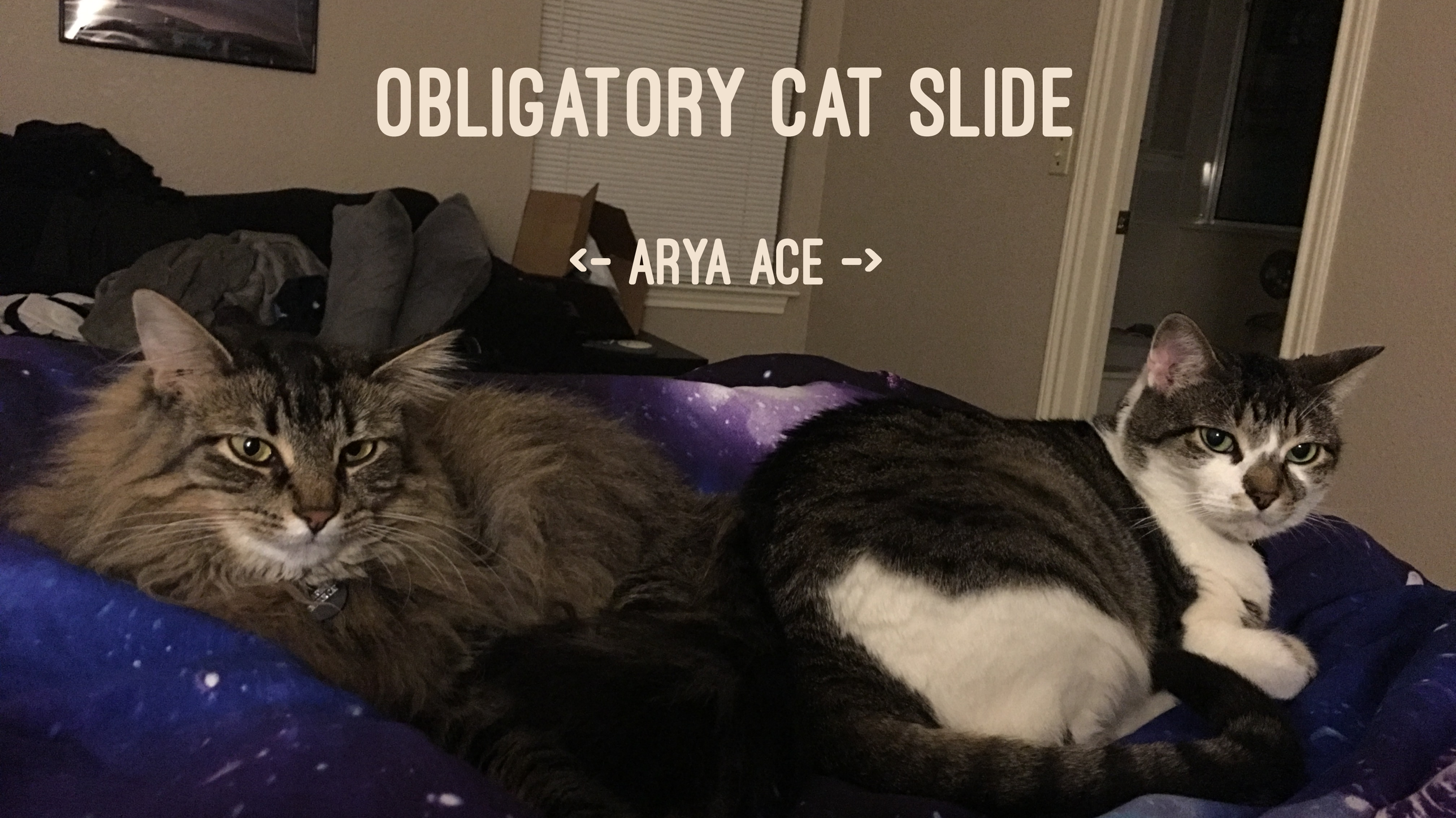
SERVERLESS ARCHITECTURE IN NODE.JS

OBLIGATORY ABOUT ME

- "THE NODEBOTANIST"
- ATX DWELLER
- CATPARENT
- ELECTRICAL ENGINEERING STUDENT AT ASU ONLINE
- DEVELOPER RELATIONS ENGINEER AT IOPIPE

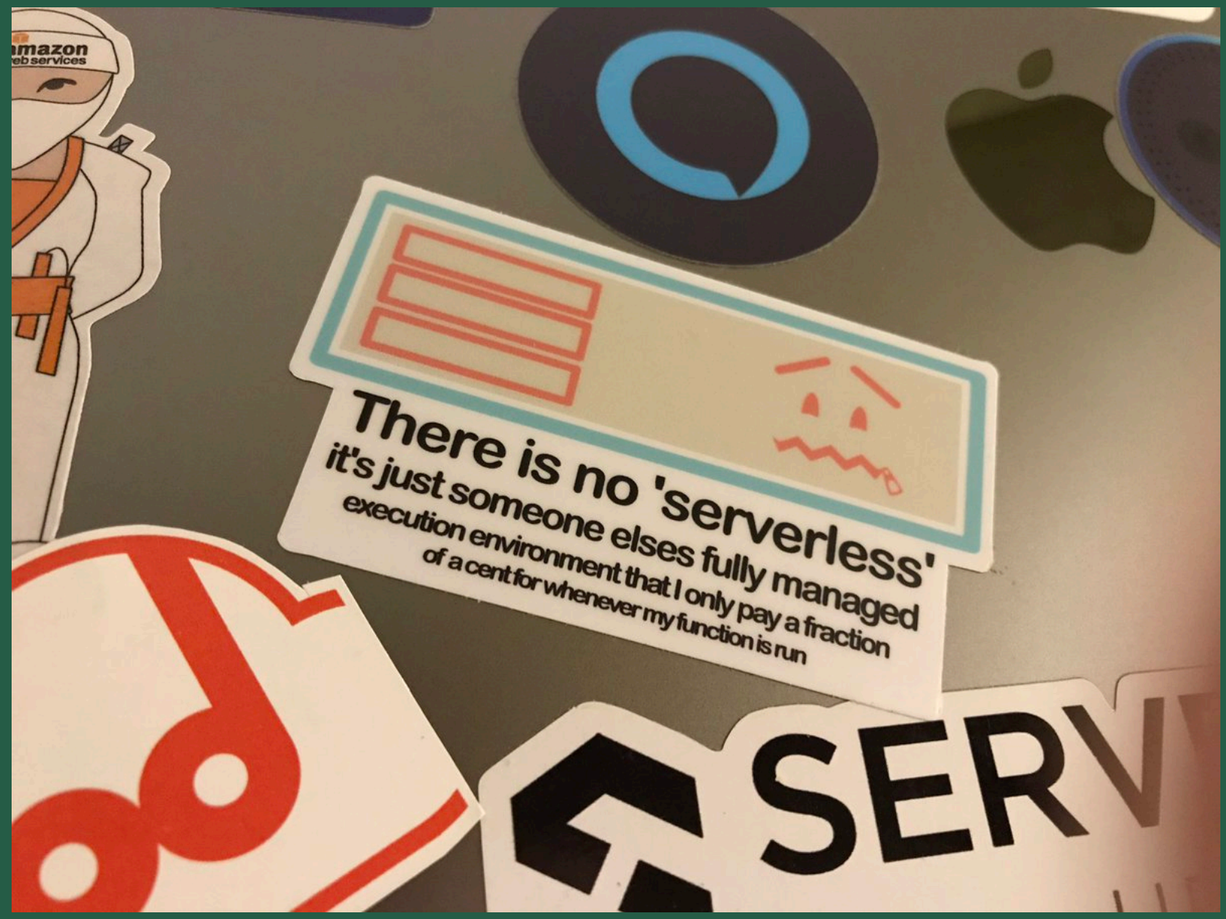
OBLIGATORY CAT SLIDE

<- ARYA ACE ->



WHAT IS SERVERLESS?

IT DOESN'T LITERALLY MEAN NO SERVERS



IT'S THE NEXT STEP FROM IAAS

- ALSO CALLED FUNCTIONS AS A SERVICE (FAAS)
- EVERYTHING FROM INFRASTRUCTURE AS A SERVICE IS ABSTRACTED AWAY PLUS LANGUAGE RUNTIME, HTTP SERVING (IN MOST CASES)
- ALL YOU NEED IS THE ACTUAL CODE FOR THE FUNCTION YOU WANT TO EXECUTE

EXAMPLE

- YOU HAVE A NODE.JS FUNCTION THAT SCRAPES A WEB SITE FOR SALES ON SOCKS
- YOU WANT TO RUN IT REPEATEDLY
- YOU DON'T WANT TO SPIN UP A NEW SERVER AND ENVIRONMENT JUST TO RUN THIS ONE THING!

EXAMPLE (CONT.)

- YOU DON'T CARE WHAT OS IT RUNS ON, WHERE IT IS, OR EVEN WHAT NODE.JS RUNTIME (AS LONG AS ITS A CERTAIN VERSION OR HIGHER)
- YOU SET UP A SERVERLESS FUNCTION TO RUN ON A SCHEDULE AND BAM!

WHO PROVIDES SERVERLESS ARCHITECTURES?

TO NAME A FEW

- AWS LAMBDA
- MICROSOFT AZURE FUNCTIONS
- WEBTASK.IO
- APACHE OPENWHISK

YEAH. OKAY. BUT WHY NODE.JS?

THAT'S A REALLY GOOD QUESTION!

THIS QUESTION IS ONE WITH MANY ANSWERS (AND MANY ONGOING SURROUNDING ARGUMENTS)!

WHY I BELIEVE NODE.JS IS UBIQUITOUS IN SERVERLESS

- IT'S ON THE WEB
- MANY SERVERLESS FUNCTIONS RELY ON ASYNCHRONOUS TASKS
- IT'S POPULAR (I'M NOT GOING TO ARGUE THE SHOULD IT BE. JUST STATING A FACT HERE)
- A LOT OF COINCIDENCE AND HAPPENSTANCE AND "WELL WE ALREADY DID IT THIS WAY"

NODE.JS ISN'T THE ONLY ANSWER!

- PYTHON, .NET
- AWS ALLOWS ANY EXECUTABLE*

*-- ANY EXEC THAT RUNS ON THEIR CONTAINER AND A FEW OTHER CAVEATS

WHY WOULDN'T YOU WANT NODE.JS?

- YOUR TEAM DOESN'T KNOW IT YET
- YOUR TASKS ARE NOT ASYNC-HEAVY
- YOU JUST PLAIN DON'T WANT TO WRITE NODE.JS (CONTRARY TO POPULAR INTERNET BELIEF. YOU CAN BE A WEB DEV AND NOT LIKE NODE.JS)

HOW DOES A SERVERLESS ARCHITECTURE WORK?

INSTEAD OF BUILDING A REST API. YOU BUILD A SET OF INTEROPERATING FUNCTIONS THAT OPERATE ON RESOURCES AND EVENTUALLY THIS BECOMES YOUR APPLICATION'S BACKEND

WAAAAAAAIT...ISN'T THIS BASICALLY MICROSERVICES?

...NOT TECHNICALLY. HOWEVER, THINKING IN TERMS OF MICROSERVICES CAN REALLY HELP YOU BUILD OUT A SERVERLESS ARCHITECTURE

LET'S TAKE A LOOK AT BUILDING AWS LAMBDA FUNCTIONS WITH THE SERVERLESS FRAMEWORK

AWESOME PARTS OF SERVERLESS ARCHITECTURE

- SCALABILITY
- QUICK DEVELOPMENT
- VENDOR TOOLING IS DEVELOPING QUICKLY!

NOT-SO-AWESOME PARTS

- DEBUGGING CAN BE TRICKY
- SO CAN MONITORING (BELIEVE ME!)
- THE TOOLING IS STILL DEVELOPING!

WANNA LEARN MORE? CHECK OUT:

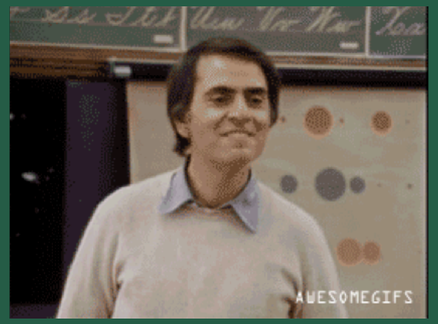
- THE REPO FOR THIS TALK (HAS SLIDES, CODE) : GITHUB.COM/NODEBOTANIST/2000K-2017
- THE SERVERLESS FRAMEWORK AT SERVERLESS.COM
- THE COMMUNITY SERVERLESS PODCAST SERVERLESSPODCAST.COM
- THE SERVERLESS FORUM SLACK CHANNEL

A BRIEF ASIDE

YOU SHOULD BE UP HERE. TOO

- YES. YOU.
- YOU HAVE INTERESTING EXPERIENCES AND THINGS TO SAY
- APPLY TO THUNDERPLAINS AND OTHER GREAT CONFERENCES
- GET IN TOUCH- I'M HAPPY TO REVIEW TALK PROPOSALS. SLIDES. ETC.

THANKS FOR LISTENING!



- THE@NODEBOTANI.ST OR KAS@IOPIPE.COM
- @NODEBOTANIST PRETTY MUCH EVERYWHERE
- COME FIND ME AT LUNCH OR THE SOCIAL-- I'M IN THE BACK HACKING ON HARDWARE!