



THE LINUX FOUNDATION



NORTH AMERICA

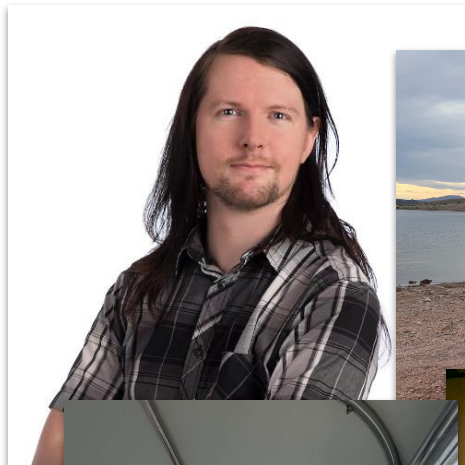
A Fun Overview of the Argo Ecosystem

by: Aaron Teague



About Me

- Works as SRE
- Always remote
- Enjoys making lessons fun
- Has really cool dog



ArgoCD



THE LINUX FOUNDATION

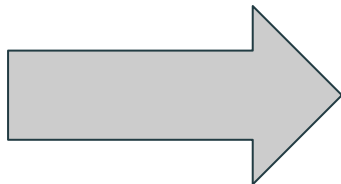
NORTH AMERICA



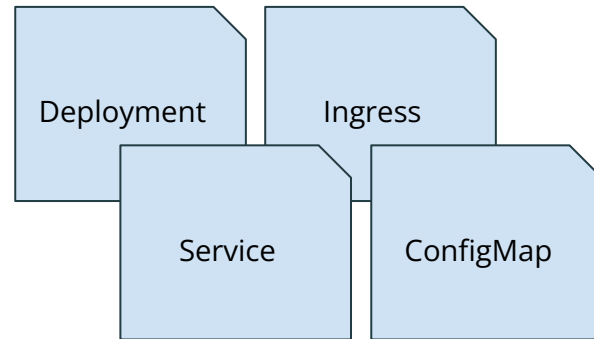
From Git to K8s Environment



- GitHub
- BitBucket
- GitLab



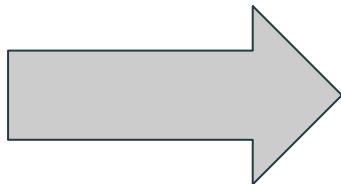
kubernetes



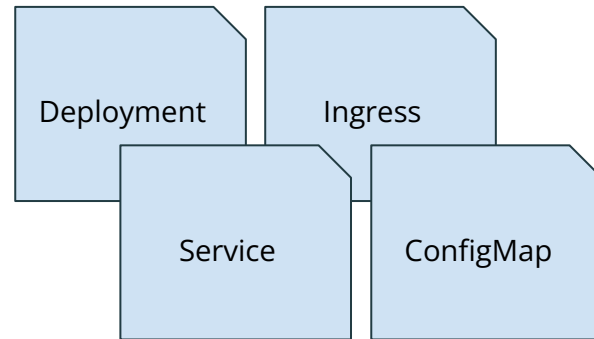
From Git to K8s Environment



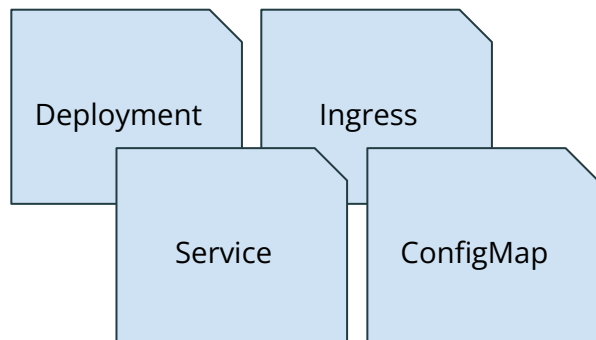
- GitHub
- BitBucket
- GitLab



kubernetes



Several Management Options



The Architecture

```
aaron@Aarons-MacBook-Pro argo-presentation % kubectl get pods -n argocd
```

NAME	READY	STATUS	RESTARTS	AGE
argo-cd-argocd-application-controller-0	1/1	Running	0	6d1h
argo-cd-argocd-applicationset-controller-6fdff4c95f-rdwbl	1/1	Running	0	6d1h
argo-cd-argocd-dex-server-6fb8d4746f-pswbj	1/1	Running	0	6d1h
argo-cd-argocd-notifications-controller-6fcff4b4bc-ftlw	1/1	Running	0	6d1h
argo-cd-argocd-redis-56d87c4dd8-hnrmm	1/1	Running	0	6d1h
argo-cd-argocd-repo-server-5ff859989b-v2sh8	1/1	Running	0	6d1h
argo-cd-argocd-server-78c46d4474-dlbl9	1/1	Running	0	6d1h



Songs Include:

- Deploying in the Moonlight
- Since U Been Synced
- I git-ta Feeling
- Hey There, Deployment
- *And More!*

Call Today!

Hello World! (featuring Nginx)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hello-world
spec:
  selector:
    matchLabels:
      app: nginx-hello-world
  template:
    metadata:
      labels:
        app: nginx-hello-world
    spec:
      containers:
        - image: nginx:latest
          name: nginx
          ports:
            - containerPort: 80
      volumeMounts:
        - name: config-vol
          mountPath: /etc/nginx/
      volumes:
        - name: config-vol
          configMap:
            name: nginx-hello-world
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-hello-world
data:
  nginx.conf: |
    events {
    }
    http {
      server {
        listen 80;
        location / {
          return 200 "hello world";
        }
      }
    }
  }
```

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-hello-world
spec:
  ports:
    - port: 80
      targetPort: 80
  selector:
    app: nginx-hello-world
```

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-hello-world
spec:
  ingressClassName: nginx
  rules:
    - host: nginx-hello-world.example.com
      http:
        paths:
          - backend:
              service:
                name: nginx-hello-world
                port:
                  number: 80
            path: /
            pathType: Prefix
```

```
kubectl apply -f all-that-jazz/
```

An Application

```
apiVersion: argoproj.io/v1alpha1
kind: Application
```

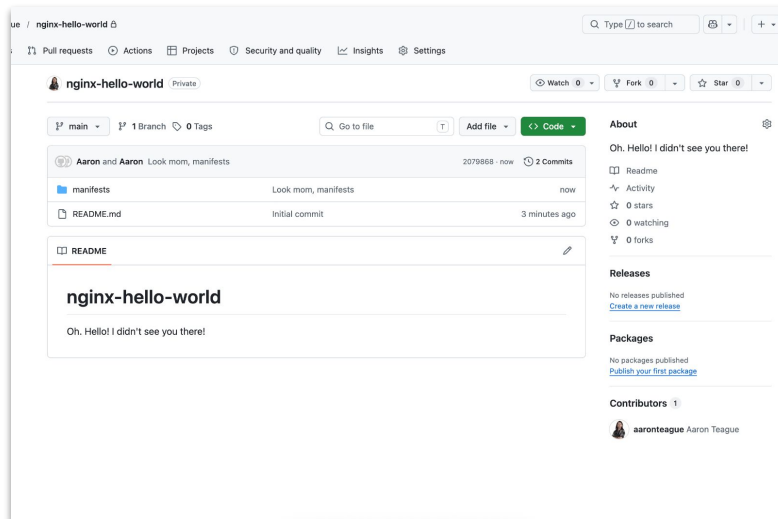
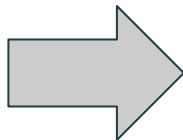
```
metadata:
  name: nginx-hello-world
  namespace: argocd
```

```
spec:
  syncPolicy:
    automated:
      enabled: true
```

```
project: default
```

```
sources:
  - repoURL: git@github.com:aaronteague/nginx-hello-world.git
    path: manifests
    targetRevision: HEAD
```

```
destination:
  server: "https://kubernetes.default.svc"
  namespace: default
```



Forming the git connection

The Result

The screenshot displays the Argo CD web interface for the application 'nginx-hello-world'. The interface is organized into several sections:

- Left Sidebar:** Contains the Argo logo (v3.3.7), navigation links for Applications, Settings, User Info, and Documentation. Below these are 'Resource filters' and a list of application resources.
- Top Navigation:** Includes buttons for DETAILS, DIFF, SYNC, SYNC STATUS, HISTORY AND ROLLBACK, DELETE, and REFRESH. A 'Log out' link is also present.
- Application Health:** Shows 'APP HEALTH' as 'Healthy' (green heart icon).
- Sync Status:** Displays 'SYNC STATUS' as 'Synced to HEAD (2079868) and (0) more' (green checkmark icon). It also shows 'LAST SYNC' as 'Sync OK to 2079868 and (0) more' (green checkmark icon), with a timestamp of 'Succeeded a few seconds ago (Sat May 02 2026 17:52:18 GMT+0500)'. The author is 'Aaron <aaron@Aarons-MacBook-Pro.local>' and the comment is 'Look mom, manifests'.
- Resource Tree:** A diagram showing the application's structure. The root node is 'nginx-hello-world' (Application). It branches into several components, each with its own status and refresh options:
 - 'cm' (ConfigMap)
 - 'svc' (Service)
 - 'deploy' (Deployment)
 - 'ing' (Ingress)
 - 'fs' (Filesystem)
 - 'pod' (Pod)

Thanks for Coming!



THE LINUX FOUNDATION

NORTH AMERICA



Embedded Linux
Conference





Get More Done with **Kubernetes**

Open source tools for Kubernetes to run workflows,
manage clusters, and do GitOps right.

[View on GitHub](#)



The big four

CD

Rollouts

Workflows

Events



Argo Rollouts

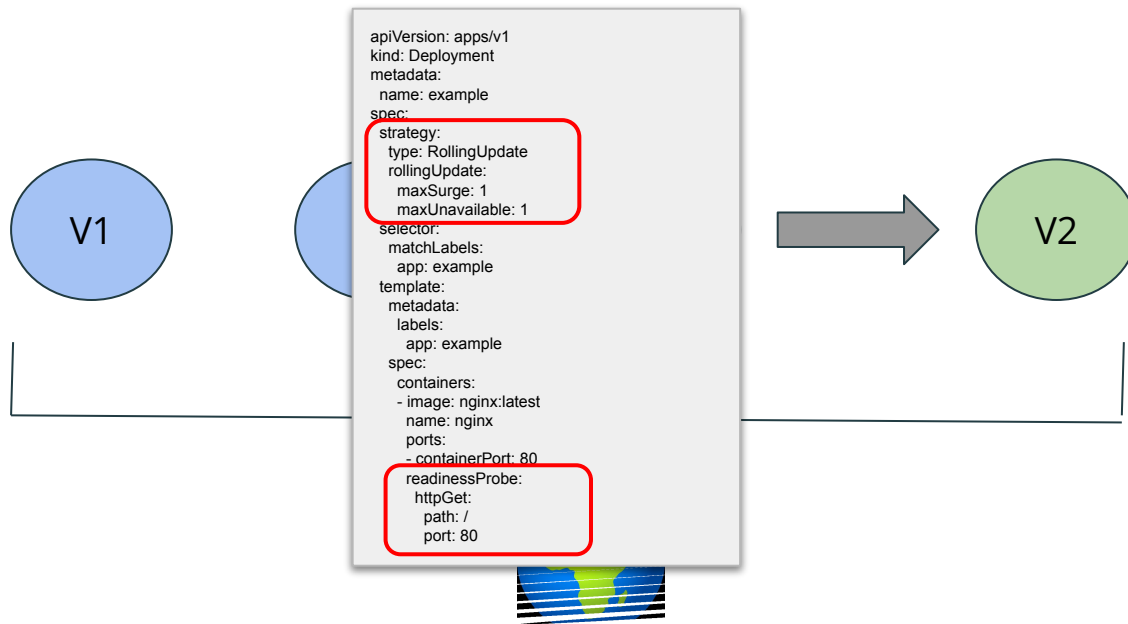
Advanced Kubernetes deployment strategies such as Canary and Blue-Green made easy.

[Documentation](#)



  3461

With finesse an grace



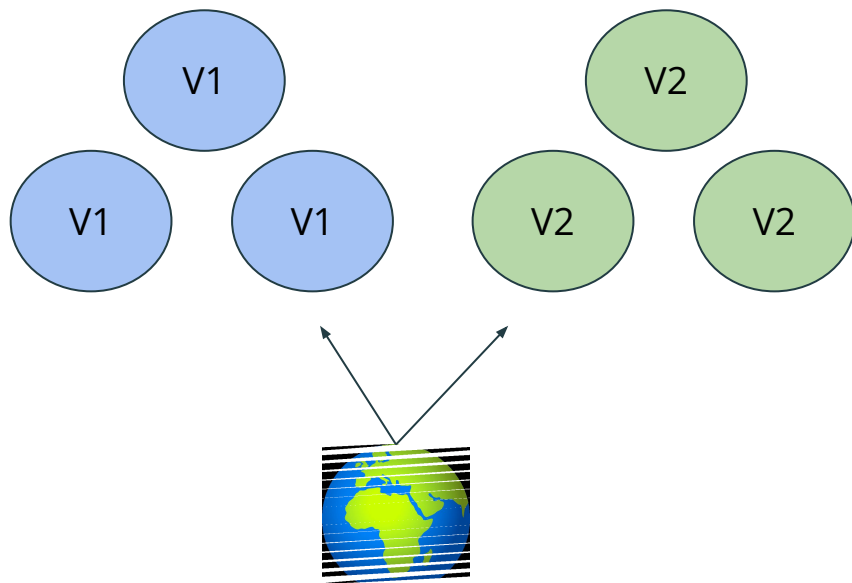
More to be desired

- New error codes
- High response times
- Heavy CPU load
- ~~Tax Fraud~~

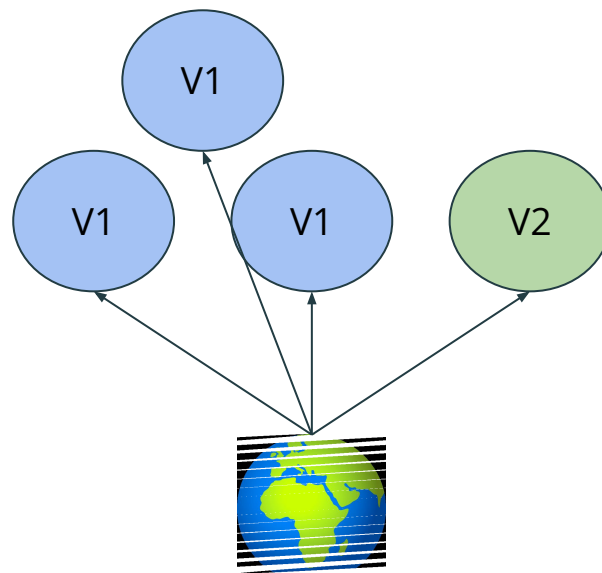


All at once? Or a little at a time?

Blue Green



Canary



A Rollout object

```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: nginx-hello-world
spec:
  replicas: 3
  strategy:
    canary:
      steps:
        - setWeight: 20
        - pause: {}
  selector:
    matchLabels:
      app: nginx-hello-world
  template:
    metadata:
      labels:
        app: nginx-hello-world
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
          volumeMounts:
            - name: config-vol
              mountPath: /etc/nginx/
          volumes:
            - name: config-vol
              configMap:
                name: nginx-hello-world
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hello-world
spec:
  replicas: 3

  selector:
    matchLabels:
      app: nginx-hello-world
  template:
    metadata:
      labels:
        app: nginx-hello-world
    spec:
      containers:
        - image: nginx:latest
          name: nginx
          ports:
            - containerPort: 80
          volumeMounts:
            - name: config-vol
              mountPath: /etc/nginx/
          volumes:
            - name: config-vol
              configMap:
                name: nginx-hello-world
```



Rollout

Pod

Pod

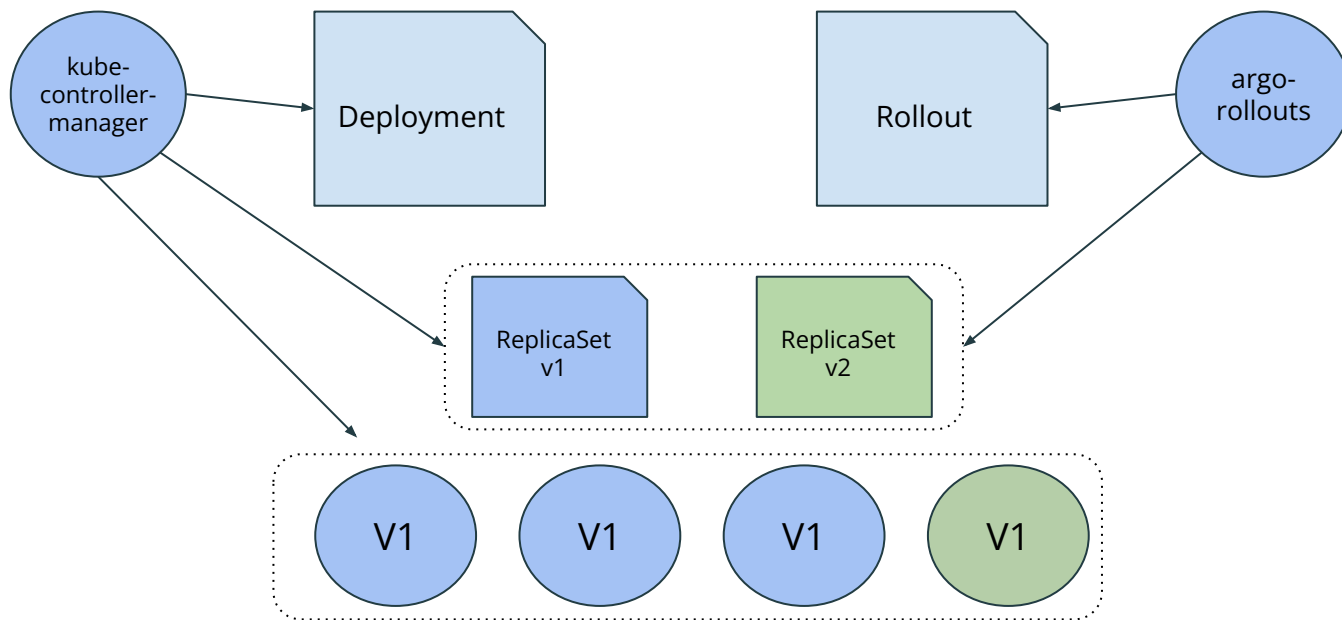
Pod

Pod

(Image credit: Michael Sohn/AP)




New Controller, same great pods



Hello World, but with Rollouts

```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: nginx-hello-world
spec:
  replicas: 3
  strategy:
    canary:
      steps:
        - setWeight: 20
        - pause: {}
  selector:
    matchLabels:
      app: nginx-hello-world
  template:
    metadata:
      labels:
        app: nginx-hello-world
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
          volumeMounts:
            - name: config-vol
              mountPath: /etc/nginx/
      volumes:
        - name: config-vol
          configMap:
            name: nginx-hello-world
```



- Applications
- Settings
- User Info
- Documentation

Applications /

DETAILS
DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH

Log out

APP HEALTH Healthy

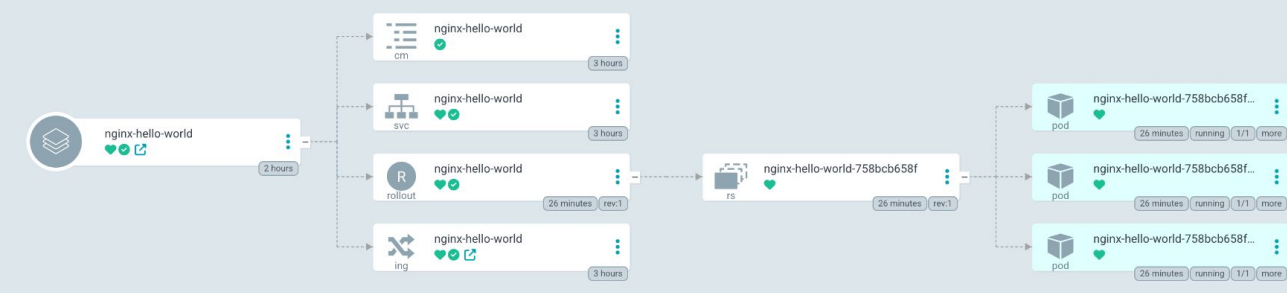
SYNC STATUS Synced to HEAD (17ccd10) and (0) more

Auto sync is enabled.
Author: Aaron <aaron@Aarons-MacBook-Pro.local> -
Comment: Migrating from deployment to rollout

LAST SYNC Sync OK to 17ccd10 and (0) more

Succeeded a few seconds ago (Sat May 02 2026 19:49:43 GMT-0500)
Author: Aaron <aaron@Aarons-MacBook-Pro.local> -
Comment: Migrating from deployment to rollout

100%



Resource filters

NAME

KINDS

SYNC STATUS

HEALTH STATUS

Let's put the canary into the coal mine

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-hello-world
data:
  nginx.conf: |
    events {
    }
    http {
      server {
        listen 80;
        location / {
          return 500 "go away!";
        }
      }
    }
  }
```

```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: nginx-hello-world
spec:
  [...]
```

template:

```
  metadata:
    labels:
      app: nginx-hello-world
  spec:
    containers:
      - name: nginx-v2
        image: nginx:latest
    ports:
      [...]
```

- DETAILS
- DIFF
- SYNC
- SYNC STATUS
- HISTORY AND ROLLBACK
- DELETE
- REFRESH

APP HEALTH **Suspended**

SYNC STATUS **Synced** to HEAD (2e64aba) and (0) more

LAST SYNC **Sync OK** to 2e64aba and (0) more

Auto sync is enabled.
 Author: Aaron <aaron@Aarons-MacBook-Pro.local> -
 Comment: Lets roll out a bad message

Succeeded a minute ago (Sat May 02 2026 19:58:07 GMT-0500)
 Author: Aaron <aaron@Aarons-MacBook-Pro.local> -
 Comment: Lets roll out a bad message

Navigation and zoom controls: list, diff, refresh, zoom in, zoom out, 100%

Resource filters

NAME

NAME

KINDS

KINDS

SYNC STATUS

Synced 4

OutOfSync 0

HEALTH STATUS

Progressing 0

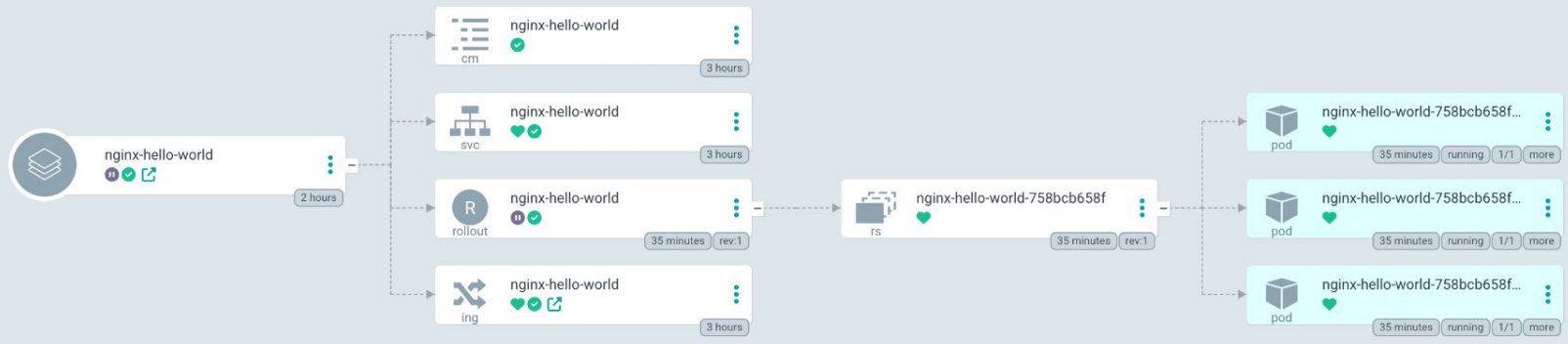
Suspended 1

Healthy 6

Degraded 0

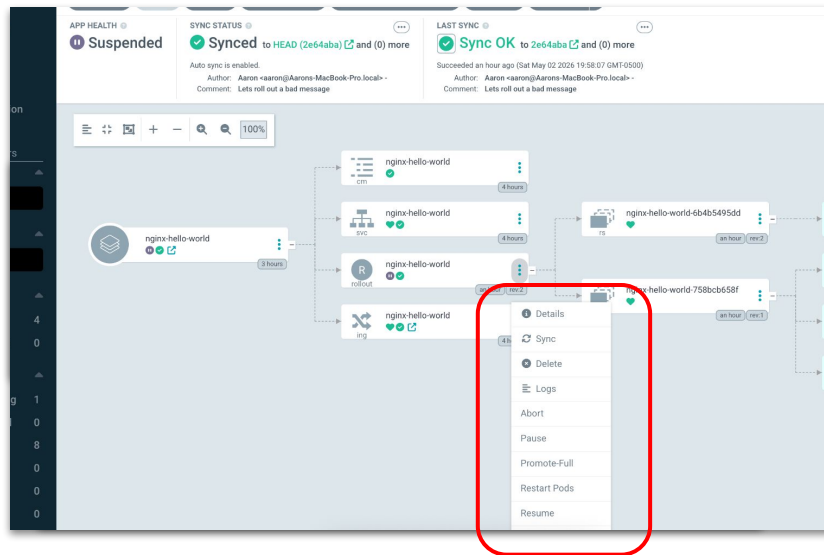
Missing 0

Unknown 0



Let's get this rolling again!

- `kubectl argo rollouts abort nginx-hello-world`
- `kubectl argo rollouts promote nginx-hello-world`
- As a CI/CD manual step
- From ArgoCD UI



If rollout good, why manual?



OPEN SOURCE SUMMIT

THE LINUX FOUNDATION

NORTH AMERICA



Embedded Linux
Conference



Time for an analysis

```
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: total-failures
```

```
spec:
  args:
  - name: service-name
  metrics:
```

```
- name: total-failures
  interval: 2m
  failureCondition: result[0] >= 0.2
  failureLimit: 1
```

```
provider:
  prometheus:
    address: http://prometheus-kube-prometheus-prometheus.prometheus:9090
```

```
query: |
sum(irate(
  nginx_ingress_controller_requests{exported_service="{{args.service-name}}", status!="5.."}[1m]
))
```

```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
```

```
metadata:
  name: nginx-hello-world
spec:
```

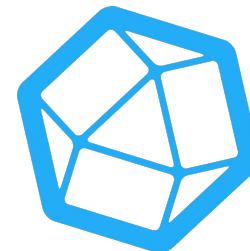
```
replicas: 3
strategy:
```

```
canary:
  steps:
  - setWeight: 20
  - pause: {duration: 10m}
```

```
analysis:
  templates:
  - templateName: total-failures
    startingStep: 0
```

```
args:
  - name: service-name
    value: nginx-hello-world
```

```
selector:
  matchLabels:
    app: nginx-hello-world
template:
[...]
```



Analysis Run

argo v3.3.7

Applications

Settings

User Info

Documentation

Resource filters

NAME

NAME

KINDS

KINDS

SYNC STATUS

Synced 5

OutOfSync 0

HEALTH STATUS

Progressing 1

Suspended 0

Healthy 9

Degraded 2

Missing 0

Unknown 0

Applications / nginx-hello-world

DETAILS DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

APP HEALTH **Degraded**

SYNC STATUS **Synced to HEAD (eaac54c) and (0) more**

LAST SYNC **Sync OK to eaac54c and (0) more**

Author: Aaron <aaron@Aarons-MacBook-Pro.local> - Comment: roll that beautiful rollout footage

Author: Aaron <aaron@Aarons-MacBook-Pro.local> - Comment: roll that beautiful rollout footage

nginx-hello-world (15 hours)

- nginx-hello-world (cm) (17 hours)
- nginx-hello-world (svc) (16 hours)
- total-failures (analysisistemplate) (22 minutes)
- nginx-hello-world (rollout) (14 hours rev:3)
- nginx-hello-world (ing) (16 hours)

- nginx-hello-world-8658b9b975 (rs) (19 minutes rev:3)
- nginx-hello-world-6b4b5495dd (rs) (13 hours rev:2)
- nginx-hello-world-758bc658f (rs) (14 hours rev:1)
- nginx-hello-world-8658b9b97... (analysisisrun) (19 minutes rev:3)
- nginx-hello-world-8658b9b97... (analysisisrun) (12 minutes rev:3)

- nginx-hello-world-6b4b5495d... (pod) (12 hours running 1/1 more)
- nginx-hello-world-6b4b5495d... (pod) (12 hours running 1/1 more)
- nginx-hello-world-6b4b5495d... (pod) (12 hours running 1/1 more)

APPLICATION DETAILS TREE

Log out

A more exact measurement of weight



OPEN SOURCE SUMMIT

THE LINUX FOUNDATION

NORTH AMERICA



Embedded Linux
Conference



Ingress Controller Manipulation

- AWS ALB Ingress Controller
- Ambassador Edge Stack
- Apache APISIX
- Google Cloud
- Gateway API
- Istio
- Kong Ingress
- Nginx Ingress Controller
- Service Mesh Interface (SMI)
- Traefik Proxy

A small percent to fail

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-hello-world
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx-hello-world
```

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-hello-world-canary
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx-hello-world
```

```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: nginx-hello-world
spec:
  replicas: 3
  strategy:
    canary:
      stableService: nginx-hello-world
      canaryService: nginx-hello-world-canary
      trafficRouting:
        nginx:
          stableIngress: nginx-hello-world
      steps:
        - setWeight: 2
        - pause: {duration: 10m}
      analysis:
        templates:
        - templateName: total-failures
      startingStep: 0
      args:
        - name: service-name
          value: nginx-hello-world
  selector:
    [...]
```

The Result

```
aaron@Aarons-MacBook-Pro nginx-hello-world % kubectl get ingress -l argo-rollouts-nginx-hello-world
```

```
nginx-  
nginx-
```

```
apiVersion: networking.k8s.io/v1  
kind: Ingress  
metadata:  
  annotations:  
    nginx.ingress.kubernetes.io/canary: "true"  
    nginx.ingress.kubernetes.io/canary-weight: "2"  
  creationTimestamp: "2026-05-03T14:46:21Z"  
  generation: 1  
  name: nginx-hello-world-nginx-hello-world-canary  
  namespace: default  
  resourceVersion: "4086955"  
  uid: 26525cc2-ab86-4a39-8a40-65caed93cb8d  
spec:  
  ingressClassName: nginx  
  rules:  
  - host: nginx-hello-world.example.com  
    http:  
      paths:  
      - backend:  
          service:  
            name: nginx-hello-world-canary  
            port:  
              number: 80  
          path: /  
          pathType: Prefix  
status:  
  loadBalancer:  
  ingress:  
  - ip: 134.199.181.43
```

```
apiVersion: v1  
kind: Service  
metadata:  
  annotations:  
    argo-rollouts.argo.dev/managed-by-rollouts: nginx-hello-world  
  creationTimestamp: "2026-05-02T21:42:11Z"  
  name: nginx-hello-world  
  namespace: default  
  resourceVersion: "4086399"  
  uid: 90024d82-4a34-4693-a638-bdc735070653  
spec:  
  clusterIP: 10.108.89.38  
  clusterIPs:  
  - 10.108.89.38  
  internalTrafficPolicy: Cluster  
  ipFamilies:  
  - IPv4  
  ipFamilyPolicy: SingleStack  
  ports:  
  - port: 80  
    protocol: TCP  
    targetPort: 80  
  selector:  
    app: nginx-hello-world  
    rollouts-pod-template-hash: 6b4b5495dd  
  sessionAffinity: None  
  type: ClusterIP  
status:  
  loadBalancer: {}
```

```
apiVersion: v1  
kind: Service  
metadata:  
  annotations:  
    argo-rollouts.argo.dev/managed-by-rollouts: nginx-hello-world  
  creationTimestamp: "2026-05-03T14:46:21Z"  
  name: nginx-hello-world-canary  
  namespace: default  
  resourceVersion: "4088290"  
  uid: 3b9fa039-c3c9-4040-a718-197f0375ca2d  
spec:  
  clusterIP: 10.108.80.198  
  clusterIPs:  
  - 10.108.80.198  
  internalTrafficPolicy: Cluster  
  ipFamilies:  
  - IPv4  
  ipFamilyPolicy: SingleStack  
  ports:  
  - port: 80  
    protocol: TCP  
    targetPort: 80  
  selector:  
    app: nginx-hello-world  
    rollouts-pod-template-hash: 758bcb658f  
  sessionAffinity: None  
  type: ClusterIP  
status:  
  loadBalancer: {}
```

```
16h  
25s
```



Argo Workflows

Kubernetes-native workflow engine supporting DAG and step-based workflows.

[Documentation](#)



Like a K8s job, but more

```
apiVersion: batch/v1
kind: Job
metadata:
  name: hello-world
spec:
  template:
    spec:
      containers:
        - name: busybox
          image: busybox
          command: [ echo ]
          args: [ "hello world" ]
      restartPolicy: Never
```

```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  name: hello-world
spec:
  entrypoint: hello-world
  templates:
    - name: hello-world
      container:
        name: busybox
        image: busybox
        command: [ echo ]
        args: [ "hello world" ]
```

One step at a time

```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  name: hello-world-with-steps
spec:
  entrypoint: hello-world-stepped
  templates:
  - name: hello-world-stepped
    steps:
    - - name: hello
      inline:
        container:
          name: busybox
          image: busybox
          command: [ echo ]
          args: [ "hello..." ]
    - - name: world
      inline:
        container:
          name: alpine
          image: alpine
          command: [ echo ]
          args: [ "...world" ]
```

```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  name: hello-world-with-steps-and-dry
spec:
  entrypoint: hello-world-stepped
  templates:
  - name: say-what-template
    inputs:
      parameters:
      - name: say-what
    container:
      name: busybox
      image: busybox
      command: [ echo ]
      args: [ "{{inputs.parameters.say-what}}" ]
  - name: hello-world-stepped
    steps:
    - - name: hello
      template: say-what-template
      arguments:
        parameters:
        - name: say-what
          value: "hello..."
    - - name: world
      template: say-what-template
      arguments:
        parameters:
        - name: say-what
          value: "...world"
```

```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  name: hello-world-with-2d-steps-and-dry
spec:
  entrypoint: hello-world-stepped
  templates:
  [...]
  - name: hello-world-stepped
    steps:
    - - name: hello
      template: say-what-template
      arguments:
        parameters:
        - name: say-what
          value: "hello..."
    - - name: say-with-hello
      template: say-what-template
      arguments:
        parameters:
        - name: say-what
          value: "im with hello"
    - - name: world
      template: say-what-template
      arguments:
        parameters:
        - name: say-what
          value: "...world"
    - - name: way-with-world
      template: say-what-template
      arguments:
        parameters:
        - name: say-what
          value: "im with world"
```



Workflow Example 1

Workflow Example 2

Argo Events



THE LINUX FOUNDATION

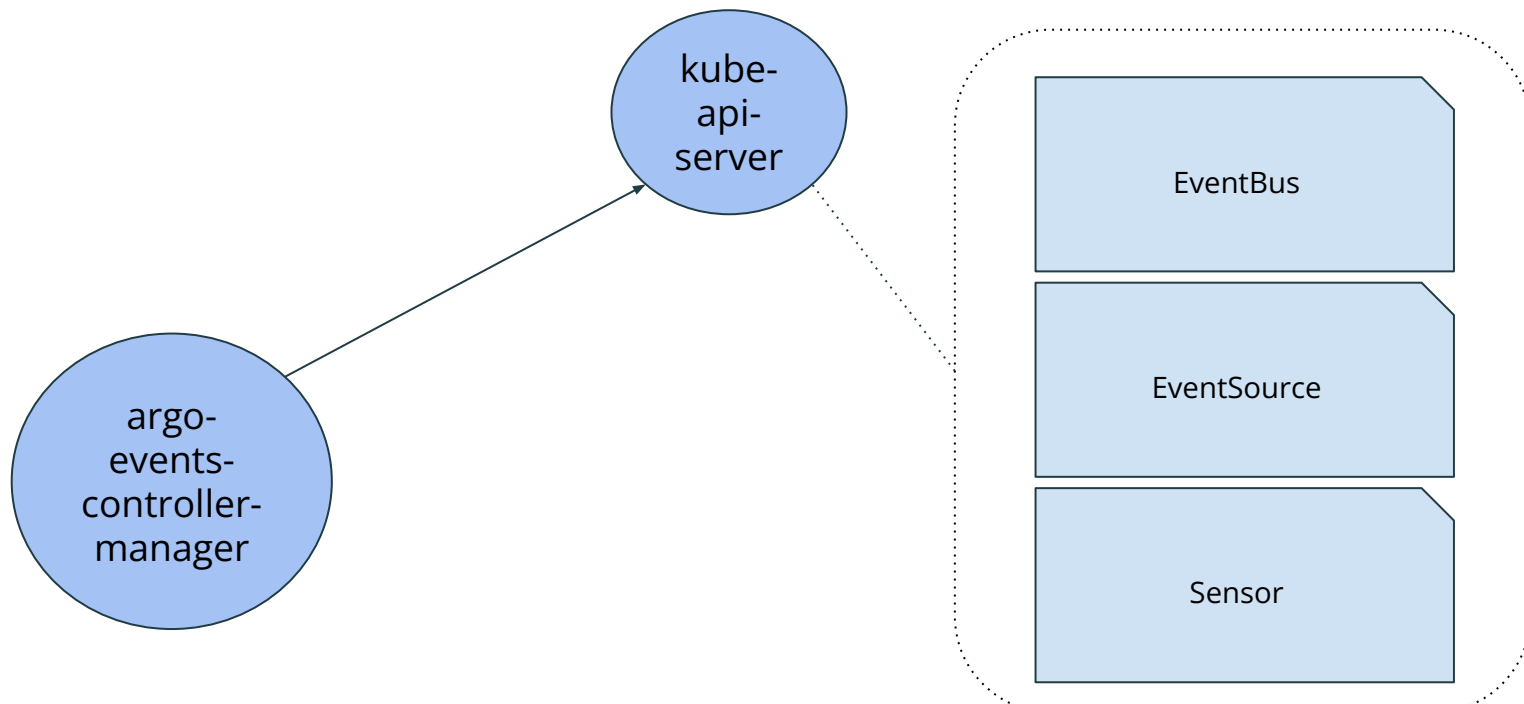
NORTH AMERICA



Embedded Linux
Conference

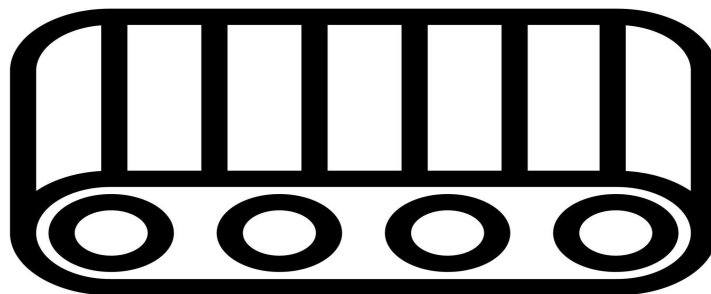
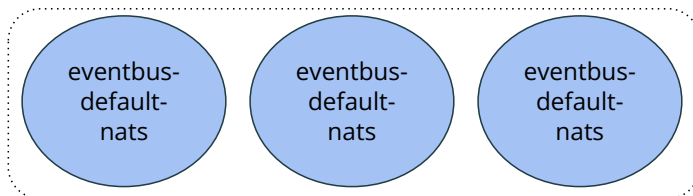


Argo Events, the controllers



All aboard the Events Queue. Choo Choo!

```
apiVersion: argoproj.io/v1alpha1
kind: EventBus
metadata:
  name: default
spec:
  nats:
    native:
      replicas: 3
      auth: token
```



Source

```
apiVersion: argoproj.io/v1alpha1
kind: EventSource
metadata:
  name: github
spec:
  service:
    ports:
      - name: webhook
        port: 12000
        targetPort: 12000
  github:
    nginx-hello-world:
      repositories:
        - owner: aaronteague
          names:
            - nginx-hello-world
      webhook:
        endpoint: /push
        port: "12000"
        method: POST
        url: https://github-webhook.example.com
      events:
        - **
  apiToken:
    name: github-access
    key: token
  insecure: true
  active: true
  contentType: json
```

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: github-webhook
spec:
  ingressClassName: nginx
  tls:
  - hosts:
    - github-webhook.example.com
    secretName: cert-secret
  rules:
  - host: github-webhook.example.com
    http:
      paths:
      - backend:
          service:
            name: github-eventsource-svc
            port:
              number: 12000
          path: /push
          pathType: Prefix
```

General Webhooks / Manage webhook

Access Collaborators

Code and automation

- Branches
- Tags
- Rules
- Actions
- Models Preview
- Webhooks**
- Copilot
- Environments
- Codespaces
- Pages

Security and quality

- Advanced Security
- Deploy keys
- Secrets and variables

Integrations

- GitHub Apps
- Email notifications

Settings Recent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *
https://github-webhook.thecloudinformer.com/push

Content type *
application/json

Secret

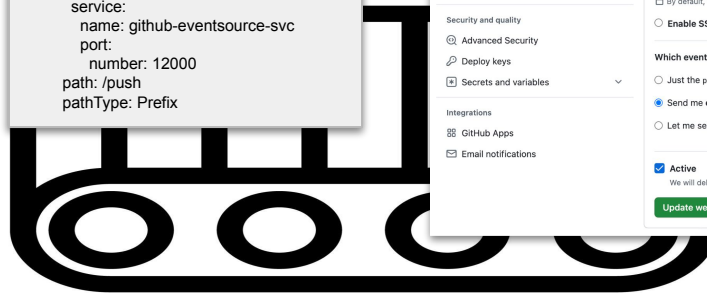
SSL verification
By default, we verify SSL certificates when delivering payloads.
 Enable SSL verification Disable (not recommended)

Which events would you like to trigger this webhook?

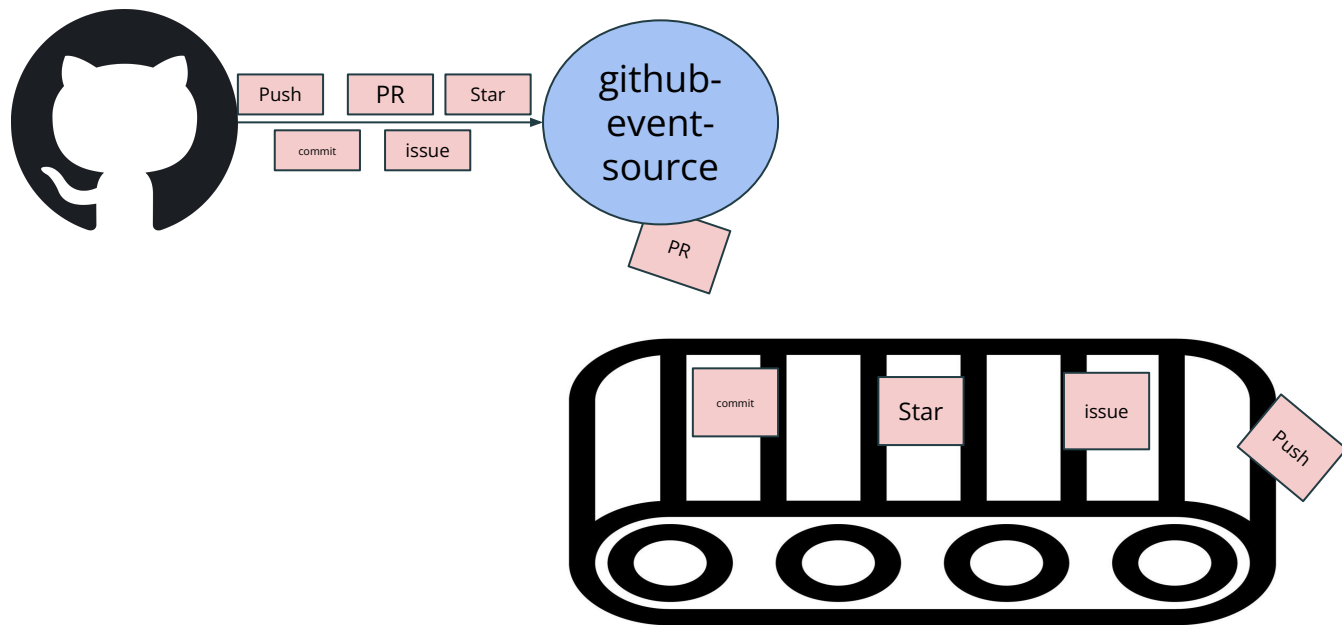
- Just the push event.
- Send me everything.
- Let me select individual events.

Active
We will deliver event details when this hook is triggered.

Update webhook Delete webhook



Let's make it rain



Finally coming to our senses



```
apiVersion: argoproj.io/v1alpha1
kind: Sensor
metadata:
  name: github
spec:
  template:
    serviceAccountName: operate-workflow-sa
  dependencies:
    - name: github-push-event
      eventSourceName: github
      eventName: nginx-hello-world
      filters:
        data:
          - path: headers.X-Github-Event
            type: string
            value:
              - push
[continued on next panel]
```

github-
event-
source

PR

commit

```
[continuation from last panel]
triggers:
  - template:
      name: argo-workflow-trigger
      argoWorkflow:
        parameters:
          - src:
              dependencyName: github-push-event
              dataKey: body.head_commit.message
              dest: spec.arguments.parameters.0.value
    operation: submit
  source:
    resource:
      apiVersion: argoproj.io/v1alpha1
      kind: Workflow
    metadata:
      generateName: workflow-template-print-message-
    spec:
      arguments:
        parameters:
          - name: commit-message
            entrypoint: say-the-line-bart
      templates:
        - name: say-the-line-bart
          inputs:
            parameters:
              - name: commit-message
            container:
              image: busybox
              command: [ echo ]
              args: [ "{{inputs.parameters.commit-message}}" ]
```

kube-
api-
server

Workflow

Event Source

- AMQP
- AWS SNS
- AWS SQS
- Azure Service Bus
- Azure Queue Storage
- Calendar
- Emitter
- File
- GCP Pub/Sub
- GitHub
- GitLab
- Bitbucket
- Kafka
- Minio
- MQTT
- NATS
- NSQ
- Redis
- Redis Streams
- Resource
- Webhook
- Pulsar
- Alibabacloud MNS/SMQ
- Slack

Triggers

- Argo Workflow Trigger
 - AWS Lambda
 - HTTP Trigger
 - NATS Trigger
 - Kafka Trigger
 - Kubernetes Object Trigger
- Log
 - OpenWhisk Trigger
 - Slack Trigger
 - Azure Event Hubs
 - Pulsar Trigger
 - Build Your Own Trigger

Putting it all together



OPEN SOURCE SUMMIT

THE LINUX FOUNDATION

NORTH AMERICA



Embedded Linux
Conference



Example 1

Example 2

The Setup

- Run a CI/CD job (Github, GitLab, BitBucket)
- Argo Events watches for “Deploy” stage
- Events starts a workflow
- Workflow pushes updated Rollout to Git
- CD updates Rollout in K8s cluster
- Workflow (*continued*)
 - Watches Rollout for updated status
 - (*if fail*) push previous Rollout to Git
 - Notify via a rest request

```

apiVersion: argoproj.io/v1alpha1
kind: Sensor
metadata:
  name: watch-github-deploy-job
spec:
dependencies:
- name: test-dep
eventName: nginx-hello-world-event
eventSourceName: github
filters:
  data:
  - path: headers.X-Github-Event
    type: string
    value:
      - workflow_job
  - path: body.workflow_job.name
    type: string
    value:
      - deploy
  - path: body.workflow_job.status
    type: string
    value:
      - completed
triggers:
- template:
  name: argo-workflow-trigger
  argoWorkflow:
    operation: submit
    source:
      resource:
        apiVersion: argoproj.io/v1alpha1
        kind: Workflow
        metadata:
          generateName: deploy-to-manifest-repo-
        spec:
          workflowTemplateRef:
            name: deploy-to-manifest-repo
        arguments:
          parameters:
            - name: image
              value: ""
      parameters:
        - src:
            dependencyName: test-dep
            dataKey: body.head_sha
            dest: "myimagerepo/nginx:{{spec.arguments.parameters.0.value}}"

```

```

apiVersion: argoproj.io/v1alpha1
kind: WorkflowTemplate
metadata:
  name: deploy-to-manifest-repo
spec:
  volumes:
  - name: github-ssh-key
    secret:
      secretName: github-ssh-key
      defaultMode: 0400
  entrypoint: deploy
  arguments:
    parameters:
      - name: image
        value: gimme_image
  templates:
  [...]

```

```

[.]
templates:
- name: deploy
  steps:
  - name: push-new-image
    template: git-push-template
    arguments:
      parameters:
        - name: image
          value: gimme_image
  - name: watch-rollout
    inline:
      script:
        image: alpine/kubectl:1.36.0
        command: [sh]
        source: |
          while true; do
            sleep 10;
            PHASE=$(kubectl get rollout nginx-h
            if [ "$PHASE" != "Running" ]; then
              echo "$PHASE" > /tmp/phase.txt
            fi
          done
        outputs:
          parameters:
            - name: phase
              valueFrom:
                default: "undefined"
                path: /tmp/phase.txt
  - name: rollback
    when: "{{(steps.watch-rollout.outputs.param
    template: git-push-template
    arguments:
      parameters:
        - name: image
          value: "{{(steps.push-new-image.outputs
  - name: notify-someone
    inline:
      http:
        url: "https://www.whatever.com"
        method: "GET"
[.]

```

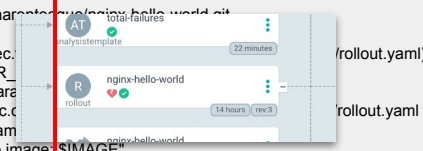
```

[...]
```

```

- name: git-push-template
  inputs:
    parameters:
      - name: image
  script:
    image: alpine/git
    command: [sh]
    volumeMounts:
      - mountPath: /root/.ssh/id_rsa
        name: github-ssh-key
        subPath: id_rsa
    source: |
      ssh-keyscan github.com >> ~/.ssh/known_hosts
      git config --global user.email "automation@automation.com"
      apk add yq
      git clone git@github.com:aaarotter/nginx-hello-world.git
      cd nginx-hello-world
      PRIOR_IMAGE=$(yq '.spec.'
      echo "prior image: $PRIOR_
      export IMAGE="{{(inputs.para
      yq e -i '.spec.template.spec.c
      git add manifests/rollout.yam
      git commit -m "Updating to image: $IMAGE"
      git push
      echo "$PRIOR_IMAGE" > /tmp/prior_image.txt
  outputs:
    parameters:
      - name: prior_image
        valueFrom:
          default: "undefined"
          path: /tmp/prior_image.txt

```



You have, the power!



OPEN SOURCE SUMMIT



Conference

NORTH AMERICA



Thanks for joining! Any Questions?



OPEN SOURCE SUMMIT

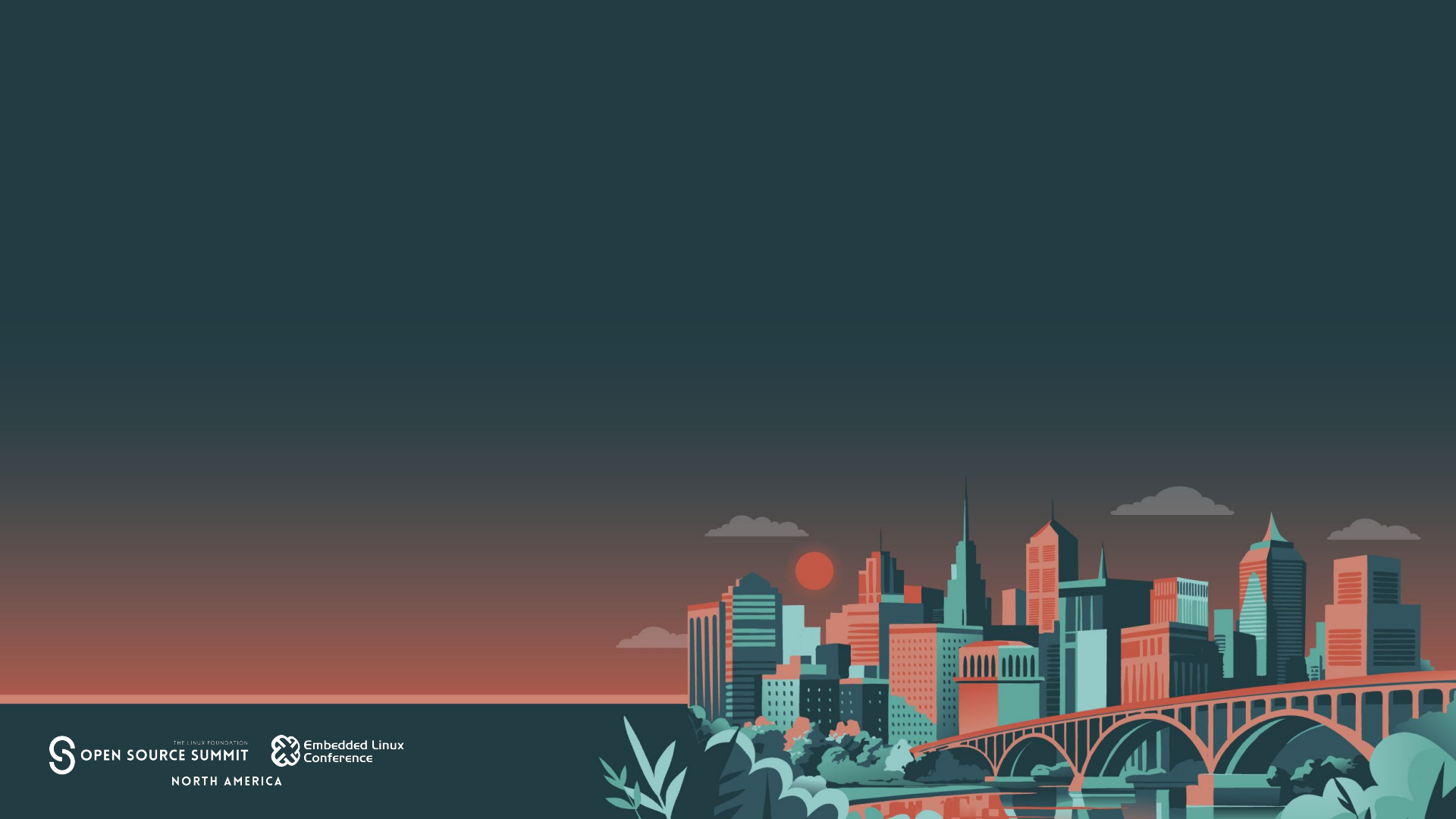
THE LINUX FOUNDATION

NORTH AMERICA



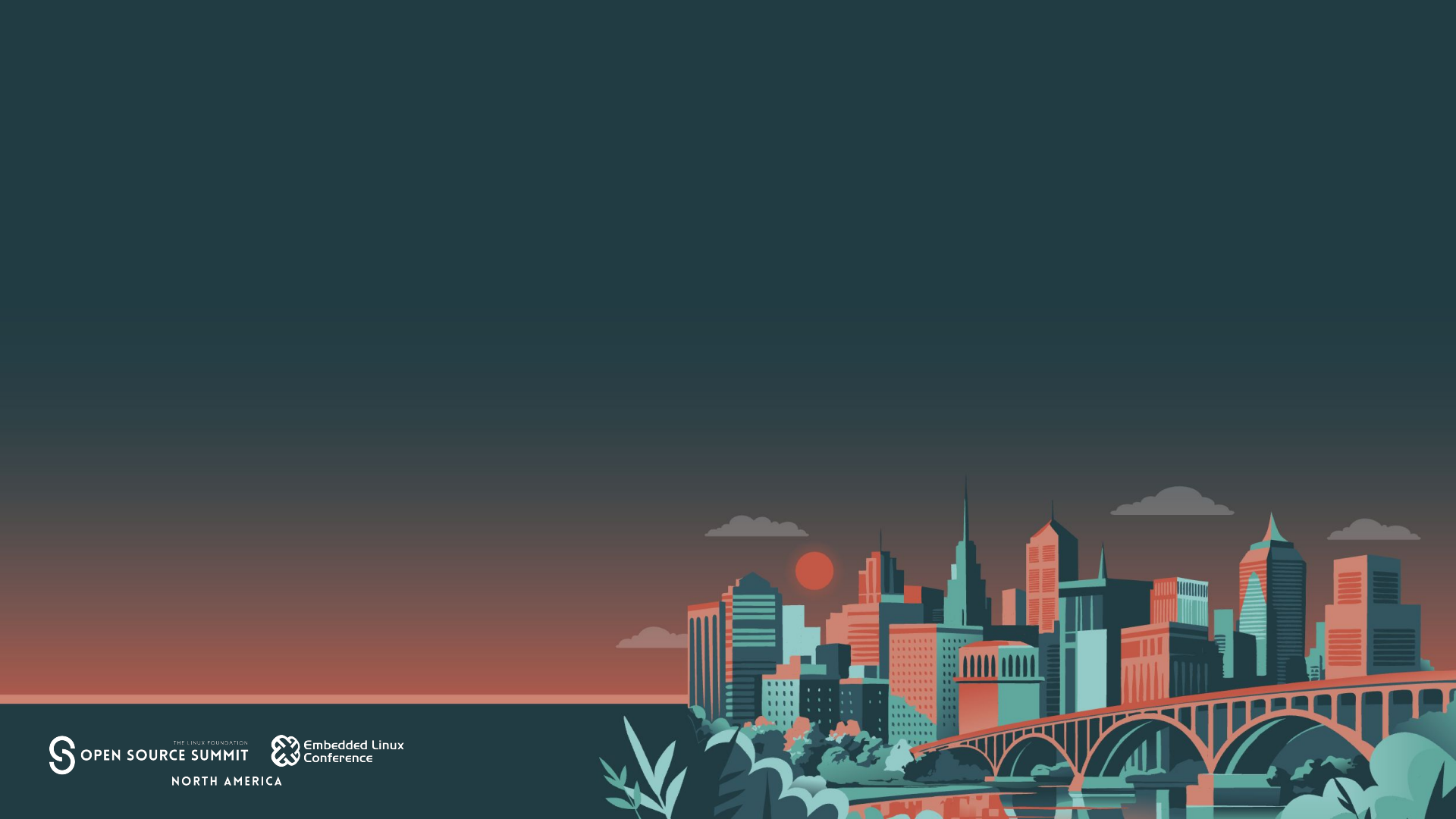
Embedded Linux
Conference





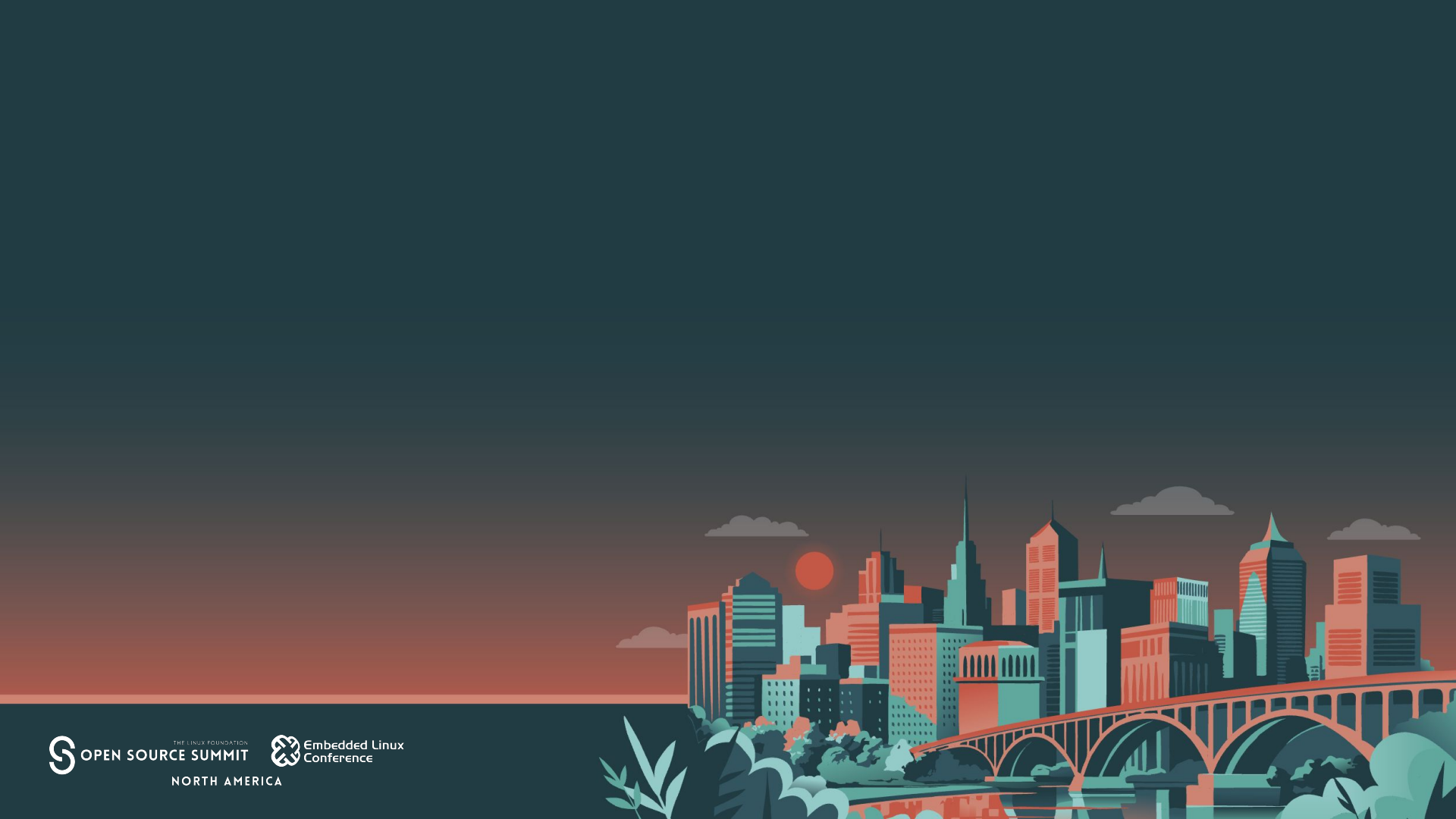
 OPEN SOURCE SUMMIT
NORTH AMERICA

THE LINUX FOUNDATION
 Embedded Linux
Conference



 OPEN SOURCE SUMMIT
NORTH AMERICA

THE LINUX FOUNDATION
 Embedded Linux
Conference



 OPEN SOURCE SUMMIT
NORTH AMERICA

THE LINUX FOUNDATION
 Embedded Linux
Conference

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: endless-curl
spec:
  selector:
    matchLabels:
      app: endless-curl
  template:
    metadata:
      labels:
        app: endless-curl
    spec:
      replicas: 1
      containers:
      - image: curlimages/curl:8.19.0
        name: curl
        command: [ "/bin/sh", "-c", "--" ]
        args: [ "while true; do sleep 1; curl -k https://nginx.thecloudinformer.com ; done;" ]
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: endless-curl
spec:
  selector:
    matchLabels:
      app: endless-curl
  template:
    metadata:
      labels:
        app: endless-curl
    spec:
      replicas: 1
      containers:
      - image: curlimages/curl:8.19.0
        name: curl
        command: [ "/bin/sh", "-c", "--" ]
        args: [ "while true; do sleep 1; curl -k https://nginx.thecloudinformer.com ; done;" ]
```



THE LINUX FOUNDATION

NORTH AMERICA

