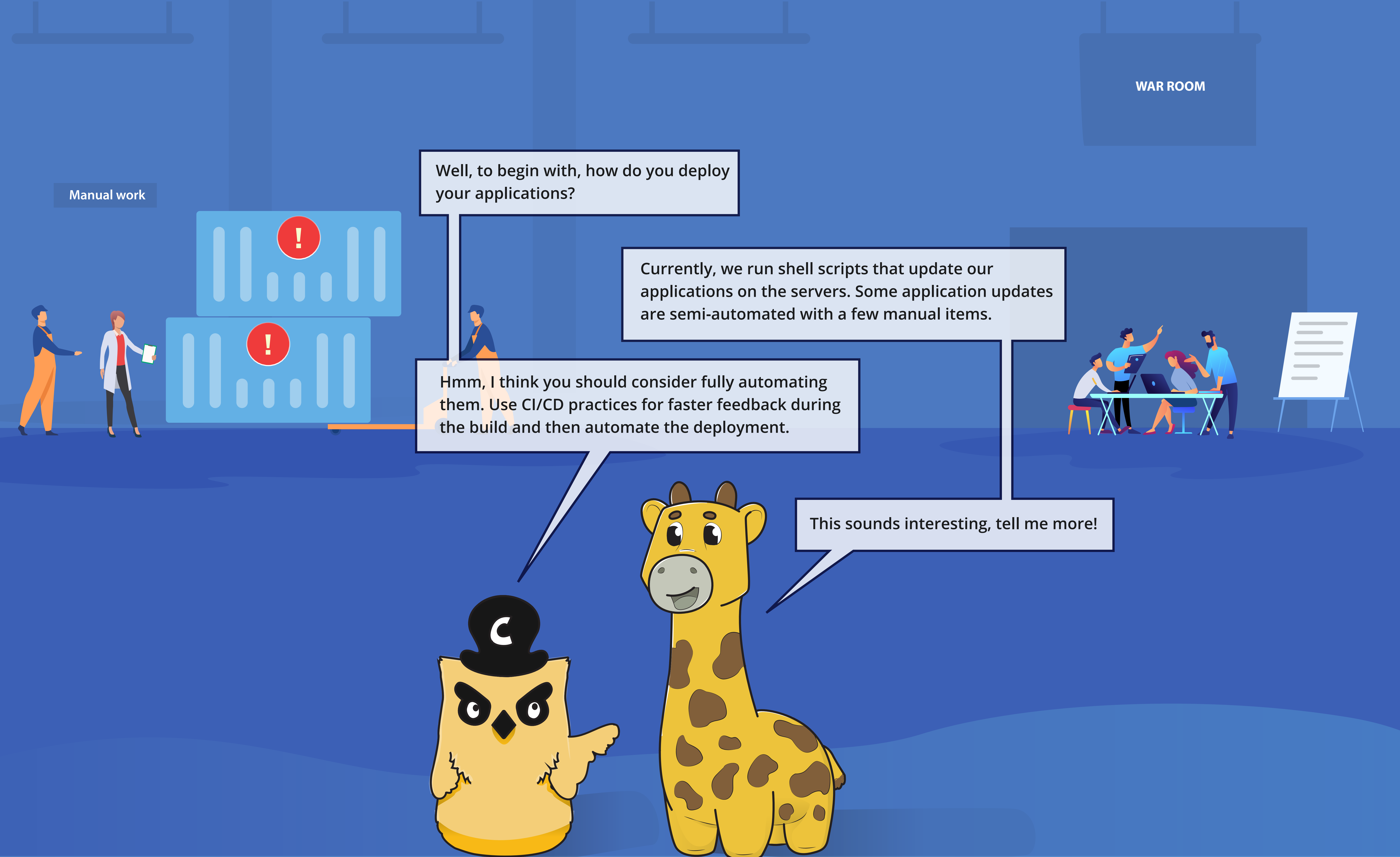
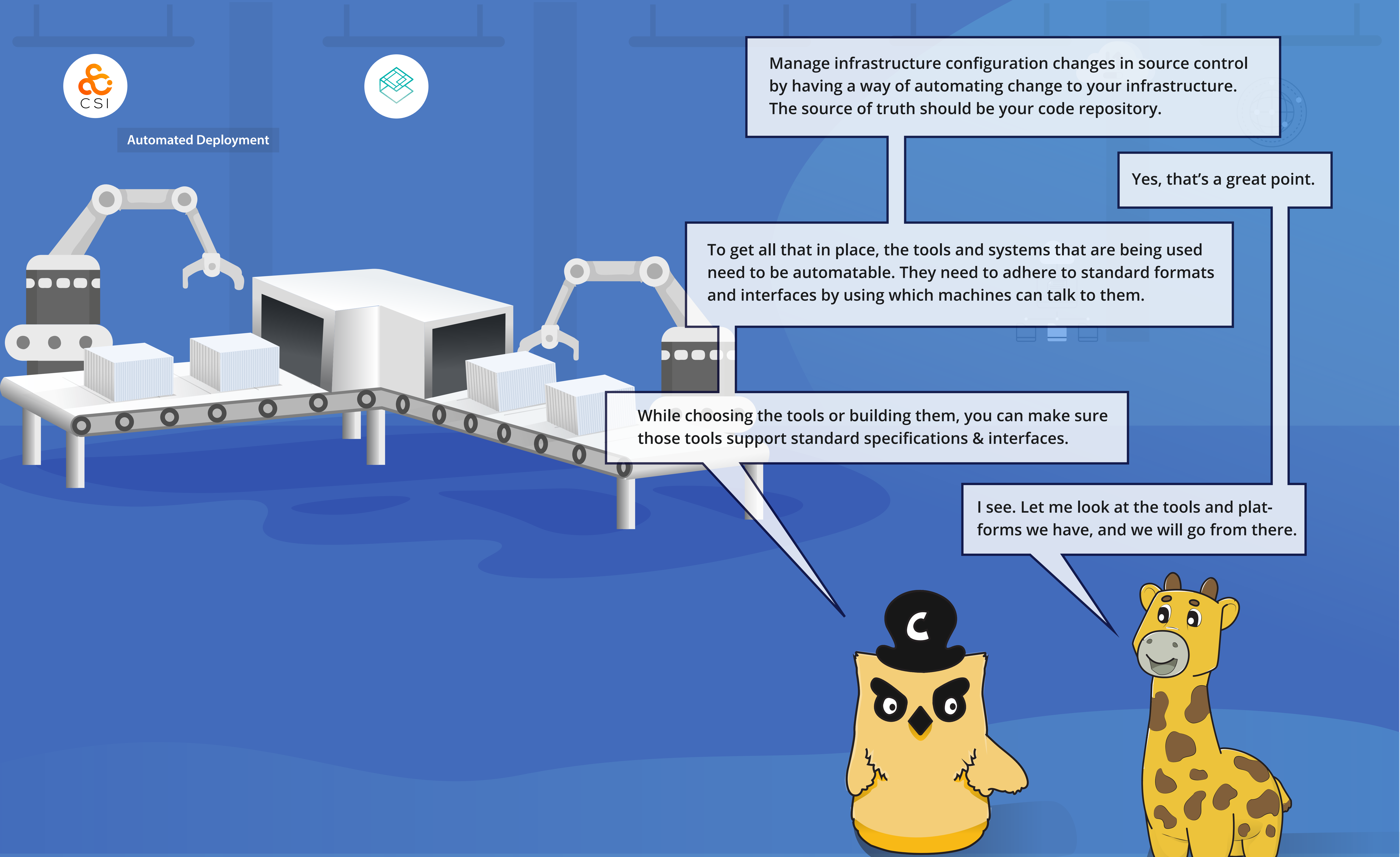
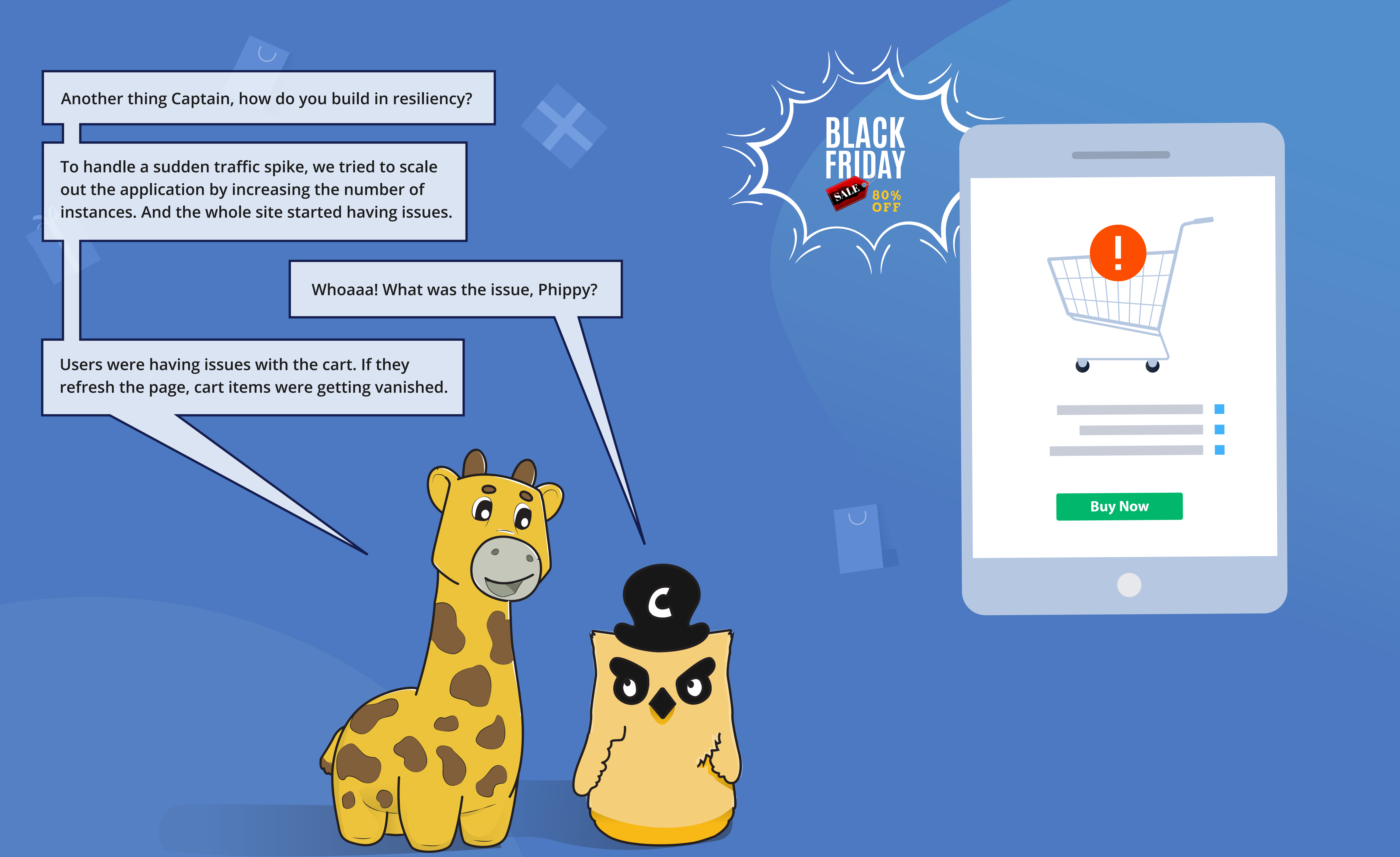
PHIPPY'S QUEST FOR CLOUD NATIVE TRANSFORMATION

Captain, I really need to prepare for this coming Black Friday sale. Being the Engg VP, I need to make sure we are always available and fault-tolerant. The last time, we had an outage, and we barely managed to band-aid it on time, but we might not be lucky enough this time.
Oh, man, that sounds scary!
BLACK FRIDAY SALE 50%

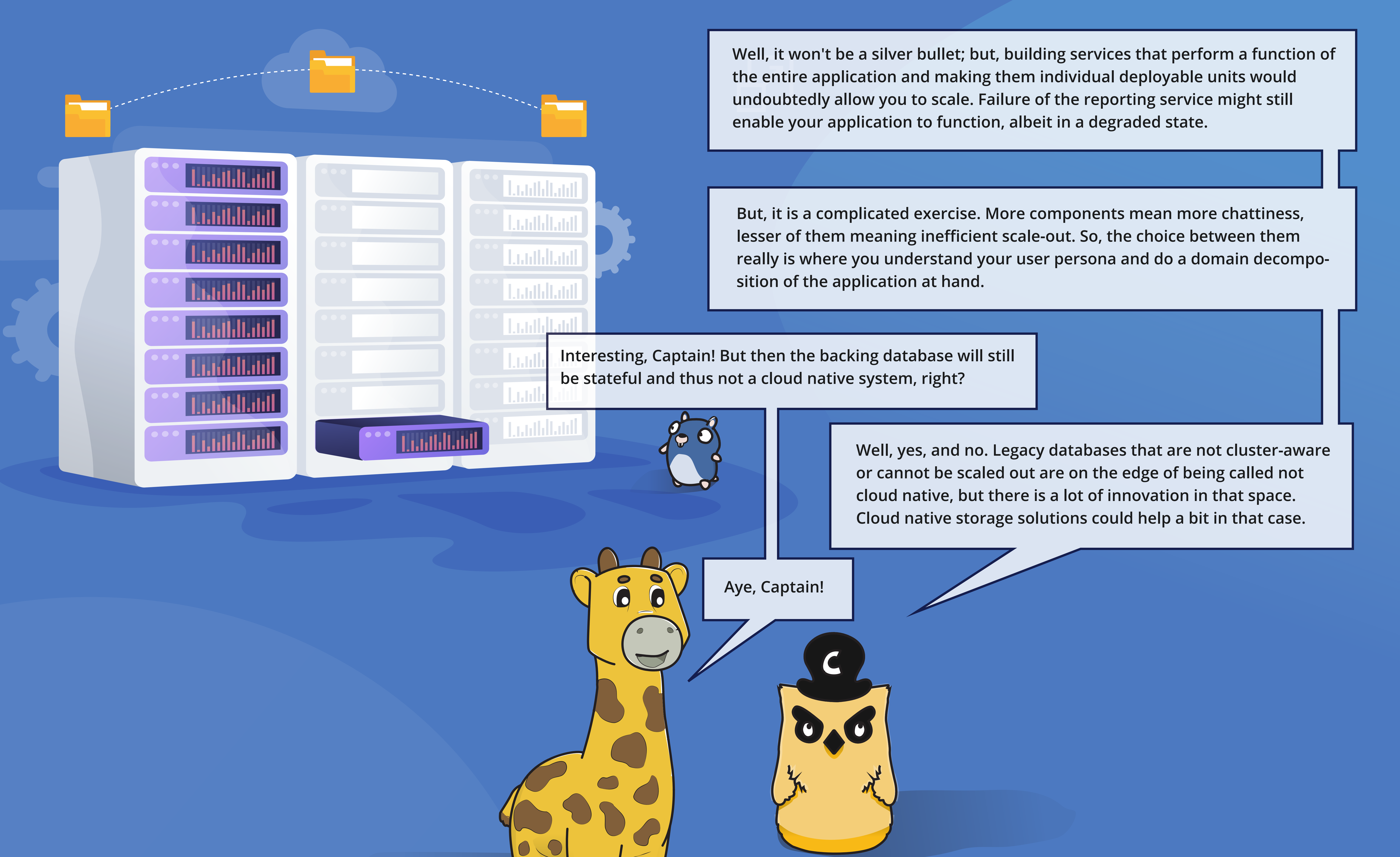
I am trying to move towards adopting the cloud native principles, but I am sort of lost with all the tools and the prescriptions that are out there. Coming from a very legacy background, I find the landscape is huge and overwhelming. I am not sure where to start and what would work for me.
I see. Yes, this is something that I would love to help you with. From what I understand and the way you described it, this is going to be an extensive and exciting transformation for you. But do not worry; we will get there, Phippy.
C
BLACK FRIDAY SALE 50%

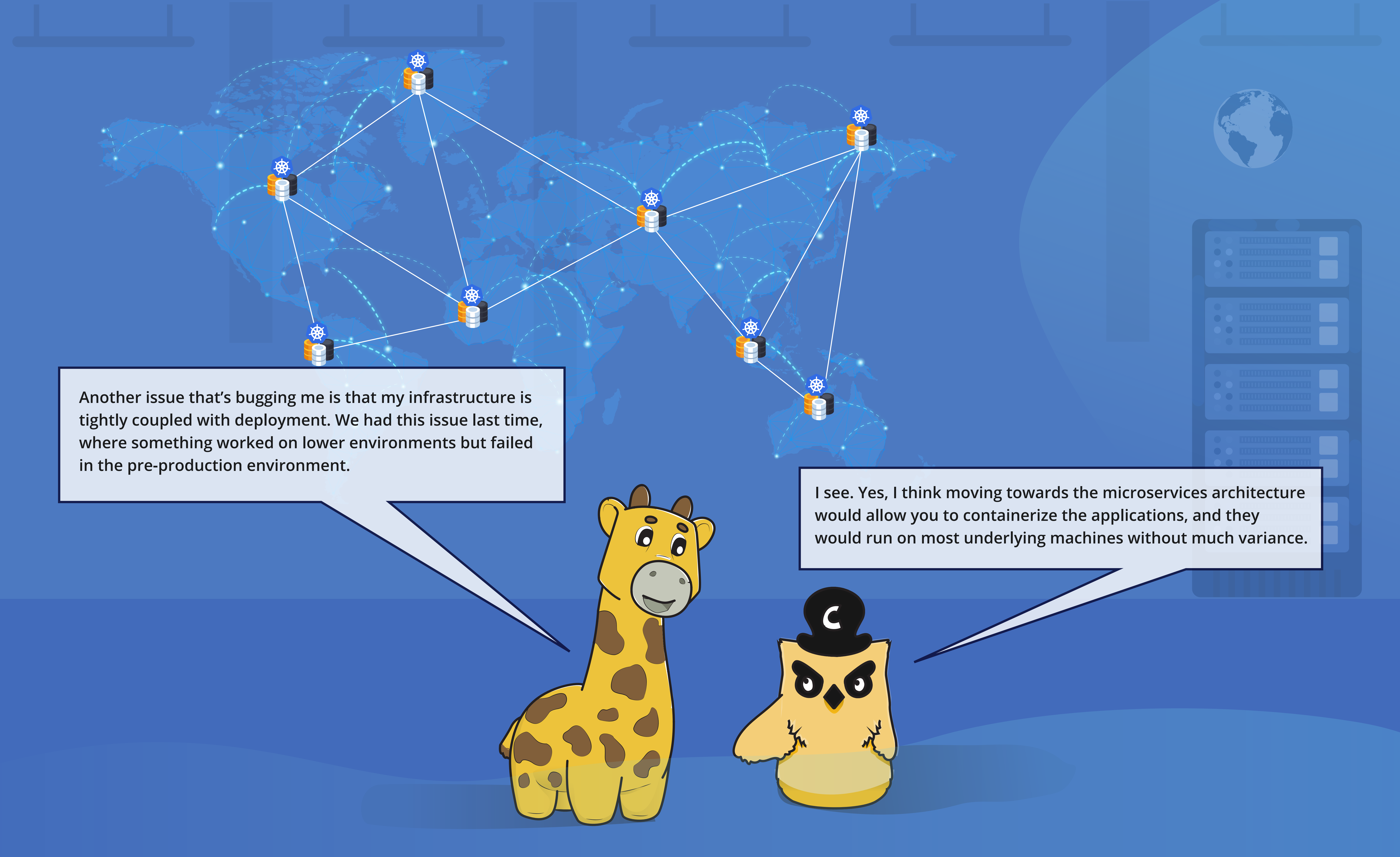

WAR ROOM
Manual work
Well, to begin with, how do you deploy your applications?
Currently, we run shell scripts that update our applications on the servers. Some application updates are semi-automated with a few manual items.
Hmm, I think you should consider fully automating them. Use CI/CD practices for faster feedback during the build and then automate the deployment.
This sounds interesting, tell me more!

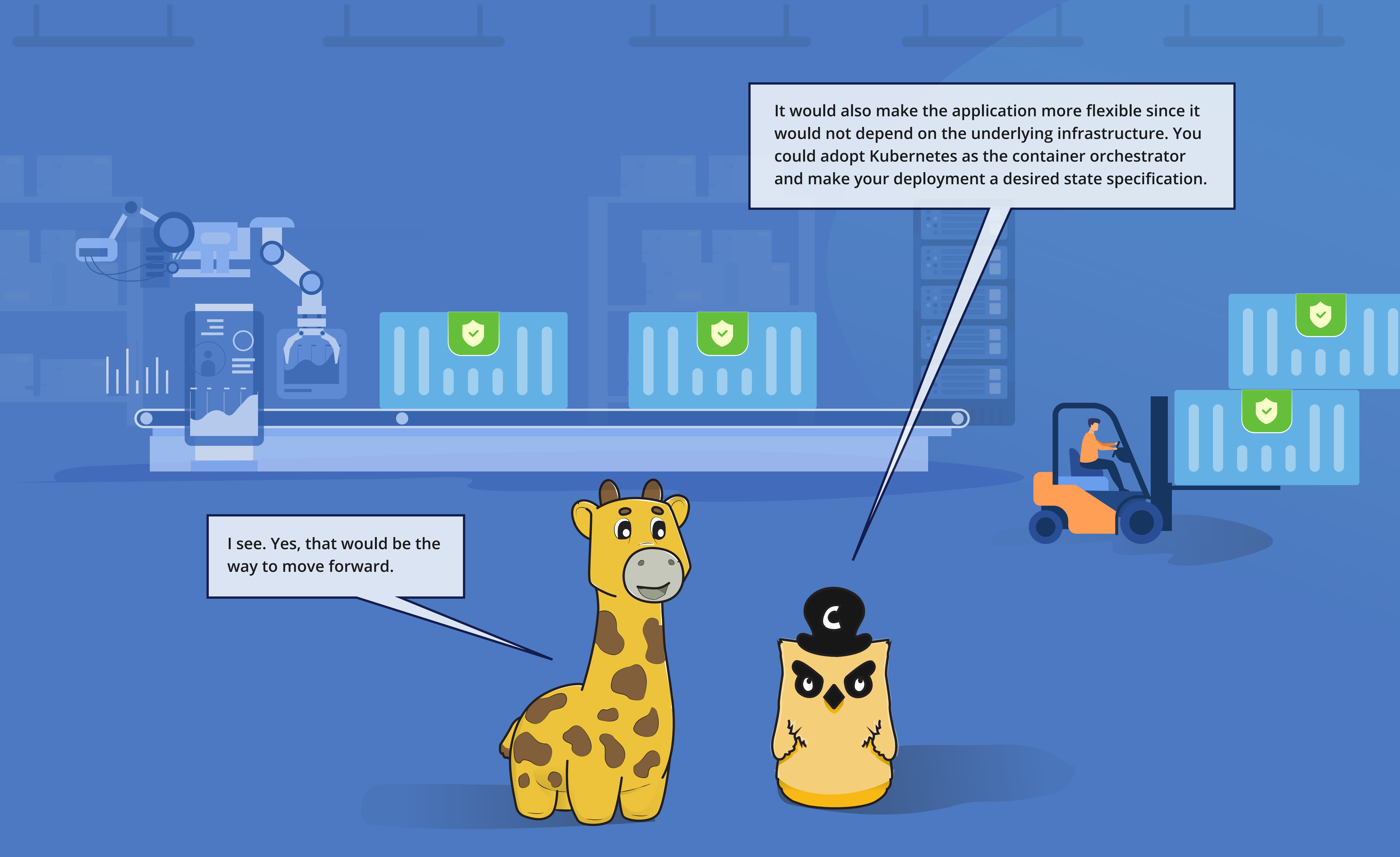

CSI
Automated Deployment
Manage infrastructure configuration changes in source control by having a way of automating change to your infrastructure. The source of truth should be your code repository.
Yes, that's a great point.
To get all that in place, the tools and systems that are being used need to be automatable. They need to adhere to standard formats and interfaces by using which machines can talk to them.
While choosing the tools or building them, you can make sure those tools support standard specifications & interfaces.
I see. Let me look at the tools and platforms we have, and we will go from there.

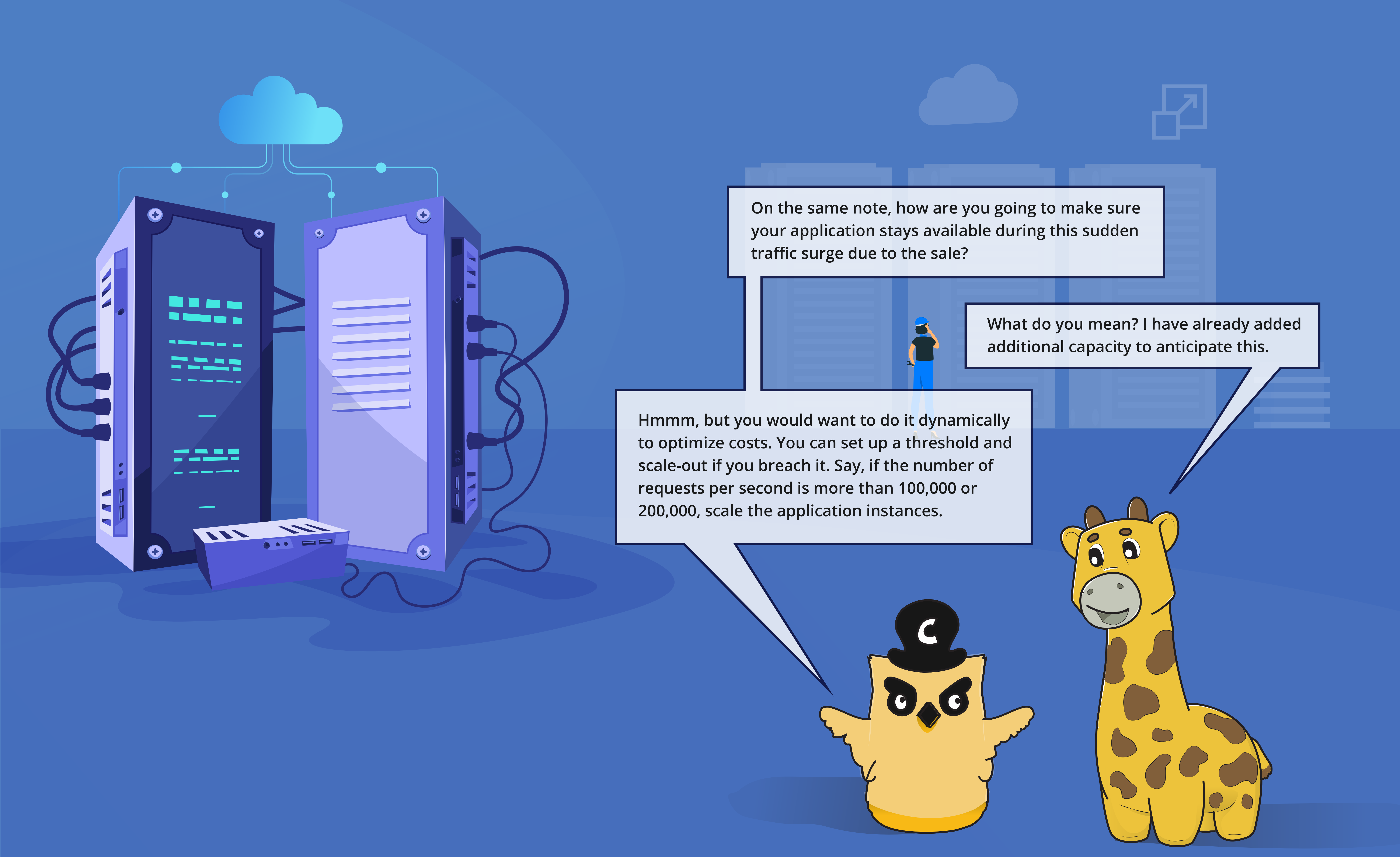

Another thing Captain, how do you build in resiliency?
To handle a sudden traffic spike, we tried to scale out the application by increasing the number of instances. And the whole site started having issues.
Whoaaa! What was the issue, Phippy?
Users were having issues with the cart. If they refresh the page, cart items were getting vanished.
BLACK FRIDAY
SALE
80% OFF
Buy Now

I see. It seems more like an application problem but let me ask you this: Do you store the cart information in memory of the application?
Ummmm, yeah, we do that. We realized that it is not a great practice unless we use a dedicated caching component.
That's right. Ideally, you should not store any state data at places like memory. Using some backing service like a caching component is a way to go.
Basically, it will make your application scalable as it becomes stateless. Any instance of the application can serve the user request in that case.
What about microservices? Do I need to go through this exercise of building services for this giant monolithic application? Would it make it more resilient and scalable???
?

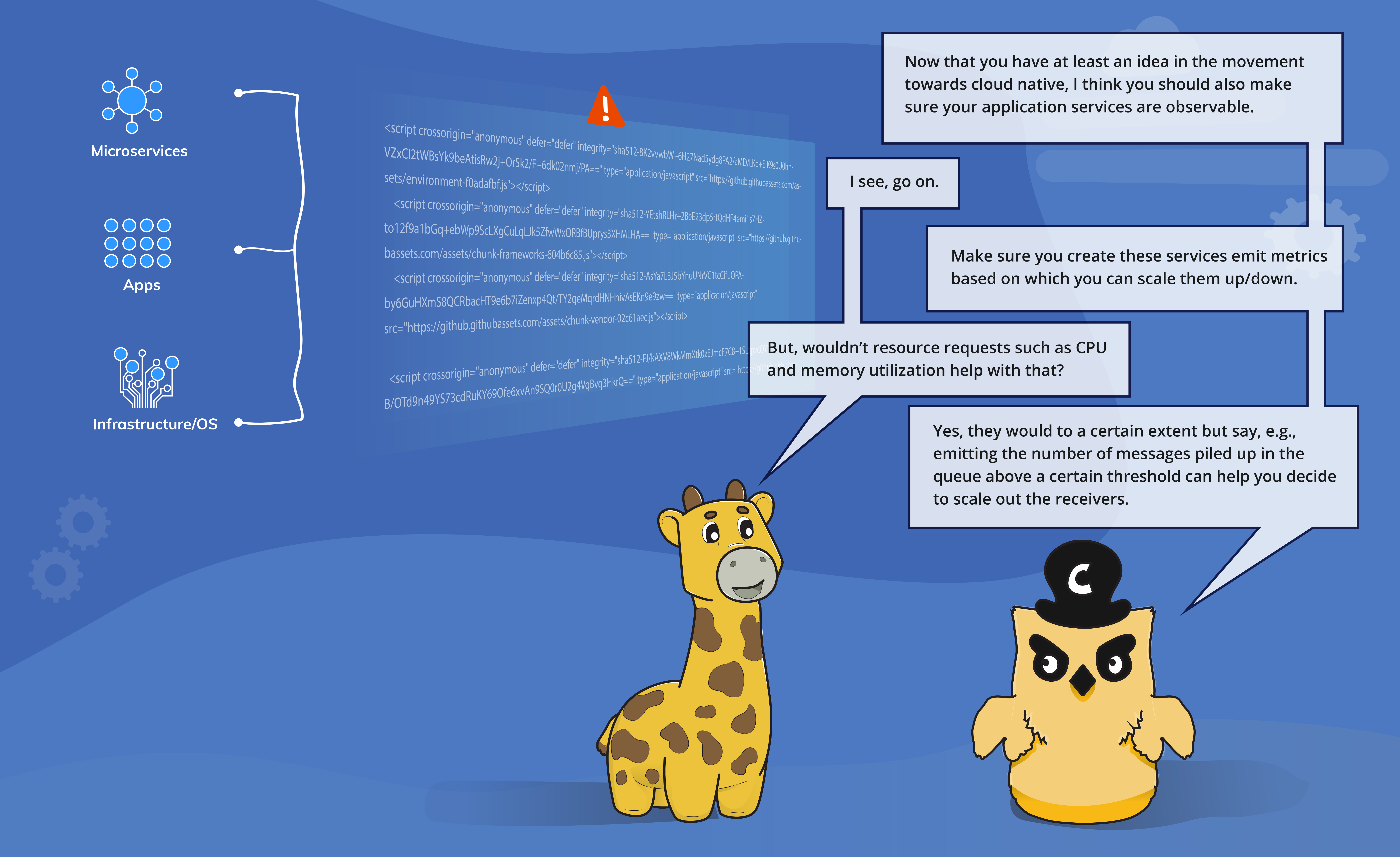

Well, it won't be a silver bullet; but, building services that perform a function of the entire application and making them individual deployable units would undoubtedly allow you to scale. Failure of the reporting service might still enable your application to function, albeit in a degraded state.
But, it is a complicated exercise. More components mean more chattiness, lesser of them meaning inefficient scale-out. So, the choice between them really is where you understand your user persona and do a domain decomposition of the application at hand.
Interesting, Captain! But then the backing database will still be stateful and thus not a cloud native system, right?
Well, yes, and no. Legacy databases that are not cluster-aware or cannot be scaled out are on the edge of being called not cloud native, but there is a lot of innovation in that space. Cloud native storage solutions could help a bit in that case.
Aye, Captain!


Another issue that’s bugging me is that my infrastructure is tightly coupled with deployment. We had this issue last time, where something worked on lower environments but failed in the pre-production environment.
I see. Yes, I think moving towards the microservices architecture would allow you to containerize the applications, and they would run on most underlying machines without much variance.


It would also make the application more flexible since it would not depend on the underlying infrastructure. You could adopt Kubernetes as the container orchestrator and make your deployment a desired state specification.
I see. Yes, that would be the way to move forward.
C


On the same note, how are you going to make sure your application stays available during this sudden traffic surge due to the sale?
What do you mean? I have already added additional capacity to anticipate this.
Hmmm, but you would want to do it dynamically to optimize costs. You can set up a threshold and scale-out if you breach it. Say, if the number of requests per second is more than 100,000 or 200,000, scale the application instances.

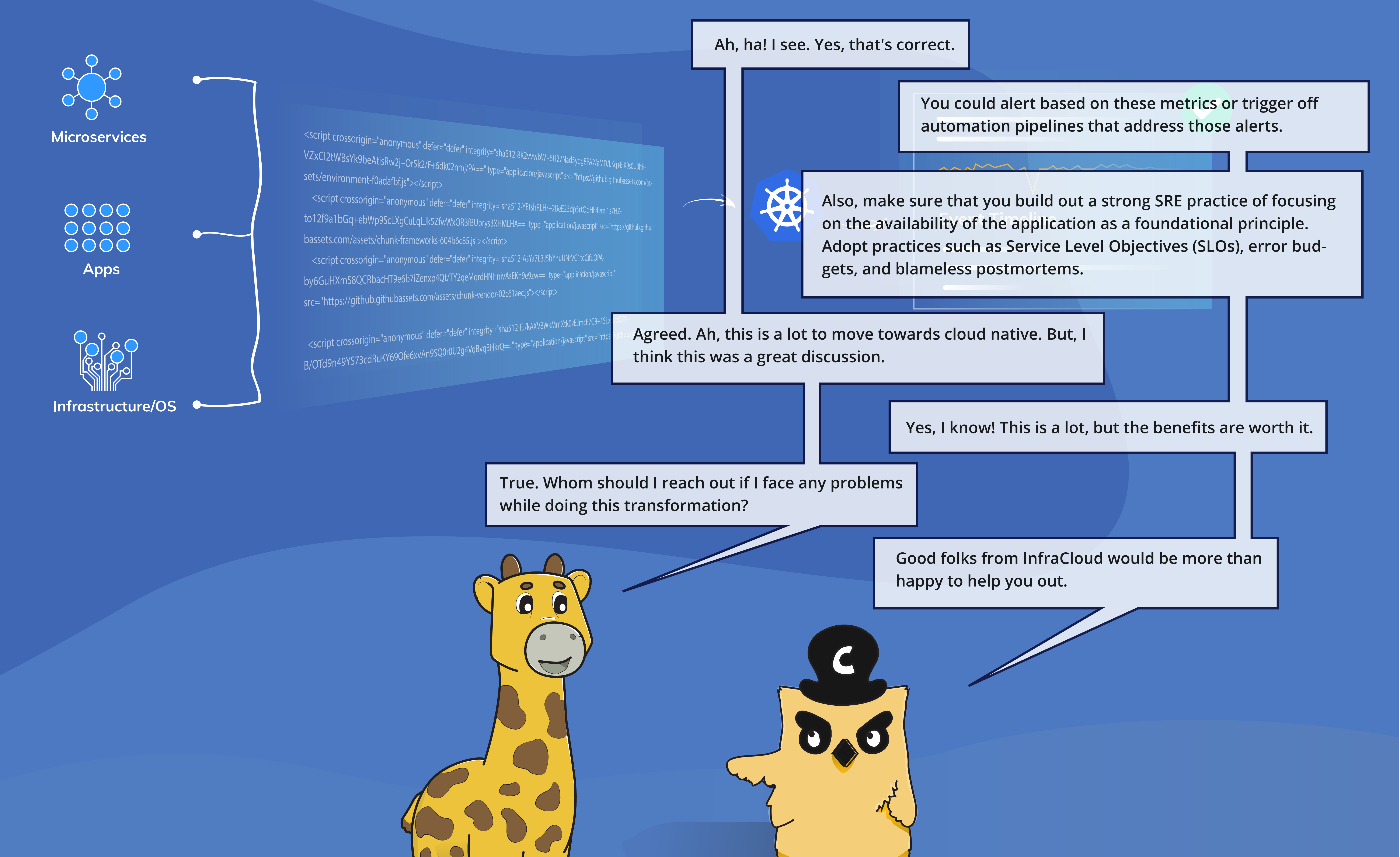
So how do we do it?
It's simple, so there is this concept of Pod and Cluster autoscaling in Kubernetes, where you can achieve this very easily. You can configure custom metrics as well.
Cool, that sounds interesting. Let me talk to the team, and then I would be able to decide how we can implement these best practices to become cloud native.
Yupp.
Capacity
Load
M T W T F S S



Microservices
Apps
Infrastructure/OS
Now that you have at least an idea in the movement towards cloud native, I think you should also make sure your application services are observable.
I see, go on.
Make sure you create these services emit metrics based on which you can scale them up/down.
But, wouldn't resource requests such as CPU and memory utilization help with that?
Yes, they would to a certain extent but say, e.g., emitting the number of messages piled up in the queue above a certain threshold can help you decide to scale out the receivers.
C

```
<script crossorigin="anonymous" defer="defer" integrity="sha512-8K2vvwbW+6H27Nad5ydg8PA2/aMD/LKq+EiK9s0U0hh-
VZxCI2tWBsYk9beAtisRw2j+Or5k2/F+6dk02nmj/PA==" type="application/javascript" src="https://github.githubassets.com/as-
sets/environment-f0adafbf.js"></script>
   <script crossorigin="anonymous" defer="defer" integrity="sha512-YEtshRLHr+2BeE23dp5rtQdHF4emi1s7HZ-
to12f9a1bGq+ebWp9ScLXgCuLqLJk5ZfwWxORBfBUprys3XHMLHA==" type="application/javascript" src="https://github.githu-
bassets.com/assets/chunk-frameworks-604b6c85.js"></script>
   <script crossorigin="anonymous" defer="defer" integrity="sha512-AsYa7L3J5bYnuUNrVC1tcCifuOPA-
by6GuHXmS8QCRbacHT9e6b7iZenxp4Qt/TY2qeMqrdHNHnivAsEKn9e9zw==" type="application/javascript"
src="https://github.githubassets.com/assets/chunk-vendor-02c61aec.js"></script>

  <script crossorigin="anonymous" defer="defer" integrity="sha512-FJ/kAXV8WkMmXtk0zEJmcF7C8+1SL
B/OTd9n49YS73cdRuKY69Ofe6xvAn9SQ0r0U2g4VqBvq3HkrQ==" type="application/javascript" src="http
```


Microservices
Apps
Infrastructure/OS
Ah, ha! I see. Yes, that's correct.
You could alert based on these metrics or trigger off automation pipelines that address those alerts.
Also, make sure that you build out a strong SRE practice of focusing on the availability of the application as a foundational principle. Adopt practices such as Service Level Objectives (SLOs), error budgets, and blameless postmortems.
Agreed. Ah, this is a lot to move towards cloud native. But, I think this was a great discussion.
Yes, I know! This is a lot, but the benefits are worth it.
True. Whom should I reach out if I face any problems while doing this transformation?
Good folks from InfraCloud would be more than happy to help you out.

```
<script crossorigin="anonymous" defer="defer" integrity="sha512-8K2vvwbW+6H27Nad5ydg8PA2/aMD/LKq+EiK9s0U0hh-
VZxCI2tWBsYk9beAtisRw2j+Or5k2/F+6dk02nmj/PA==" type="application/javascript" src="https://github.githubassets.com/as-
sets/environment-f0adafbf.js"></script>
  <script crossorigin="anonymous" defer="defer" integrity="sha512-YEtshRLHr+2BeE23dp5rtQdHF4emi1s7HZ-
to12f9a1bGq+ebWp9ScLXgCuLqLJk5ZfwWxORBfBUprys3XHMLHA==" type="application/javascript" src="https://github.githu-
bassets.com/assets/chunk-frameworks-604b6c85.js"></script>
  <script crossorigin="anonymous" defer="defer" integrity="sha512-AsYa7L3J5bYnuUNrVC1tcCifuOPA-
by6GuHXmS8QCRbacHT9e6b7iZenxp4Qt/TY2qeMqrdHNHnivAsEKn9e9zw==" type="application/javascript"
src="https://github.githubassets.com/assets/chunk-vendor-02c61aec.js"></script>

 <script crossorigin="anonymous" defer="defer" integrity="sha512-FJ/kAXV8WkMmXtk0zEJmcF7C8+1SLz
B/OTd9n49YS73cdRuKY69Ofe6xvAn9SQ0r0U2g4VqBvq3HkrQ==" type="application/javascript" src="https
```

INFRACLOUD

Begin your cloud native journey with InfraCloud

Interested in diving into what Phippy did and learning more about the cloud native transformation?

Talk to an Expert