intuit
simplify the business of life

Argo Rollouts

Progressive Delivery on Kubernetes

Danny Thomson

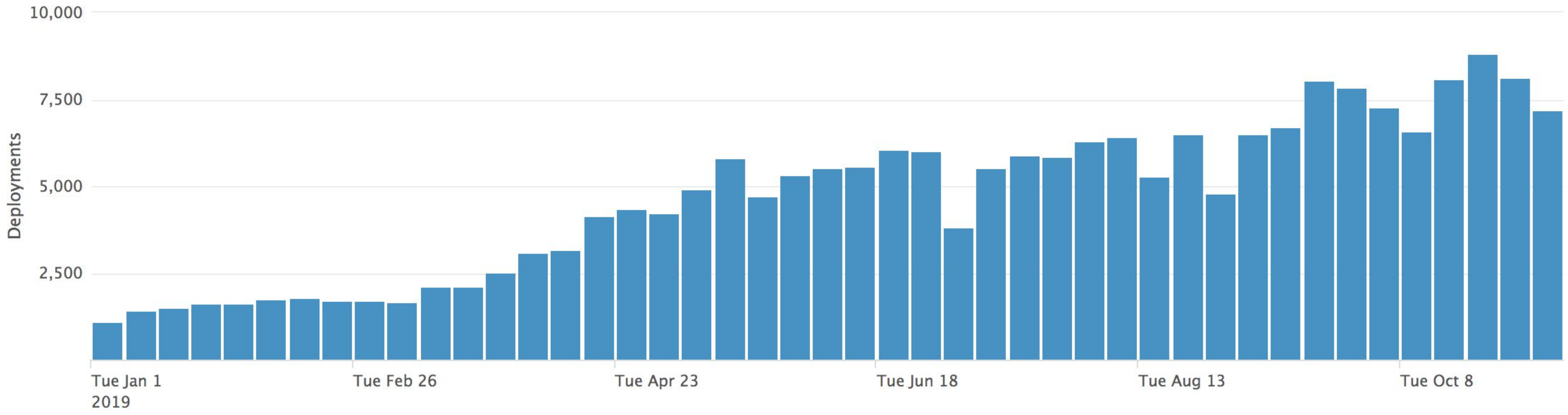
Some Intuit Statistics

intuit

- 4 business units
- 30 business segments
- 1,200+ developers using Kubernetes

kubernetes

- 160+ clusters (Intuit managed)
- 6,600 nodes
- 5,400 namespaces
- 62,000 pods
- **1,300 deploys a day**



Problem with native Deployment

- No advance strategies like BlueGreen/Canary
- Rolling Update provides few controls over speed
- Container readiness probes are not enough
    - Unsuitable for deeper or temporary checks
    - Unable to use external metrics
- Able to halt the progression, but not reverse

Use Cases

How do I…

- orchestrate advance strategies like BlueGreen/Canary
- automatically rollback an update due to failed metrics
- fine-tune my success and failure criteria
- insert a manual judgement step
- use my own business metrics for analysis
- experiment with multiple versions of my service
- (e.g. baseline vs. canary, A/B testing)
- and others...

Argo Rollouts

Phase 1: Deployment++

- Drop-in replacement for a Deployment
- Additional deployment strategies: blue-green and canary
- Declarative and GitOps friendly

Rollout

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: canary-demo
spec:
  replicas: 5
  template:
    spec:
      containers:
      - name: app
        image: argoproj/rollouts-demo:blue
    ...
  strategy:
    canary:
      steps:
      - setWeight: 40
      - pause: {duration: 3600}
      - setWeight: 60
      - pause: {duration: 10}
      - setWeight: 80
      - pause: {duration: 10}
```

- Manages creation, scaling, and deletions of ReplicaSets

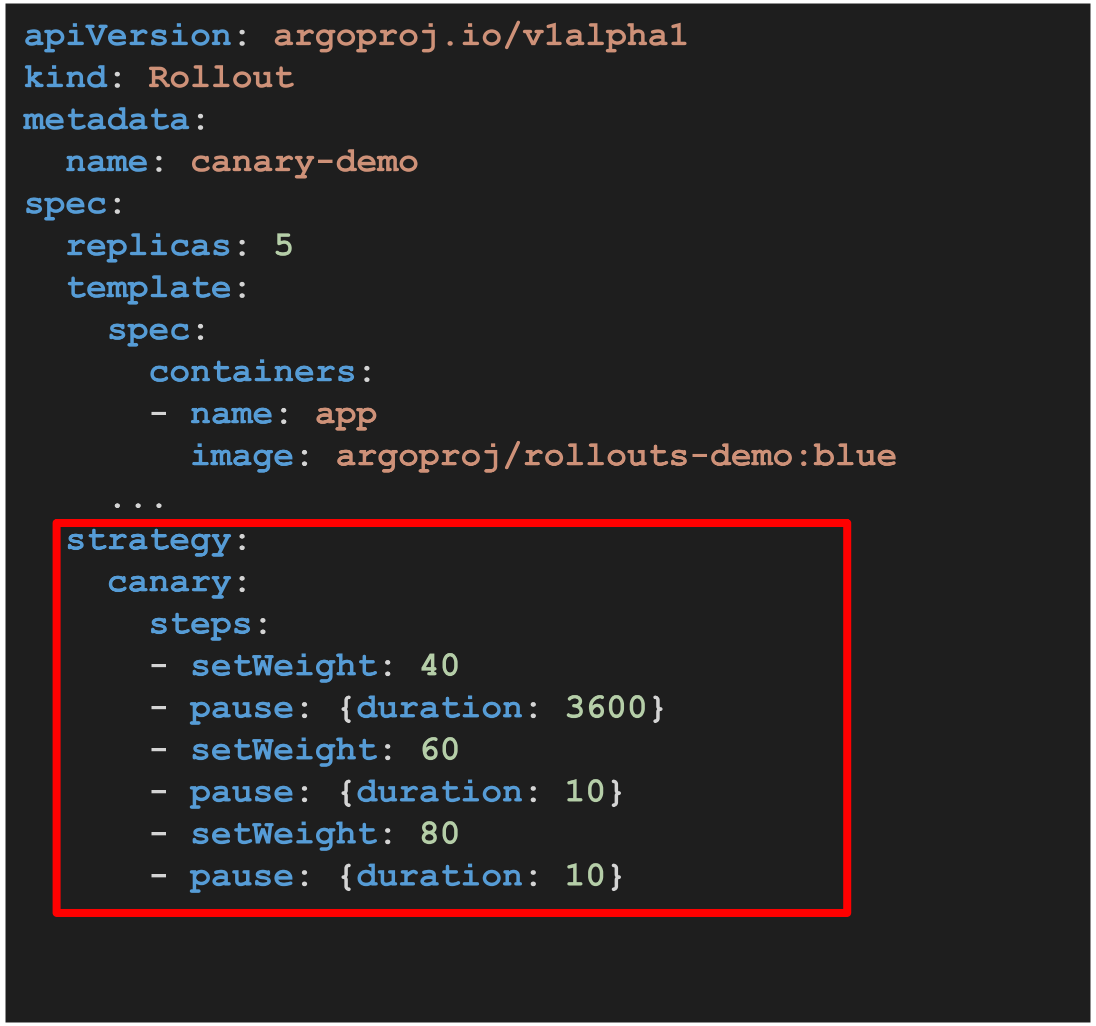
Rollout

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: canary-demo
spec:
  replicas: 5
  template:
    spec:
      containers:
      - name: app
        image: argoproj/rollouts-demo:blue
    ...
  strategy:
    canary:
      steps:
      - setWeight: 40
      - pause: {duration: 3600}
      - setWeight: 60
      - pause: {duration: 10}
      - setWeight: 80
      - pause: {duration: 10}
```

- Spec is mostly identical to Deployment

Demo

Rollout



```yaml
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: canary-demo
spec:
  replicas: 5
  template:
    spec:
      containers:
      - name: app
        image: argoproj/rollouts-demo:blue
    ...
  strategy:
    canary:
      steps:
      - setWeight: 40
      - pause: {duration: 3600}
      - setWeight: 60
      - pause: {duration: 10}
      - setWeight: 80
      - pause: {duration: 10}
```

- New blue-green and canary strategies provides control over how to update the stable version to new version

Rollout

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: canary-demo
spec:
  replicas: 5
  template:
    spec:
      containers:
      - name: app
        image: argoproj/rollouts-demo:blue
    ...
  strategy:
    canary:
      steps:
      - setWeight: 40
      - pause: {duration: 3600}
      - setWeight: 60
      - pause: {duration: 10}
      - setWeight: 80
      - pause: {duration: 10}
```

- New blue-green and canary strategies provides control over how to update the stable version to new version

Argo Rollouts

Phase 1: Deployment++

- Drop-in replacement for a Deployment
- Additional deployment strategies: blue-green and canary
- Declarative and GitOps friendly

Phase 2: Progressive Delivery

- Analysis
- Experimentation

Analysis CRD

- Brings observability to the delivery process
- Defines how to perform a canary analysis:
  - What metrics to measure and when
  - What values are considered successful, failed, inconclusive
- Automates promotion & rollback

Rollout Integration

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: canary-demo
spec:
  replicas: 5
  template:
    spec:
      containers:
      - name: app
        image: argoproj/rollouts-demo:blue
    ...
  strategy:
    canary:
      analysis:
        templateName: success-rate
      steps:
      - setWeight: 40
      - pause: {duration: 3600}
      - setWeight: 60
      - pause: {duration: 10}
      - setWeight: 80
      - pause: {duration: 10}
```

Canary Analysis

- Analysis is performed in the background, while the rollout is progressing through its steps
- Started at the beginning of a rollout, and stopped when the rollout is complete

Analysis Template

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: success-rate
spec:
  args:
  - name: ingress
  metrics:
  - name: success-rate
    interval: 5m
    count: 5
    successCondition: result[0] > 0.90
    failureLimit: 2
    provider:
      prometheus:
        address: http://prometheus-svc.prometheus-ns:9090
        query: >-
          sum(rate(nginx_ingress_controller_requests
              {ingress="{{args.ingress}}"status!~"[4-5].*"}[5m]))
          /
          sum(rate(nginx_ingress_controller_requests
              {ingress="{{args.ingress}}"}[5m]))
```

Defines one or more key metrics to monitor during a rollout

Support for many providers:

- Prometheus
- Job
- Kayenta
- Web
- Wavefront

Analysis Template - Prometheus Provider

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: success-rate
spec:
  args:
  - name: ingress
  metrics:
  - name: success-rate
    interval: 5m
    count: 5
    successCondition: result[0] > 0.90
    failureLimit: 2
    provider:
      prometheus:
        address: http://prometheus-svc.prometheus-ns:9090
        query: >-
          sum(rate(nginx_ingress_controller_requests
              {ingress="{{args.ingress}}"status!~"[4-5].*"}[5m]))
          /
          sum(rate(nginx_ingress_controller_requests
              {ingress="{{args.ingress}}"}[5m]))
```

Prometheus Provider

Address - prometheus server

Query - PromQL query

Example (HTTP success rate):

$$\frac{\text{\# of non-4xx/5xx HTTP requests}}{\text{\# of total HTTP requests}}$$

Analysis Template - Success Condition

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: success-rate
spec:
  args:
  - name: ingress
  metrics:
  - name: success-rate
    interval: 5m
    count: 5
    successCondition: result[0] > 0.90
    failureLimit: 2
    provider:
      prometheus:
        address: http://prometheus-svc.prometheus-ns:9090
        query: >-
          sum(rate(nginx_ingress_controller_requests
              {ingress="{{args.ingress}}"status!~"[4-5].*"}[5m]))
          /
          sum(rate(nginx_ingress_controller_requests
              {ingress="{{args.ingress}}"}[5m]))
```

- An **expression** which interprets the result of a measurement
- Results can return as:
  - Scalars
  - Vectors
  - structured objects
- Built-in functions like any(), all(), filter(), map()
- Results can also be Inconclusive to allow for manual judgements

Analysis Template - Interval & Count

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: success-rate
spec:
  args:
  - name: ingress
  metrics:
  - name: success-rate
    interval: 5m
    count: 5
    successCondition: result[0] > 0.90
    failureLimit: 2
    provider:
      prometheus:
        address: http://prometheus-svc.prometheus-ns:9090
        query: >-
          sum(rate(nginx_ingress_controller_requests
              {ingress="{{args.ingress}}"status!~"[4-5].*"}[5m]))
          /
          sum(rate(nginx_ingress_controller_requests
              {ingress="{{args.ingress}}"}[5m]))
```

- **Interval**
  - How frequent to query the provider
- **Count**
  - Number of times to take a measurement
  - Runs indefinitely if omitted (or until failure)

Analysis Template - Arguments

```yaml
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: success-rate
spec:
  args:
  - name: ingress
  metrics:
  - name: success-rate
    interval: 5m
    count: 5
    successCondition: result[0] > 0.90
    failureLimit: 2
    provider:
      prometheus:
        address: http://prometheus-svc.prometheus-ns:9090
        query: >-
          sum(rate(nginx_ingress_controller_requests
              {ingress='{{args.ingress}}'status!~"[4-5].*"}[5m]))
          /
          sum(rate(nginx_ingress_controller_requests
              {ingress='{{args.ingress}}'}[5m]))
```

- Arguments make Analysis Templates parameterizable
- Enables templates to be reusable/standardized across organizations and communities
- Makes templates building blocks for higher levels resources

Q/A

Links

- Argo Rollouts Repo
- Argo Rollouts Docs
- Join #argo-rollouts in ArgoProj Slack: