Phased upgrades of Strimzi Managed Kafka Fleets

Thomas Cooper

Senior Principal Software Engineer

Red Hat

What am I going to talk about...

- Strimzi Upgrades

  How do they work and what you need to watch out for

- How could we do phased upgrades?

  What options are there for implementing a more controlled approach

- Concurrent Strimzi deployments with Kustomize

  A proposed setup that will work with Strimzi out of the box

- Phased upgrade process and future steps

  How to use the proposed process to do upgrades and what else this setup enables

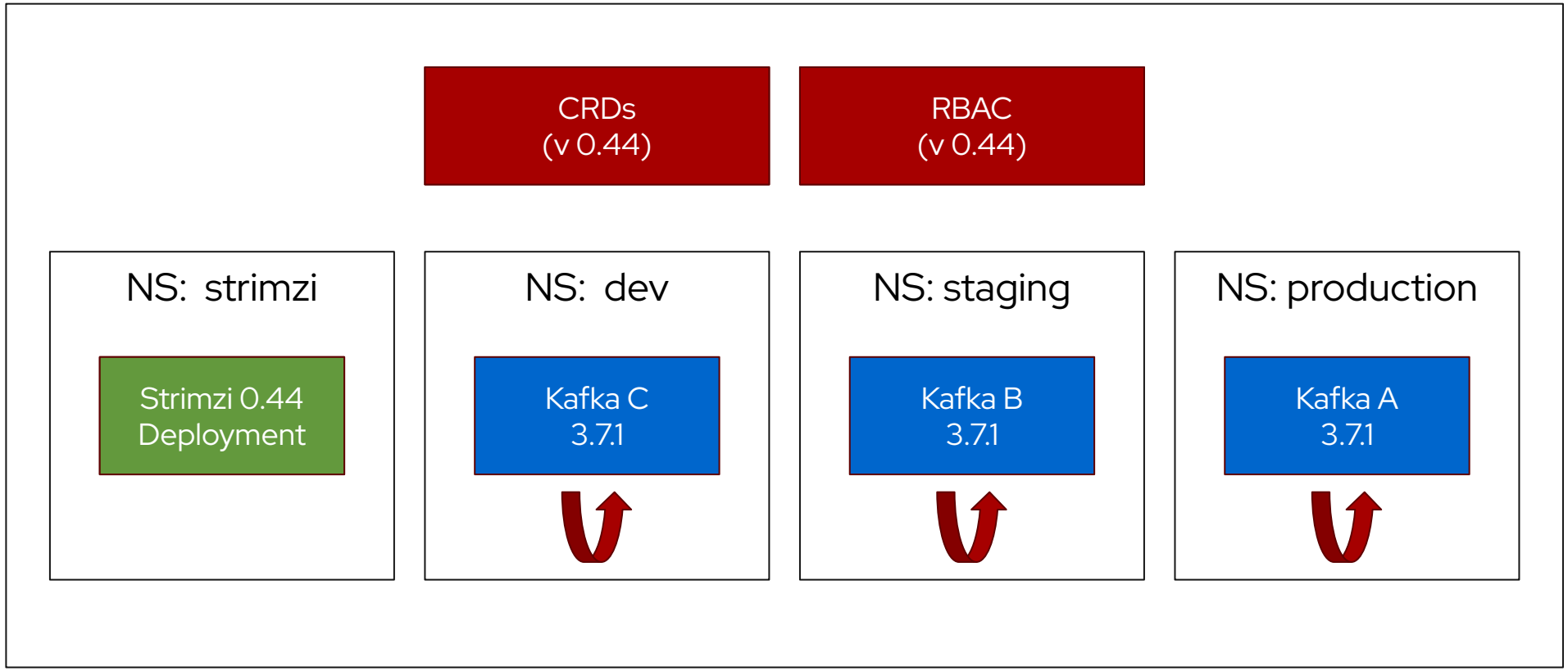
This presentation should have been a blog post...

- https://strimzi.io/blog/2025/04/10/phased-strimzi-upgrade-example/

Show me the code!

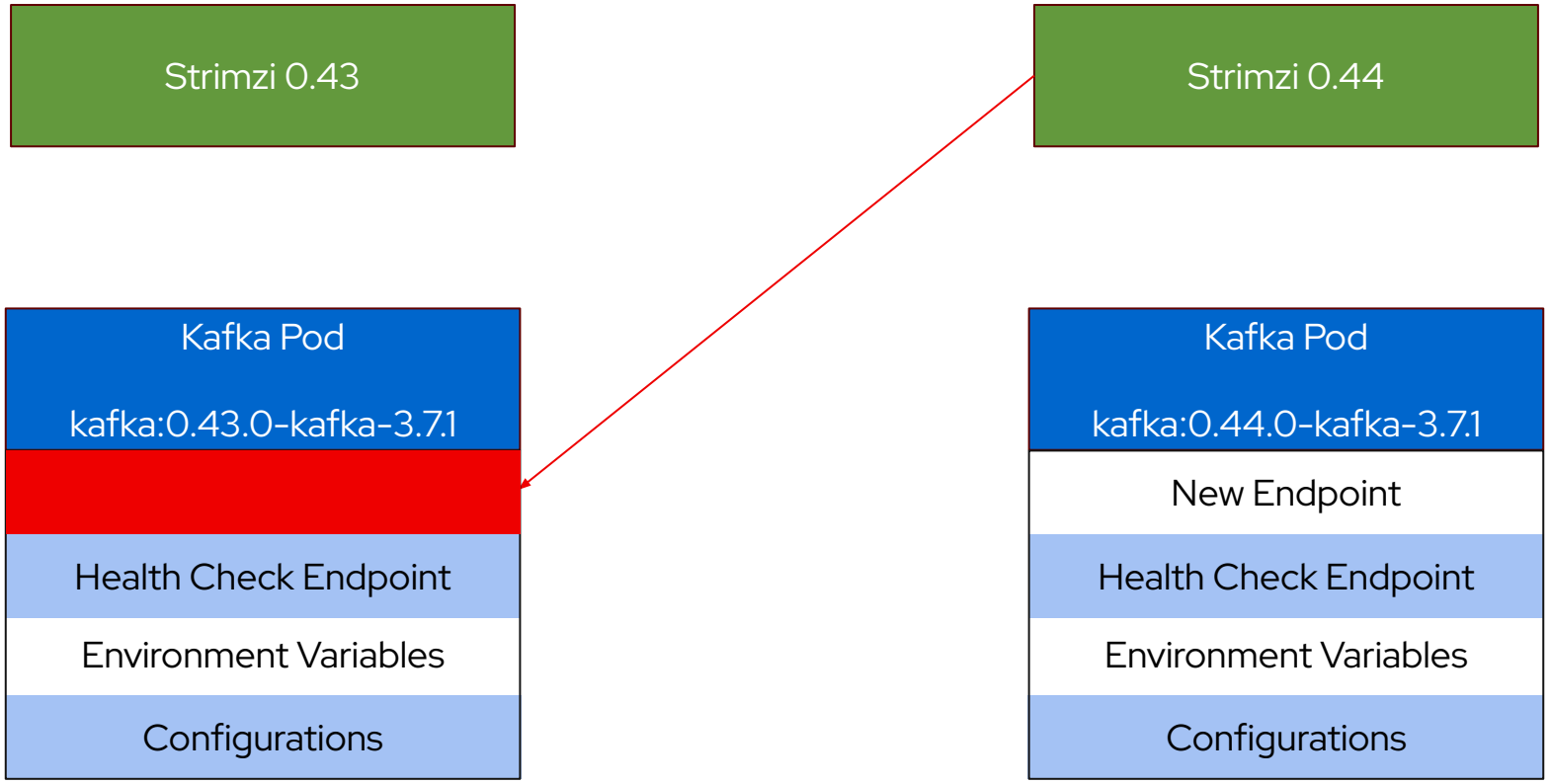
- https://github.com/tomncooper/strimzi-phased-upgrade

Strimzi Upgrades

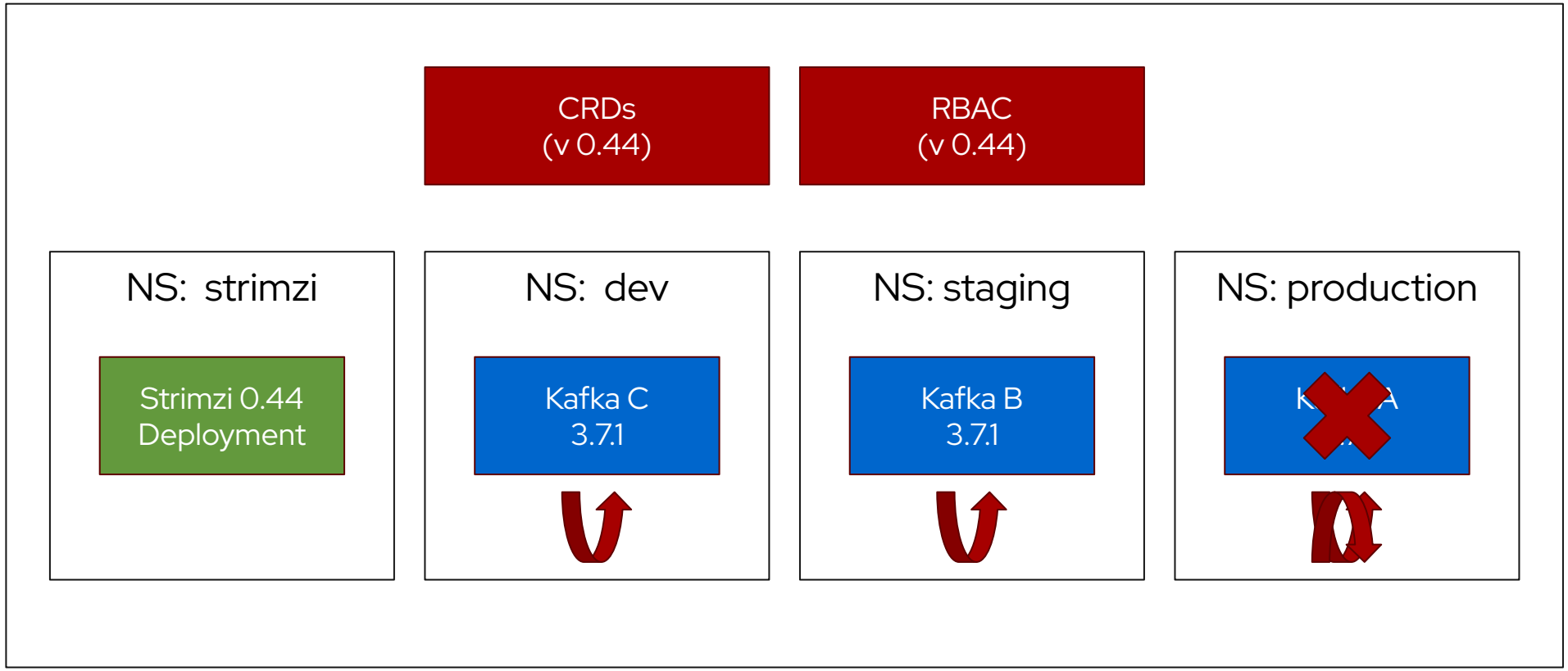


```
apiVersion: apps/v1
kind: Deployment
spec:
  template:
    spec:
      containers:
        - name: strimzi-cluster-operator
          ...
          env:
          ...
          - name: STRIMZI_KAFKA_IMAGES
            value: |
              3.7.0=quay.io/strimzi/kafka:0.43.0-kafka-3.7.0
              3.7.1=quay.io/strimzi/kafka:0.43.0-kafka-3.7.1
              3.8.0=quay.io/strimzi/kafka:0.43.0-kafka-3.8.0
          - name: STRIMZI_KAFKA_CONNECT_IMAGES
            value: |
              3.7.0=quay.io/strimzi/kafka:0.43.0-kafka-3.7.0
              3.7.1=quay.io/strimzi/kafka:0.43.0-kafka-3.7.1
              3.8.0=quay.io/strimzi/kafka:0.43.0-kafka-3.8.0
          ...
```

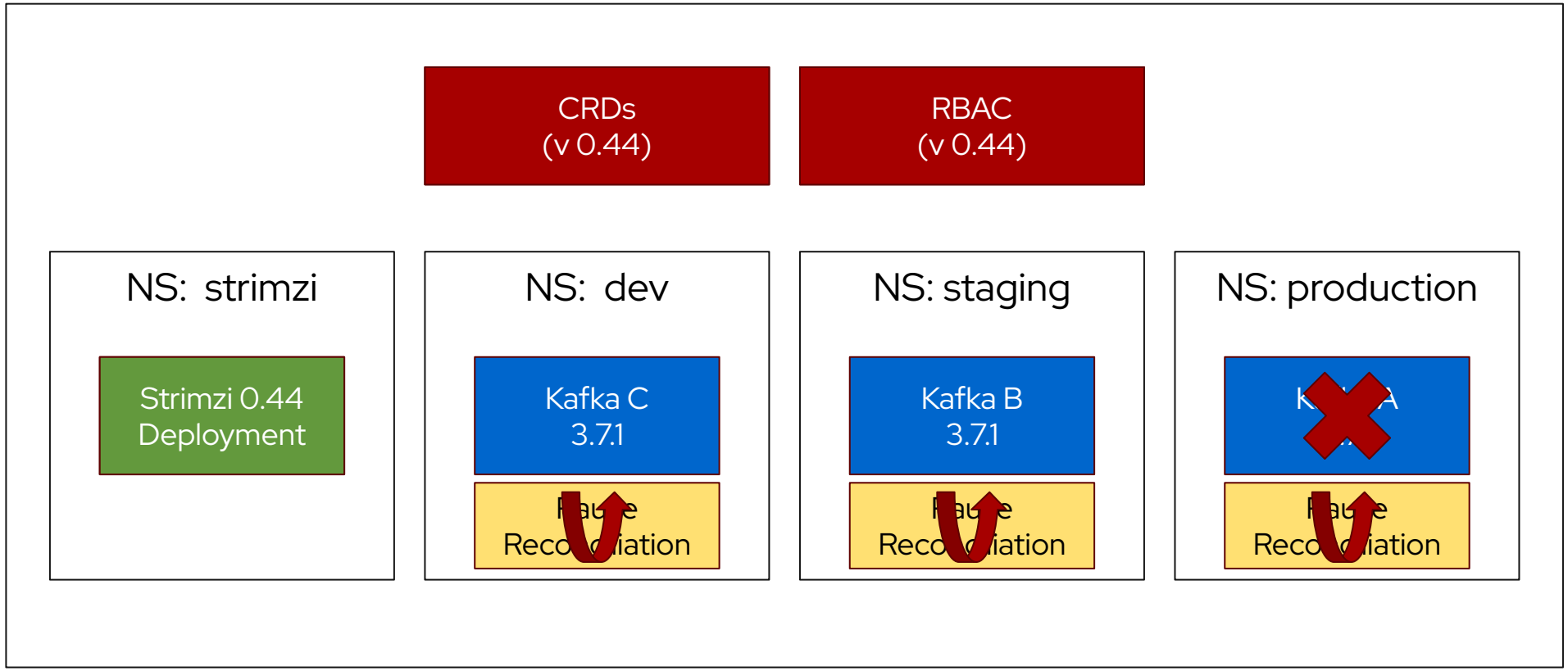

```
apiVersion: apps/v1
kind: Deployment
spec:
  template:
    spec:
      containers:
        - name: strimzi-cluster-operator
          ...
          env:
          ...
          - name: STRIMZI_KAFKA_IMAGES
            value: |
              3.7.0=quay.io/strimzi/kafka:0.44.0-kafka-3.7.0
              3.7.1=quay.io/strimzi/kafka:0.44.0-kafka-3.7.1
              3.8.0=quay.io/strimzi/kafka:0.44.0-kafka-3.8.0
          - name: STRIMZI_KAFKA_CONNECT_IMAGES
            value: |
              3.7.0=quay.io/strimzi/kafka:0.44.0-kafka-3.7.0
              3.7.1=quay.io/strimzi/kafka:0.44.0-kafka-3.7.1
              3.8.0=quay.io/strimzi/kafka:0.44.0-kafka-3.8.0
          ...
```



Strimzi 0.43
Strimzi 0.44
Kafka Pod
kafka:0.43.0-kafka-3.7.1
Health Check Endpoint
Environment Variables
Configurations
Kafka Pod
kafka:0.44.0-kafka-3.7.1
New Endpoint
Health Check Endpoint
Environment Variables
Configurations

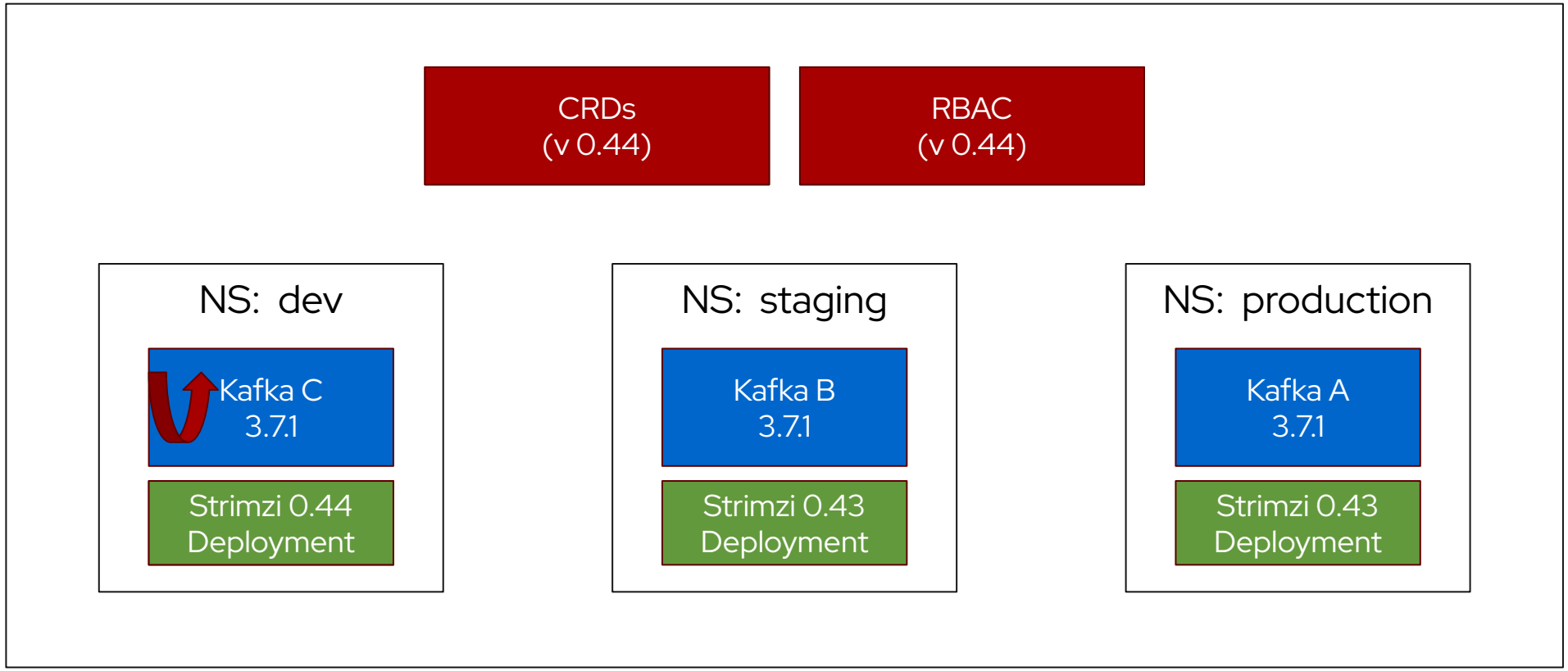
Phased Upgrades



Pause Reconciliation



Operator per Namespace

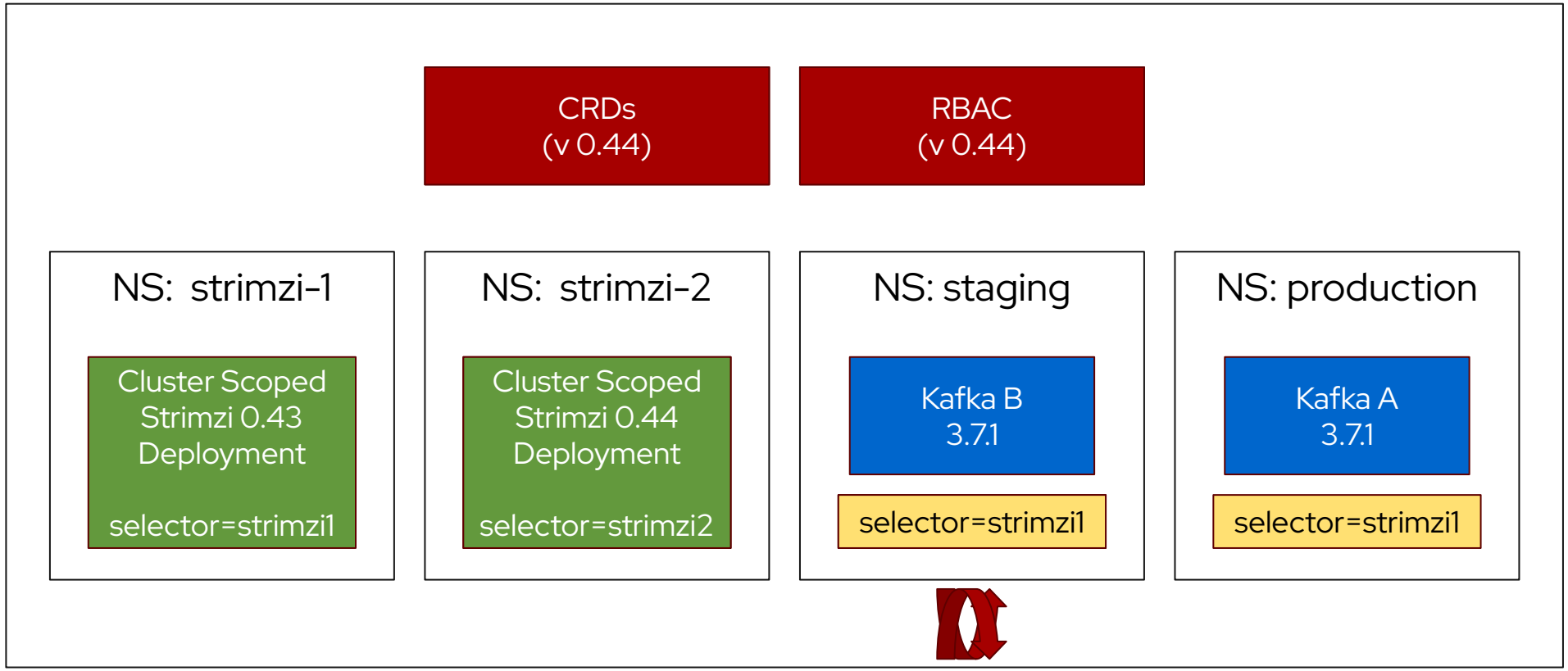


Resource Selector

`STRIMZI_CUSTOM_RESOURCE_SELECTOR` Optional. The label selector to filter the custom resources handled by the Cluster Operator. The operator will operate only on those custom resources that have the specified labels set. Resources without these labels will not be seen by the operator. The label selector applies to `Kafka`, `KafkaConnect`, `KafkaBridge`, `KafkaMirrorMaker`, and `KafkaMirrorMaker2` resources. `KafkaRebalance` and `KafkaConnector` resources are operated only when their corresponding Kafka and Kafka Connect clusters have the matching labels.

```
env:
    - name: STRIMZI_CUSTOM_RESOURCE_SELECTOR
      value: label1=value1,label2=value2
```


Resource Selector



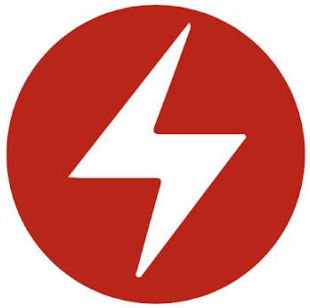
Multiple Concurrent Strimzi Versions

- There can only be **ONE** version of the CRDs installed
- Strimzi's CRDs are **backwards compatible**. Older Strimzi versions can run on Kubernetes clusters with newer CRDs.
- They are **not forwards compatible**, so the installed CRDs need to match the most recent version of Strimzi you have installed.
- Compatibility is only guaranteed between **ADJACENT VERSIONS**
- CRD Kubernetes API versions upgrades (eg v1beta1 to v1beta2) are a special case.

Multiple Concurrent Strimzi Versions

- We need a way to deploy 2 adjacent versions of Strimzi
- Install the CRD version matching the most recent of the two Strimzi versions
- Alter RBAC and other installation resources so that don't collide or interfere
- Each Strimzi deployment should have a unique Custom Resource Selector label

Multiple Concurrent Strimzi Versions



OPERATOR
LIFECYCLE MANAGER

Multiple Concurrent Strimzi Versions


```
install-files
├── cluster-operator-0.43.0
│   ├── 010-ServiceAccount-strimzi-cluster-operator.yaml
│   ├── …
│   ├── 060-Deployment-strimzi-cluster-operator.yaml
│   ├── patches
│   │   ├── patch-deployment.yaml
│   │   └── patch-role-references.yaml
│   ├── kustomization.yaml
├── cluster-operator-0.44.0
│   ├── 010-ServiceAccount-strimzi-cluster-operator.yaml
│   ├── …
│   ├── 060-Deployment-strimzi-cluster-operator.yaml
│   ├── patches
│   │   ├── patch-deployment.yaml
│   │   └── patch-role-references.yaml
│   ├── kustomization.yaml
├── crds
│   ├── 04A-Crd-kafkanodepool.yaml
│   ├── …
│   └── kustomization.yaml
└── kustomization.yaml
```


Multiple Concurrent Strimzi Versions

install-files/cluster-operator-0.43.0/kustomization.yaml:

```
resources:
  - 010-ServiceAccount-strimzi-cluster-operator.yaml
  - …
  - 060-Deployment-strimzi-cluster-operator.yaml

nameSuffix: '-0-43-0'

patches:
  - path: patches/patch-deployment.yaml
    target:
      kind: Deployment
      name: strimzi-cluster-operator

  - path: patches/patch-role-references.yaml
    target:
      kind: (RoleBinding|ClusterRoleBinding)
      name: ".*"
```


Multiple Concurrent Strimzi Versions

install-files/cluster-operator-0.43.0/patches/Patch-deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: strimzi-cluster-operator
spec:
  template:
    spec:
      containers:
        - name: strimzi-cluster-operator
          env:
            - name: STRIMZI_NAMESPACE
              value: "*"
              valueFrom: null
            - name: STRIMZI_CUSTOM_RESOURCE_SELECTOR
              value: strimzi-resource-selector=strimzi-0-43-0
            - name: STRIMZI_LEADER_ELECTION_LEASE_NAME
              value: "strimzi-cluster-operator-0-43-0"
```

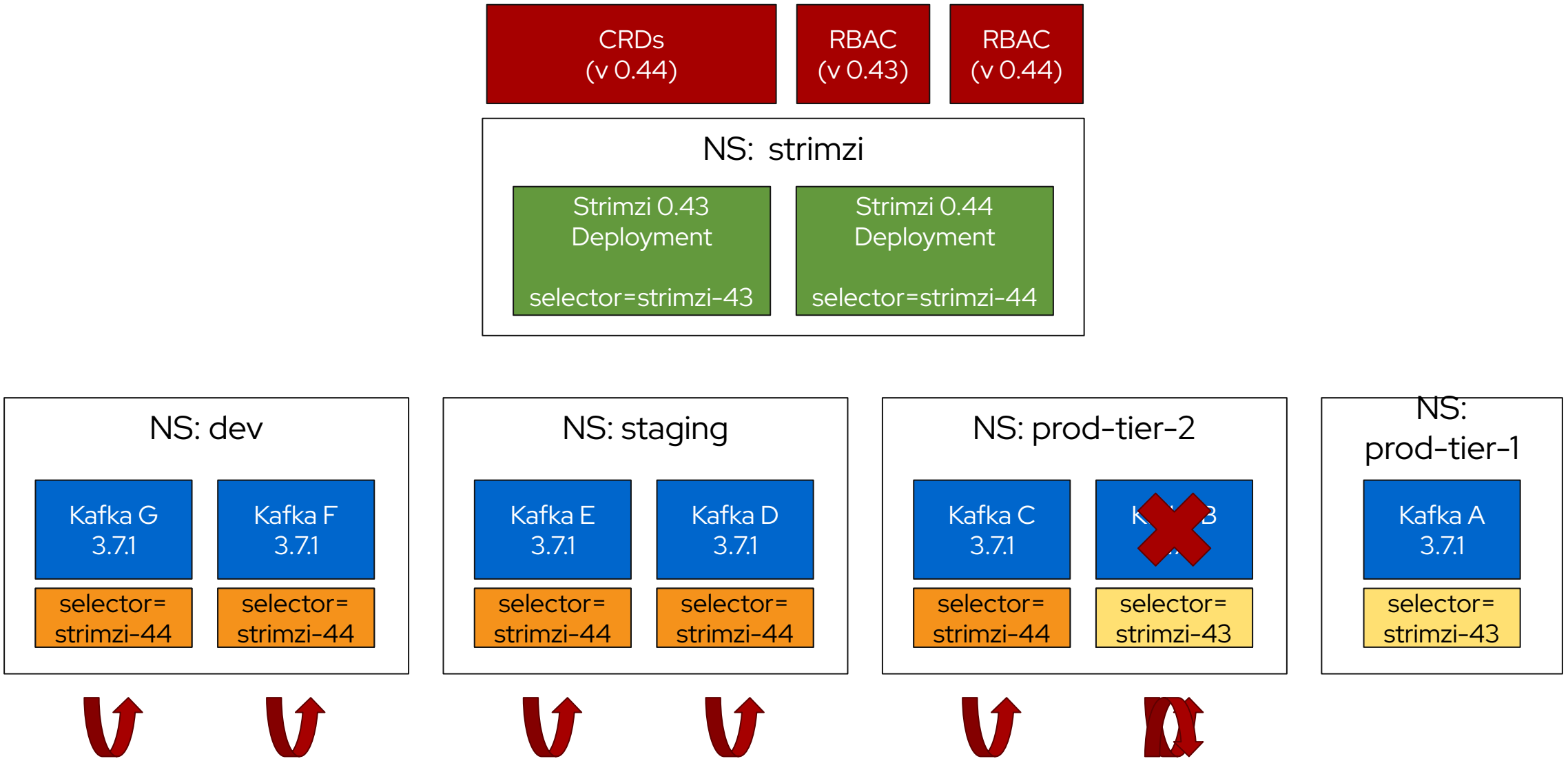
install-files/cluster-operator-0.43.0/patches/patch-role-references.yaml:

```
- op: replace
  path: /subjects/0/name
  value: strimzi-cluster-operator-0-43-0
- op: replace
  path: /subjects/0/namespace
  value: strimzi
```

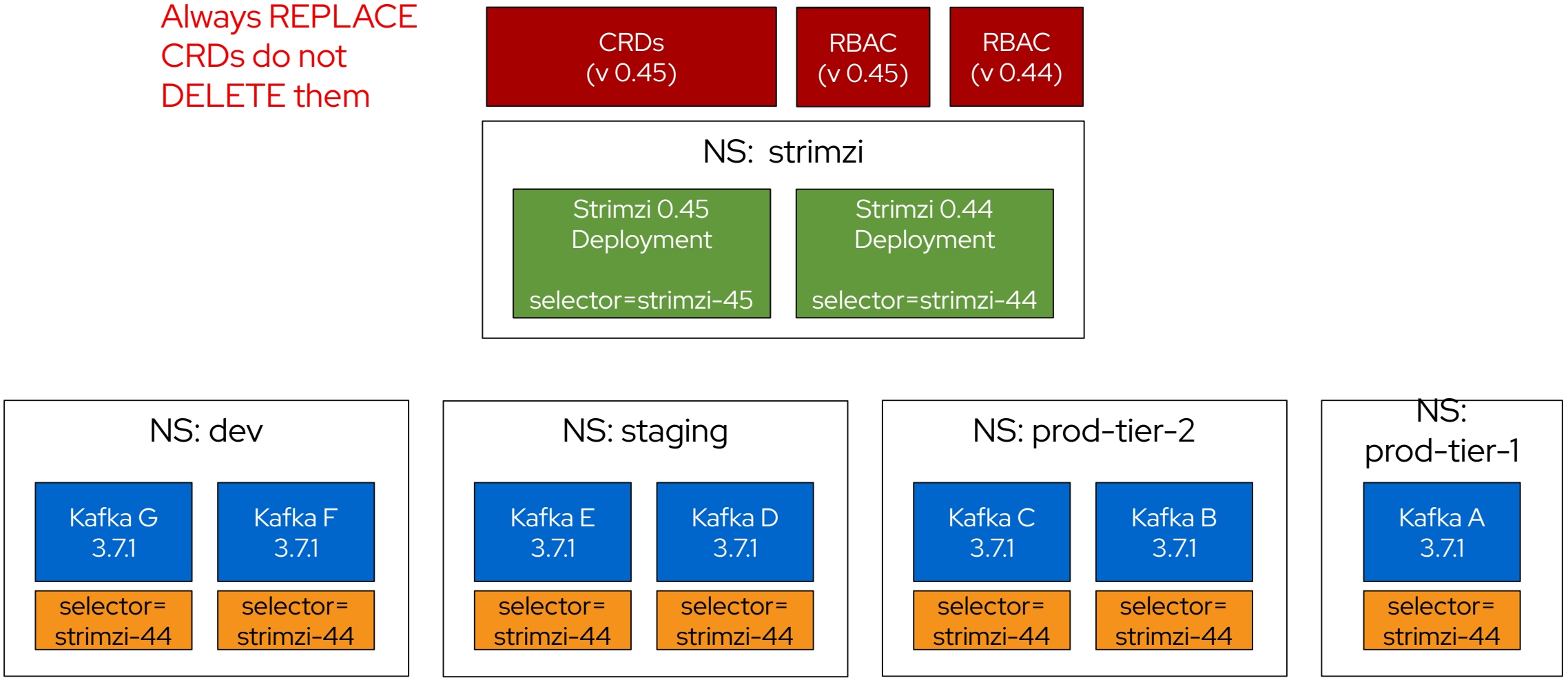

Multiple Concurrent Strimzi Versions


```
install-files
├── cluster-operator-0.43.0
│   ├── 010-ServiceAccount-strimzi-cluster-operator.yaml
│   ├── …
│   ├── 060-Deployment-strimzi-cluster-operator.yaml
│   ├── patches
│   │   ├── patch-deployment.yaml
│   │   └── patch-role-references.yaml
│   └── kustomization.yaml
├── cluster-operator-0.44.0
│   ├── 010-ServiceAccount-strimzi-cluster-operator.yaml
│   ├── …
│   ├── 060-Deployment-strimzi-cluster-operator.yaml
│   ├── patches
│   │   ├── patch-deployment.yaml
│   │   └── patch-role-references.yaml
│   └── kustomization.yaml
├── crds
│   ├── 04A-Crd-kafkanodepool.yaml
│   ├── …
│   └── kustomization.yaml
└── kustomization.yaml
```

```
kubectl -n strimzi apply -k install-files
```



CRDs
(v 0.44)
RBAC
(v 0.43)
RBAC
(v 0.44)
NS: strimzi
Strimzi 0.43
Deployment
selector=strimzi-43
Strimzi 0.44
Deployment
selector=strimzi-44
NS: dev
Kafka G
3.7.1
selector=
strimzi-44
Kafka F
3.7.1
selector=
strimzi-44
NS: staging
Kafka E
3.7.1
selector=
strimzi-44
Kafka D
3.7.1
selector=
strimzi-44
NS: prod-tier-2
Kafka C
3.7.1
selector=
strimzi-44
selector=
strimzi-43
NS:
prod-tier-1
Kafka A
3.7.1
selector=
strimzi-43

Always REPLACE CRDs do not DELETE them



CRD API Version Upgrades

- There is a special case when you move CRD Kubernetes API version
- For example moving from v1beta1 in Strimzi 0.21 to v1beta2 in Strimzi 0.23
- You need to move ALL resources onto the version below the API upgrade as that supports both API versions
- Remove the old version that only supports the old API version
- Perform the CRD API upgrade steps
- Add the new Strimzi version with the new API version

Wrapping up

- Use the **STRIMZI_CUSTOM_RESOURCE_SELECTOR** environment variable to allow you to limit what CRs an particular operator manages
- Install 2 adjacent Strimzi versions concurrently:
  - One version of the CRDs installed from the latest Atrimzi version
  - Isolate the install resources from each other in your deployment system
  - Only adjacent Strimzi versions are a tested configuration
- Hand Kafka CRs off between them with labels
- Rollbacks are just a simple label change
- Take care with CRD upgrades!
- You can also use this setup to have resource isolation between multiple deployments of the same Strimzi version.
- https://strimzi.io/blog/2025/04/10/phased-strimzi-upgrade-example/
- https://github.com/tomncooper/strimzi-phased-upgrade