

# The operator antipattern

Kubernetes Community Days London 2024

Gerald Schmidt



**Cindy Sridharan**

@copyconstruct · [Follow](#)



As more and more exec types begin to wonder why companies are overstaffed, we're going to enter an era where engineers are going to be questioned why they run such complex systems - think every tech trend of the 2010's - when much simpler ones would suffice.

Brace yourselves.

4:19 PM · Nov 7, 2022



596



Reply

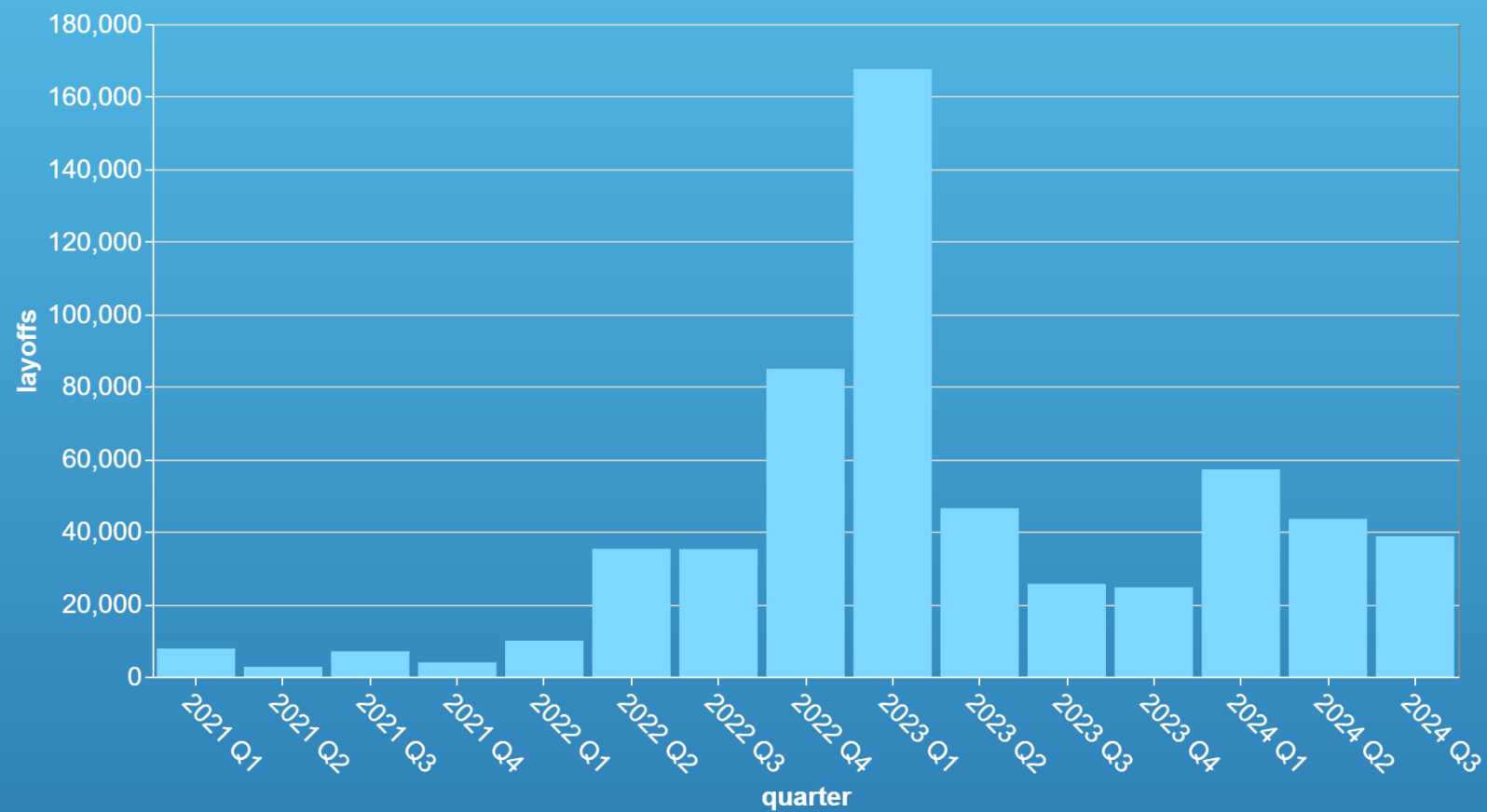


Copy link

[Read 16 replies](#)

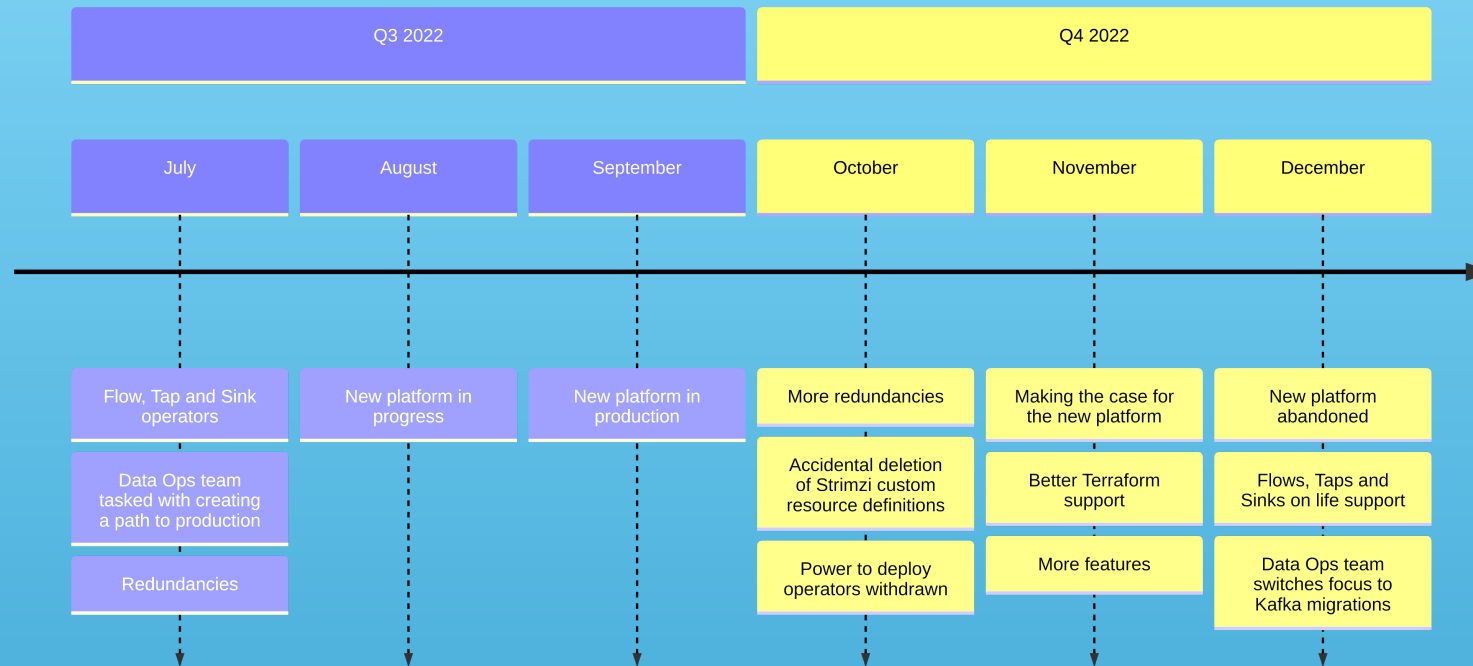
Classic Sridharan:

[Testing Microservices, the sane way \(2017\)](#) | [Testing in Production, the safe way \(2018\)](#) | [Testing in Production: the hard parts \(2019\)](#)



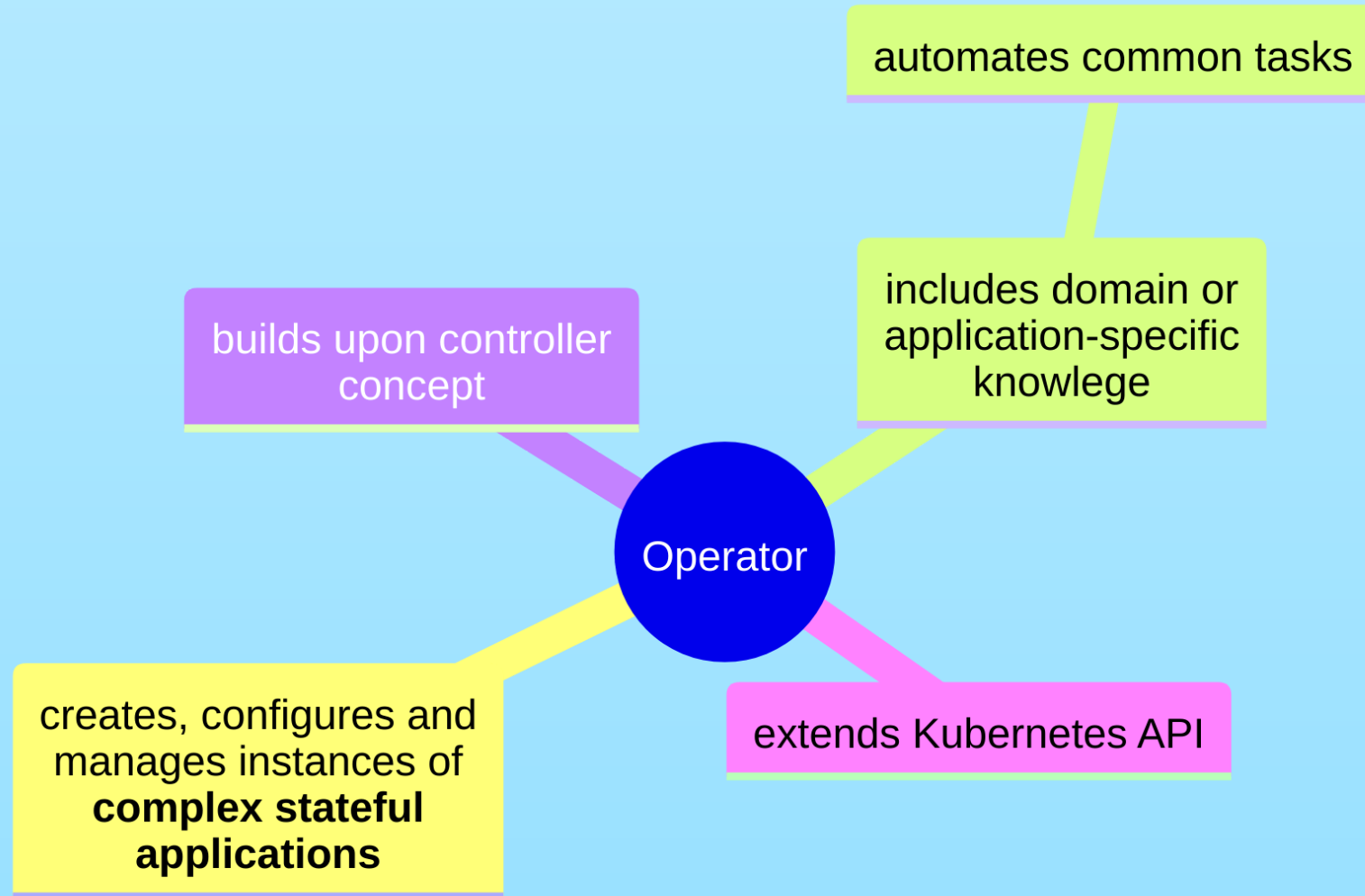
Source: layoffs.fyi

# Startup world in 2022



An Operator is an application-specific controller that extends the Kubernetes API to create, configure, and manage instances of complex stateful applications on behalf of a Kubernetes user. It builds upon the basic Kubernetes resource and controller concepts but includes domain or application-specific knowledge to automate common tasks.

Brandon Philips, [Introducing Operators: Putting Operational Knowledge into Software](#) (2016)



## Great expectations ①

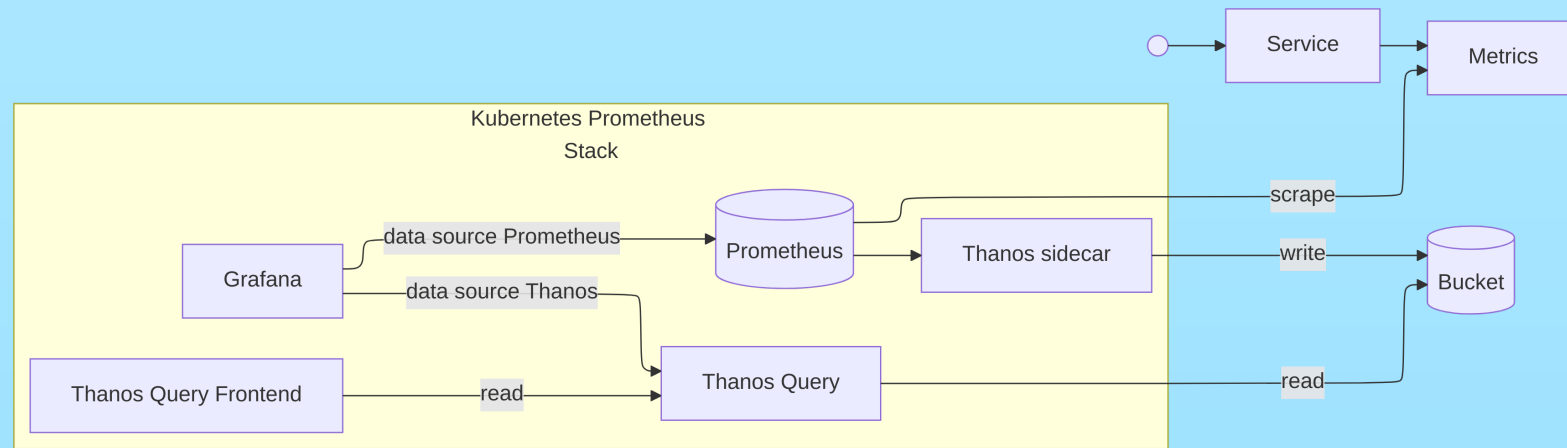
Operators promised to solve the problem of stateful applications on Kubernetes.

One issue is that operators didn't do so convincingly.

Another is that arguably Kubernetes never had a stateful application problem; it had a persistent volume problem.

See [Object storage for stateful applications on Kubernetes \(2022\)](#).

# Tiered storage: Thanos



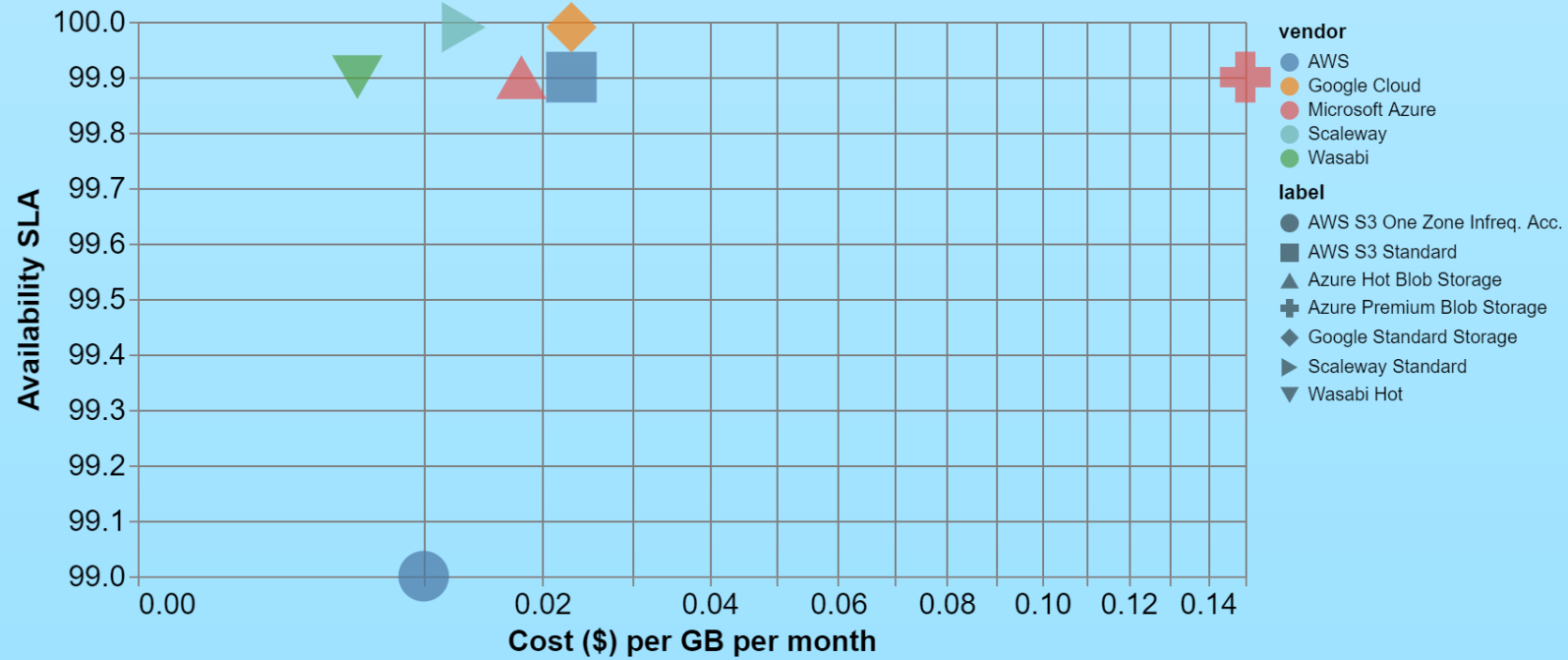


# Object storage only: WarpStream

‘Zero disks would be better.’

Richard Artoul, [Tiered Storage Won't Fix Kafka](#) (28 April 2024). On 9 September 2024, Confluent announced that it had [acquired WarpStream](#).

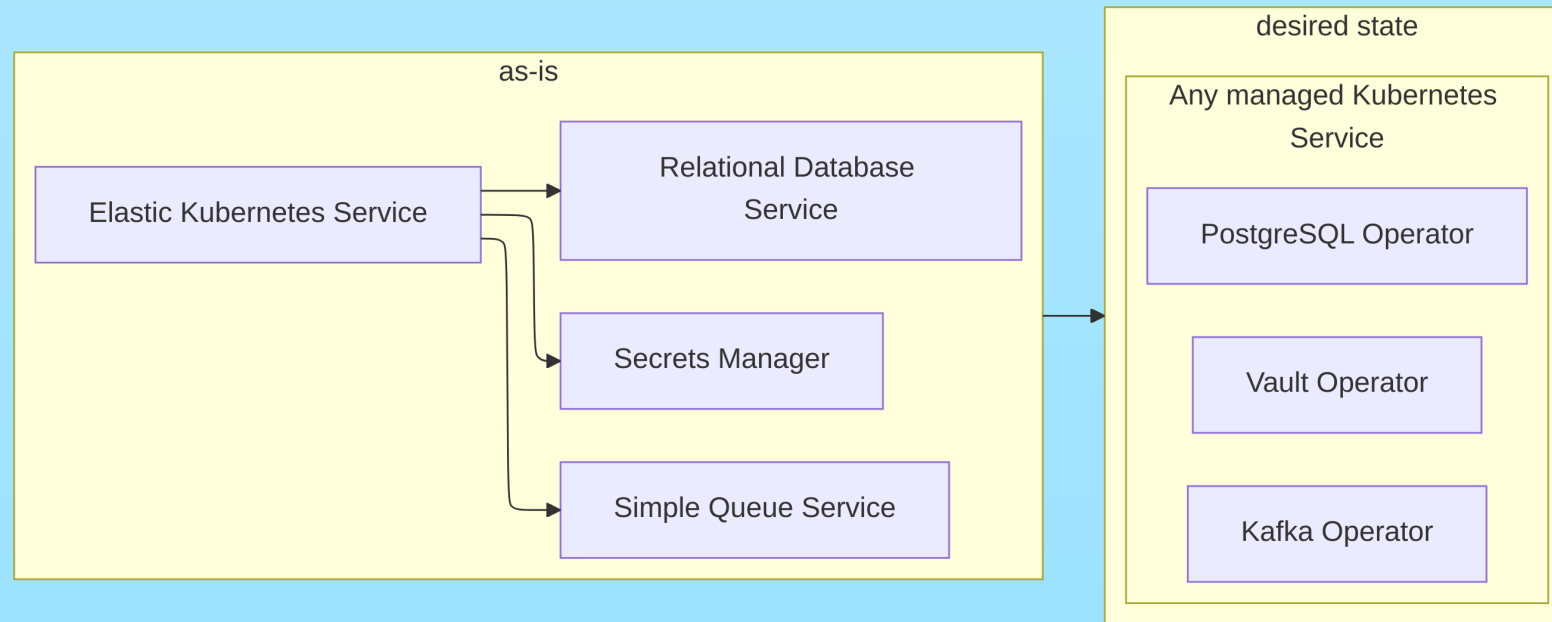
# Waiting for COSI



See [container-object-storage-interface.github.io](https://github.com/container-storage-interface/spec).

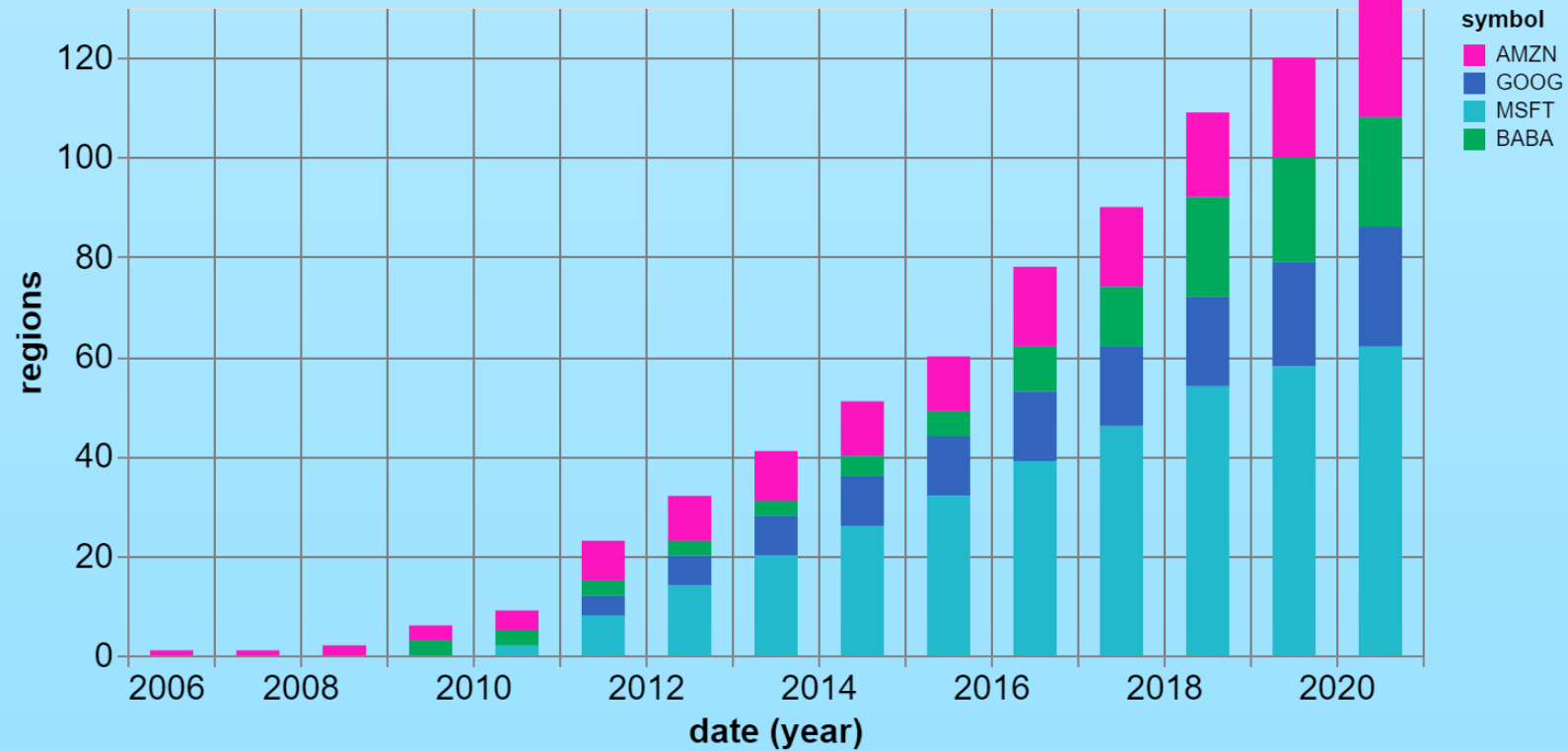
## Great expectations ②

Operators promised to help us move from managed and mostly proprietary services to portable Kubernetes environments hosting many open source operators.



# There was another group cheering us on

That group is best described as everyone except Amazon.

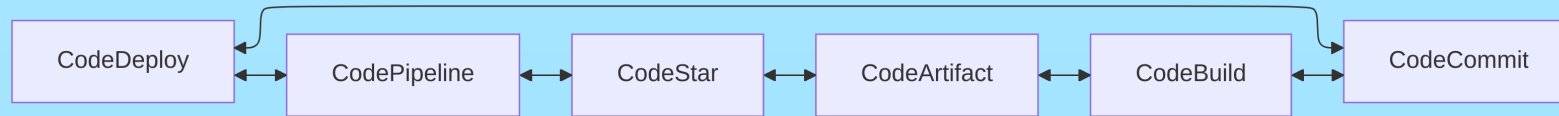


Regions differ significantly in size. Some values have been interpolated.

# What a three-year head start gives you

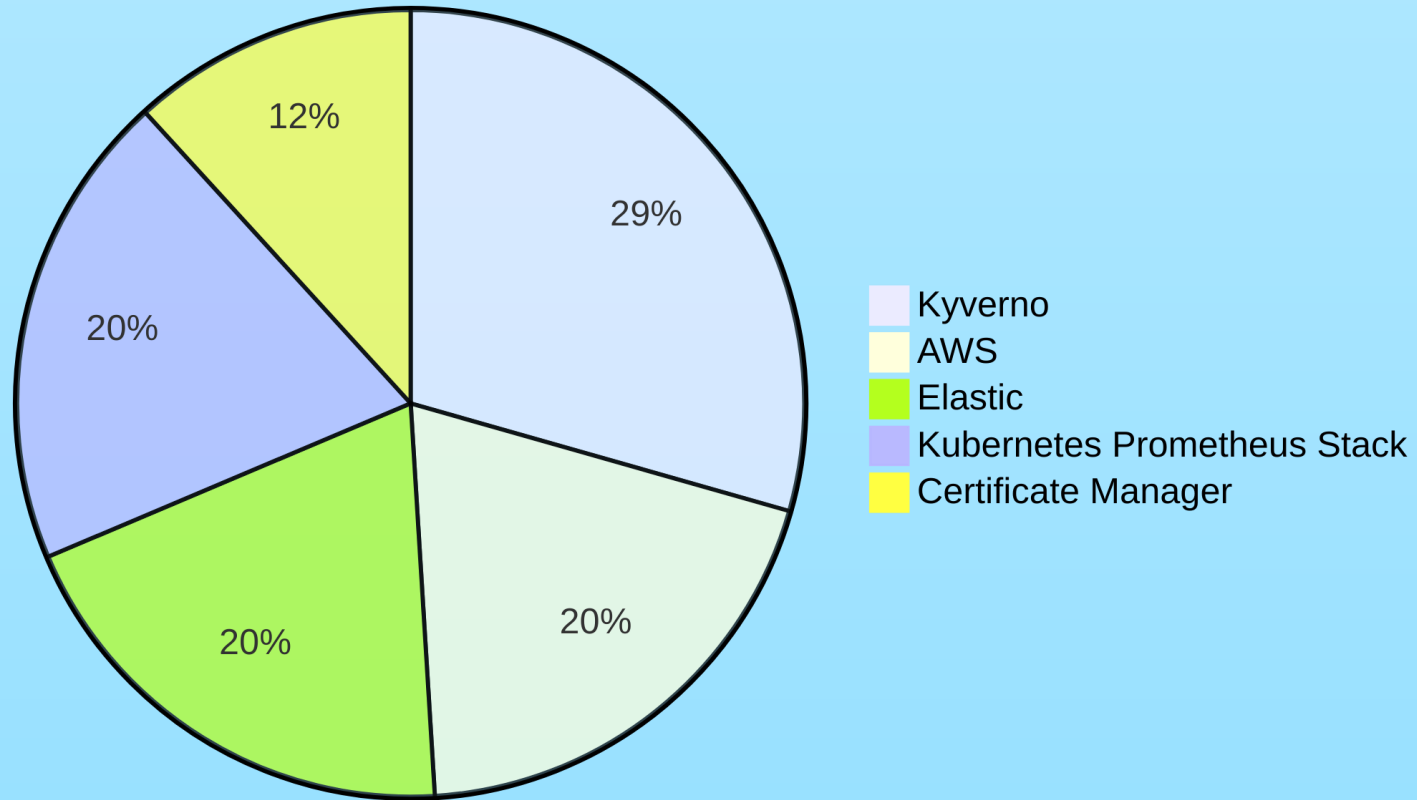
Alexa for Business, Amazon AppFlow, Amazon Augmented AI, Amazon Bedrock, Amazon Braket, Amazon Chime, Amazon CodeGuru, Amazon Comprehend, Amazon Connect, Amazon DocumentDB, Amazon EventBridge, Amazon DataZone, Amazon Forecast, Amazon Fraud Detector, Amazon GameLift, Amazon Honeycode, Amazon Interactive Video Service, Amazon Kendra, Amazon Keyspaces, Amazon Lex, Amazon Macie, Amazon Managed Blockchain, Amazon MQ, Amazon Personalize, Amazon Polly, Amazon QLDB, Amazon Redshift, Amazon Rekognition, Amazon SageMaker, Amazon Sumerian, Amazon Textract, Amazon Transcribe, Amazon Translate, API Gateway, Application Discovery Service, AppStream 2.0, Artifact, Athena, AWS Amplify, AWS App Mesh, AWS AppConfig, AWS AppSync, AWS Auto Scaling, AWS Backup, AWS Budgets, AWS Chatbot, AWS Cloud Map, AWS Compute Optimizer, AWS Cost Explorer, AWS Data Exchange, AWS DeepComposer, AWS DeepLens, AWS DeepRacer, AWS Firewall Manager, AWS Glue, AWS IQ, AWS Lake Formation, AWS License Manager, AWS Marketplace Subscriptions, AWS Migration Hub, AWS Organizations, AWS Outposts, AWS RoboMaker, AWS Single Sign-On, AWS Snow Family, AWS Transfer Family, AWS Well-Architected Tool, Batch, Certificate Manager, Cloud9, CloudFormation, CloudFront, CloudHSM, CloudSearch, CloudTrail, CloudWatch, CodeArtifact, CodeBuild, CodeCommit, CodeDeploy, CodePipeline, CodeStar, Cognito, Config, Control Tower, Data Pipeline, Database Migration Service, DataSync, Detective, Device Farm, Direct Connect, Directory Service, DynamoDB, EC2, EC2 Image Builder, EFS, Elastic Beanstalk, Elastic Container Registry, Elastic Container Service, Elastic Kubernetes Service, Elastic Transcoder, ElastiCache, Elasticsearch Service, Elemental Appliances & Software, EMR, FreeRTOS, FSx, Global Accelerator, Ground Station, GuardDuty, IAM, Inspector, IoT 1-Click, IoT Analytics, IoT Core, IoT Device Defender, IoT Device Management, IoT Events, IoT Greengrass, IoT SiteWise, IoT Things Graph, Key Management Service, Kinesis, Kinesis Video Streams, Lambda, Launch Wizard, Lightsail, Managed Services, MediaConnect, MediaConvert, MediaLive, MediaPackage, MediaStore, MediaTailor, Mobile Hub, MSK, Neptune, OpsWorks, Personal Health Dashboard, Pinpoint, QuickSight, RDS, Resource Access Manager, Route 53, S3, S3 Glacier, Secrets Manager, Security Hub, Server Migration Service, Serverless Application Repository, Service Catalog, Simple Email Service, Simple Notification Service, Simple Queue Service, Step Functions, Storage Gateway, Support, SWF, Systems Manager, Trusted Advisor, VPC, WAF & Shield, WorkDocs, WorkLink, WorkMail, WorkSpaces, X-Ray

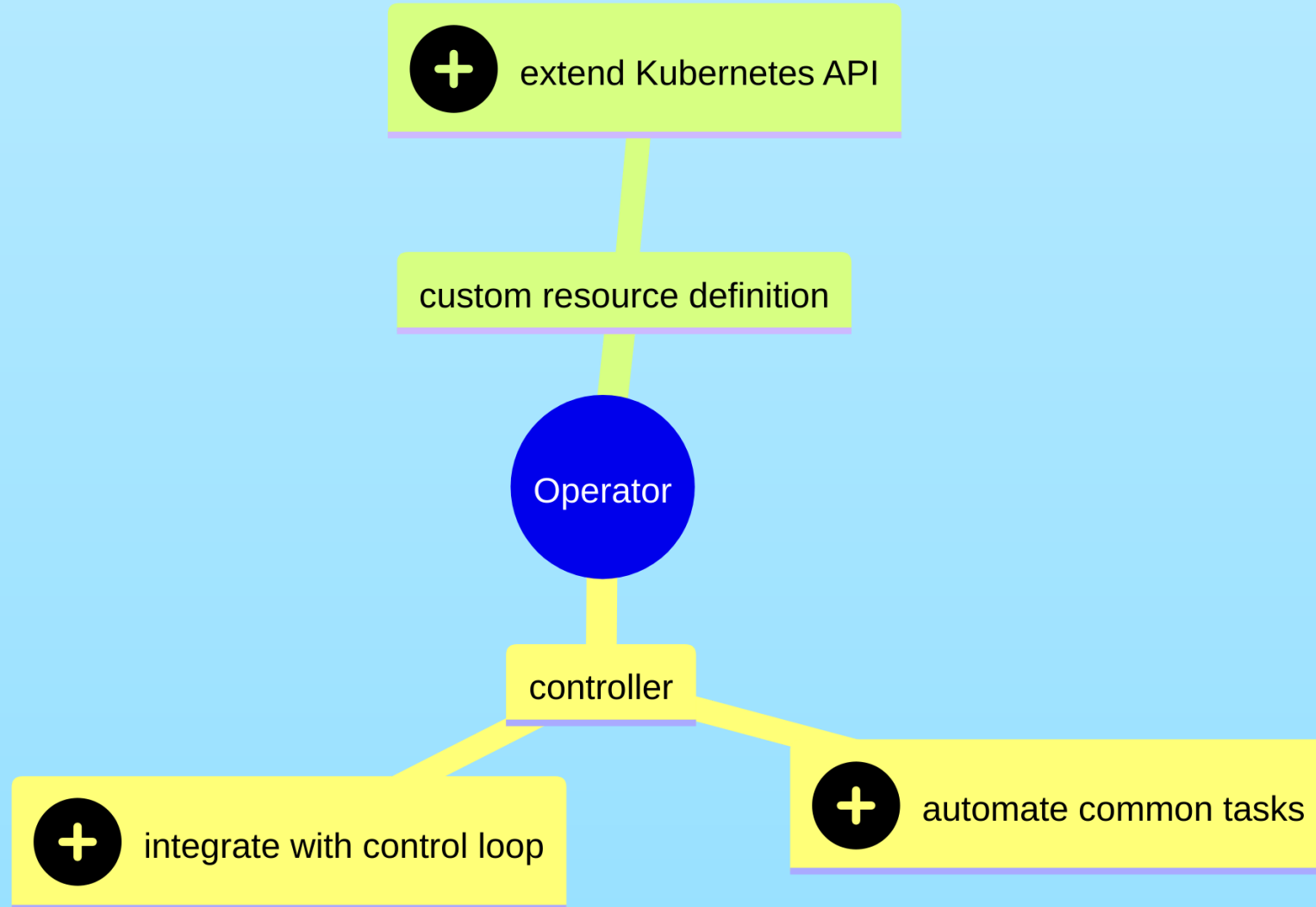
## Competitive advantage or liability



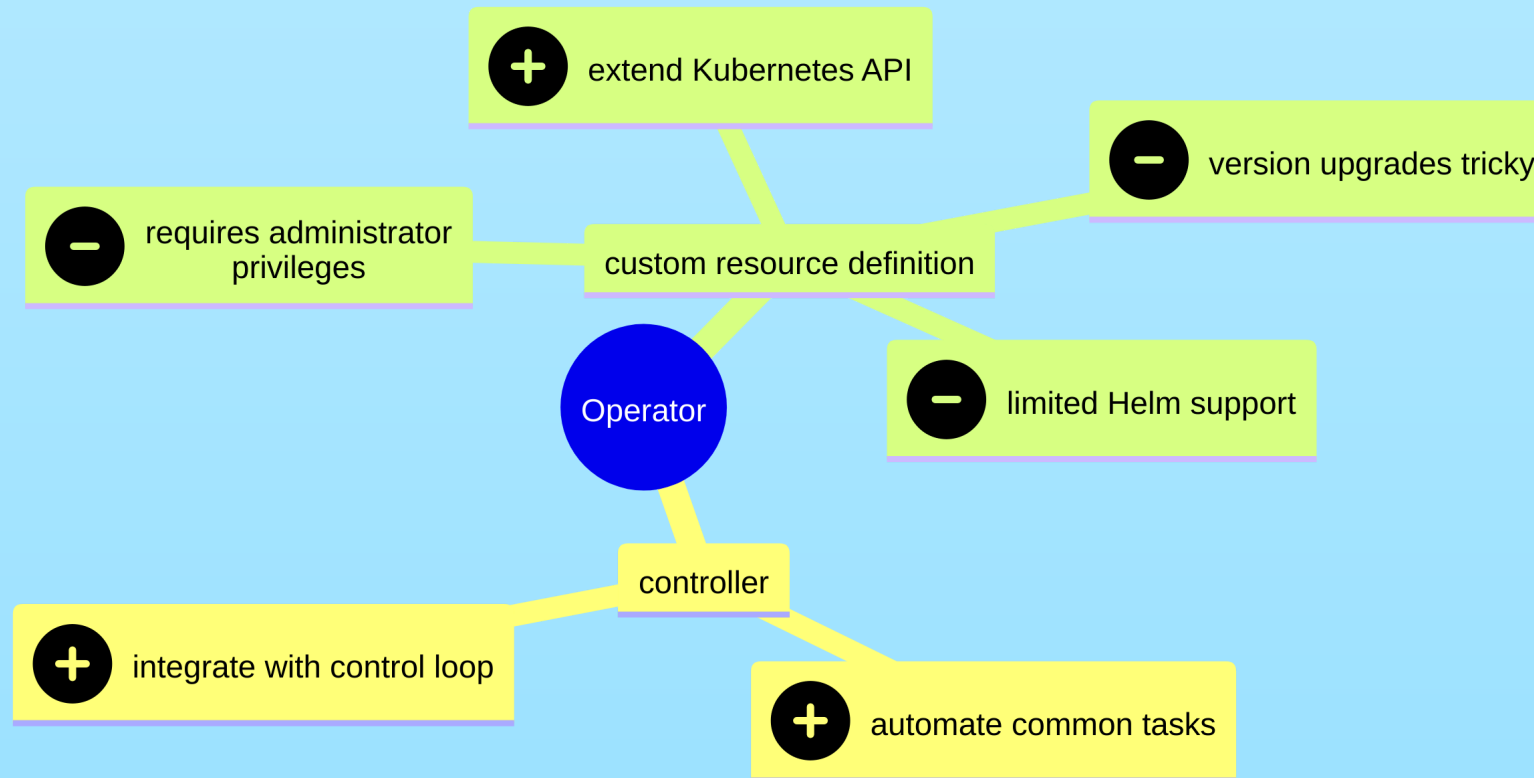
# The operators we got weren't the ones we expected

We kept using managed services as before and every self-respecting open source project added custom resource definitions.









# Developer experience

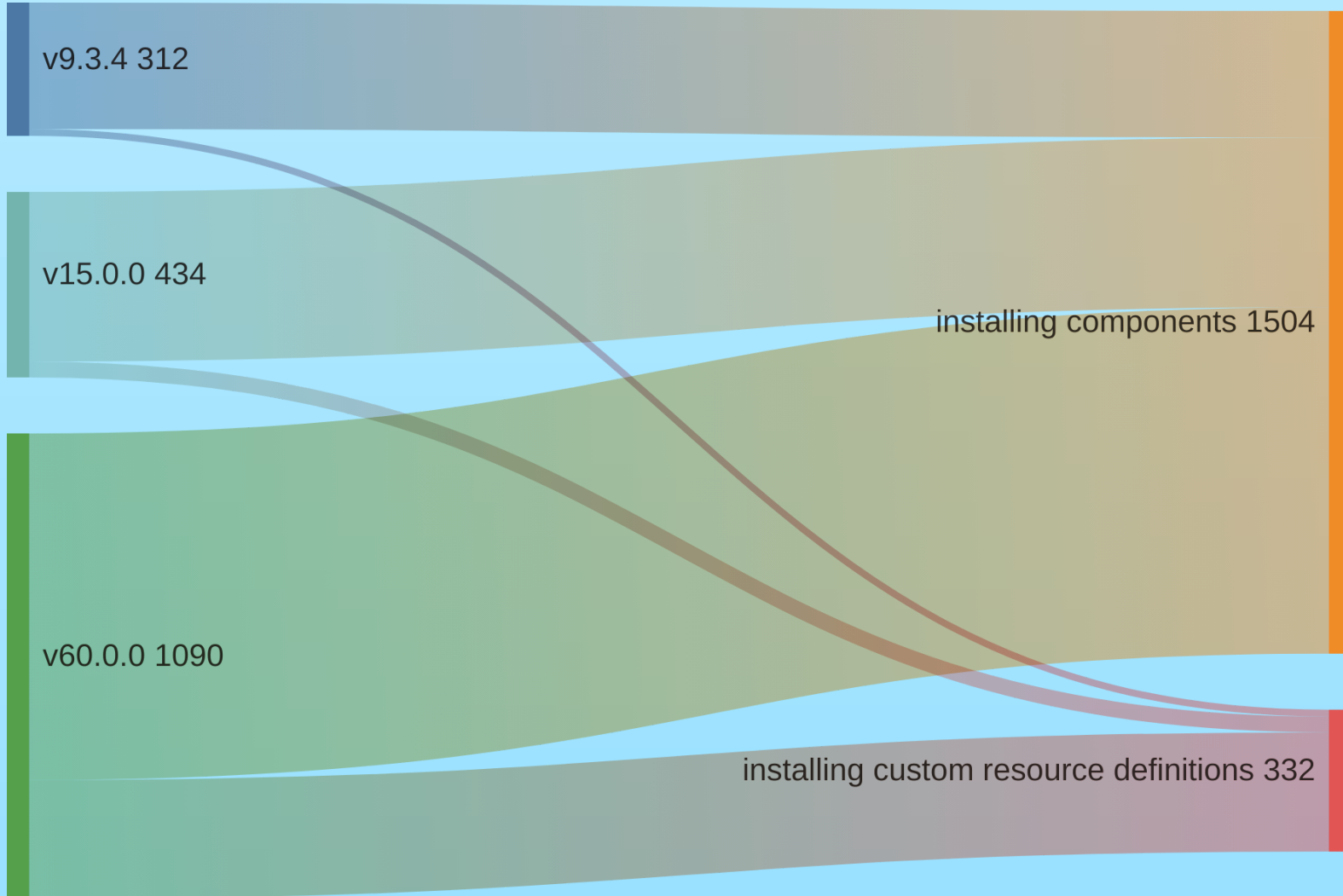
‘Nothing is simple about writing a CRD.’

Adam Jacob, [Kubernetes is an anti-platform](#), Ship It (18 October 2024)

# User experience

Let's take a look at the Kubernetes Prometheus Stack README.

```
1  ### From 64.x to 65.x
2
3  This version upgrades Prometheus-Operator to v0.77.1
4
5  Run these commands to update the CRDs before applying the upgrade.
6
7  kubectl apply ... -f https://.../alertmanagerconfigs.yaml
8  kubectl apply ... -f https://.../alertmanagers.yaml
9  kubectl apply ... -f https://.../podmonitors.yaml
10 kubectl apply ... -f https://.../probes.yaml
11 kubectl apply ... -f https://.../prometheusagents.yaml
12 kubectl apply ... -f https://.../prometheuses.yaml
13 kubectl apply ... -f https://.../prometheusrules.yaml
14 kubectl apply ... -f https://.../scrapeconfigs.yaml
15 ...
```



## Service targeting Prometheus v1.0.0

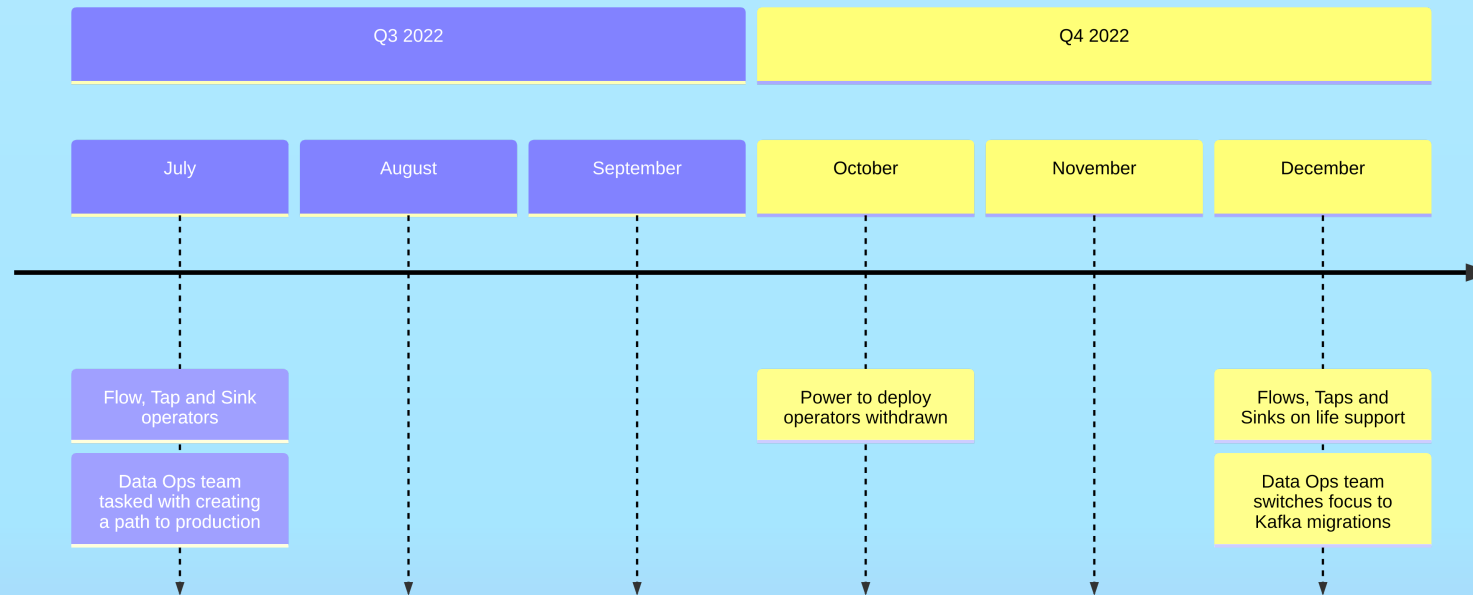
```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   annotations:
5     prometheus.io/port: "2112"
6     prometheus.io/scrape: "true"
```

## Service targeting current Prometheus

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    ...
5  ---
6  apiVersion: monitoring.coreos.com/v1
7  kind: ServiceMonitor
8  metadata:
9    namespace: monitoring
10 spec:
11   endpoints:
12     ...
```

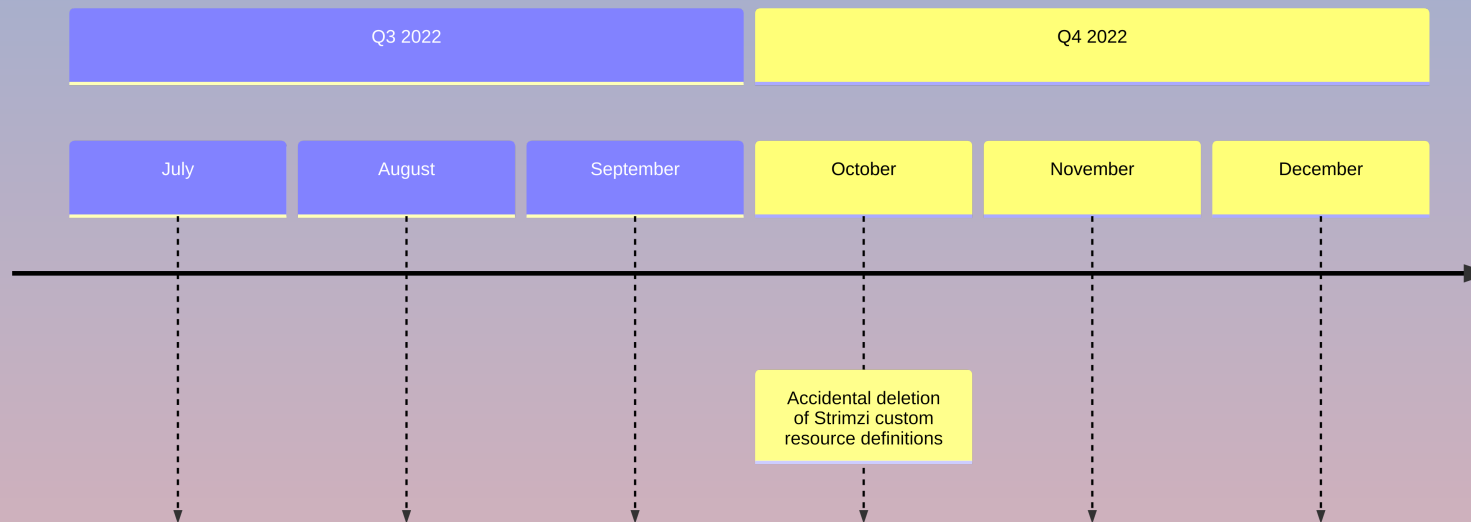
# Antipattern ① – developers building operators

e.g. Flow, Tap and Sink



## Antipattern ② – tight coupling with external resources

e.g. Strimzi





## Antipattern ③ – versioning is hard

e.g. AWS Controllers for Kubernetes

```
1 apiVersion: s3.services.k8s.aws/v1alpha1  
2 kind: Bucket
```

Is the old version still served? Have we provided a conversion webhook?

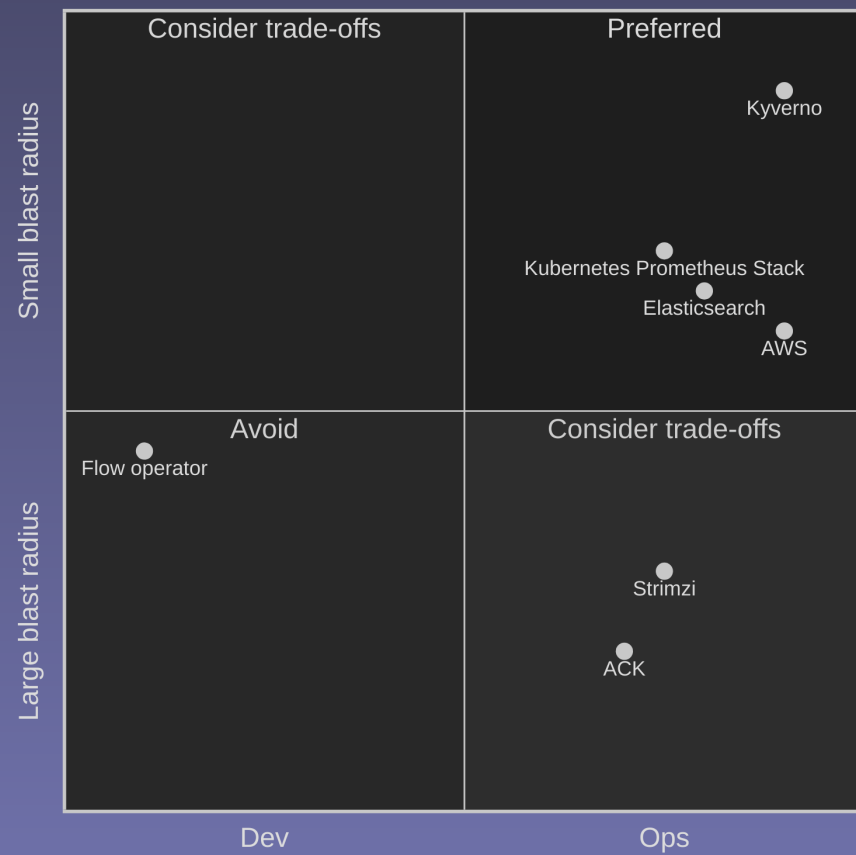
We are introducing new failure modes and edge cases.

## There are many outstanding operators

Kyverno for example stands out as an operator that has the look and feel of an in-tree policy engine.

Policy violations create detailed events and the new resources (Policy, ClusterPolicy) fit well into the existing set of resources.

```
$ kubectl get events --sort-by='{.lastTimestamp}'
TYPE      REASON          MESSAGE
Warning PolicyViolation policy require-ro-rootfs/validate-
              readOnlyRootFilesystem fail: validation
              error: Root filesystem must be read-only.
              rule validate-readOnlyRootFilesystem failed at
              path /spec/template/spec/containers/0/
              securityContext/readOnlyRootFilesystem/
```

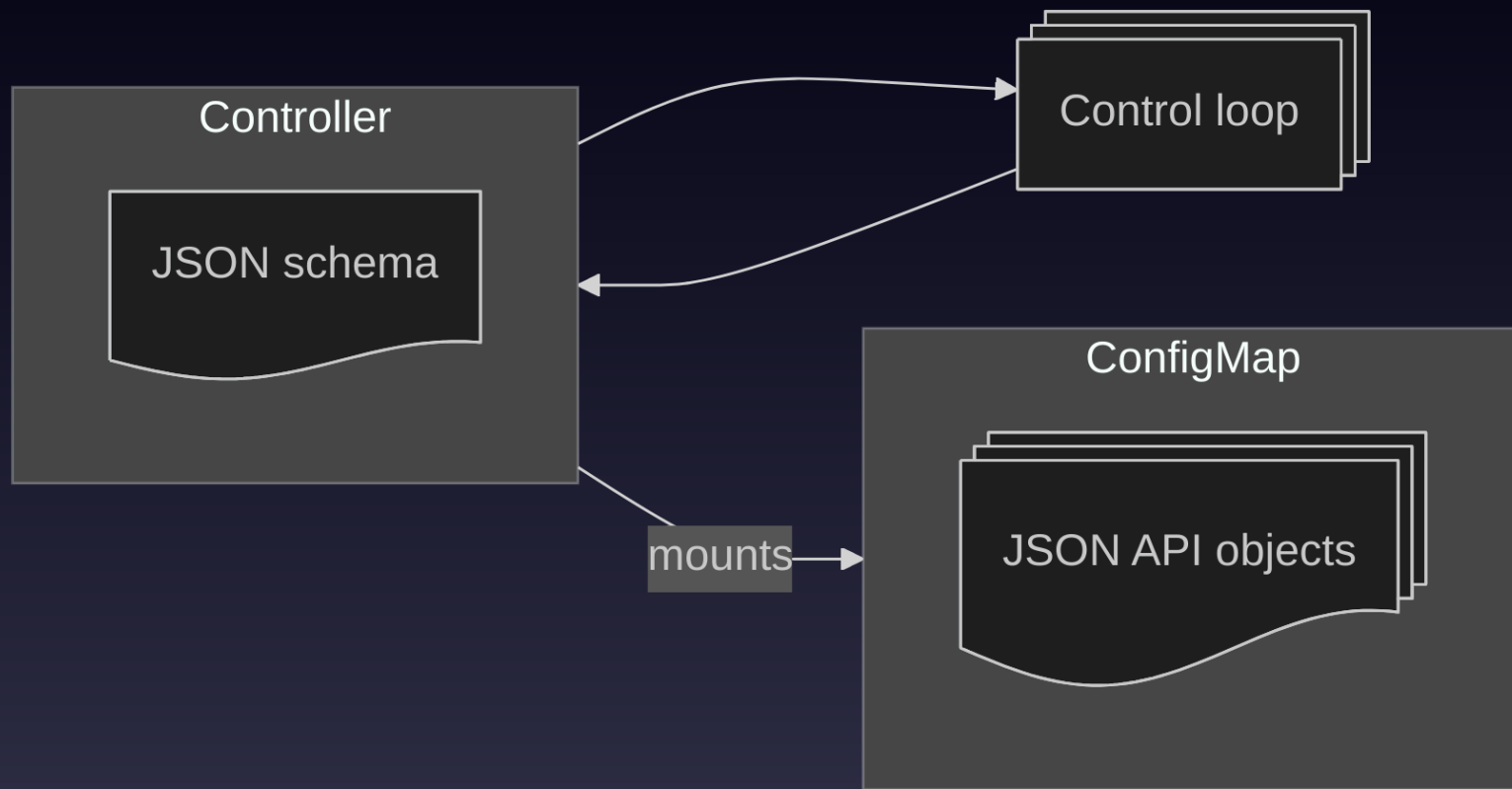


# A controller revival is overdue

Grafana loads ConfigMaps with label `grafana_dashboard` set to value `1`.

The only format required is Grafana's native JSON format.

```
1 for DASHBOARD in \
2   $(ls kube-prometheus-stack/dashboards/*.json)
3 do
4   CONFIGMAP=$(basename "${DASHBOARD}" | cut -d'.' -f1)
5   kubectl create configmap "${CONFIGMAP}" \
6     -n monitoring \
7     --dry-run=client \
8     --from-file="${DASHBOARD}" -o yaml | \
9     kubectl apply -f -
10  kubectl label configmap "${CONFIGMAP}" \
11    -n monitoring \
12    --overwrite grafana_dashboard="1"
13 done
```




# Thank you

 [gerald1248/operator-antipattern-slides](https://github.com/gerald1248/operator-antipattern-slides)

 [gerald1248@gmail.com](mailto:gerald1248@gmail.com)

 [www.linkedin.com/in/gerald1248](https://www.linkedin.com/in/gerald1248)

 [03spirit](https://twitter.com/03spirit)