Here is a comprehensive list of Kubernetes resource types (kinds) with their API versions, along with a brief explanation of each resource type and why we use it:

* v1
  + APIVersions: Represents the available API versions in the Kubernetes cluster.
  + ComponentStatus: Provides information about the health of various cluster components.
  + ConfigMap: Stores configuration data as key-value pairs.
  + Endpoints: Exposes network endpoints for a service.
  + Event: Captures events related to objects in the cluster.
  + Namespace: Provides a logical separation of resources within a cluster.
  + Node: Represents a worker node in the cluster.
  + PersistentVolume: Represents a storage volume in the cluster.
  + PersistentVolumeClaim: Requests storage resources from a PersistentVolume.
  + Pod: Represents a running instance of a containerized application.
  + PodTemplate: Defines a template for creating Pods.
  + ReplicationController: Ensures a specified number of Pod replicas are running.
  + ResourceQuota: Limits resource consumption within a namespace.
  + Secret: Stores sensitive information, such as passwords or API keys.
  + Service: Exposes an application running on a set of Pods.
  + ServiceAccount: Provides an identity for Pods to access cluster resources.
* apps/v1
  + ControllerRevision: Represents a specific revision of a Deployment or StatefulSet.
  + Deployment: Manages the deployment and scaling of application Pods.
  + ReplicaSet: Ensures a specified number of Pod replicas are running.
  + StatefulSet: Manages the deployment and scaling of stateful applications.
* batch/v1
  + Job: Runs a task to completion, ensuring that a specified number of completions are successful.
  + CronJob: Runs Jobs on a schedule defined in Cron format.
* policy/v1
  + PodDisruptionBudget: Ensures the availability of a specified number of Pods during disruptions.
* rbac.authorization.k8s.io/v1
  + ClusterRoleBinding: Binds ClusterRoles to subjects within the cluster.
  + ClusterRole: Defines a set of permissions for cluster-level access.
  + RoleBinding: Binds Roles to subjects within a namespace.
  + Role: Defines a set of permissions for namespace-level access.
* autoscaling/v1
  + HorizontalPodAutoscaler: Automatically scales the number of Pods based on resource utilization.
* networking.k8s.io/v1
  + Ingress: Exposes HTTP and HTTPS routes to services within the cluster.
  + NetworkPolicy: Defines network access policies between Pods.
* storage.k8s.io/v1
  + StorageClass: Defines the class of storage that can be requested by PersistentVolumeClaims.
  + VolumeAttachment: Attaches a PersistentVolume to a node.
* certificates.k8s.io/v1beta1
  + CertificateSigningRequest: Represents a request for a signed certificate.
* coordination.k8s.io/v1
  + Lease: Used to coordinate actions between different components.