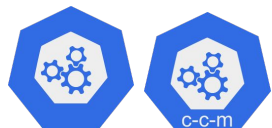
**Controller Manager:** Kubernetes controller manager.



**Kube-proxy:** The Kubernetes network proxy runs on each node. This reflects services as defined in the Kubernetes API on each node.



**Scheduler:** In charge of ensuring Pods placement.

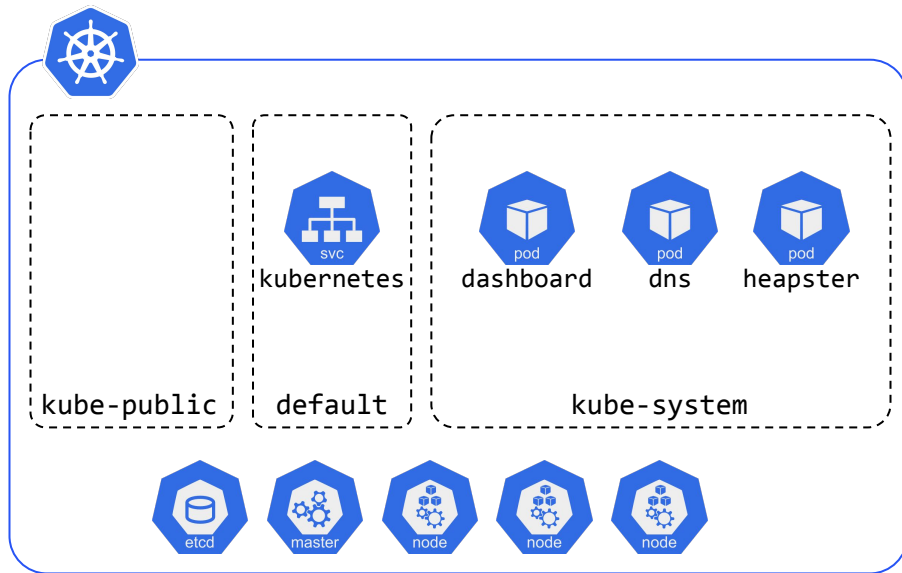


**Cloud Controller Manager:** Optional and External Cloud controller (experimental).

# Groups and links



**Namespace:** Namespace provides a scope for Names. Use of multiple namespaces is optional.

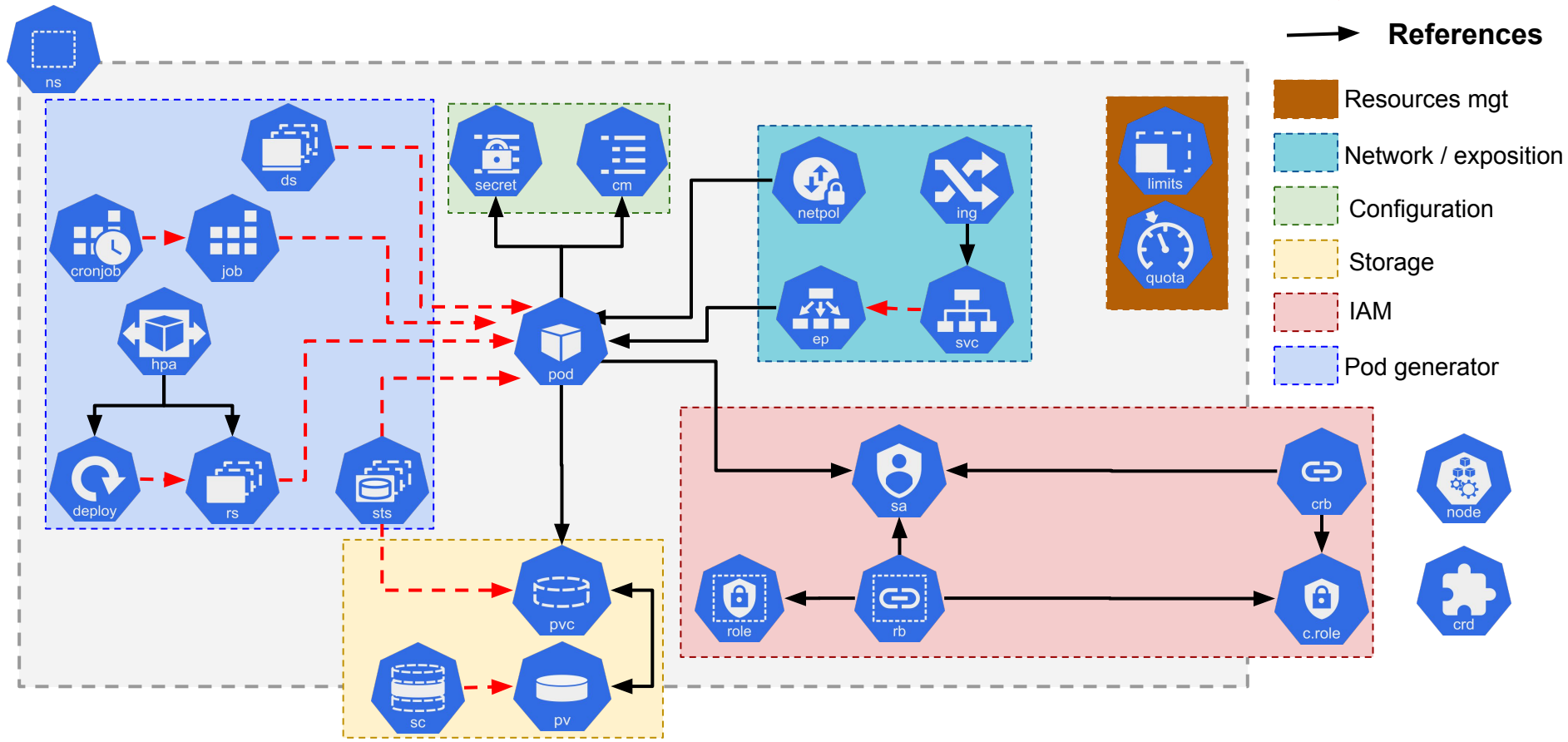


Use to represent a reference between components, reference can be through various selector (label, name ...)



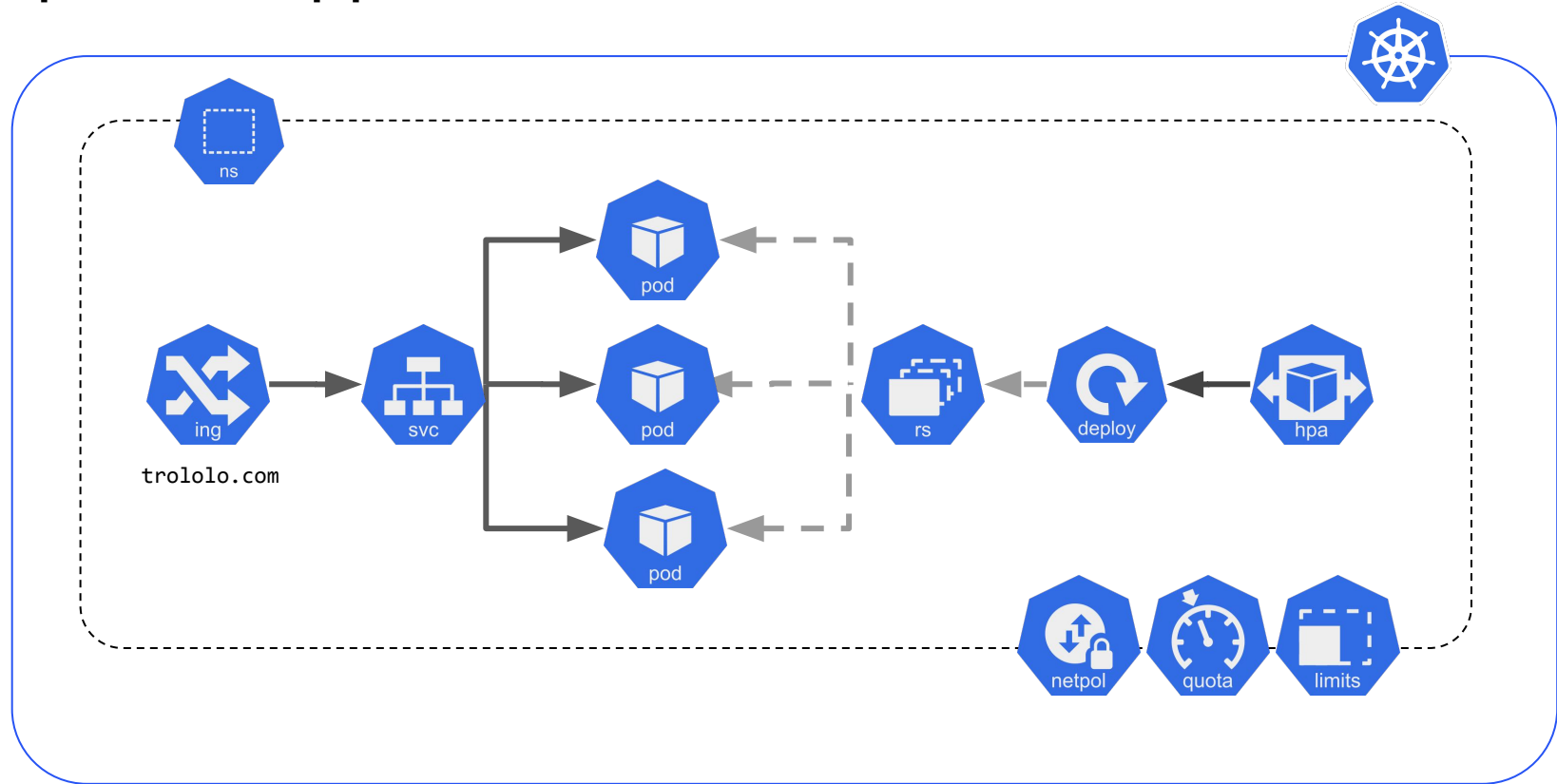
Use to represent a generation, resource generate other resource

# Kubernetes Ressources Map

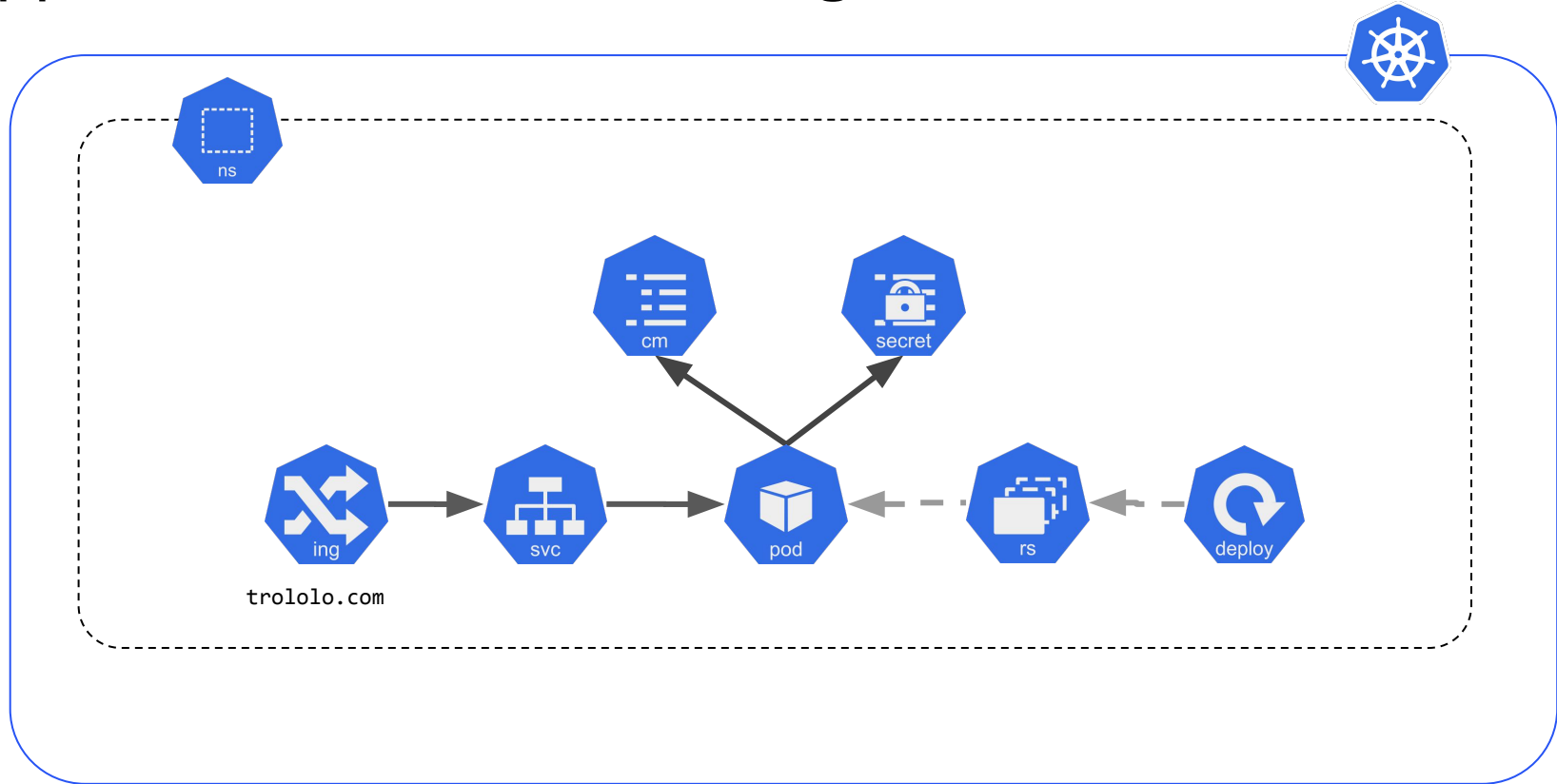


**Diagrams examples**

# Exposed Application

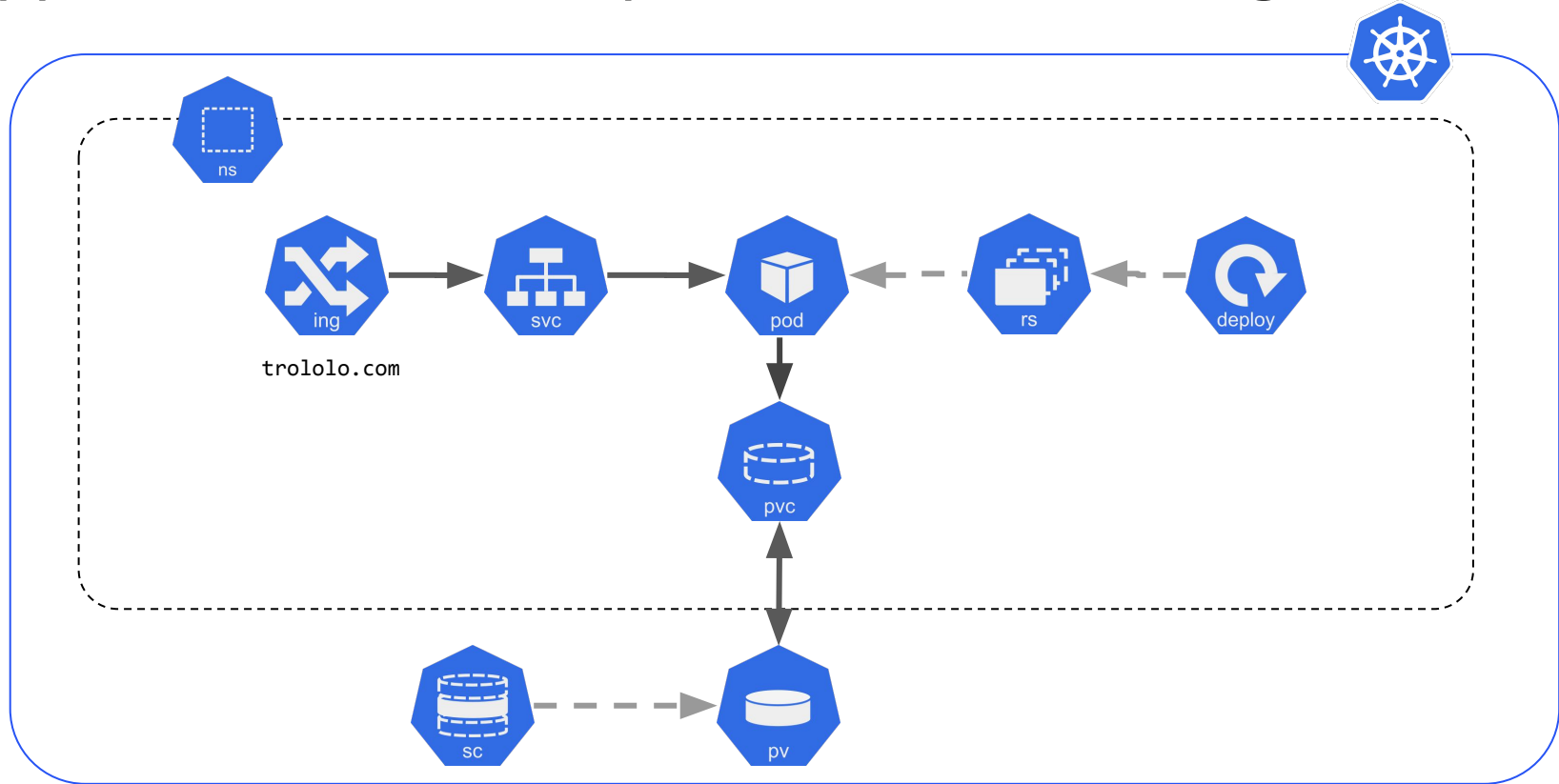


# Application with configuration

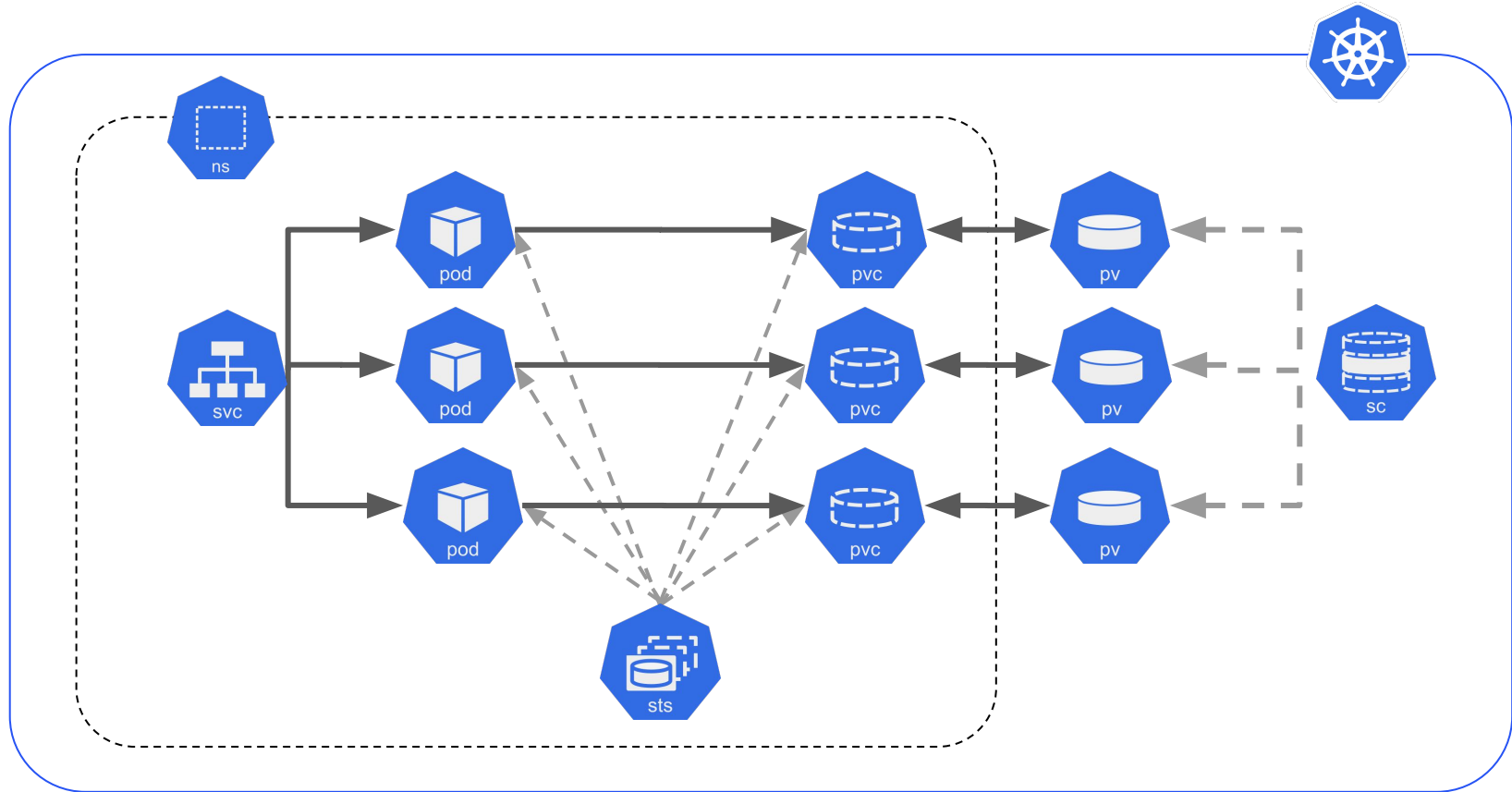




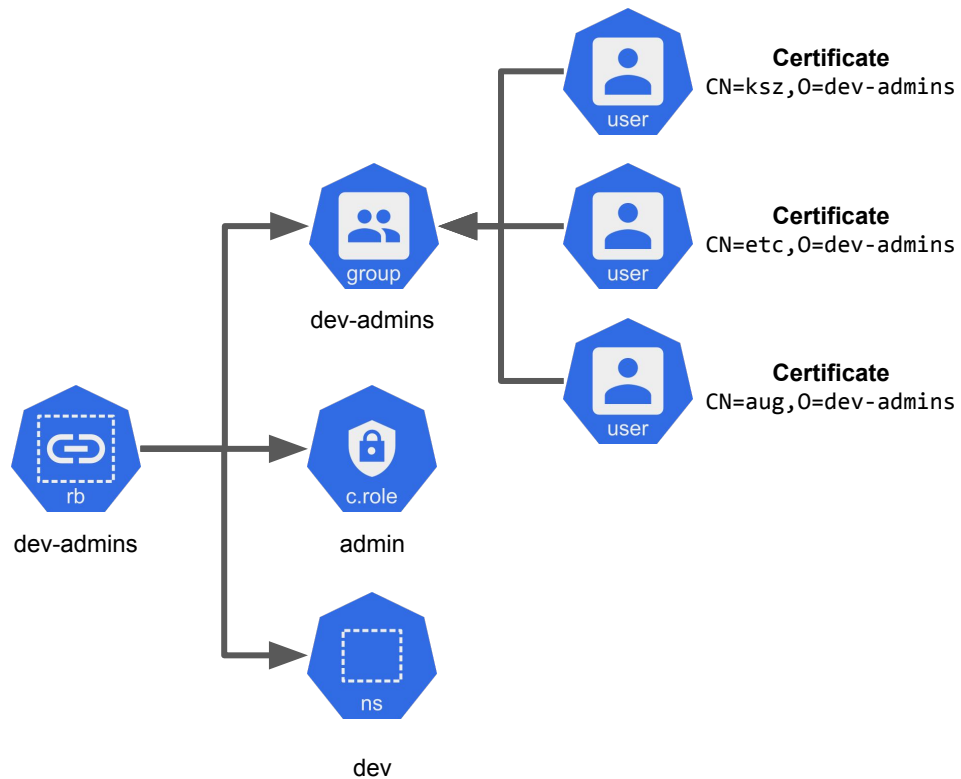
# Application with persistent storage



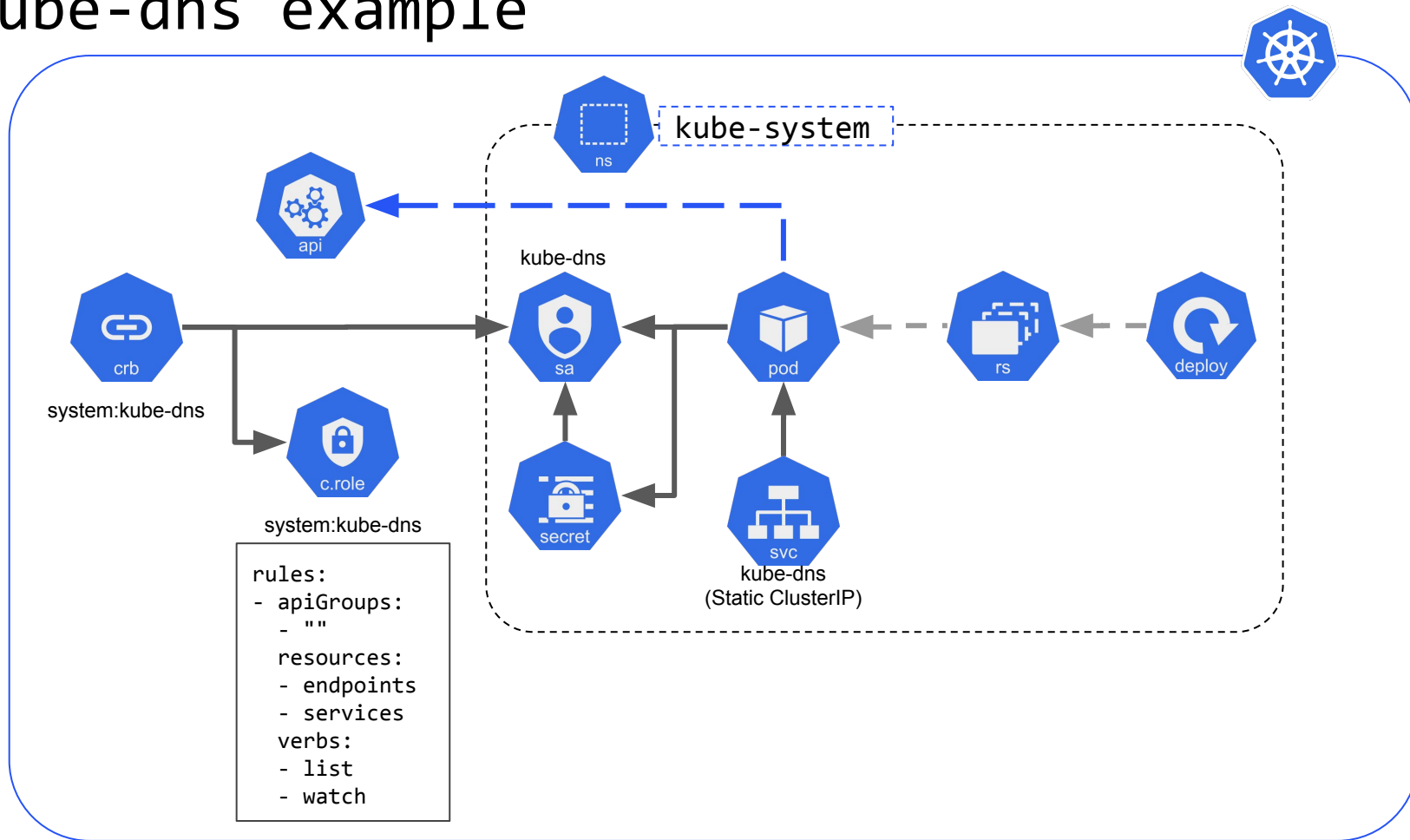
# StatefulSet Architecture



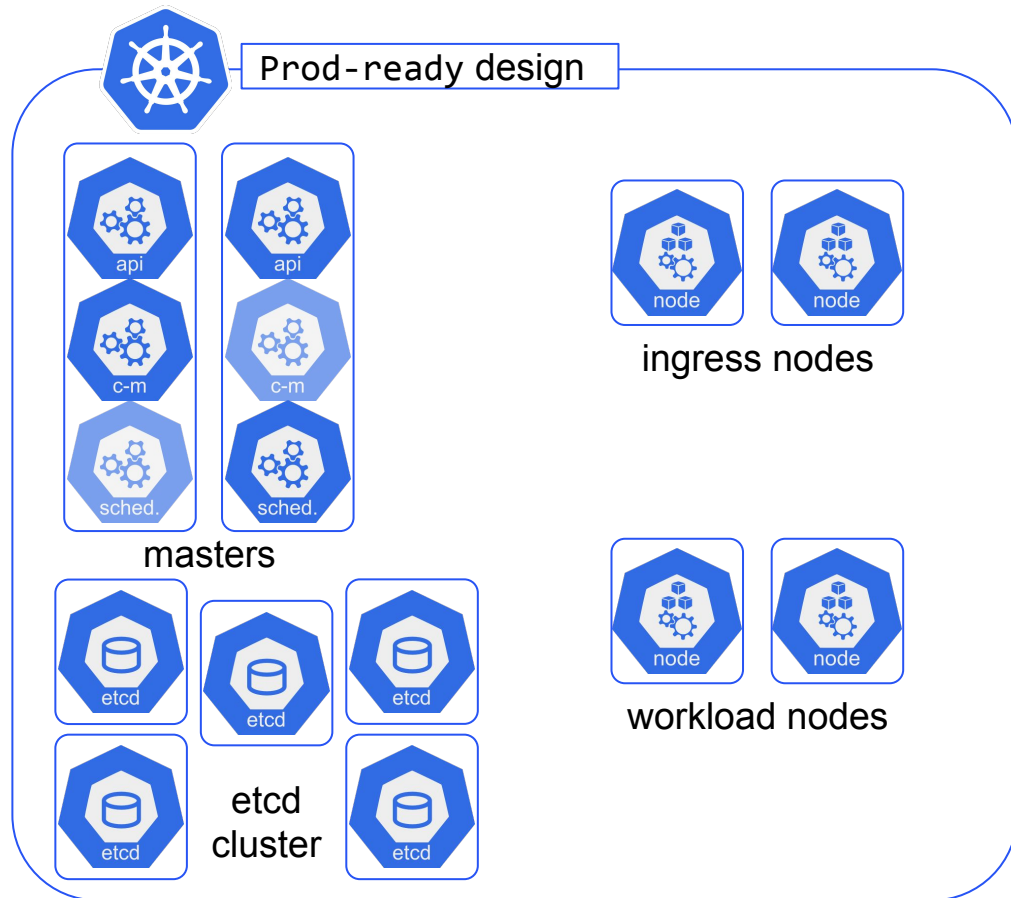
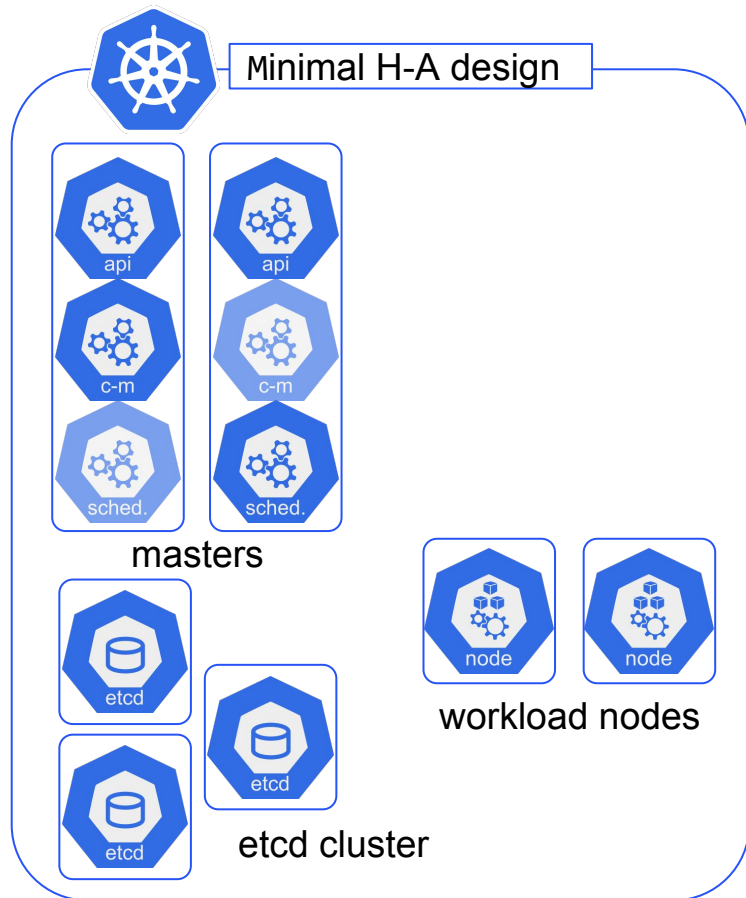
# RBAC model



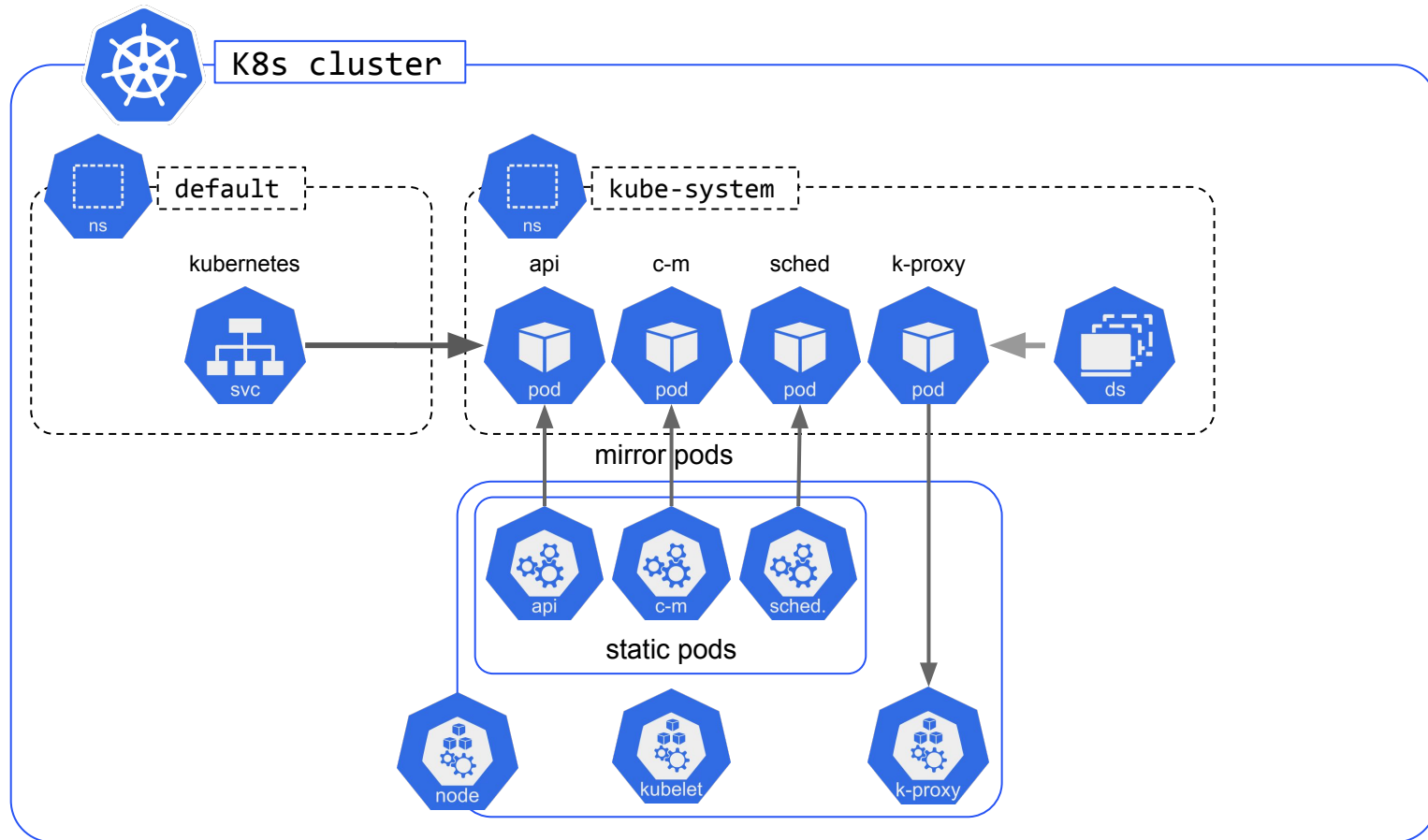
# Kube-dns example



# Server implementation



# K8s components startup



# Appendices

