



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Operators in Action

**Making Kubernetes Work
for You**

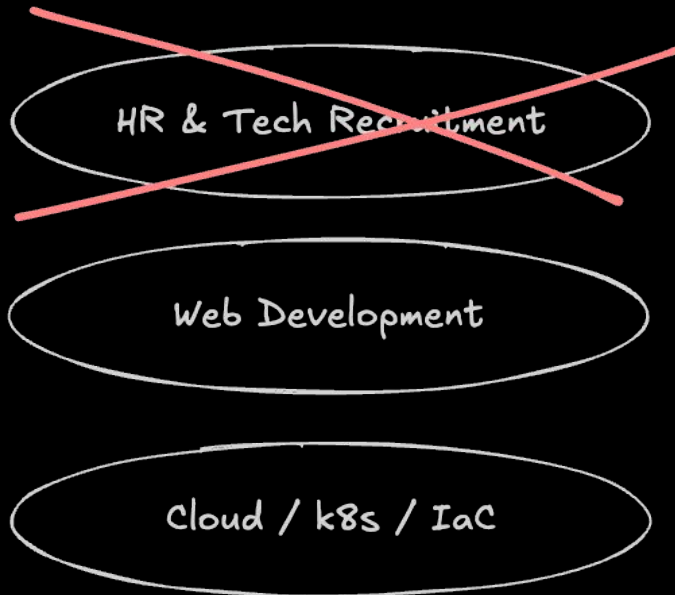
Verena Traub



Hi, I'm Verena 🖐️



Cloud & DevOps
Consultant
@ b'nerd



True love ...

b'nerd



KubeCon



CloudNativeCon

Europe 2026

b'



KubeCon



CloudNativeCon

Europe 2026

b'



KubeCon



CloudNativeCon

Europe 2026

Client's
onboarding
app



b'

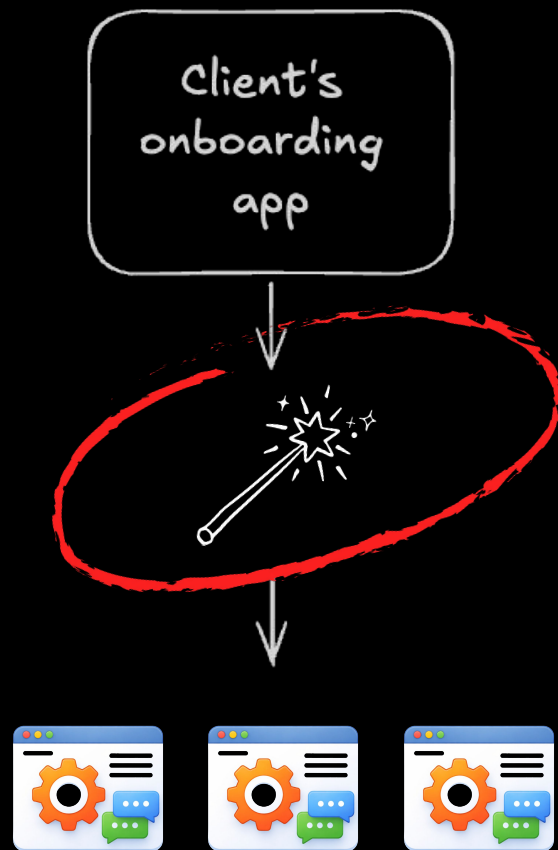


KubeCon



CloudNativeCon

Europe 2026



b'





Kubernetes Operators



Extension



KubeCon

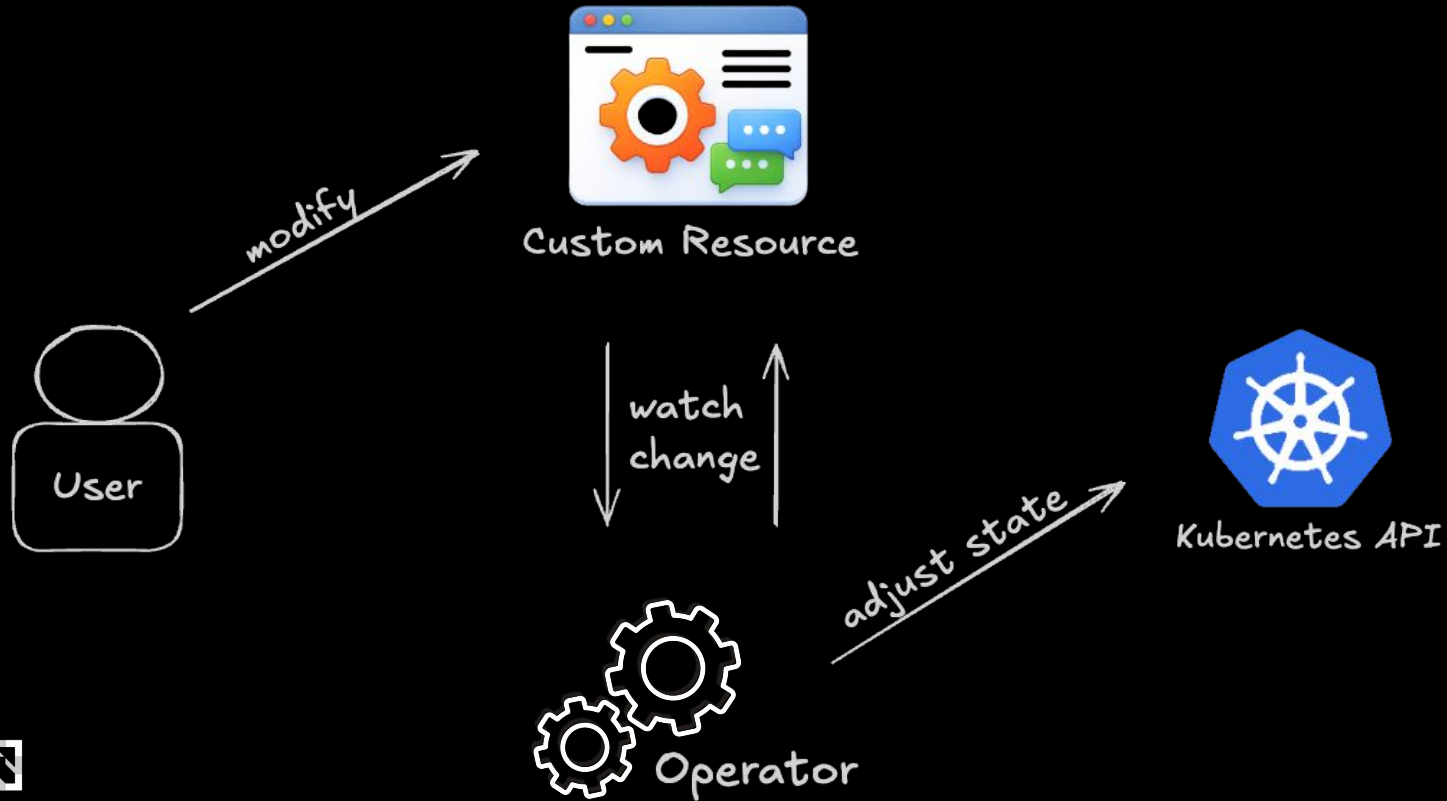


CloudNativeCon

Europe 2026

b'

What operators are all about



What operators are all about



new API
type



desired
state



reconciles
resources

CustomResourceDefinition

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: helloworld.apps.demo.local
spec:
  group: apps.demo.local
  names:
    kind: HelloWorldApp
    listKind: HelloWorldAppList
    singular: helloworldapp
    plural: helloworldapps
    shortNames:
      - hwa
  scope: Namespaced
```

kubectl get helloworldapps
kubectl get hwa

CustomResourceDefinition

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: helloworld.apps.demo.local
spec:
  group: apps.demo.local
  names:
    kind: HelloWorldApp
    listKind: HelloWorldAppList
    singular: helloworldapp
    plural: helloworldapps
    shortNames:
      - hwa
  scope: Namespaced
```

```
versions:
- name: v1alpha1
  schema:
    openAPIV3Schema:
      type: object
      properties:
        spec: # desired state
        properties:
          image:
            type: string
          replicas:
            type: integer
          port:
            type: integer
        status: # observed state
        properties:
          availableReplicas:
            type: integer
```



KubeCon



CloudNativeCon

Europe 2026

b'

Custom Resource

```
apiVersion: apps.demo.local/v1alpha1
kind: HelloWorldApp
metadata:
  name: demo
spec:
  image: hello-world
  replicas: 3
  port: 8080
```

Reconciliation Loop

```
func (r *HelloWorldAppReconciler) Reconcile(ctx context.Context, req ctrl.Request) {  
  
    // 1. Read the custom resource  
    app := getHelloWorldApp(req)  
  
    // 2. Ensure desired resources exist  
    ensureDeployment(app.spec.image, app.spec.replicas)  
    ensureService(app.spec.port)  
  
    // 3. Update status  
    app.status.availableReplicas = deployment.status.availableReplicas  
    updateStatus(app)  
}
```



KubeCon



CloudNativeCon

Europe 2026





KubeCon



CloudNativeCon
Europe 2026

b'

**External services
on non-k8s resources**

Start-up time

**Automatic name and
credential generation**

DNS integration

Different profiles

App Dependencies



KubeCon



CloudNativeCon
Europe 2026

b'



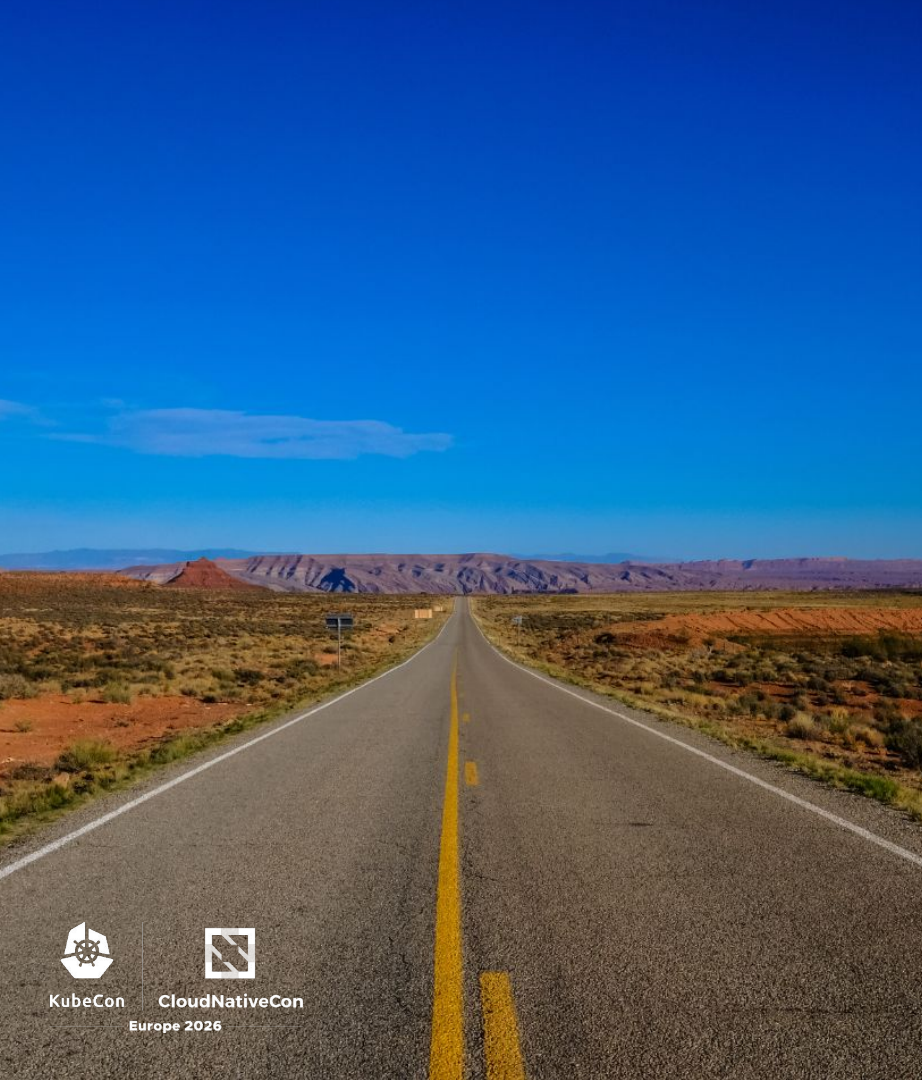
KubeCon



CloudNativeCon

Europe 2026

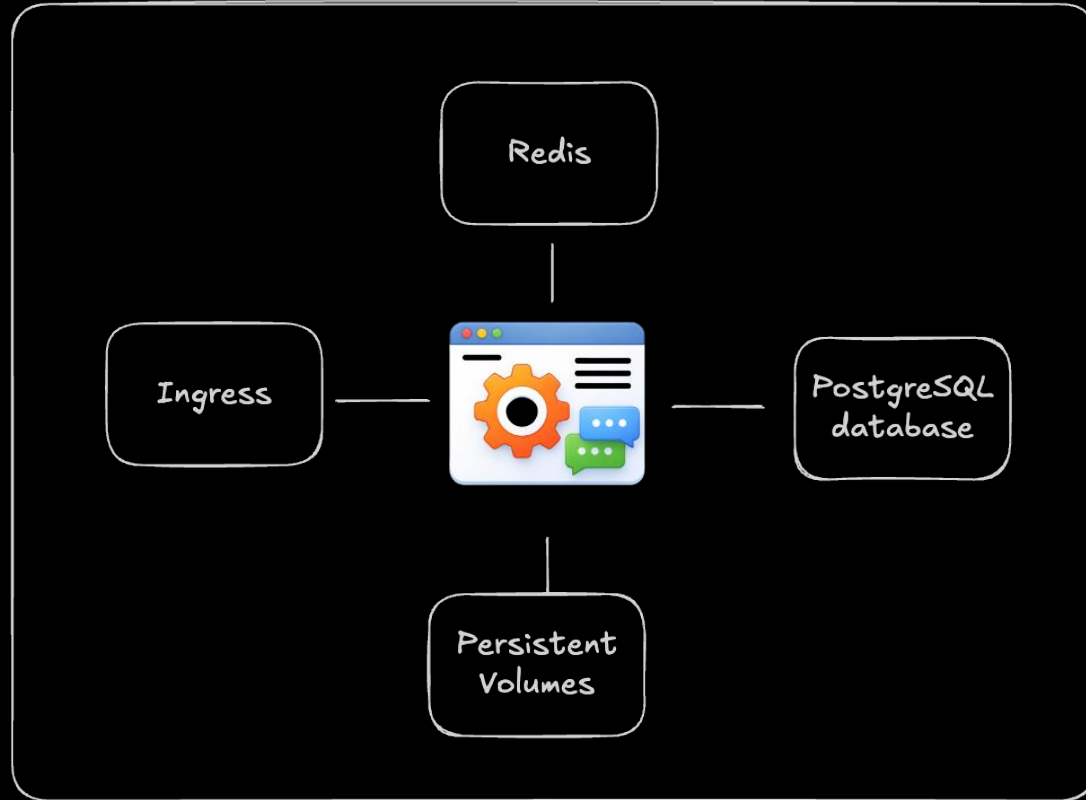
b'



Handling app dependencies

Handling app dependencies

Kubernetes cluster



KubeCon



CloudNativeCon

Europe 2026

b'

CRD

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  // ...
spec:
  names:
    kind: ClientApp
    // ...
  scope: Namespaced
  // ...
```

```
schema:
  openAPIV3Schema:
    type: object
    properties:
      spec:
        // ...
        database:
          // ...
        redis:
          // ...
        s3:
          // ...
        admin:
          // ...
        ingress:
          // ...
        // ...
```



KubeCon



CloudNativeCon

Europe 2026

b'

Dependency handling

```
// Check database dependency
if !databaseReady(app) {
    log.Info("Database not ready - requeueing in 30s")
    updateStatusCondition(app, "DatabaseReady", "False", "Pending", "waiting for
database")
    return ctrl.Result{RequeueAfter: 30 * time.Second, nil}
}
updateStatusCondition(app, "DatabaseReady", "True", "Ready", "Database is ready")

// Check Redis & PVC readiness
// ...

log.Info("Dependencies ready")
```



KubeCon



CloudNativeCon

Europe 2026



Credentials - Dependency handling

```
// Postgres
credSecretName := clusterName + "-pguser-clientapp"
credSecret := &corev1.Secret{}

err := r.Get(ctx, types.NamespacedName{
    Name:      credSecretName,
    Namespace: namespace,
}, credSecret)

if err != nil {
    log.Info("Postgres secret not ready - requeueing")
    return ctrl.Result{RequeueAfter: 30 * time.Second}, nil
}

log.Info("Postgres secret available")
```

Automatic name and credential generation



KubeCon



CloudNativeCon

Europe 2026



b'

“Haiku-ify” names & namespaces

```
var adjectives = []string{
    "bold", "calm", "dark", "fast", "gold", "nice", "kind", "lean",
    // more adjectives
}

var nouns = []string{
    "leaf", "wave", "peak", "dawn", "dusk", "mist", "flower", "snow",
    // more nouns
}

func GenerateHaikuName() (string, error) {
    b := make([]byte, 3)
    if _, err := rand.Read(b); err != nil {
        return "", fmt.Errorf("failed to generate random bytes: %w", err)
    }

    adj := adjectives[int(b[0])%len(adjectives)]
    noun := nouns[int(b[1])%len(nouns)]
    suffix := strings.ToLower(base64.RawURLEncoding.EncodeToString(b[2:]))[:6]

    return fmt.Sprintf("ca-%s-%s-%s", adj, noun, suffix), nil
}
```



ca-gold-leaf-j5cvv1

“Haiku-ify” names & namespaces

```
namespace := &corev1.Namespace{  
    ObjectMeta: metav1.ObjectMeta{  
        Name: name,  
    },  
}
```

```
instance := &appsv1alpha1.ClientApp{  
    ObjectMeta: metav1.ObjectMeta{  
        Name:      name,  
        Namespace: name,  
    },  
    // ...  
}
```



KubeCon



CloudNativeCon

Europe 2026



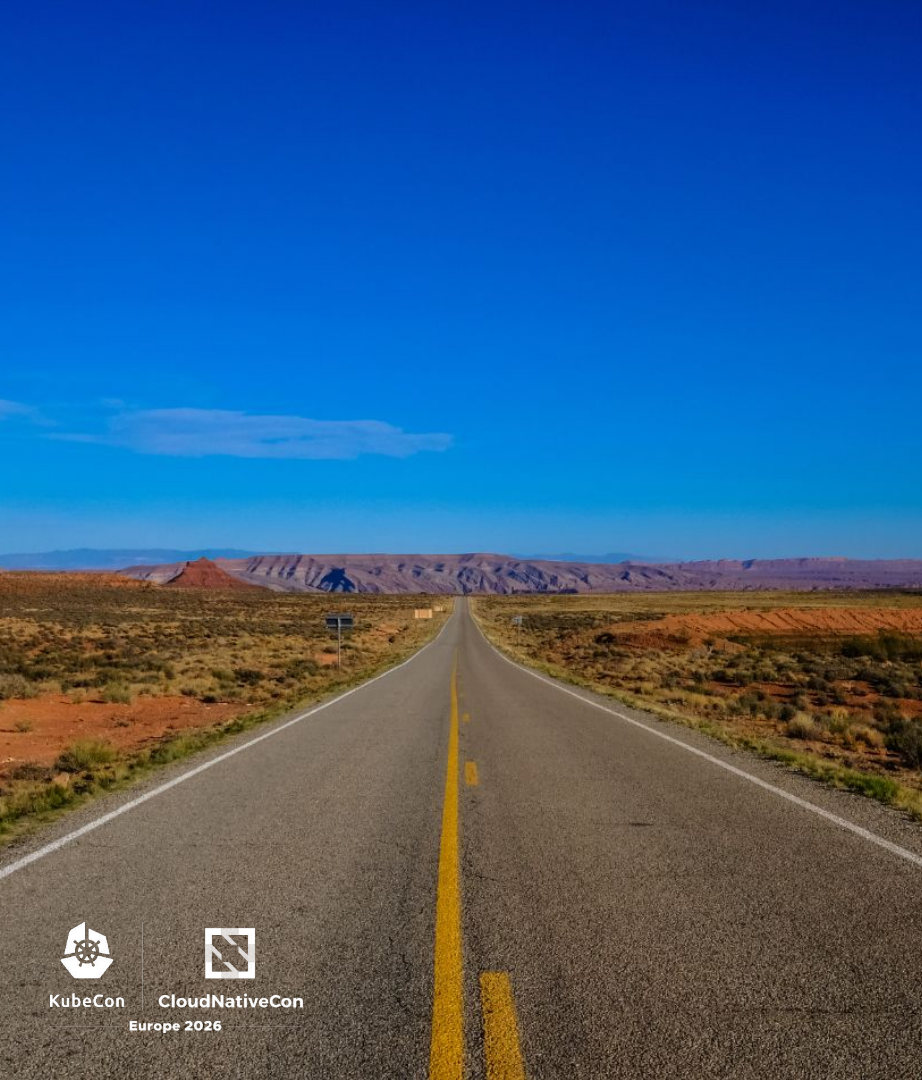
Credentials - App credentials

```
secretName := instance.Spec.AdminSecret
var adminPassword string

if secretName == "" {
    secretName = fmt.Sprintf("%s-admin-secret", instanceName)
    adminPassword = utils.GeneratePassword(32)

    secret := &corev1.Secret{
        ObjectMeta: metav1.ObjectMeta{
            Name:      secretName,
            Namespace: namespace,
        },
        StringData: map[string]string{
            "adminPassword": adminPassword,
        },
        Type: corev1.SecretTypeOpaque,
    }

    if err := r.Client.Create(ctx, secret); err != nil {
        log.Error(err, "Failed to create admin secret")
        return ctrl.Result{}, err
    }
}
```



Different profiles

New CRD

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  // ...
spec:
  group: k8s.bnerd.com
  names:
    kind: ClientAppProfile
    plural: clientappprofiles
    singular: clientappprofiles
    shortNames: [caprofile, profile]
  scope: Cluster
  // ...
  versions:
    - name: v1alpha1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              required: [defaults]
```



KubeCon



CloudNativeCon

Europe 2026

b'

New CRD

```
spec:
  defaults:
    type: object
    x-kubernetes-preserve-unknown-fields: true
  properties:
    apps:
      type: object
      properties:
        app1:
          type: object
          properties:
            enabled:
              type: boolean
              default: false
            backends:
              type: array
              items:
                type: object
                properties:
                  server: string
                  secret: string
        app2:
          type: object
          properties:
            enabled:
              type: boolean
              default: true
      // more apps
```



KubeCon



CloudNativeCon

Europe 2026

b'

CR then

```
apiVersion: apps.demo.local/v1alpha1
kind: ClientAppProfile
metadata:
  // ...
spec:
  apps:
    app1:
      enabled: false
    app2:
      enabled: true
    app3:
      enabled: true
  // more apps
```

Startup Time



New CRD

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  // ...
spec:
  group: k8s.bnerd.com
  names:
    kind: ClientAppPool
    plural: clientapppools
    singular: clientapppools
    shortNames:
      - cap
  scope: Cluster
  versions:
    - name: v1alpha1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              required:
                - replicas
                - instanceNamespacePattern
```



KubeCon



CloudNativeCon

Europe 2026



New CRD

```
spec:
  type: object
  required:
    - replicas
    - instanceNamespacePattern
  properties:
    replicas:
      type: integer
      description: "Number of unassigned instances to maintain in the pool"
      minimum: 0
      default: 5
    instanceNamespacePattern:
      type: string
      description: "Namespace pattern for instances. Use * as placeholder for name"
      default: "*"
    profileRef:
      type: object
      required:
        - name
      properties:
        name:
          type: string
          // ...
```



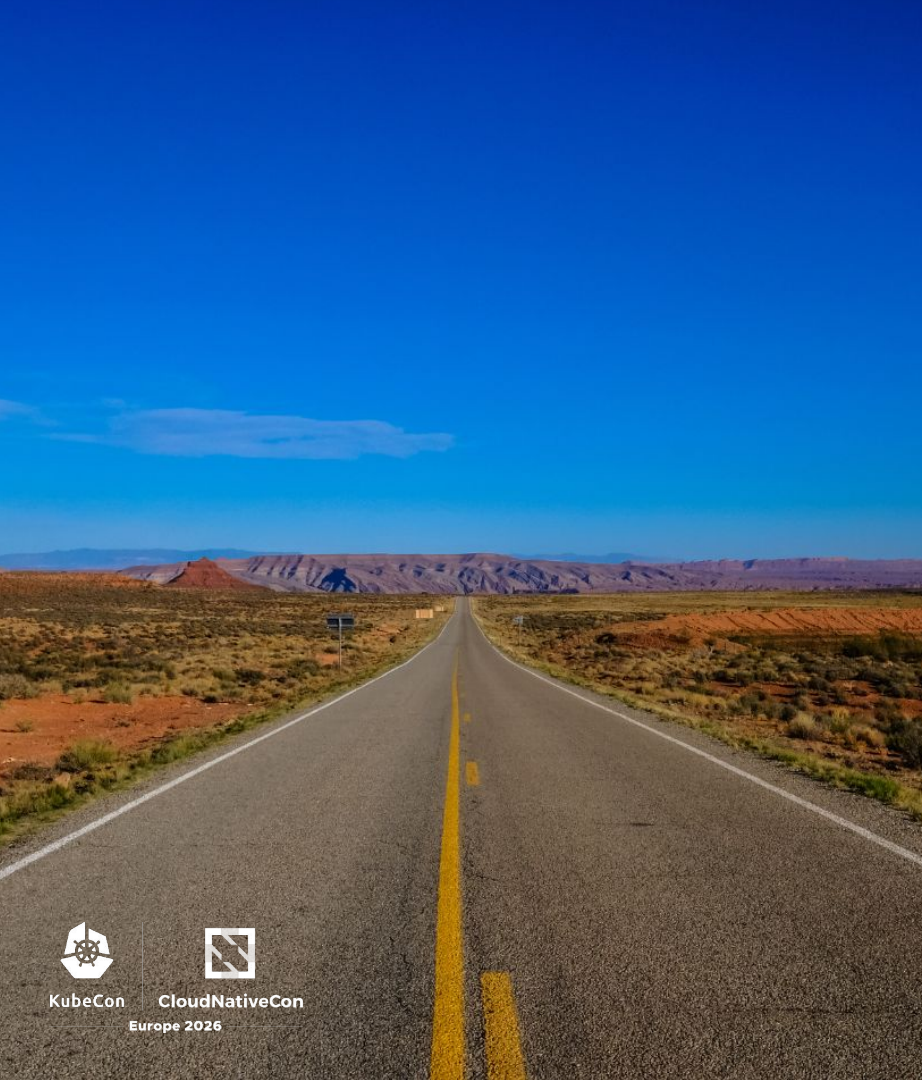
KubeCon



CloudNativeCon

Europe 2026

b'



KubeCon

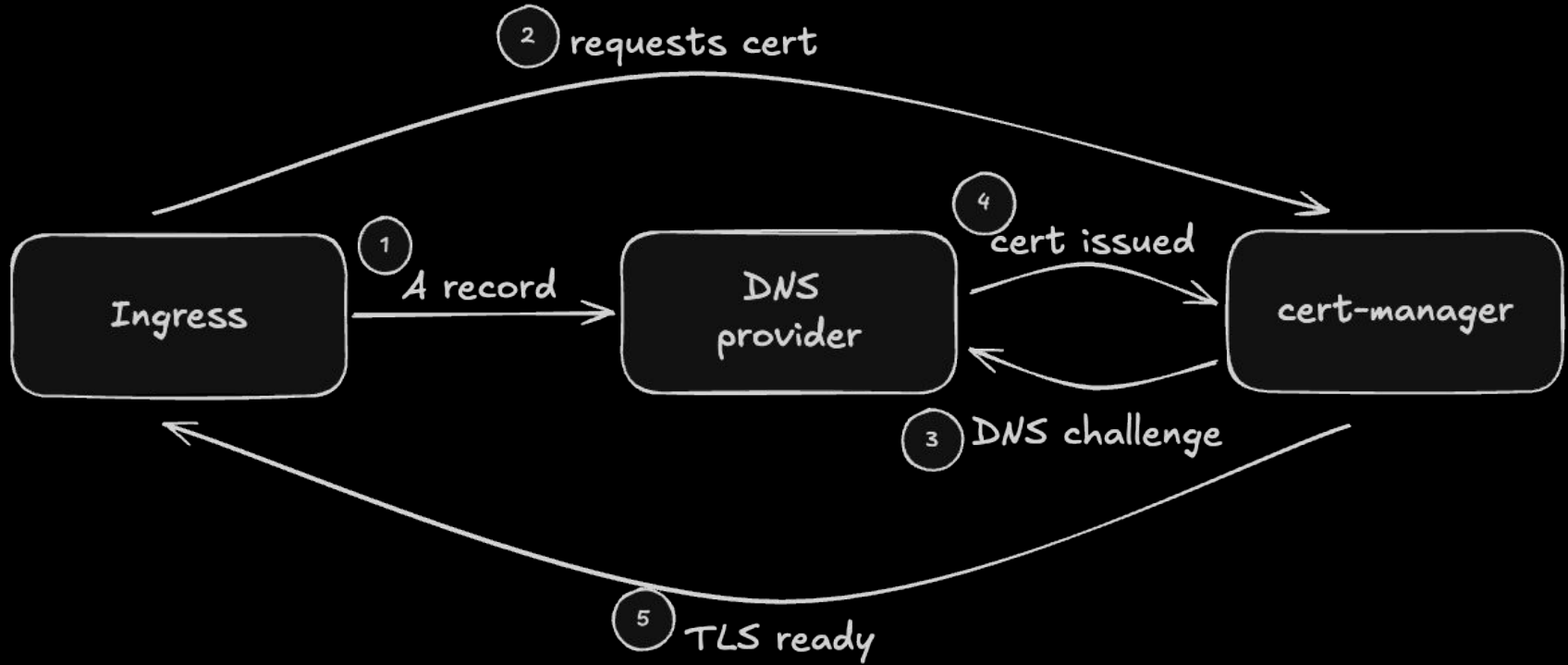


CloudNativeCon

Europe 2026

DNS Integration

b'



```
func buildIngress(name, host string) *networkingv1.Ingress {
    pathType := networkingv1.PathTypePrefix
    return &networkingv1.Ingress{
        ObjectMeta: metav1.ObjectMeta{
            Name: name,
            Annotations: map[string]string{
                "cert-manager.io/cluster-issuer": "letsencrypt-prod",
            },
        },
        Spec: networkingv1.IngressSpec{
            TLS: []networkingv1.IngressTLS{{Hosts: []string{host}, SecretName: name + "-tls"}}},
            Rules: []networkingv1.IngressRule{{Host: host}},
        },
    }
}
```



KubeCon



CloudNativeCon

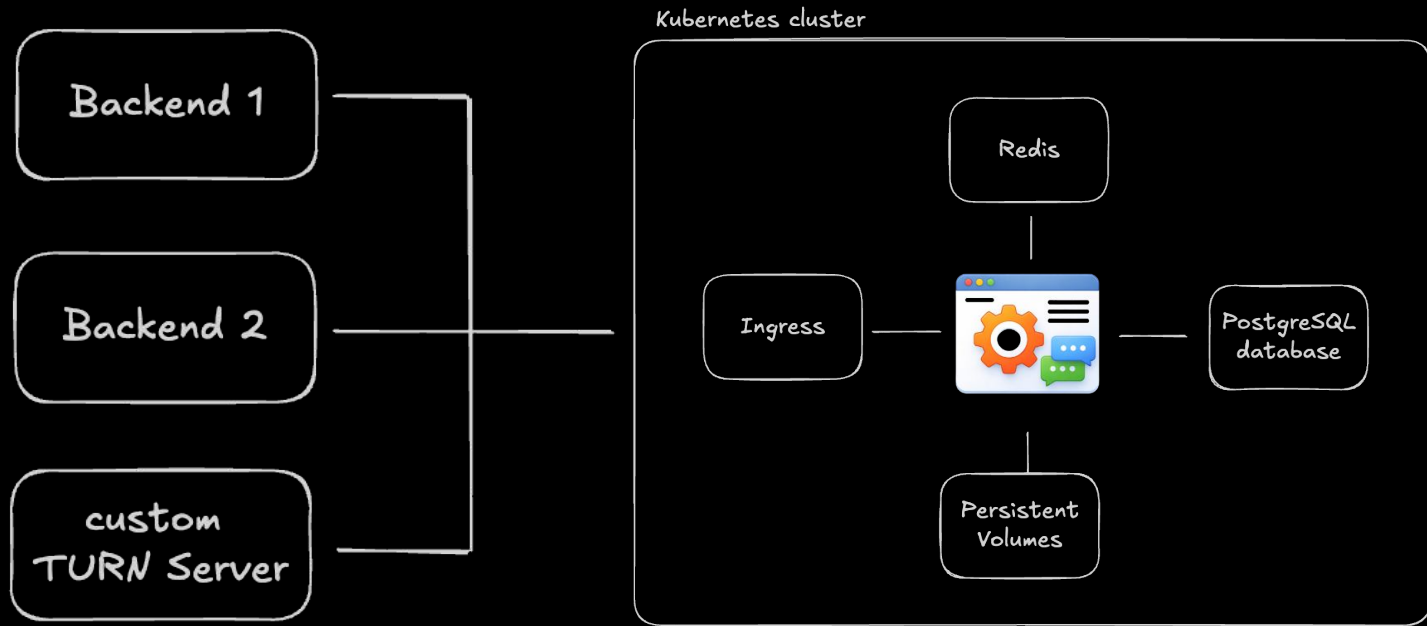
Europe 2026

b'

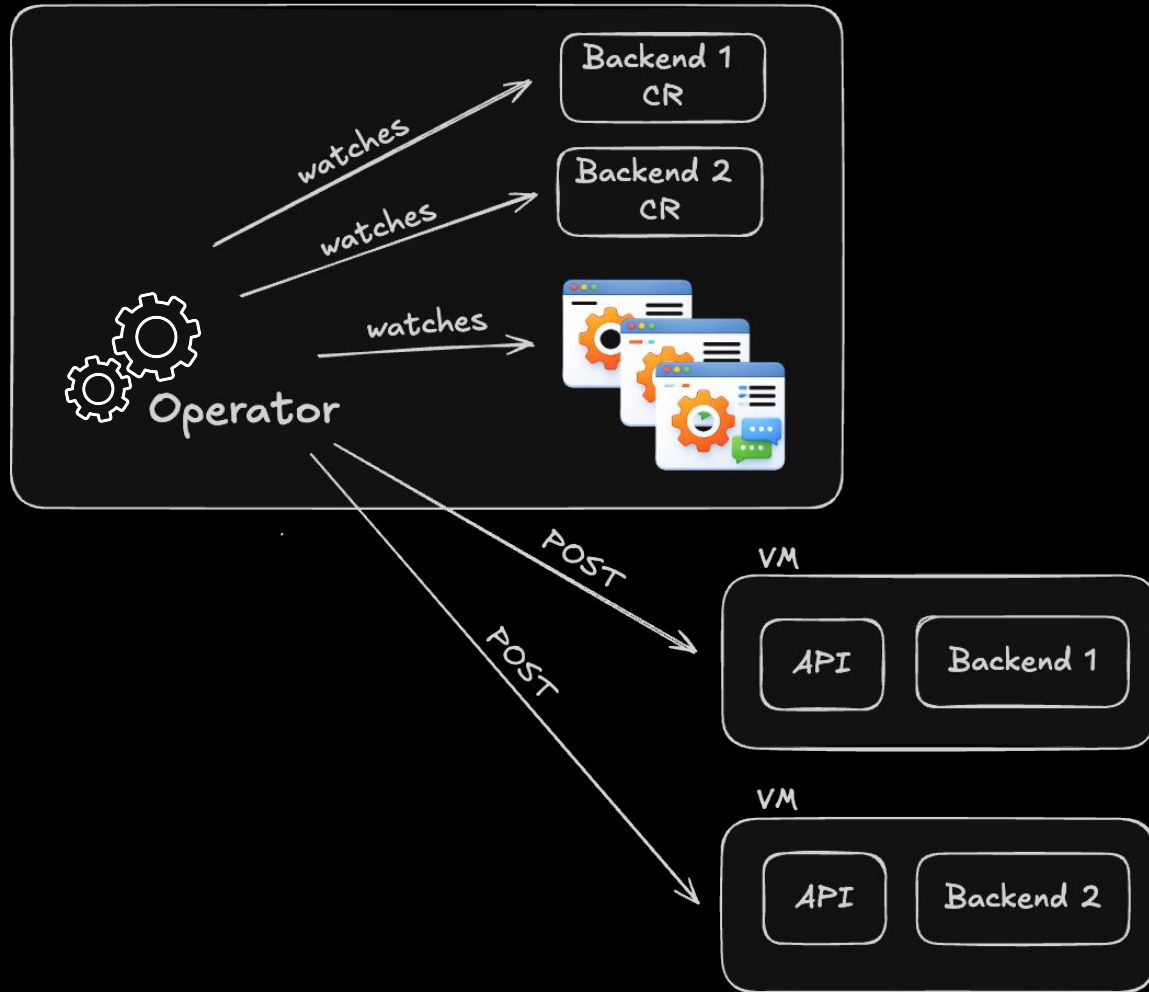
External Components



Separate components outside of Kubernetes



Kubernetes Cluster



```

func (r *ClientAppReconciler) Reconcile(ctx context.Context, req ctrl.Request) (ctrl.Result, error)
{
    var bo v1alpha1.BackendOne
    r.Get(ctx, req.NamespacedName, bo)

    current, _ := listBackends(bo.Spec.APIEndpoint, apiKey) // GET backend-one-api
    desired, _ := r.listInstancesWithAppOne(ctx)

    for id, b := range desired {
        if _, exists := current[id]; !exists {
            registerBackend(bo.Spec.APIEndpoint, apiKey, id, b) // POST backend-one-api
        }
    }

    for id := range current {
        if _, exists := desired[id]; !exists {
            deregisterBackend(bo.Spec.APIEndpoint, apiKey, id) // DELETE backend-one-api
        }
    }

    return ctrl.Result{RequeueAfter: 60 * time.Second}, nil
}

```



KubeCon



CloudNativeCon

Europe 2026

b'



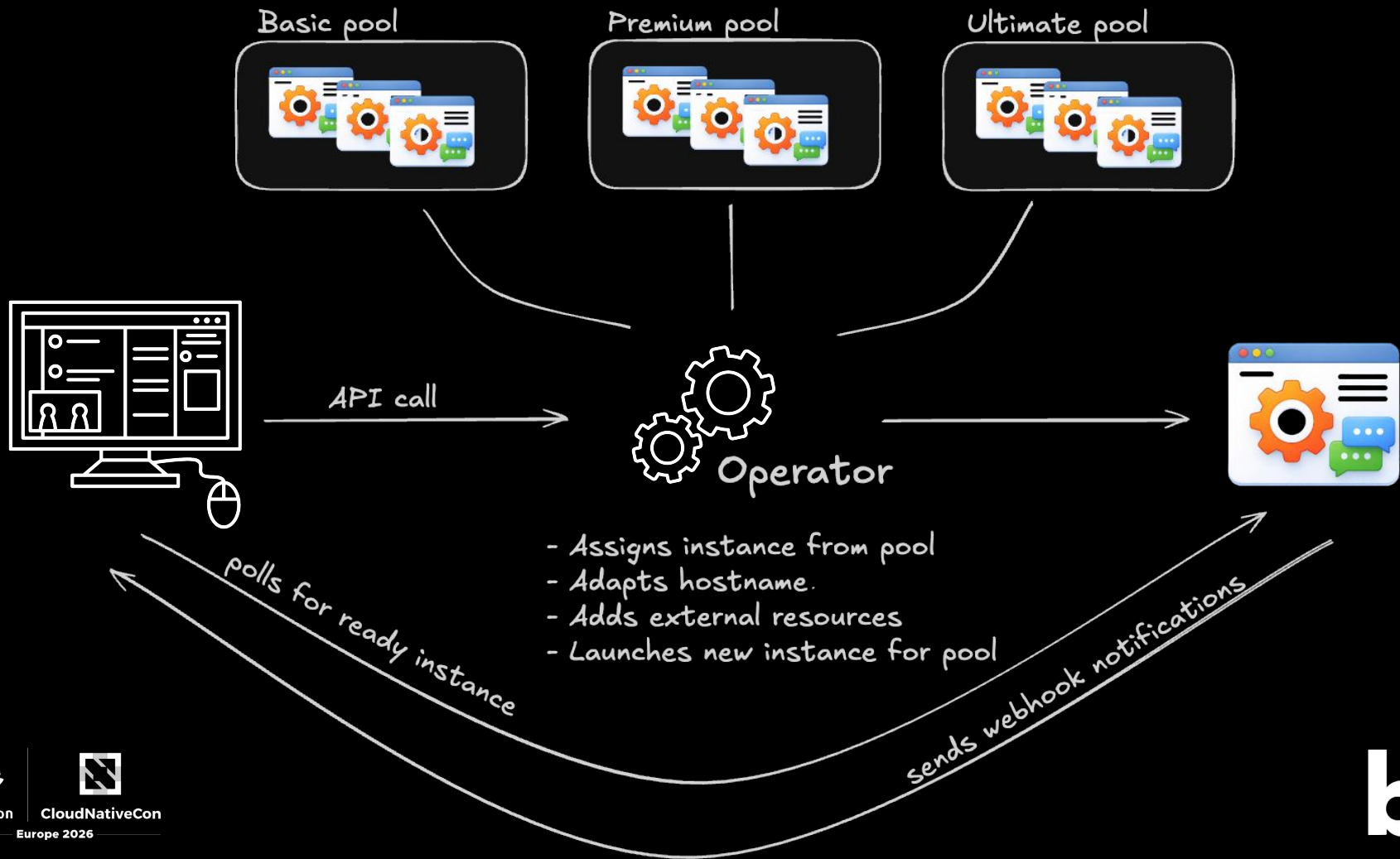
KubeCon



CloudNativeCon

Europe 2026

b'





CRD design
matters a lot



KubeCon

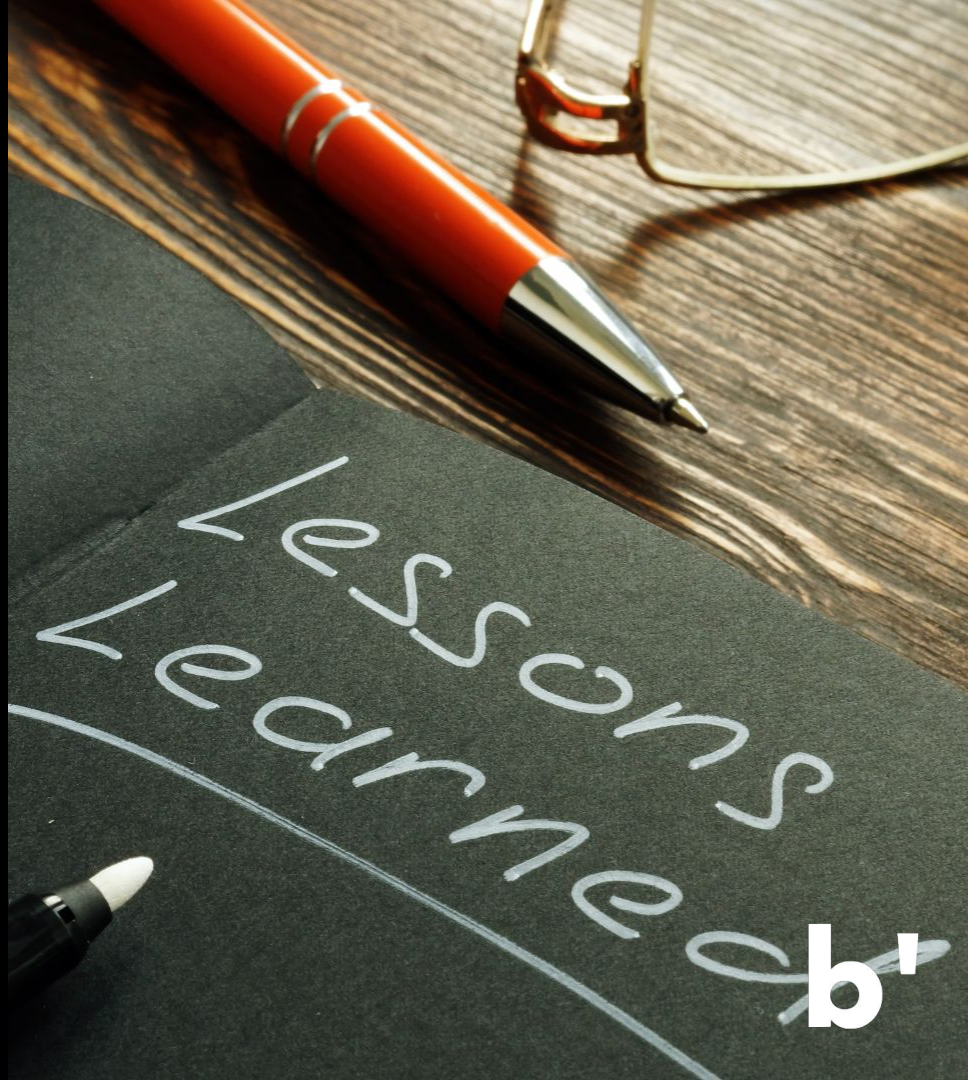


CloudNativeCon

Europe 2026

b'

Reconciliation
complexity grows
quickly





Cleaning up
resources needs
more love than
expected



KubeCon



CloudNativeCon

Europe 2026

b'

Testing (and
debugging)
operators is tricky

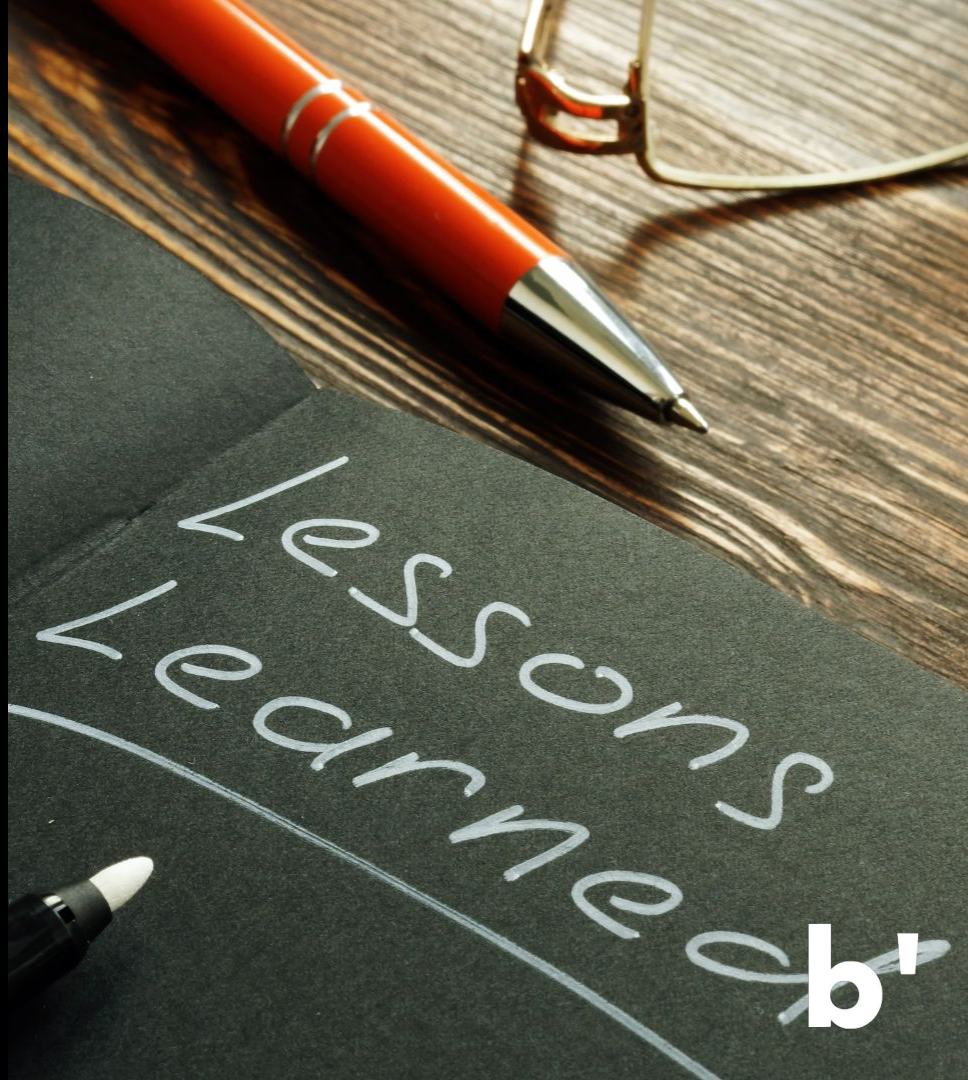


KubeCon



CloudNativeCon

Europe 2026



b'



Clear logs, events &
status reporting
matter



KubeCon



CloudNativeCon

Europe 2026

b'



KubeCon



CloudNativeCon

Europe 2026

Thank you! Questions?



Verena Traub
verena@bnerd.com

