RANCHER

CNCF Partner Webinar

Declarative Host Upgrades From Within Kubernetes

March 4, 2020

Who's who

Adrian Goins
Director of Community
Rancher Labs

Dax McDonald
Software Engineer
Rancher Labs

Jacob Blain Christen
Principal Software Engineer
Rancher Labs

AGENDA

1 **Rancher Overview**
Adrian Goins, Director of Community

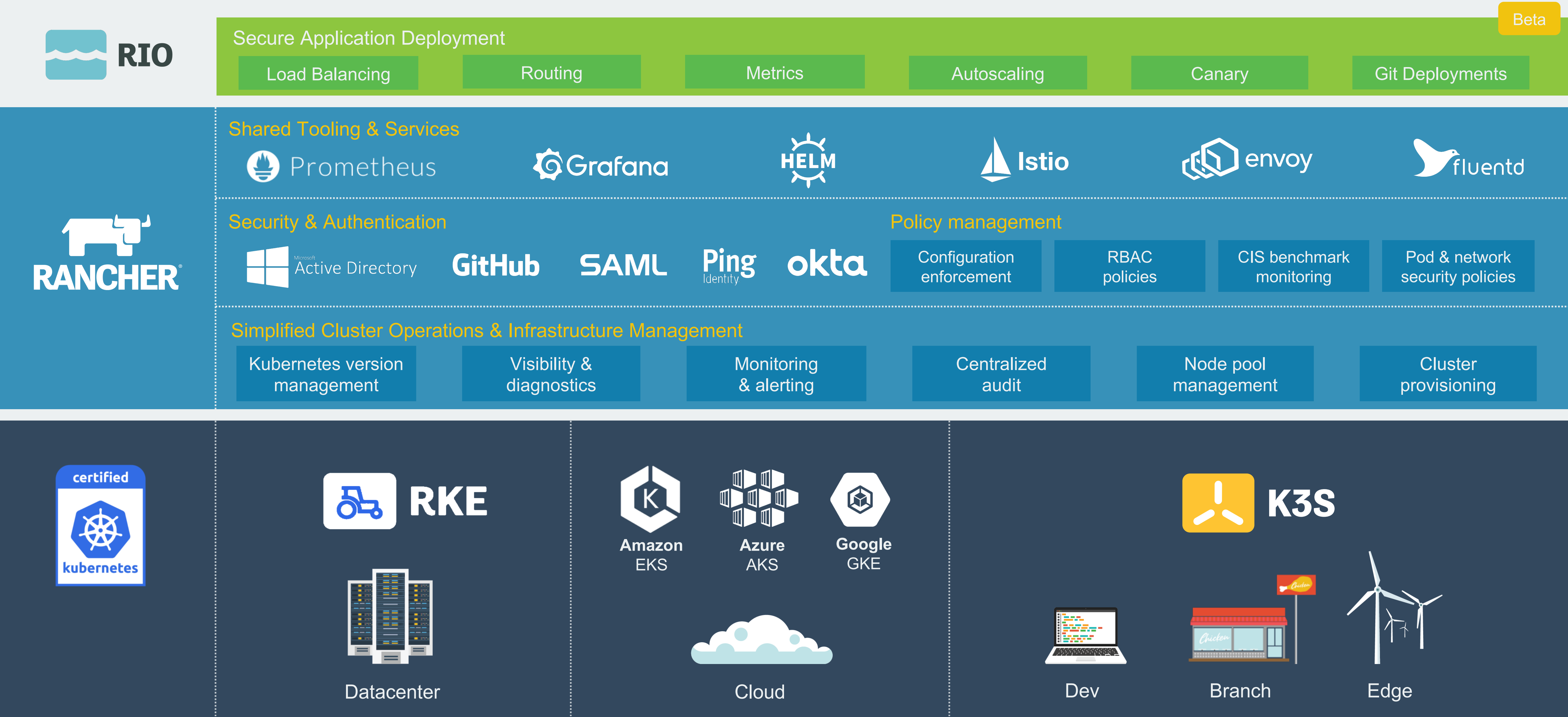
2 **An Overview of the SUC**
Dax McDonald, Software Engineer

3 **The SUC in Action**
Jacob Blain Christen,Principal Software Engineer

4 **Q&A**

Rancher Overview

Rancher’s recipe for production quality Kubernetes at scale



RIO

Beta

Secure Application Deployment

- Load Balancing
- Routing
- Metrics
- Autoscaling
- Canary
- Git Deployments

RANCHER

Shared Tooling & Services

- Prometheus
- Grafana
- HELM
- Istio
- envoy
- fluentd

Security & Authentication

- Microsoft Active Directory
- GitHub
- SAML
- Ping Identity
- okta

Policy management

- Configuration enforcement
- RBAC policies
- CIS benchmark monitoring
- Pod & network security policies

Simplified Cluster Operations & Infrastructure Management

- Kubernetes version management
- Visibility & diagnostics
- Monitoring & alerting
- Centralized audit
- Node pool management
- Cluster provisioning

certified kubernetes

RKE

Datacenter

Amazon EKS

Azure AKS

Google GKE

Cloud

K3S

Dev

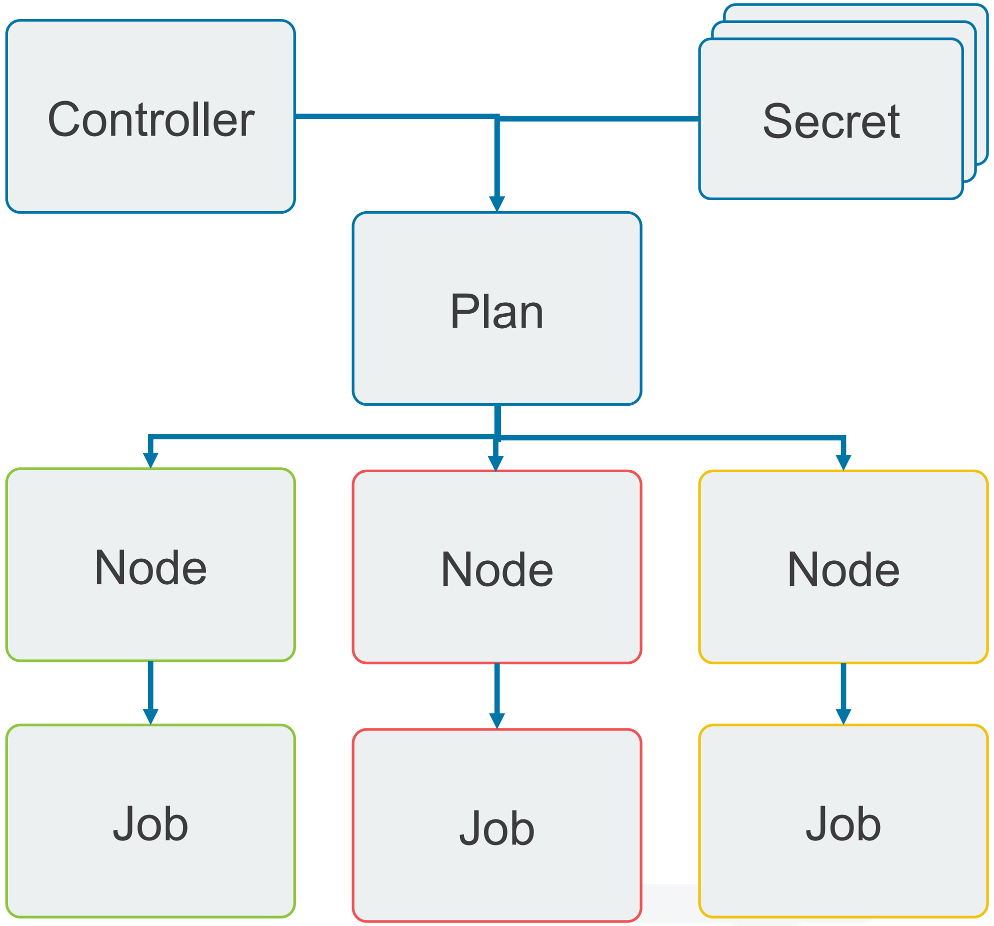
Branch

Edge

System Upgrade Controller

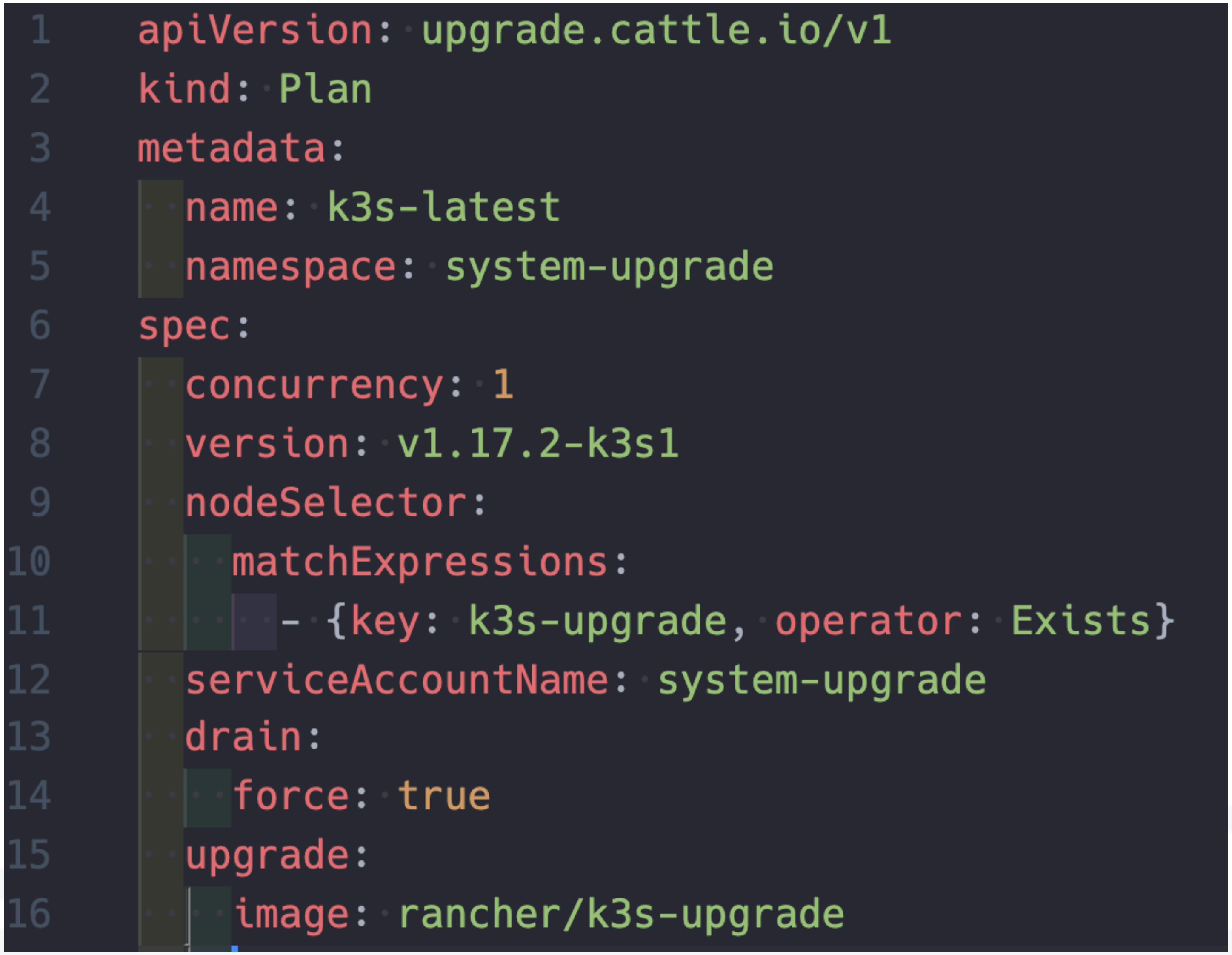
System Upgrade Controller Architecture

The ***Controller*** manages ***Plans*** by selecting ***Nodes*** to run upgrade ***Jobs*** on. A ***Plan*** defines which ***Nodes*** are eligible for upgrade by specifying a label selector. When a ***Job*** has run to completion successfully the ***Controller*** will label the ***Node*** on which it ran according to the ***Plan*** that was applied by the ***Job***.



Plan CRD

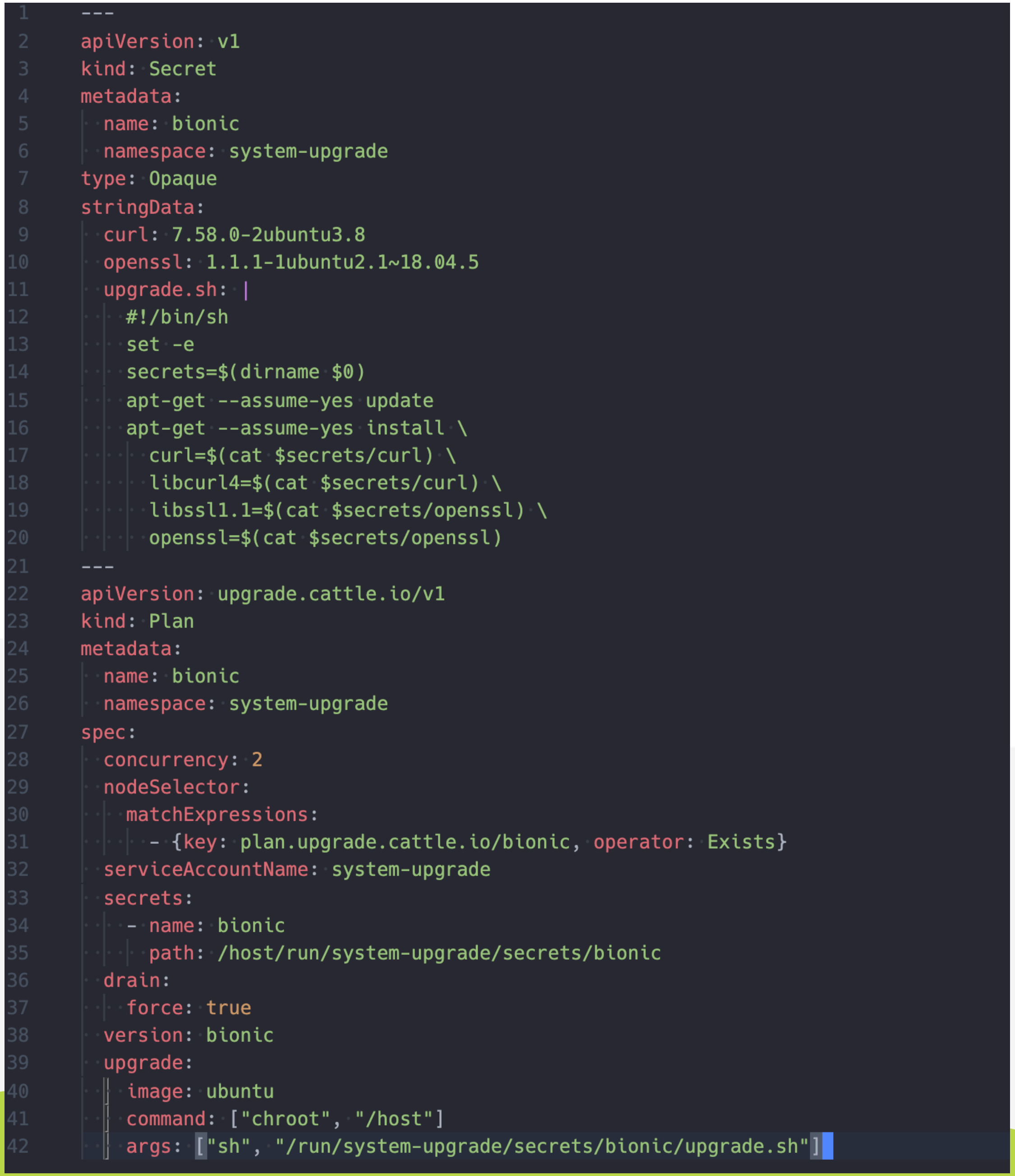
- Leveraging Kubernetes CRDs with the system-upgrade controller creates a declarative API.
- A Plan is an outstanding intent to mutate a system
- Its a declarative definition to mutate a system
- Image-based approach



```yaml
apiVersion: upgrade.cattle.io/v1
kind: Plan
metadata:
  name: k3s-latest
  namespace: system-upgrade
spec:
  concurrency: 1
  version: v1.17.2-k3s1
  nodeSelector:
    matchExpressions:
      - {key: k3s-upgrade, operator: Exists}
  serviceAccountName: system-upgrade
  drain:
    force: true
  upgrade:
    image: rancher/k3s-upgrade
```

Plan CRD continued

- Optional support for secrets to parameterize plans
- Allows for more granular updates at the package level



```yaml
---
apiVersion: v1
kind: Secret
metadata:
  name: bionic
  namespace: system-upgrade
type: Opaque
stringData:
  curl: 7.58.0-2ubuntu3.8
  openssl: 1.1.1-1ubuntu2.1~18.04.5
  upgrade.sh: |
    #!/bin/sh
    set -e
    secrets=$(dirname $0)
    apt-get --assume-yes update
    apt-get --assume-yes install \
      curl=$(cat $secrets/curl) \
      libcurl4=$(cat $secrets/curl) \
      libssl1.1=$(cat $secrets/openssl) \
      openssl=$(cat $secrets/openssl)
---
apiVersion: upgrade.cattle.io/v1
kind: Plan
metadata:
  name: bionic
  namespace: system-upgrade
spec:
  concurrency: 2
  nodeSelector:
    matchExpressions:
      - {key: plan.upgrade.cattle.io/bionic, operator: Exists}
  serviceAccountName: system-upgrade
  secrets:
    - name: bionic
      path: /host/run/system-upgrade/secrets/bionic
  drain:
    force: true
  version: bionic
  upgrade:
    image: ubuntu
    command: ["chroot", "/host"]
    args: ["sh", "/run/system-upgrade/secrets/bionic/upgrade.sh"]
```

From https://github.com/rancher/system-upgrade-controller/blob/master/examples/ubuntu/bionic.yaml

Demo Time!

Thank you

@Rancher_Labs