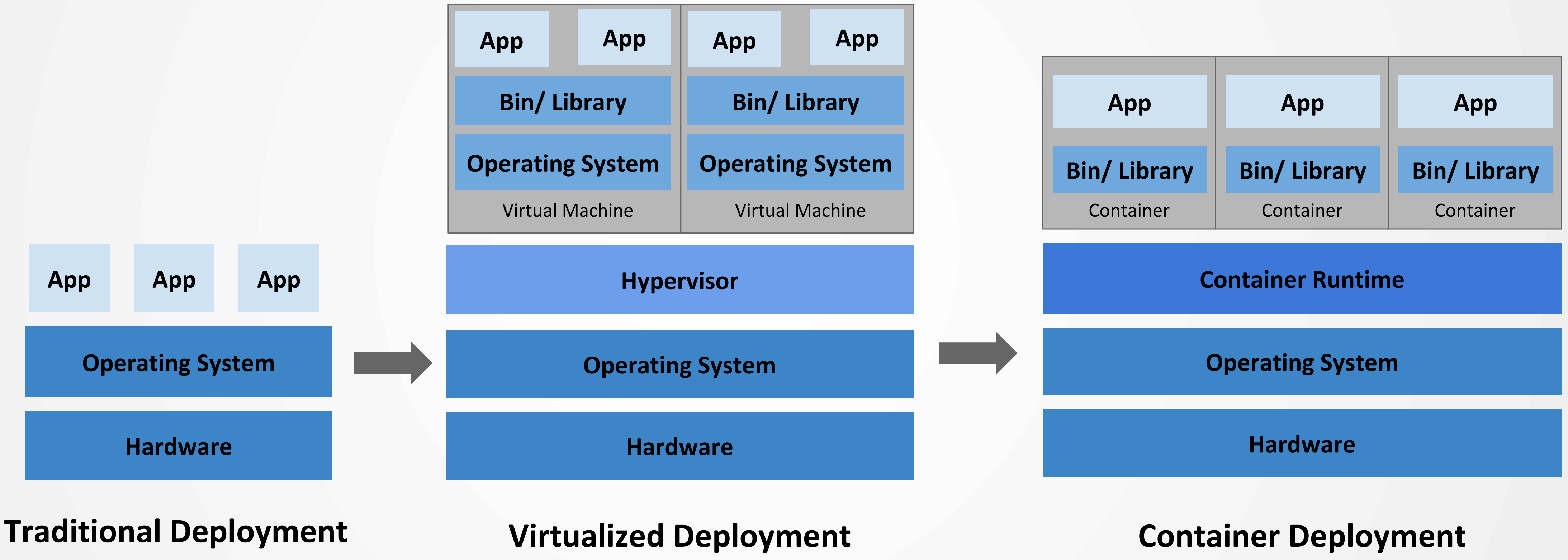
kubernetes

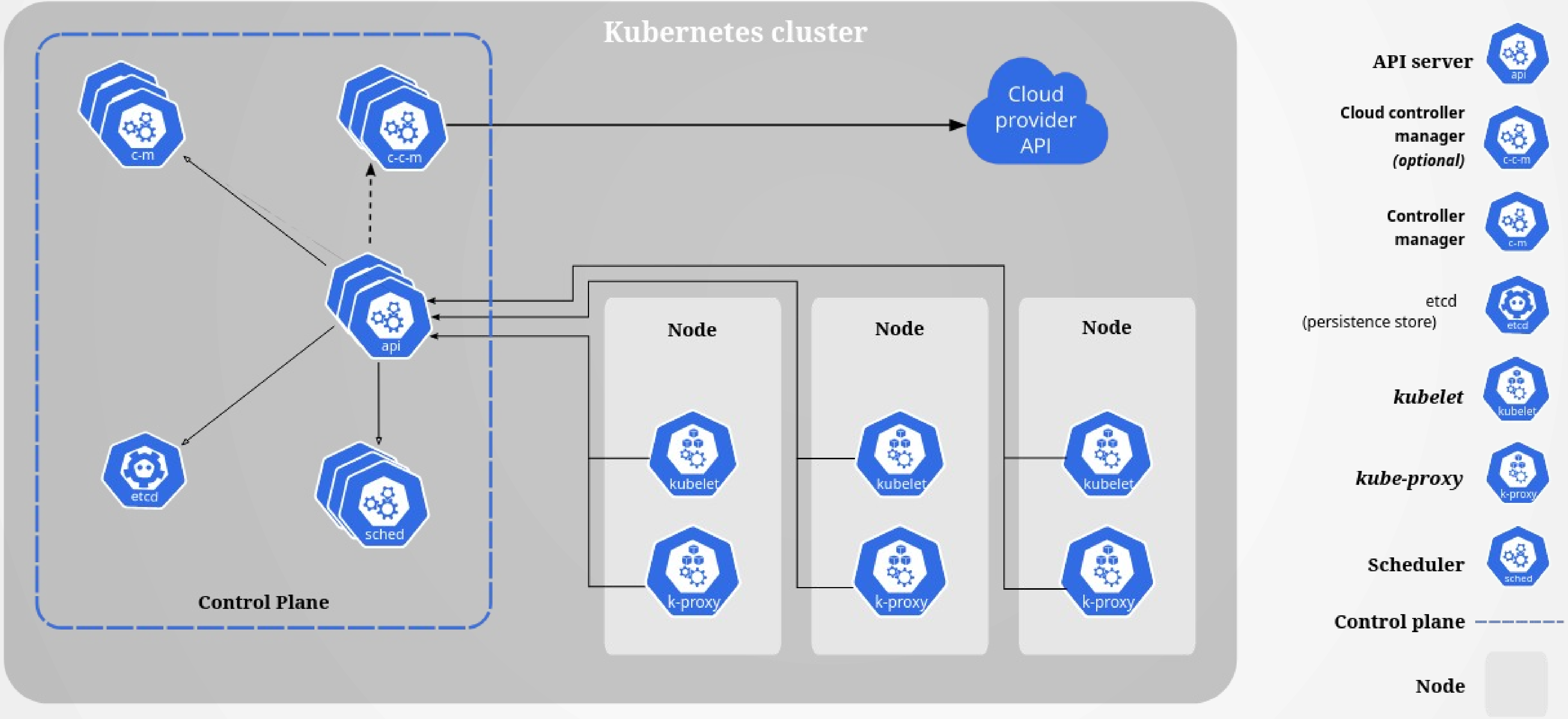
Honza Dražil

Seznam.cz

Vývoj


App
App
App
Operating System
Hardware
Traditional Deployment
App
App
App
App
Bin/ Library
Bin/ Library
Operating System
Operating System
Virtual Machine
Virtual Machine
Hypervisor
Operating System
Hardware
Virtualized Deployment
App
App
App
Bin/ Library
Bin/ Library
Bin/ Library
Container
Container
Container
Container Runtime
Operating System
Hardware
Container Deployment

Kubernetes



Kubernetes cluster
c-m
c-c-m
Cloud provider API
api
etcd
sched
Control Plane
Node
Node
Node
kubelet
kubelet
kubelet
k-proxy
k-proxy
k-proxy
API server
api
Cloud controller manager (optional)
c-c-m
Controller manager
c-m
etcd (persistence store)
etcd
kubelet
kubelet
kube-proxy
k-proxy
Scheduler
sched
Control plane
Node

Základní Workloads

Namespace

- Umožňuje logicky oddělit jednotlivé resources v k8s
- Odstraněním namespace se odstraní i vše v něm uložené
- Práva v k8s je možné nastavit na namespace

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: production
```

Pod

- Nejmenší možná konfigurace, která umožňuje provést výpočet.
- Obsahuje alespoň jeden kontejner (image).
- Dočasný immutable resource

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: counters
spec:
  containers:
  - name: counter-containerv1
    image: counter:v1
  - name: counter-containerv2
    image: counter:v2
```

Replicaset

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: counters
spec:
  replicas: 3
  selector:
    matchLabels:
      app: counter-rs
  template:
    metadata:
      name: counter-pod
      labels:
        app: counter-rs
    spec:
      containers:
      - name: counter-container
        image: counter:v1
```

Replicaset

- Zajišťuje, že běží nastavený počet podů
- Používá selector k "počítání" podů
- Přímo se často nepoužívá
- Pojmenování podů:
  - `<ReplicaSet>-<PodId>`

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: counters
spec:
  replicas: 3
  selector:
    matchLabels:
      app: counter-rs
  template:
    metadata:
      name: counter-pod
      labels:
        app: counter-rs
    spec:
      containers:
      - name: counter-container
        image: counter:v1
```

Deployment

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: counter-app
spec:
  replicas: 10
  strategy:
    rollingUpdate:
      maxSurge: 3
      maxUnavailable: 1
  selector:
    matchLabels:
      app: cool-counter
  template:
    metadata:
      name: cool-counter
      labels:
        app: cool-counter
    spec:
      containers:
      - name: counter-container
        image: counter:v1
```

Deployment

- Bezstavové aplikace
- Vytváří ReplicatSety, které vlastní pody
- Nová konfigurace aktivuje (ve výchozím stavu) RollingUpdate:
  - Pody v replicasetu se postupně ukončují a nahrazují novou instancí
  - Starý replicaset běží dokud neběží nový replica set
  - Možnost vrátit se ke starému replica setu
    - `kubectl rollout undo deployment <deployment>`

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: counter-app
spec:
  replicas: 10
  strategy:
    rollingUpdate:
      maxSurge: 3
      maxUnavailable: 1
  selector:
    matchLabels:
      app: cool-counter
  template:
    metadata:
      name: cool-counter
      labels:
        app: cool-counter
    spec:
      containers:
      - name: counter-container
        image: counter:v1
```

Konfigurace

Proměnné prostředí

https://kubernetes.io/docs/reference/kubernetes-api/workload-resources/pod-v1/

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: environment-dumper
spec:
  replicas: 1
  selector:
    matchLabels:
      app: environment-dumper
  template:
    metadata:
      name: environment-dumper
      labels:
        app: environment-dumper
    spec:
      containers:
      - name: dumper-container
        image: dump-env:continuous
```

Proměnné prostředí

https://kubernetes.io/docs/reference/kubernetes-api/workload-resources/pod-v1/

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: environment-dumper
spec:
  replicas: 1
  selector:
    matchLabels:
      app: environment-dumper
  template:
    metadata:
      name: environment-dumper
      labels:
        app: environment-dumper
    spec:
      containers:
      - name: dumper-container
        image: dump-env:continuous
        env:
        - name: OWNER
          value: Tady je Honzovo
```

Proměnné prostředí

- Změna v manifestu způsobí přenasazení podů

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: environment-dumper
spec:
  replicas: 1
  selector:
    matchLabels:
      app: environment-dumper
  template:
    metadata:
      name: environment-dumper
      labels:
        app: environment-dumper
    spec:
      containers:
      - name: dumper-container
        image: dump-env:continuous
        env:
        - name: OWNER
          value: Tady je Honzovo
```

ConfigMap

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  custom-config: |
    This is custom config
    Try to change me
  another-config: |
    This is another config
  unused-config: |
      This is another config
```

ConfigMap

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: config-reader
spec:
  selector:
    matchLabels:
      app: config-reader
  template:
    metadata:
      name: config-reader
      labels:
        app: config-reader
    spec:
      containers:
      - name: config-reader
        image: read-cfg:v1
```

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  cfg-file: /app/k8s.cfg
  custom-config: |
    This is custom config
    Try to change me
  another-config: |
    This is another config
  unused-config: |
      This is another config
```

ConfigMap

```
...
  template:
    ...
    spec:
      volumes:
      - name: config-volume
        configMap:
          name: app-config
          items:
          - key: custom-config
            path: k8s.cfg
          - key: another-config
            path: another.cfg
      containers:
      - name: config-reader
        image: read-cfg:v1
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  cfg-file: /app/k8s.cfg
  custom-config: |
    This is custom config
    Try to change me
  another-config: |
    This is another config
  unused-config: |
      This is another config
```

ConfigMap

```
...
  template:
    ...
    spec:
      volumes:
      - name: config-volume
        configMap:
          name: app-config
          items:
          - key: custom-config
            path: k8s.cfg
          - key: another-config
            path: another.cfg
      containers:
      - name: config-reader
        image: read-cfg:v1
        volumeMounts:
        - name: app-config
          mountPath: /app
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  cfg-file: /app/k8s.cfg
  custom-config: |
    This is custom config
    Try to change me
  another-config: |
    This is another config
  unused-config: |
      This is another config
```

ConfigMap

```
...
    spec:
      volumes:
      - name: config-volume
        configMap:
          name: app-config
          items:
          - key: custom-config
            path: k8s.cfg
          - key: another-config
            path: another.cfg
      containers:
      - name: config-reader
        image: read-cfg:v1
        volumeMounts:
        - name: app-config
          mountPath: /app
        env:
        - name: CFG_FILE
          valueFrom:
            configMapKeyRef:
              name: app-config
              key: cfg-file
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  cfg-file: /app/k8s.cfg
  custom-config: |
    This is custom config
    Try to change me
  another-config: |
    This is another config
  unused-config: |
      This is another config
```

ConfigMap

```yaml
...
    spec:
      volumes:
      - name: config-volume
        configMap:
          name: app-config
          items:
          - key: custom-config
            path: k8s.cfg
          - key: another-config
            path: another.cfg
      containers:
      - name: config-reader
        image: read-cfg:v1
        volumeMounts:
        - name: app-config
          mountPath: /app
        env:
        - name: CFG_FILE
          valueFrom:
            configMapKeyRef:
              name: app-config
              key: cfg-file
```

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  cfg-file: /app/k8s.cfg
  custom-config: |
    This is new config. What happened?
  another-config: |
    This is another config
  unused-config: |
      This is another config
```

ConfigMap

```
...
    spec:
      volumes:
      - name: config-volume
        configMap:
          name: app-config
          items:
          - key: custom-config
            path: k8s.cfg
          - key: another-config
            path: another.cfg
      containers:
      - name: config-reader
        image: read-cfg:v1
        volumeMounts:
        - name: app-config
          mountPath: /app
        env:
        - name: CFG_FILE
          valueFrom:
            configMapKeyRef:
              name: app-config
              key: cfg-file
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  cfg-file: /app/another.cfg
  custom-config: |
    This is new config. What happened?
  another-config: |
    This is another config
  unused-config: |
      This is another config
```

ConfigMap

- Umožňuje držet konfiguraci mimo aplikaci
- Manifest obsahující seznam klíčů s hodnotou
- Hodnota pro každý klíč může být
  - Mountuta jako soubor do filesystmu kontejneru
  - Předána jako proměnná prostředí
  - Kombinace předchozího
- Pokud se změní ConfigMap, Pod který ji používá se automaticky **nerestartuje**
  - Mountnutý soubor v podu se upravý
  - Pokud se upraví hodnota, která se používá jako proměnná prostředí, v podu se tato změna neprojeví

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  cfg-file: /app/another.cfg
  custom-config: |
    This is new config. What happened?
  another-config: |
    This is another config
  unused-config: |
      This is another config
```

Secrets

- Stejné chování jako u ConfigMap
- Vlastní resource, umožňuje řídit přístup k nim v kubernetu
- V produkčních clusterech jsou typicky synchroniyzováný s externím secret managerem

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: server-secret
type: Opaque
stringData:
  password: P0wn3d
```

Komunikace

3 rules:
-no wishing for death
-no falling in love
-no bring back dead people
I wish to understand kubernetes
There are 4 rules

Mezi kontejnery v jednom podu

- Volumes
  - Sdílený adresář
    - Soubory
    - UNIX socket
    - Pipe
- Persistent Volume Claim (PVC)
  - Storage v podobě filesystému, který přežije vypnutí podu
- Network
  - Všechny kontejnery v podu sdílí linux namespace -> localhost
  - Pozor na stejné porty
- Signály

Sdílený adresář

```
apiVersion: apps/v1
kind: Deployment
...
  template:
    ...
    spec:
      containers:
      # Serves static pages from /usr/share/nginx/html
      - name: nginx
        image: nginx:1.21.1

      # Generates static pages with counter
      - name: data-generator
        image: data-generator:v1

      # Logs pages served by nginx
      - name: local-client
        image: http-client:v1
```

Sdílený adresář

```yaml
apiVersion: apps/v1
kind: Deployment
...
  template:
    ...
    spec:
      containers:
      # Serves static pages from /usr/share/nginx/html
      - name: nginx
        image: nginx:1.21.1

      # Generates static pages with counter
      - name: data-generator
        image: data-generator:v1

      # Logs pages served by nginx
      - name: local-client
        image: http-client:v1
        env:
        - name: TARGET
          value: localhost
```

Sdílený adresář

```
apiVersion: apps/v1
kind: Deployment
...
  template:
    ...
    spec:
      volumes:
      - name: html-root
        emptyDir: { }
      containers:
      # Serves static pages from /usr/share/nginx/html
      - name: nginx
        image: nginx:1.21.1
```

```
# Generates static pages with counter
- name: data-generator
  image: data-generator:v1

# Logs pages served by nginx
- name: local-client
  image: http-client:v1
  env:
  - name: TARGET
    value: localhost
```

Sdílený adresář

```yaml
apiVersion: apps/v1
kind: Deployment
...
  template:
    ...
    spec:
      volumes:
      - name: html-root
        emptyDir: { }
      containers:
      # Serves static pages from /usr/share/nginx/html
      - name: nginx
        image: nginx:1.21.1
        volumeMounts:
        - mountPath: /usr/share/nginx/html
          name: html-root
```

```yaml
# Generates static pages with counter
- name: data-generator
  image: data-generator:v1
  volumeMounts:
  - mountPath: /data
    name: html-root

# Logs pages served by nginx
- name: local-client
  image: http-client:v1
  env:
  - name: TARGET
    value: localhost
```

Mezi pody v kubernetu

- Vyžaduje se Service
  - Service má vlastní IP
  - Vytvoření service -> vytvoří endpointy s IP podů (`kubectl get endpoints`)
  - Pody pak směřují na IP service z ní se provede překlad na konkrétní pod
- Persistent Volumes
  - Sdílí se část filesystému, např přes S3
  - Vhodné pro velmi obskurdní případy

Komunikace přes síť

```
apiVersion: v1
kind: Service
metadata:
  name: web-server
spec:
  selector:
    app: comm-server
  ports:
  - name: http
    port: 80
    targetPort: 80
```

Komunikace přes síť

```yaml
apiVersion: v1
kind: Service
metadata:
  name: web-server
spec:
  selector:
    app: comm-server
  ports:
  - name: http
    port: 80
    targetPort: 80
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: comm-client
spec:
  replicas: 1
  selector:
    matchLabels:
      app: comm-client
  template:
    metadata:
      name: comm-client
      labels:
        app: comm-client
    spec:
      containers:
      - name: local-client
        image: http-client:v1
        env:
        - name: TARGET
          value: ???
```

Komunikace přes síť

```yaml
apiVersion: v1
kind: Service
metadata:
  name: web-server
spec:
  selector:
    app: comm-server
  ports:
  - name: http
    port: 80
    targetPort: 80
```

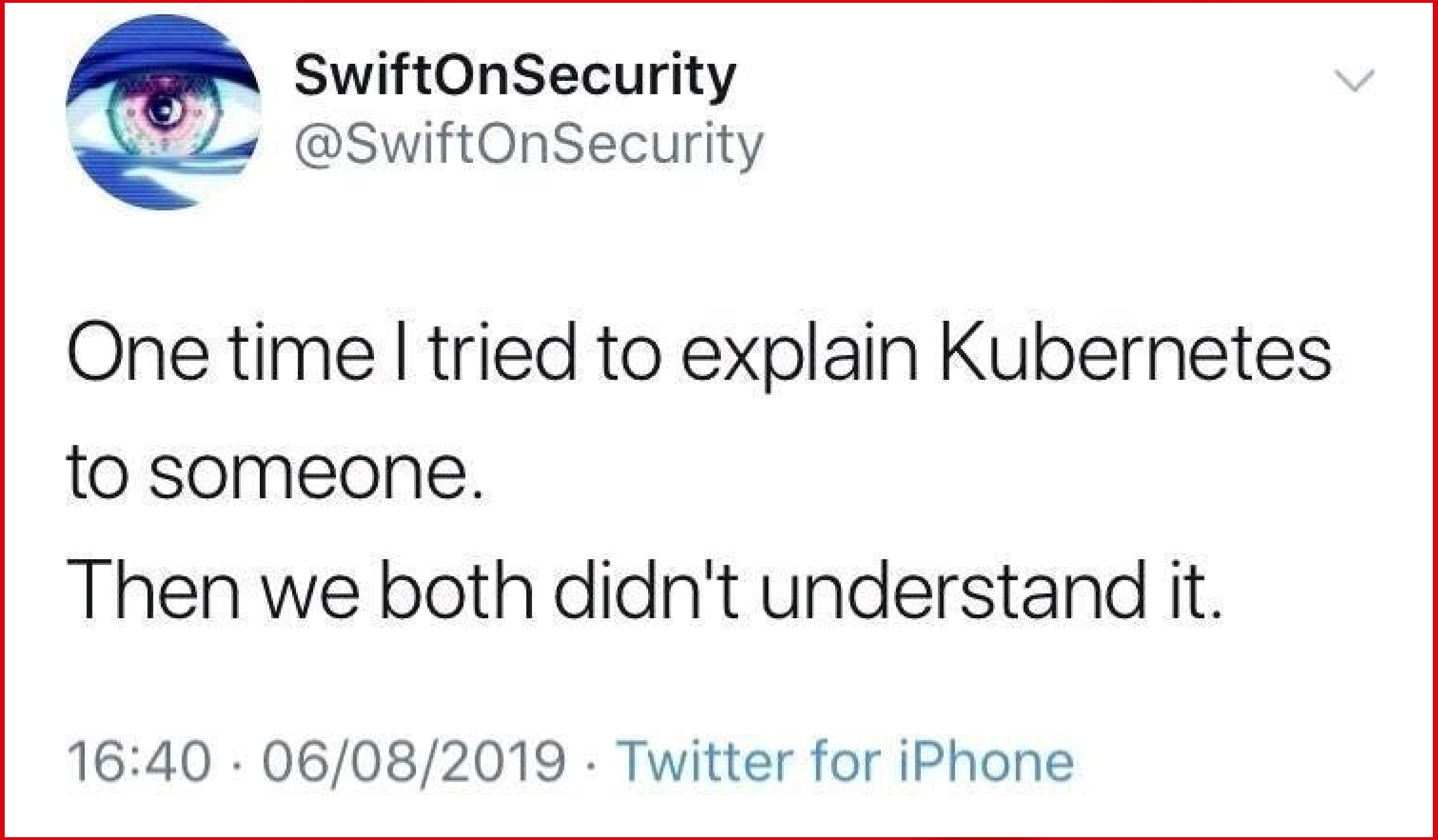
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: comm-client
spec:
  replicas: 1
  selector:
    matchLabels:
      app: comm-client
  template:
    metadata:
      name: comm-client
      labels:
        app: comm-client
    spec:
      containers:
      - name: local-client
        image: http-client:v1
        env:
        - name: TARGET
          value: web-server
```

Přístup mimo kubernet

- Typicky nginx, který běží v kubernetu
  - Provoz na servicy směruje podle:
    - Doménového jména
    - Cesty v URL

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: comm
spec:
  rules:
  - host: communication.k8s
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web-server
            port:
              name: http
```

Workloads 2



SwiftOnSecurity
@SwiftOnSecurity

One time I tried to explain Kubernetes to someone.
Then we both didn't understand it.

16:40 · 06/08/2019 · Twitter for iPhone

Statefulset

```yaml
apiVersion: v1
kind: Service
metadata:
  name: counter-sts
spec:
  clusterIP: None
```

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: stateful-counter
spec:
  replicas: 3
  serviceName: counter-sts
  selector:
    matchLabels:
      app: counter-sts
  template:
    metadata:
      name: counter-pod
      labels:
        app: counter-sts
    spec:
      containers:
      - name: counter-container
        image: counter:v1
```

Statefulset

- Bez replica setů
- Vlastní pody
- Pod si zachovává
  - síťovou identitu (nezáleží na počtu restartů vždy bude mít stejný hostname)
  - storage identitu (poze při použití PVC, vždy dostane stejný storage)
- Vyžaduje **`Headless Service`**.
- Pojmenování podů:
  - `<StatefulSet>-<counter>`

```
apiVersion: v1
kind: Service
metadata:
  name: counter-sts
spec:
  clusterIP: None
```

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: stateful-counter
spec:
  replicas: 3
  serviceName: counter-sts
  selector:
    matchLabels:
      app: counter-sts
  template:
    metadata:
      name: counter-pod
      labels:
        app: counter-sts
    spec:
      containers:
      - name: counter-container
        image: counter:v1
```

Job

- Vlastní pody
- Spustí sekvenčně nebo paralelně pody
- Joby a Pody zůstavají v K8s po skončení pro check logů
  - TTL mechanismus pro úklid
  - Např. batchové zpracování dat z front
- Pojmenování podů:
  - `<Job>-<PodId>`

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: dump-job
spec:
  template:
    spec:
      restartPolicy: OnFailure
      containers:
      - name: counter-container
        image: dump-env:one-shot
```

DaemonSet

- Vlastní pody
- Zajistí spuštění podů na každém nodu v clusteru
- Pozor na tainty (omezení nastavené na Nodu určující co na něm může běžet)
- Používá se pro **`CronJob`**
- Užitečné s **`HostPath`**
- Pojmenování podů:
  - `<DaemonSet>-<PodId>`

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: counters-everywhere
spec:
  selector:
    matchLabels:
      app: counter
  template:
    metadata:
      name: counter-pod
      labels:
        app: counter
    spec:
      containers:
      - name: counter-container
        image: counter:v1
```

That's all Folks!