

Here is a short explanation with real-world examples 👇  
  
We will look at Kubernetes native objects in two parts. Here is part 01.  
  
- 𝗣𝗼𝗱: Deployment Unit - Running a single-instance, e.g., Nginx web server  
  
- 𝗦𝗲𝗿𝘃𝗶𝗰𝗲: Networking - Exposing a set of pods to other pods within the cluster. e.g., Exposing a set of Redis server pods.  
  
- 𝗩𝗼𝗹𝘂𝗺𝗲: Storage - Storing database files for a MySQL server running in a pod.  
  
- 𝗡𝗮𝗺𝗲𝘀𝗽𝗮𝗰𝗲: Workload Isolation - Segregating apps/teams/projects in a dedicated/shared cluster. e.g., Different namespaces for apps or stages like development, testing, and production.  
  
- 𝗥𝗲𝗽𝗹𝗶𝗰𝗮𝗦𝗲𝘁: Replication - Running five replicas of a web server application. e.g. Nginx server with multiple replicas for load balancing.  
  
- 𝗗𝗲𝗽𝗹𝗼𝘆𝗺𝗲𝗻𝘁: Management - Uses Replicaset + Rolling out a new version of a web server application. e.g. Upgrading from Nginx version 1.19 to 1.20.  
  
- 𝗦𝘁𝗮𝘁𝗲𝗳𝘂𝗹𝗦𝗲𝘁: State Management - Scaling a distributed database like Cassandra. e.g. Cassandra cluster with multiple nodes.  
  
- 𝗗𝗮𝗲𝗺𝗼𝗻𝗦𝗲𝘁: Node Operation - Running a log collection daemon on every node. e.g. Fluentd or Filebeat for log collection.  
  
- 𝗝𝗼𝗯: Task Execution - Processing a large compute job using several workers. e.g. A data processing job using Apache Spark.  
  
- 𝗖𝗿𝗼𝗻𝗝𝗼𝗯: Scheduled Tasks - Running a batch job at specific times. e.g. A nightly backup job.  
  
- 𝗦𝗲𝗰𝗿𝗲𝘁: Sensitive Data - Storing the password for a database. e.g. MongoDB password.  
  
- 𝗖𝗼𝗻𝗳𝗶𝗴𝗠𝗮𝗽: Configuration - Storing the configuration for a web server. e.g. Nginx configuration file.  
  
- 𝗜𝗻𝗴𝗿𝗲𝘀𝘀: External Access - Exposing a web application to the internet. e.g. A web application running on Apache.  
  
- 𝗡𝗲𝘁𝘄𝗼𝗿𝗸𝗣𝗼𝗹𝗶𝗰𝘆: Network Rules - Defining how pods communicate with each other. e.g. Allowing traffic from a specific IP range or bewtween namespace ot pods with specific labels.  
  
- 𝗛𝗼𝗿𝗶𝘇𝗼𝗻𝘁𝗮𝗹 𝗣𝗼𝗱 𝗔𝘂𝘁𝗼𝘀𝗰𝗮𝗹𝗲𝗿 (𝗛𝗣𝗔): Scalability - Automatically scaling a web server application based on CPU usage. e.g. An auto-scaling Nginx deployment.  
  
- 𝗣𝗲𝗿𝘀𝗶𝘀𝘁𝗲𝗻𝘁𝗩𝗼𝗹𝘂𝗺𝗲 (𝗣𝗩): Persistent Storage - Providing a file system for a MongoDB database pod.  
  
- 𝗣𝗲𝗿𝘀𝗶𝘀𝘁𝗲𝗻𝘁𝗩𝗼𝗹𝘂𝗺𝗲𝗖𝗹𝗮𝗶𝗺 (𝗣𝗩𝗖): Storage Request - Requesting storage for a PostgreSQL database pod.  
  
- 𝗘𝗻𝗱𝗽𝗼𝗶𝗻𝘁𝘀𝗹𝗶𝗰𝗲𝘀: Network endpoint Points - Storing IP addresses for a service. e.g., IP addresses of pods running an Nginx server.  
  
- 𝗦𝗲𝗿𝘃𝗶𝗰𝗲𝗔𝗰𝗰𝗼𝘂𝗻𝘁: Authentication - Giving a pod the necessary permissions to interact with the Kubernetes API.  
  
- 𝗥𝗼𝗹𝗲/𝗖𝗹𝘂𝘀𝘁𝗲𝗿𝗥𝗼𝗹𝗲: Authorization - Granting read access to pods in a specific namespace.