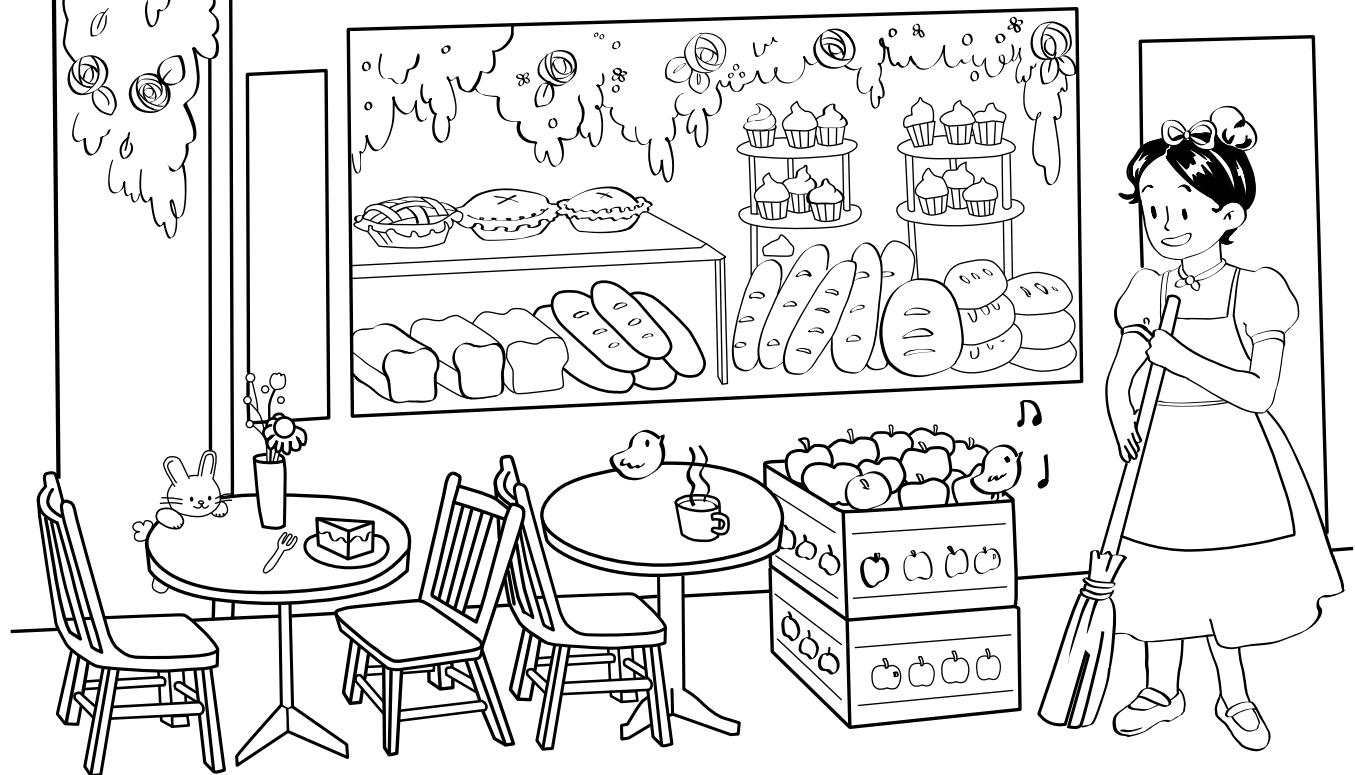


Once Upon a Time...

Cute Cuvette

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

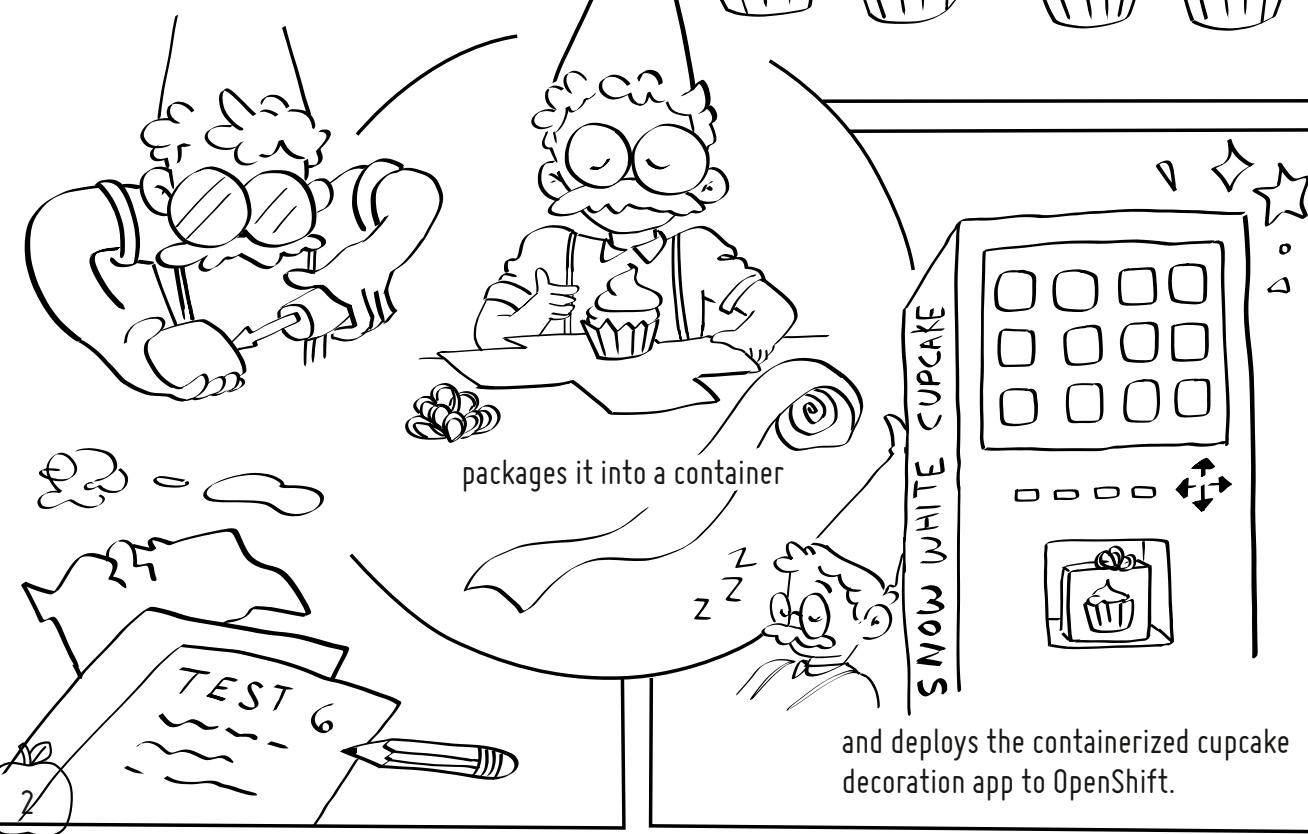


When Snow White develops a new application for her BAKERY- say, a cupcake decoration application-



TEKTON takes her app code through a software development pipeline.

He builds & tests the application code-



and deploys the containerized cupcake decoration app to OpenShift.

SERVE KNATIVE is a server dwarf

Anyone need anything?



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

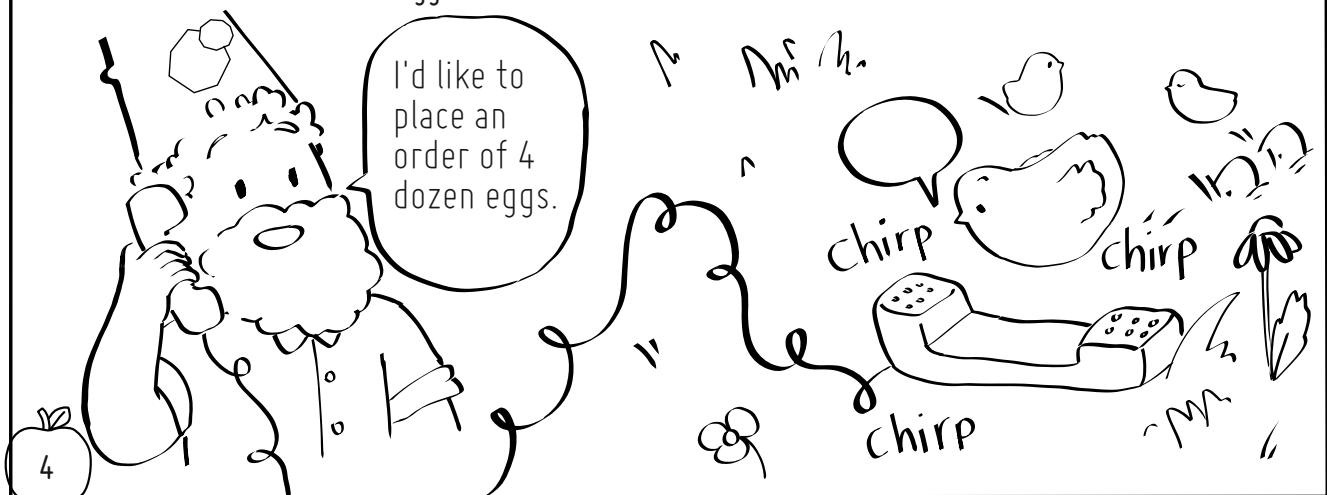
EVENT KNATIVE is an eventing dwarf



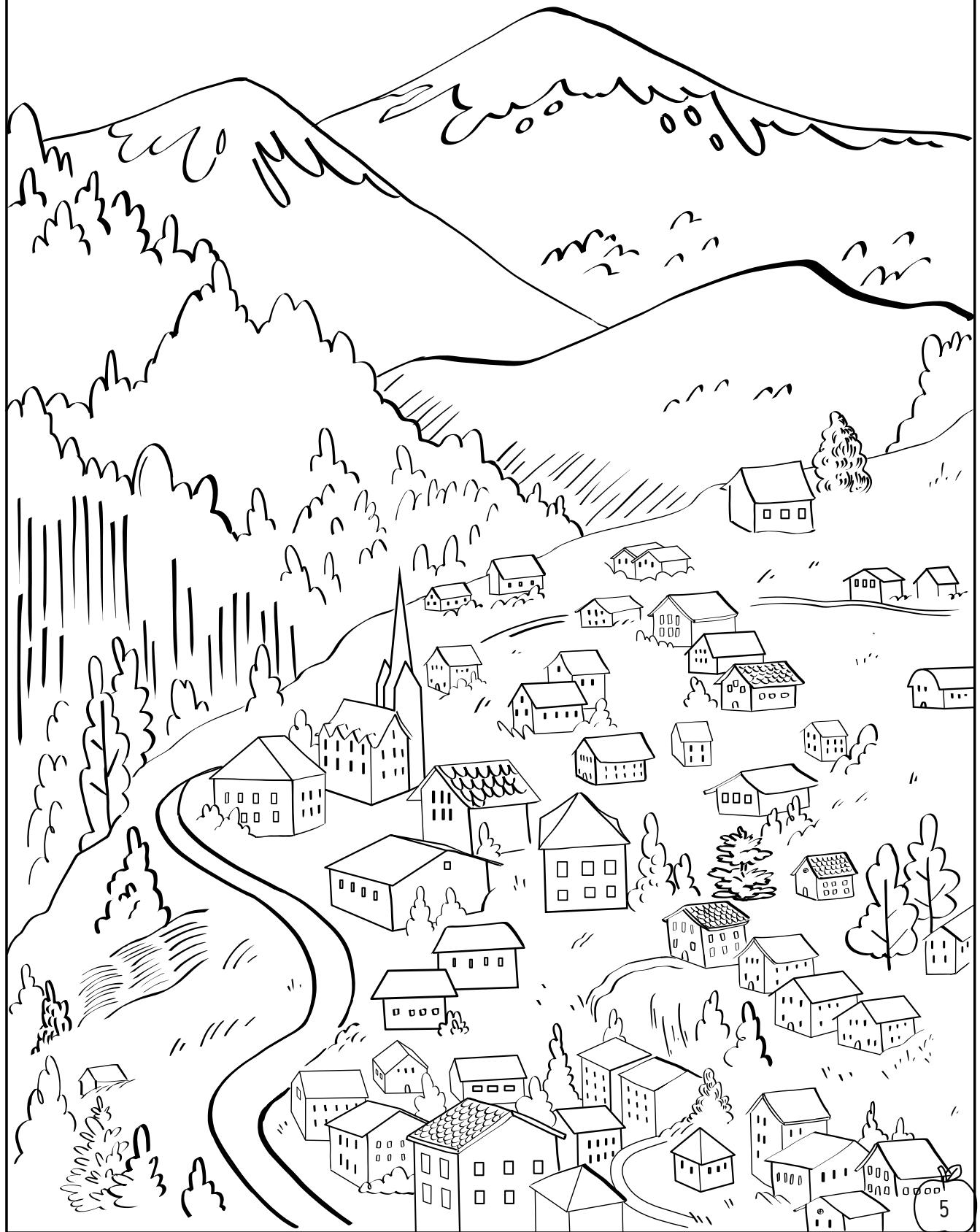
For example, as SNOW WHITE uses eggs up in her bakery, it may produce egg usage events



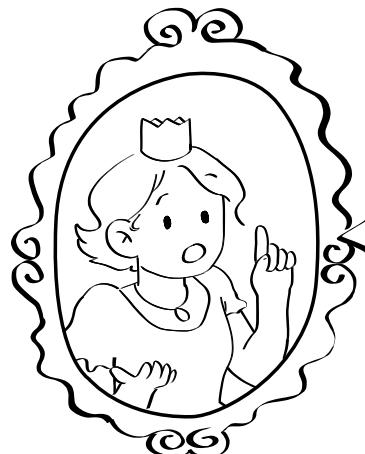
When it consumes enough egg usage events above a certain threshold, EVENT KNATIVE may place an order with CHICKEN LITTLE for more eggs.



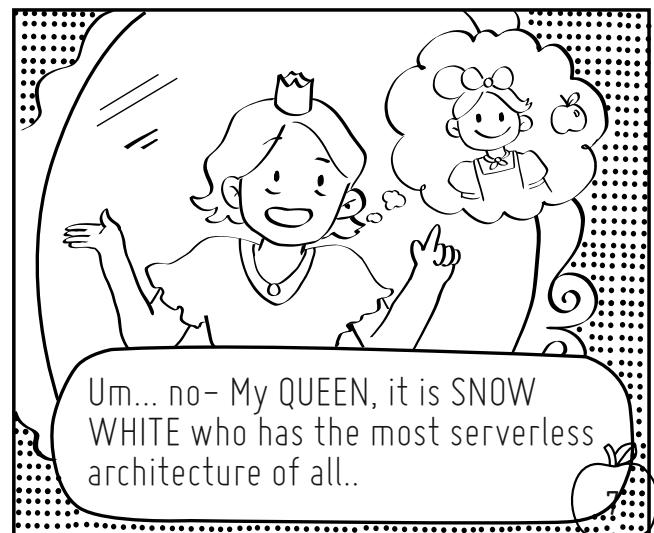
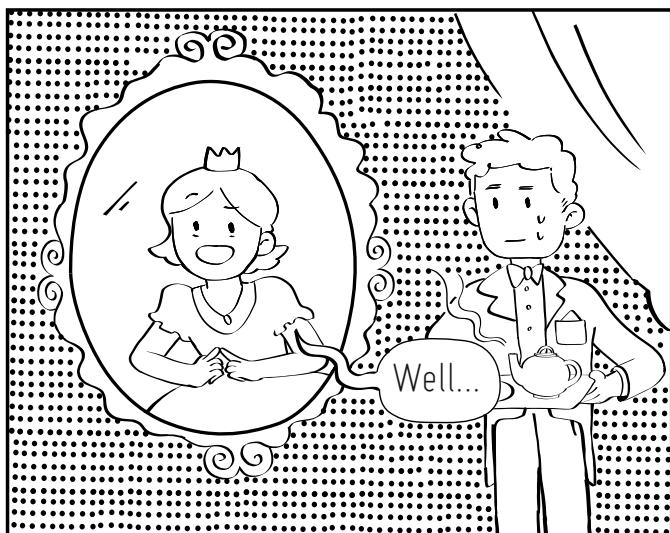
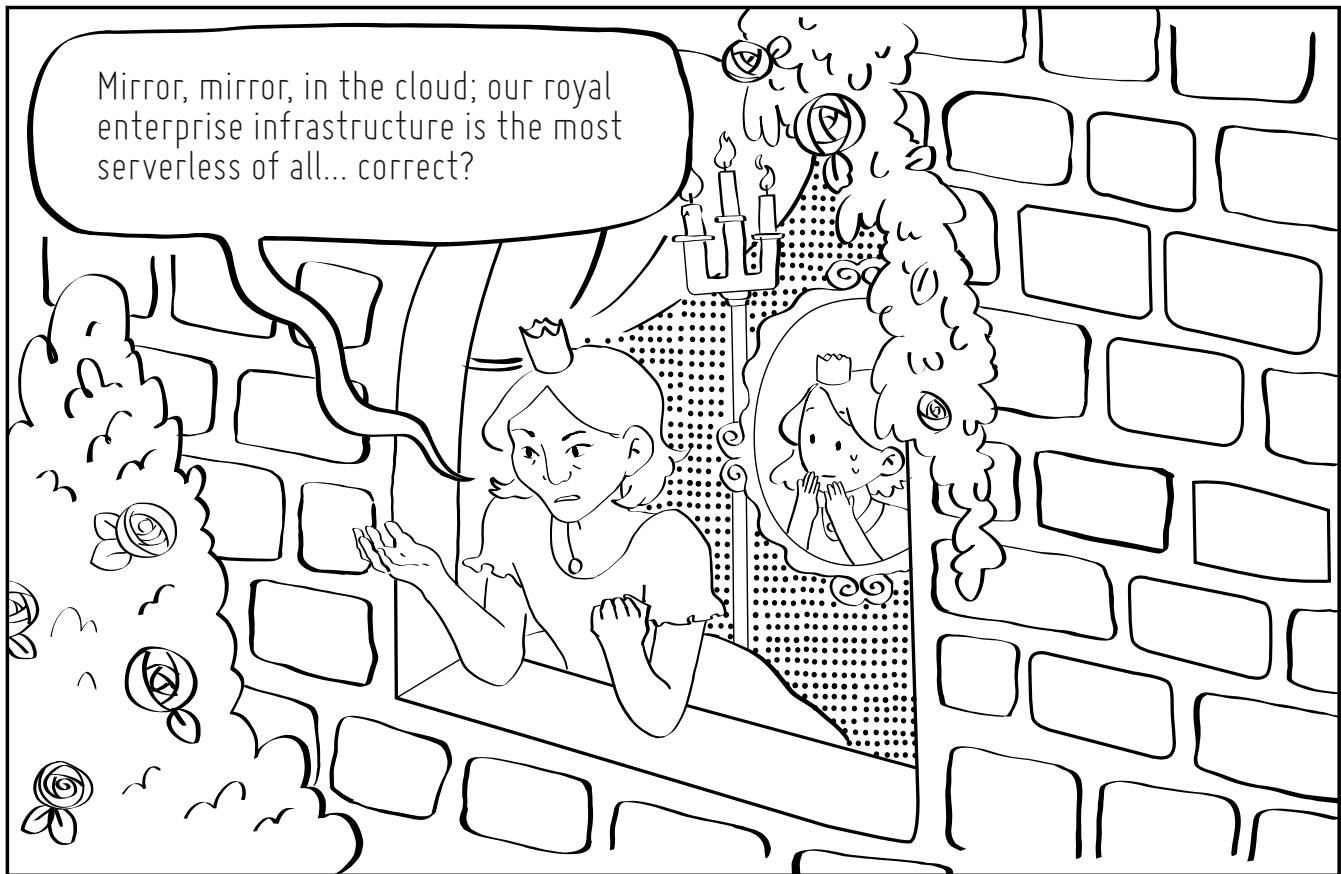
Meanwhile, in the lair of the EVIL QUEEN MALICIOUS,
on a mountain top overlooking the Kubernetes Kingdom...

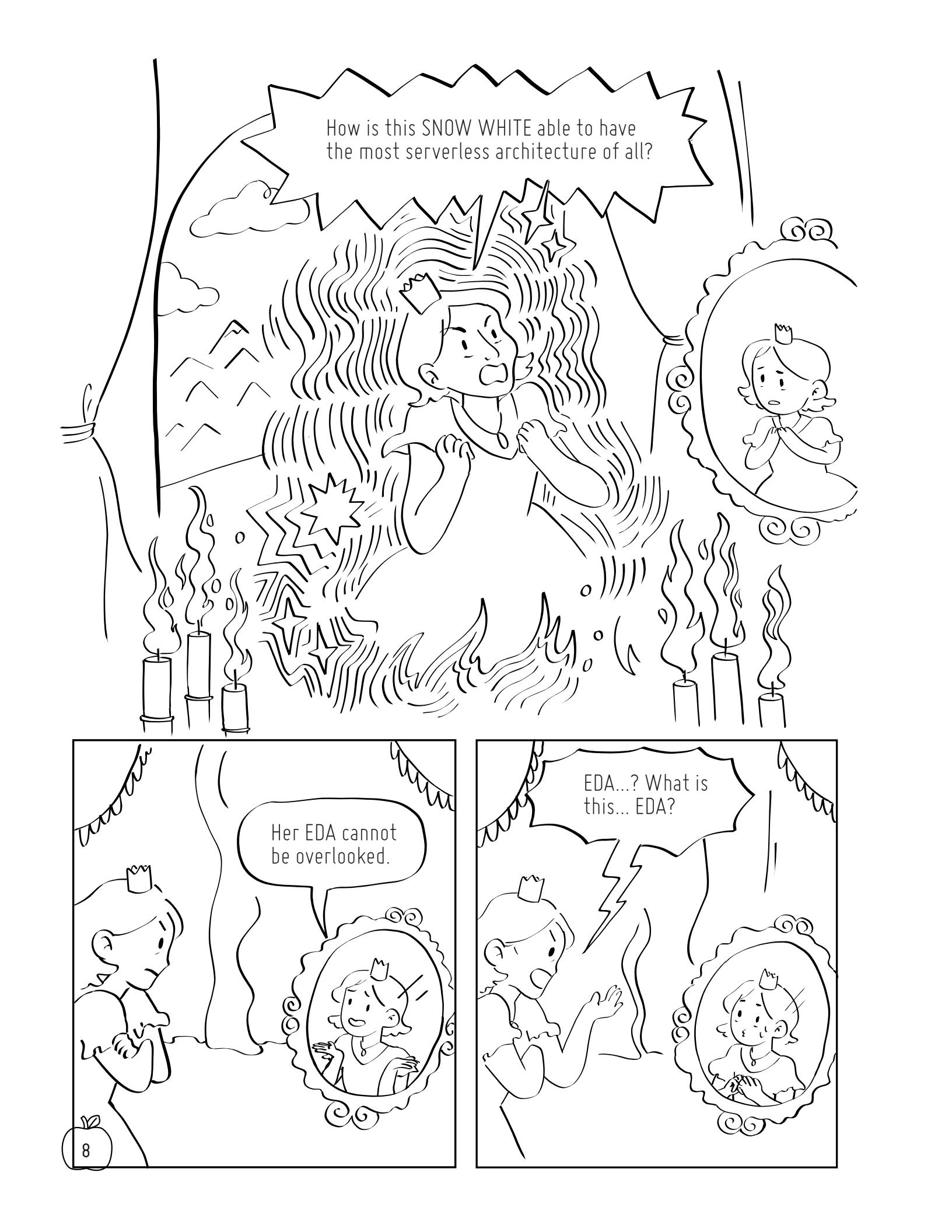






Serverless doesn't mean there are no servers, MALICIOUS. It means applications don't run 24/7-they only run if they are needed and stop when they are not. It results in a more efficient allocation of resources-and the cakes and pies are always fresh.





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

Her EDA cannot
be overlooked.

EDA...? What is
this... EDA?

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.



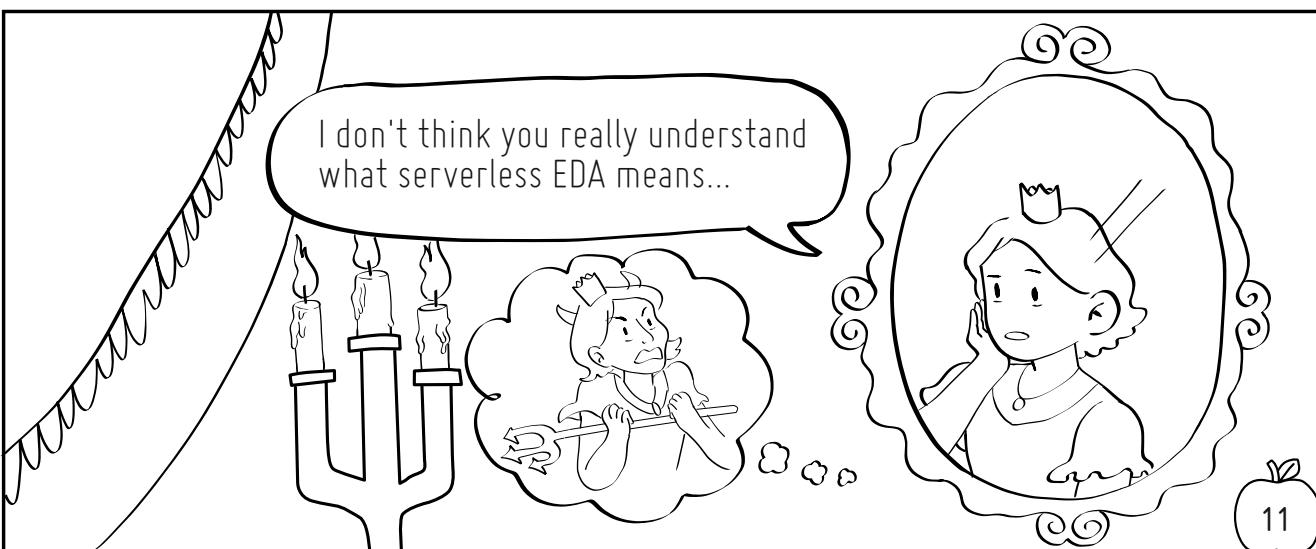
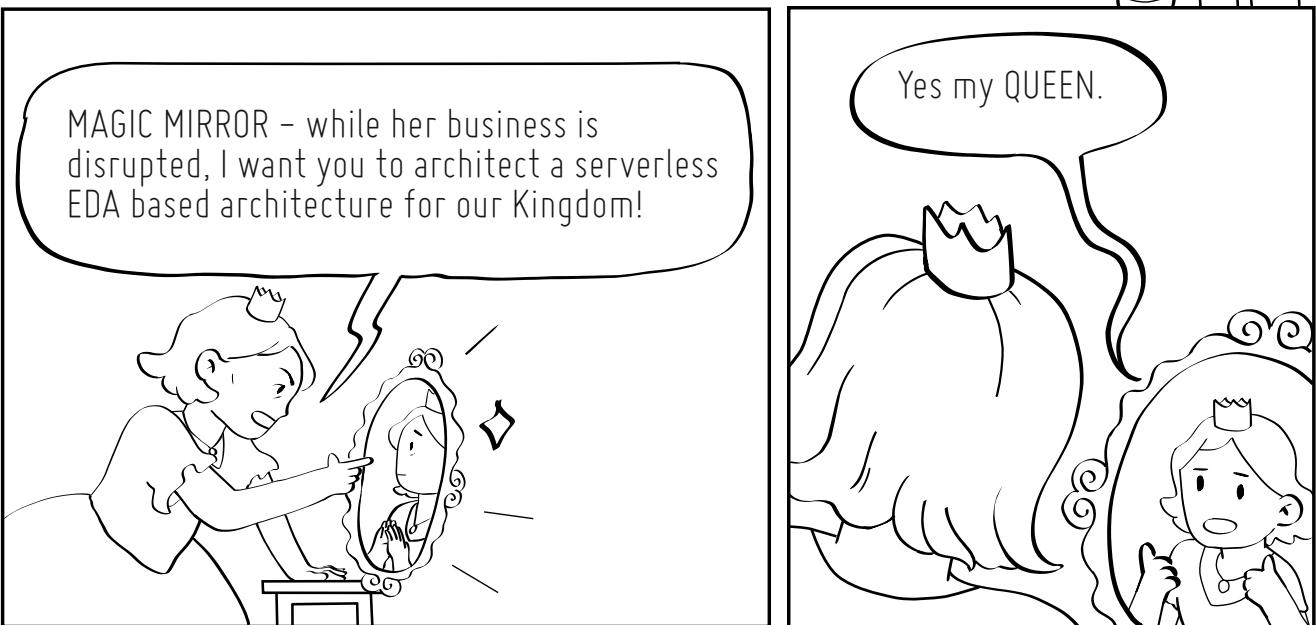
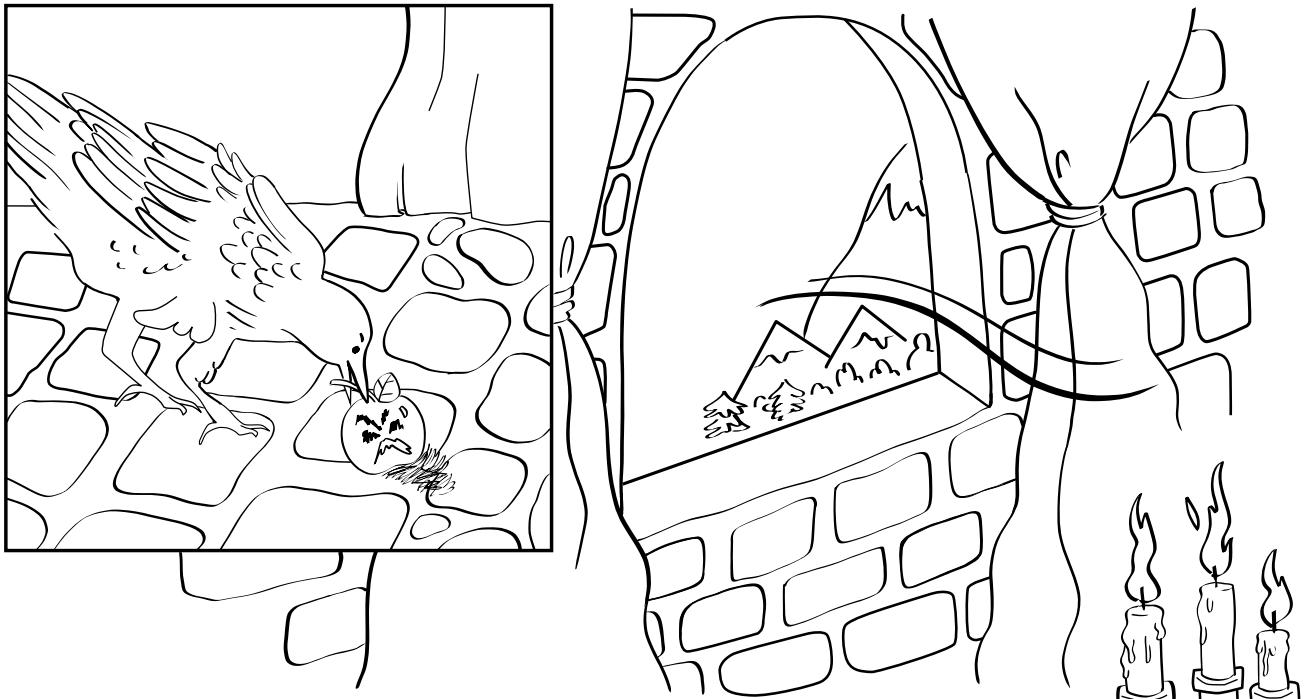
When the winter holidays rolled in a few months later, the production for cranberry pie & regular pumpkin pie was scaled up.

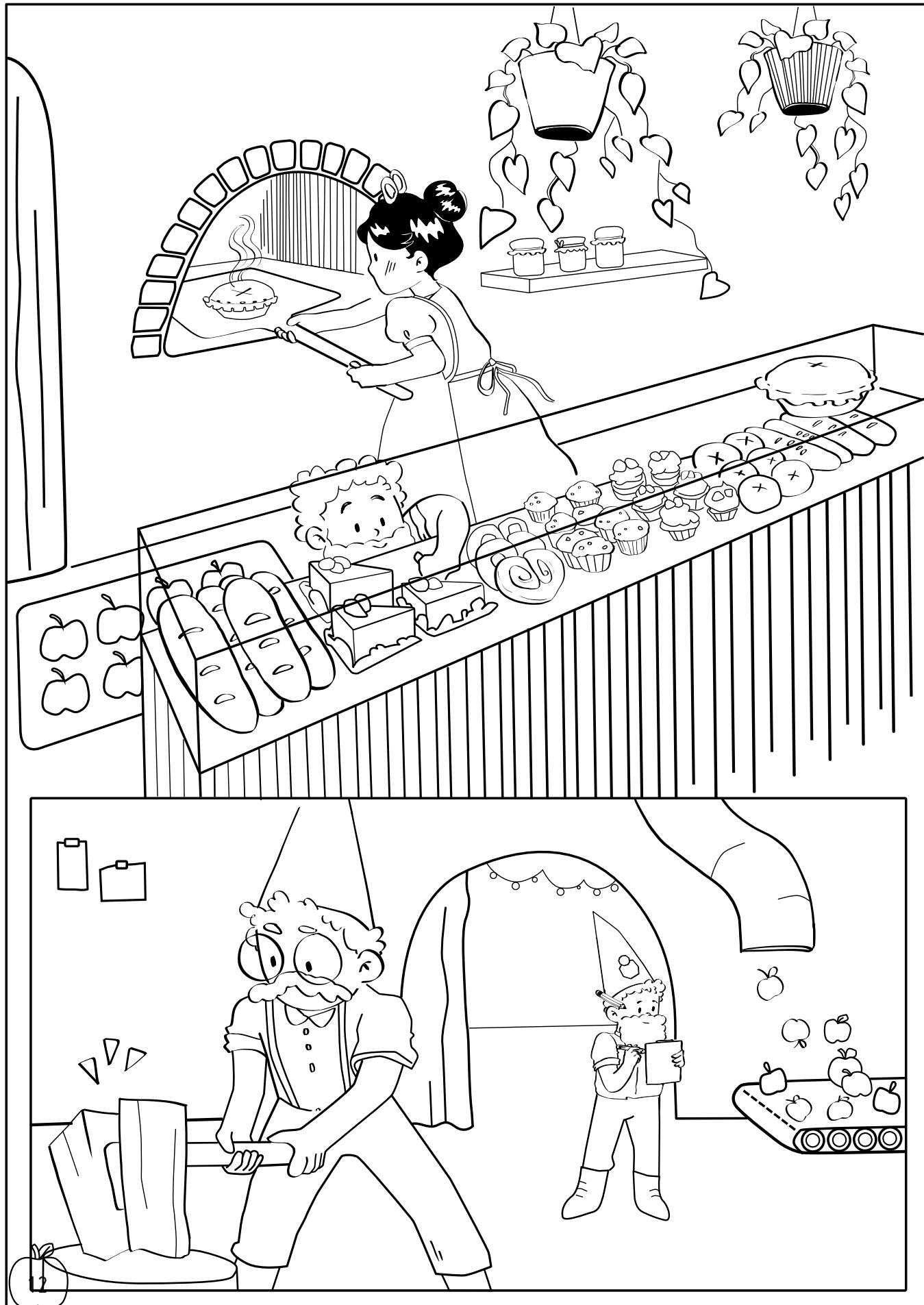


After the holidays, everyone is so full of pies and leftovers that they scale down to zero pies for as long as they need.

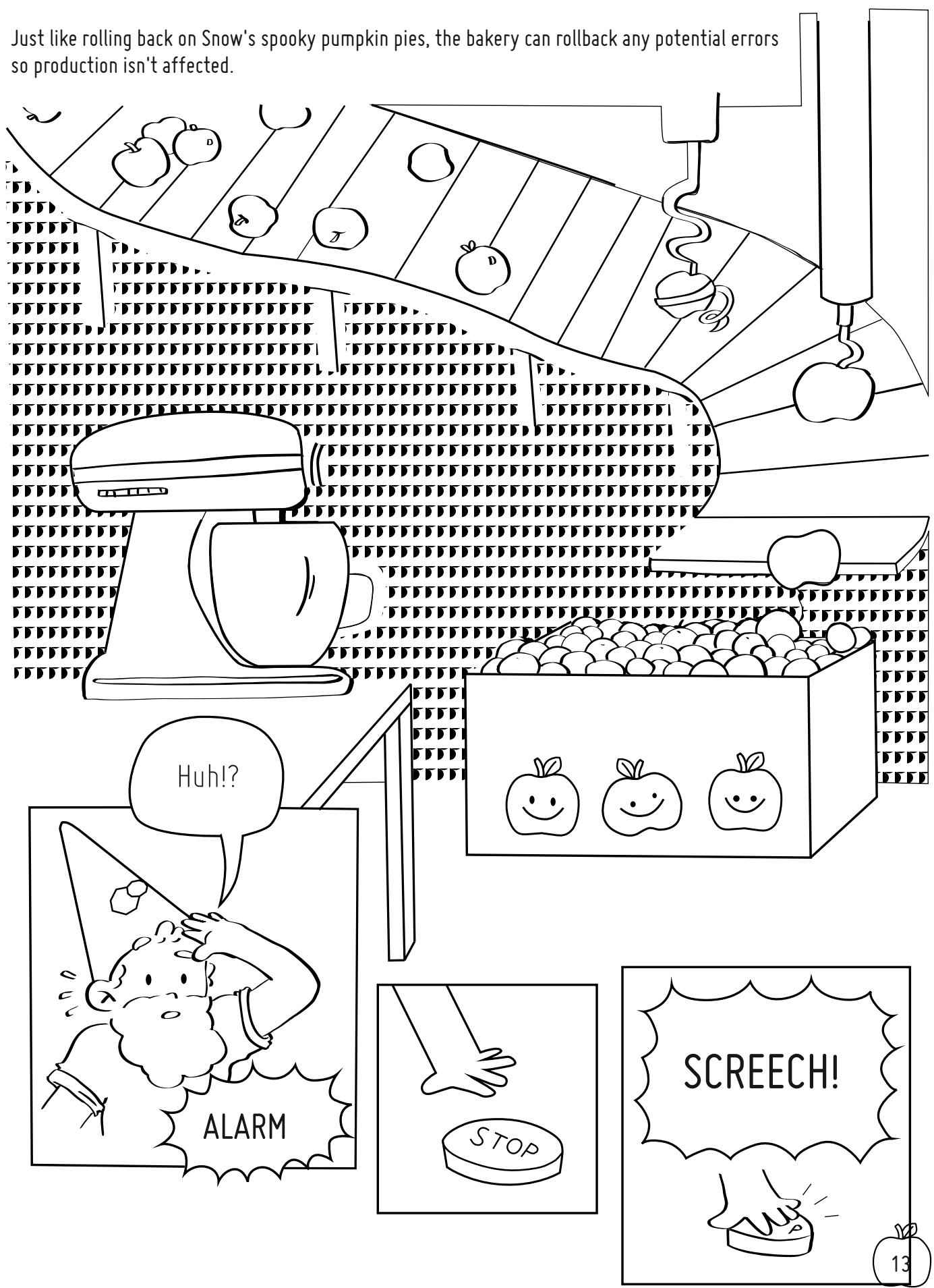


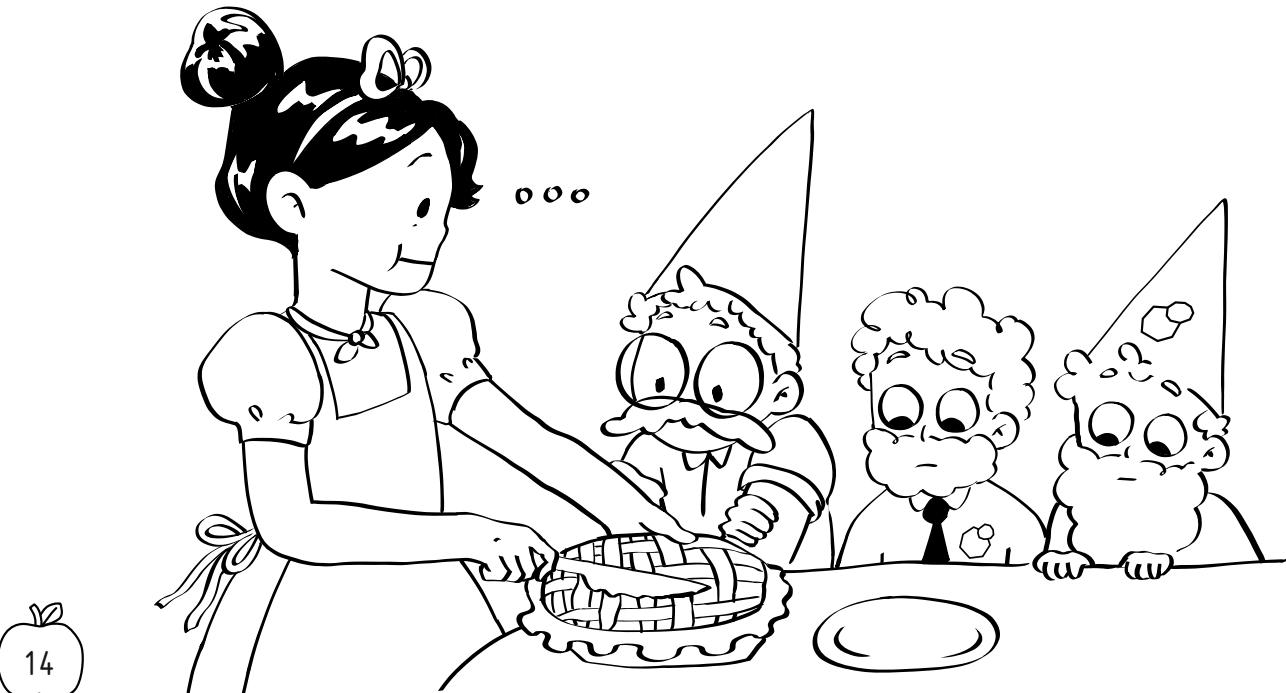






Just like rolling back on Snow's spooky pumpkin pies, the bakery can rollback any potential errors so production isn't affected.





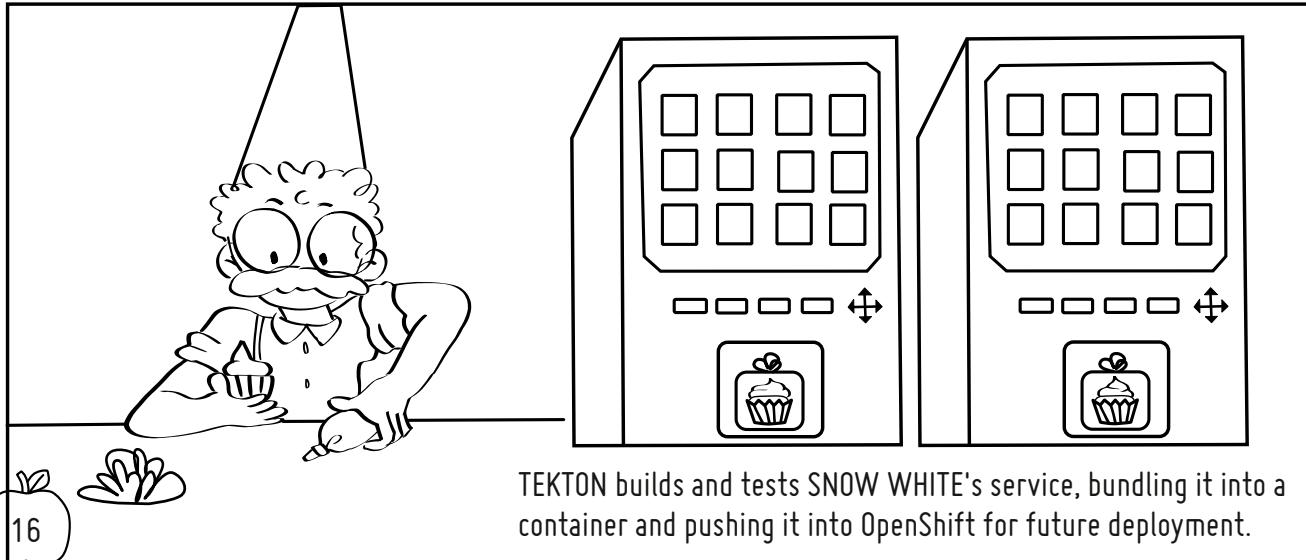
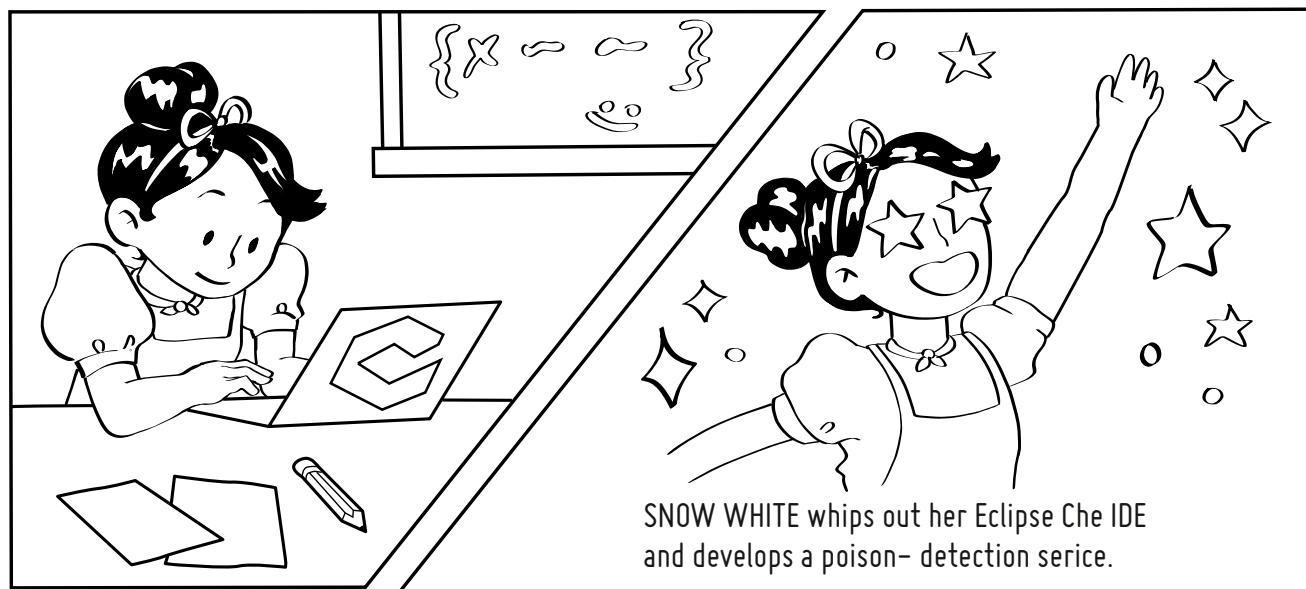


Is this... a poisoned apple in our filling?

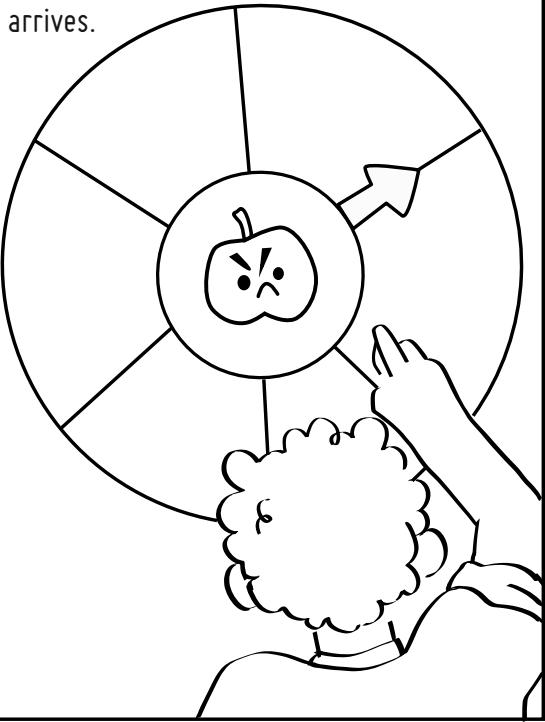
It IS!

Blegh

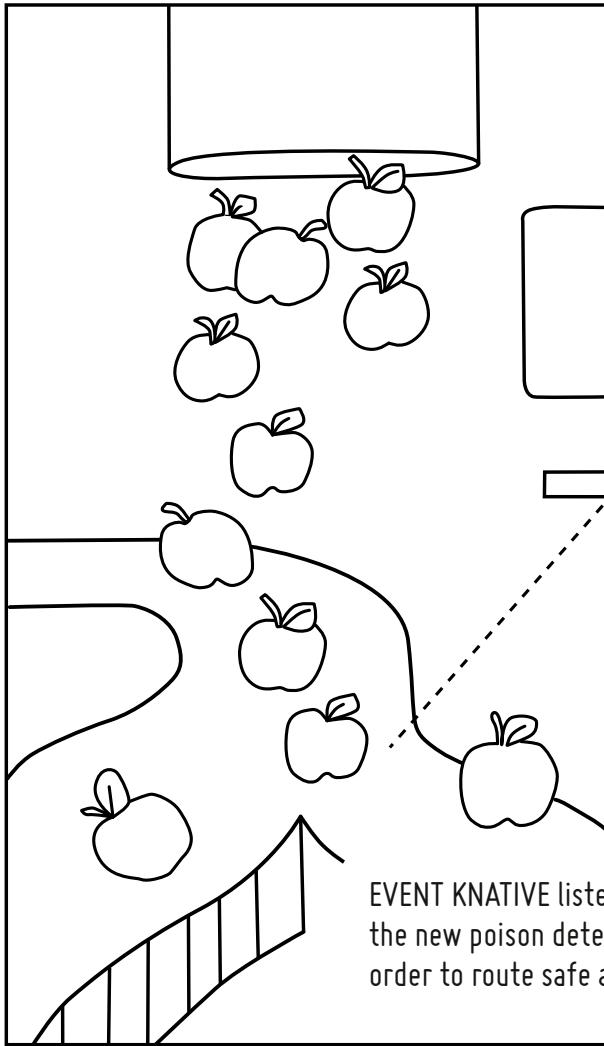




SERVE increases the number of poison detection services as a new apple delivery arrives.



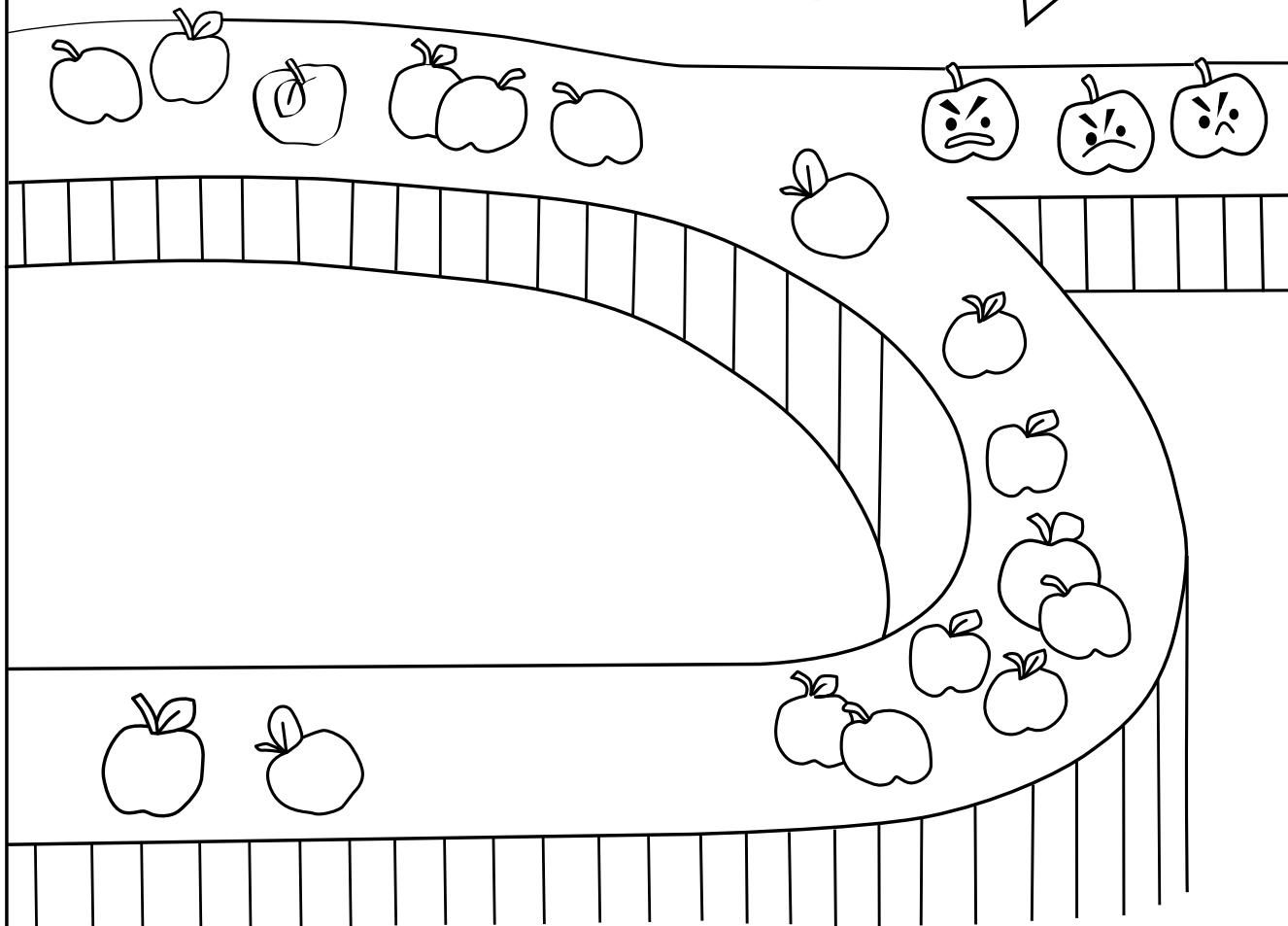
When the delivery finishes, he will stop some of the instances to free up resources for other bakery tasks.



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?

Um... My QUEEN...

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

