



EDA and the THREE DWARVES

written by **MÁIRÍN DUFFY**

illustrated by **MADELINE PECK**



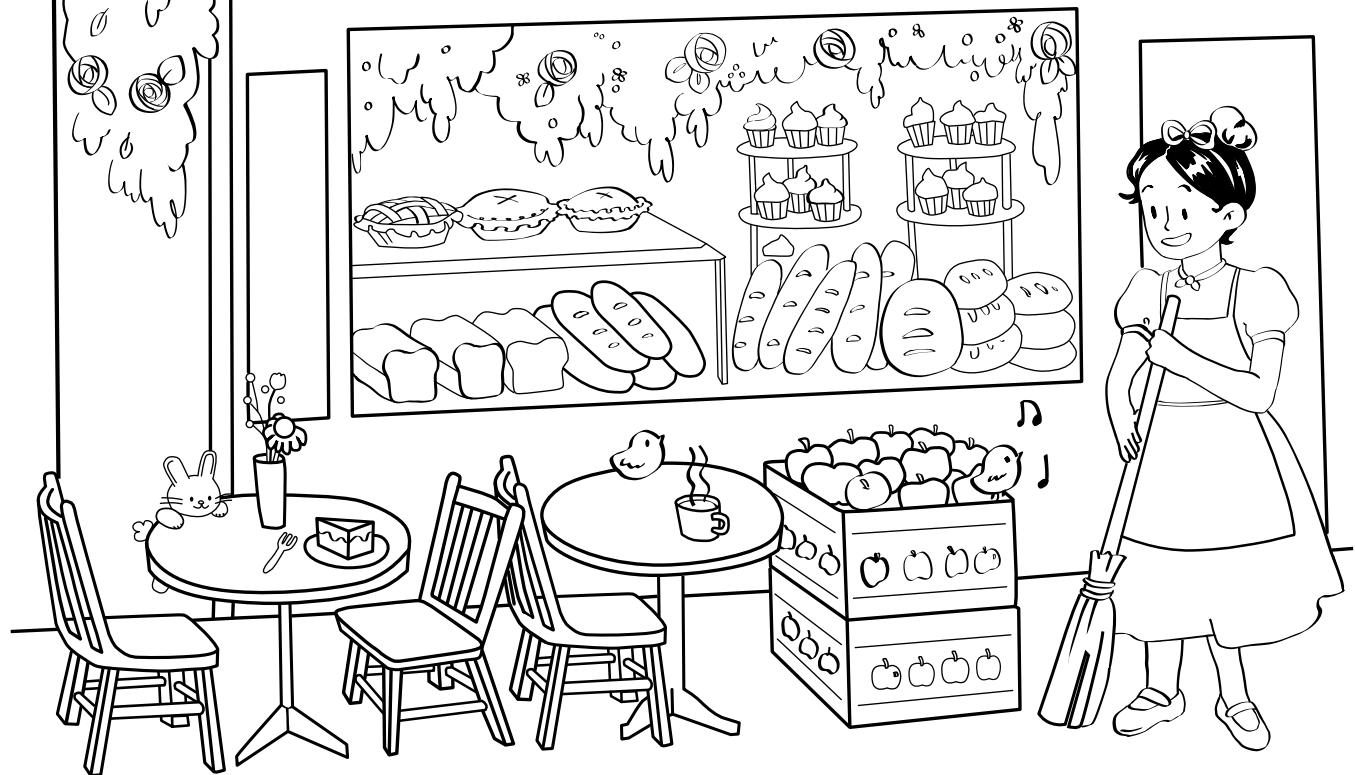
**LEARN
as you
COLOR!**

Once Upon a time...

Cute Cawaii

There was a baker named Snow White. She ran a bakery in the Kubernetes Kingdom.
Her specialty? Apple pies.

白雪姫のパン屋さん



The three dwarves TEKTON, and the KNATIVE brothers SERVE and EVENT, help her keep her BAKERY running.

TEKTON is a builder dwarf



When Snow White develops a new application for her BAKERY- say, a cupcake decoration application-

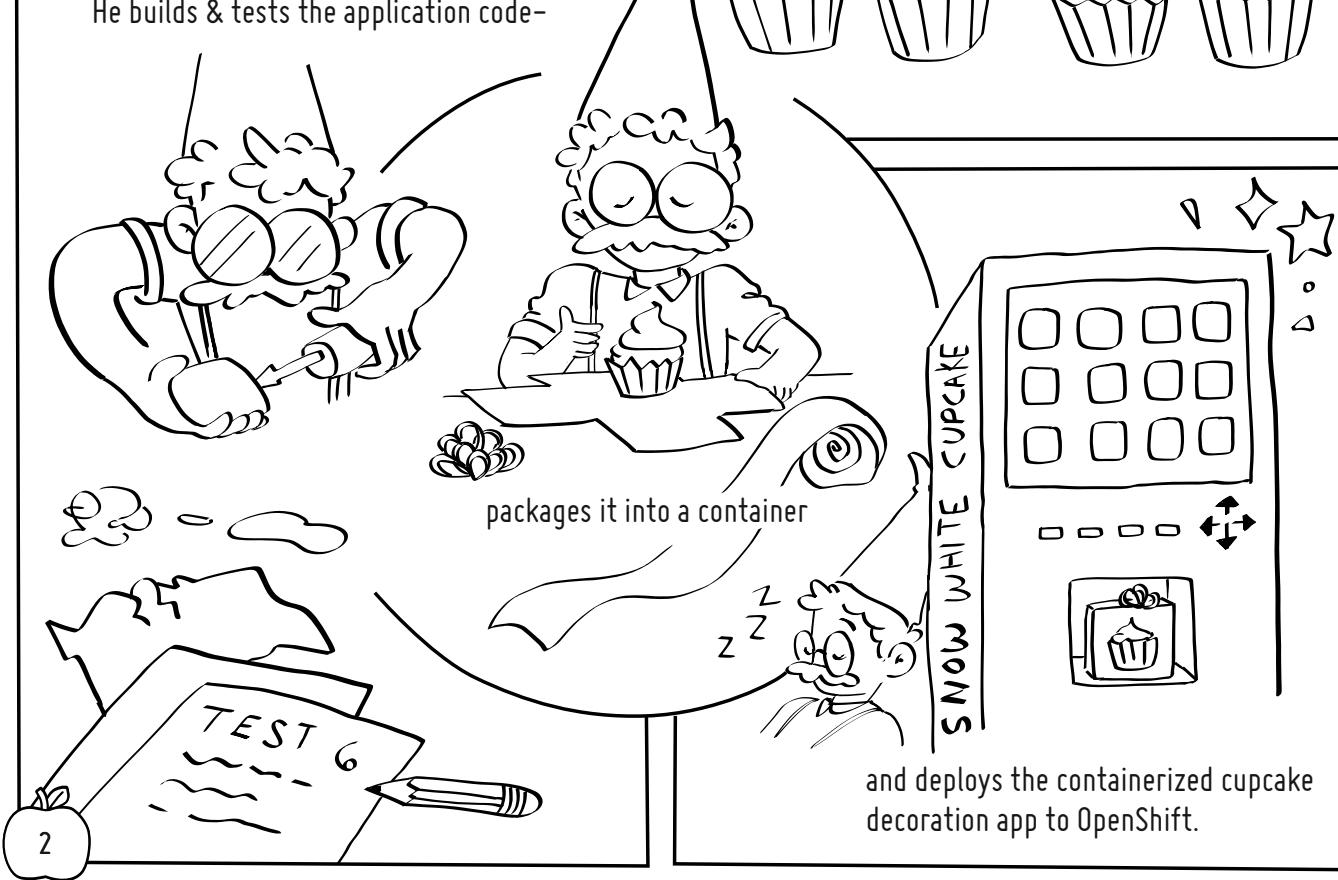


TEKTON takes her app code through a software development pipeline.

He builds & tests the application code-



packages it into a container



and deploys the containerized cupcake decoration app to OpenShift.

SERVE KNATIVE is a server dwarf



TEKTON builds application containers, and SERVE serves them up in a serverless fashion in Kubernetes.



If SNOW WHITE is no longer using her cupcake decoration app because there are no open orders -



SERVE KNATIVE can stop the cupcake app deployment so it is no longer running.



As another cupcake order comes through -



SERVE can deploy the cupcake app again just in time.

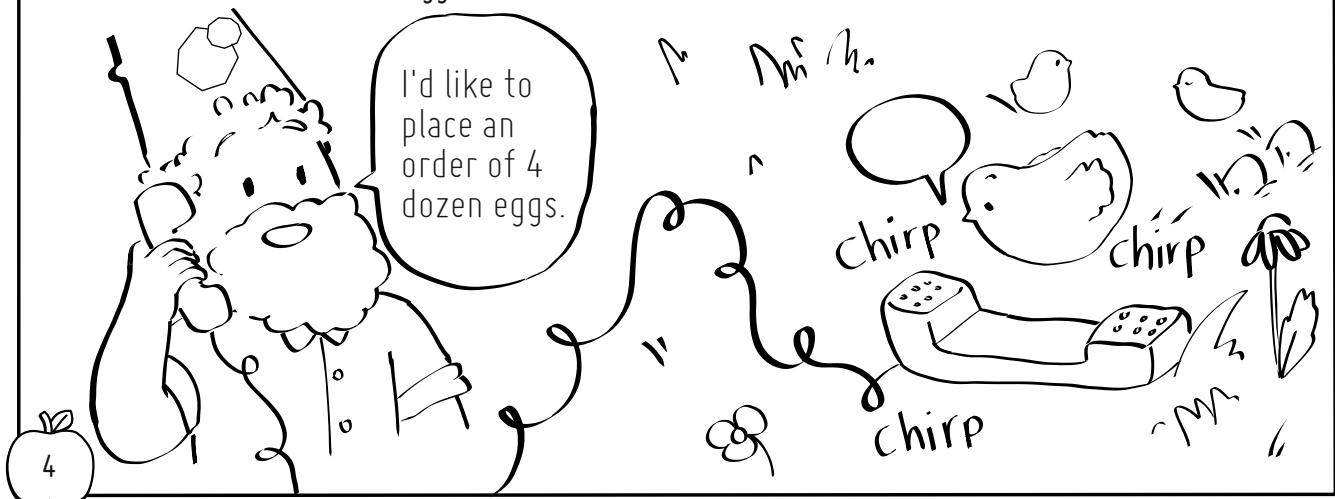
EVENT KNATIVE is an eventing dwarf



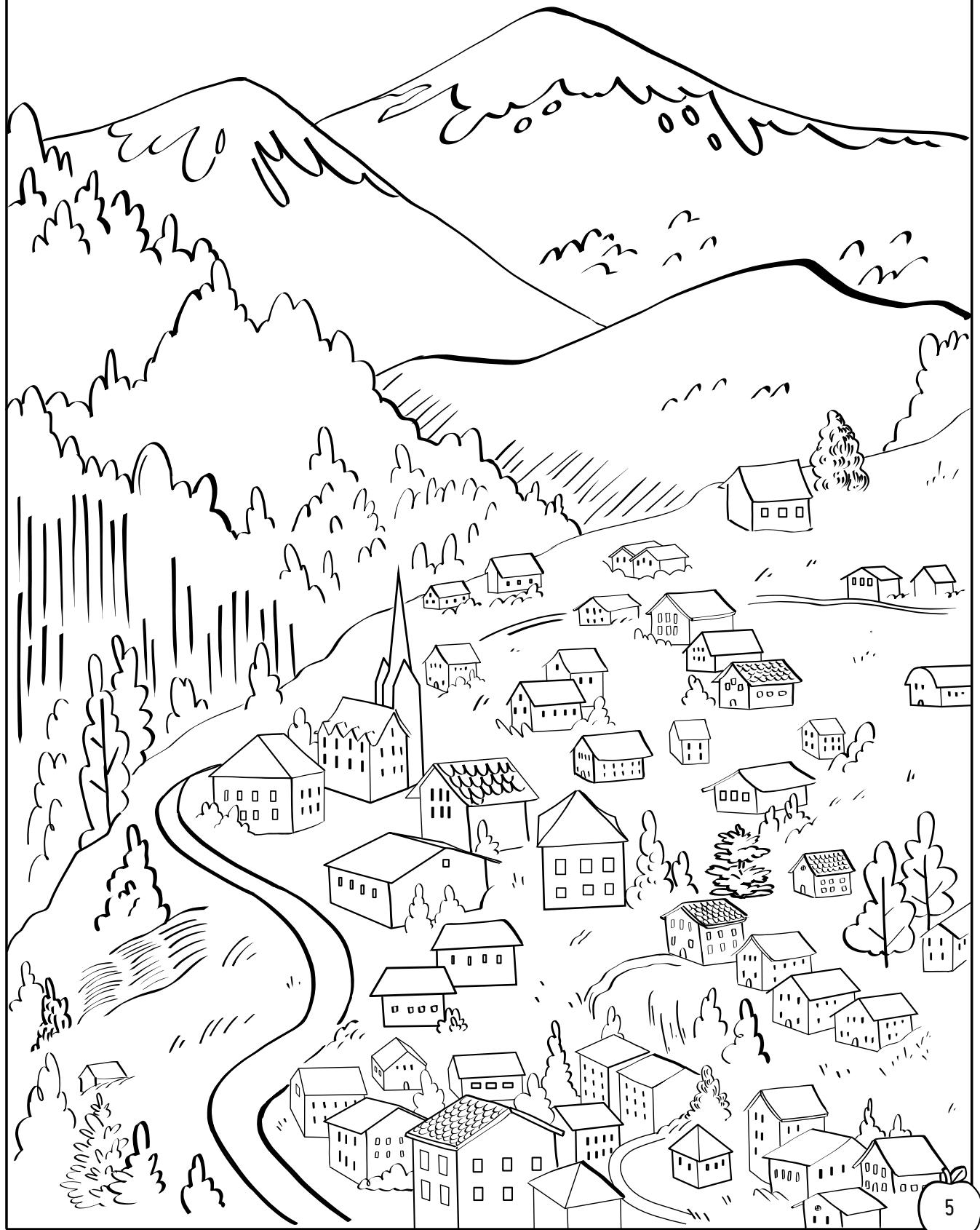
For example, as SNOW WHITE uses eggs up in her bakery, it may produce egg usage events



When it consumes enough egg usage events above a certain threshold, EVENT KNATIVE may place an order with CHICKEN LITTLE for more eggs.



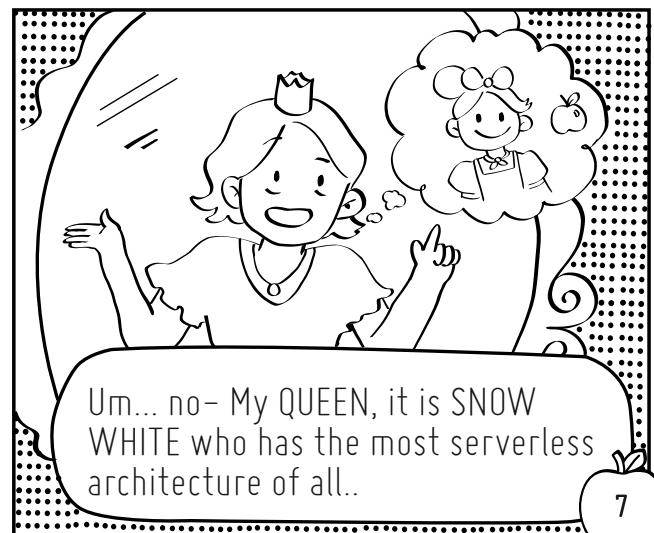
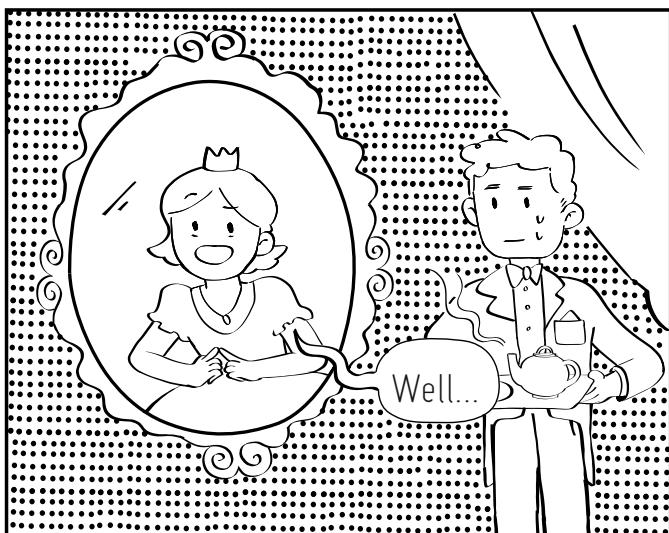
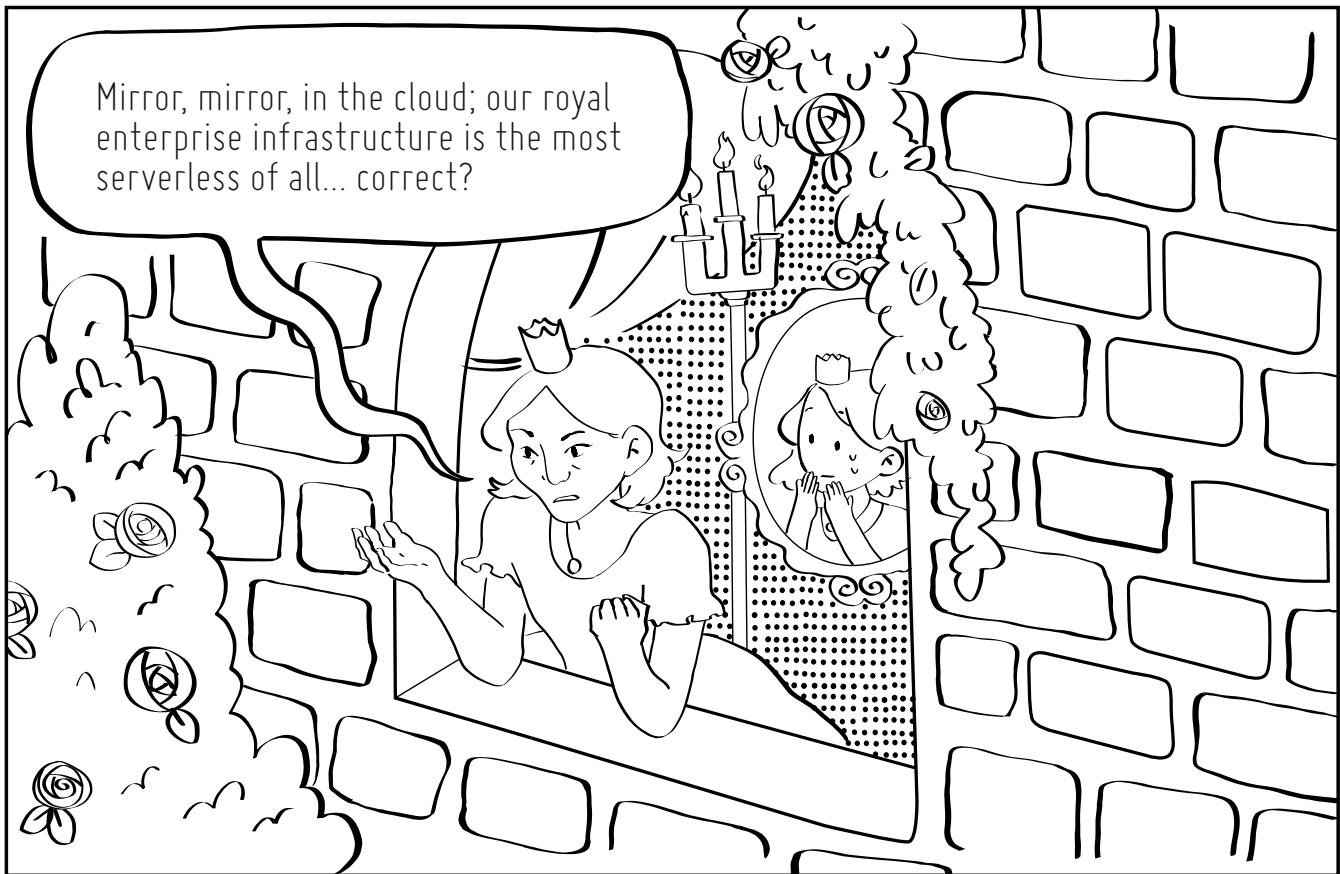
Meanwhile, in the lair of the EVIL QUEEN MALICIOUS,
on a mountain top overlooking the Kubernetes Kingdom...





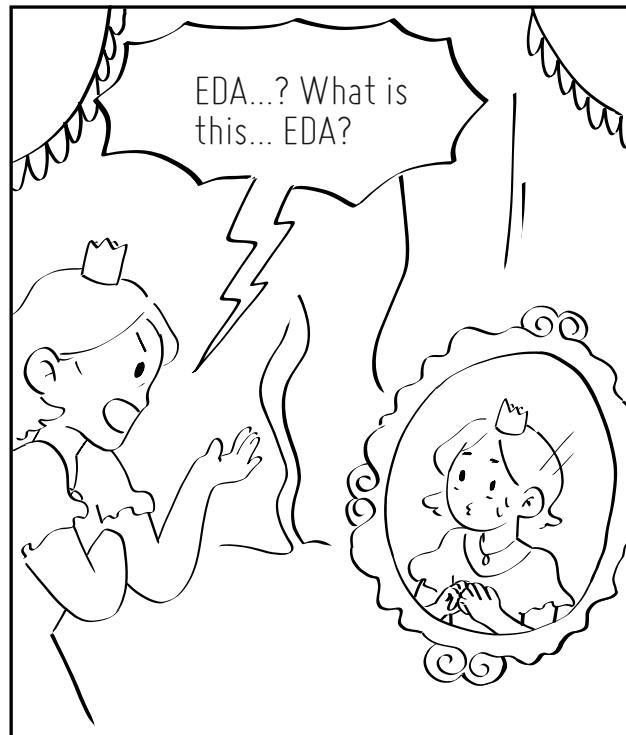


Serverless doesn't mean there are no servers, MALICIOUS. It means applications don't run 24/7—they only run if they are needed and stop when they are not. It results in a more efficient allocation of resources—and the cakes and pies are always fresh.





How is this SNOW WHITE able to have
the most serverless architecture of all?



EDA- Event Driven Architecture. It's an application model based around real-world events rather than manual or periodic requests. Her bakery's architecture can scale, stop, & deploy applications based on business-related events.



For example, this past Autumn when customers were scared of her spooky pumpkin pies- she was able to roll back on production.

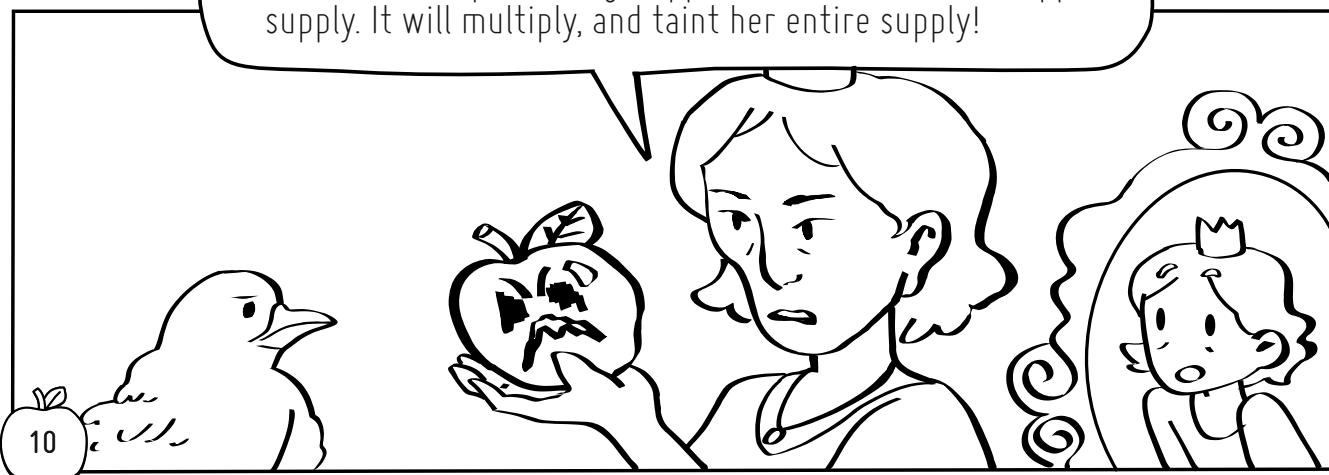
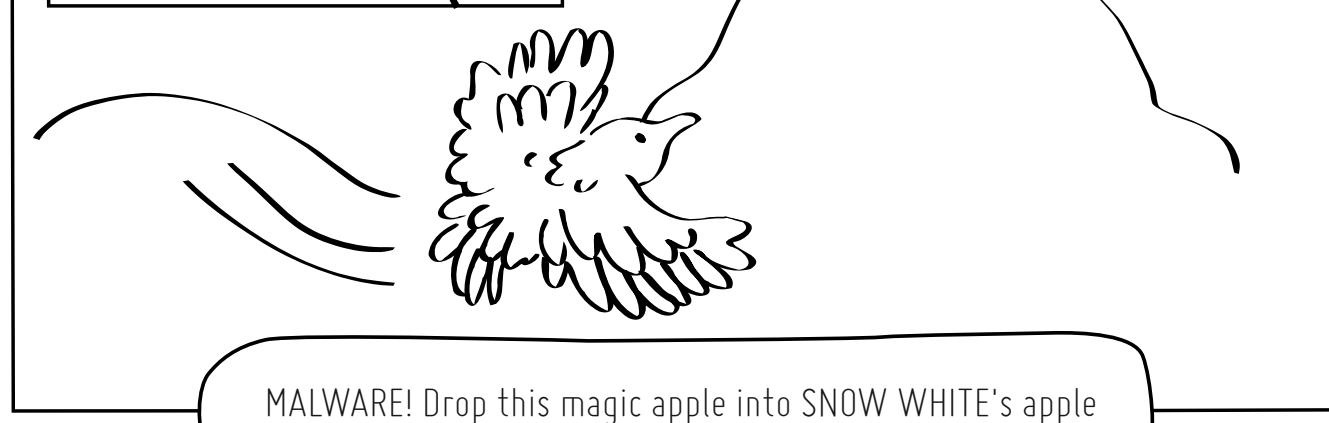
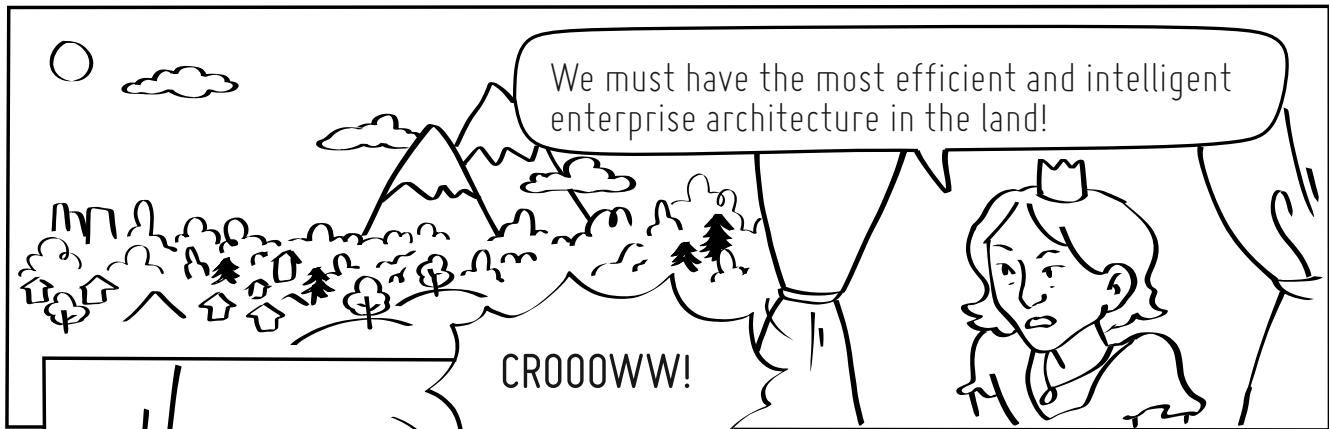


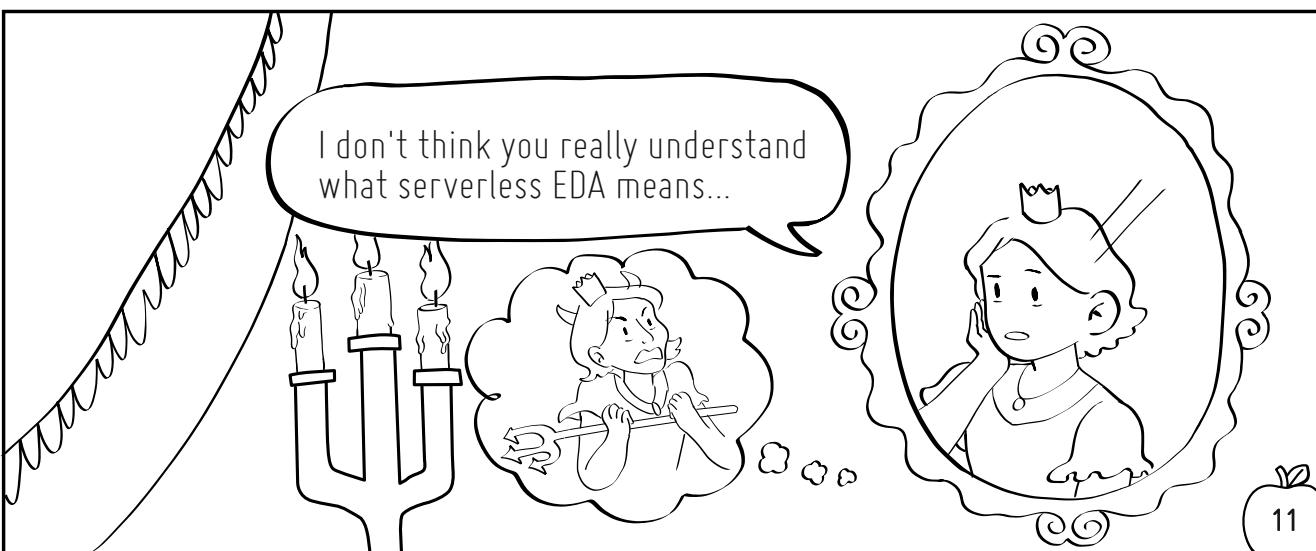
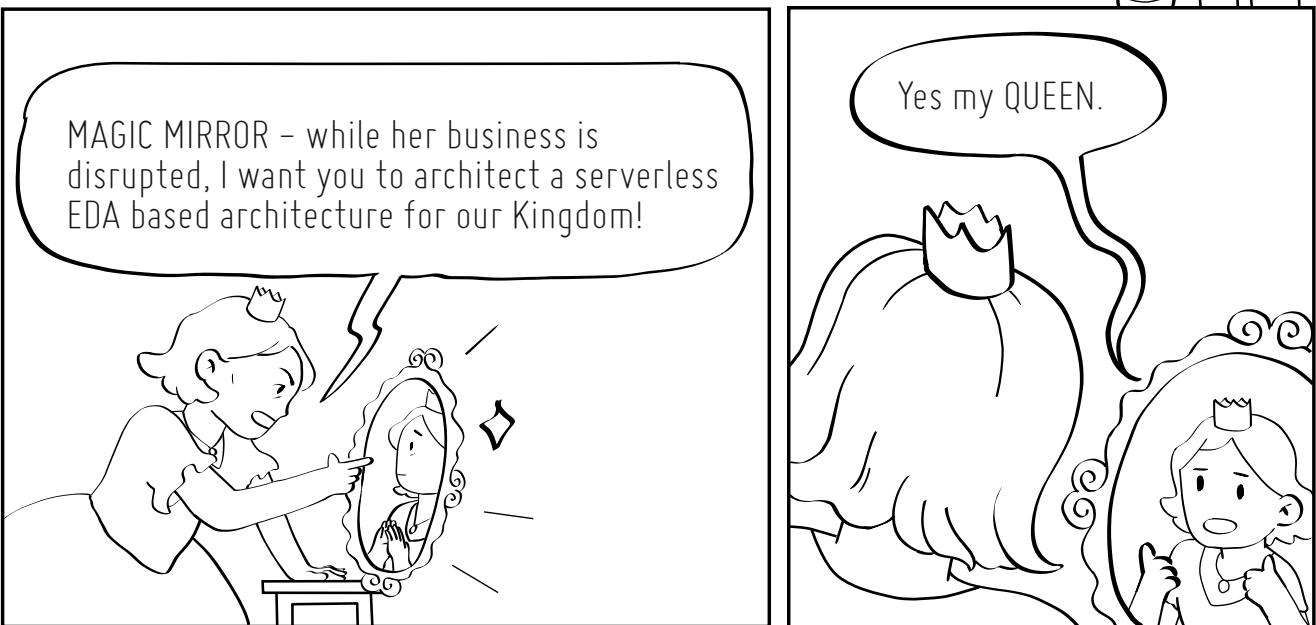
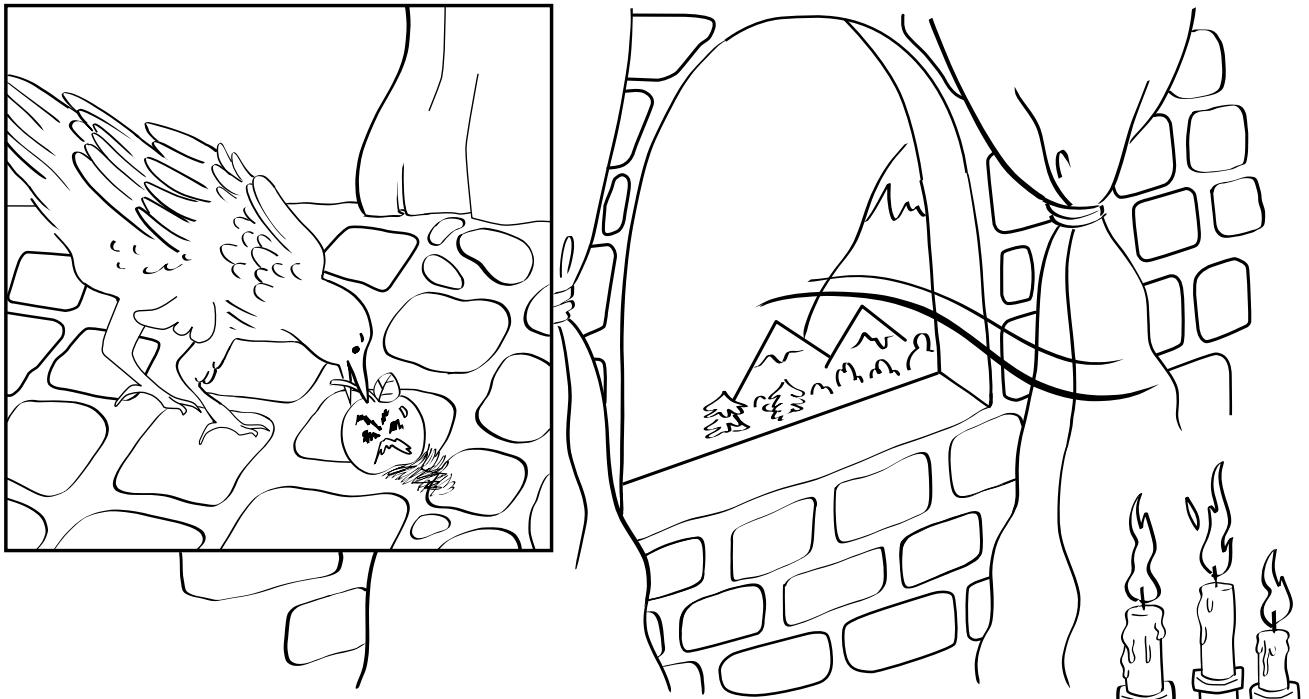
When the winter holidays rolled in a few months later, the production for cranberry pie & regular pumpkin pie was scaled up.

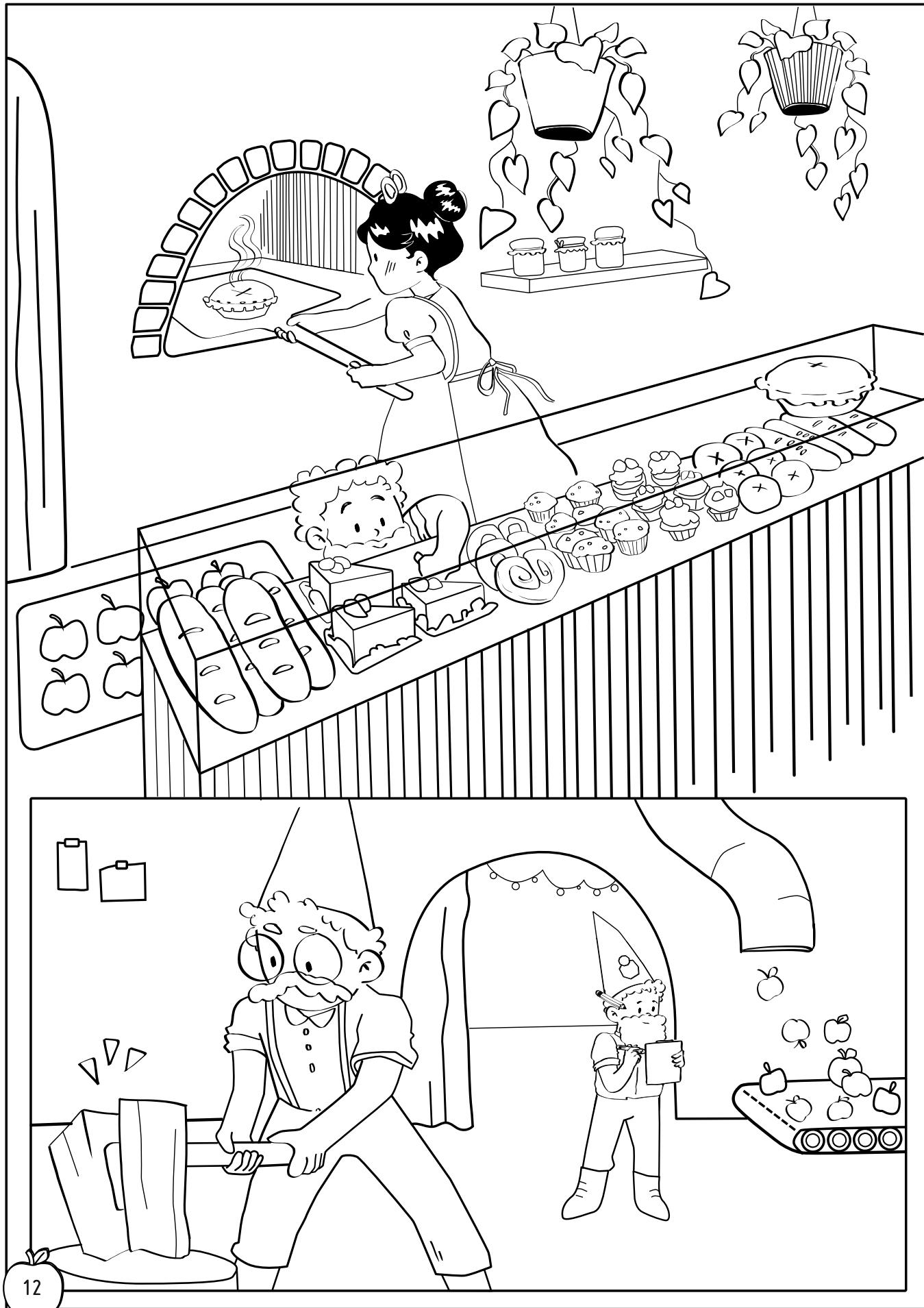


After the holidays, everyone is so full of pies and leftovers that they scale down to zero pies for as long as they need.

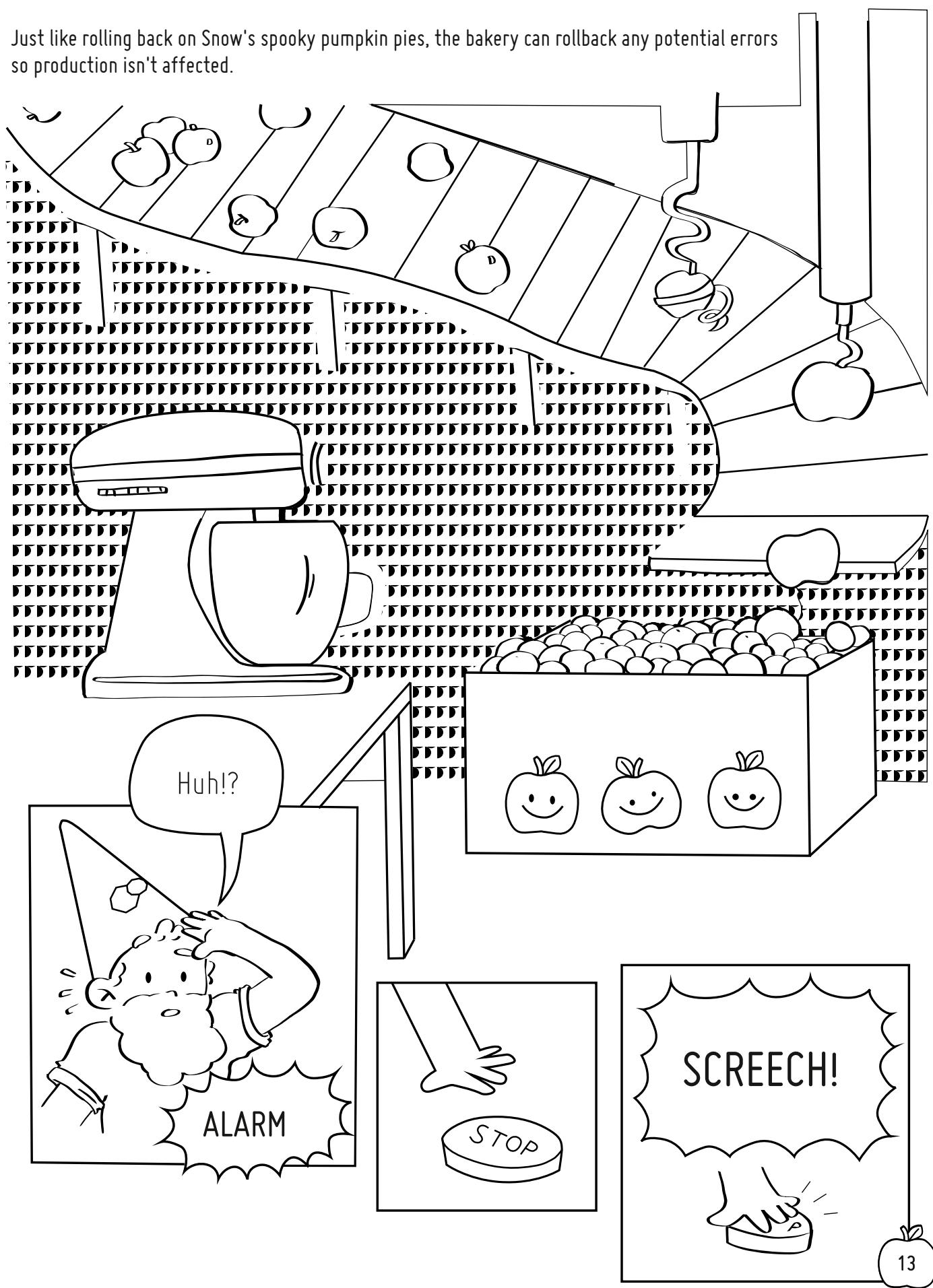


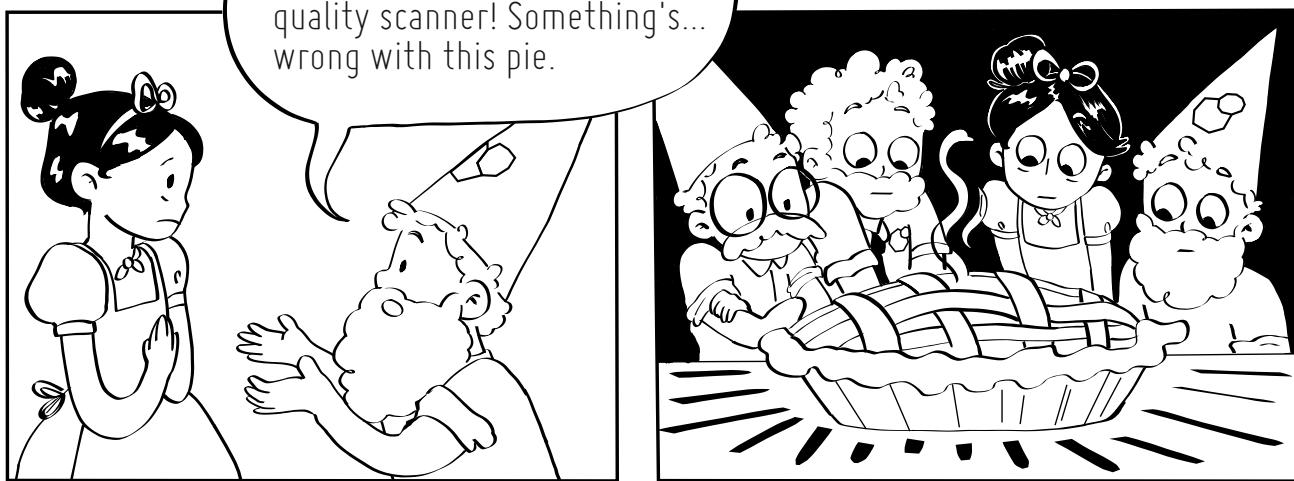






Just like rolling back on Snow's spooky pumpkin pies, the bakery can rollback any potential errors so production isn't affected.





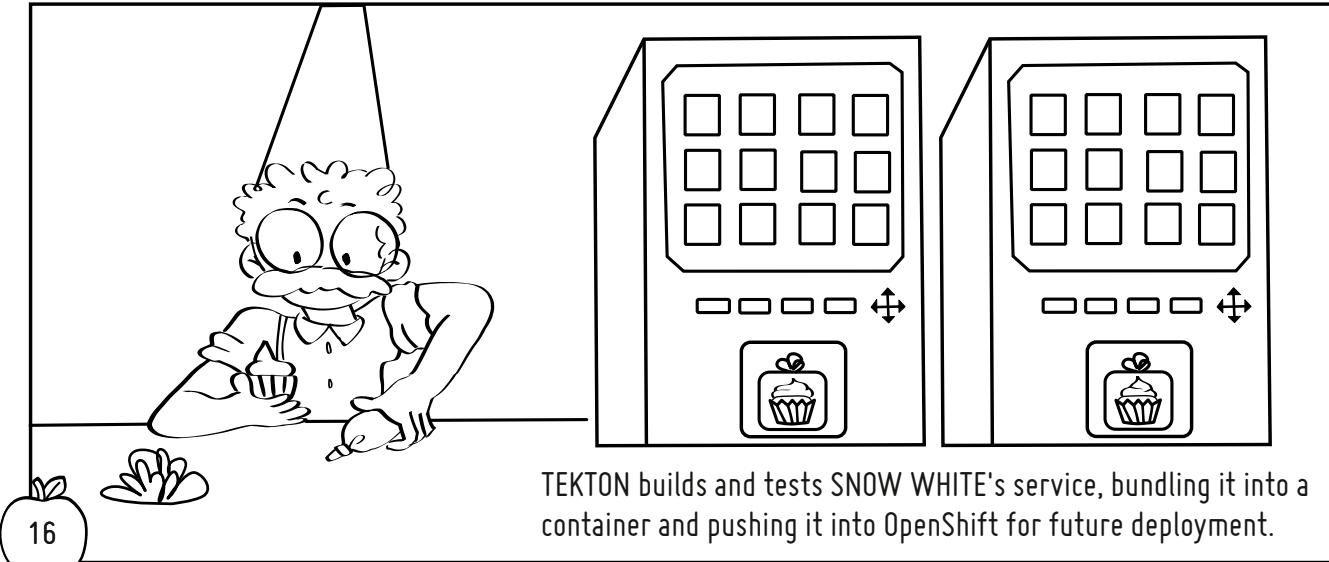


Is this... a poisoned apple in our filling?

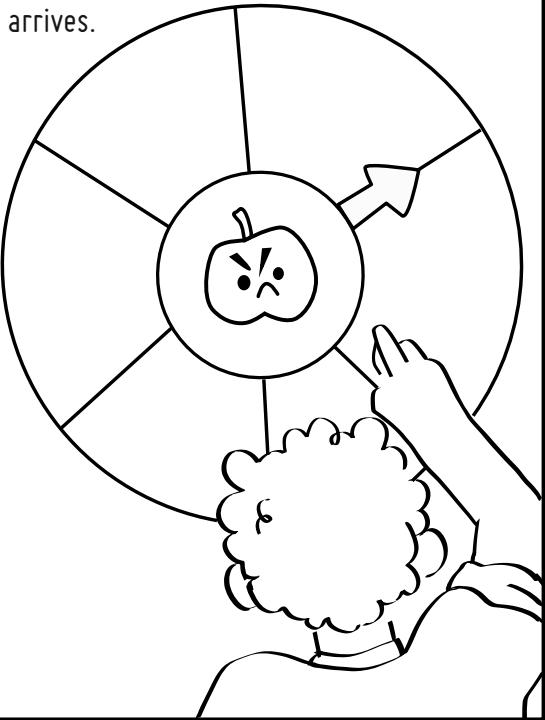
It IS!

Blegh

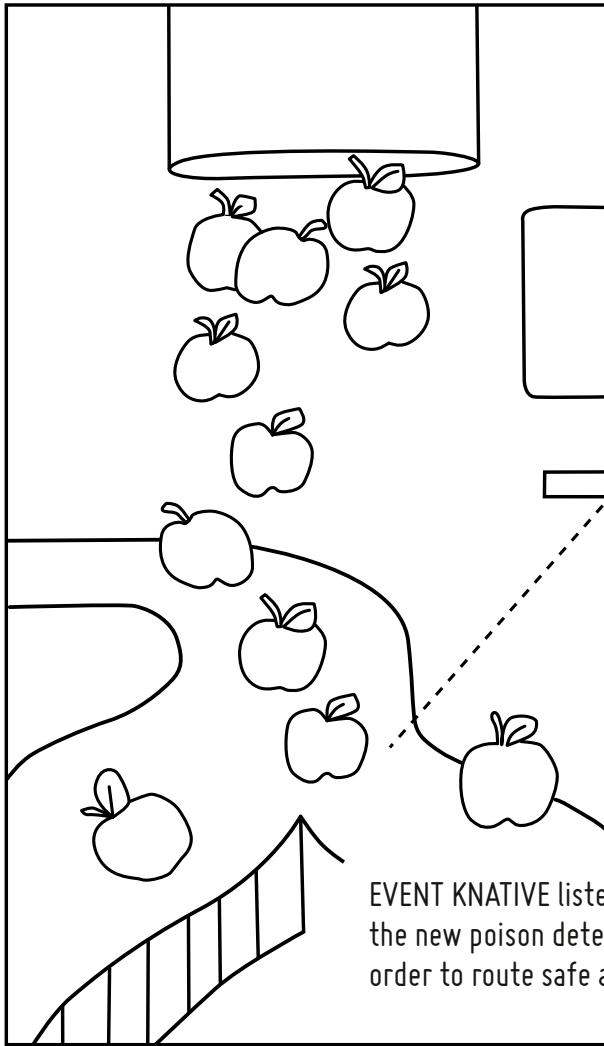




SERVE increases the number of poison detection services as a new apple delivery arrives.



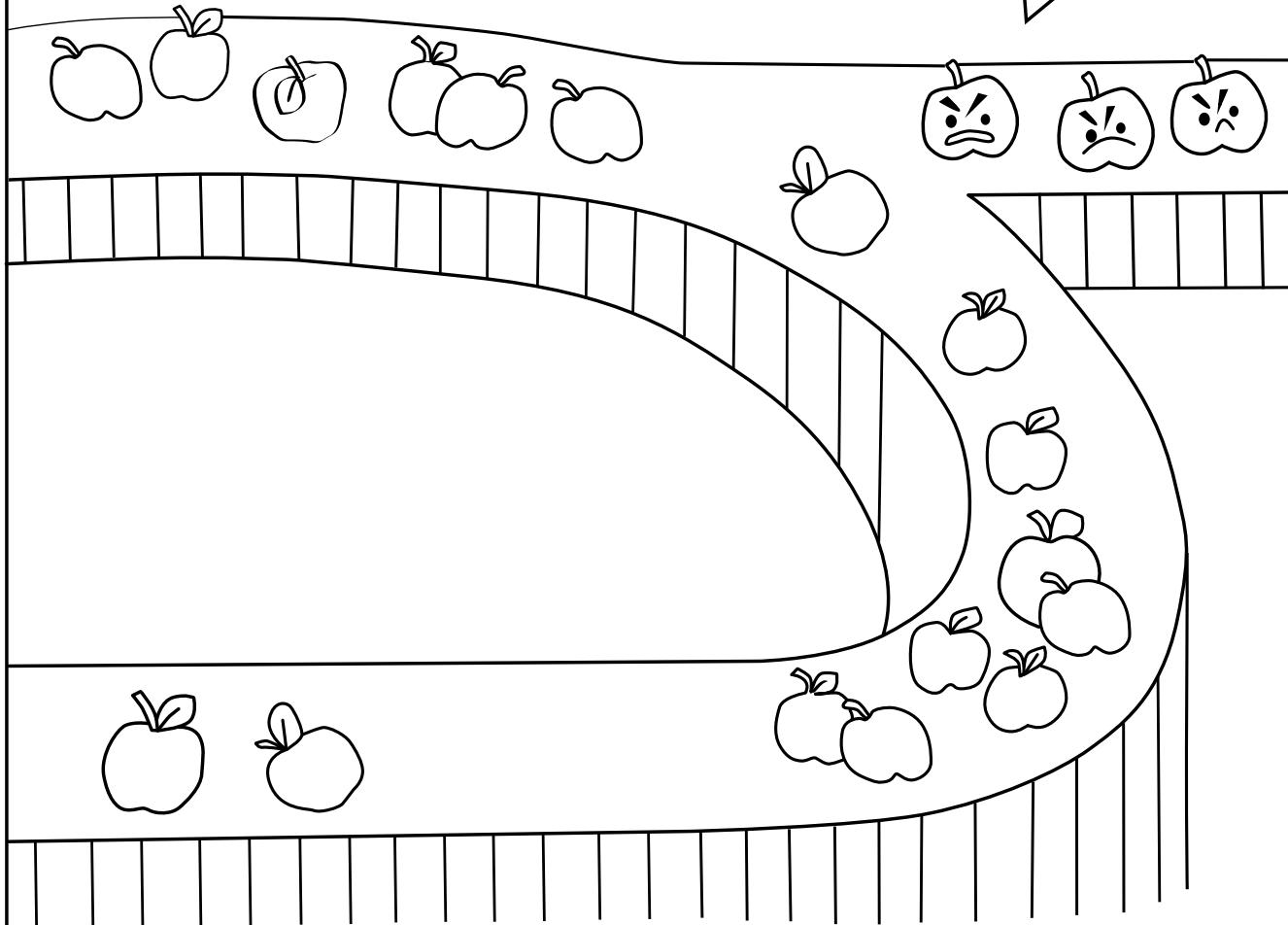
When the delivery finishes, he will stop some of the instances to free up resources for other bakery tasks.



EVENT KNATIVE listens to the event channel, producing events when the new poison detector goes off, and consuming those events in order to route safe and poisoned apples.

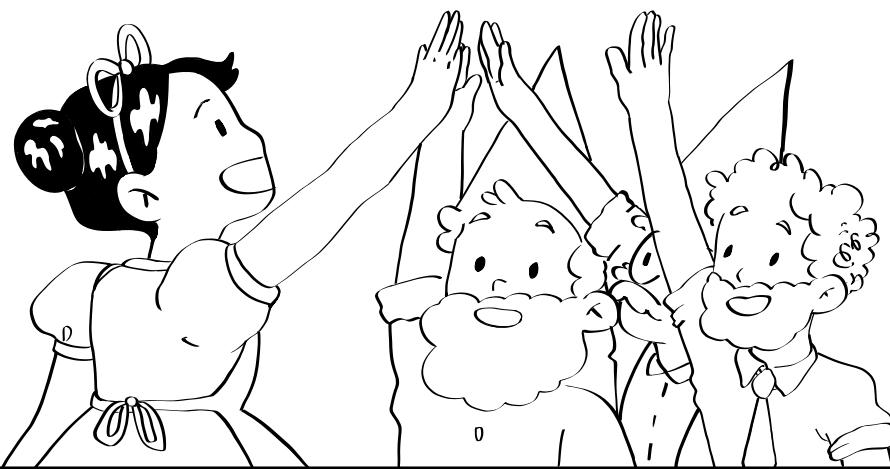
Positioned apples are routed to a new conveyor belt system where they are bagged, tagged, and sent to the Kingdom's CRIME LAB for analysis.

LAB



Safe apples are routed to the peeling and coring conveyor belt system to be baked into pies.

YIPPEE !



Meanwhile, at the evil
QUEEN's lair...

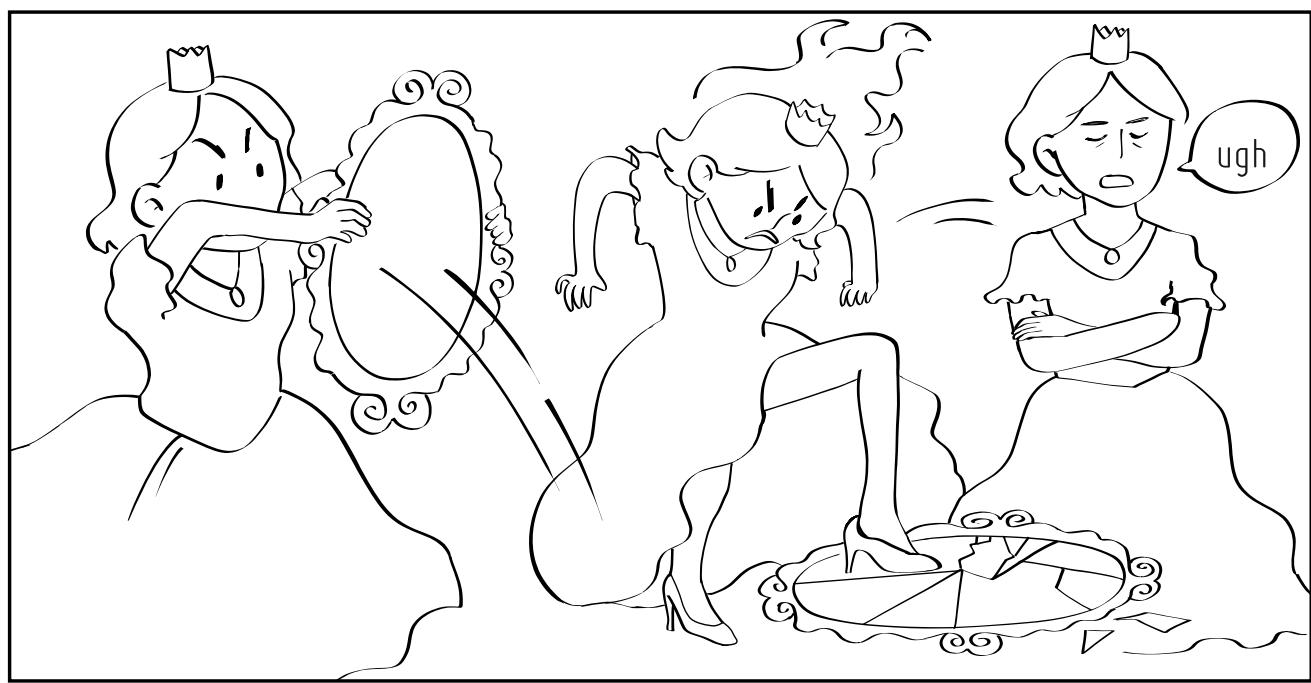


Mirror, mirror,
on the wall...

Who's enterprise
architecture is best
of them all?

It is still SNOW WHITE's bakery,
that has a serverless event-driven
architecture, and not our own
architecture.

Um... My QUEEN...



The
End



Special thanks to everyone who contributed to this coloring book:



Dan Juengst



William Henry



Langdon White



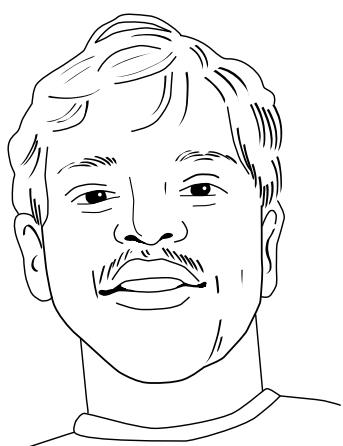
Kimberly Craven



Brian Harrington



Brian Tannous



Kamesh Sampath



Madeline Peck



Máirín Duffy

As well as Mrunal Patel, Emma Kidney, Jess Chitas and Victoria Peck.

GLOSSARY

Kubernetes – a secure, efficient and reliable open-source distributed system that runs on top of an operating system while interacting with pods of containers running on the physical computers (node). Kubernetes takes the commands from an administrator and relays those instructions to the computing machines. It automates software deployment, scaling, and management.

Software Development Pipeline – a process that drives software development through a path of building, testing, and deploying code. By automating the process, the objective is to minimize human error and maintain a consistent process for how software is released.

Container – an application packaging standard that provides all the files necessary for the application's entire runtime environment. It's portable and makes it easy to move the contained application between environments (dev, test, production, etc.) while retaining full functionality.

OpenShift – owned by Red Hat it is one of the leading distributions of Kubernetes. It's a consistent hybrid cloud for building and scaling containerized applications.

Serverless – Serverless computing is a cloud-native development model that allows developers to build and run applications without having to manage servers for a short period of time. Physical servers are used but don't need to be managed by developers.

Deployment – Deployment automation provides the ability to move your software between testing and production environments by using automated processes. This leads to repeatable and reliable deployments across the software delivery cycle.

Event – The record that something happened within an application, often originating from the external physical environment. The event is generated in one of the architecture components and is then handled by the software.

EDA – Event-driven architecture is a software architecture and model for application design. With an event-driven system, the capture, communication, processing, and persistence of events are the core structure of the solution. This differs from a traditional request-driven model. A tax preparation site might use EDA to spawn additional web servers during tax time for example.

Enterprise Infrastructure – Made up of hardware, software, networking, and services required to manage an environment. The result is a standardized platform across physical, virtual, private cloud, and public cloud environments, with solutions that work as well with one another as they do with your existing technologies and processes.

Eclipse Che IDE – An integrated development environment, is software for building applications that combines common developer tools into a single graphical user interface. It is an open-source multi-user development platform that is based in Java.

Learn more at redhat.com:



[https://
www.redhat.com/en/
blog/importance-
event-driven-
architecture-digital-
world](https://redhat.com/en/blog/importance-event-driven-architecture-digital-world)



[https://red.ht/
coloring](https://red.ht/coloring)