Kubernetes Components with YAML Examples

1. Pod

Description: A Pod is the smallest and simplest Kubernetes object. It represents a single instance of a running process in your cluster.

Use: Used to run a single container or multiple tightly coupled containers.

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
  - name: my-container
    image: nginx
```

2. Deployment

Description: A Deployment provides declarative updates for Pods and ReplicaSets.

Use: Used for managing stateless applications and scaling Pods.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
      - name: my-container
        image: nginx
```

3. Service

Description: A Service is an abstraction that defines a logical set of Pods and a policy by which to access them.

Use: Used to expose a set of Pods as a network service.

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: myapp
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
```

4. Ingress

Description: Ingress manages external access to the services in a cluster, typically HTTP.

Use: Used for load balancing and SSL termination.

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: myapp.example.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: my-service
            port:
              number: 80
```

5. ConfigMap

Description: ConfigMap allows you to decouple environment-specific configuration from your container images.

Use: Used to inject environment variables and configuration files.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-config
data:
  APP_MODE: production
  APP_DEBUG: "false"
```

6. Secret

Description: Secret is used to store and manage sensitive information, such as passwords, OAuth tokens, and ssh keys.

Use: Used to store sensitive data securely.

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
type: Opaque
data:
  password: cGFzc3dvcmQ=  # base64 encoded
```

7. Namespace

Description: Namespace provides a way to divide cluster resources between multiple users.

Use: Used to organize cluster resources.

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-namespace
```

8. PersistentVolume

Description: PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator.

Use: Used to provide storage resources to Pods.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: my-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /mnt/data
```

9. PersistentVolumeClaim

Description: PersistentVolumeClaim (PVC) is a request for storage by a user.

Use: Used to request storage defined by a PersistentVolume.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Mi
```

10. StatefulSet

Description: StatefulSet is used to manage stateful applications.

Use: Used when applications require persistent storage and stable network identities.

```
apiVersion: apps/v1
```

```
kind: StatefulSet
metadata:
  name: my-statefulset
spec:
  serviceName: "my-service"
  replicas: 2
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
      - name: my-container
        image: nginx
        volumeMounts:
        - name: data
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: data
    spec:
      accessModes: [ "ReadWriteOnce" ]
      resources:
        requests:
          storage: 1Gi
```

11. HorizontalPodAutoscaler

Description: Automatically scales the number of pods in a deployment depending on CPU utilization or other select metrics.

Use: Used to auto-scale applications based on load.

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: my-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
```

```
    kind: Deployment
    name: my-deployment
  minReplicas: 1
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

12. Job

Description: Job creates one or more pods and ensures that a specified number of them successfully terminate.

Use: Used for batch and one-time tasks.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: my-job
spec:
  template:
    spec:
      containers:
      - name: my-container
        image: busybox
        command: ["echo", "Hello from Kubernetes Job!"]
      restartPolicy: Never
```

13. DaemonSet

Description: A DaemonSet ensures that all (or some) Nodes run a copy of a Pod.

Use: Used for running background tasks on all nodes.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: my-daemonset
spec:
```

```
  selector:
    matchLabels:
      app: my-daemon
  template:
    metadata:
      labels:
        app: my-daemon
    spec:
      containers:
      - name: my-container
        image: busybox
           command: ["sh", "-c", "while true; do echo Hello from DaemonSet; sleep 10; done"]
```