



ArgoCon

EUROPE

Argo Rollouts + HPA



ArgoCon
EUROPE

1 April 2025

London, England



Anastasiia Gubska
CNCF Ambassador
SRE/DevOps
BT Group



Kapelonis Kostis
Argo Team
Developer Advocate
Octopus Deploy

QUIZ

Are you familiar with?



Argo Rollouts

  2619

Advanced Kubernetes deployment strategies such as Canary and Blue-Green made easy.

[Learn More](#)

Using in Production?



Argo Rollouts

  2619

Advanced Kubernetes deployment strategies such as Canary and Blue-Green made easy.

[Learn More](#)



What does this affect?

previewReplicaCount: 5

What does this affect?

dynamicStableScale: true

Are they the same thing?



setWeight: 20

**setCanaryScale:
weight: 20**

Who will win?



Argo Rollouts



Horizontal
Autoscaler

Agenda



1. What is Argo Rollouts
2. How the Horizontal Pod Autoscaler works
3. The cost problem for Rollouts
4. Why use Argo Rollouts + HPA
5. Use cases and different scenarios with HPA

Argo Rollouts

Progressive Delivery strategies

Blue/Green Deployment

1 INITIAL VERSION



2 NEW VERSION DEPLOYED



3 SWITCH TRAFFIC



4 FINISH



Canary Release

1 INITIAL VERSION



2 NEW VERSION USED BY 10% OF USERS



3 NEW VERSION USED BY 33% OF USERS



4 NEW VERSION USED BY ALL USERS

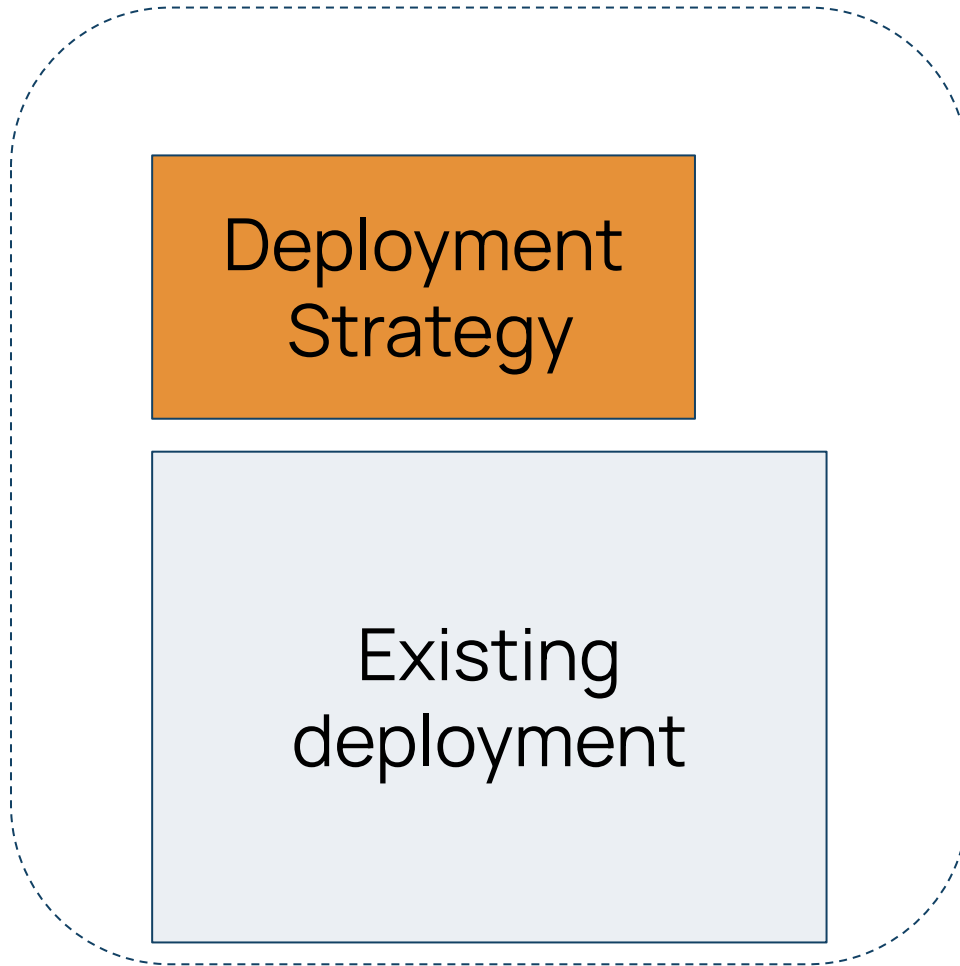


Argo Rollouts

- Kubernetes native
- Standalone project
- Does **NOT** depend on Argo CD
- Blue/Green support
- Canary support
- A/B testing and other Experiments
- Zero downtime releases
- Automatic rollbacks based on metrics
- Installed on each deployment cluster

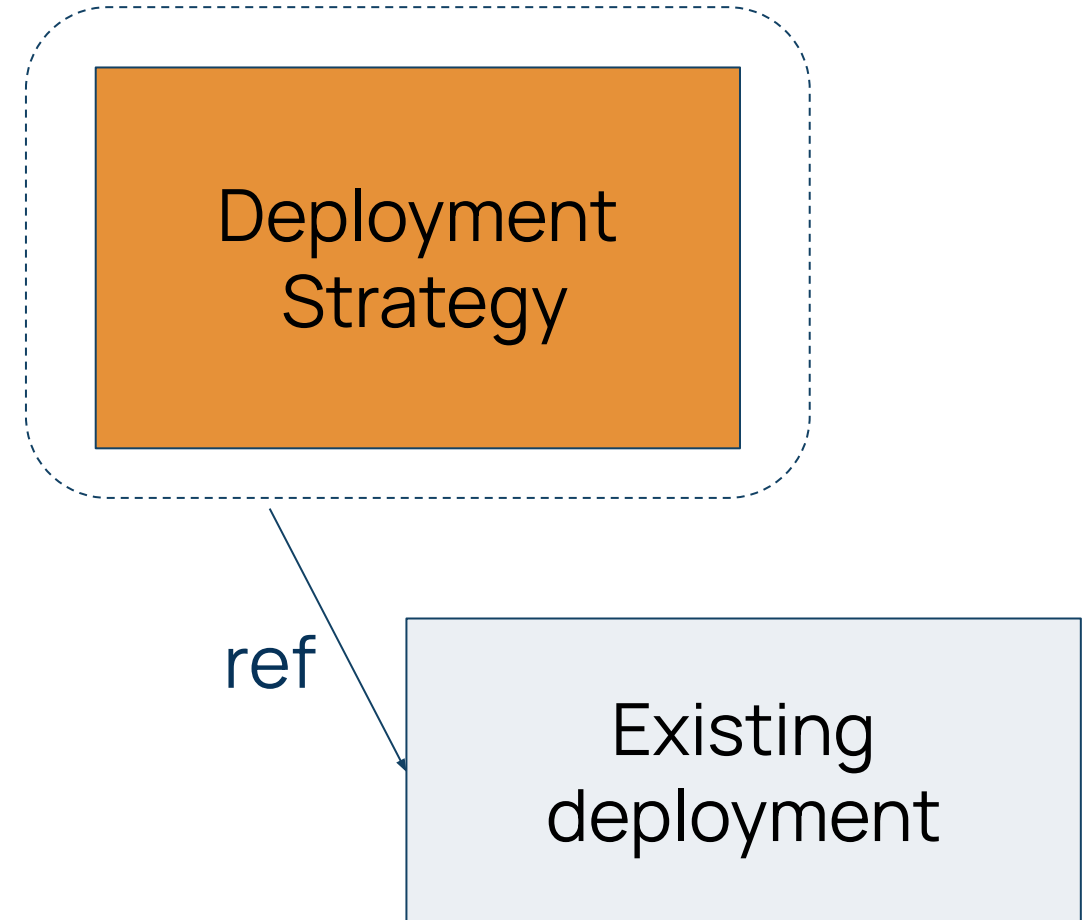


How Rollouts work



Rollout CRD

Rollout CRD



Rollout manifest

```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: canary-example
spec:
  replicas: 10
  revisionHistoryLimit: 2
  selector:
    matchLabels:
      app: cost-demo
  template:
    metadata:
      labels:
        app: cost-demo
    spec:
      containers:
        - name: cost-demo
          image: ghcr.io/kostis-codefresh/rollouts-autoscaling-example:v1
          ports:
            - name: http
              containerPort: 8080
              protocol: TCP
      strategy:
        canary:
          canaryService: argo-rollouts-canary-service
          stableService: argo-rollouts-stable-service
          steps:
            - setWeight: 10
            - pause:
                duration: 1h
            - setWeight: 20
            - pause: {}
```

```
strategy:
  canary:
    canaryService: argo-rollouts-canary-service
    stableService: argo-rollouts-stable-service
    steps:
      - setWeight: 10
      - pause:
          duration: 1h
      - setWeight: 20
      - pause: {}
```

Rollout = deployment PLUS strategy

Graphical dashboard



rollouts-demo 

 Restart

 Retry

 Abort

 Promote

 PromoteFull

Steps

 Set Weight: 20%

 Pause

 Set Weight: 40%

 Pause: 10s

 Set Weight: 60%

 Pause: 10s

 Set Weight: 80%

 Pause: 10s

Summary

Strategy

 Canary

Step

 1/8

Set Weight

 20

Actual Weight

 20

Containers

 Edit

rollouts-demo

argoproj/rollouts-demo:yellow




Revisions

Revision 2



argoproj/rollouts-demo:yellow

 canary

rollouts-demo-6cf78c66c5



Revision 1

 Rollback



argoproj/rollouts-demo:blue

 stable

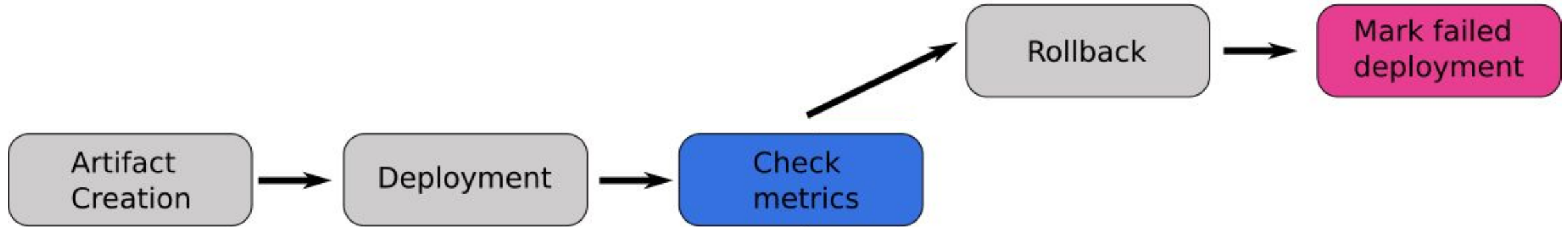
rollouts-demo-687d76d795



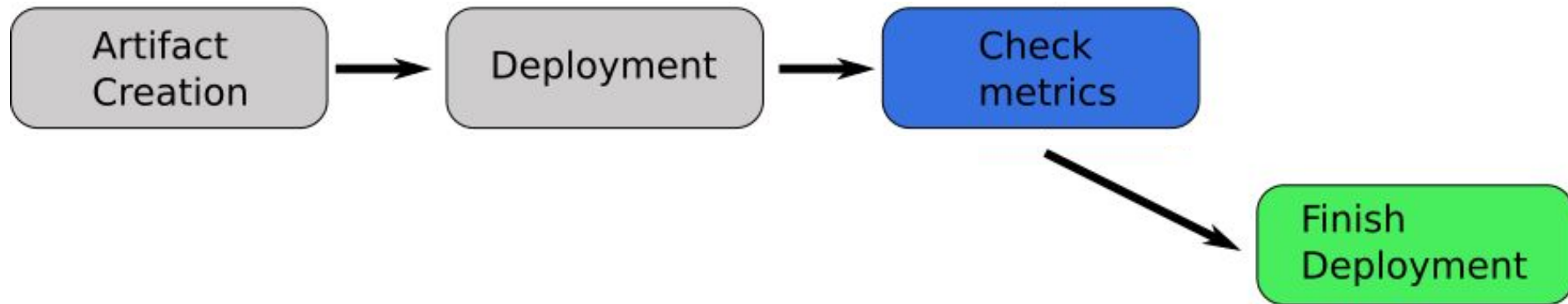
Rollback automatically



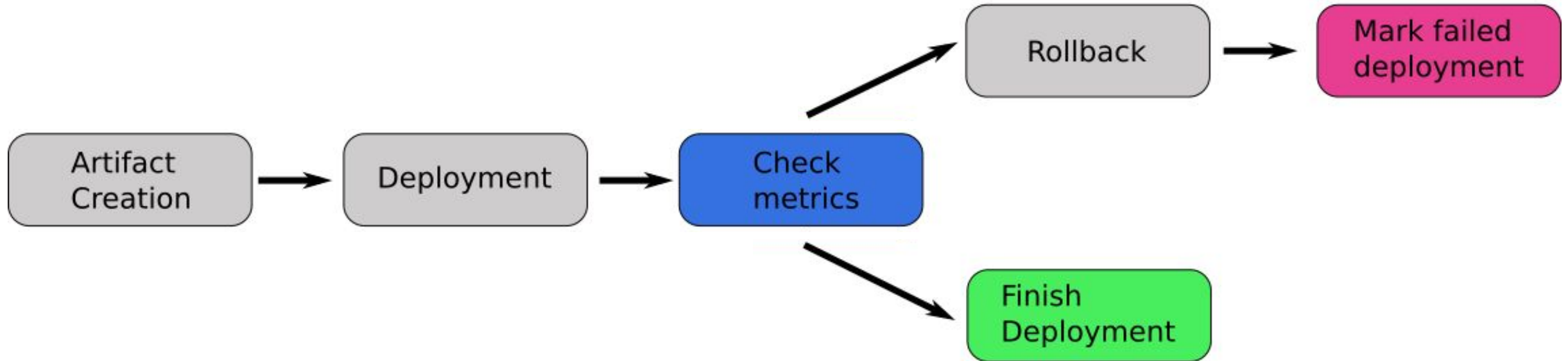
Rollback automatically



Rollback automatically



Rollback automatically



Deploy on Friday at 5:00



Deployment happens at 5.00 pm on Friday

5:15 the whole team is at the pub

Kubernetes Autoscaling (HPA)

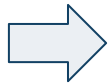
Kubernetes scaling options



- **Horizontal Pod Autoscaler (HPA)**
- Vertical Pod Autoscaler (VPA)
- Cluster Autoscaler
- Custom Metrics Autoscaler
- Kubernetes Event-driven Autoscaler (KEDA)

HPA manifest

- HPA monitors the metrics (like CPU, memory usage or RPS) of the pods and adjust the number of replicas to match the desired state.
- Effective for variable traffic



...



Before HPA

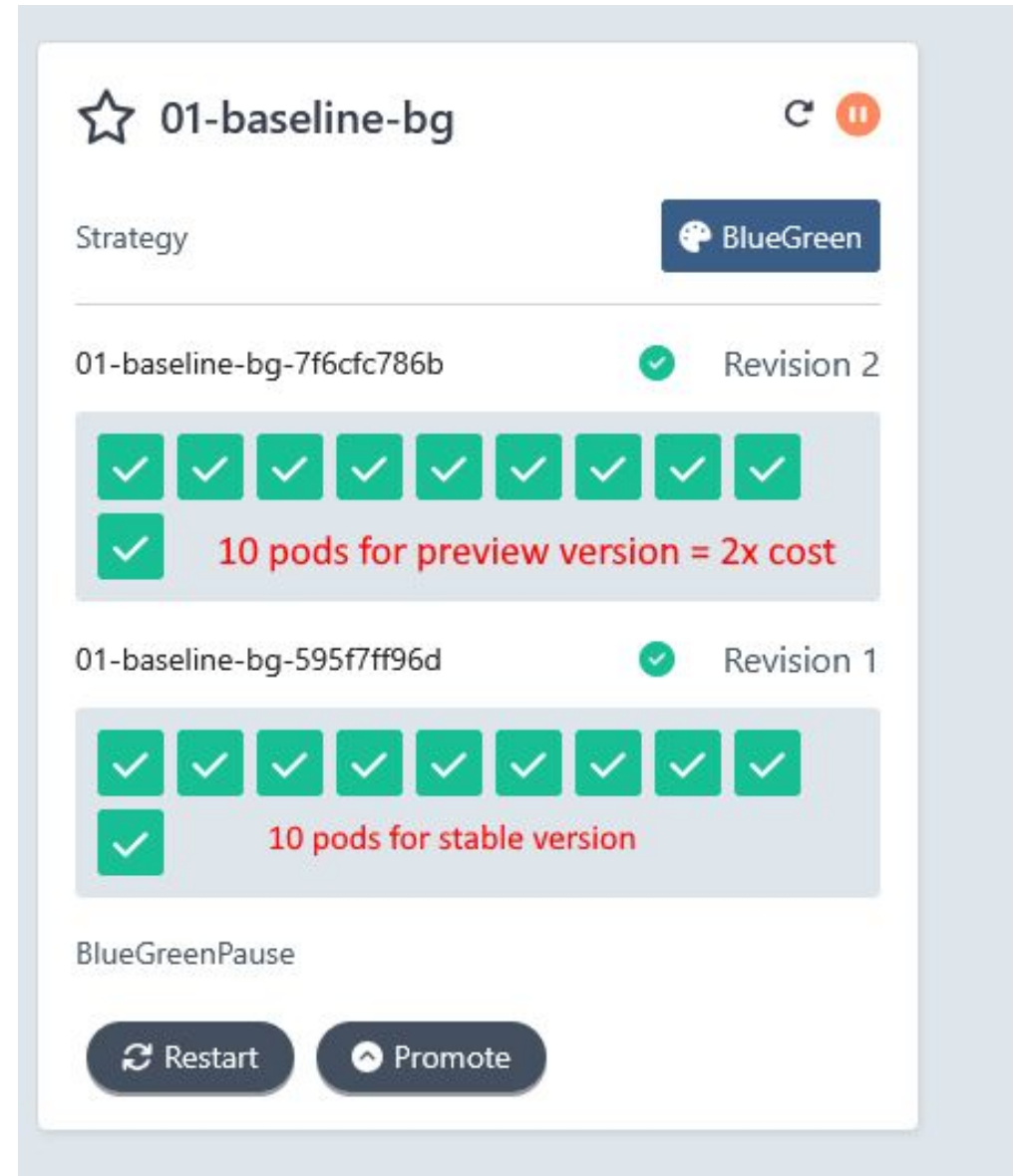
After HPA

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: demo-hpa
  labels:
    app: cost-demo
spec:
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: memory
        target:
          type: AverageValue
          averageValue: 16Mi
```


The problem of cost

Blue/Green is 2x cost

- Launching a new color doubles the cost
- Preview version needed for smoke tests or other integration tests
- In a big company with many applications, costs quickly add up



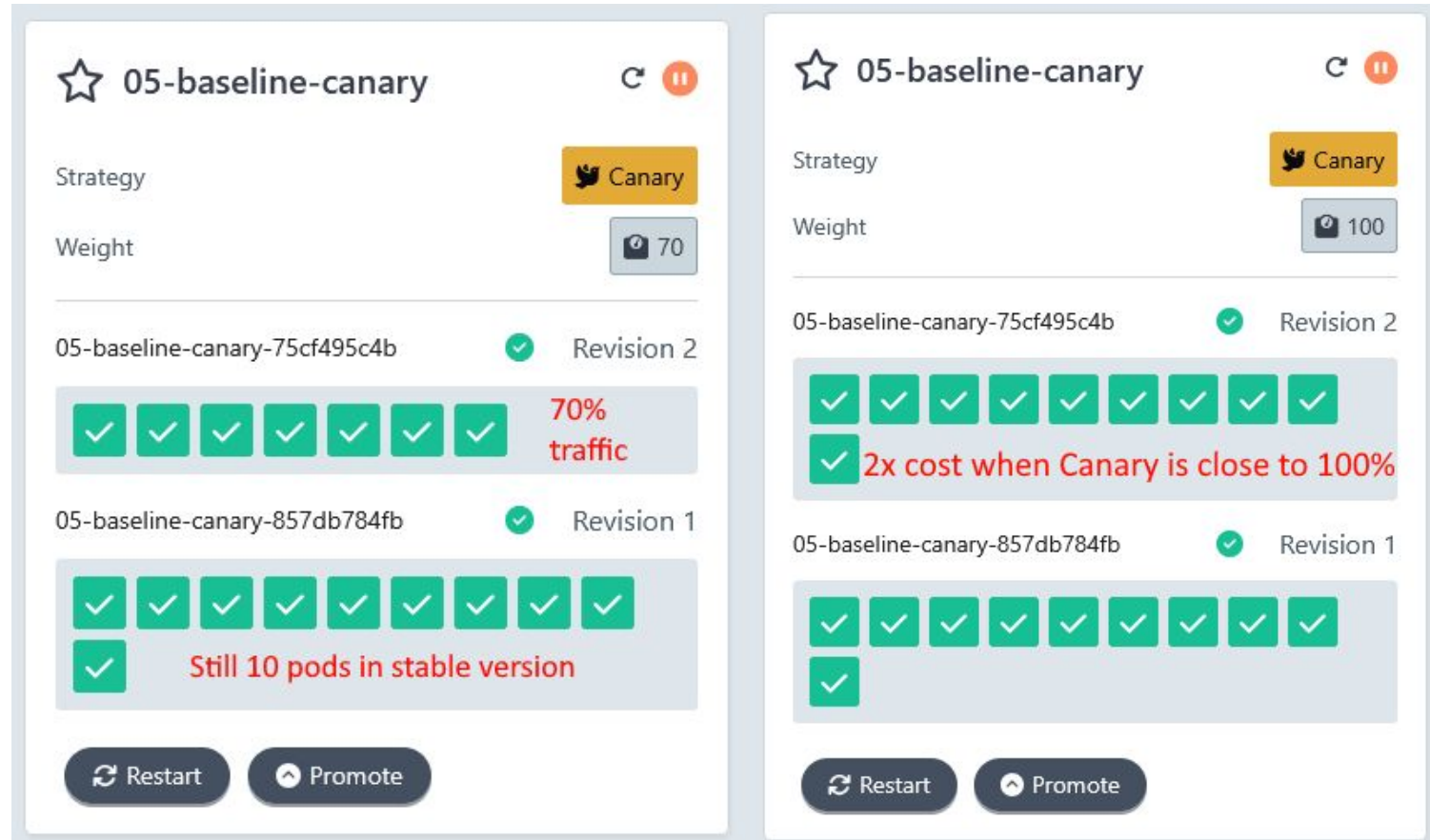
The screenshot displays the Argo CD interface for a deployment named '01-baseline-bg'. The deployment strategy is set to 'BlueGreen'. It shows two revisions:

- Revision 2:** Labeled '01-baseline-bg-7f6cfc786b', it shows 10 pods for the preview version, with a red text annotation stating '10 pods for preview version = 2x cost'.
- Revision 1:** Labeled '01-baseline-bg-595f7ff96d', it shows 10 pods for the stable version, with a red text annotation stating '10 pods for stable version'.

At the bottom, there are buttons for 'Restart' and 'Promote', and a 'BlueGreenPause' label.

Canary can be 2x cost

- Cost is increasing as canary is progressing
- Noticeable impact for a company with many applications



The image displays two side-by-side screenshots of the Argo Rollouts user interface, illustrating the cost implications of a canary deployment as it progresses.

Left Screenshot (70% traffic):

- 05-baseline-canary** (Revision 2): Shows 7 green checkmarks representing pods, with a label indicating **70% traffic**.
- 05-baseline-canary-857db784fb** (Revision 1): Shows 10 green checkmarks, with a label indicating **Still 10 pods in stable version**.

Right Screenshot (100% traffic):

- 05-baseline-canary** (Revision 2): Shows 10 green checkmarks, with a label indicating **2x cost when Canary is close to 100%**.
- 05-baseline-canary-857db784fb** (Revision 1): Shows 1 green checkmark, indicating that the stable version has been scaled down.

Both screenshots include a **Strategy** dropdown set to **Canary** and a **Weight** input field (70 on the left, 100 on the right). At the bottom of each panel are **Restart** and **Promote** buttons.

Argo Rollouts in the wild



Understand your use case

Argo Rollouts is perfect for all progressive delivery scenarios as explained in [the concepts page](#).

You should *NOT* use Argo Rollouts for preview/ephemeral environments. For that use case check the [Argo CD Pull Request generator](#).

The recommended way to use Argo Rollouts is for brief deployments that take 15-20 minutes or maximum 1-2 hours. If you want to run new versions **for days or weeks** before deciding to promote, then Argo Rollouts is probably not the best solution for you.

Keeping parallel releases for long times, complicates the deployment process a lot and opens several questions where different people have different views on how Argo Rollouts should work.

People really do want to
tests canaries for
days/weeks 😞

<https://argo-rollouts.readthedocs.io/en/latest/best-practices/>

Autoscaling with Argo Rollouts

Argo Rollouts + HPA



1. People want to reduce costs for their applications
2. Complete control over preview/stable pods
3. Current Argo Rollouts HPA documentation is lacking
4. Not really clear what happens when both active

Can I disable HPA while the argo rollout is progressing? #3925

Unanswered sungho-rpls asked this question in Q&A



sungho-rpls on Oct 30, 2024

HPA scales out replicas in my environment because of high CPU usage while the argo rollout progresses. High CPU usage for startup time is acceptable for me, but too many replicas by HPA cause problems with latency, resource usage, and database connections. Is it possible to disable HPA scaling when the argo rollout is progressing?



Jacob Hagstedt Dec 19th, 2022 at 2:36 PM

Perhaps one thing that is unclear to me is how Argo Rollouts is handling a HPA during a canary rollout?



Jaysheel Shah Mar 11th at 9:51 PM

Hello. I was working on implementing Argo Rollouts for my applications and I noticed an odd behavior with the HPA configuration. If a HPA config is added for an argo rollout, it will only auto-scale the pods in replica sets that are tied to the blue version (current/latest version) or previously deployed replica sets (old blue versions). The replica set associated with the preview version (green version) will not be auto scaled. Is there any way to configure the same autoscaling behavior with the preview version as well without working with the `previewReplicaCount` each time I want to update the number of replicas? Any guidance/suggestions you may have is appreciated. Thank you.



Sam J Jan 6th at 10:39 PM

Hey all, would really appreciate some insight here. I'm trying to use the canary strategy with no traffic management. I have the following `strategy`:

```
canary:
  maxSurge: 10
  maxUnavailable: 0
  steps:
    - setWeight: 50
    - pause:
        duration: 5m
    - setWeight: 100
```



crumb 8:09 PM

Hello, I've tried using `dynamicStableScale: true` in a canary and when I aborted the rollout it seems the stable scaled itself up to nearly double of what my HPA maximum was.

I've been having issues with rollouts scaling way up during the setWeight / pause period. Is there anything I can do to mitigate running more pods than necessary?

...along with an `HPA` with `minReplicas=10` and `maxReplicas=20`. (Note that I am not using traffic management features!)



Shruthi 12:53 AM

Hi, I am trying to use Argo Rollouts with deployment that has HPA configured.

1. I have updated HPA's `scaleTargetRef` Kind to Rollout so the rollout is aware of the HPA otherwise rollout doesn't seem to be working with the HPA
2. With this setting the HPA scales both canary and stable pods resulting in doubling the number of pods, in order to avoid that added `dynamicStableScale: true` setting to rollout.
3. and `dynamicStableScale` only works when traffic routing is enabled so I created 2 services (1 for canary and 1 for stable) and updated rollout to use traffic routing
4. I set replicas to 0 in deployment and replicas to 1 in rollout and let the HPA scale the replicas based on min replica count

Can I get some help validating these steps?



David Curran Nov 6th, 2023 at 5:51 PM

<https://argoproj.github.io/argo-rollouts/migrating/#reference-deployment-from-rollout>

If I reference a deployment in my rollout. How does that work with HPA? I assume HPA will scale my deployment back up but not the rollout? I'd end up with double the pods running?



Jacob Hagstedt Sep 21st, 2022 at 4:03 PM

`hpa` `canary strategy`

Hey! I've noticed that it is possible to use HPA's with rollouts, but how does it actually work? I have a concern about e.g. having a metric like this

```
metrics:
- type: Resource
  resource:
    name: cpu
    target:
      type: Utilization
```

which means we want to scale based on CPU usage. But how does that work when using the canary rollout strategy?

Because we will have both 'stable' and 'canary' pods up and running. What I would like to happen is that the HPA only looks at the 'stable' pods for CPU usage. Is that the case?



Lukasz Boldys Apr 21st, 2023 at 4:49 PM

Hi, I tried to google for it but I couldn't find the answer. I have Rollout configured with HPA.

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: app-name-hpa
spec:
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 50
  scaleTargetRef:
    apiVersion: argoproj.io/v1alpha1
    kind: Rollout
    name: app-name
    minReplicas: 5
    maxReplicas: 5
```

And Rollout using Canary strategy. But for some reason when new Canary pods are started the HPA is not scaling down the Stable RS. So in the middle of deployment I can have 4 Canary pods and 5 Stable pods running. Only after deployment is promoted fully the Stable is being scaled down to zero (or rather, Canary becomes stable and it's 5 at that moment). Does anybody have any idea what I might have messed with the configuration?

Several combinations

[Example 01 - Base case for Blue/Green](#)

[Example 02 - Custom number for pods for preview version](#)

[Example 03 - Blue/Green with autoscaling](#)

[Example 04 - Blue/Green with autoscaling and custom number of pods](#)

[Example 05 - Base case for Canary](#)

[Example 06 - Canary without Traffic manager](#)

[Example 07 - Canary with dynamic scaling](#)

[Example 08 - Canary with decoupled traffic split](#)

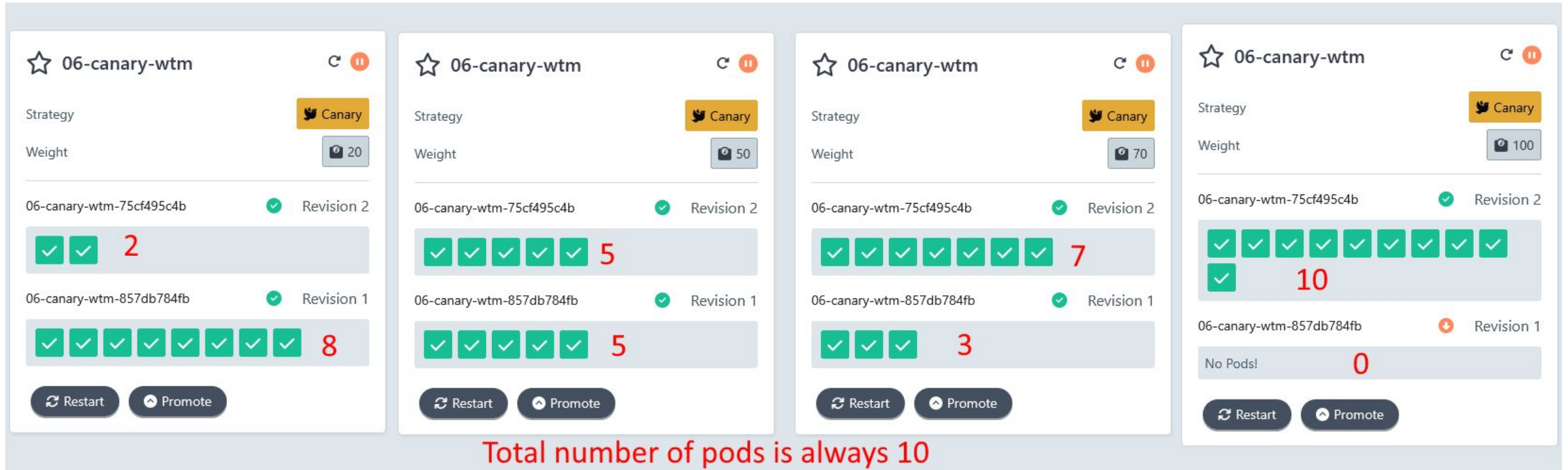
[Example 09 - Canary with autoscaling](#)

[Example 10 - Canary with autoscaling and decoupled traffic split](#)



<https://github.com/kostis-codefresh/rollouts-autoscaling-example>

See visually what happens



<https://github.com/kostis-codefresh/rollouts-autoscaling-example>

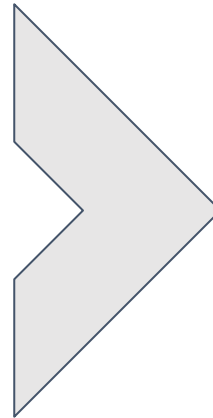


Override HPA decisions

Winner



Argo Rollouts



Horizontal
Autoscaler

Controlling costs without HPA

Pin down number of pods



Blue/Green

Use **previewReplicaCount** Property

This locks down number of pods for new color to specific number

Canaries

Use **dynamicStableScale** property

OR

Use **setCanaryScale** to specify exact number of canary pods

Specify pods for Green color

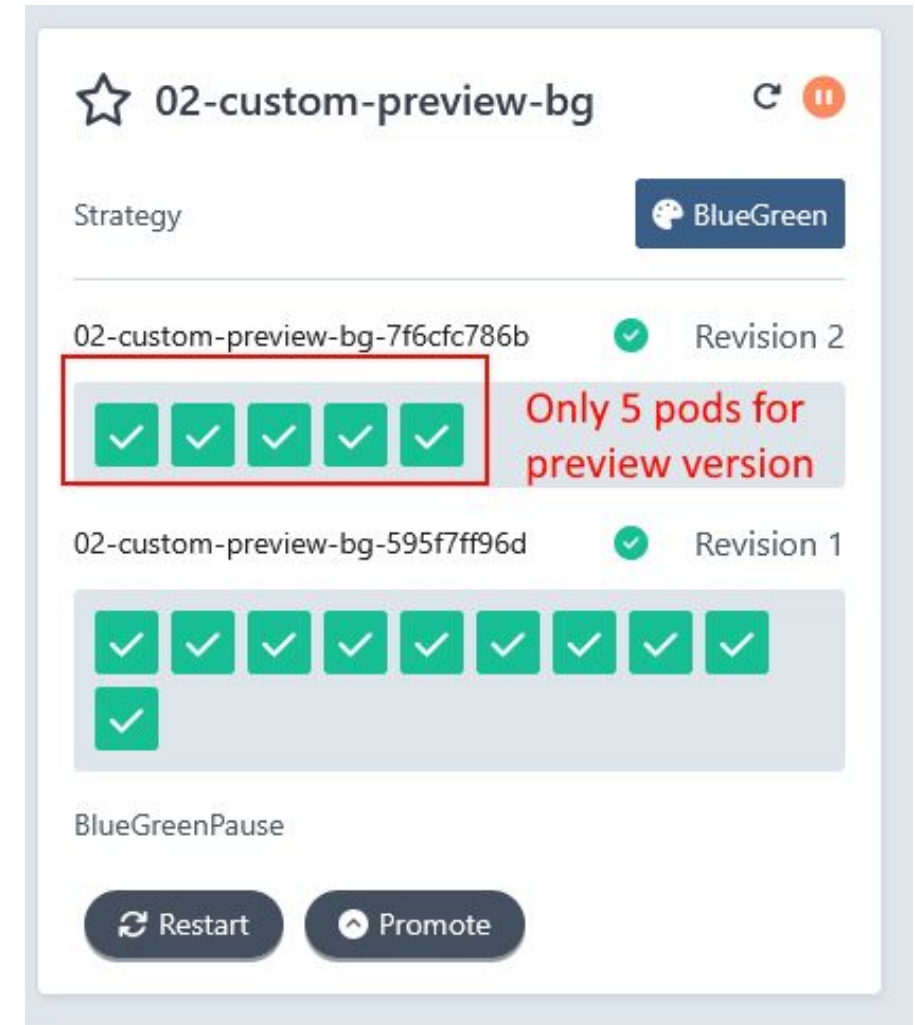
- Cut extra costs by 50% in this example
- Explicitly set the pods of new color to 5



```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: 02-custom-preview-bg
spec:
  replicas: 10
  strategy:
    blueGreen:
      previewService: argo-rollouts-preview-service
      activeService: argo-rollouts-stable-service
      previewReplicaCount: 5
      autoPromotionEnabled: false
  revisionHistoryLimit: 2
  selector:
```

Specify pods for Green color

- Valid only while blue/green is in progress
- Once promoted preview pods will go to 10 (match stable)

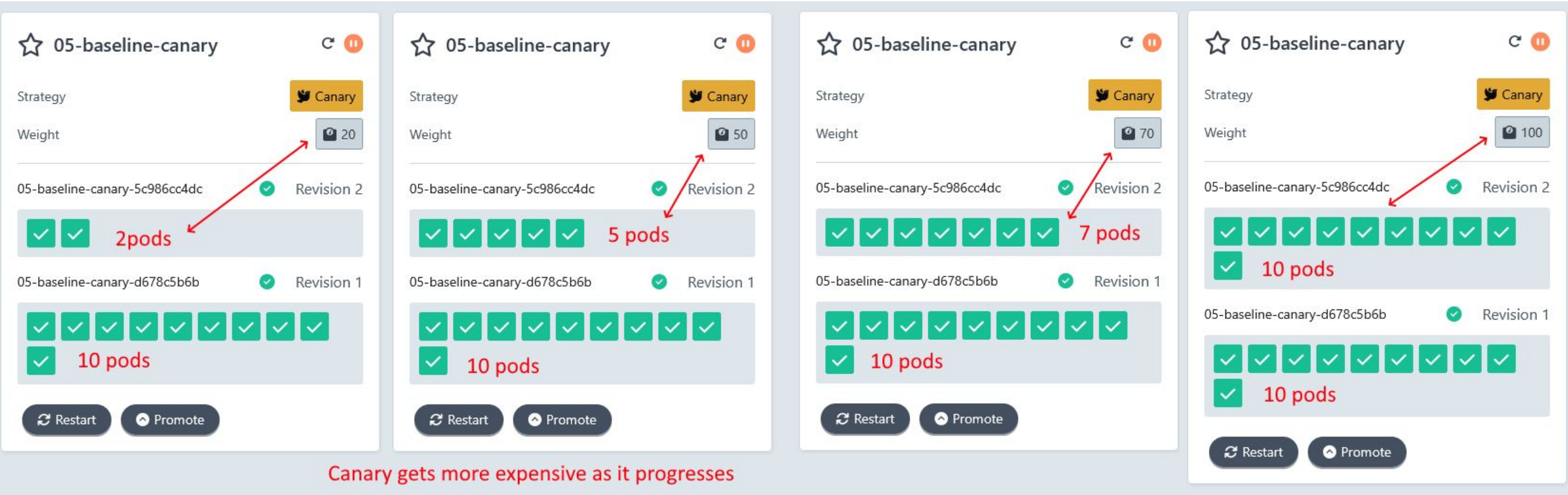


The screenshot shows the Argo Rollouts interface for a deployment named "02-custom-preview-bg". The strategy is set to "BlueGreen". Two revisions are shown:

- Revision 2:** Labeled "02-custom-preview-bg-7f6cfc786b". It shows 5 green checkmarks in a row, which are highlighted by a red box. A red text overlay says "Only 5 pods for preview version".
- Revision 1:** Labeled "02-custom-preview-bg-595f7ff96d". It shows 10 green checkmarks arranged in two rows of five.

At the bottom, there are buttons for "Restart" and "Promote".

Default Canary behavior



The image displays four sequential panels of the Argo CD interface, illustrating the default canary deployment strategy. Each panel shows a deployment named '05-baseline-canary' with a 'Canary' strategy. The weight of the canary revision (Revision 2) increases from 20 to 100 across the panels. The canary revision is shown with a series of green checkmarks representing pods, and the number of pods is displayed in red text. The baseline revision (Revision 1) is shown with a single green checkmark representing a pod. The panels are arranged horizontally, showing the progression of the deployment. Red arrows point from the 'Weight' field to the canary revision's pod count. At the bottom, a red text label states: 'Canary gets more expensive as it progresses'.

Weight	Canary Revision 2 Pods	Baseline Revision 1 Pods
20	2 pods	10 pods
50	5 pods	10 pods
70	7 pods	10 pods
100	10 pods	10 pods

Canary gets more expensive as it progresses

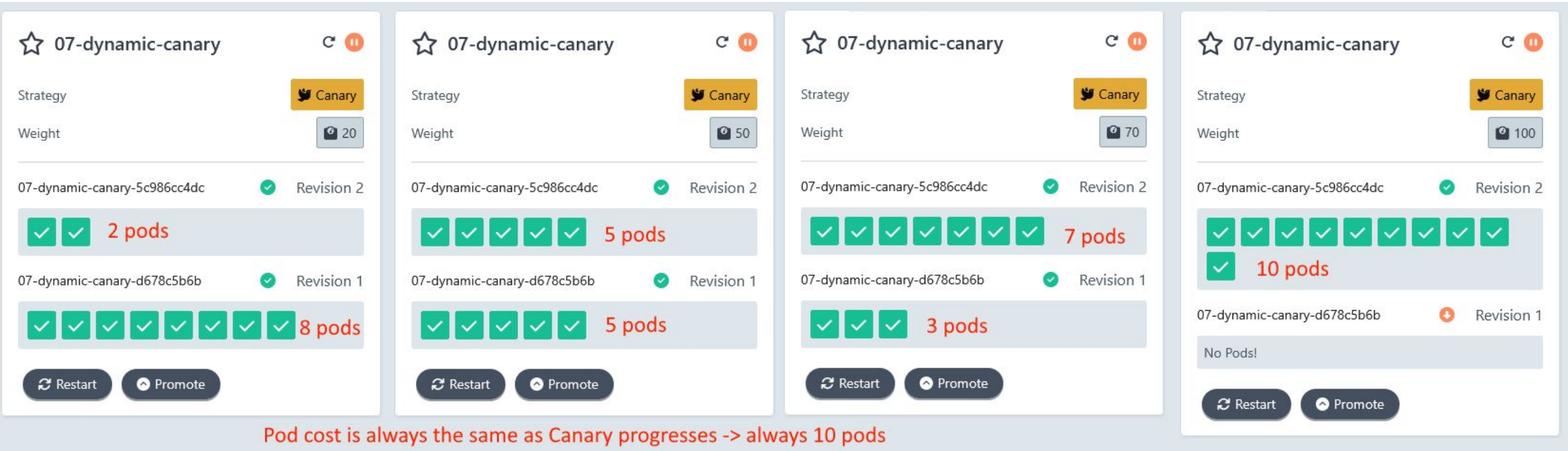
Proportional scaling

- Enabled **dynamicStableScale** property
- Stable pods will decrease as canary pods increase



```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: 07-dynamic-canary
spec:
  replicas: 10
  strategy:
    canary:
      canaryService: rollout-canary-preview
      stableService: rollout-canary-stable
      trafficRouting:
        traefik:
          weightedTraefikServiceName: traefik-service
      dynamicStableScale: true
      steps:
        - setWeight: 20
        - pause: {}
        - setWeight: 50
        - pause: {}
        - setWeight: 70
        - pause: {}
        - setWeight: 100
        - pause: {}
  revisionHistoryLimit: 2
```


Proportional scaling



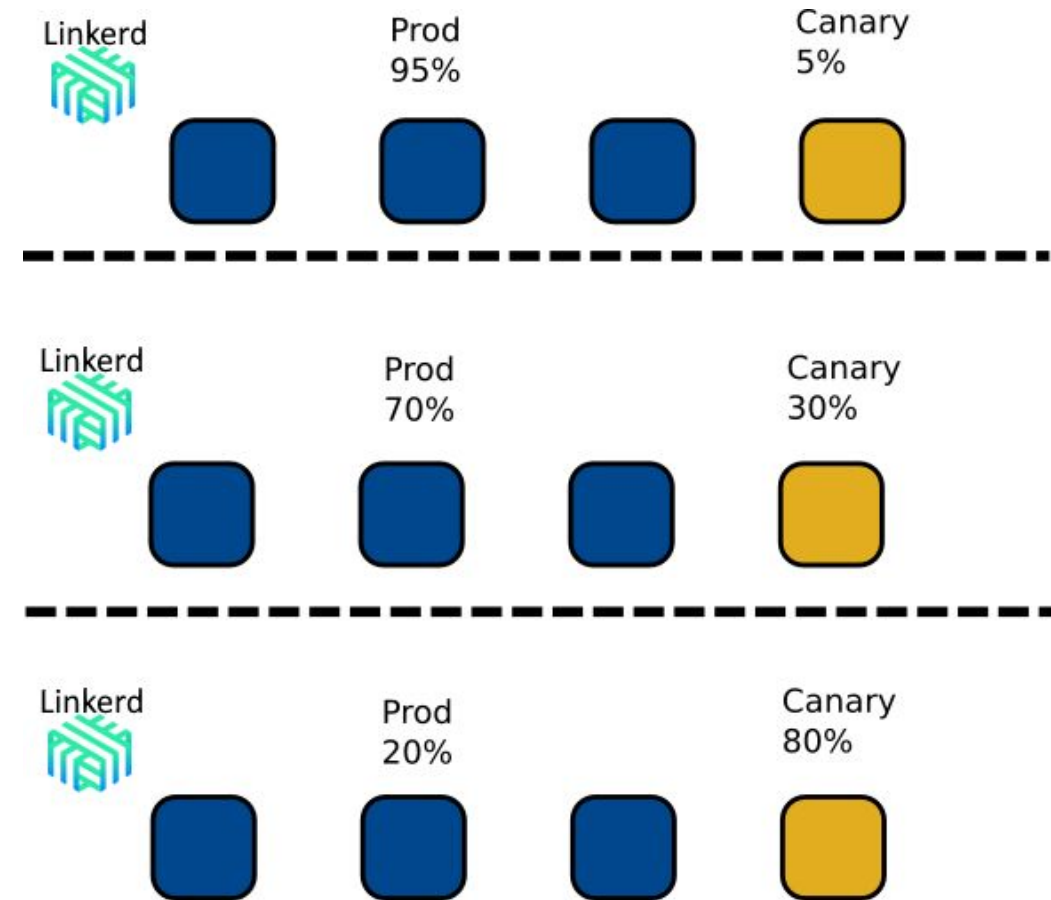
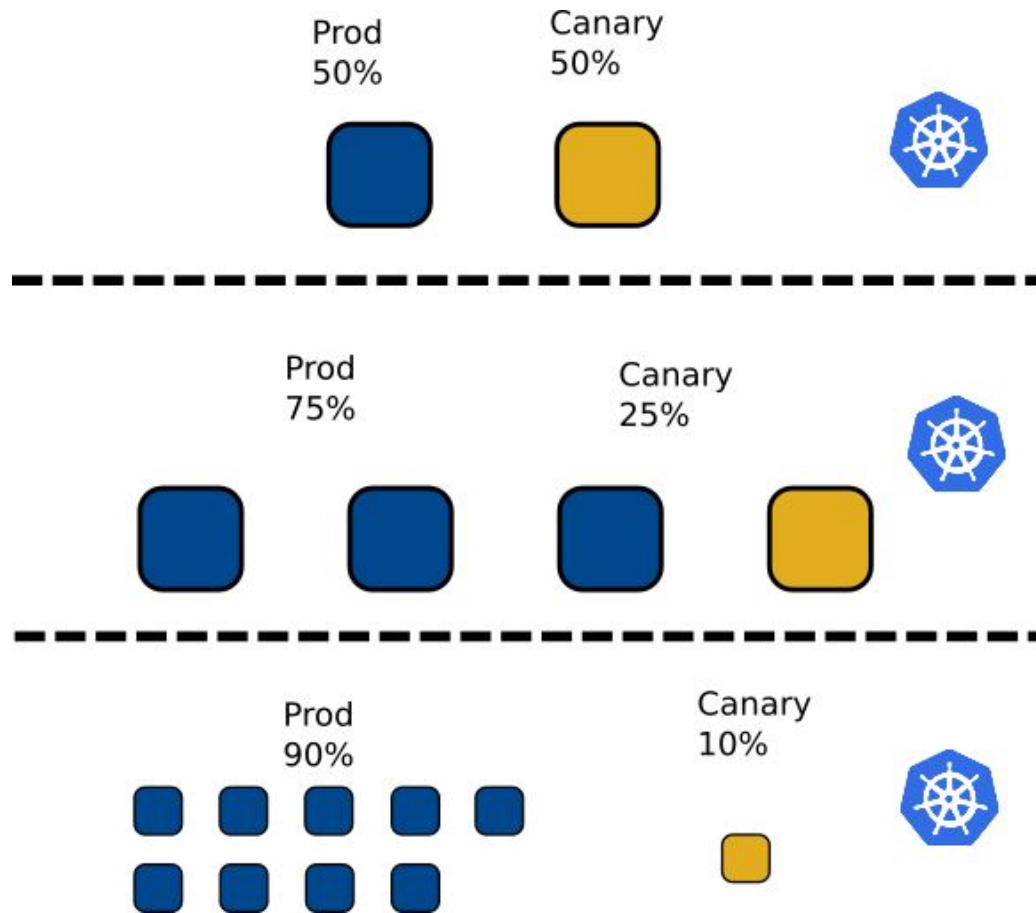
Decouple canary traffic from canary pods

Control canary traffic



- Default behavior -> number of canary pods get proportionate traffic
- You **CAN** change this and send **ANY** number of traffic to **ANY** number of canary pods
- Requires a traffic manager

Without/With traffic controller



Control canary traffic

- Send 20% of canary traffic to just 1 pod
- Send 50% of canary traffic to 3 pods
- Send 90% of canary traffic to 5 pods
- Send 100% of canary traffic to 8 pods



```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: 08-decoupled-canary
spec:
  replicas: 10
  strategy:
    canary:
      canaryService: rollout-canary-preview
      stableService: rollout-canary-stable
      trafficRouting:
        traefik:
          weightedTraefikServiceName: traefik-service
      steps:
        - setWeight: 20
        - setCanaryScale:
            replicas: 1
        - pause: {}
        - setWeight: 50
        - setCanaryScale:
            replicas: 3
        - pause: {}
        - setWeight: 90
        - setCanaryScale:
            replicas: 5
        - pause: {}
        - setWeight: 100
        - setCanaryScale:
            replicas: 8
        - pause: {}
  revisionHistoryLimit: 2
```

Control canary traffic

☆ 08-decoupled-canary

Strategy

Weight replicas: 1

Canary 20

08-decoupled-canary-75cf495c4b ✓ Revision 2

✓ 1 pod gets 20% of traffic and not 10%

08-decoupled-canary-857db784fb ✓ Revision 1

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

Restart Promote

☆ 08-decoupled-canary

Strategy

Weight replicas: 3

Canary 50

08-decoupled-canary-75cf495c4b ✓ Revision 2

✓ ✓ ✓ 3 pods get 50% of traffic and not 30%

08-decoupled-canary-857db784fb ✓ Revision 1

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

Restart Promote

☆ 08-decoupled-canary

Strategy

Weight replicas: 5

Canary 90

08-decoupled-canary-75cf495c4b ✓ Revision 2

✓ ✓ ✓ ✓ ✓ 5 pods get 90% and not 50%

08-decoupled-canary-857db784fb ✓ Revision 1

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

Restart Promote

Traffic of canary does not depend on the number of preview pods anymore



Employing HPA + Argo Rollouts

HPA and Argo Rollouts



Rollout CRD

```
subresources:  
  status: {}  
  scale:  
    labelSelectorPath: .status.selector  
    specReplicasPath: .spec.replicas  
    statusReplicasPath: .status.replicas
```



Horizontal
Autoscaler



HPA (memory based)

- Watch memory usage
- Scale 1-10 pods
- Point to a Rollout CRD

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: demo-hpa
  labels:
    app: cost-demo
spec:
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: memory
        target:
          type: AverageValue
          averageValue: 16Mi
  minReplicas: 1
  scaleTargetRef:
    apiVersion: argoproj.io/v1alpha1
    kind: Rollout
    name: 03-hpa-bg
```

Example app

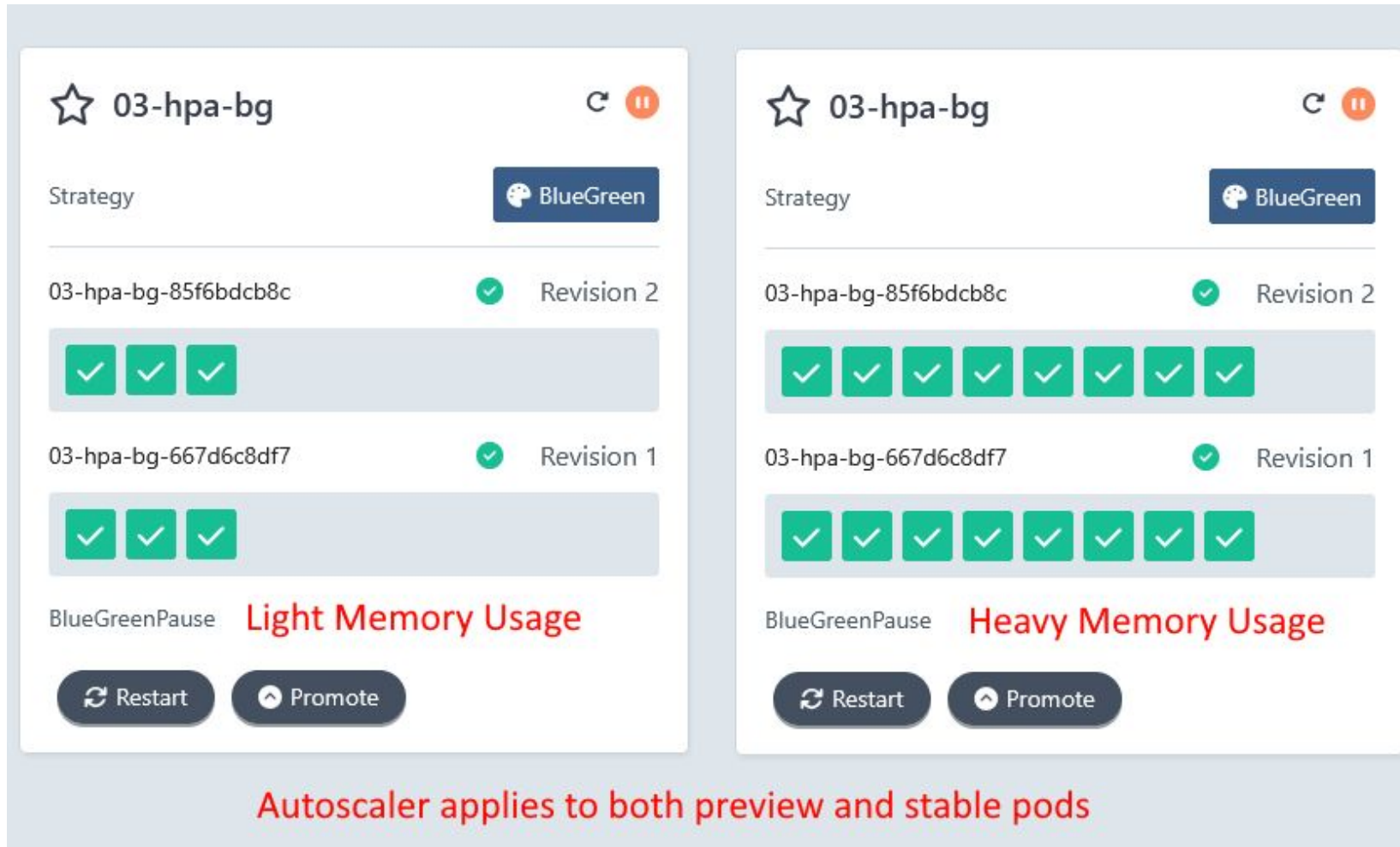
- Each HTTP call reserves 1MB memory
- /clear releases all reserved memory
- Available as container image



```
func handler(w http.ResponseWriter, r *http.Request) {  
    // Seed the random number generator with the current time  
    myRandom := rand.New(rand.NewSource(int64(new(maphash.Hash).Sum64())))  
  
    // Create a buffer of 1MB  
    buffer := make([]byte, 1024*1024)  
  
    // Fill the buffer with completely random data  
    _, err := myRandom.Read(buffer)  
    if err != nil {  
        http.Error(w, "Failed to generate random data", http.StatusInternalServerError)  
        return  
    }  
  
    // Run the SHA1 algorithm on the buffer  
    hash := sha1.New()  
    hash.Write(buffer)  
    sha1Result := hex.EncodeToString(hash.Sum(nil))  
}
```

<https://github.com/kostis-codefresh/rollouts-autoscaling-example/tree/main/source-code>

Blue/Green + HPA



☆ 03-hpa-bg

Strategy BlueGreen

03-hpa-bg-85f6bdc8c ✓ Revision 2

03-hpa-bg-667d6c8df7 ✓ Revision 1

BlueGreenPause **Light Memory Usage**

Restart Promote

☆ 03-hpa-bg

Strategy BlueGreen

03-hpa-bg-85f6bdc8c ✓ Revision 2

03-hpa-bg-667d6c8df7 ✓ Revision 1

BlueGreenPause **Heavy Memory Usage**

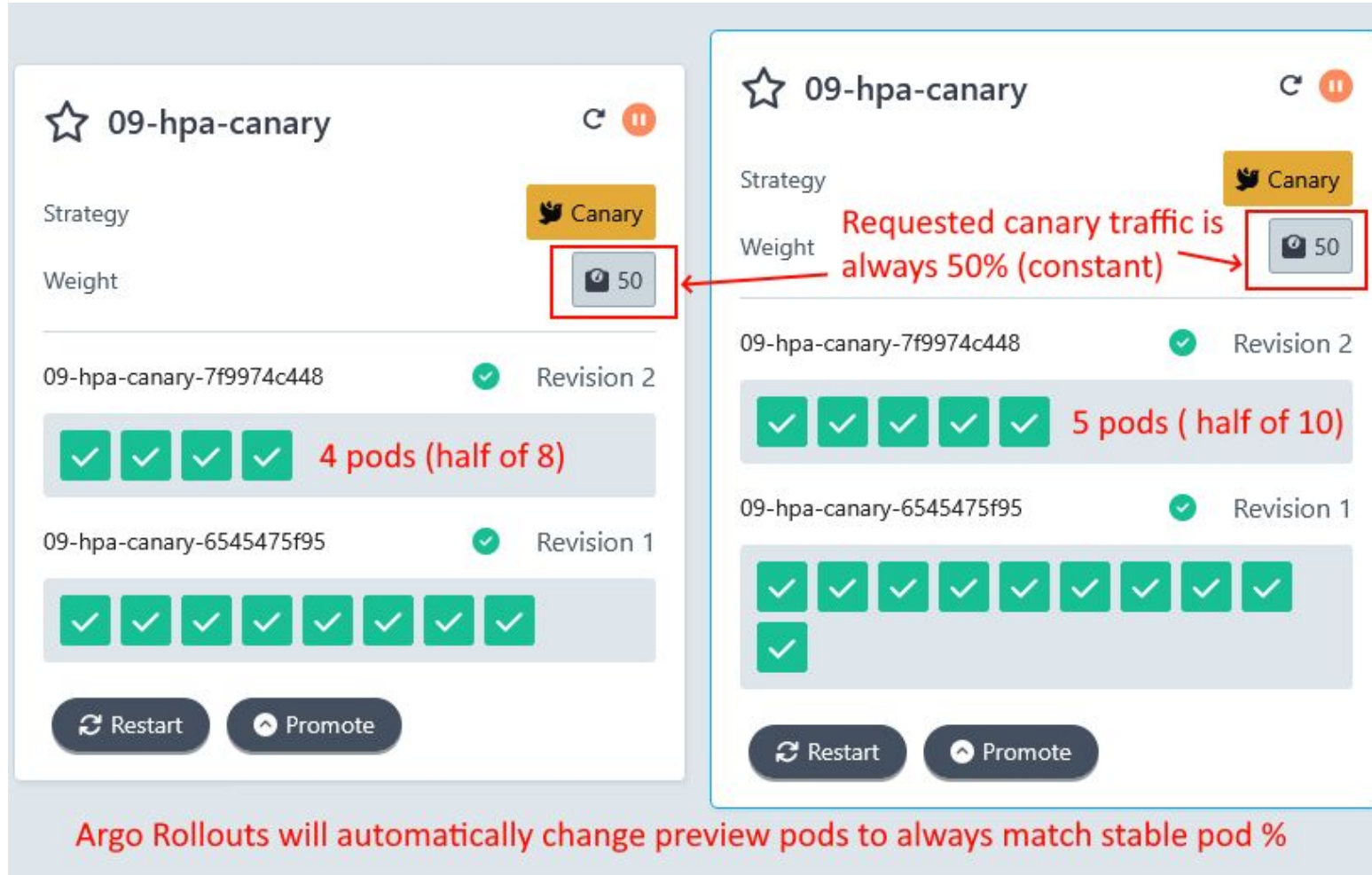
Restart Promote

Autoscaler applies to both preview and stable pods

<https://github.com/kostis-codefresh/rollouts-autoscaling-example/tree/main/03-hpa-bg>



Canary + HPA



09-hpa-canary

Strategy: Canary

Weight: 50

09-hpa-canary-7f9974c448 ✓ Revision 2

4 pods (half of 8)

09-hpa-canary-6545475f95 ✓ Revision 1

8 pods

Restart Promote

09-hpa-canary

Strategy: Canary

Weight: 50

Requested canary traffic is always 50% (constant)

09-hpa-canary-7f9974c448 ✓ Revision 2

5 pods (half of 10)

09-hpa-canary-6545475f95 ✓ Revision 1

9 pods

Restart Promote

Argo Rollouts will automatically change preview pods to always match stable pod %



HPA versus Argo Rollouts



Question:

- What happens if we use **previewReplicaCount** and HPA?
- What happens if we use **setCanaryScale** and HPA?

HPA versus Argo Rollouts

Question:

- What happens if we use **previewReplicaCount** and HPA?
- What happens if we use **setCanaryScale** and HPA?



Answer:

Argo Rollouts **overrides** HPA

Blue/Green + HPA + Count

☆ 04-hpa-bg-custom



Strategy

BlueGreen

04-hpa-bg-custom-85f6bdc8c ✓ Revision 2



Only 3 pods for preview
because of previewReplicaCount

04-hpa-bg-custom-667d6c8df7 ✓ Revision 1



BlueGreenPause

Light memory usage

Restart

Promote

☆ 04-hpa-bg-custom



Strategy

BlueGreen

04-hpa-bg-custom-85f6bdc8c ✓ Revision 2



Still 3 pods only for preview
regardless of actual load

04-hpa-bg-custom-667d6c8df7 ✓ Revision 1



BlueGreenPause

Heavy memory usage

Restart

Promote

```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: 04-hpa-bg-custom
spec:
  strategy:
    blueGreen:
      previewService: argo-rollouts-preview-service
      activeService: argo-rollouts-stable-service
      previewReplicaCount: 3
      autoPromotionEnabled: false
      revisionHistoryLimit: 2
  selector:
```

Setting previewReplicaCount = 3
overrides HPA. Autoscaling now only happens on stable pods



Canary + HPA + custom traffic



10-hpa-canary-decoupled

Strategy: Canary

Weight: 50

10-hpa-canary-decoupled-7f9974c448 ✓ Revision 2

3 pods get 50%

10-hpa-canary-decoupled-6545475f95 ✓ Revision 1

4 pods

Restart Promote

10-hpa-canary-decoupled

Strategy: Canary

Weight: 50

10-hpa-canary-decoupled-7f9974c448 ✓ Revision 2

3 pods get 50%

10-hpa-canary-decoupled-6545475f95 ✓ Revision 1

8 pods

Restart Promote

10-hpa-canary-decoupled

Strategy: Canary

Weight: 50

10-hpa-canary-decoupled-7f9974c448 ✓ Revision 2

3 pods get 50%

10-hpa-canary-decoupled-6545475f95 ✓ Revision 1

9 pods

Restart Promote

Autoscaler only affects stable pods. Preview pods are always 3 since this was defined explicitly in the Rollout Spec

Conclusion

Key points



- There are several properties you can fine tune
- Evaluate cost vs flexibility vs ease of rollback
- Using HPA is not required for simple scenarios
- When HPA and Argo Rollouts disagree,

Argo Rollouts wins and it will override HPA

Combinations

Example	Strategy	Options	Stable pods	Preview pods
01	Blue/Green	Default options	Rollouts Spec	Equal to stable pods
02	Blue/Green	Pin preview count	Rollouts Spec	Explicit number
03	Blue/Green	HPA	Controlled by HPA	Controlled by HPA
04	Blue/Green	Pin + HPA	Controlled by HPA	Explicit number
05	Canary	Default options	Rollouts Spec (Constant)	Rollouts Spec
06	Canary	Without Traffic manager	Inverse of Preview pods	Rollouts Spec
07	Canary	Dynamic scaling	Inverse of Preview pods	Rollouts Spec
08	Canary	Decoupled traffic	Rollouts Spec	Explicit number
09	Canary	HPA	Controlled by HPA	Controlled by HPA and current canary weight
10	Canary	HPA + Decoupled	Rollouts Spec	Explicit number

<https://github.com/kostis-codefresh/rollouts-autoscaling-example>





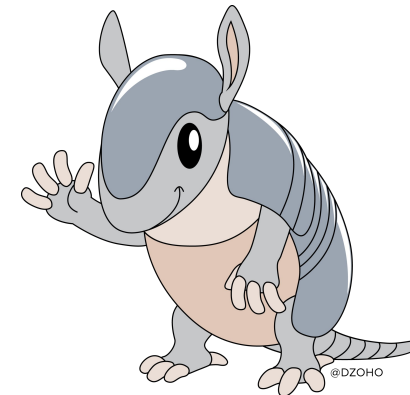
TAG Contributor Strategy

DEAF & HARD OF HEARING
WORKING GROUP

Wondering what it is like to be **deaf in tech**?

Want to know what our community can do
to **improve accessibility**?

Come chat with us!

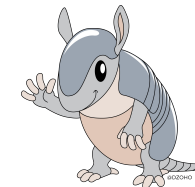


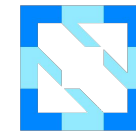


Become a **Mentor** for underrepresented groups in Open Source!

Passionate about fostering an inclusive open source community?

Sign Up today!





TAG Contributor Strategy

DEAF & HARD OF HEARING
WORKING GROUP

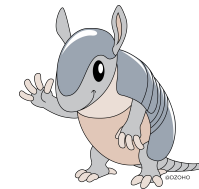
Sign Language Crash Course

Did you know? There are over **300** sign languages worldwide!

Join our course to learn a few signs in **ASL** (**BSL** available on demand).

Thu, 17:00 ~ 18:00 | DEI Community Hub

#deaf-and-hard-of-hearing (CNCf Slack)



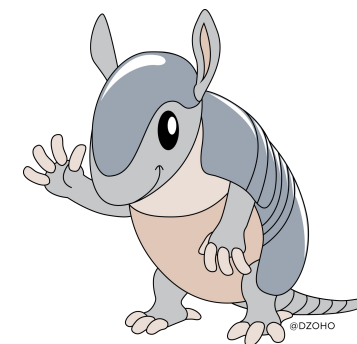
Thank you



Give us feedback! ->



- <https://argoproj.github.io/rollouts/>
- <https://argo-rollouts.readthedocs.io/en/stable/features/hpa-support/>
- <https://github.com/kostis-codefresh/rollouts-autoscaling-example>
- <https://contribute.cncf.io/about/deaf-and-hard-of-hearing/>



Argo Rollouts + HPA

Gubska Anastasiia (*BT Group*)
Kapelonis Kostis (*Octopus Deploy*)