Instructions: This zine is meant for you *to fill in your own art!*
There are prompts peppered but this zine is for you *to draw in!*
If you use Twitter, please share your sketches with
#serverlesszine so other people can see them!

So you've uploaded your code to AWS Lambda, and you're ready to try this "serverless" thing, but how does your code actually run?

Draw your code uploading to the ~cloud~

I hope you've learned a bit about how serverless runs! Use the rest of this page to draw out your next serverless architecture :)

How does code run in a “serverless” environment?

Once that trigger is called, your code runs!

Draw an event trigger, or your code taking off at speeeeed!

When it can, the platform (your provider) will re-use an existing container which can make running these functions really *fast*. Your code is primed to run!

Your code runs, succeeds fails, boom! You're good to go! If it failed, your function will re-try based on the settings you configured for it.

** Note that AWS Lambda guarantees "at least once" execution, so keep that in mind in how you handle events!**

If all the containers are busy, and you haven't run into your concurrency limit, or how many containers you're allowing the function to take up when it's running, your function will spin up a new container.

Draw a container spinning up!

When it's loaded into a container for the first time, it's called a **cold start** because the container is **cold** – it's just seeing your dependencies and all that for the first time!

Draw what a cold start feels like for you!

By default on AWS, your concurrency limit is the sum of *all* your functions running. But once you see how easy it is to deploy a service with just a function's worth of compute, it's not the best idea to trust your functions to share nicely!

Draw your functions arguing over containers!

You can set the concurrency limit *per function* to make sure everyone plays nice.

Draw your functions getting along happily :)

Now that you've uploaded, your code is now sitting in storage and ready to run when one of the triggers you define is called.

Draw an event triggering your code!

Triggers can come from many sources you can configure, and these will be different on different clouds (because different clouds offer different services).

You might trigger a function on an HTTP call, or when an object is added to storage, or when there's a new item in a queue. It's all up to you!