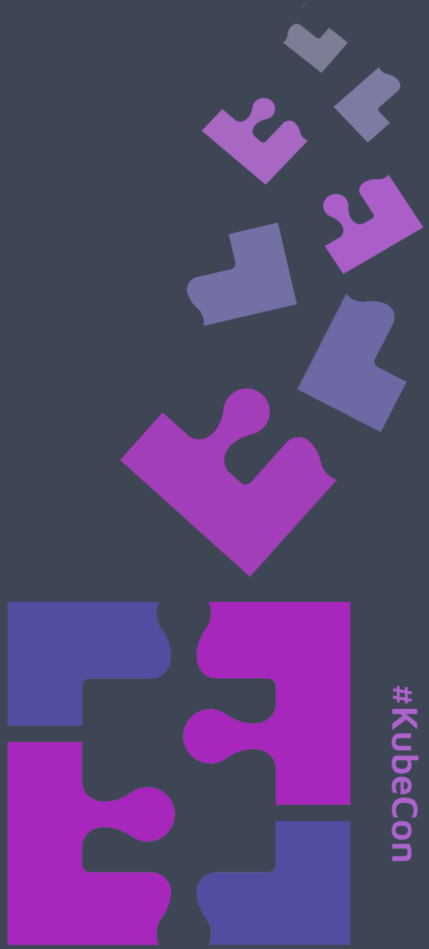
Tutorial

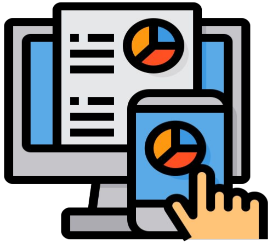
Becoming a Kubernetes Developer: Writing Your First Operator

Abby Bangser (she/her)
@a_bangser

SYNTASSO

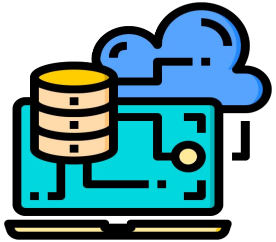


Common K8s interaction models



User

- May use `kubectl` or other GUI portals
- Focused on specific deployed applications



Administrator

- Infrastructure focused
- Likely creates and administers clusters
- Sets standards for clusters and users

Developer

- Likely some level of user and admin experience
- Using Kubernetes native solutions to address user and admin needs

So which am I?

And why am I here presenting on this topic?

- ➔ I have been a **user** during my time as a test engineer
- ➔ I have been an **administrator** during my time on platform engineering teams
- ➔ Today I am a **developer** as I build a tool to help platform teams manage their offerings as an extension to Kubernetes called Kratix

What will you achieve today?

- Scratch the surface of understanding when and why you may extend Kubernetes as a developer
- Over 1 hour of hands on learning time
- Build a CRUD complete operator
- Know where to return to continue this tutorial after this session

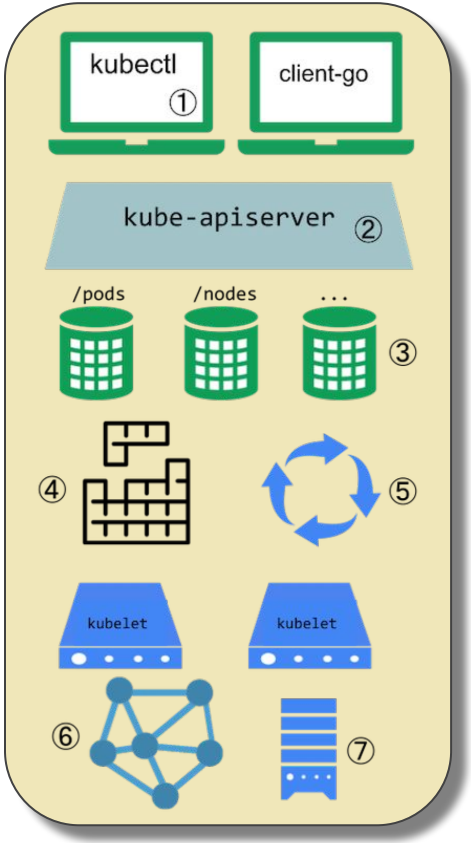
Developing for Kubernetes

1 Krew

3 crd

6 CSI

5

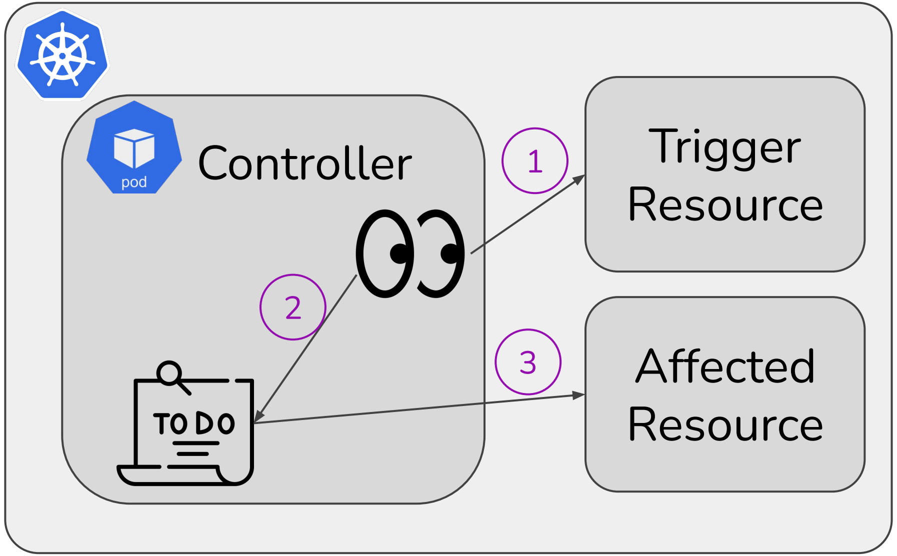


https://kubernetes.io/docs/concepts/extend-kubernetes/#extension-points

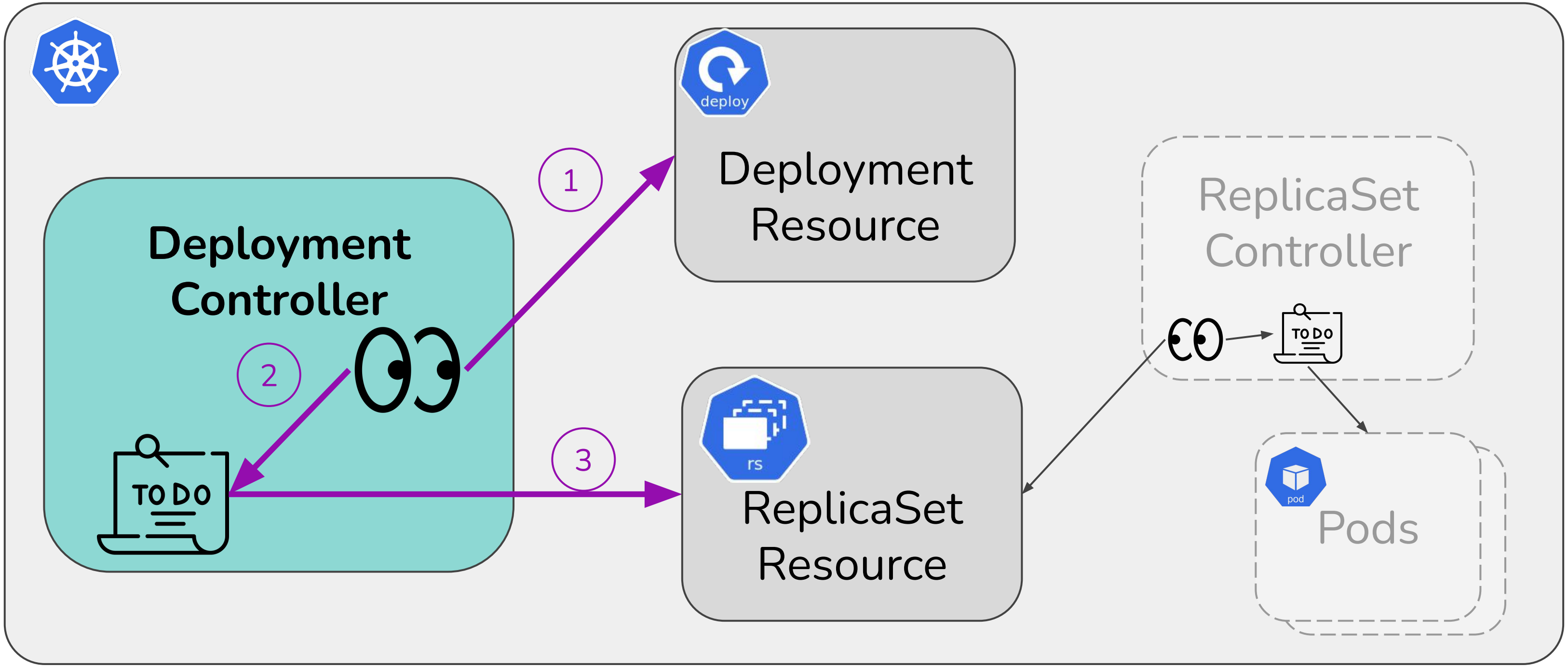
All operators start with a controller

Controllers are control loops that **watch the state of your cluster**, then make or request changes where needed.

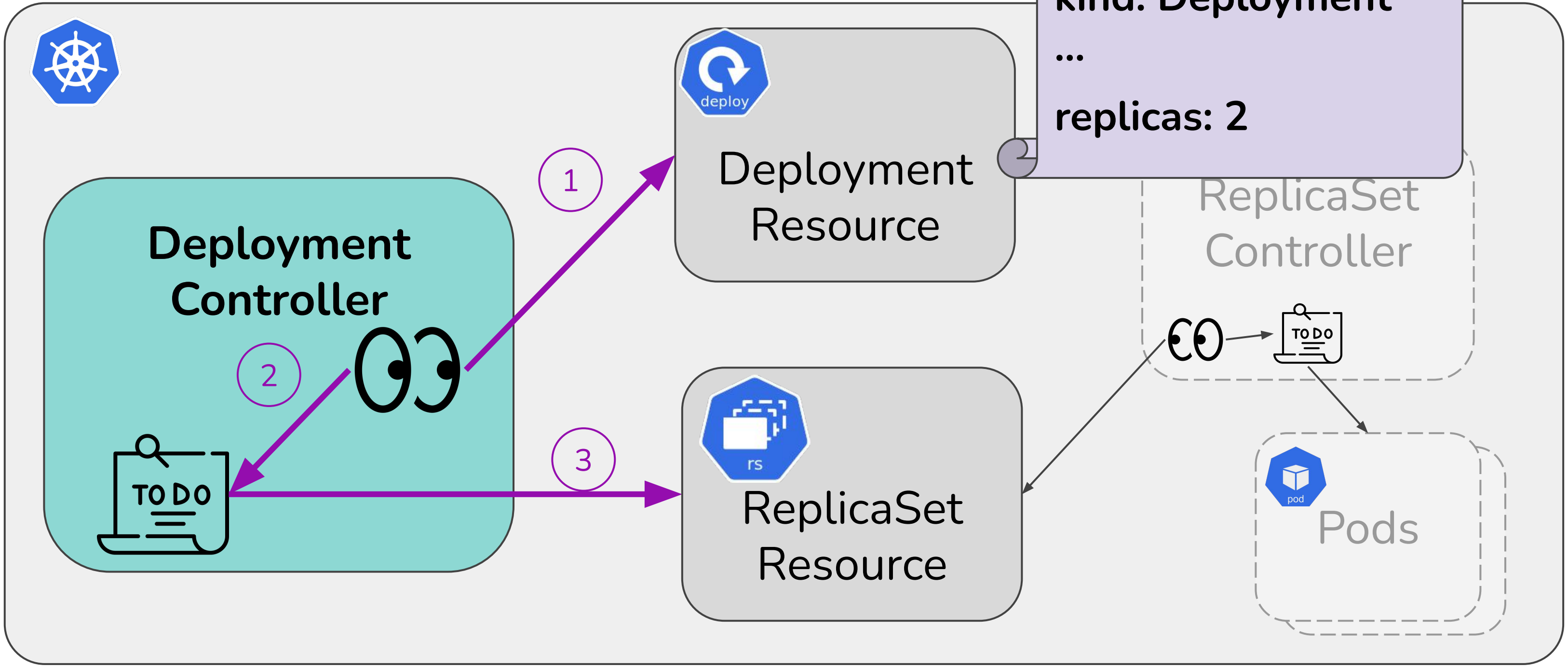
Each controller **tries to move the current cluster state closer to the desired state.**



`Deployment` is actually a controller!

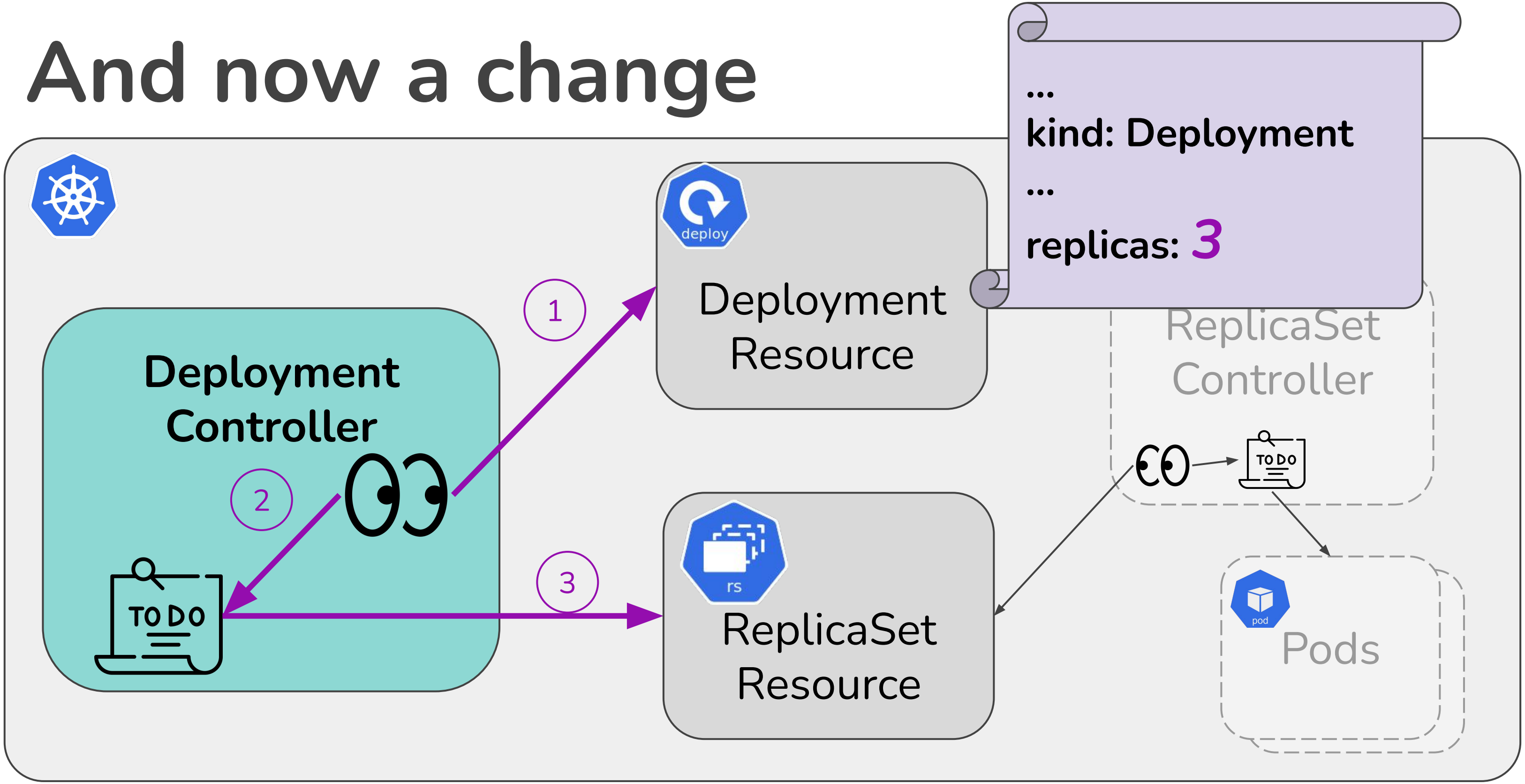


Example of current state



```
...
kind: Deployment
...
replicas: 2
```

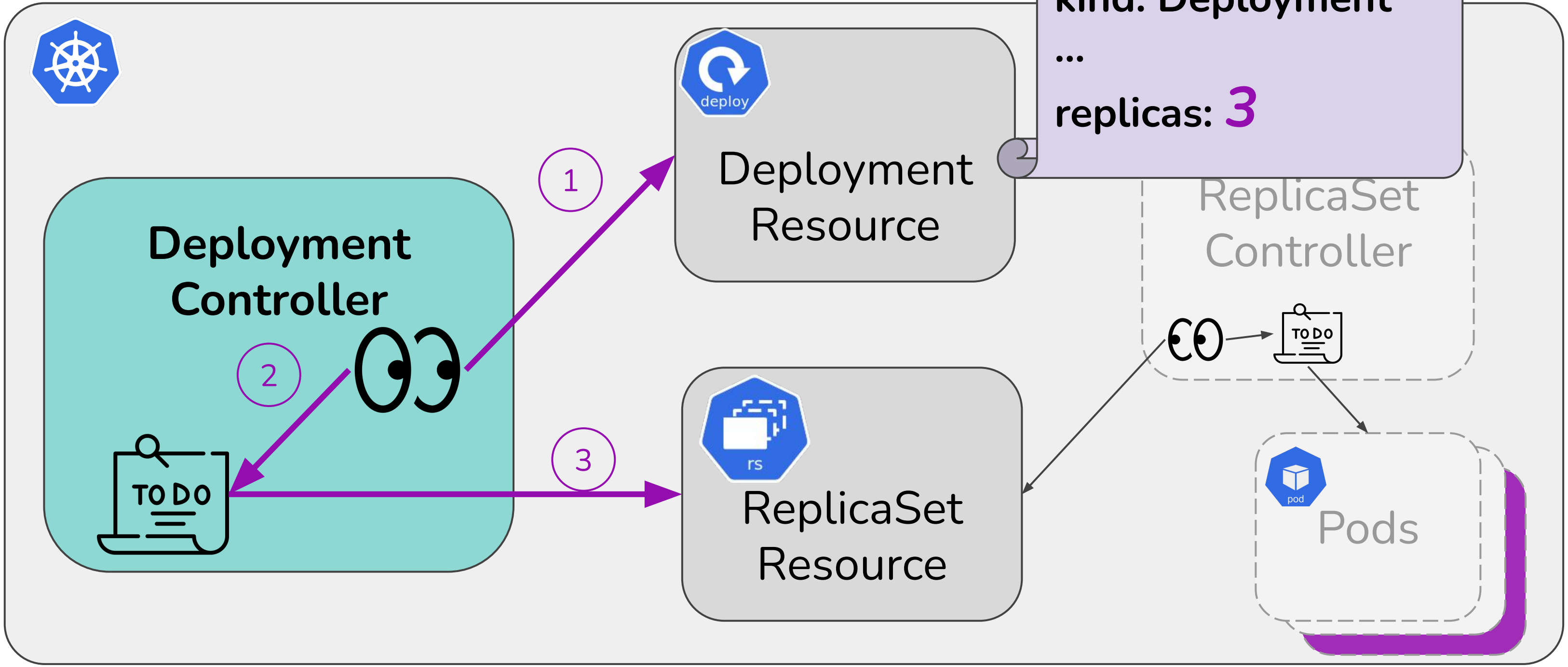
And now a change



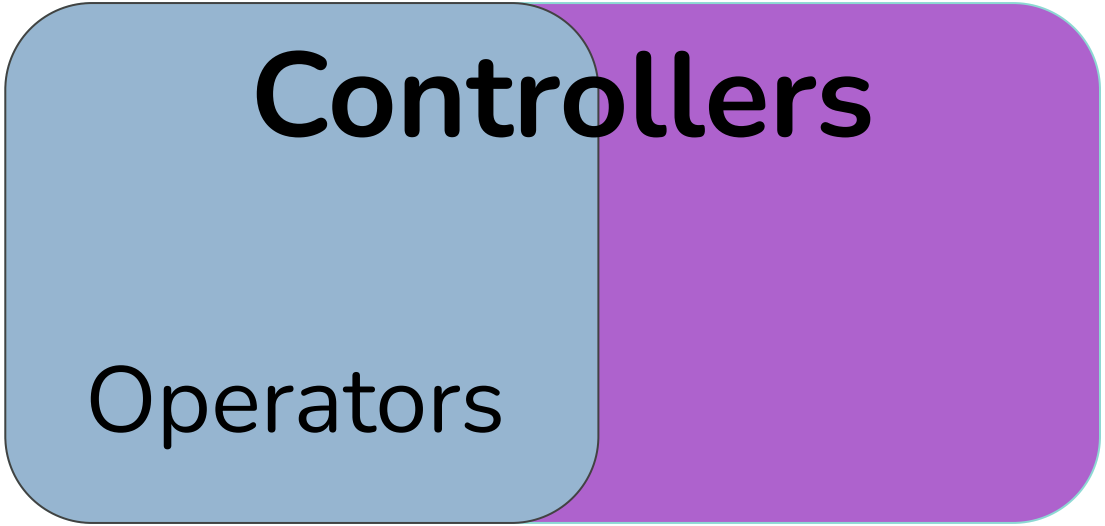
...
kind: Deployment
...
replicas: 3

So the controller goes to work

```
...
kind: Deployment
...
replicas: 3
```

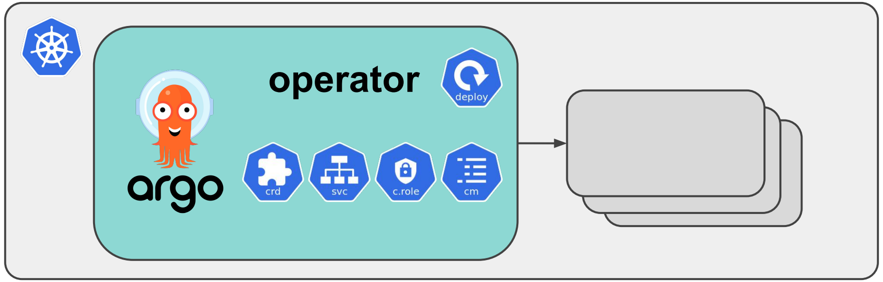


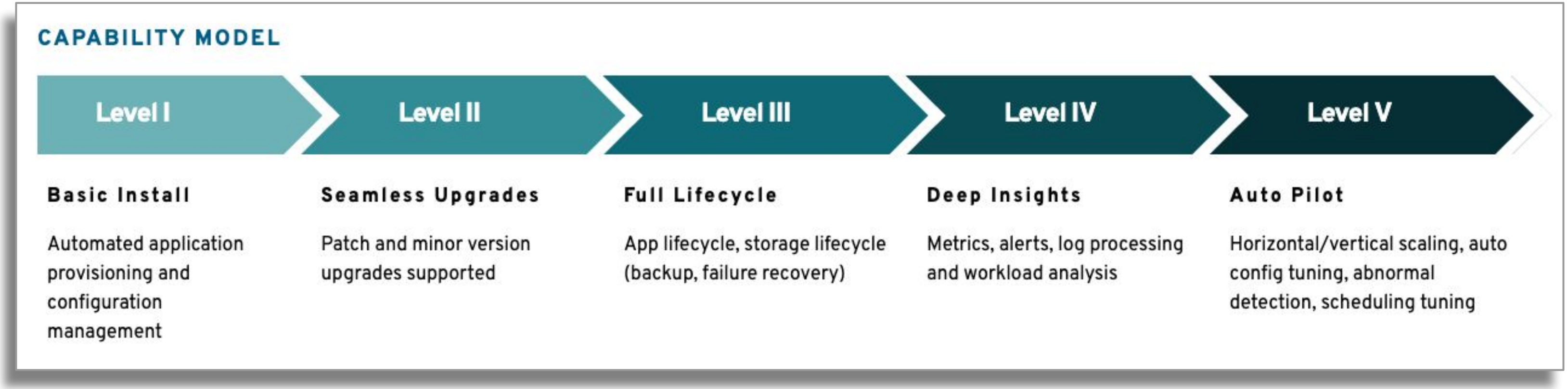
How controllers relate to operators



Then what is an operator?

An Operator represents human **operational knowledge in software** *(a controller!)*, to reliably manage an application.

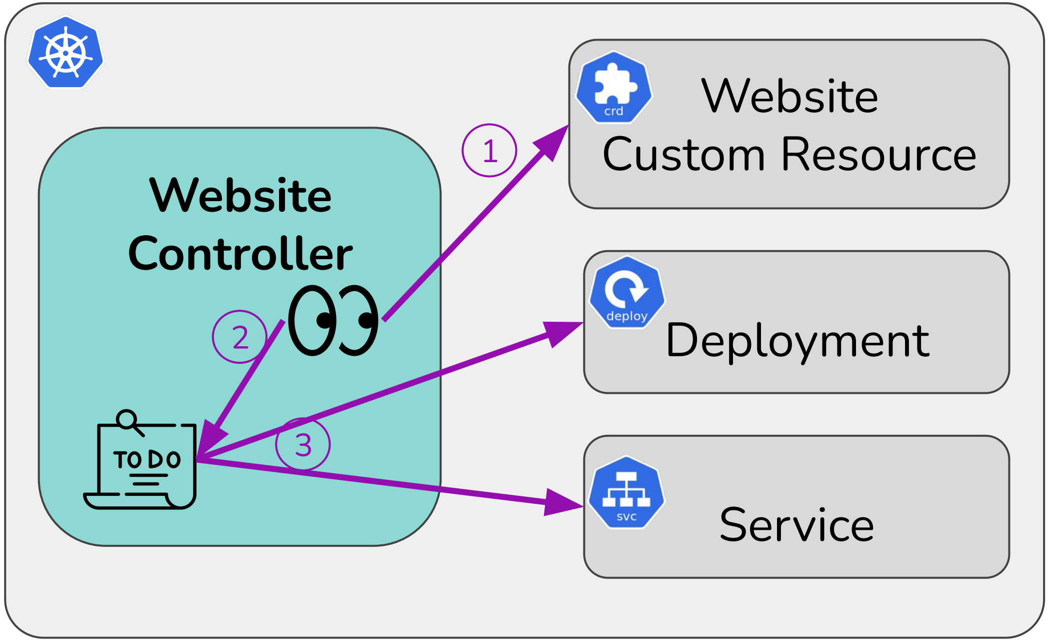


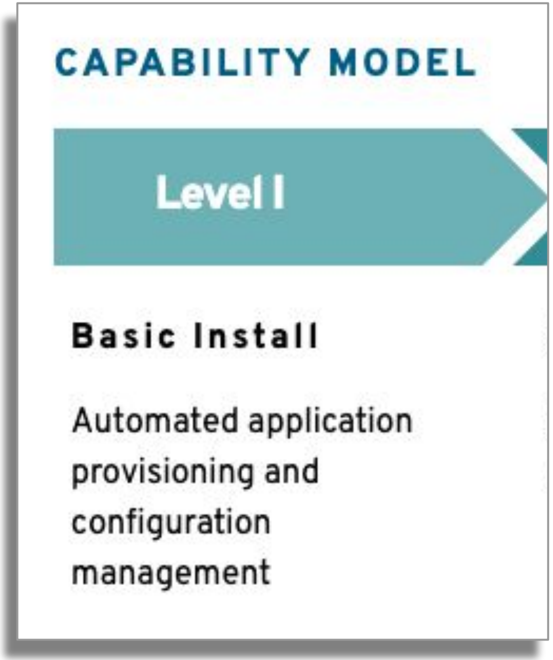


CAPABILITY MODEL

Level I	Level II	Level III	Level IV	Level V
Basic Install	**Seamless Upgrades**	**Full Lifecycle**	**Deep Insights**	**Auto Pilot**
Automated application provisioning and configuration management	Patch and minor version upgrades supported	App lifecycle, storage lifecycle (backup, failure recovery)	Metrics, alerts, log processing and workload analysis	Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

The operator you will be building





CAPABILITY MODEL

Level I

Basic Install

Automated application provisioning and configuration management

How to get started

instruqt

instruqt.com
@instruqt

It is time to get hands on!

https://abangser.gitbook.io/kubecon2022 *case sensitive!*

https://ahaslides.com/KUBECON

There will be some music put on.
Please ask for this to be stopped via Q&A (anonymous welcome) if this presents an accessibility concern for you.

Wrapping up

Learn more about the platform at SU7

Each session capped at 2 hours

Skip to

You can skip to where you last left off

The website is a permanent link!

https://abangser.gitbook.io/kubecon2022 *case sensitive!

Thank you!

https://sched.co/182F1

https://abangser.gitbook.io/kubecon2022

SYNTASSO

#KubeCon

@a_bangser

Resources

- ➔ More ideas on how to extend kubernetes:
  https://kubernetes.io/docs/concepts/extend-kubernetes/
- ➔ Original operator announcement:
  https://web.archive.org/web/20210210032403/https://coreos.com/blog/introducing-operators.html
- ➔ Blog suggesting when not to use operators:
  https://rm-rf.ca/posts/2020/when-not-to-write-kubernetes-operator/
- ➔ An example controller using Kubebuilder:
  http://kratix.io/
- ➔ An example operator following Kubebuilder patterns:
  https://github.com/argoproj-labs/argocd-operator

Icons sourced from ***FlatIcon.com***

Artists:

➔ Itim2101

➔ Freepik

➔ Phatplus

➔ Royyan Wijaya

➔ SBTS2018

➔ Unicornlabs