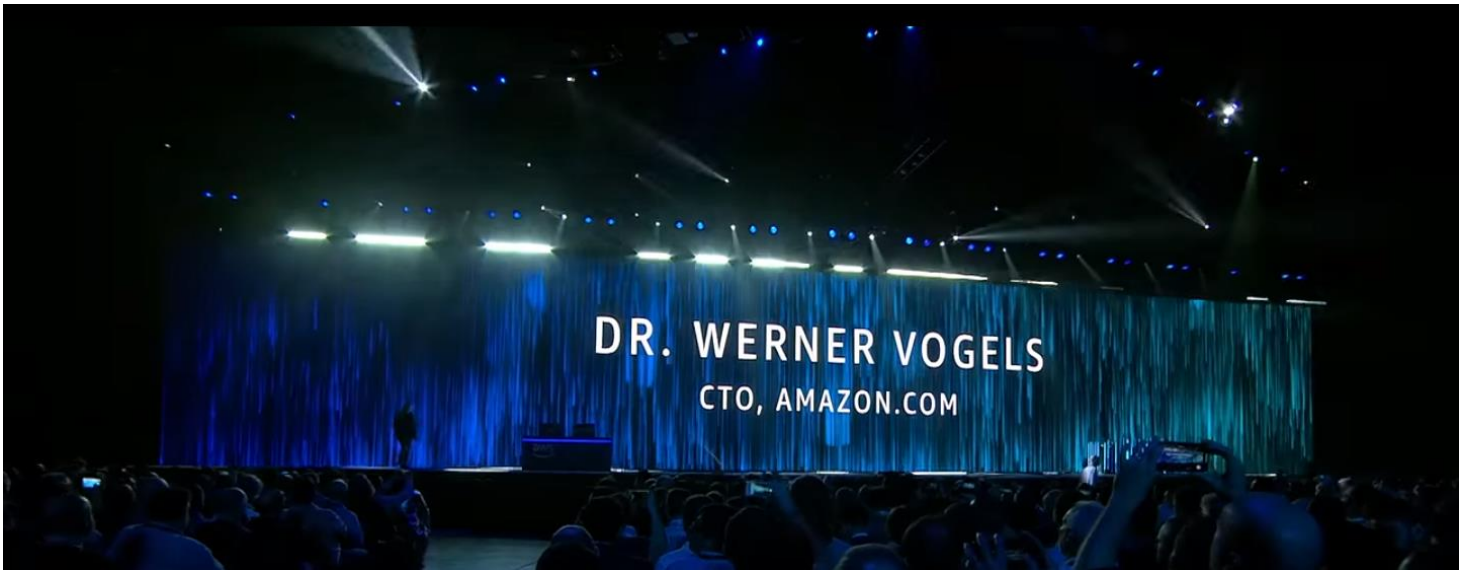
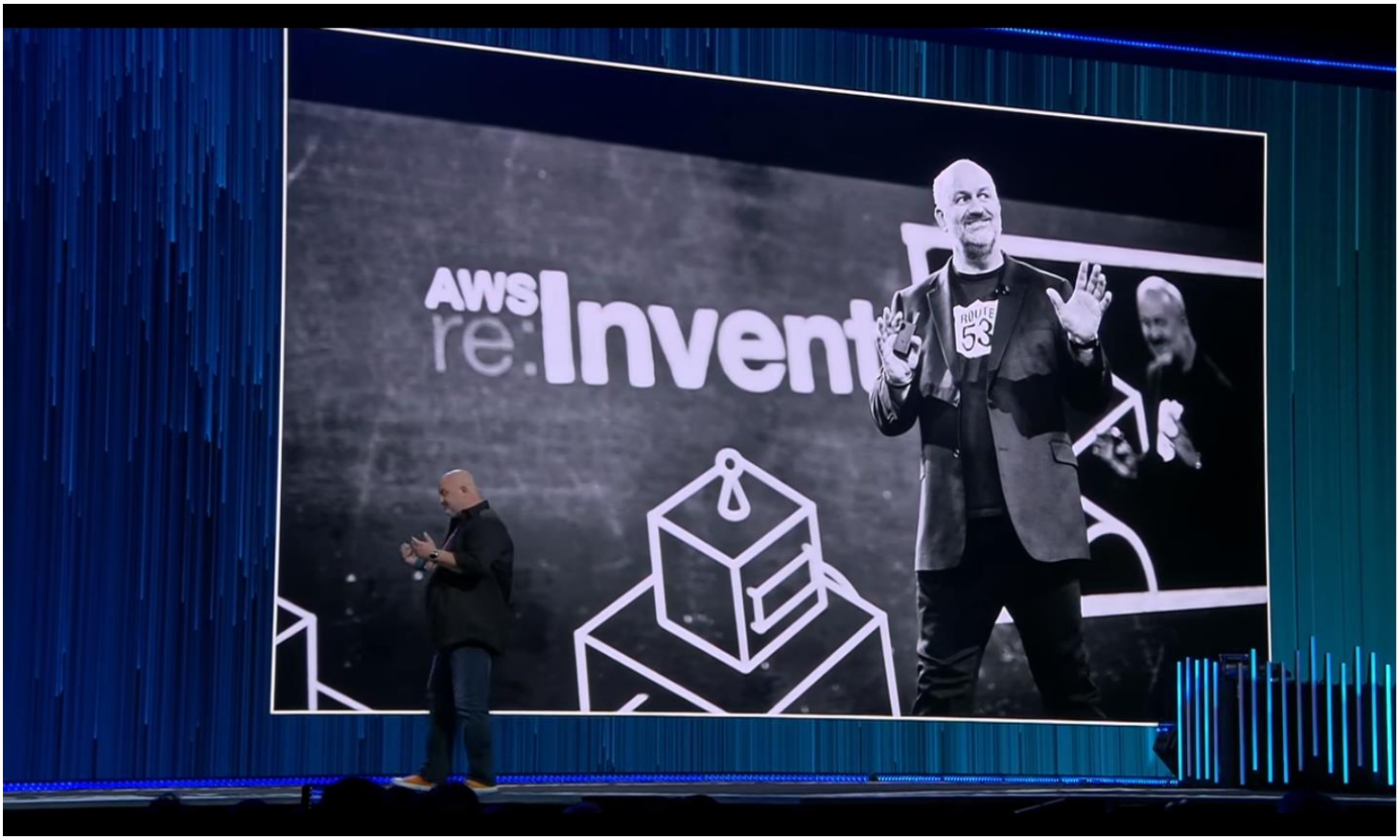

DR. WERNER VOGELS
CTO, AMAZON.COM

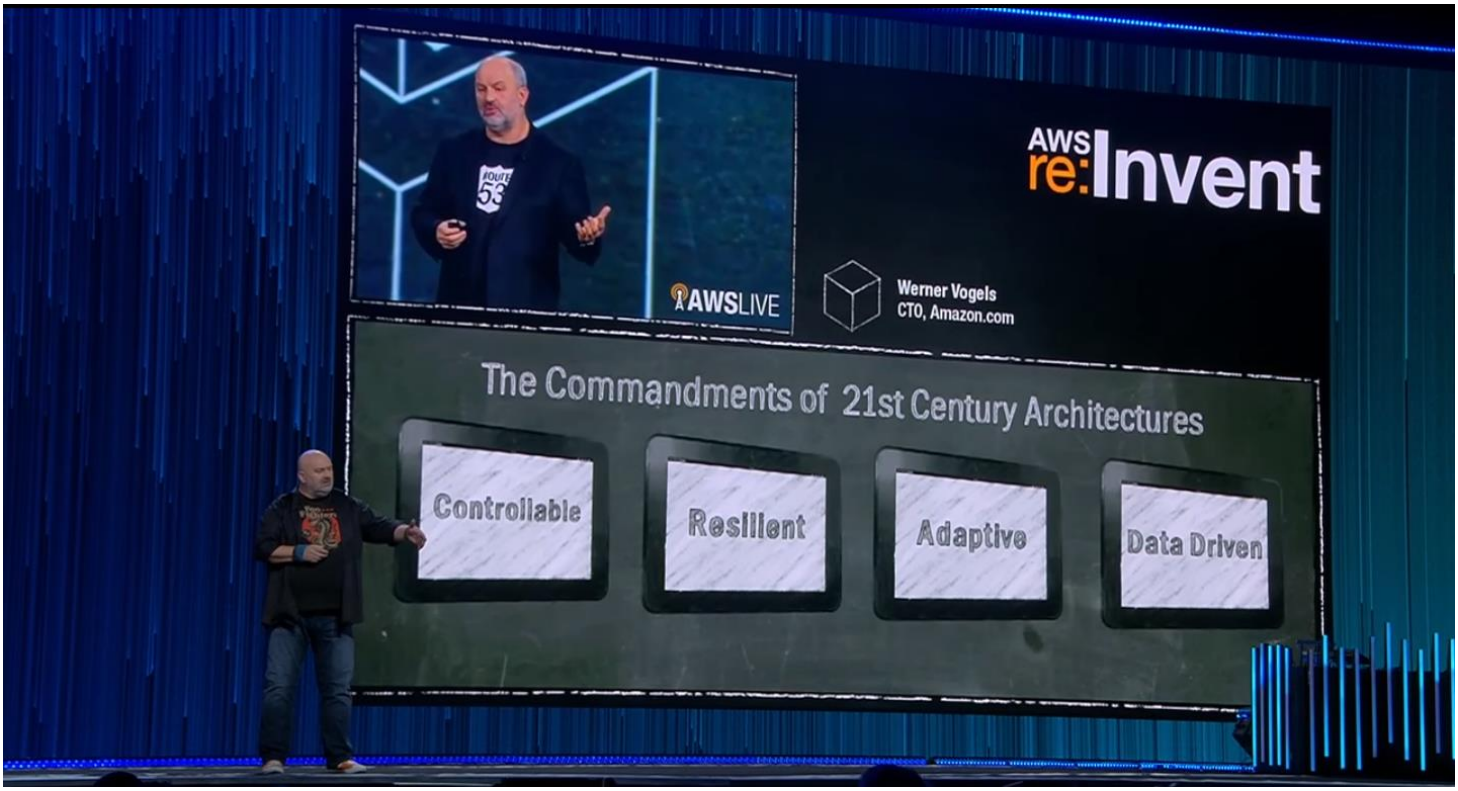

Foo Fighters
DR. WERNER VOGELS
CTO, AMAZON.COM


AWS
re:Invent

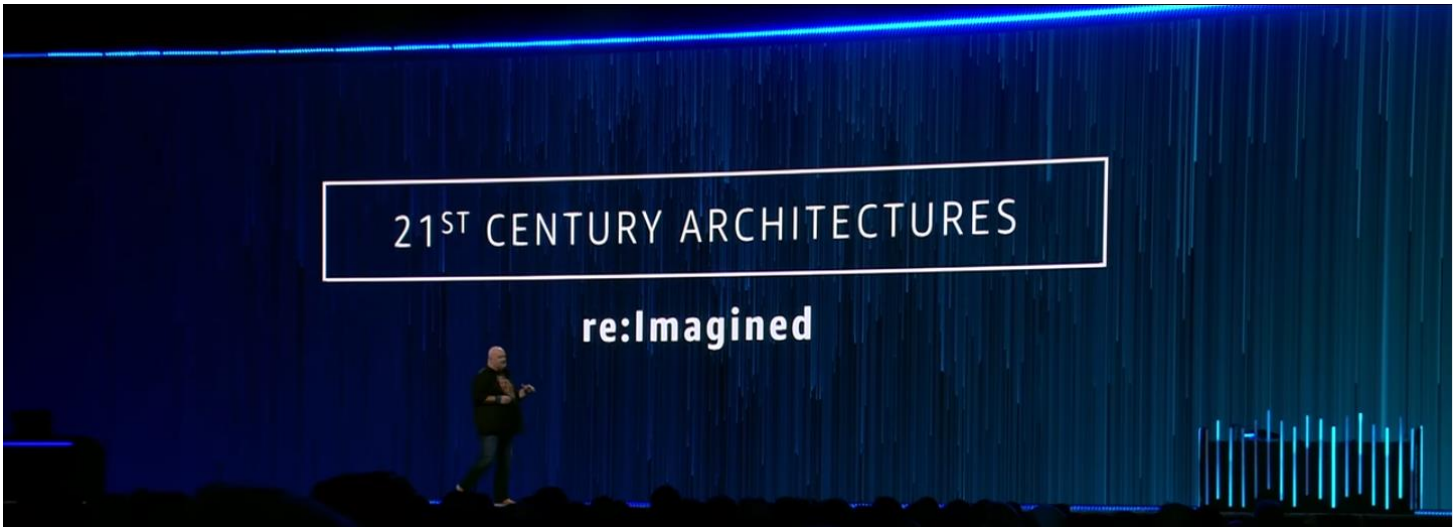

21ST CENTURY
ARCHITECTURES
aws
re:Invent

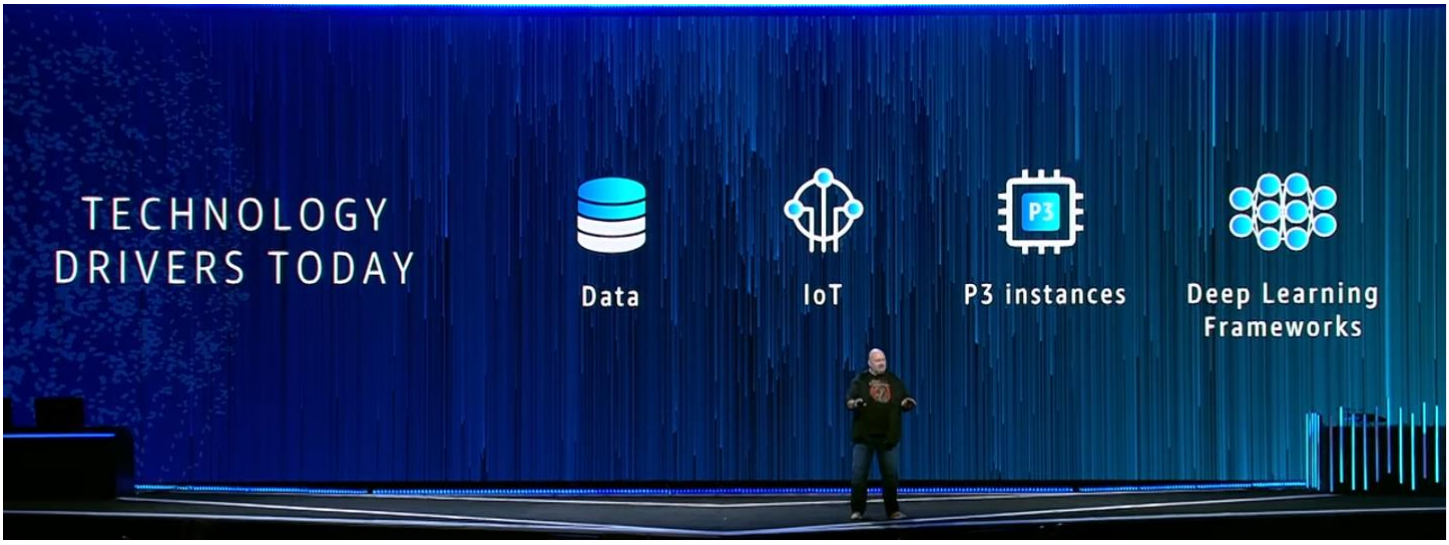

AWS
re:Invent
ROUTE
53


21ST CENTURY ARCHITECTURES
AWS re:Invent
AWSLIVE
Werner Vogels
CTO, Amazon.com
The Commandments of 21st Century Architectures
Controllable
Resilient
Adaptive
Data Driven
aws


AWS re:Invent
AWSLIVE
Werner Vogels
CTO, Amazon.com
The Commandments of 21st Century Architectures
Controllable
Resilient
Adaptive
Data Driven

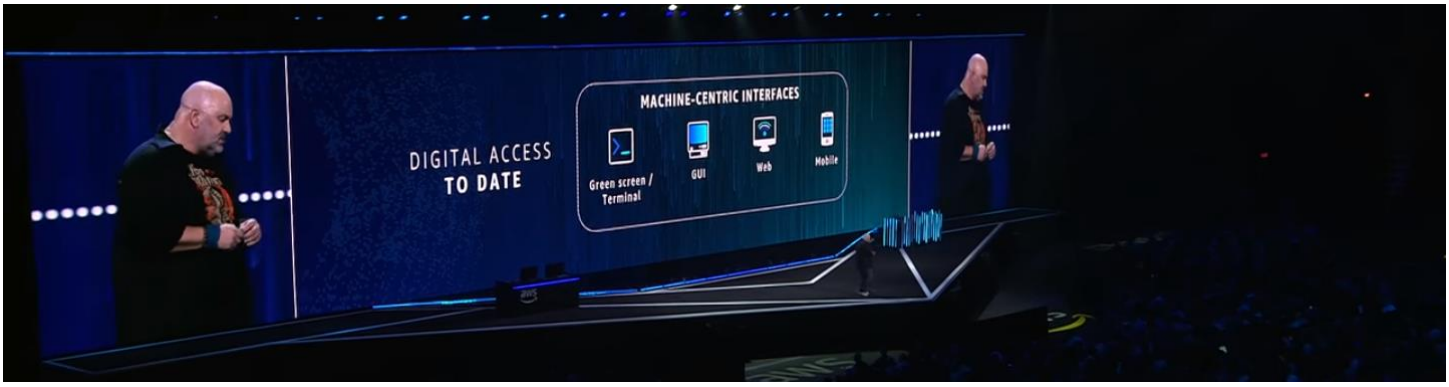

3,951 NEW FEATURES AND SERVICES

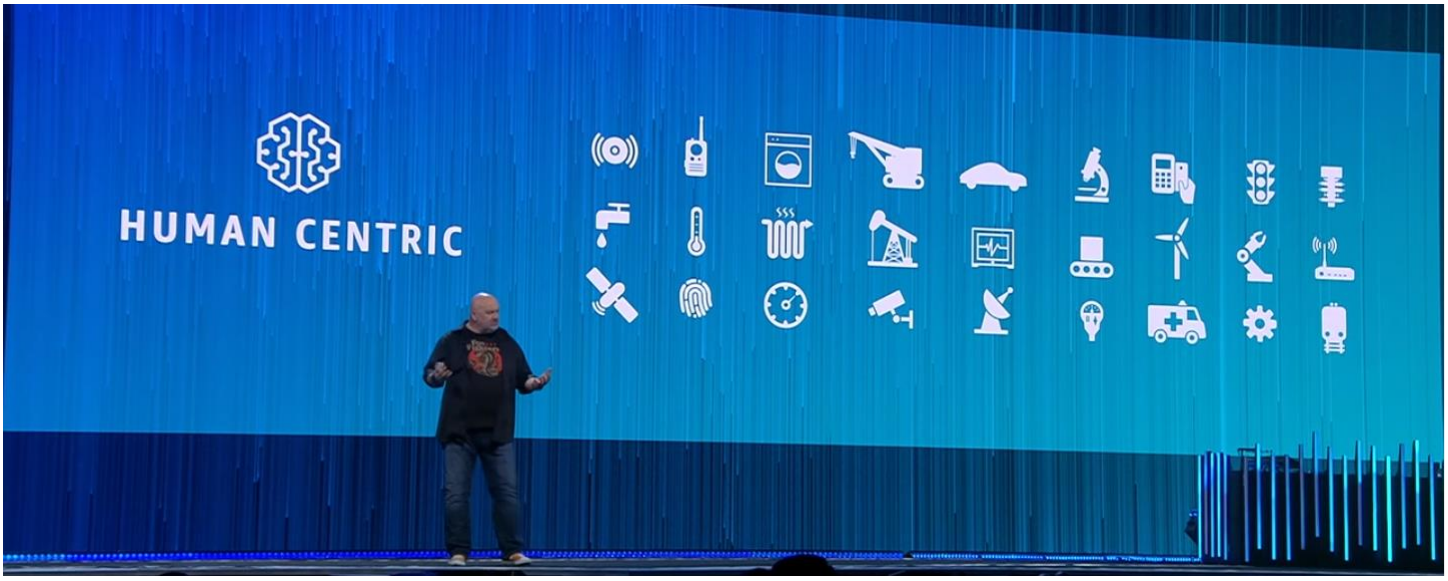




We are now able to execute neural networks in real-time.


DIGITAL ACCESS
TO DATE
Green screen / Terminal
GUI
Web
Mobile


DIGITAL ACCESS
TO DATE
MACHINE-CENTRIC INTERFACES
Green screen / Terminal
GUI
Web
Mobile

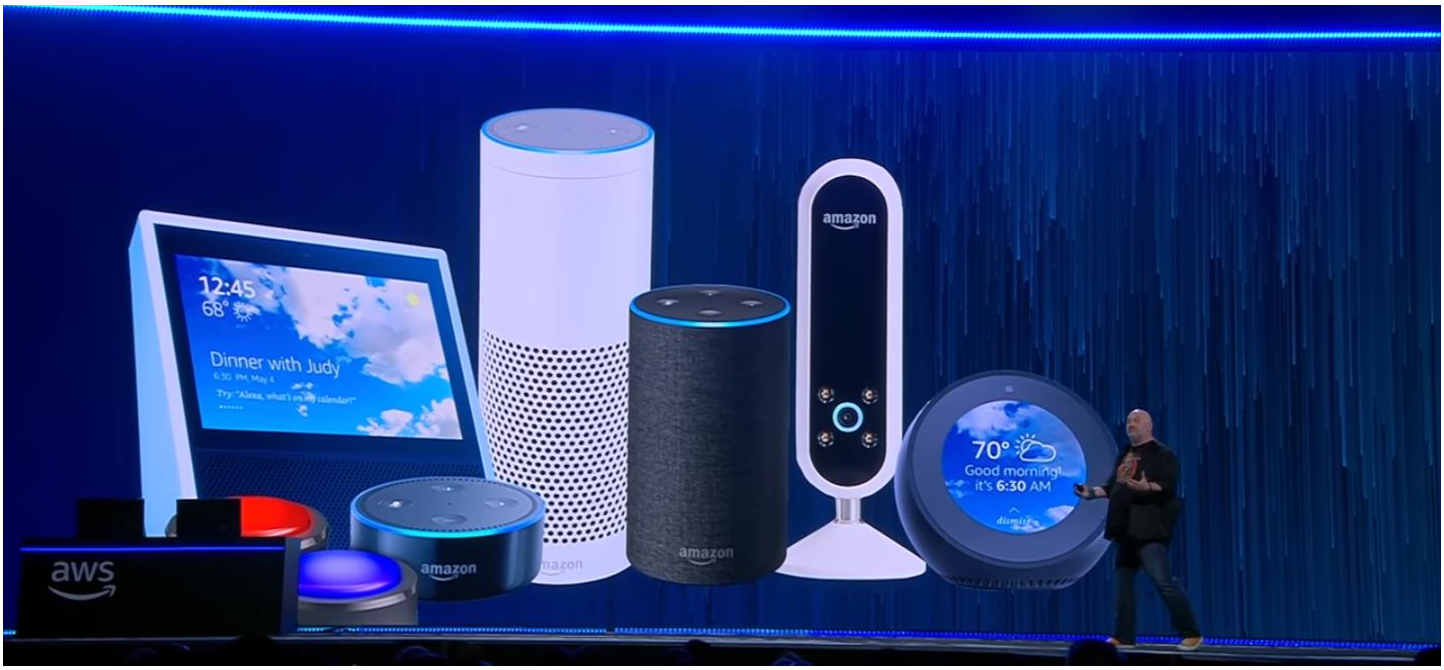

HUMAN CENTRIC

The first step to this is going to be via Voice, voice is the most natural interface for interacting in the world. We need to get this into our digital systems of the future.

The surgeons in the operating room should be able to use voice to interact with the machines around instead of having to take their hands off the patient. A sick father wants to talk to a device in natural language to get information about how to help his young child.

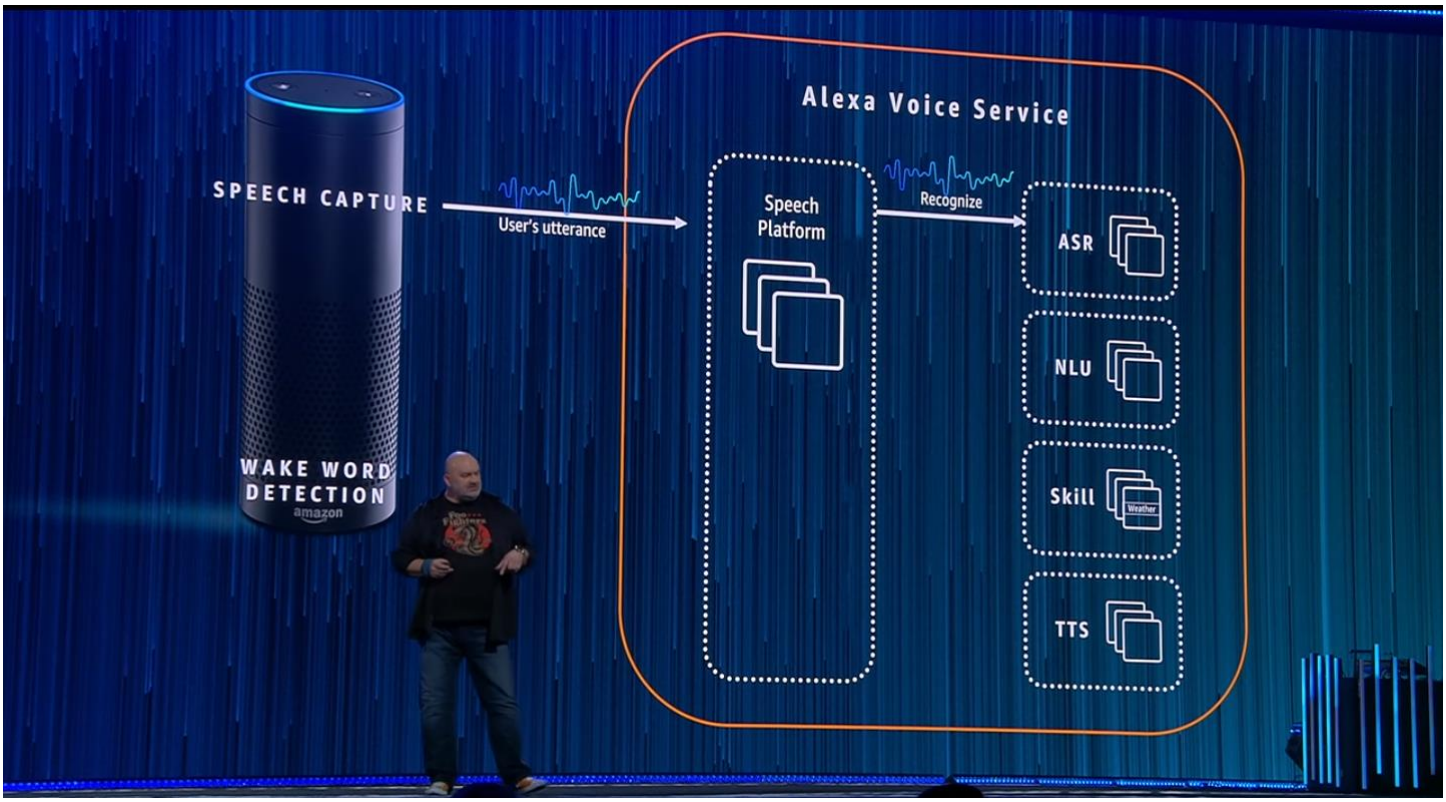
We needed to start building Voice Driven Systems.

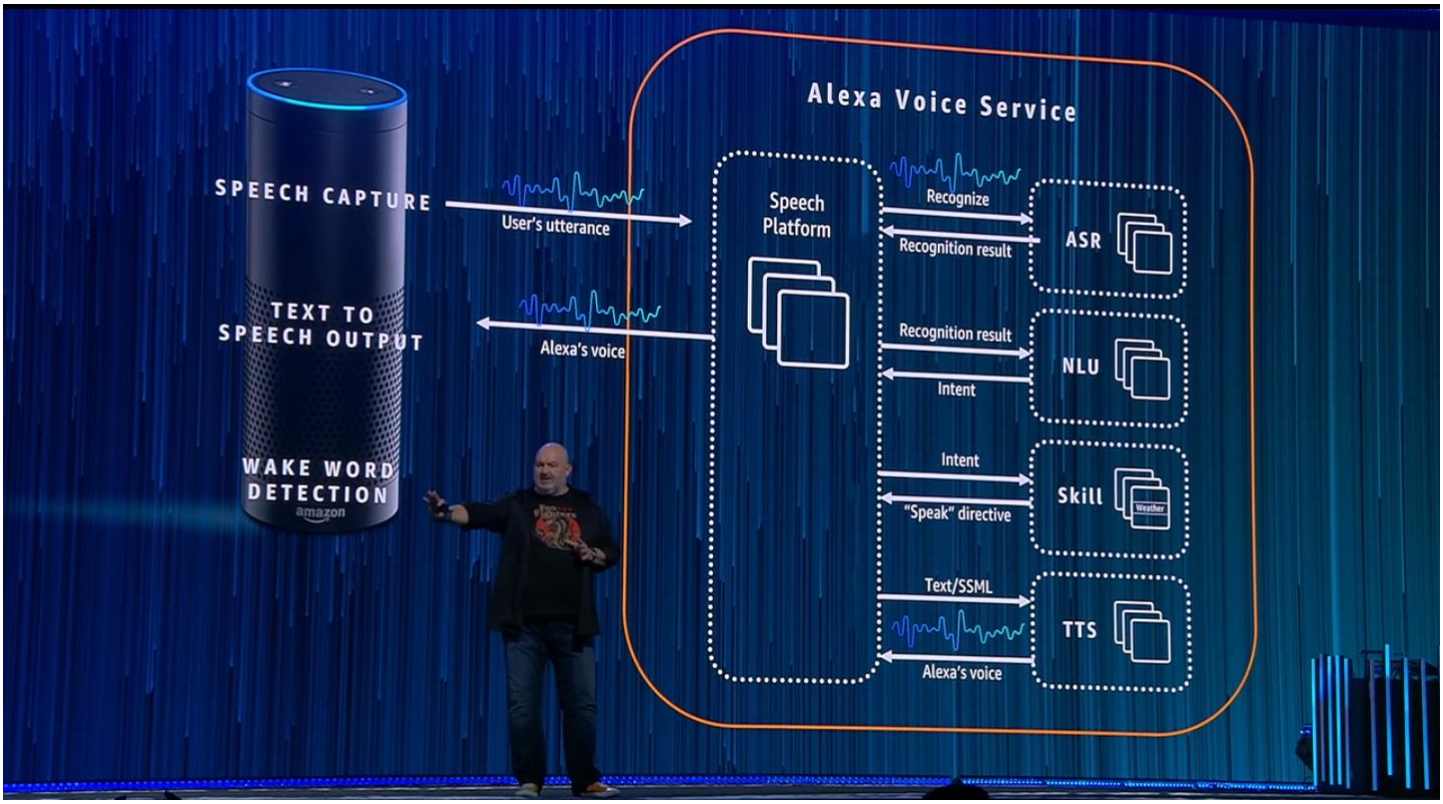
They put a voice interface to a digital system that a farmer can describe their plot of land and the machine learning system can query the digital system for information about what type and quantity of fertilizer to apply



All the smarts of these devices live in the cloud as the Alexa Voice Service and the Alexa Skills Kit.

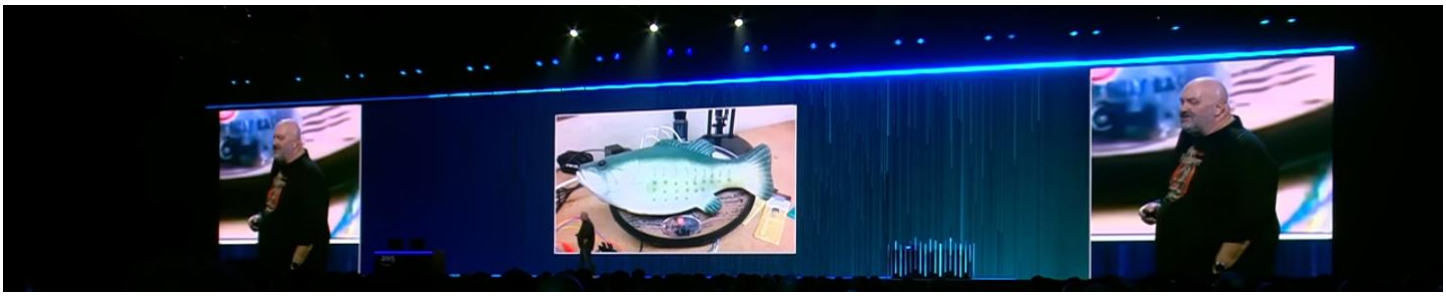
The wake word is ‘Alexa’, then the device starts recording and listening after that.

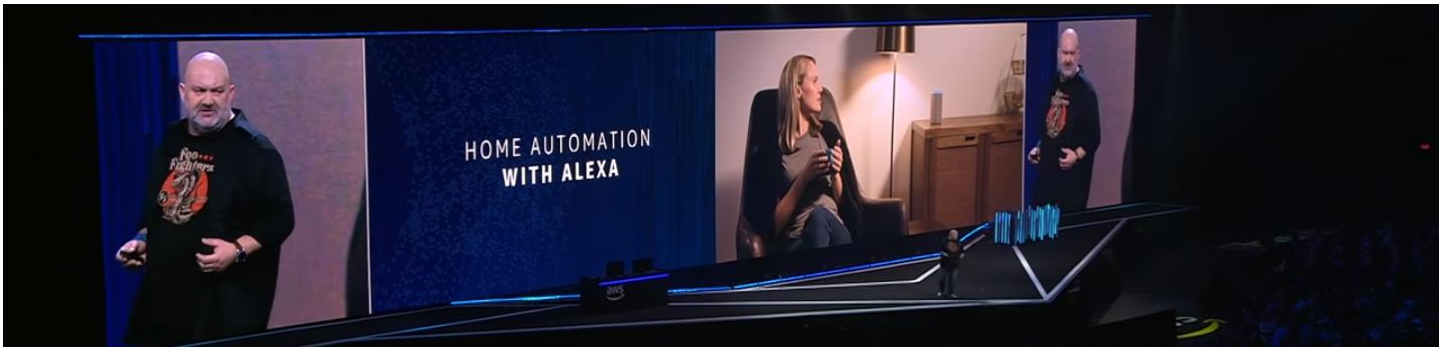




TTS creates the audio file that gets sent back to the device, which is then read out to the user. This is a pure cloud service. In AWS, Text-To-Speech (TTS) is the ***Polly service***, the Natural Language Understanding (NLU) is the ***Lex service***, you can build your own Alexa system if you want.

Anybody can integrate the Alexa Voice Service into their devices,



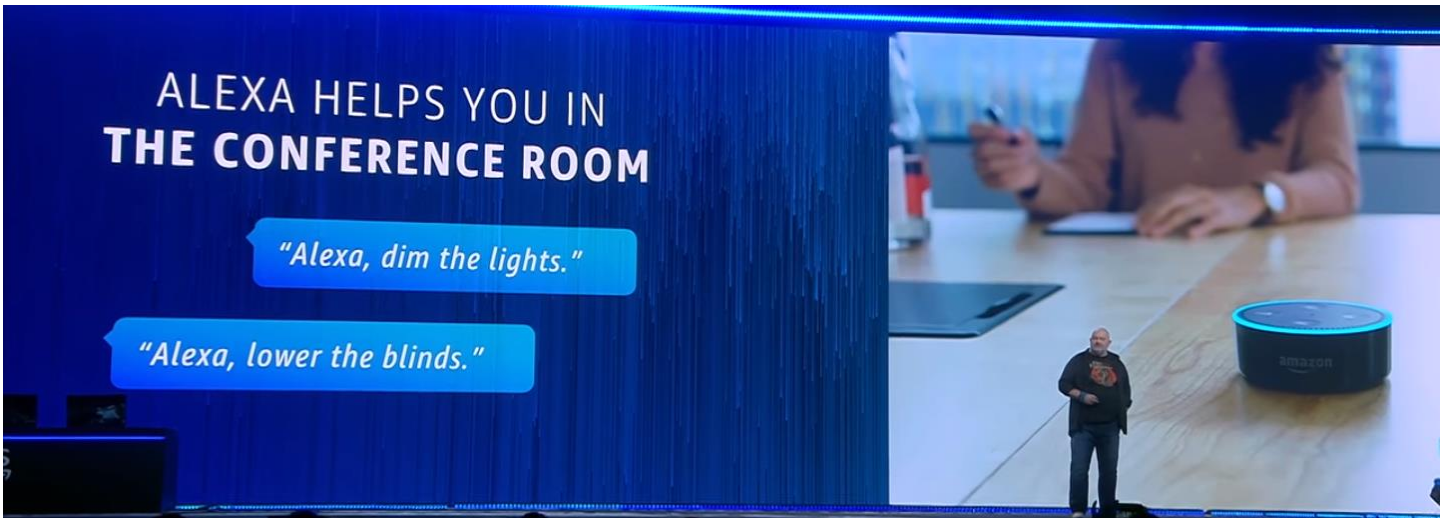

HOME AUTOMATION
WITH ALEXA


DEVICES IN THE CONFERENCE ROOM


MONTH-ENDS
DEVICES ON YOUR DESK


NEW!
Alexa for Business
Your intelligent assistant for work
Generally available today


ALEXA HELPS YOU IN
THE CONFERENCE ROOM
"Alexa, start the meeting."
"Alexa, dial 555-8000."
aws


ALEXA HELPS YOU IN
THE CONFERENCE ROOM
"Alexa, dim the lights."
"Alexa, lower the blinds."


ALEXA HELPS YOU IN
THE CONFERENCE ROOM
"Alexa, ask Teem to extend the meeting."
"Alexa, ask Teem to find an open conference room."
teem


ALEXA HELPS YOU
AT YOUR DESK
"Alexa, join my meeting."
"Alexa, call Jeff."


ALEXA HELPS YOU
AT YOUR DESK
"Alexa, when is my next meeting?"
"Alexa, cancel my staff meeting."
"Alexa, schedule a meeting with Jeff."
MON OCT 23
Weekly Business Review
10am - 11am
One on One with Kevin
11am-11:30am
Lunch with Sadie
12pm-1pm


ALEXA HELPS YOU
AT YOUR DESK
"Alexa, ask Ring Central to read my voicemail."
RingCentral


ALEXA HELPS YOU
AT YOUR DESK
"Alexa, ask Salesforce which big deals closed today."
salesforce

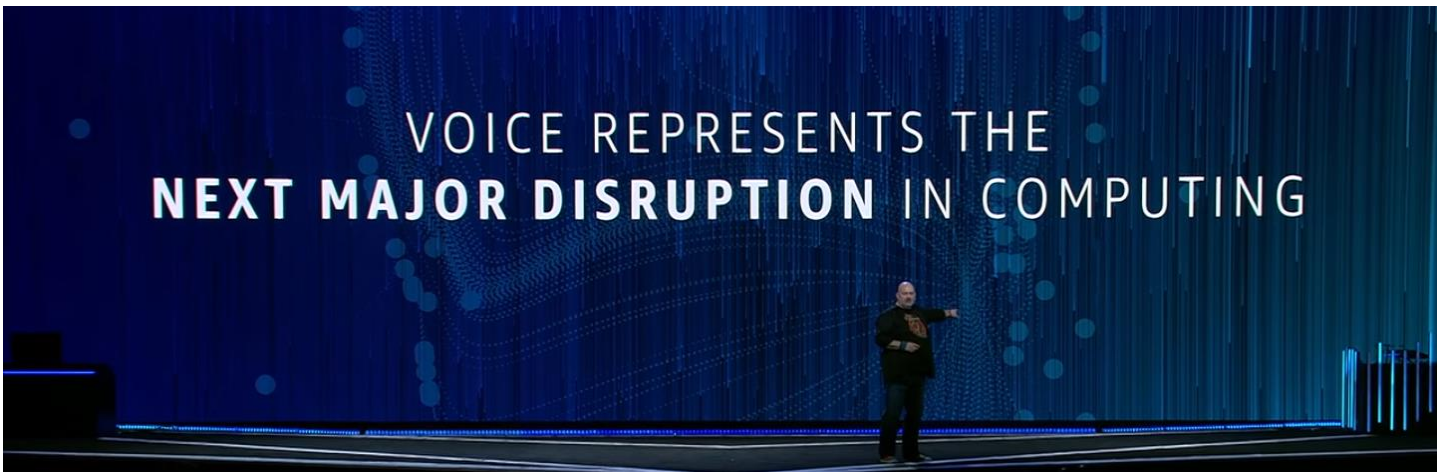

ALEXA HELPS YOU
AT YOUR DESK
"Alexa, ask Concur, 'what time does my flight leave?'"
CONCUR


ALEXA HELPS YOU
AT YOUR DESK
"Alexa, ask SuccessFactors to file PTO for February 14."
SAP SuccessFactors


ALEXA HELPS YOU
AT YOUR DESK
"Alexa, ask Splunk, 'how many Amazon EC2 instances are running in South America?'"
splunk>


ALEXA HELPS YOU
AT YOUR DESK
"Alexa, ask Acumatica, 'how many laptops do we have in stock?'"
Acumatica
The Cloud ERP
aws

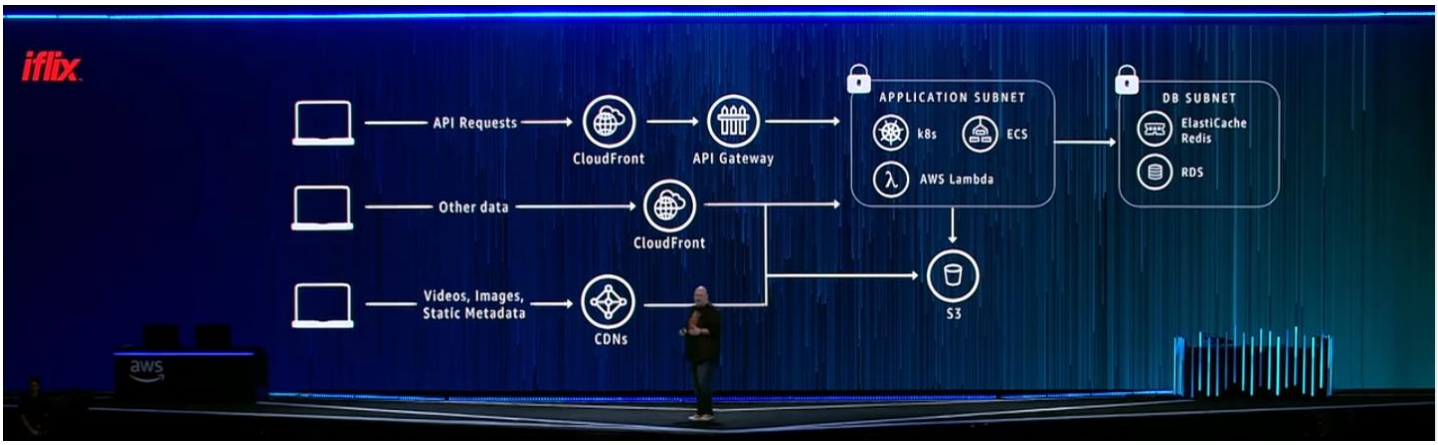
The ***wynn hotel*** is now putting an echo in each of their rooms so that guests can ask questions and get answers easily.



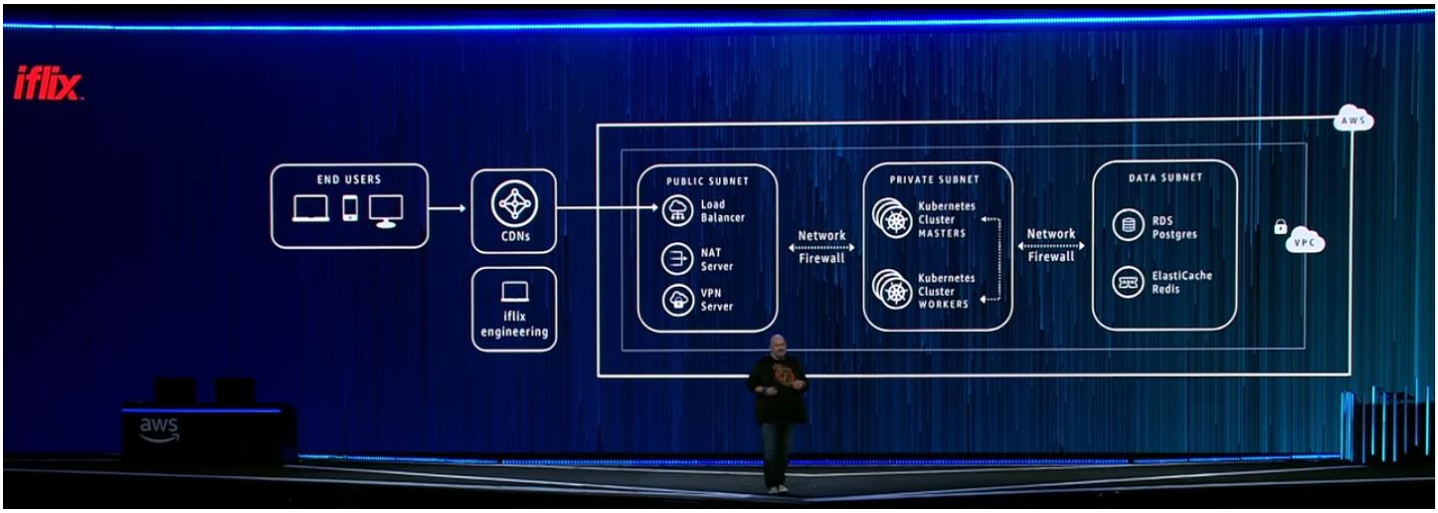
It will also mean that you are going to build your backend systems differently, the next generation of systems will be built using ***conversational interfaces*** because this is our systems will be winners in future.

21st century architectures WILL have natural interfaces for voice, there are 3 different planes that systems developers need to be aware of

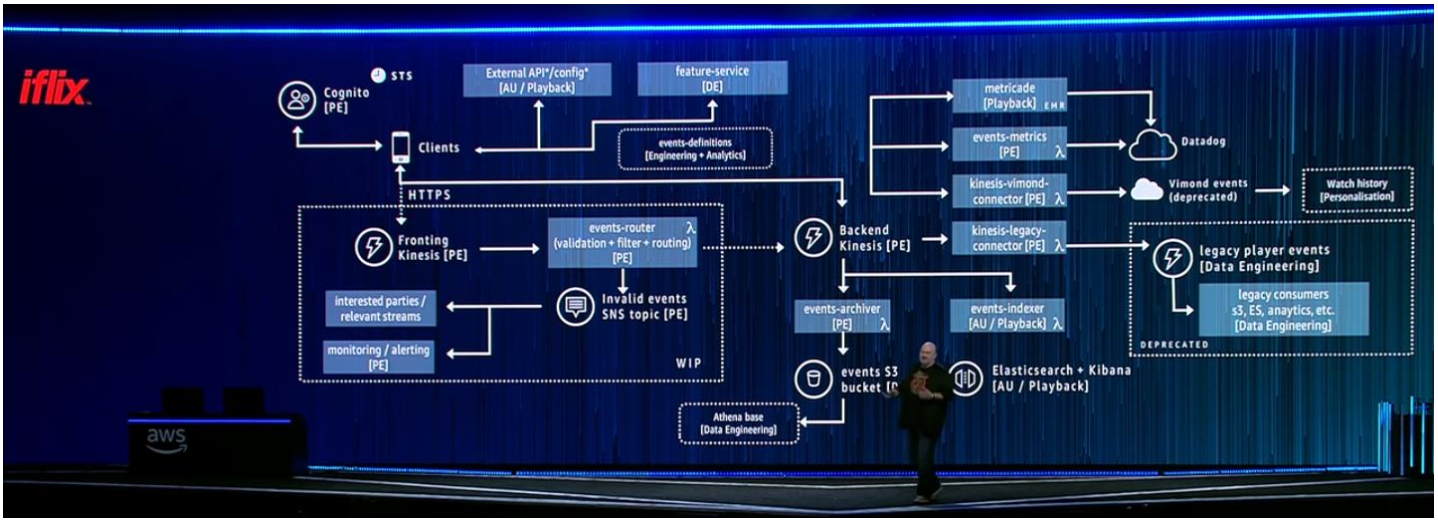
The Admin plane includes your deployment cycles, how you deploy, manage your containers and functions. The Control plane manages your deployed resources, the Data plane gives you access to your resources. There are different reliability and security requirements for these 3 planes.



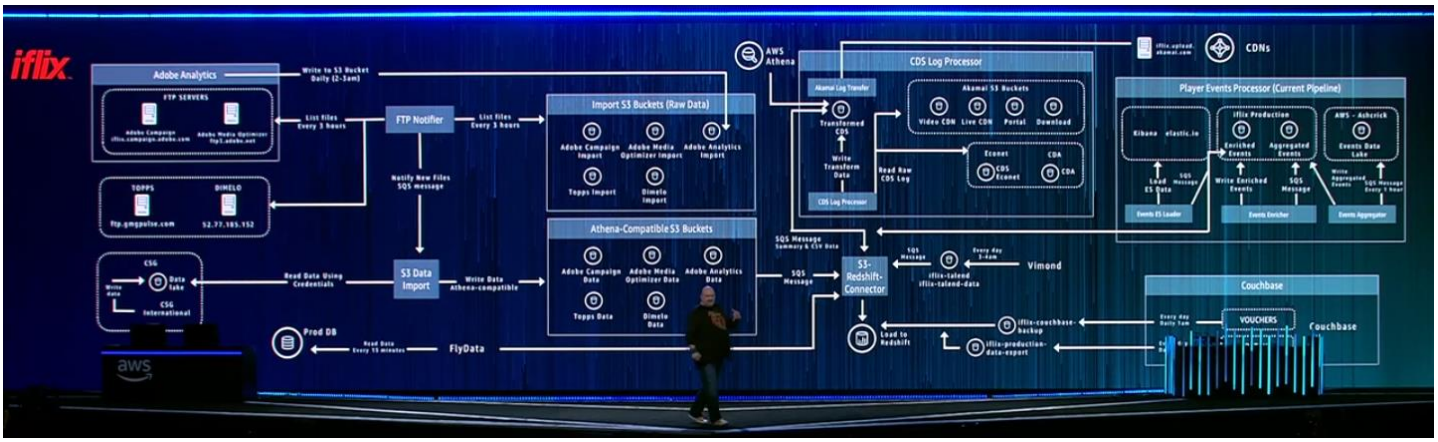
Iflix is a video streaming company in EMEA, the typical technology architecture for a video streaming company is shown above but this is not all of it. Let's see what the rest looks like



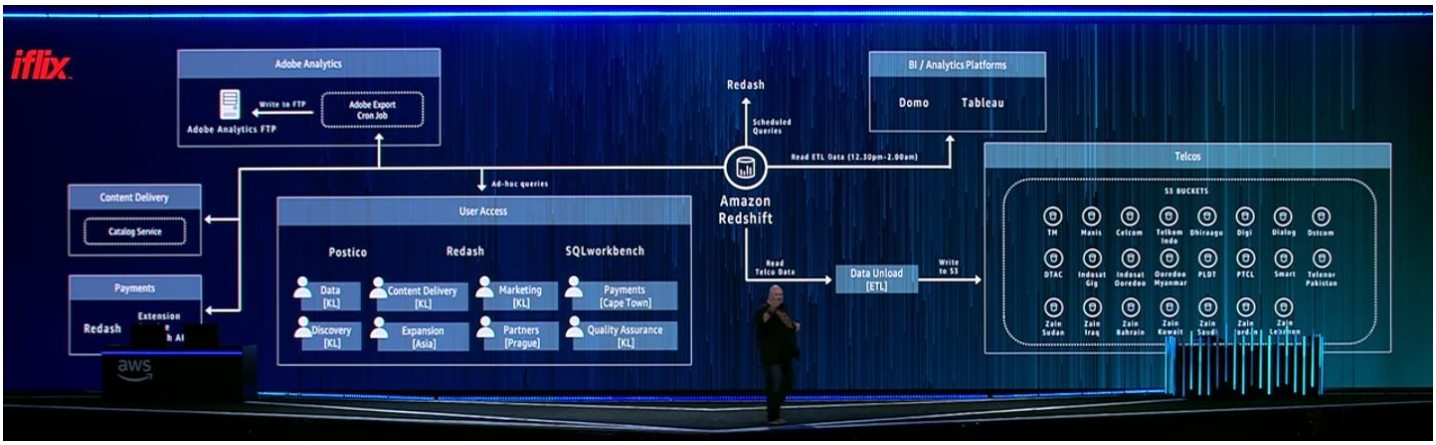
This is their ***application layout*** with subnets, VPC, k8s clusters and their associated databases running



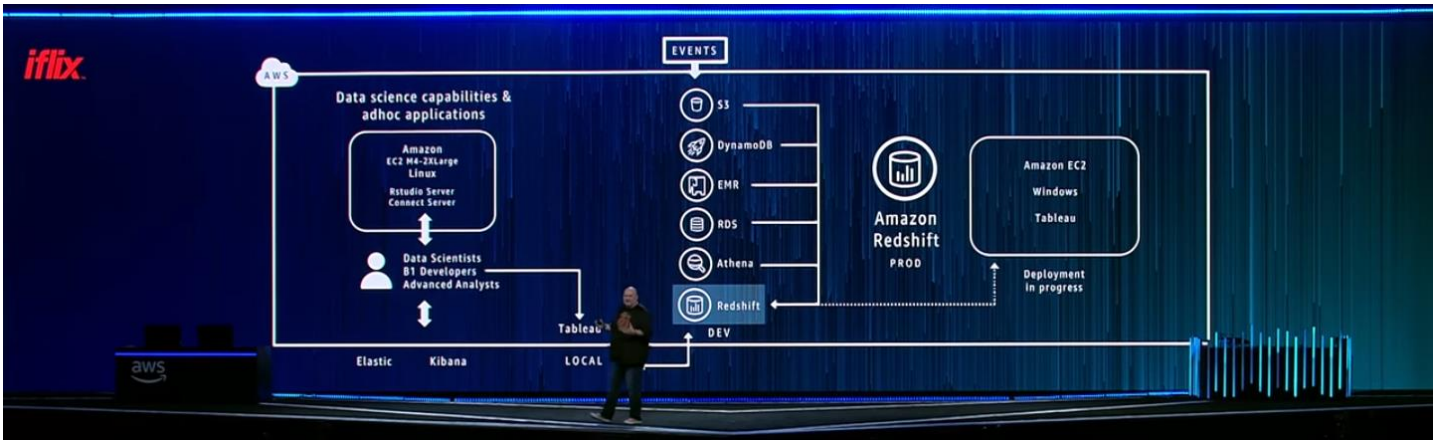
This is their ***Eventing infrastructure/architecture*** showing how events flow around their system



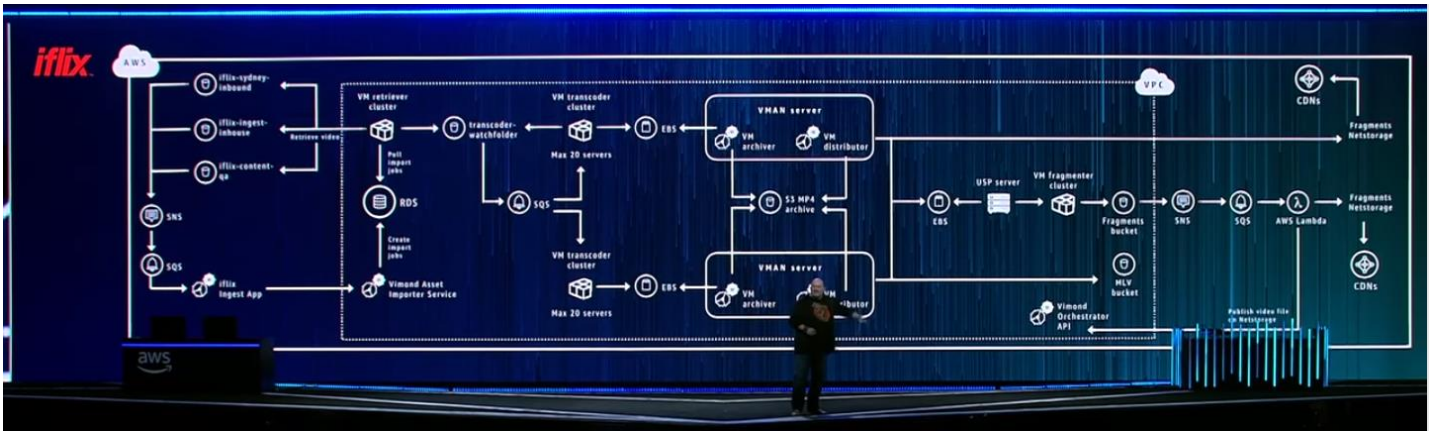
This shows where the ***events*** go



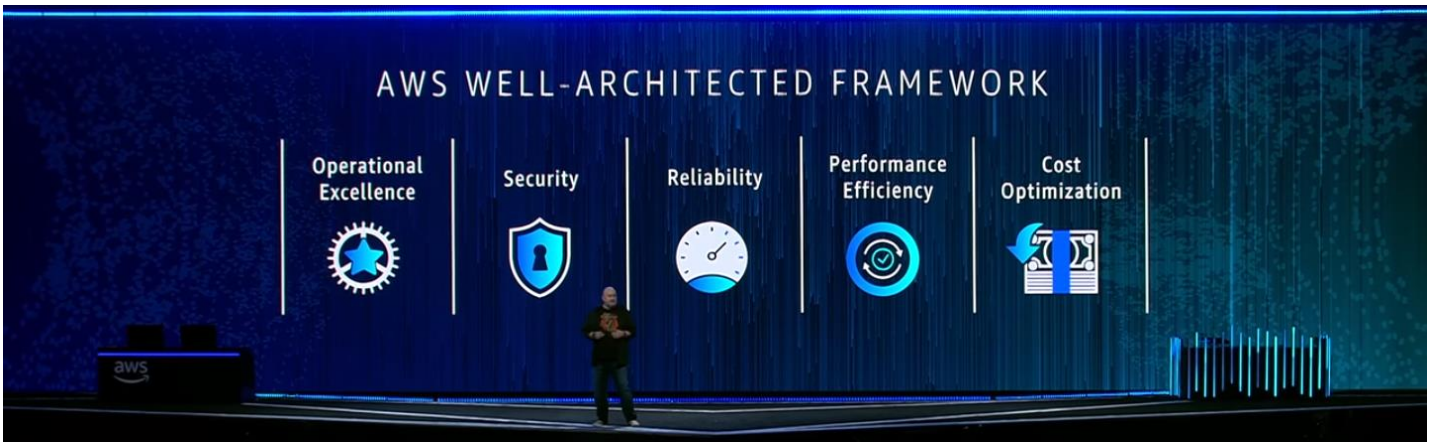
This is their Analytics infrastructure.

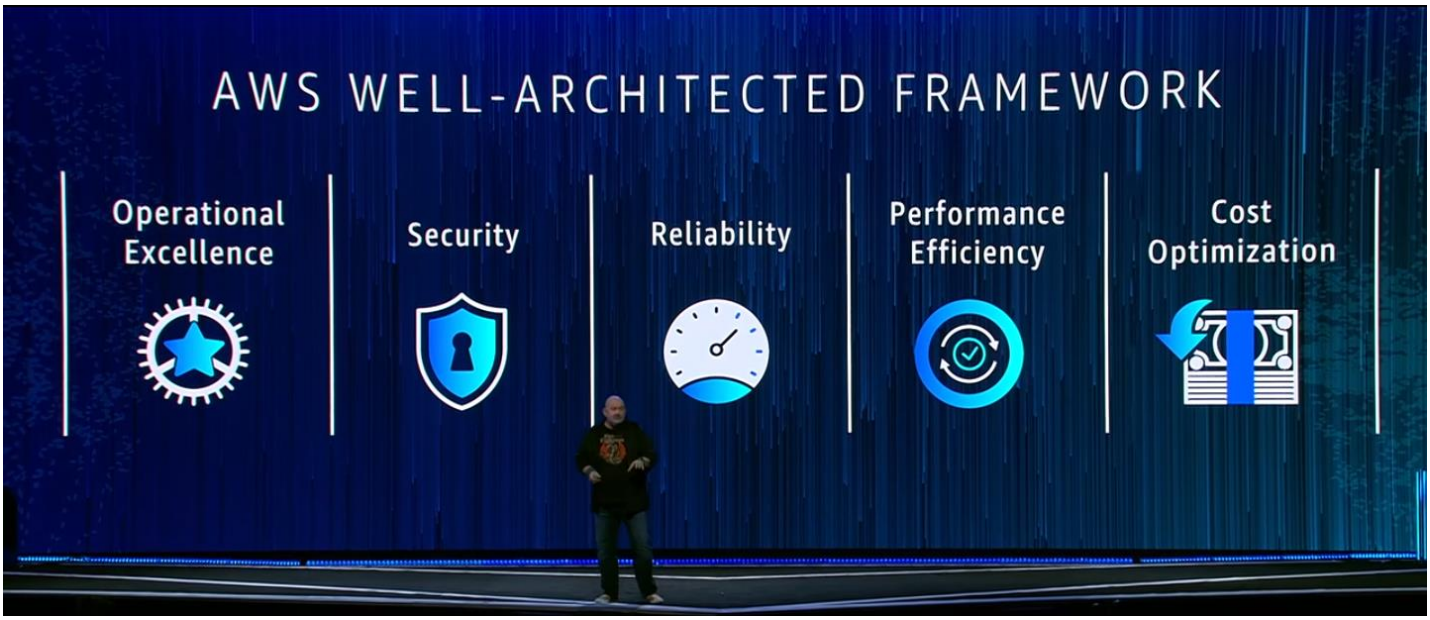


This is the environment where their ***data scientists can experiment*** with the data being collected to develop new products

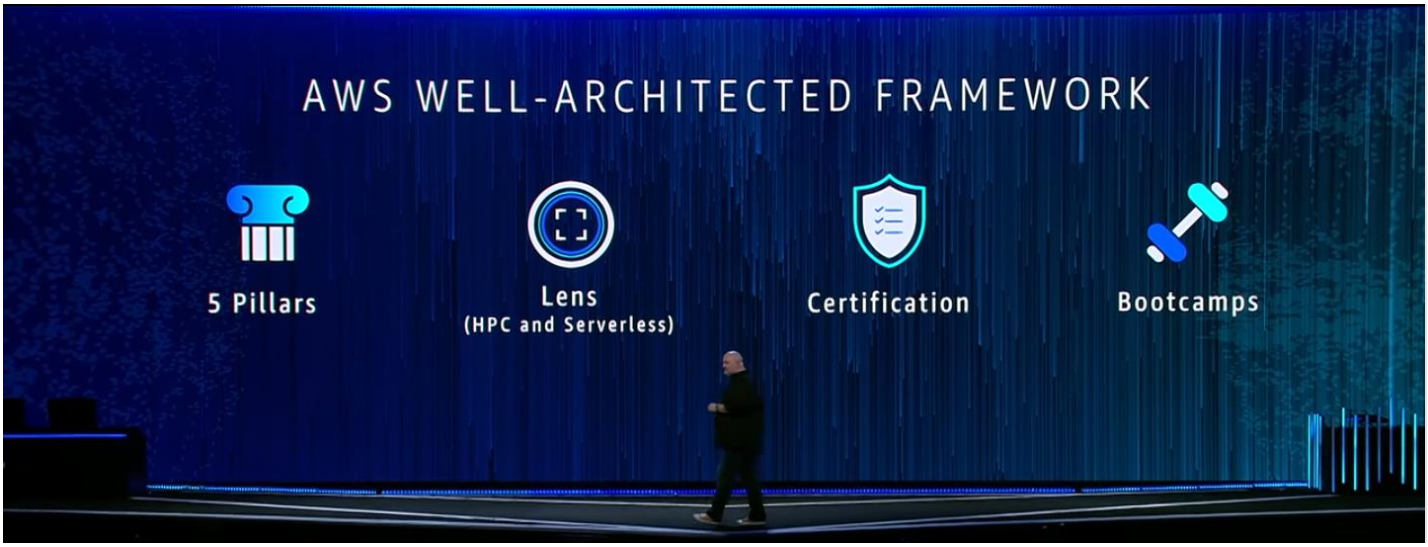


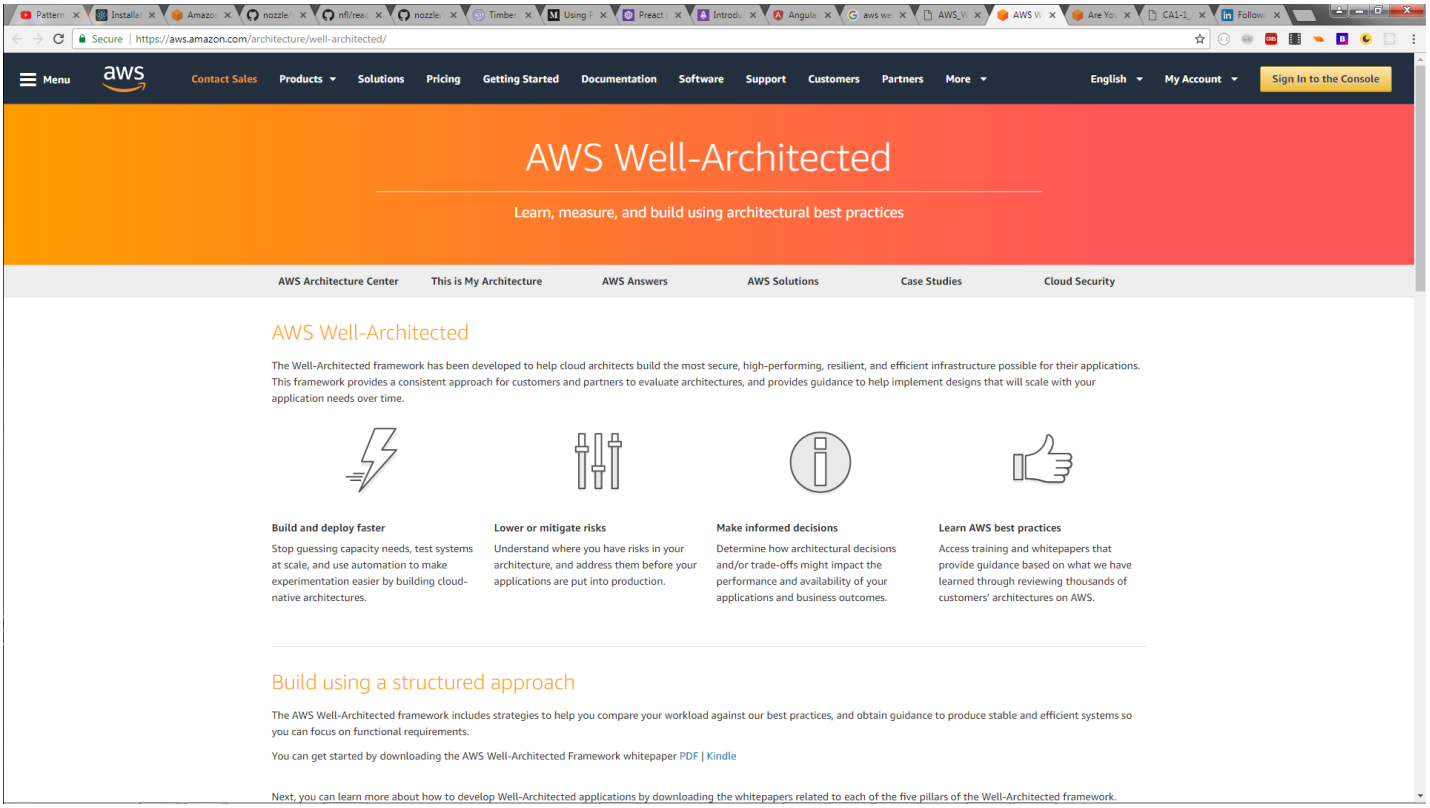
This is their ***Video Data Ingestion environment*** about how to ingest video data in real time. AWS helps you to control complexity and keep things simple when building real architectures.



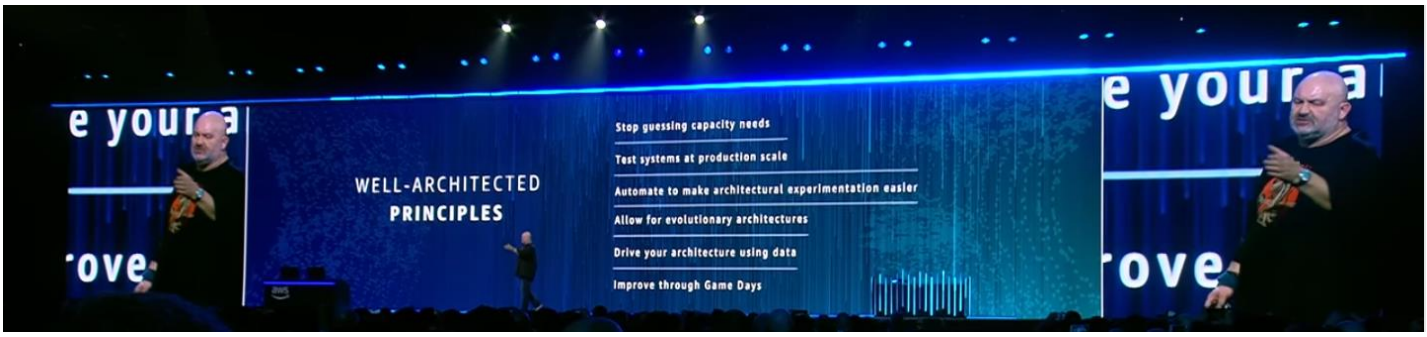


AWS Solution Architects have done extensive work in proposing 5 main pillars and give advice about to build very extensive architectures on top of AWS





https://aws.amazon.com/architecture/well-architected/

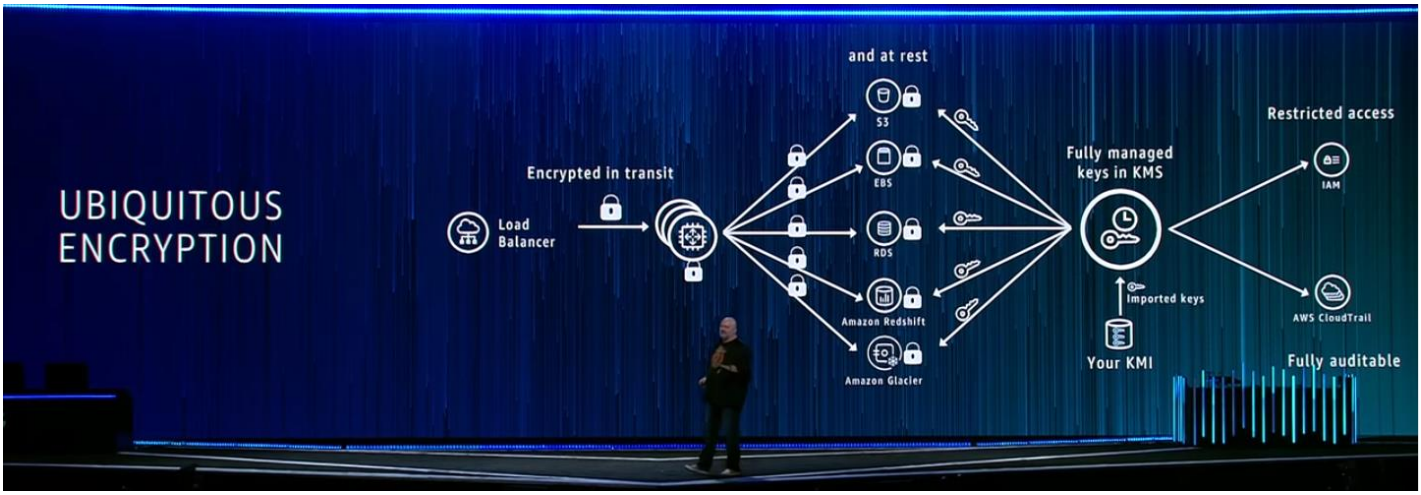


Make sure that you can evolve your systems over time and be able to be decomposed into smaller building blocks independently over time without having to change the overall system in one go.

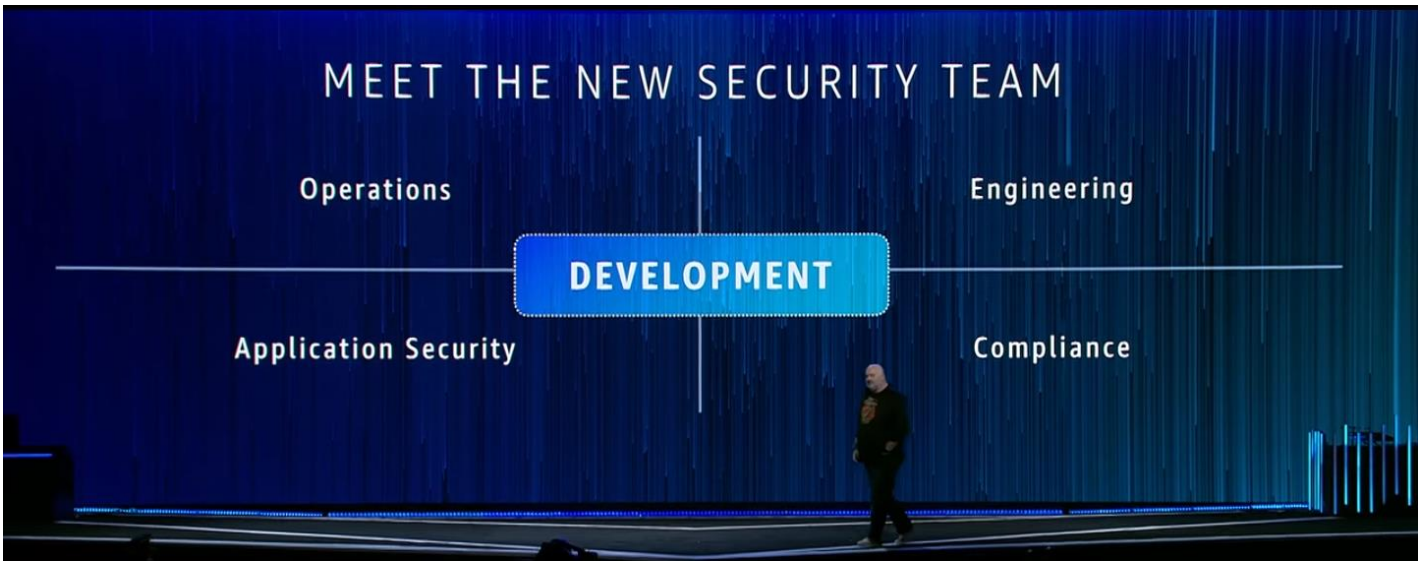
Protecting your customer and your business should be your number one priority, this comes before any feature development

Start taking away IAM roles until your engineers can no longer do their jobs, automate your best practices. You also need to have your playbooks and your runbooks ready every time

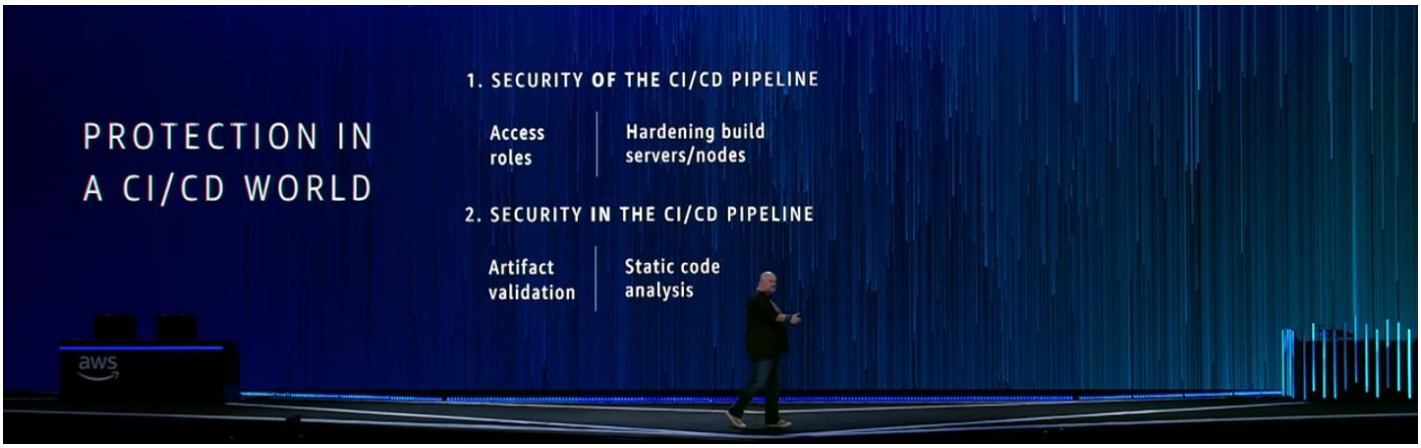
Encryption is the only way to know that you are the only one with access to your data. Signal-To-Noise toolkit can help you with this to encrypt data at rest and in transit

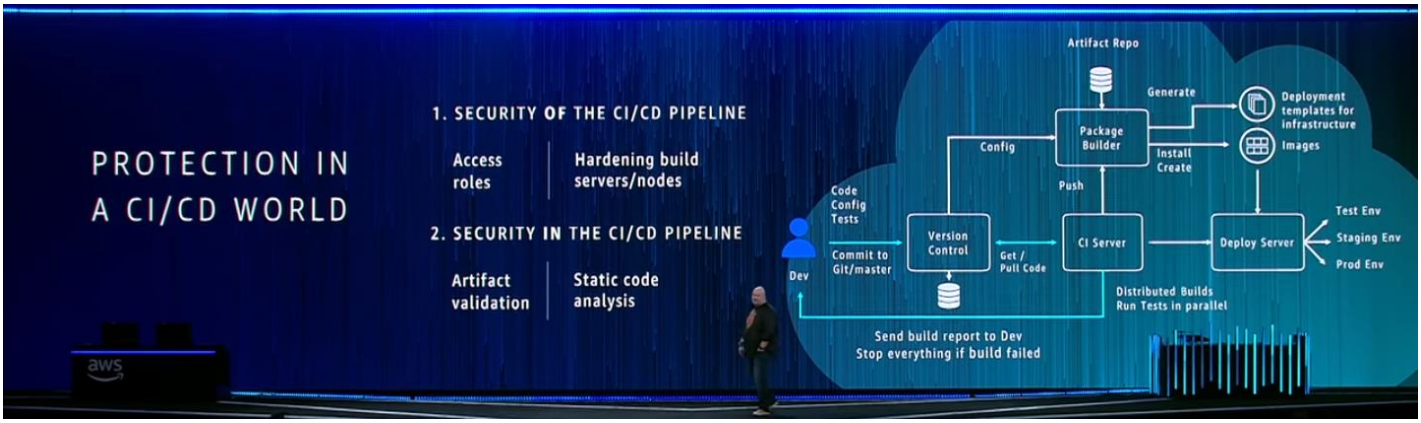


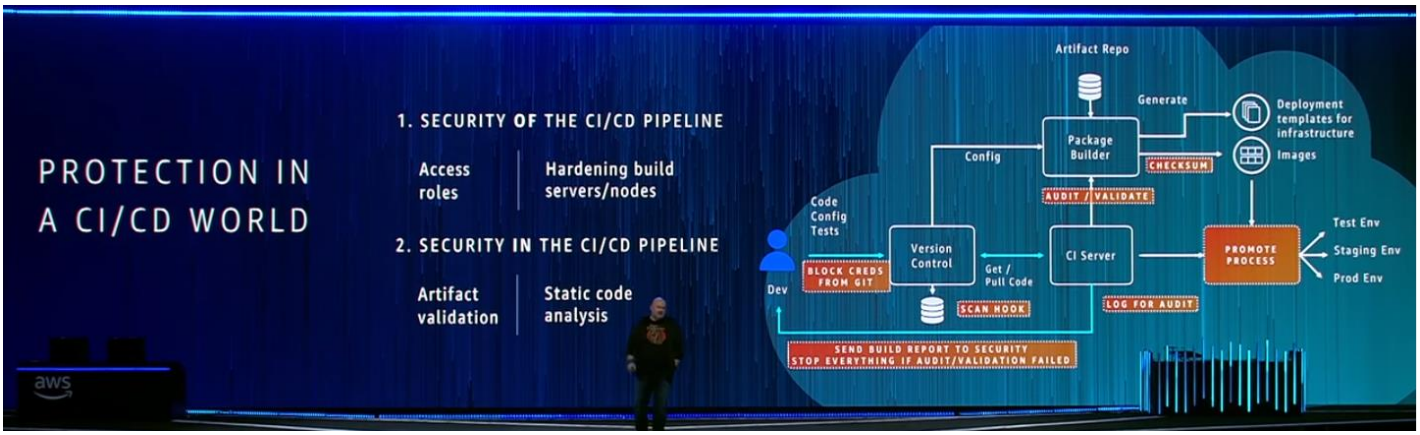
Everyone should now be a security engineer, we all have to protect our customers



The best way to do this is via Automation

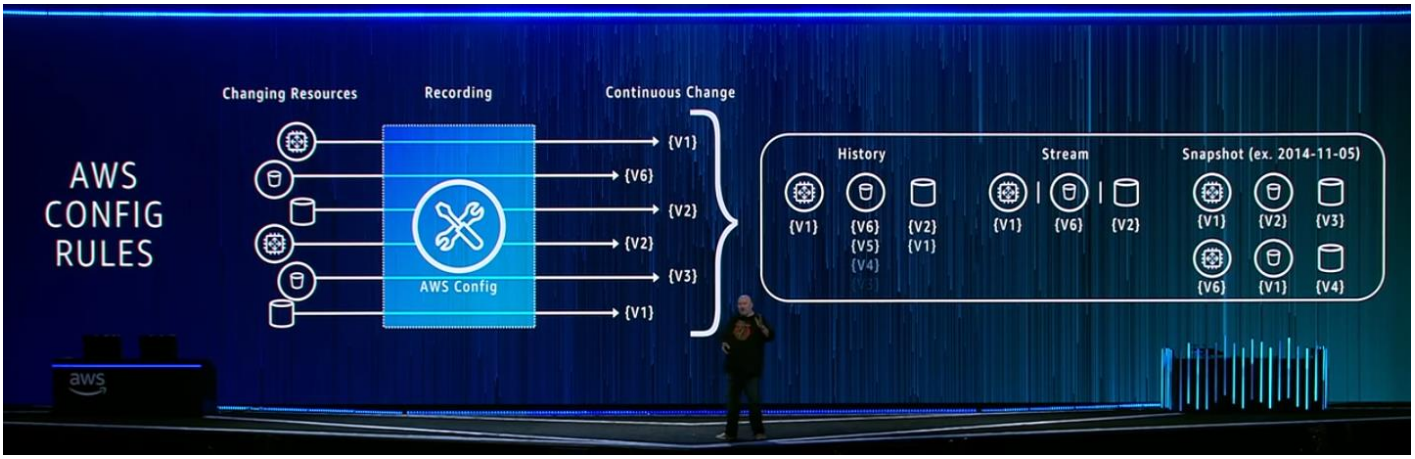




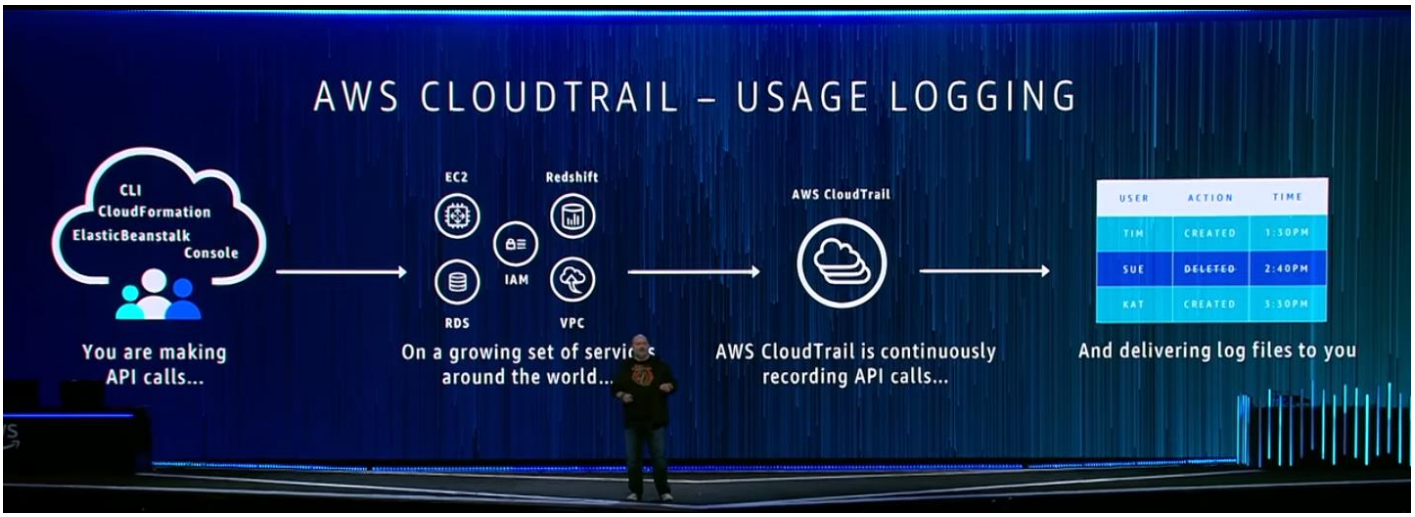


This is how your CI/CD pipeline should look like, each piece in the architecture should get security integrated into it to make sure that the software being developed is a s secure as it can be.

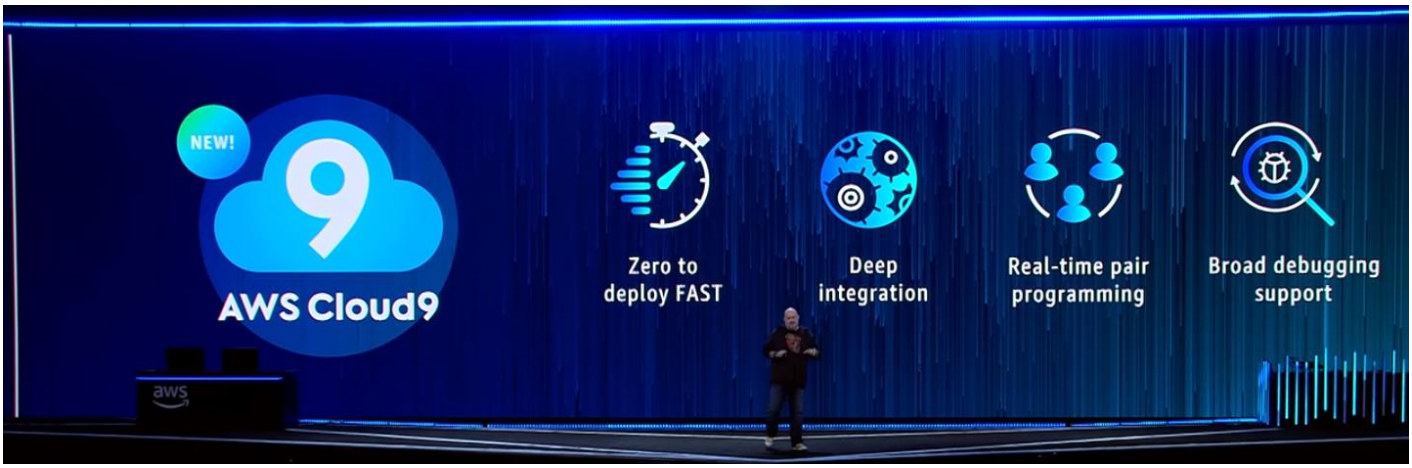
Any change to IAM needs to be manually inspected, every change should be audited



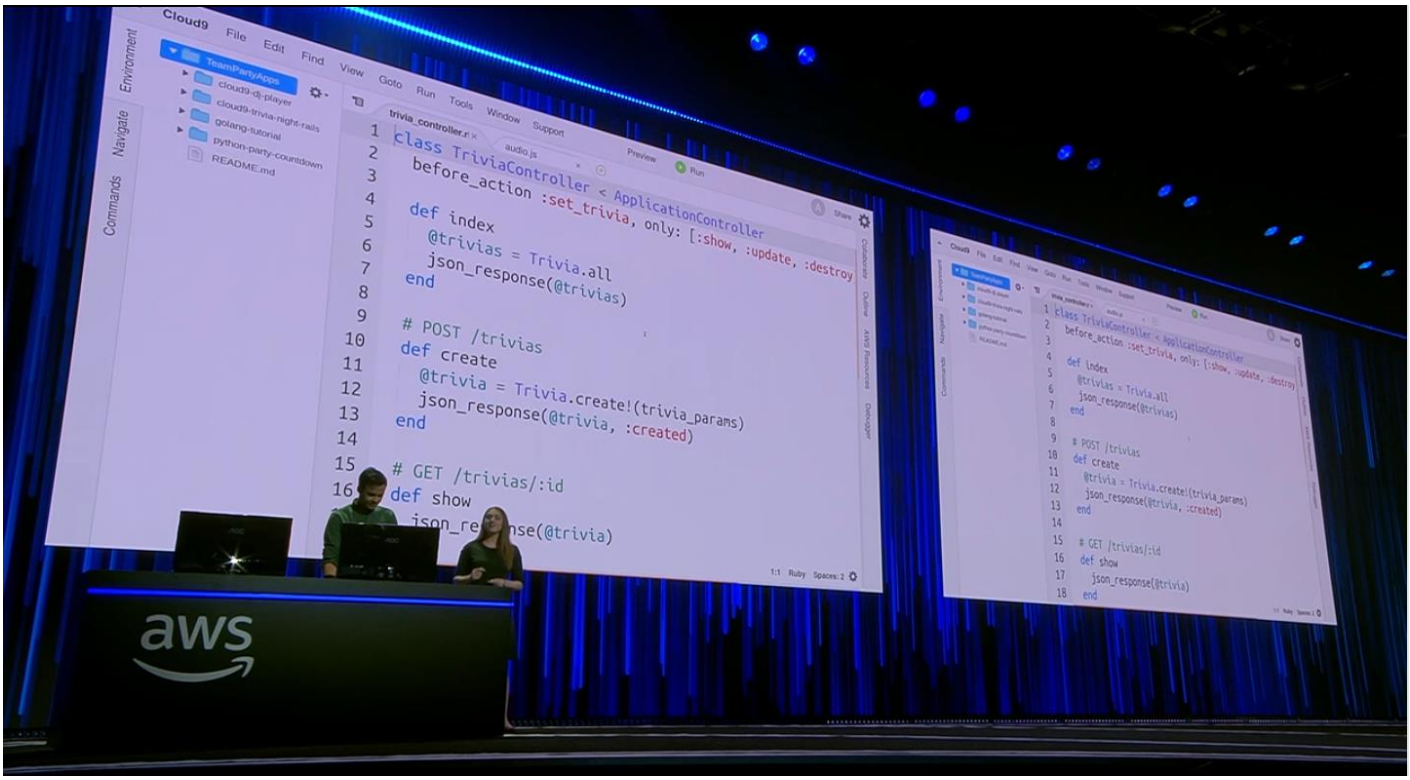
AWS Config Rules service can help you discover your whole environment, all the tools, services, and resources that you are using, and then it can start building an audit trail about all the changes that you are making to the resources that you are using. You can get streams as text, SNS notifications, create snapshots and compare them to how your system looked 2 weeks ago, etc.



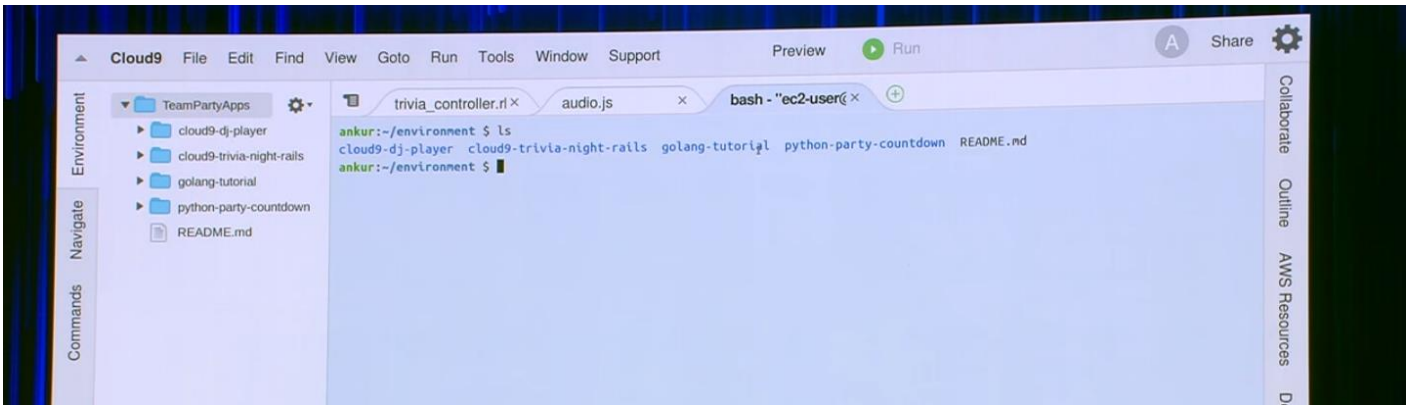
AWS CloudTrail will log every API call to every one of your service, then put the logs into secure buckets in S3 for you to do analysis on who is accessing your systems and put alarms on them

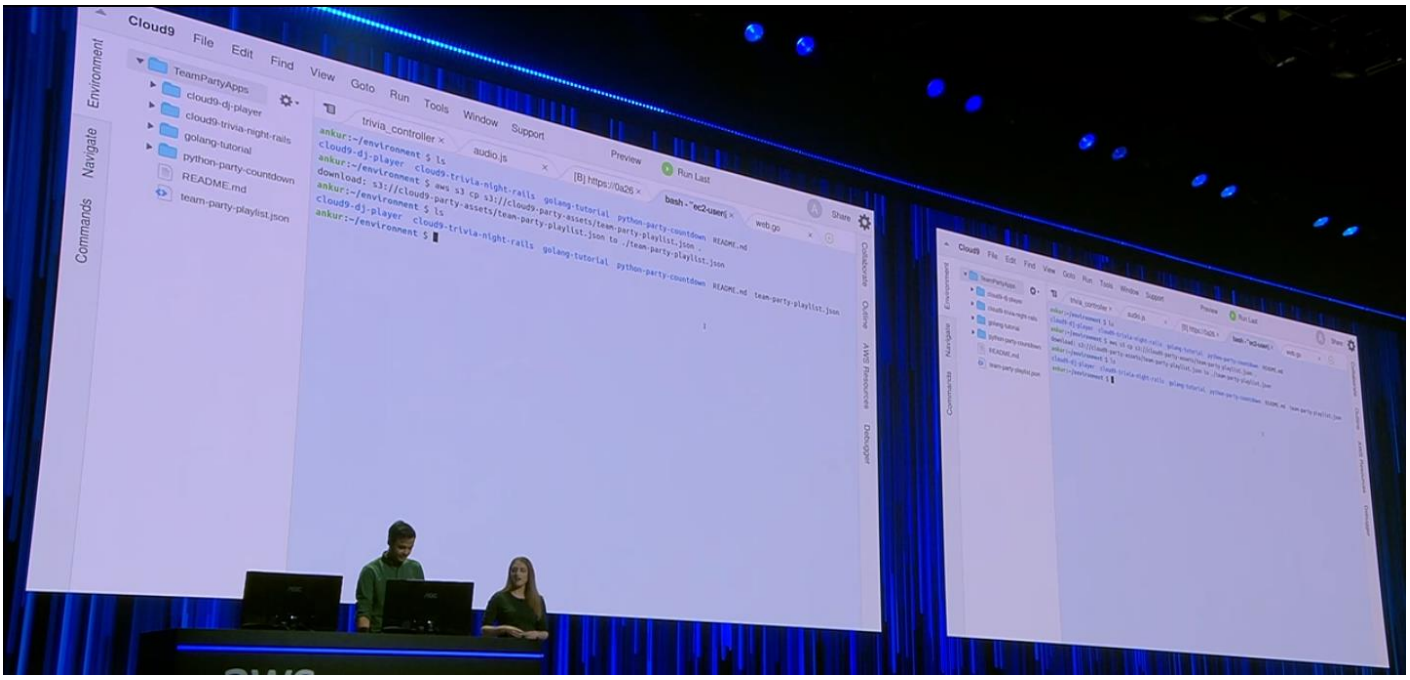


Includes the ability to debug your lambda functions in the cloud

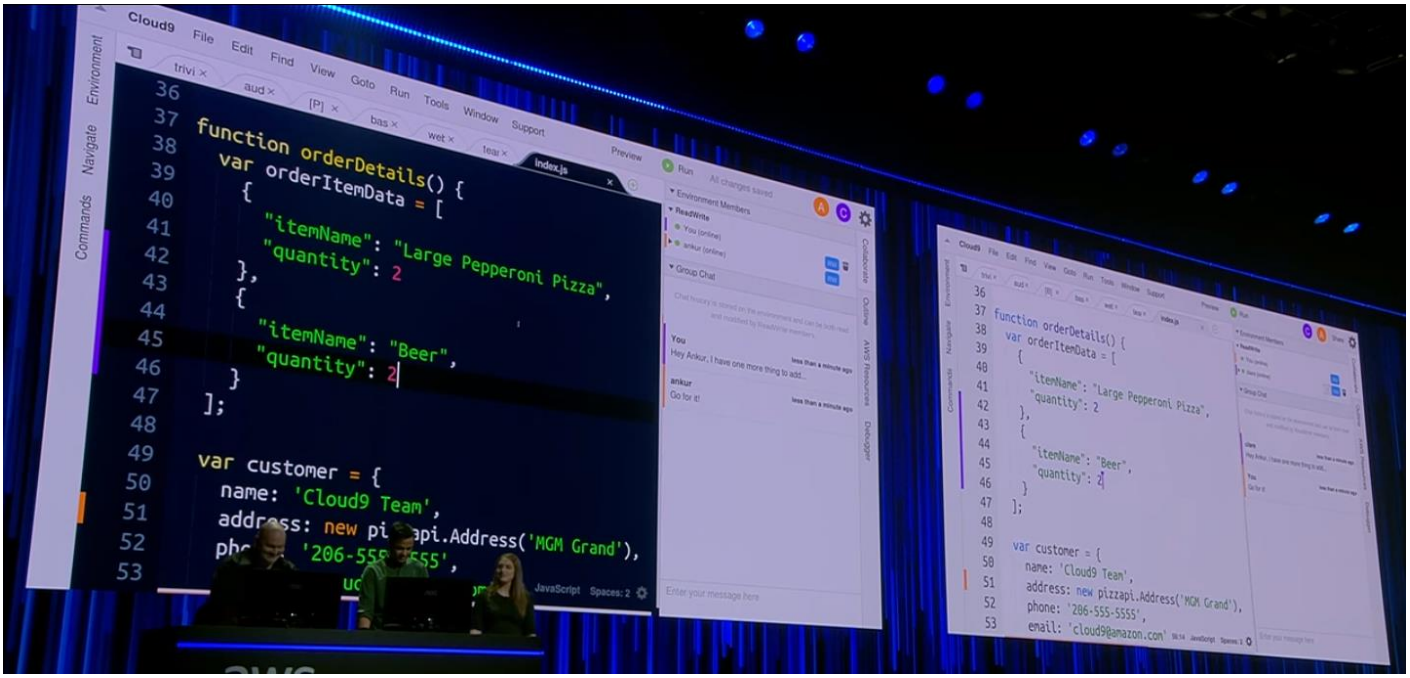


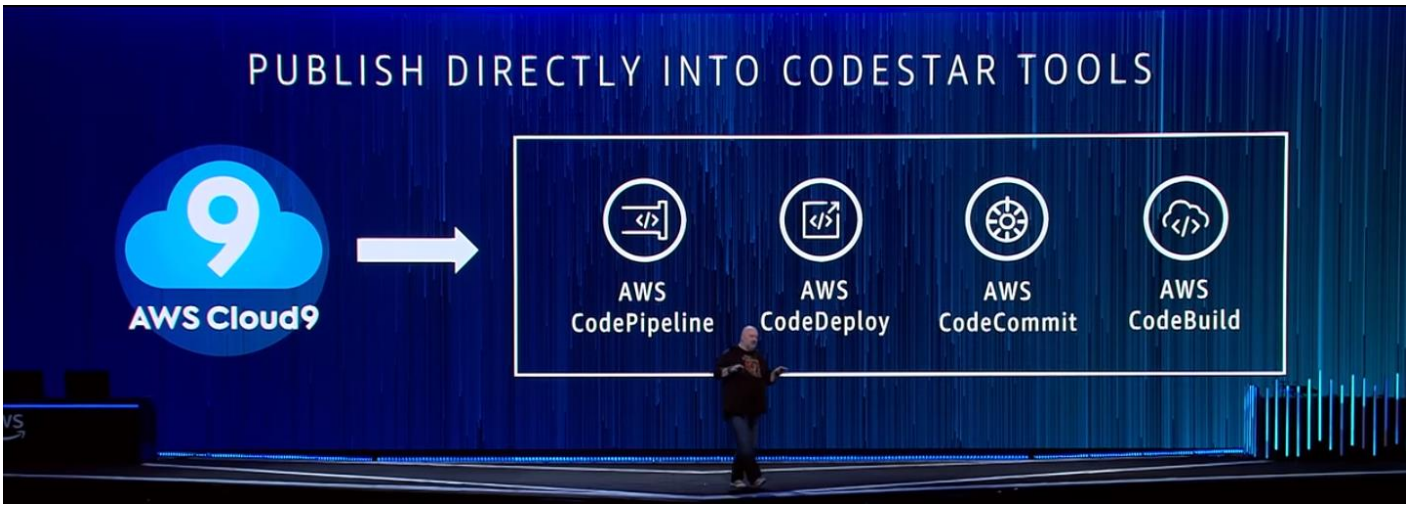
This is Ankur’s Cloud9 environment that we are able to pull up on any browser from anywhere,

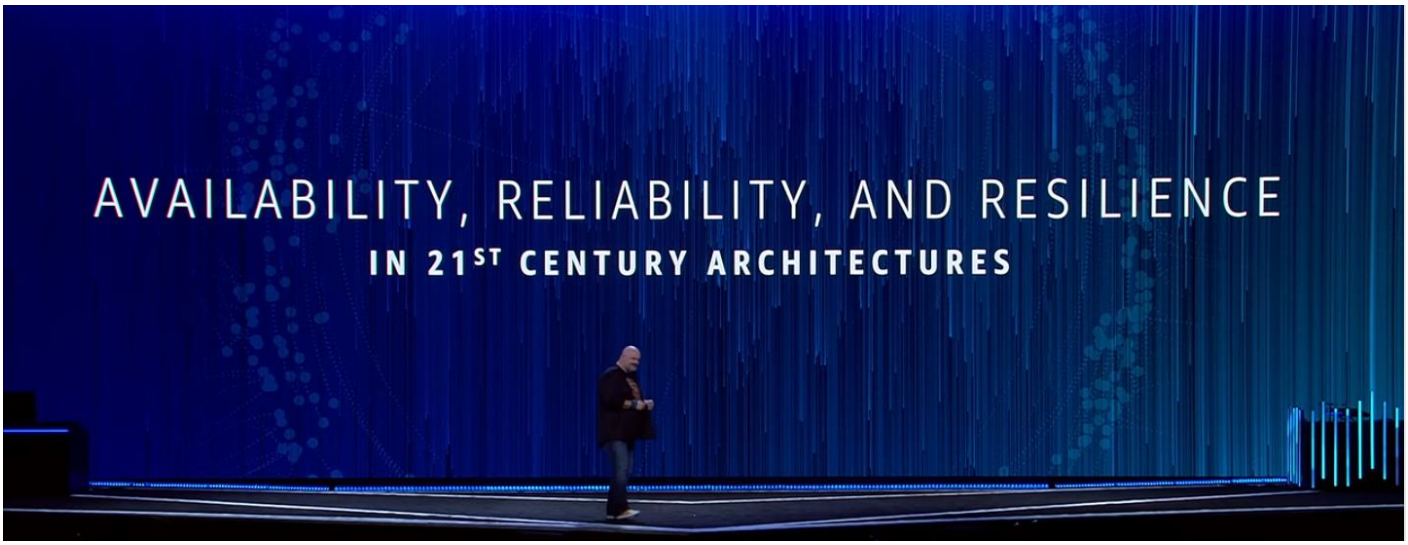




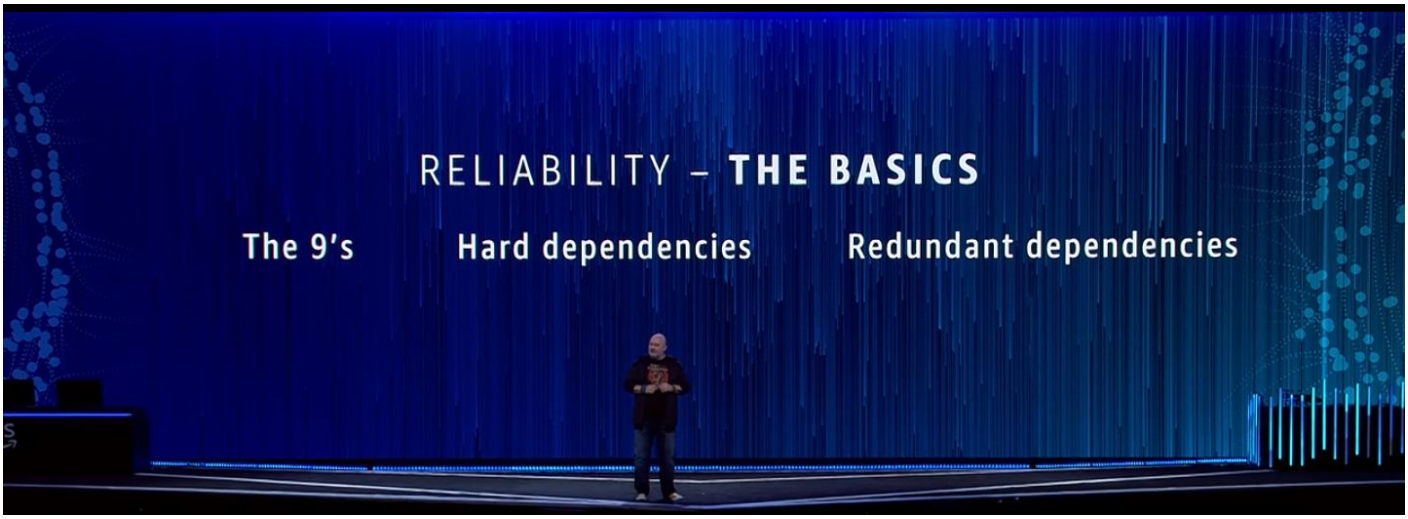
The AWS CLI comes pre-installed and can be used to access AWS resources using your automatically attached account credentials



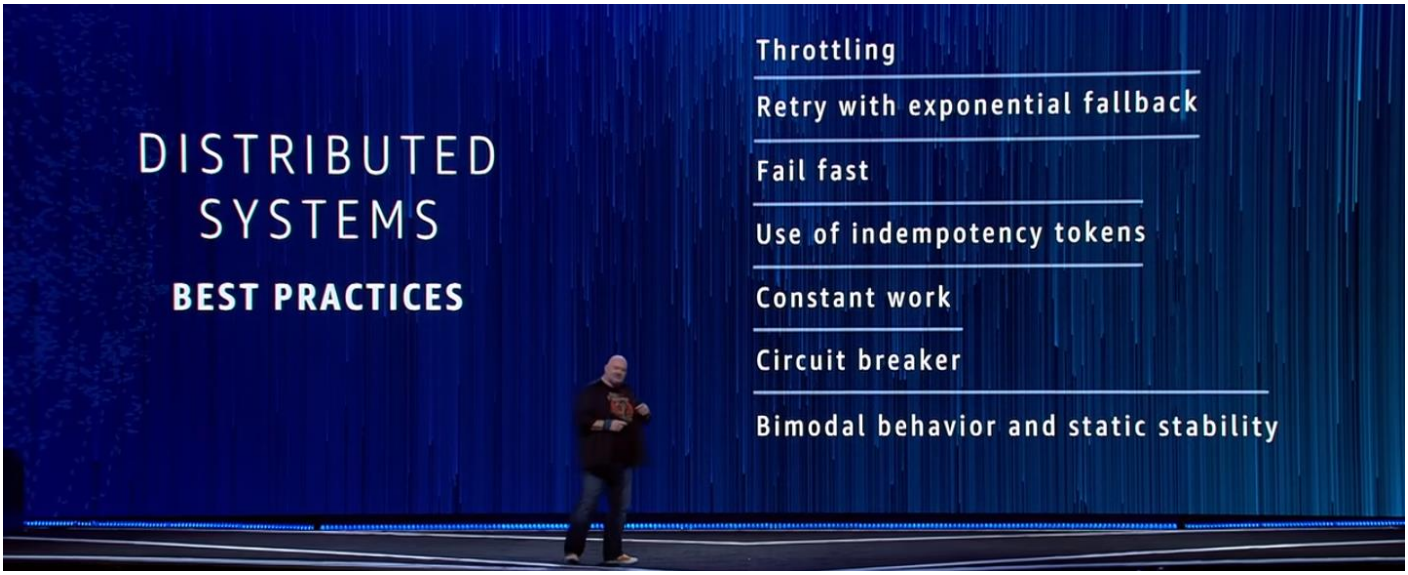

PUBLISH DIRECTLY INTO CODESTAR TOOLS
AWS Cloud9
AWS CodePipeline
AWS CodeDeploy
AWS CodeCommit
AWS CodeBuild


AVAILABILITY, RELIABILITY, AND RESILIENCE
IN 21ST CENTURY ARCHITECTURES

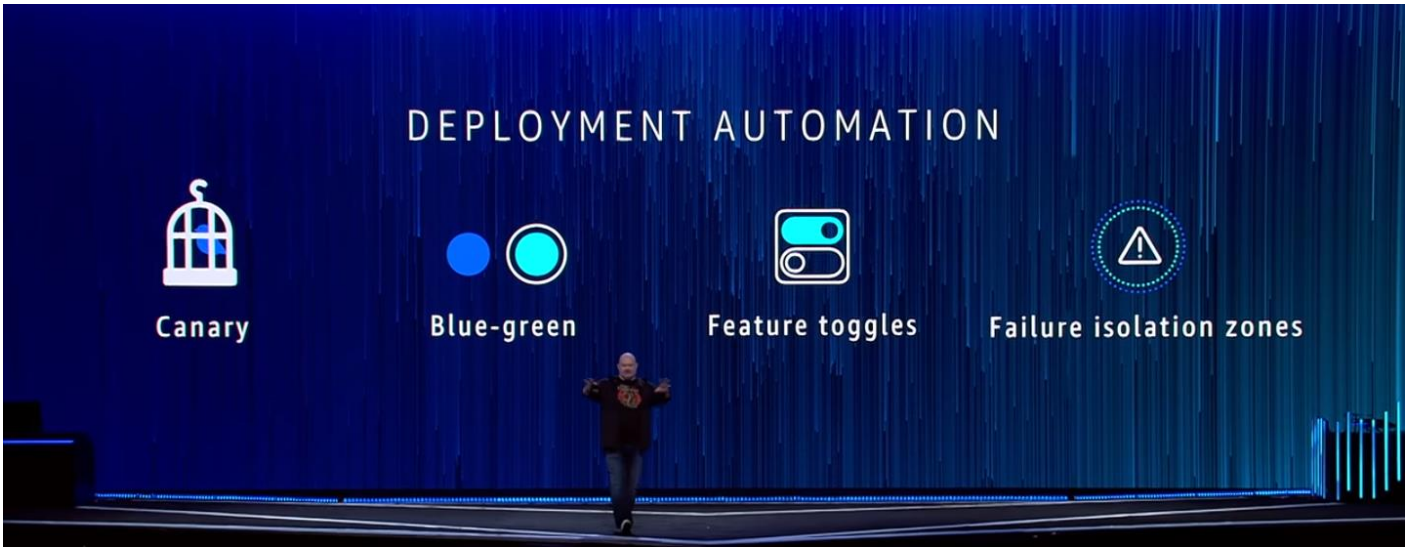

AVAILABILITY,
RELIABILITY,
AND RESILIENCE
IN 21ST CENTURY ARCHITECTURES
Test recovery procedures
Automatically recover from failure
Scale horizontally to improve availability
Stop guessing capacity
Manage change through automation

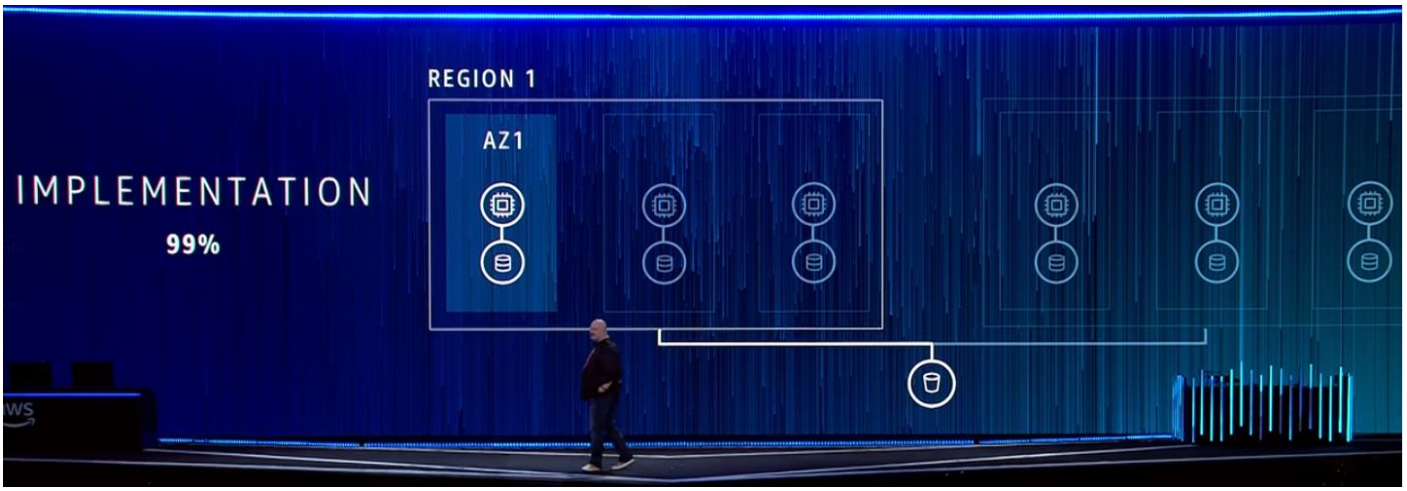


This requires that you build isolated cells of functionality within your application to prevent failure in one part taking down the entire application. You need to make sure that critical inter-cell communication is kept to a minimum, this reduces the blast radius of a cell is limited to that cell only.

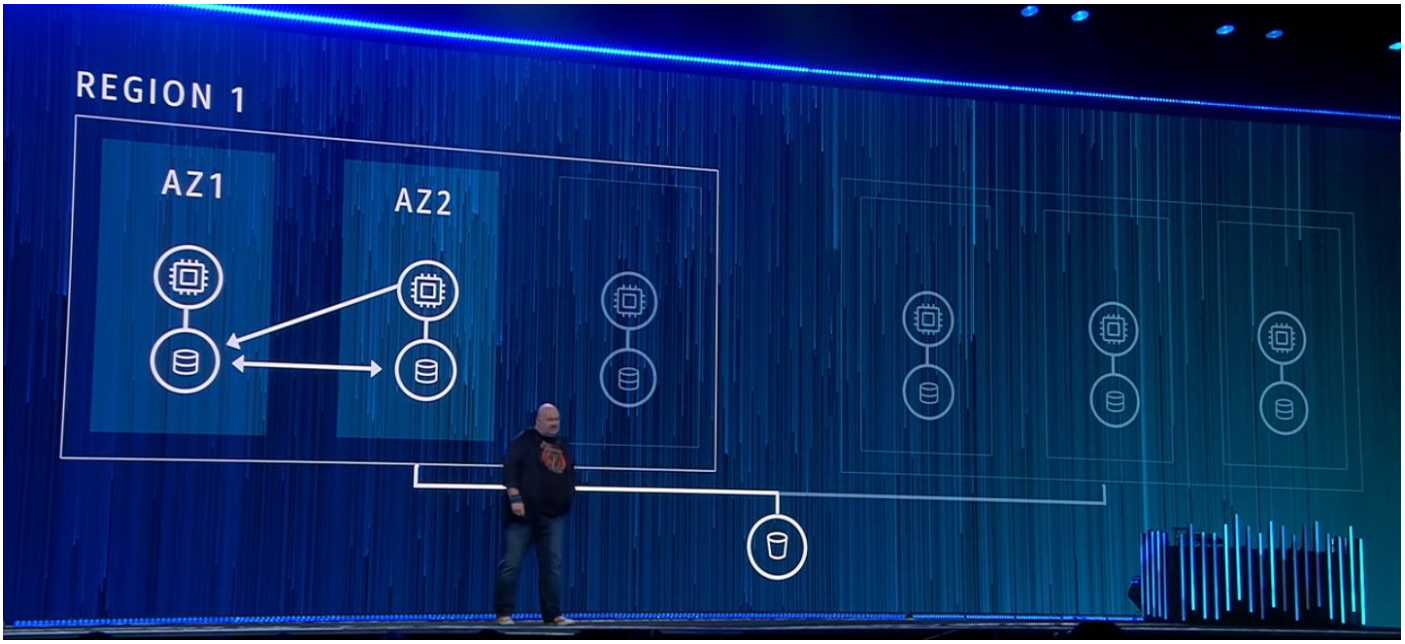


Throttle when you get overwhelmed, don't try to satisfy all calls during a load spike. Fail fast if you are going to fail so that you release resources back quickly, use circuit breakers to take bad nodes out of the loop.

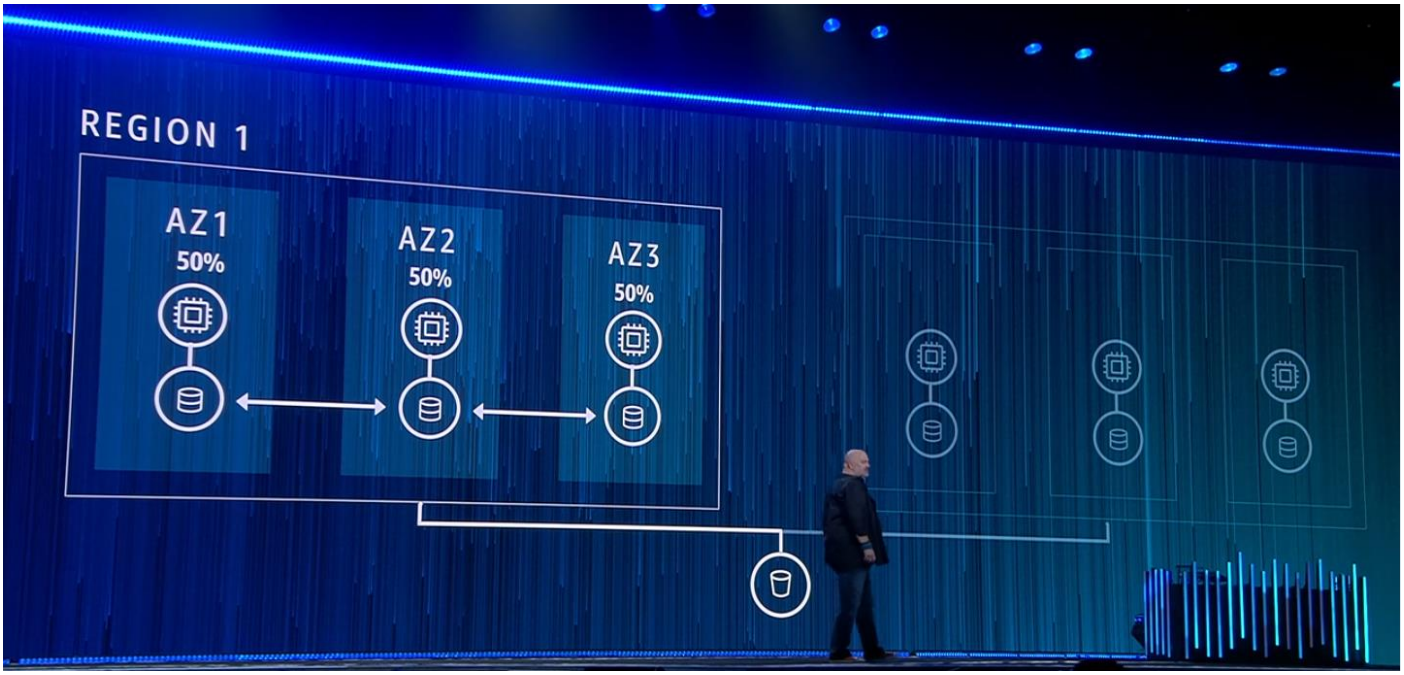




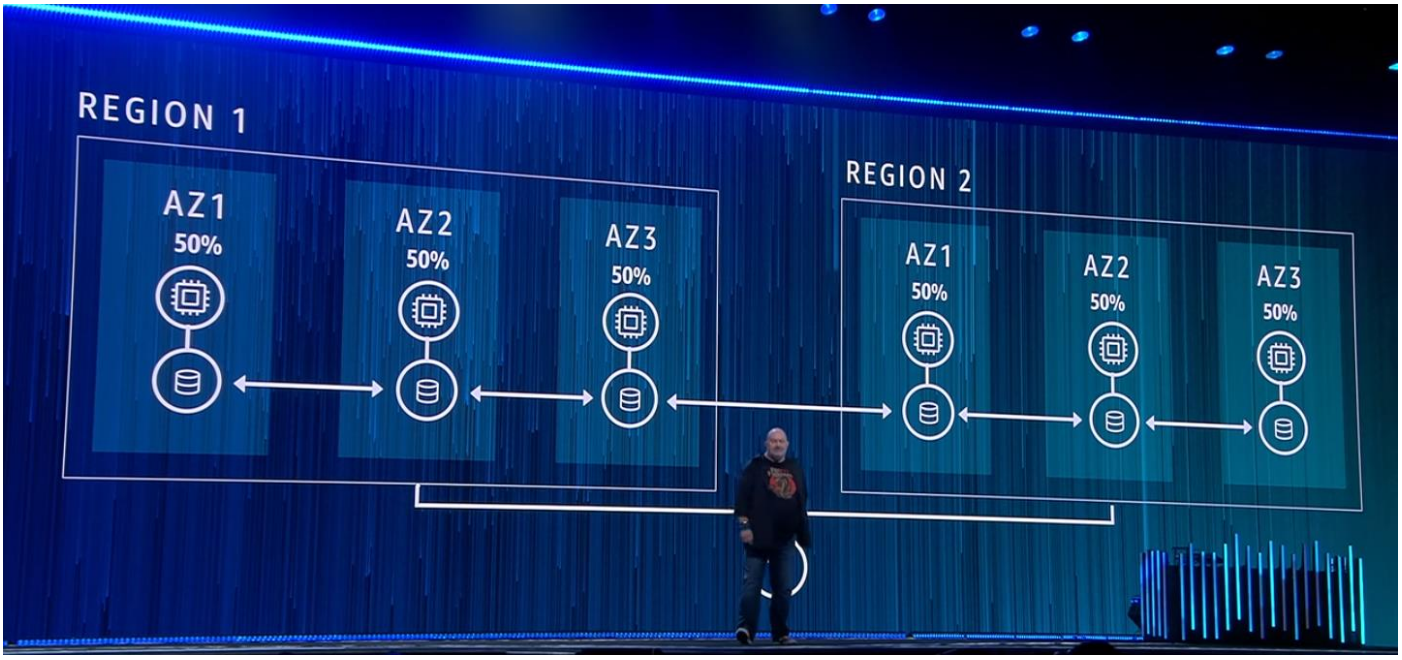
This is cheap but gives 99% availability, outages are costly and longer though



This architecture will give you a 99.5% availability when you deploy into 2 availability zones for fail overs. This limits your down time to about 30 minutes in a year.



This will give you a 99.99% availability when you deploy into 2 availability zones for fail overs. This means that if you lose an entire system in an AZ, you will still have 100% of your resources available to your users. This limits your downtime to just a few minutes in a year.



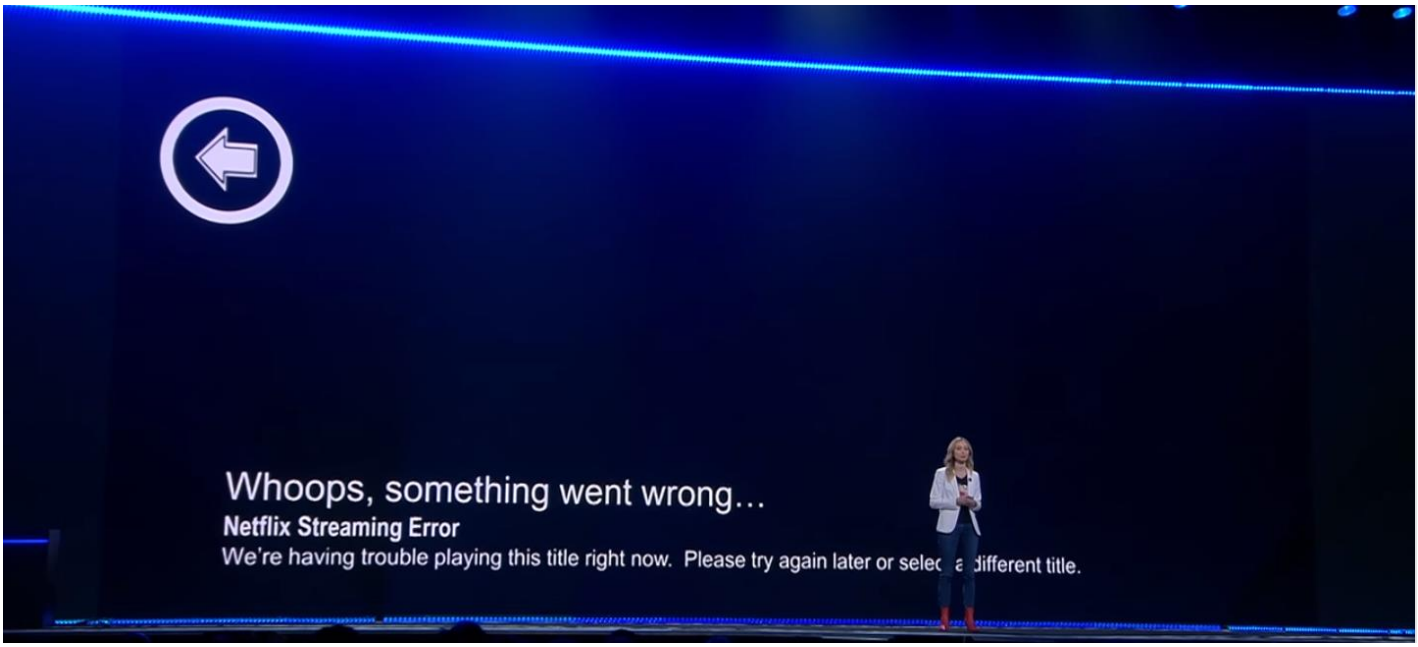
For 99.999% availability, you can deploy this active-active configuration over 2 regions and make use of route53 and use DynamoDB Global tables to give you replication of data over multiple regions. This gives you no downtime in a year availability scenario.

SERVICE	CONTROL PLANE	SINGLE AZ DP	MULTI AZ DP	SERVICE
EC2	99.95	99.95	99.99	
RDS	99.95	99.95	99.99	
ECS	99.95			99.99
EBS	99.95			99.999
Kinesis				99.99
DynamoDB				99.99
Cloudfront	99.9			99.99
Route 53	99.95			100
IAM	99.9			99.995
KMS	99.99			99.995

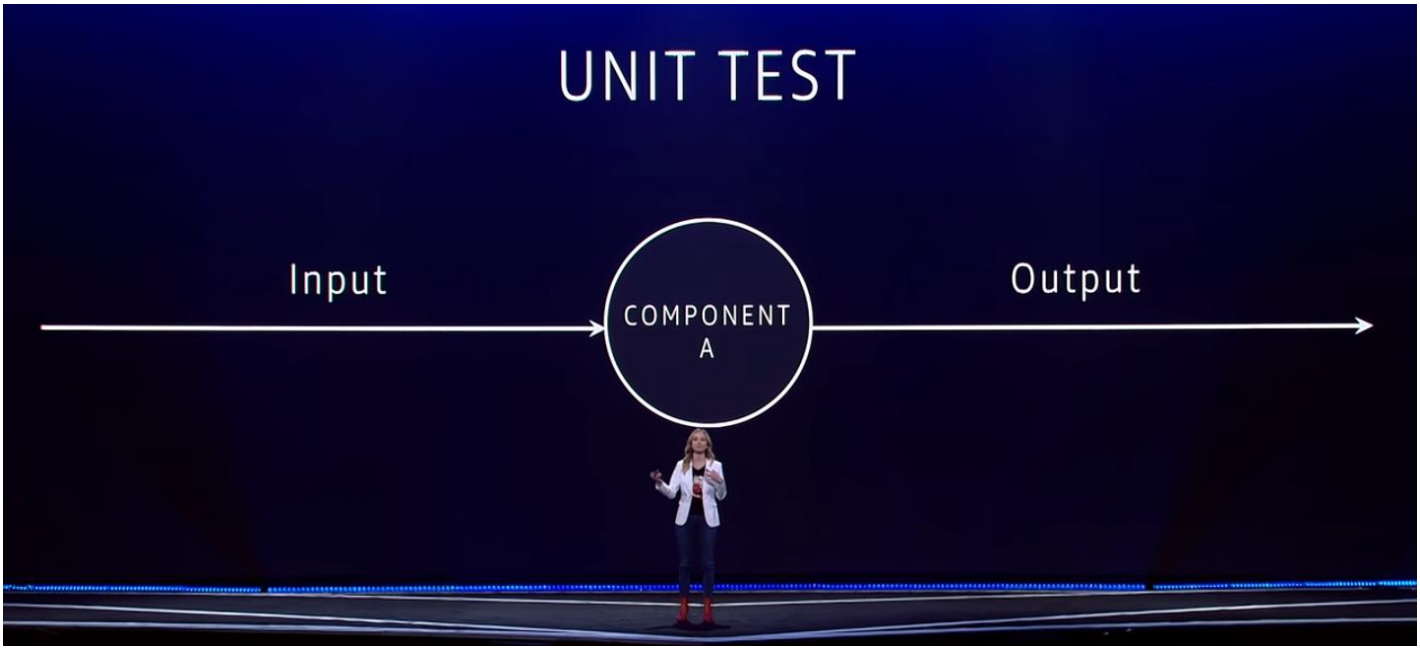
Check this numbers out in the reliability tables within the Well Architected Framework document

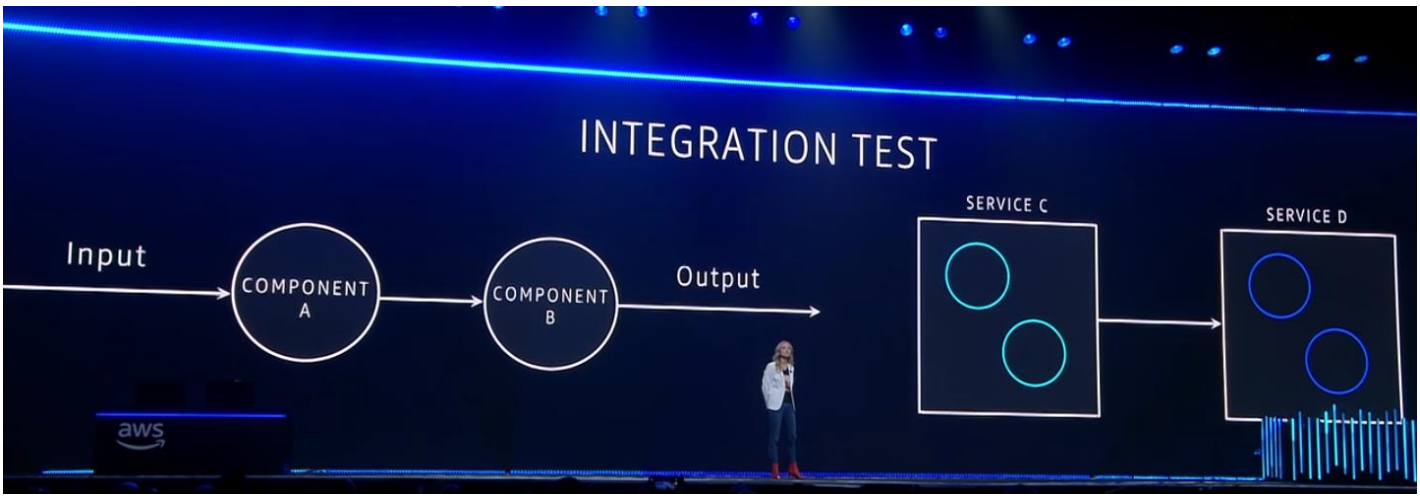
Break your systems and see how they respond to failures, there is a lot to be learnt about your system architecture

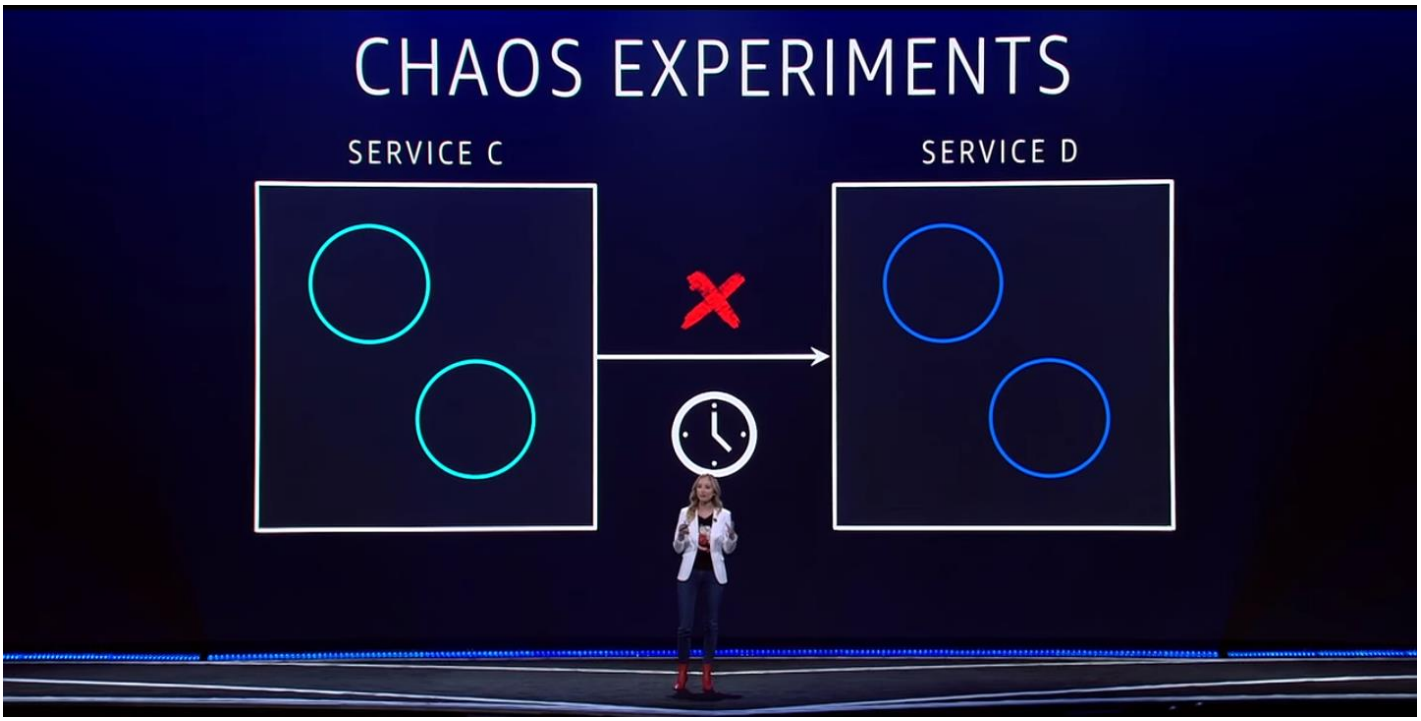
Suppose a section of your screen isn't loading its data, this should result in us not showing you that section anymore and instead show the other things available



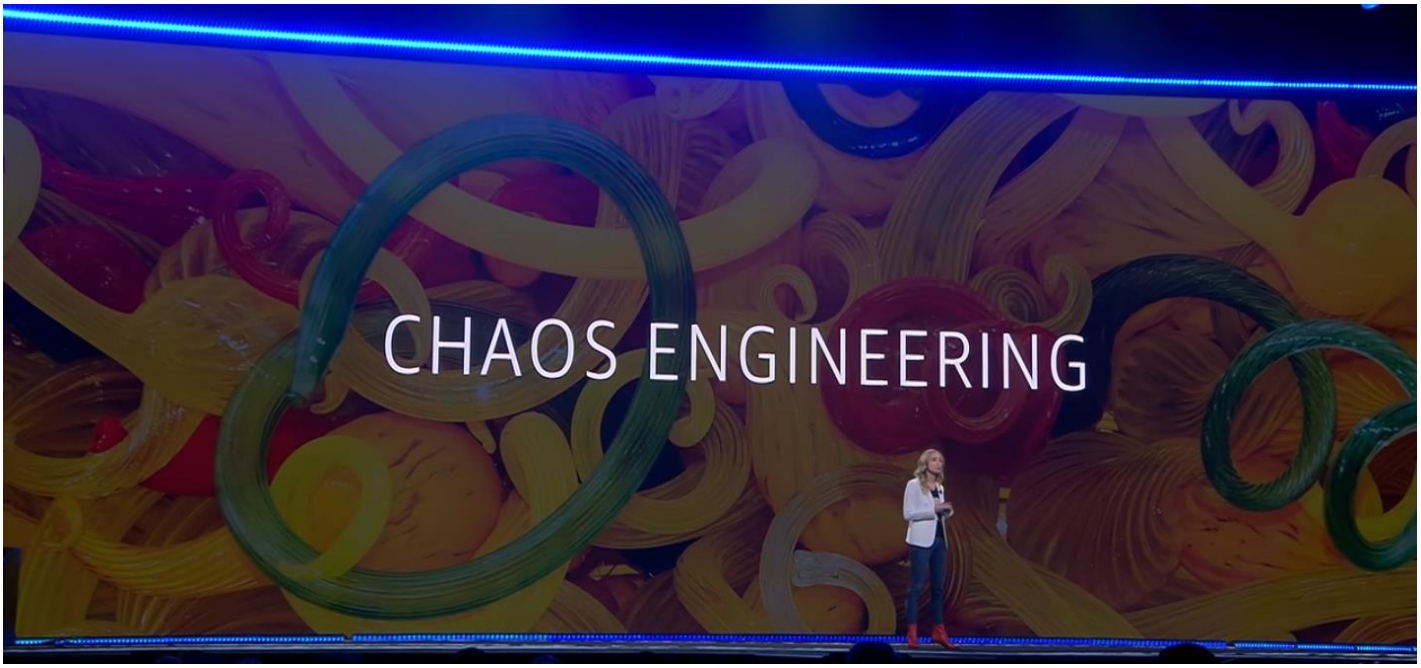
The service failing should not result in this type of screen ever! We use chaos expecriments to guarantee this never happens.





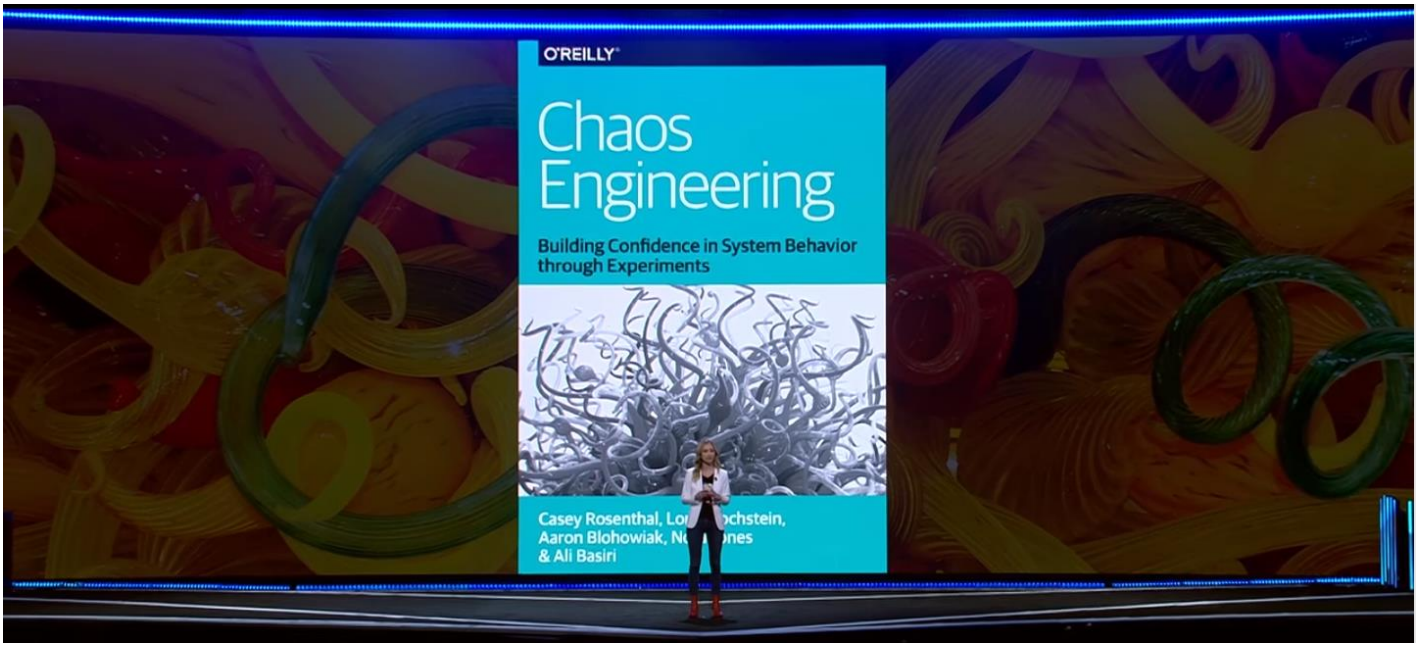


In ***Chaos experiments***, there is an option to add a failure or add a time latency to the response going back to the calling service between calls. Here we have unknowns and are resilient to these failures



This is about experimenting on a distributed system in order to build confidence in the systems capability to withstand turbulence in production. Chaos engineering in not replacing other types of tests, it is for guaranteeing resilience

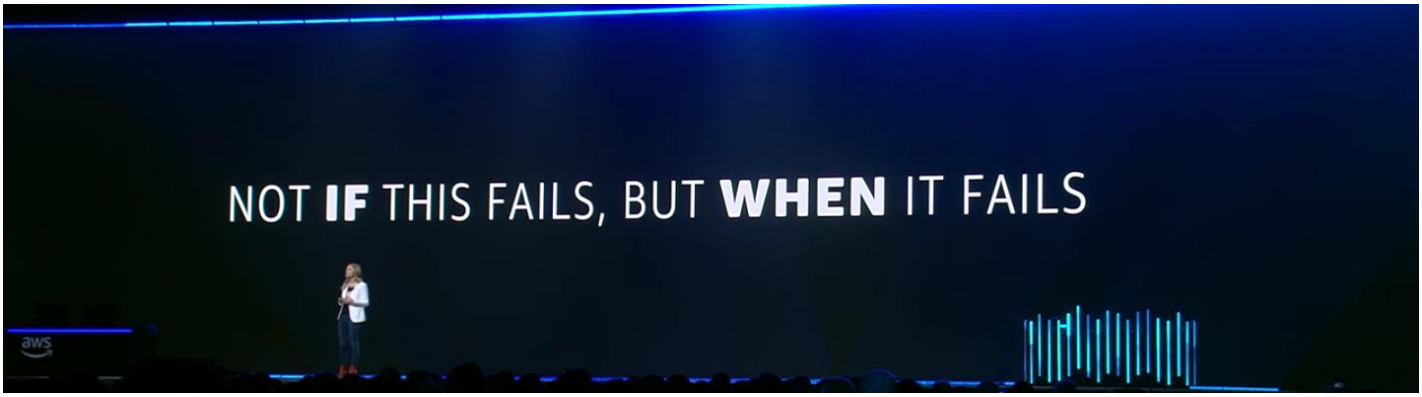

NETFLIX


O'REILLY
Chaos Engineering
Building Confidence in System Behavior through Experiments
Casey Rosenthal, Lo chstein,
Aaron Blohowiak, N nes
& Ali Basiri

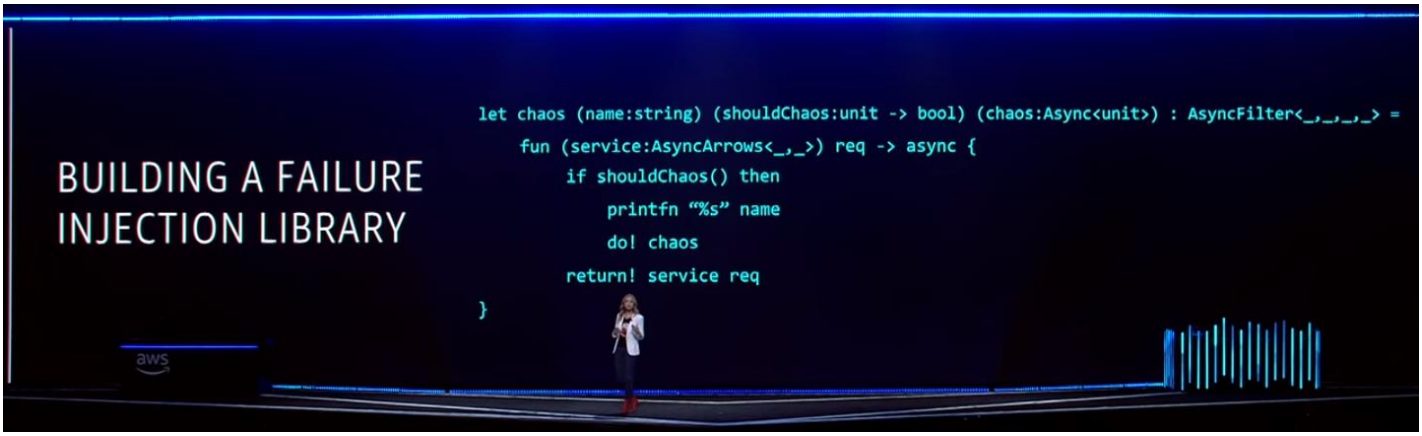
This is also known as Chaos monkeys, bringing down a single instance of some service

Here we started trying regional failover, using Kafka and deleting topics at random

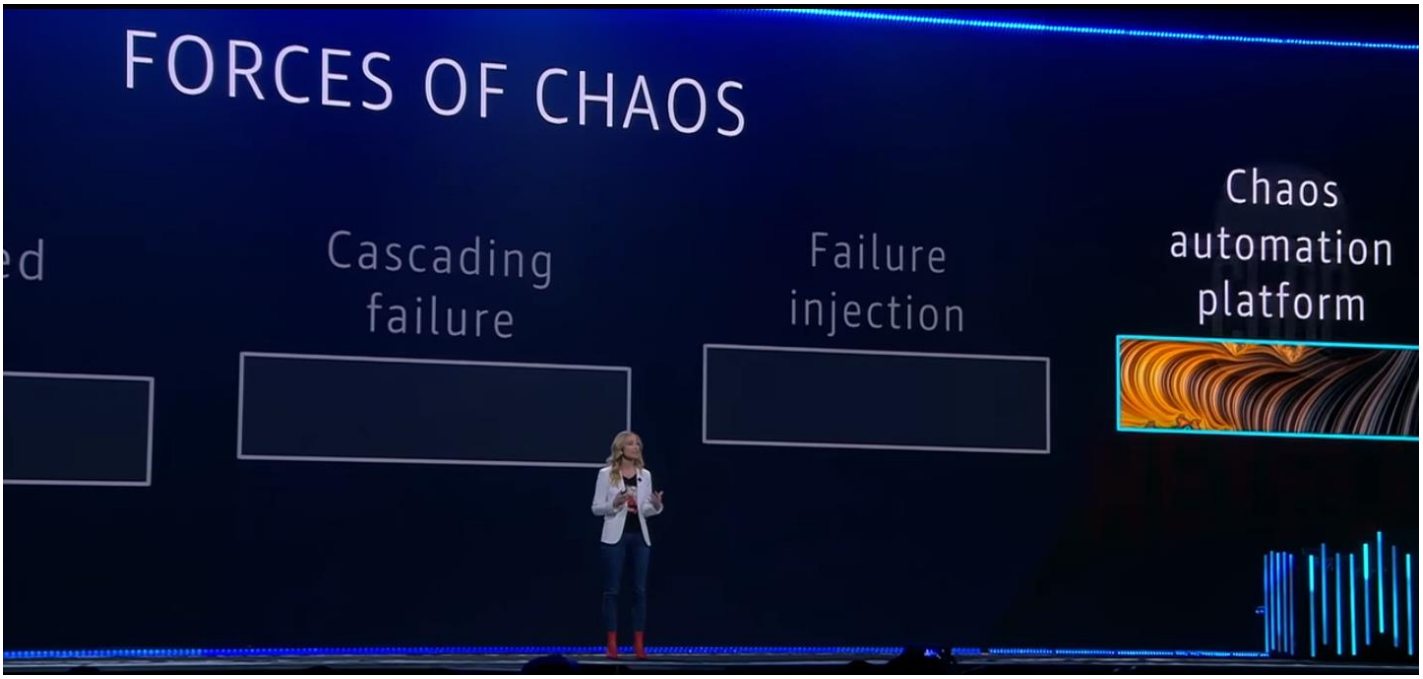
This is when failure in some part of the system triggers failures in other parts, then in other systems. We tried to cause cascading failures and watch what happens on some monitoring screen. We are doing this in QA and NOT production!

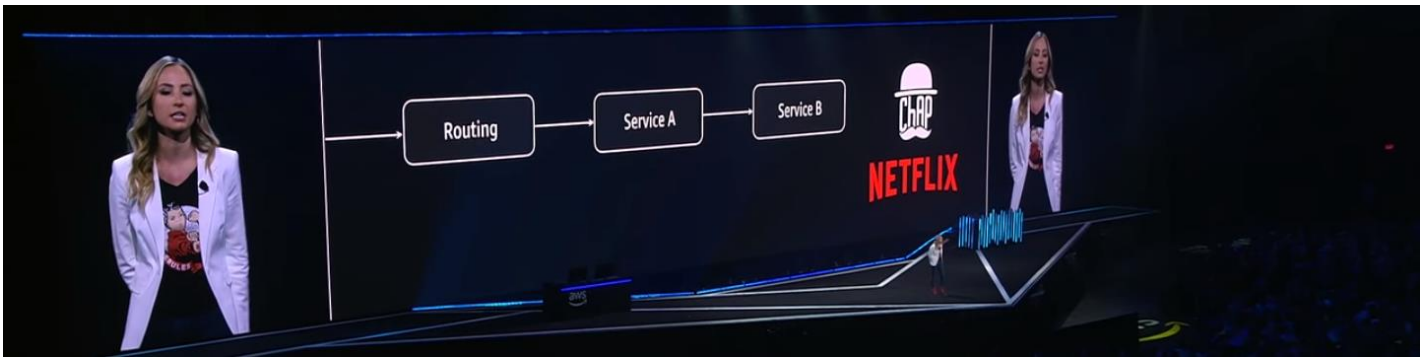


Netflix does failure injection testing where service owners can add latency or failures to the code for resiliency testing if predefined criteria conditions are met

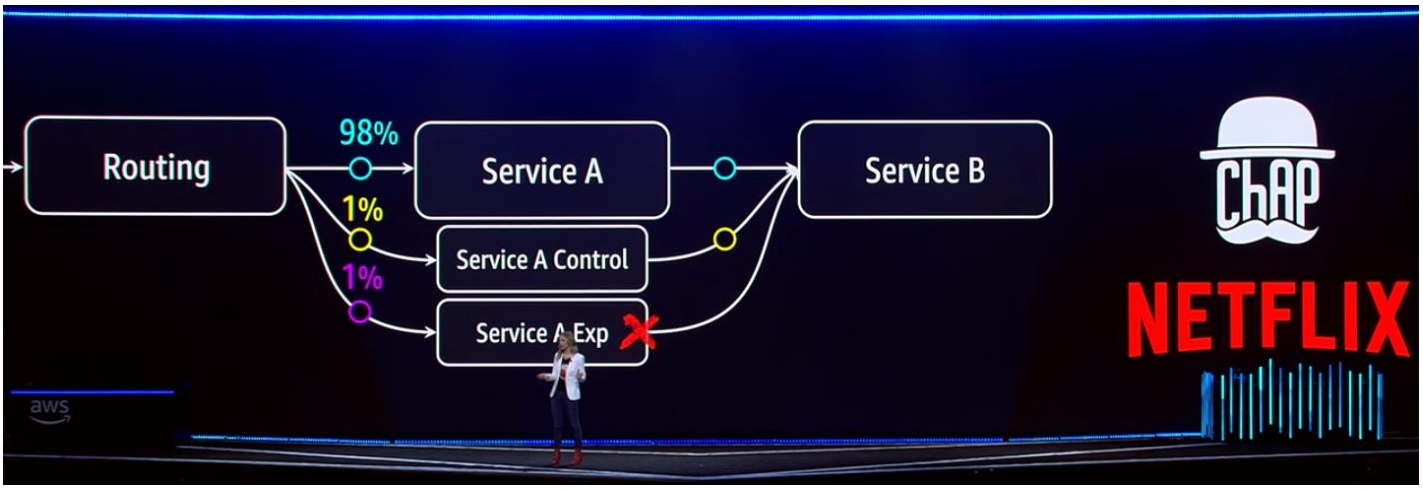


With the failure injection testing framework, Netflix as built an automated chaos monitoring platform called ***Chap*** to determine how much services traffic they are impacting at any moment in time

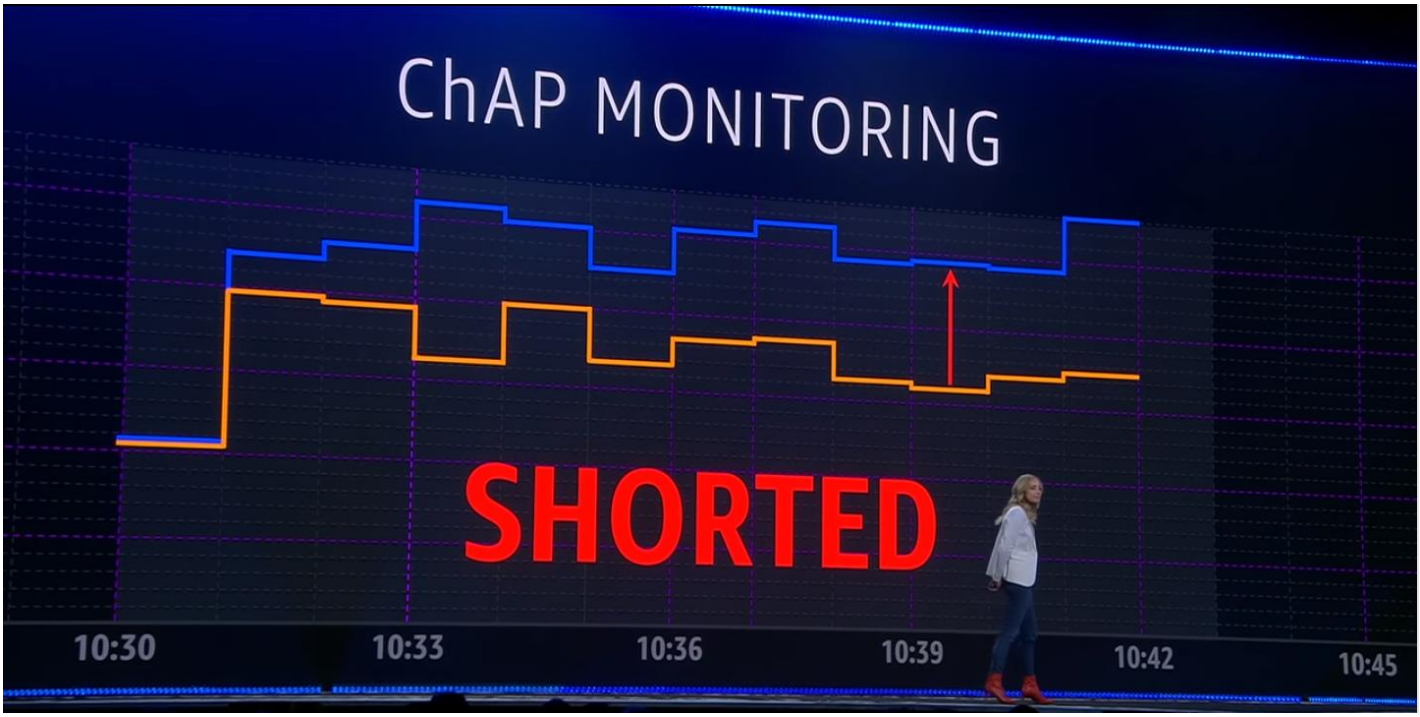




The ***key business metric at Netflix is whether or not the user can press 'Play'*** and nothing starts. This metric is calculated during chaos experiments

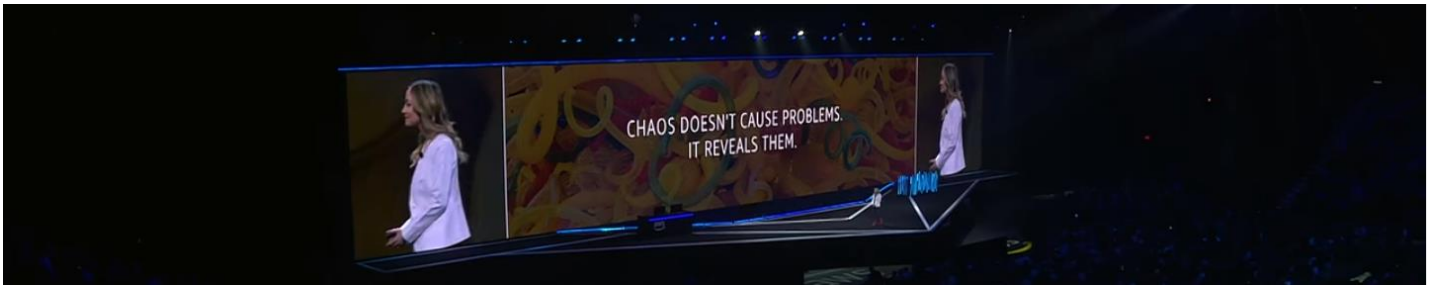


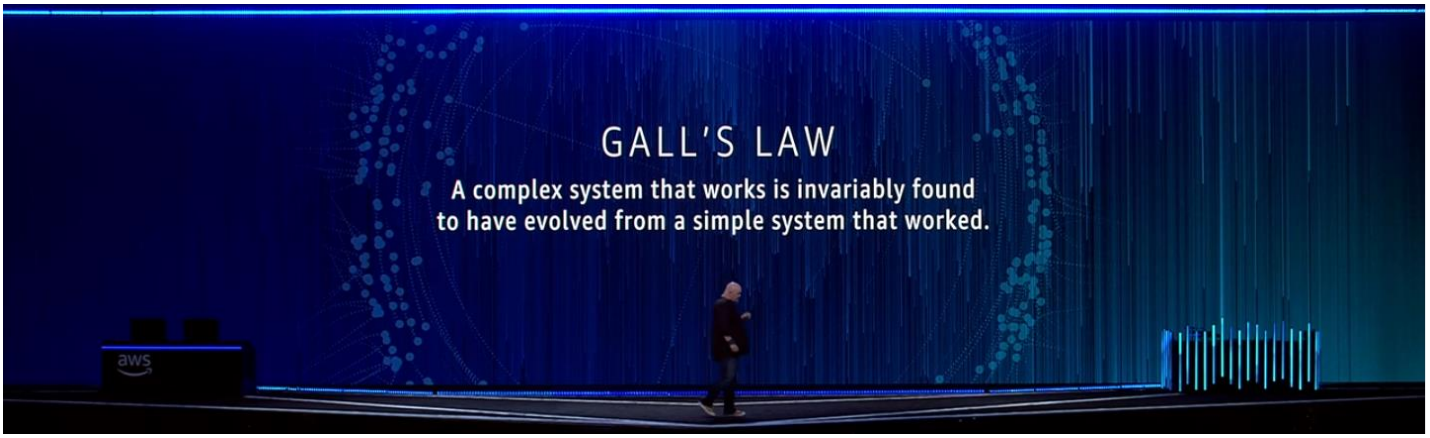
We create 2 clusters for control and experiments and watch the KPI



Too much deviation means that we stop the experiments ongoing and engineers can start to debug the issue

We now have an automated chaos experiments generating algorithm, this determines what services needs to be tested in certain scenarios.

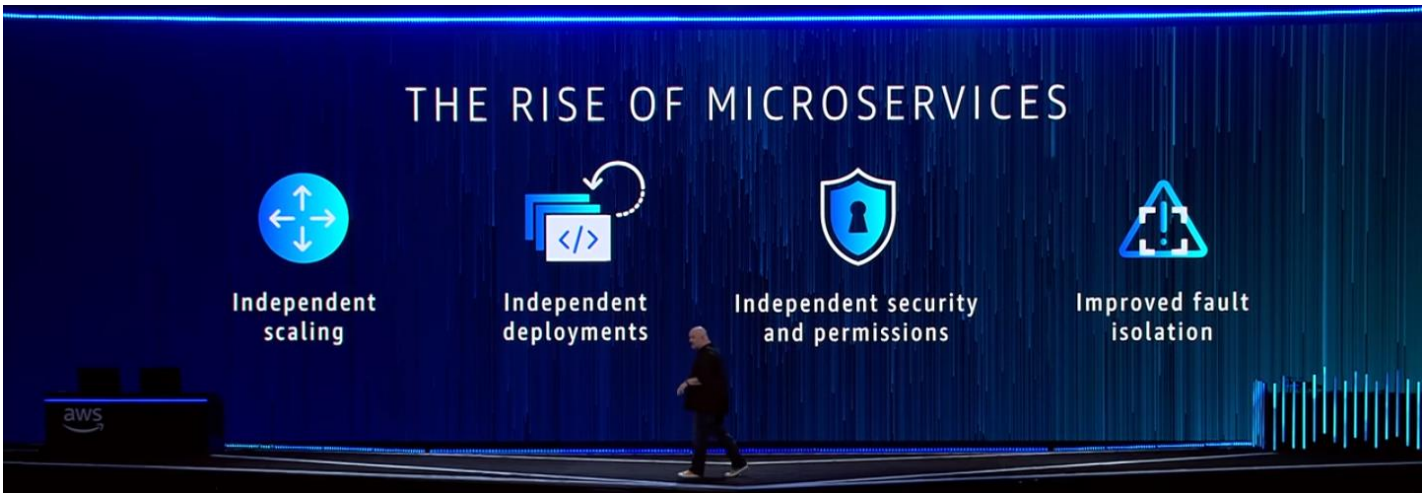




You need to start off by building smaller, simpler systems and then aggregate them into larger a complex system

The more managed services you use, the less moving parts you have regarding the control and data planes


21ST CENTURY ARCHITECTURES
re:Imagined
COMPLETELY MANAGED
CloudFront
ElasticSearch
S3
DATA
Aurora Serverless
DynamoDB
ElastiCache
Amazon Cognito
IDENTITY
Cognito Userpools
ECS
COORDINATION
Step Functions
SYSTEMS DEVELOPMENT
IoT
Mobile
ALB
NETWORK
API Gateway
Amazon MQ
Kinesis
SQS
AWS Lambda
COMPUTE
Containers/ Fargate
CloudWatch
MONITORING & AUDITING
CloudTrail
X-Ray
AWS CodeStar
AWS Cloud9
DEVELOPMENT & DEPLOYMENT
aws


THE RISE OF MICROSERVICES
Independent scaling
Independent deployments
Independent security and permissions
Improved fault isolation
aws


ENABLED BY
CONTAINER TECHNOLOGY
aws


ABBY FULLER
SENIOR TECHNICAL EVANGELIST, AMAZON WEB SERVICES


CONTAINERS ON AWS
Amazon ECS
AWS Fargate for ECS
PREVIEW
Amazon EKS
COMING SOON
AWS Fargate for EKS

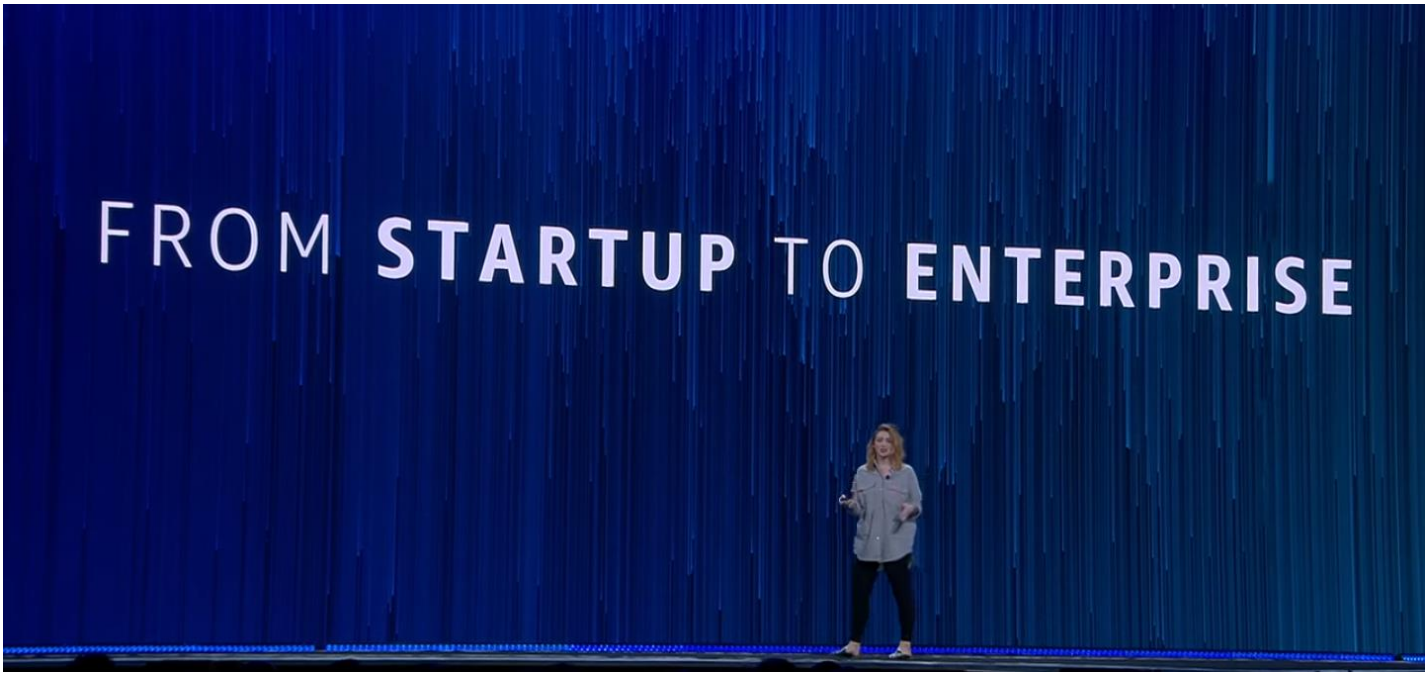

THE POWER IS
IN THE CHOICES


BUT HOW CAN YOU GET STARTED?


LET'S REWIND
On AWS since 2014 with ECS

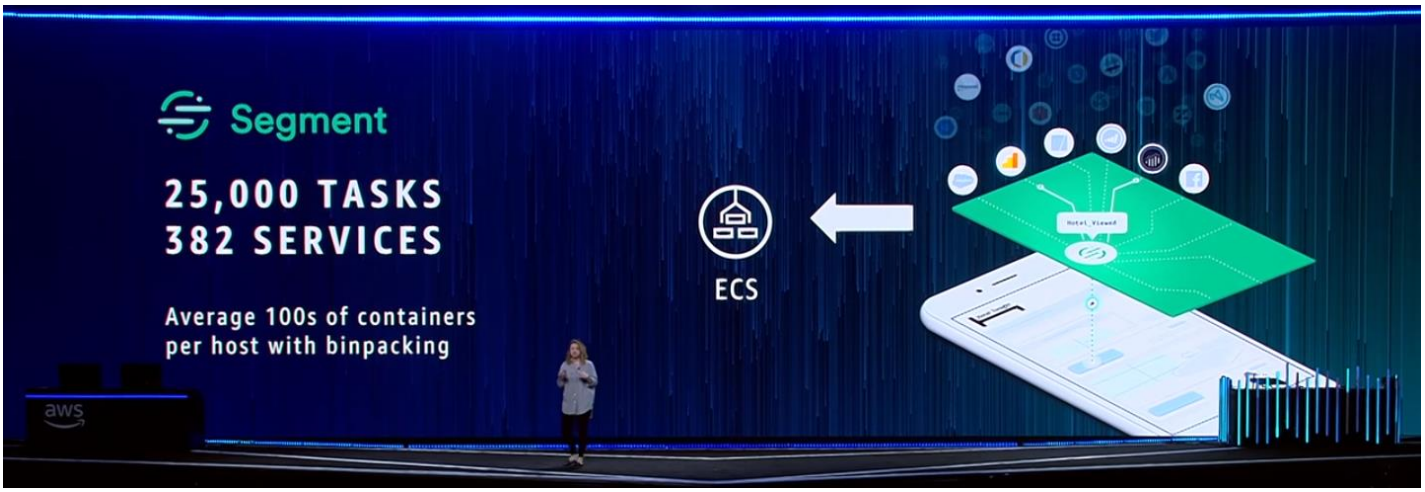

ECS
Highly scalable, high performance container management system
A MANAGED PLATFORM
Cluster Management
Container Orchestration
Deep AWS integration


CUSTOMERS ARE DOING AMAZING THINGS IN PRODUCTION WITH ECS


FROM STARTUP TO ENTERPRISE


CapitalOne
CAPITAL ONE RUNS SERVICES ON
ECS AND DOCKER FOR A FASTER
PACE OF DEVELOPMENT.


Segment
MARKETING DATA STARTUP
MAJORITY OF SERVICES
RUNNING ON ECS

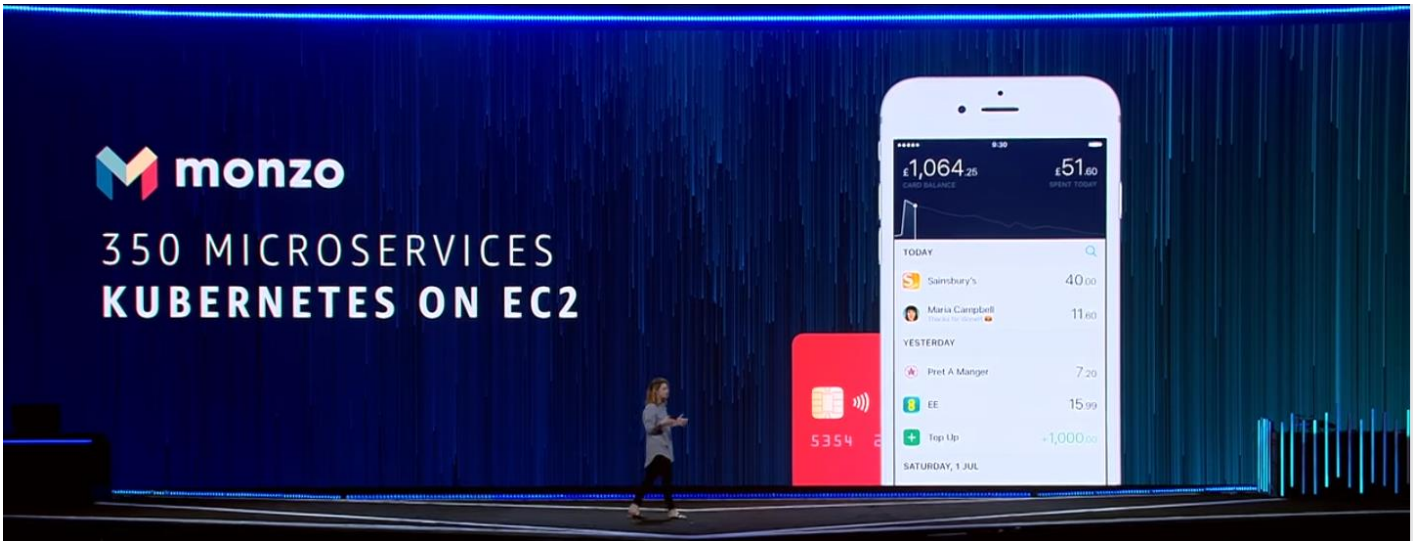

Segment
25,000 TASKS
382 SERVICES
Average 100s of containers
per host with binpacking
ECS


Segment
ECS IS NOT JUST FOR EVENT PROCESSING
ECS
Redshift
RUNNING ETL JOBS ON ECS
aws

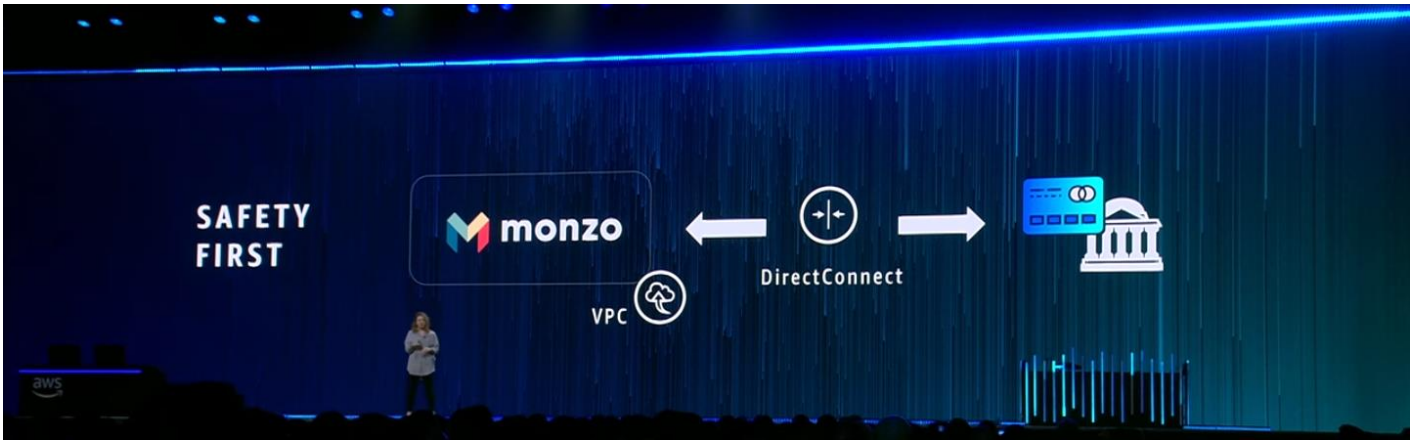

ECS ISN'T THE ONLY WAY
TO RUN CONTAINERS IN PRODUCTION
aws

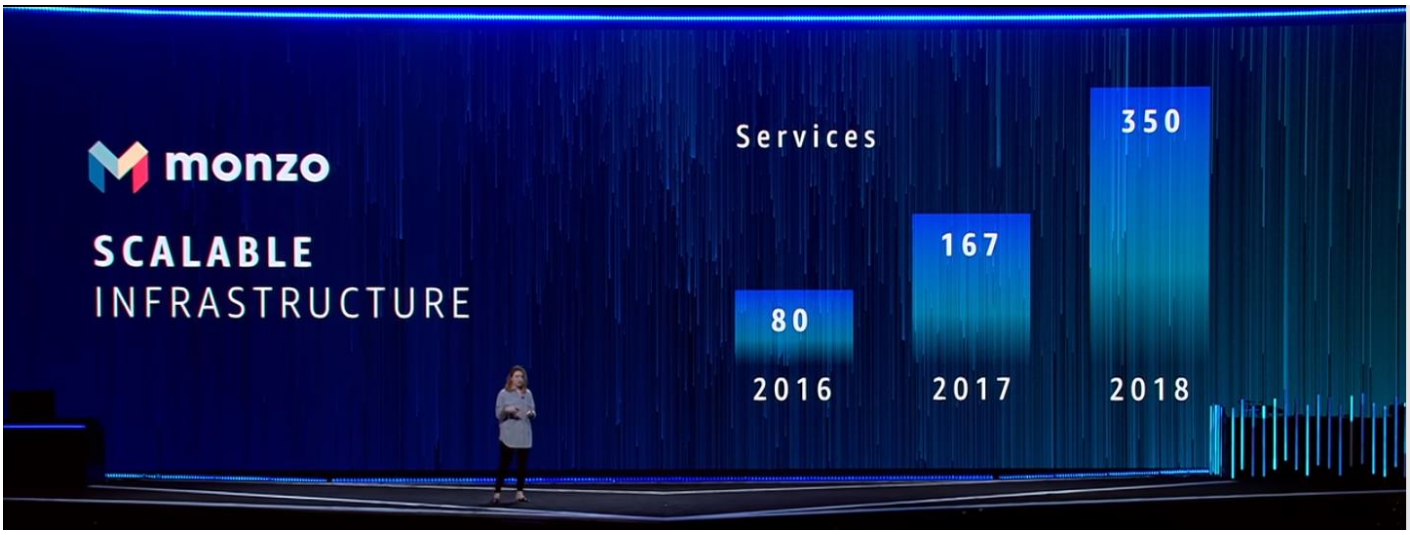

MORE CUSTOMERS RUNNING
KUBERNETES ON AWS THAN
ANYWHERE ELSE
aws


monzo
FINTECH STARTUP
KUBERNETES ON AWS


monzo
350 MICROSERVICES
KUBERNETES ON EC2
£1,064.25
CARD BALANCE
£51.60
SPENT TODAY
TODAY
Sainsbury's
40.00
Maria Campbell
11.60
YESTERDAY
Pret A Manger
7.20
EE
15.99
Top Up
+1,000.00
SATURDAY, 1 JUL
5354

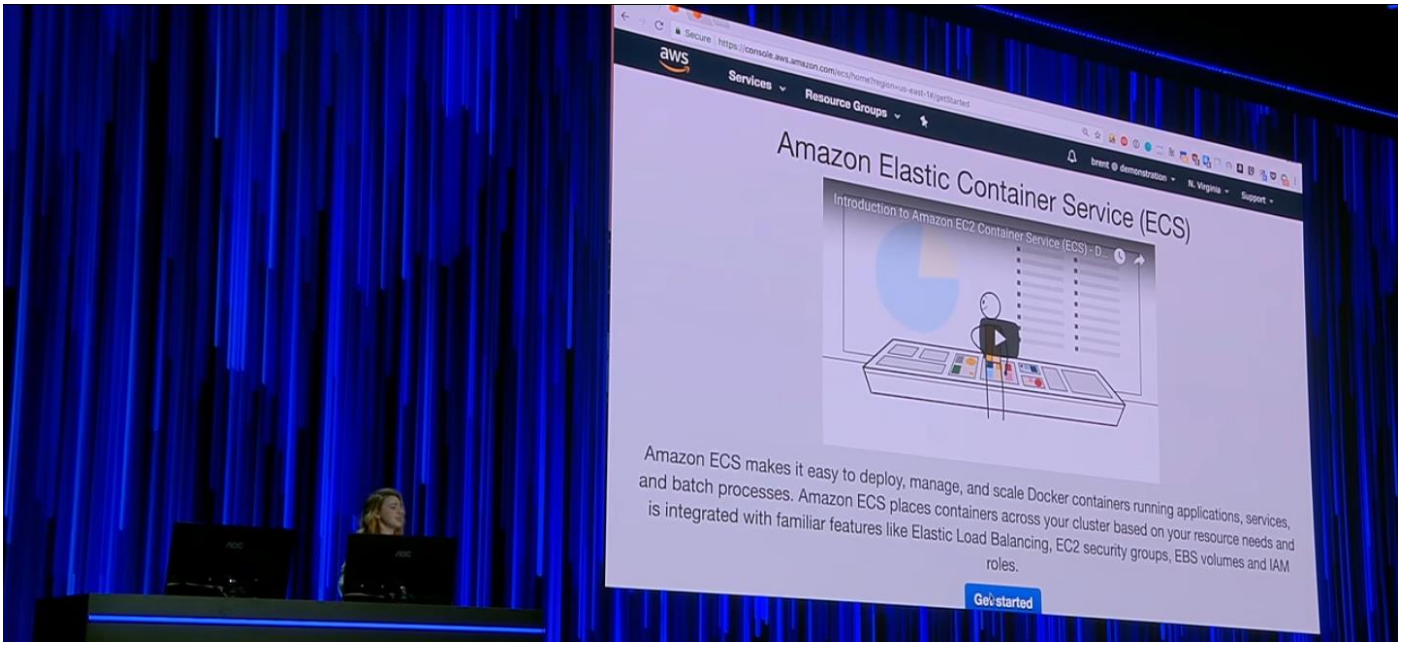

monzo
HIGH AVAILABILITY
AT EVERY LEVEL
Multi-Master
Hundreds of Workers
High Availability etcd


SAFETY
FIRST
monzo
VPC
DirectConnect

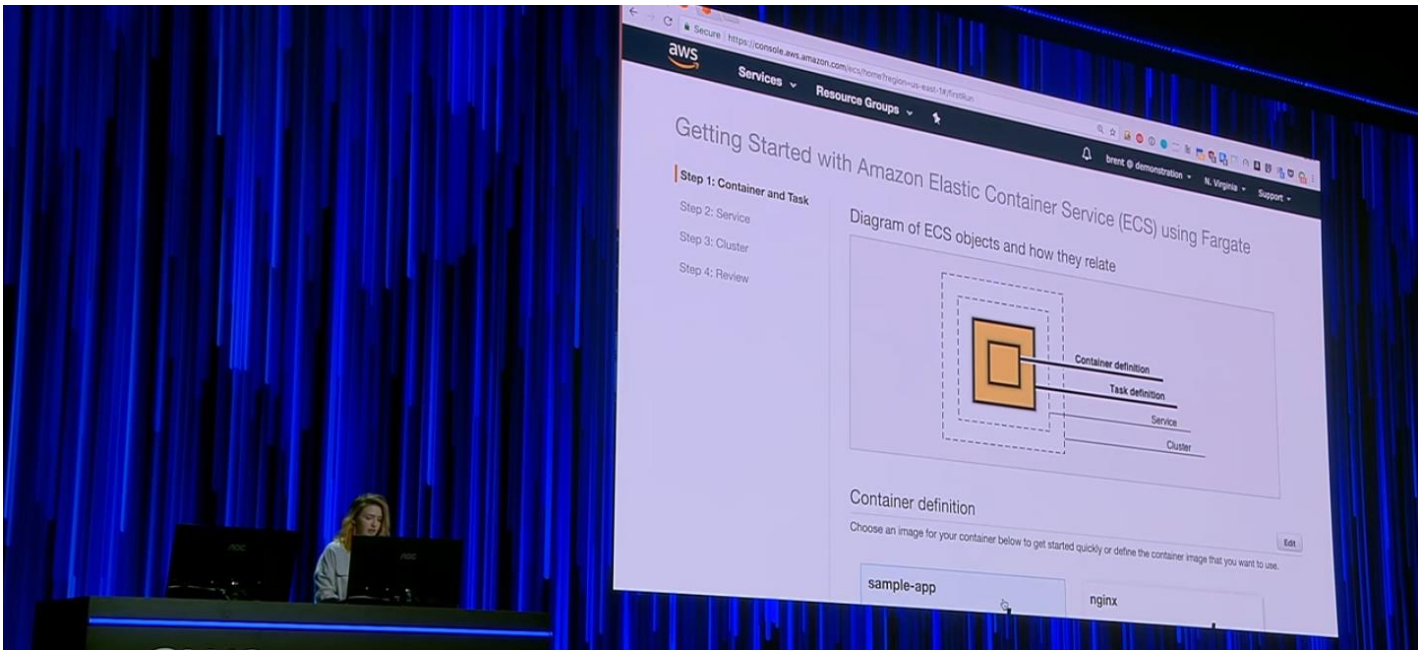

monzo
SCALABLE
INFRASTRUCTURE
Services
80
167
350
2016
2017
2018

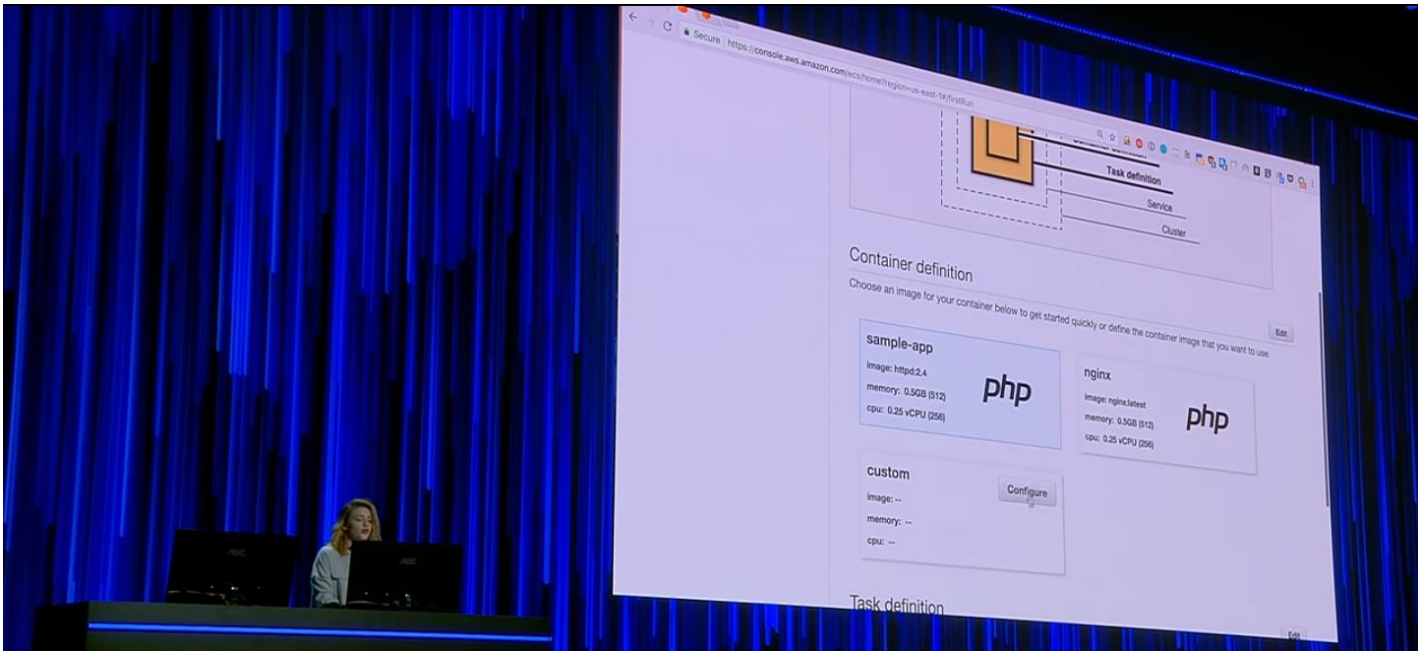

"RUN KUBERNETES FOR ME"

AWS Fargate is not a service but is an underlying technology that you can use to power your container workloads. Just take your task definition or Pod, drop it into Fargate, add some resources, and away you go.

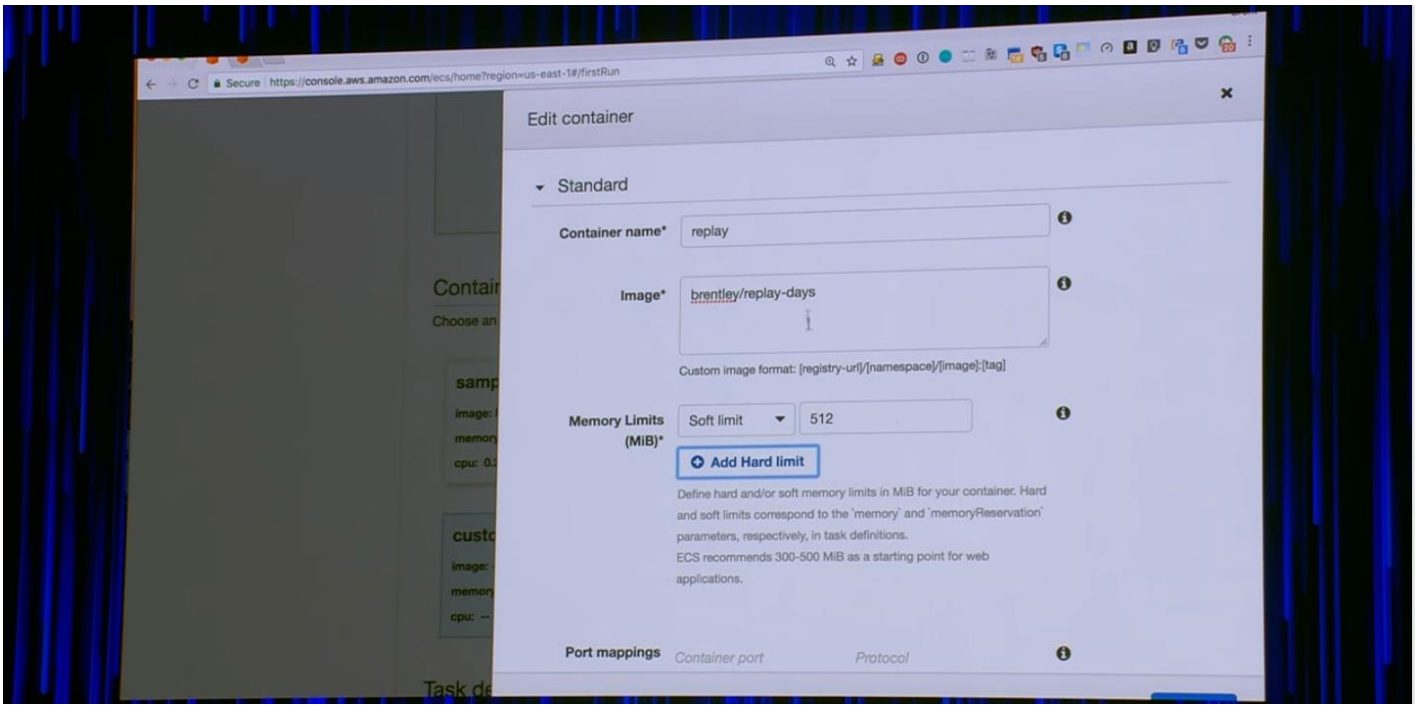


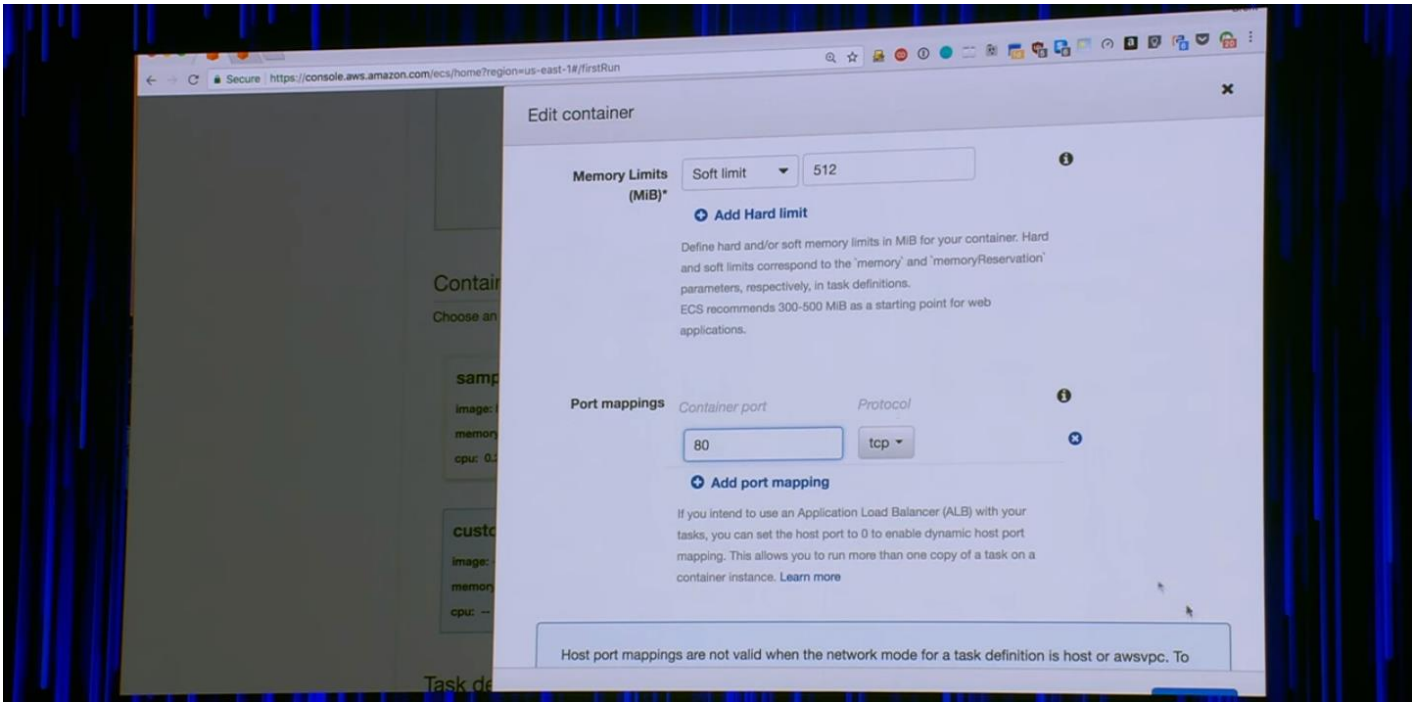
Let’s see a demo of how to deploy an app and see Fargate in action.

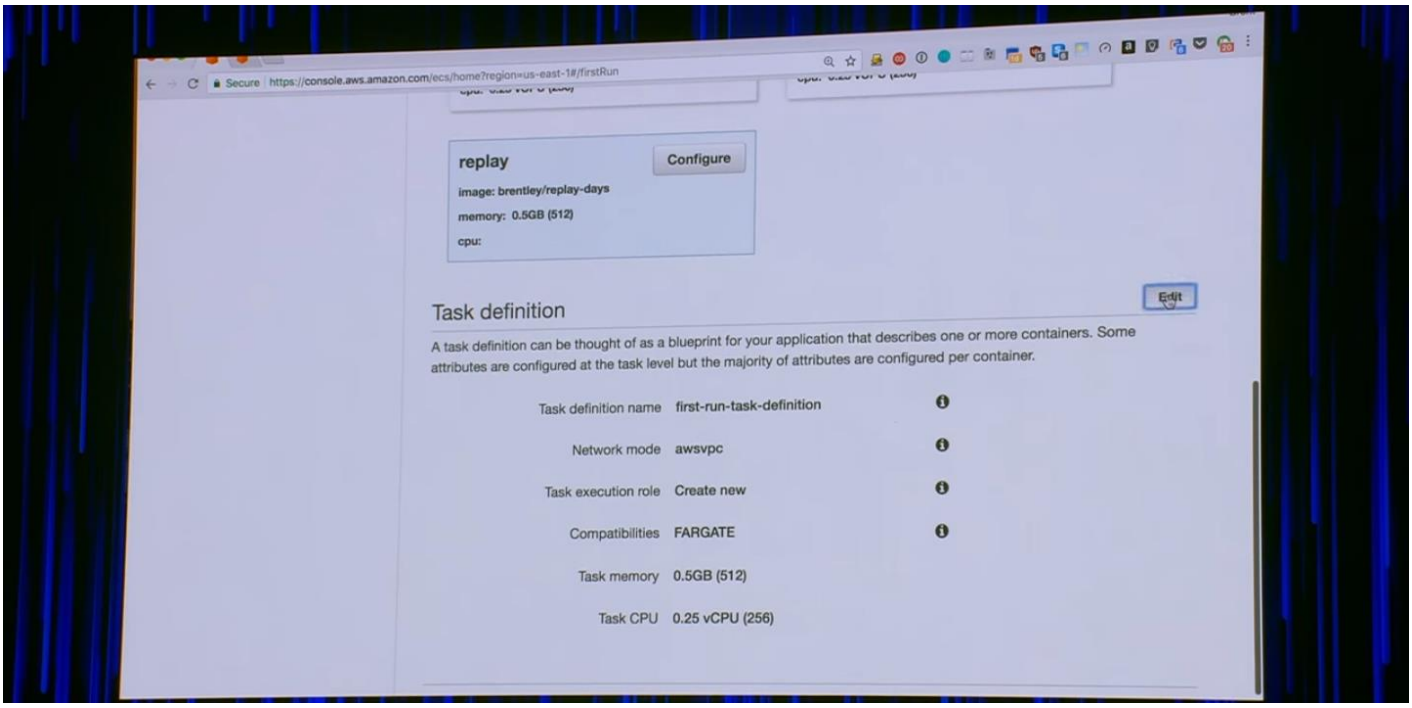


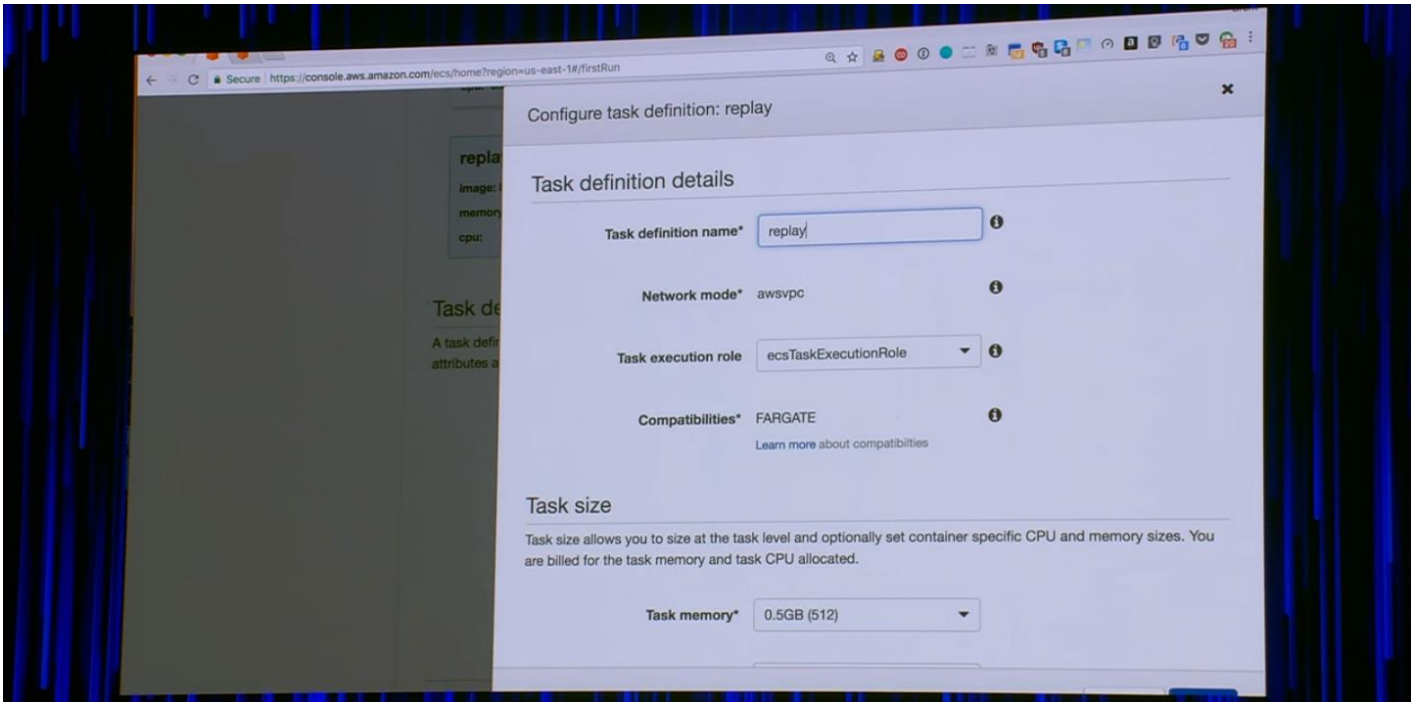


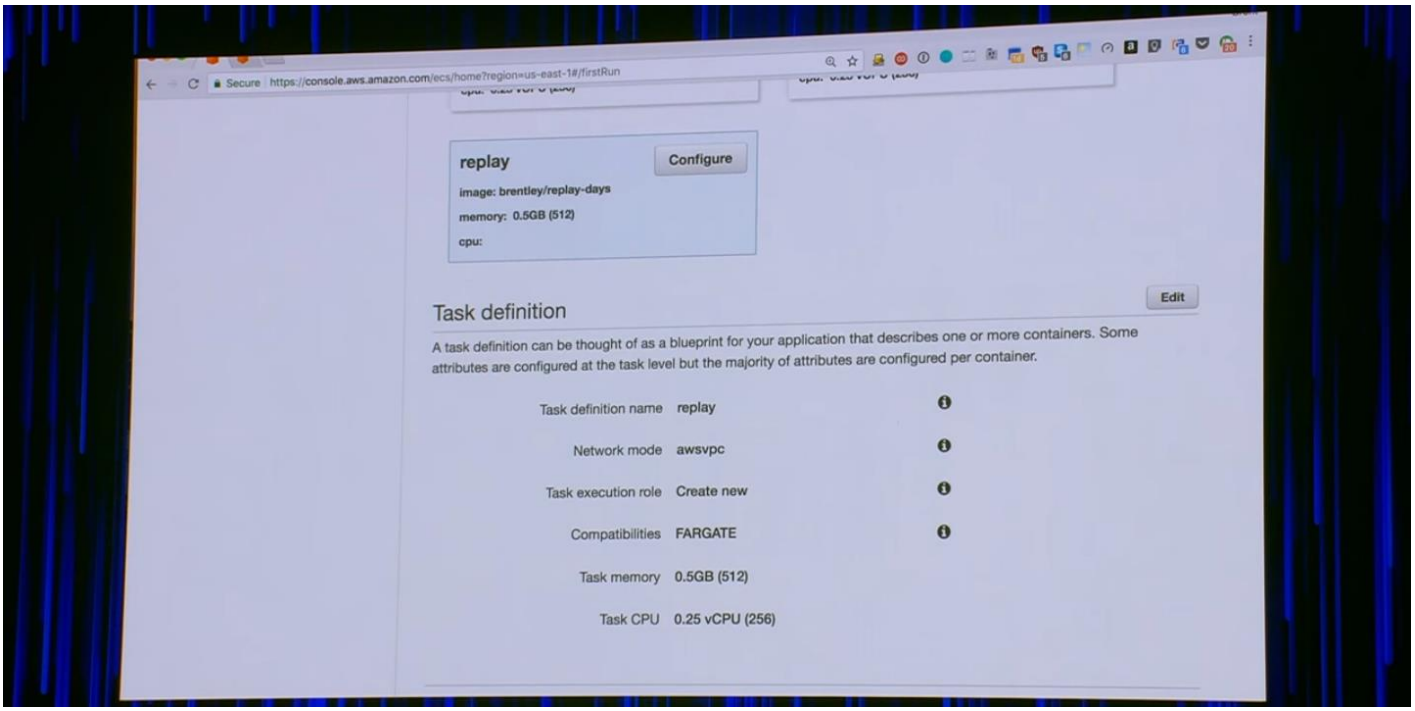
We start with our task, within the Fargate experience

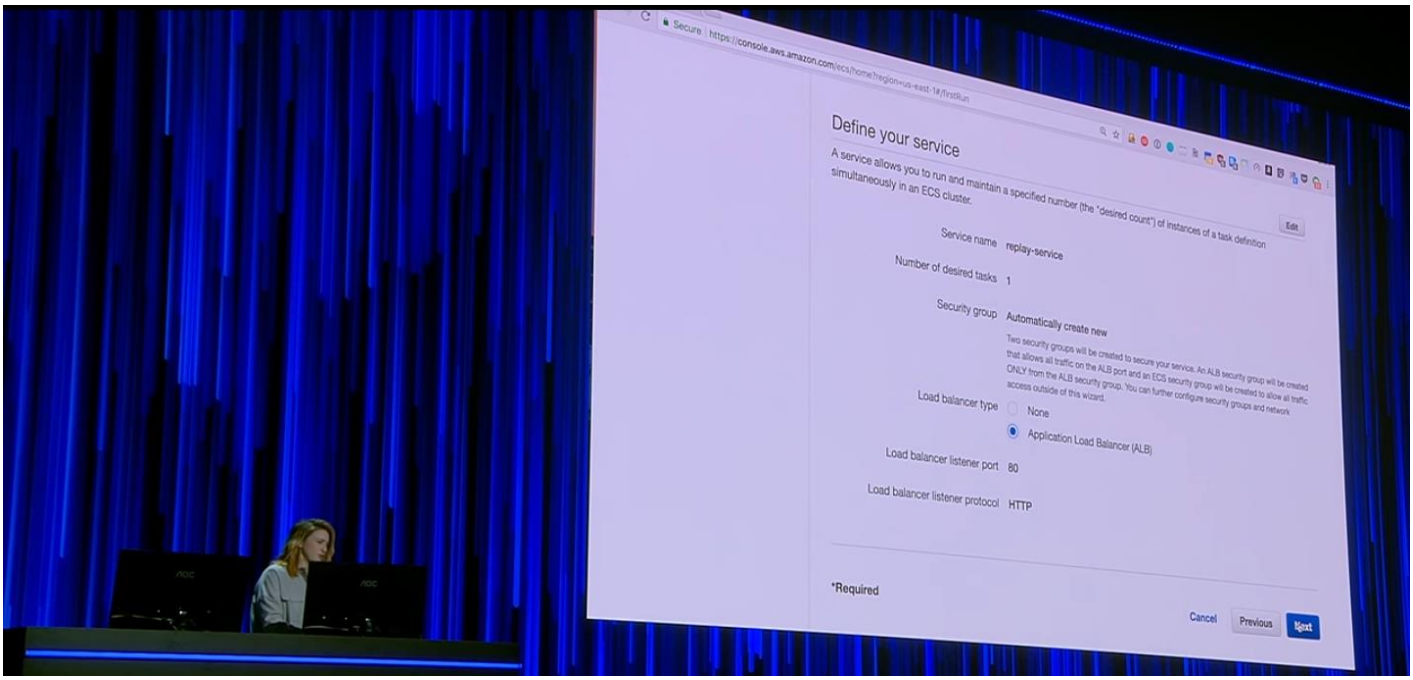

Secure https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
Edit container
Standard
Container name*
replay
Image*
brentley/replay-days
Custom image format: [registry-url]/[namespace]/[image]:[tag]
Memory Limits (MiB)*
Soft limit
512
Add Hard limit
Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions.
ECS recommends 300-500 MiB as a starting point for web applications.
Port mappings
Container port
Protocol

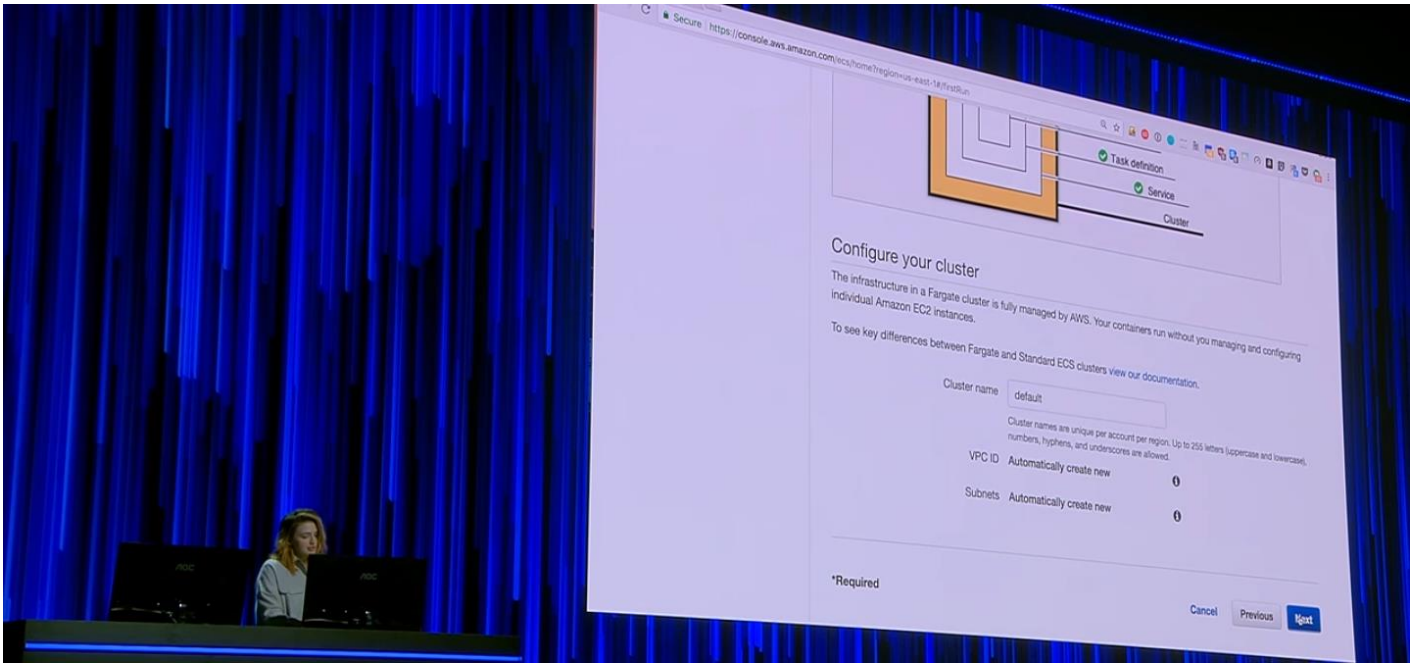

Secure https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
Edit container
Memory Limits (MiB)*
Soft limit
512
Add Hard limit
Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions.
ECS recommends 300-500 MiB as a starting point for web applications.
Port mappings
Container port
Protocol
80
tcp
Add port mapping
If you intend to use an Application Load Balancer (ALB) with your tasks, you can set the host port to 0 to enable dynamic host port mapping. This allows you to run more than one copy of a task on a container instance. Learn more
Host port mappings are not valid when the network mode for a task definition is host or awsvpc. To

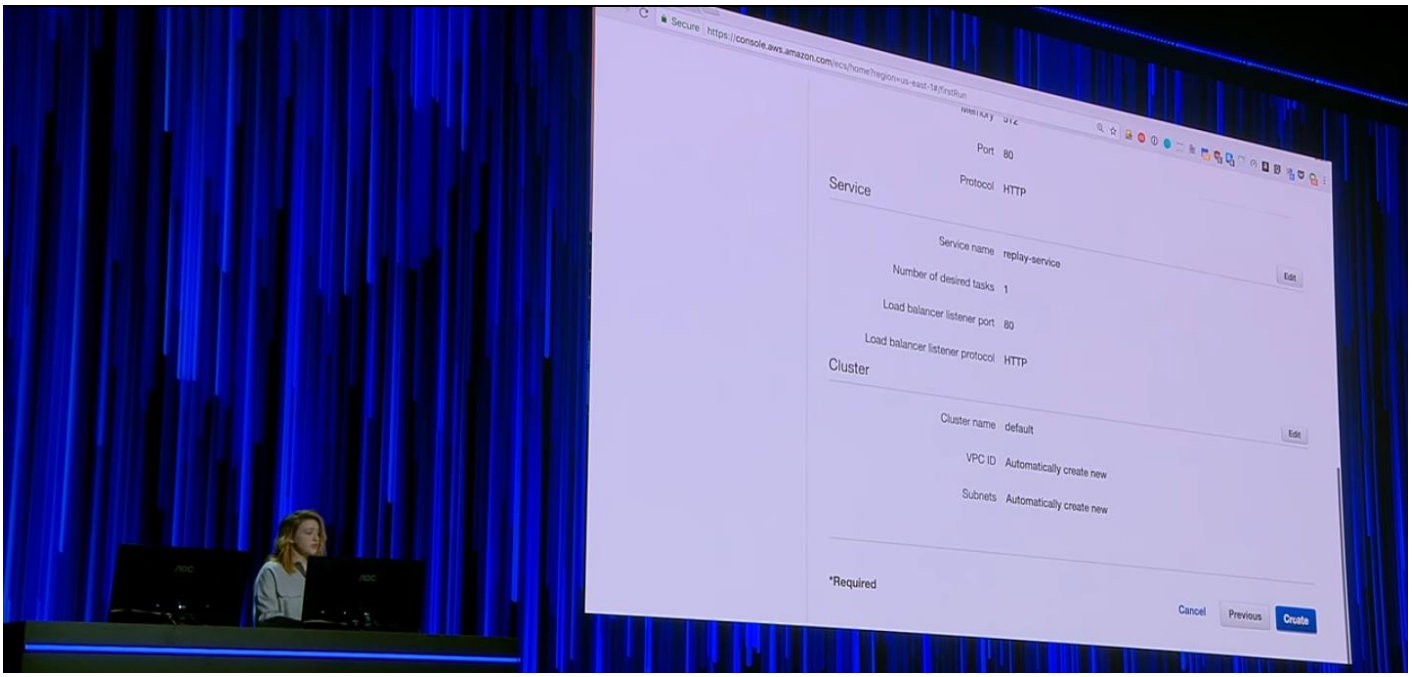

Secure https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
replay
Configure
image: brentley/replay-days
memory: 0.5GB (512)
cpu:
Task definition
Edit
A task definition can be thought of as a blueprint for your application that describes one or more containers. Some attributes are configured at the task level but the majority of attributes are configured per container.
Task definition name first-run-task-definition
Network mode awsvpc
Task execution role Create new
Compatibilities FARGATE
Task memory 0.5GB (512)
Task CPU 0.25 vCPU (256)

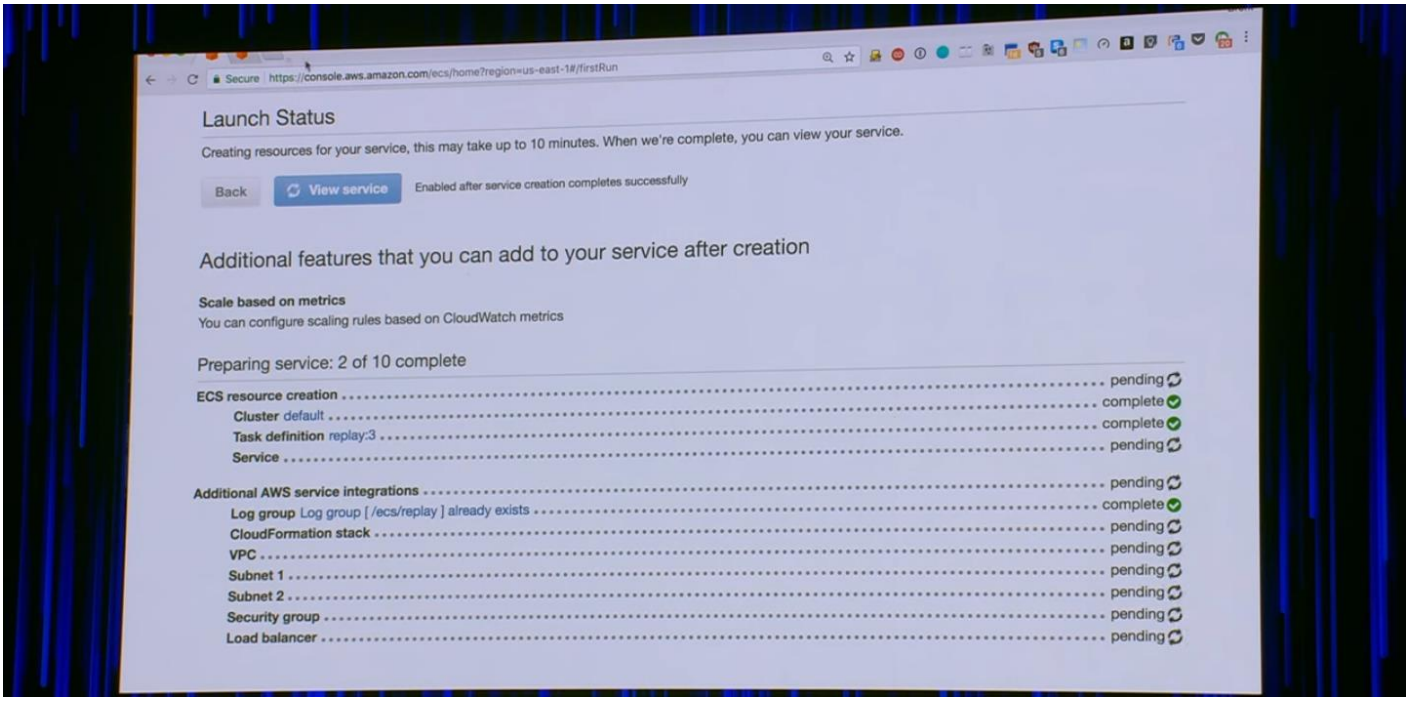

Secure https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
Configure task definition: replay
Task definition details
Task definition name* replay
Network mode* awsvpc
Task execution role ecsTaskExecutionRole
Compatibilities* FARGATE
Learn more about compatibilties
Task size
Task size allows you to size at the task level and optionally set container specific CPU and memory sizes. You are billed for the task memory and task CPU allocated.
Task memory* 0.5GB (512)

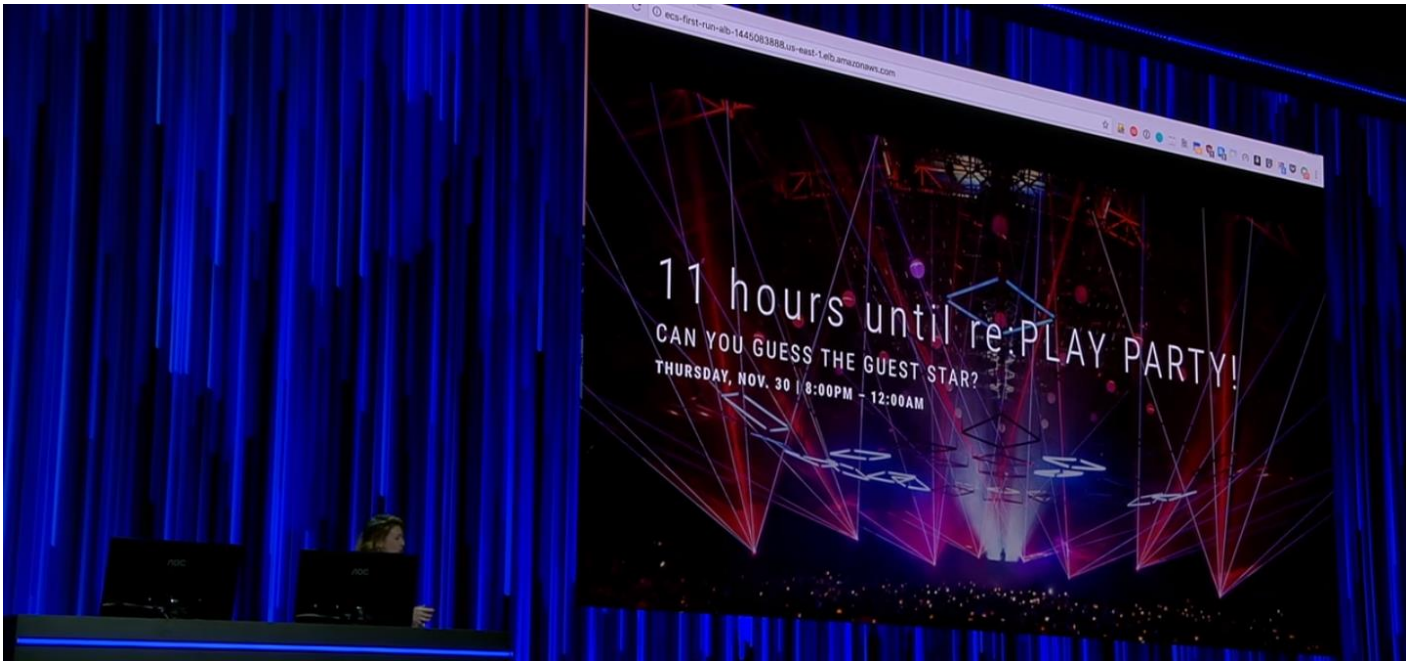

Secure https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
replay
Configure
image: brentley/replay-days
memory: 0.5GB (512)
cpu:
Task definition
Edit
A task definition can be thought of as a blueprint for your application that describes one or more containers. Some attributes are configured at the task level but the majority of attributes are configured per container.
Task definition name replay
Network mode awsvpc
Task execution role Create new
Compatibilities FARGATE
Task memory 0.5GB (512)
Task CPU 0.25 vCPU (256)


Secure https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
Define your service
A service allows you to run and maintain a specified number (the "desired count") of instances of a task definition simultaneously in an ECS cluster.
Edit
Service name replay-service
Number of desired tasks 1
Security group Automatically create new
Two security groups will be created to secure your service. An ALB security group will be created that allows all traffic on the ALB port and an ECS security group will be created to allow all traffic ONLY from the ALB security group. You can further configure security groups and network access outside of this wizard.
Load balancer type None
Application Load Balancer (ALB)
Load balancer listener port 80
Load balancer listener protocol HTTP
*Required
Cancel
Previous
Next

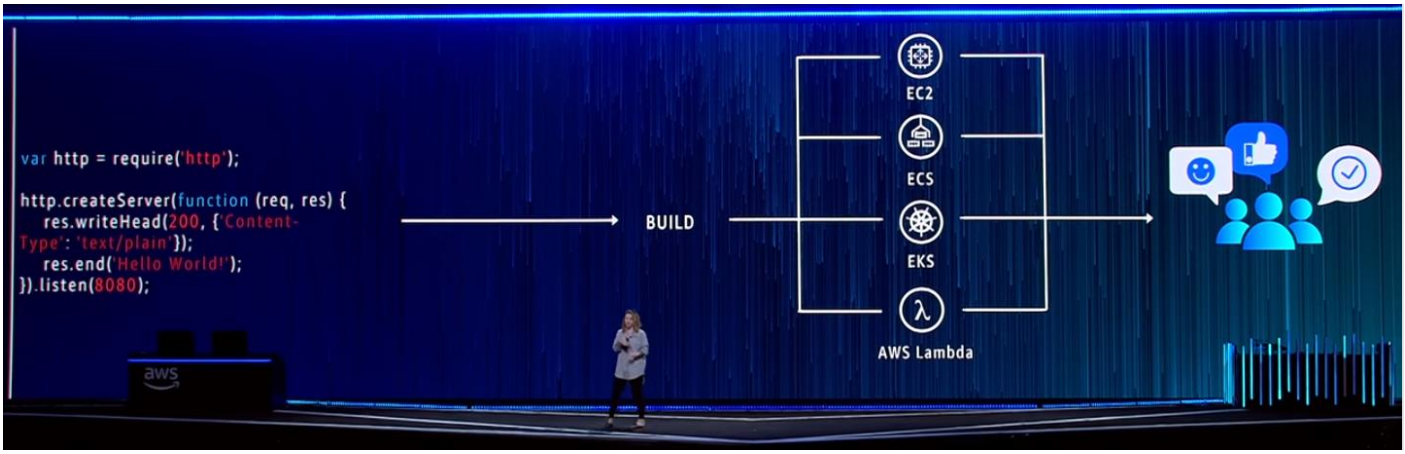

Secure https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
Task definition
Service
Cluster
Configure your cluster
The infrastructure in a Fargate cluster is fully managed by AWS. Your containers run without you managing and configuring individual Amazon EC2 instances.
To see key differences between Fargate and Standard ECS clusters view our documentation.
Cluster name default
Cluster names are unique per account per region. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.
VPC ID Automatically create new
Subnets Automatically create new
*Required
Cancel
Previous
Next


Secure https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
Port 80
Protocol HTTP
Service
Service name replay-service
Edit
Number of desired tasks 1
Load balancer listener port 80
Load balancer listener protocol HTTP
Cluster
Cluster name default
Edit
VPC ID Automatically create new
Subnets Automatically create new
*Required
Cancel
Previous
Create





The app is now deployed.



AWS Fargate lets you focus on the workload only

We will be able to build with a couple of basic primitive – the container


WORKLOAD FIRST, FULL STOP.