Service

“ a method for exposing a network application that is running as one or more Pods in your cluster”

Demo

`k8s/03_service/multiple_services.yaml`
`(DNS resolution)`

Service vs Deployment

DevOps perspective:

microservice = k8s Service

Developer perspective:

microservice = k8s Deployment

Service = it’s mainly about networking

Deployment = it’s mainly about a workload

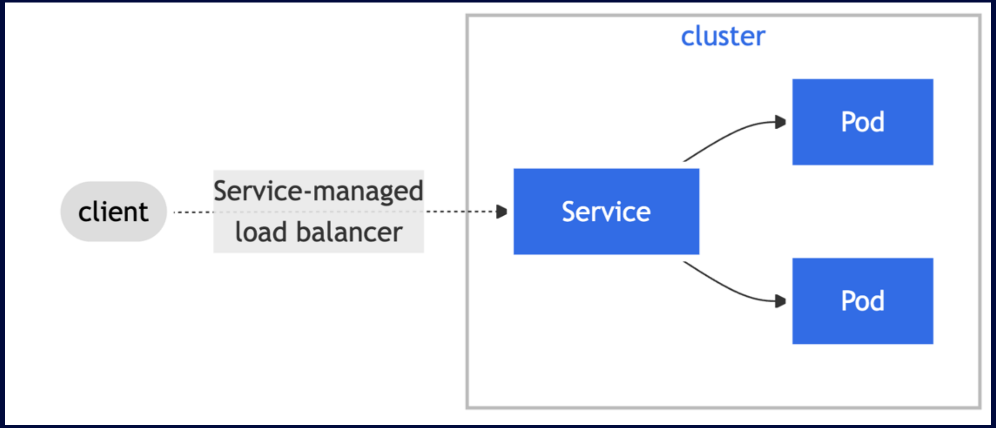
Ingress

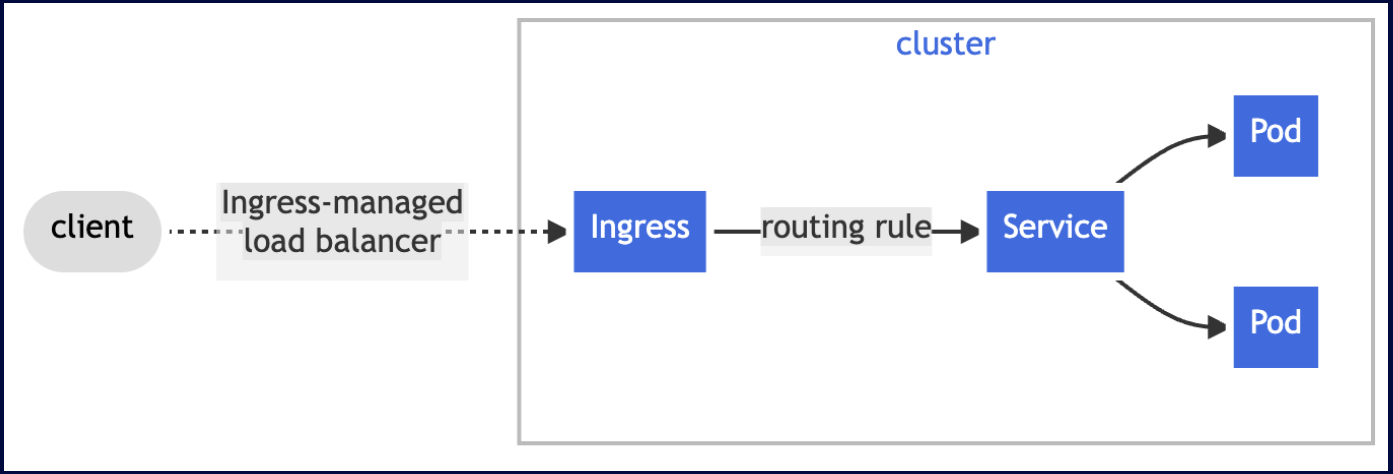
“lets you map traffic to different backends”

Service = handles just 1 Deployment

Ingress – handles 1..N Deployments (via Services)

Ingress vs Service





Demo

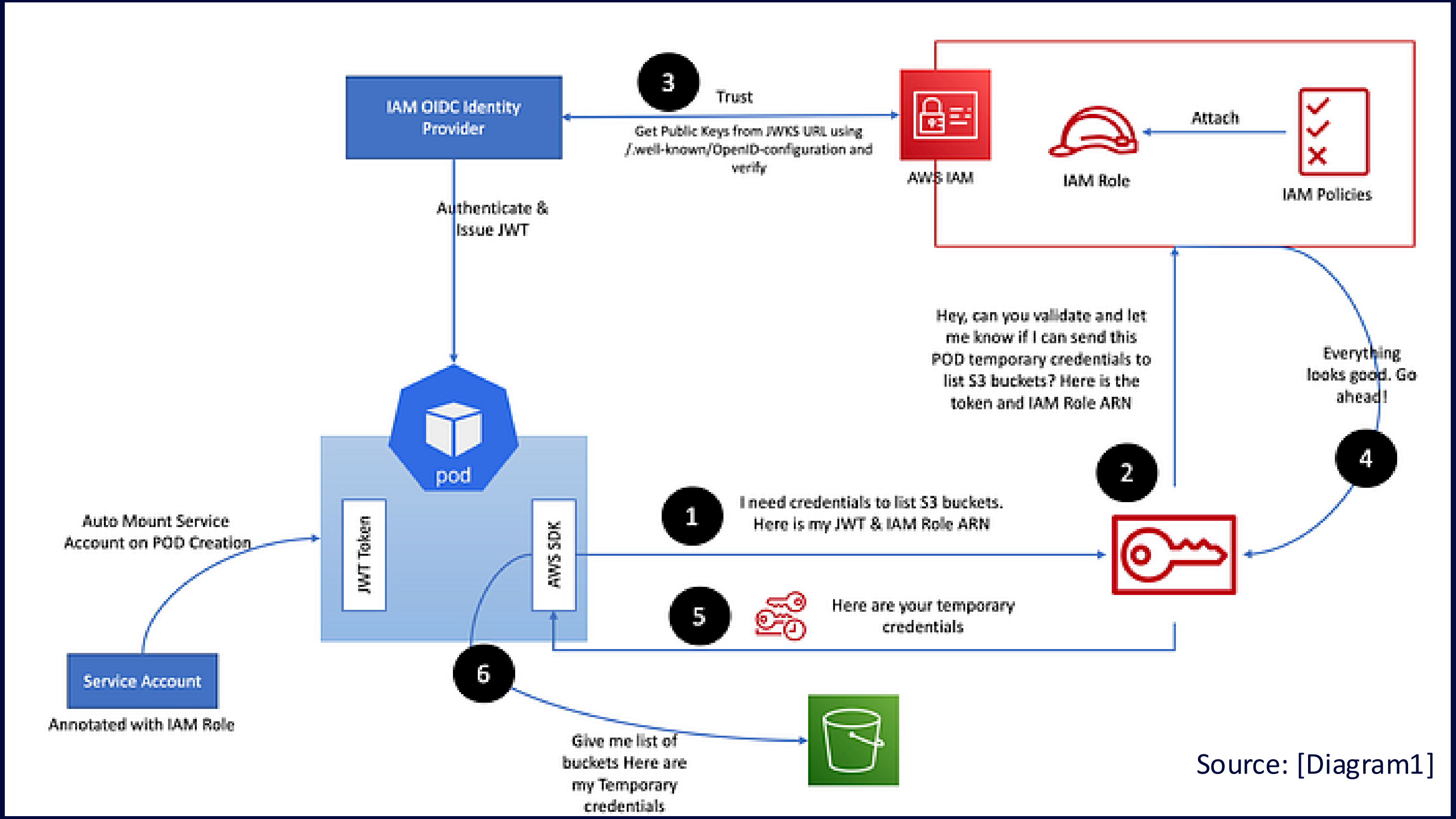
`k8s/04_ingress/two_services.yaml`

Service Accounts

“Application Pods, system components, […] can use a specific ServiceAccount's credentials to identify as that ServiceAccount.”

i.e Pods can identify as a specific ServiceAccount

IRSA = IAM Roles for Service Accounts



IAM OIDC Identity Provider
3
Trust
Get Public Keys from JWKS URL using /.well-known/OpenID-configuration and verify
AWS IAM
IAM Role
Attach
IAM Policies
Authenticate & Issue JWT
Hey, can you validate and let me know if I can send this POD temporary credentials to list S3 buckets? Here is the token and IAM Role ARN
Everything looks good. Go ahead!
pod
2
4
Auto Mount Service Account on POD Creation
JWT Token
AWS SDK
1
I need credentials to list S3 buckets. Here is my JWT & IAM Role ARN
5
Here are your temporary credentials
Service Account
Annotated with IAM Role
6
Give me list of buckets Here are my Temporary credentials

Source: [Diagram1]

Amazon EKS Pod Identity Webhook

ServiceAccount (sa-test) annotated with:

```
eks.amazonaws.com/role-arn: "arn:aws:iam::{account-id}:role/{iam-role}"
```

(new) Pod using:

```
spec.serviceAccountName: sa-test
```

Injections (by web hook):

```
Environment variables:
        - AWS_DEFAULT_REGION
        - AWS_REGION
        - AWS_ROLE_ARN
        - AWS_WEB_IDENTITY_TOKEN_FILE
        - AWS_STS_REGIONAL_ENDPOINTS
Volumes:
    - var/run/secrets/eks.amazonaws.com/serviceaccount/token
```

```kotlin
val client = S3Client
    .builder().credentialsProvider(
        StsAssumeRoleWithWebIdentityCredentialsProvider.builder().build()
    )
    .region(US_EAST_1)
    .build()
```

```java
private StsWebIdentityTokenFileCredentialsProvider(Builder builder) {
    super(builder, "sts-assume-role-with-web-identity-credentials-provider");
    Path webIdentityTokenFile =
        builder.webIdentityTokenFile != null
                ? builder.webIdentityTokenFile
                : Paths.get(
                   trim(
                         SdkSystemSetting.AWS_WEB_IDENTITY_TOKEN_FILE.getStringValueOrThrow()
                   )
                );
```

Demo

aws/create/7_setup_irsa.sh

k8s/05_irsa

Helm

package manager for Kubernetes

A ***Chart*** is a Helm package

(Helm uses a packaging format called *charts*. A chart is a collection of files that describe a related set of Kubernetes resources.)

[used by UP]

A ***Repository*** is the place where charts can be collected and shared

[not used by UP]

A ***Release*** is an instance of a chart running in a Kubernetes cluster.

[not used by UP]

User Platform facilitates only one command of Helm

(under the hood – it is called by ArgoCD):

```
helm template
```

GO templates

https://pkg.go.dev/text/template (engine)

https://pkg.go.dev/github.com/Masterminds/sprig (extra template functions - supported out of the box)

"Actions"--data evaluations or control structures--are delimited by "{{" and "}}";

all text outside actions is copied to the output unchanged.

Demo

aws/create/5_create_ingress_controller.sh
aws/create/8_install_keda.sh

(as regular package manager: helm install)

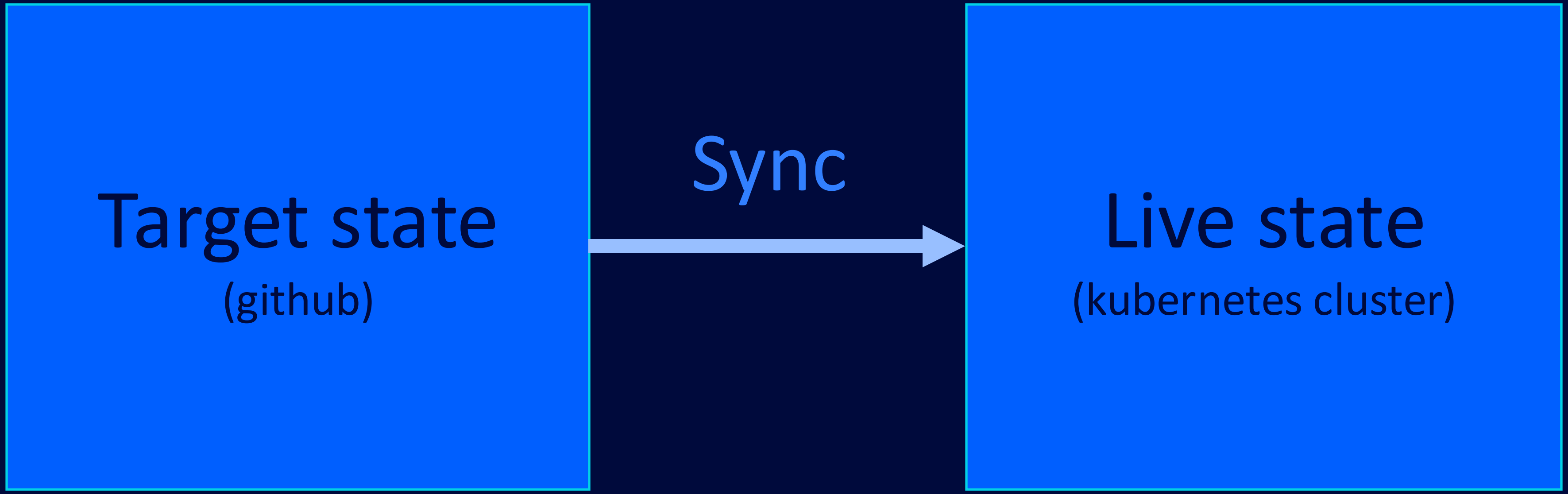
Demo

`k8s/06_helm`

`(helm template)`

ArgoCD

continuous delivery tool for Kubernetes


Target state
(github)
Sync
Live state
(kubernetes cluster)

Demo

`k8s/07_argo`

UP applications setup in EKS

ApplicationSet – generator #1 (tenants)

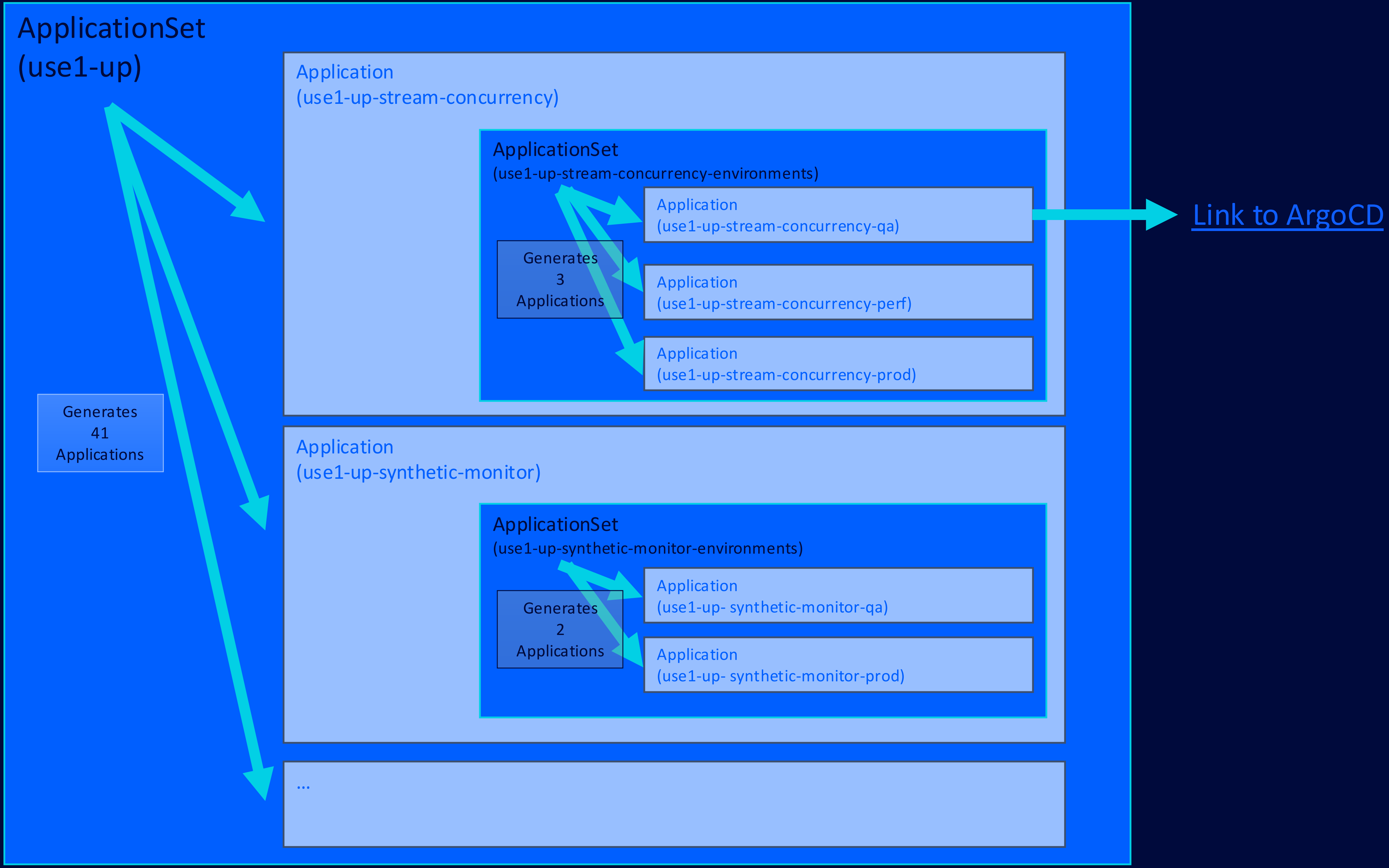
(line 23: - path: "applications/eu-west-1/*.yaml”)

UP applications - input to #1 generator

ApplicationSet – generator #2

(line 18: {{- range $d := .Values.destinations }})

UP environments – input to #2 generator


ApplicationSet
(use1-up)
Generates
41
Applications
Application
(use1-up-stream-concurrency)
ApplicationSet
(use1-up-stream-concurrency-environments)
Generates
3
Applications
Application
(use1-up-stream-concurrency-qa)
Application
(use1-up-stream-concurrency-perf)
Application
(use1-up-stream-concurrency-prod)
Link to ArgoCD
Application
(use1-up-synthetic-monitor)
ApplicationSet
(use1-up-synthetic-monitor-environments)
Generates
2
Applications
Application
(use1-up- synthetic-monitor-qa)
Application
(use1-up- synthetic-monitor-prod)
…

Demo

apply change in GH repo **(source)** and see changes reflected in k8s cluster **(target)**

UP pull request merge

utilities/eks-rolling-update-deploy :

1. Cloning repository https://[...]/viacomcbs/up-k8s-applications.git
2. python3 -u argocd-update-helm-values.py
3. git commit […]
4. git push --set-upstream origin main
5. python3 -u argocd-deployment-validation.py

```
{
    helm_values_file: braze-integration/use1-qa.yaml,
    key_to_update: pod.containers.braze-integration.image,
    container_image: […].[…]/up-braze-integration:0.1.290,
    patch_values: {
        pod: {
            containers: {
                braze-integration: {
                    environmentVariables: {
                        NR_VERSION: 0.1.290
}}}}}
```

```
def restart_argocd_application(argocd_server, argocd_app, token):
    token_opt = f"--auth-token={token}"
    command =  " ".join([
        f"argocd app actions run {argocd_app}",
        "restart --kind Deployment",
        f"{token_opt if token != '' else ''}",
        f"--server {argocd_server}",
        "--grpc-web",
    ])
```

```
argocd app actions run use1-up-braze-integration-qa restart \
  --kind Deployment {token_opt} \
  --server {argocd_server} \
  --grpc-web
```

! Triggers Kubernetes rolling update

UP service restart (e.g. after configuration change)

user-platform/up-eks-application-restart

utilities/argocd-restart-application

! Triggers Kubernetes rolling update

```
argocd app actions run use1-up-braze-integration-qa restart \
  --kind Deployment  --all '--auth-token=****' \
  --server argocd.tools.paramount.tech \
  --grpc-web
```

UP ‘perf’ service horizontal scaling

user-platform/up-scale-perf-eks-services:

1. Cloning repository https://[...]/viacomcbs/up-k8s-applications.git
2. sed -i '/replicas:/s/:.*/: 0/' api-gateway/use1-perf.yaml
3. git commit […]
4. git push

! ArgoCD autosync triggers Kubernetes rolling update

UP autoscaling (Keda)

1. **Open a PR for the service that needs an autoscaling change:**
   - E.g. https://github.com/viacomcbs/up-k8s-applications/blob/main/activation-code/use1-prod.yaml
2. **Merge PR**
3. **Wait for changes to be picked up by ArgoCD autosync**

```yaml
autoscaling:
  keda:
    enabled: true
    maxReplicas: 64
    minReplicas: *replicas
    pollingInterval: 60
    advancedBehavior:
      scaleUp:
        stabilizationWindowSeconds: 120
```

Sources

Docs1 - https://kubernetes.io/docs/home/

Docs2 - https://helm.sh/docs/topics/charts/

Docs3 - https://argo-cd.readthedocs.io/en/stable/

Diagram1 [IRSA] - https://mohaamer5.medium.com/iam-roles-for-service-accounts-with-eks-irsa-good-bye-aws-credentials-1cdf1fa5192

Terraform