

Paul Underwood, AWS  
John Keilty, Third Rock Ventures

# AWS re:Invent

Born in the Cloud  
Built like a Startup

November 28, 2017

AWS  
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



**How to think like a startup when deploying  
your next workload on AWS.**

**Whether you work for an enterprise or a  
small business.**

## My expectations

- 100 Level AWS constructs
- Familiarity with AWS services



**AWS**  
**re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



## My expectations

- 100 Level AWS constructs
- Familiarity with AWS services

## Architectures we will cover:

- N-tier
- Containerized
- Serverless

## Implications on:

- Cost
- Performance
- Team structure

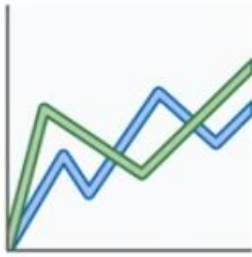
**Third Rock Ventures: 44 startups in 10 Years**

**AWS**  
**re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



What are startups thinking about?



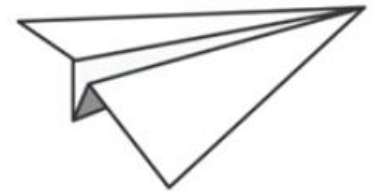
Expecting  
scale



Focus on  
features

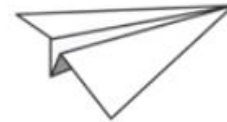
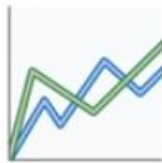


Lean IT  
department



Low cost ==  
Long runway

**Isn't everybody?**



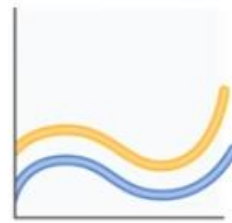
# What else are we thinking about?



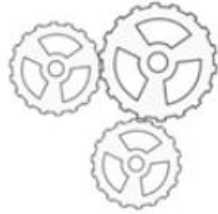
Reliability



Performance



Cost efficiency



Operational  
excellence



Security

**AWS  
re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



## Can building like a startup align these goals?



Expecting  
scale



Focus on  
features



Lean IT  
department



Low cost ==  
Long runway



Reliability



Performance



Cost efficiency



Security



Operational  
excellence



**AWS  
re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



# OK, so how have people done this historically?



express



METEOR



## Using these kinds of frameworks



AWS  
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



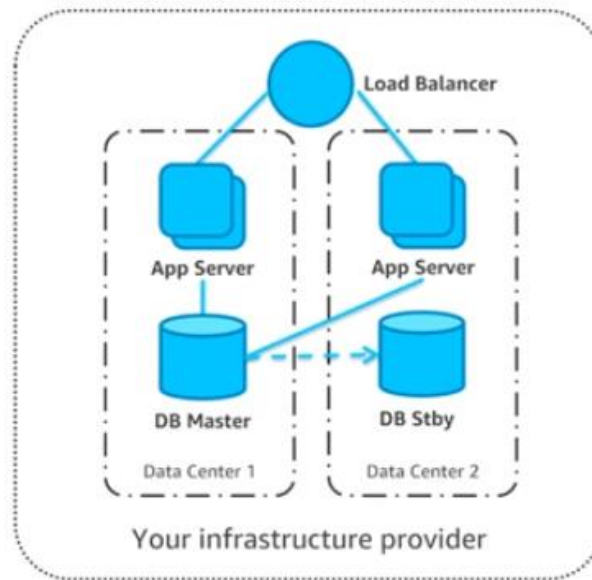
web development,  
one drop at a time

django





# How? Monolithic/N-tier architectures



AWS  
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

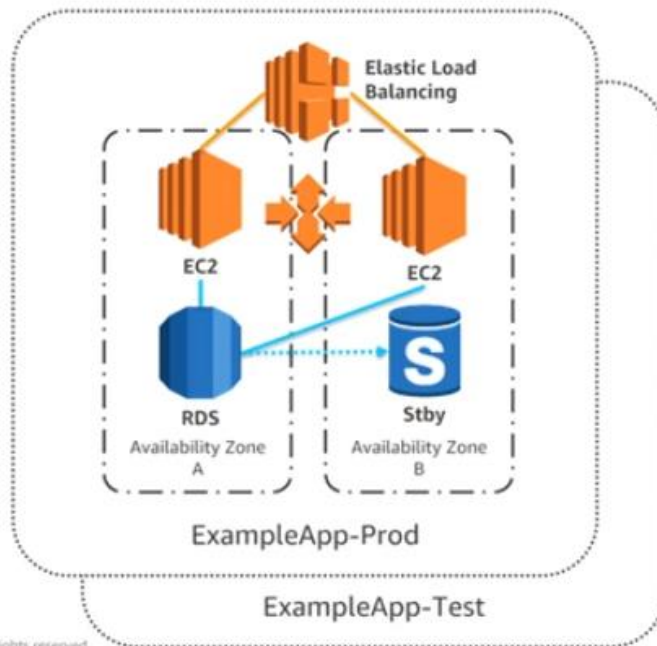


## How do startups do this on AWS?

```
git clone git://myrepo && cd myrepo
eb init
eb create prod
pg_restore -v -h mydb.rds.amazonaws.com latest.dump
eb setenv SHARED_KEY_OF_SOME_SORT=34dsa...2x32vxj
/// Changes
eb create test
git add . && git commit -m "profound change"
eb deploy
/// Test
eb switch prod && eb deploy
```

Typically using an Elastic Beanstalk ELB CLI environment to launch the apps for us automatically.

# This gets us VM-based N-tier on AWS



**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

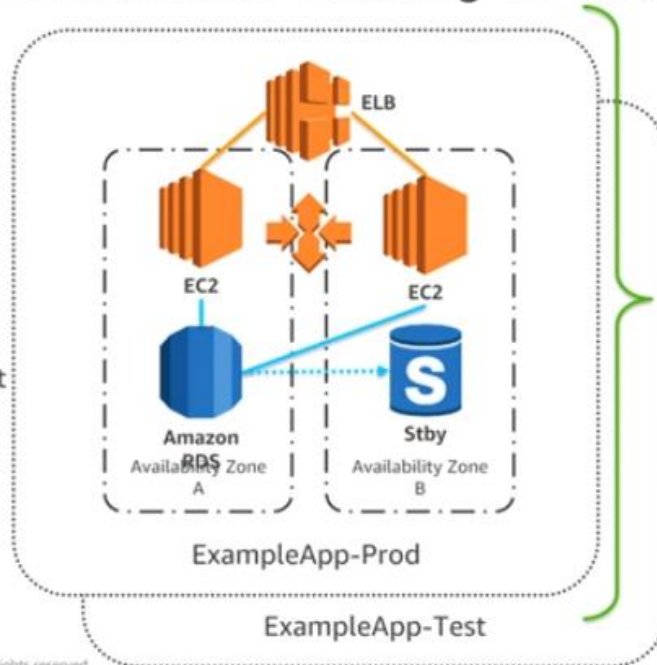


## AWS Elastic Beanstalk: Making N-tier easier

- AWS Security**
  - Identity and Access Management
  - VPC networking
- AWS CloudWatch**
  - Service-wide resource monitoring
  - Log management
- Specialist Services**
  - Block/object storage
  - Caching
  - DNS

**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



The quickest path to best practices



- AWS Elastic Beanstalk**
  - Builds you into best practices from the start
  - Integrates with developer workflows
  - Use the Elastic Beanstalk Command Line Interface



This all can also be done using the same elastic beanstalk N-tier architecture environment ELB CLI inside your scripts without using the console.

# What does this cost?

## “Development-grade” stack

Tier	Spec	Monthly cost
Load Balancer	1x	\$18.30
Application server	1 x t2.micro	\$9.52
Database server	1 x t2.micro 100 GB	\$23.95
Total monthly		\$51.77

## “Production-grade” stack

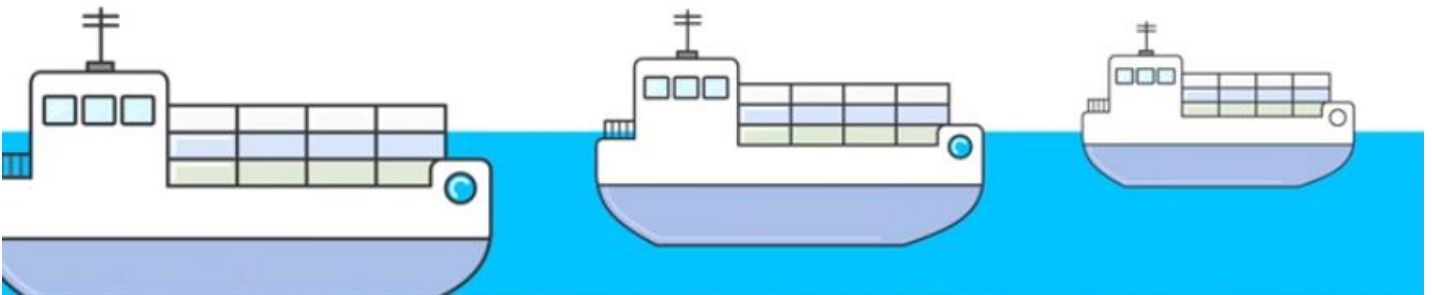
Tier	Spec	Monthly cost
Load Balancer	1x	\$18.30
Application server	2 x m4.large	\$121.18
Database server	2 x m4.large 100 GB	\$198.93
Total monthly		\$320.11

**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



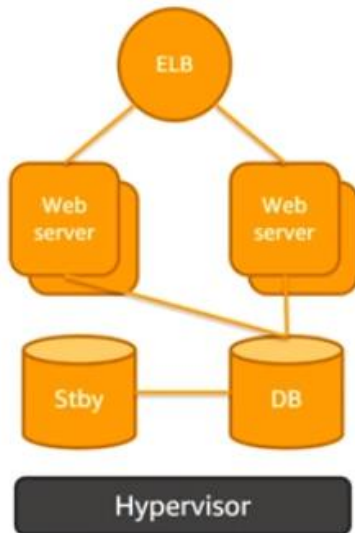
## Containerized architectures





Containers in theory look like conventional N-tier

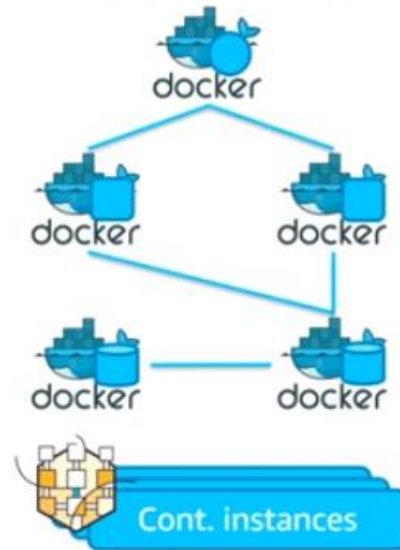
VM-based N-tier:



AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

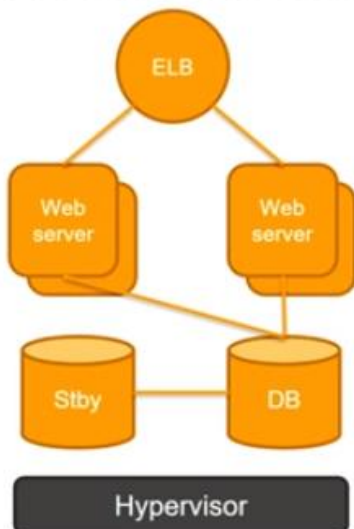
Container-based N-tier:



aws

In practice, leverage the platform...

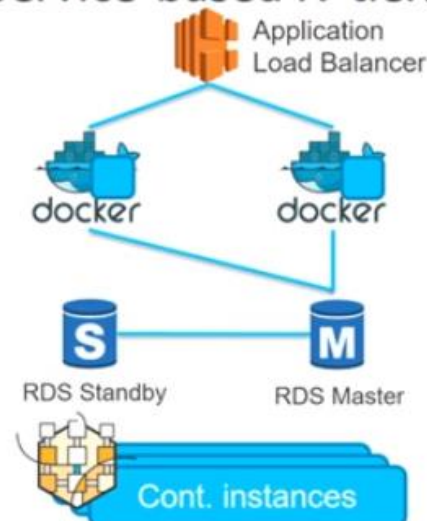
VM-based N-tier:



AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Amazon EC2 Container Service-based N-tier:



aws

# How startups are building containers on AWS

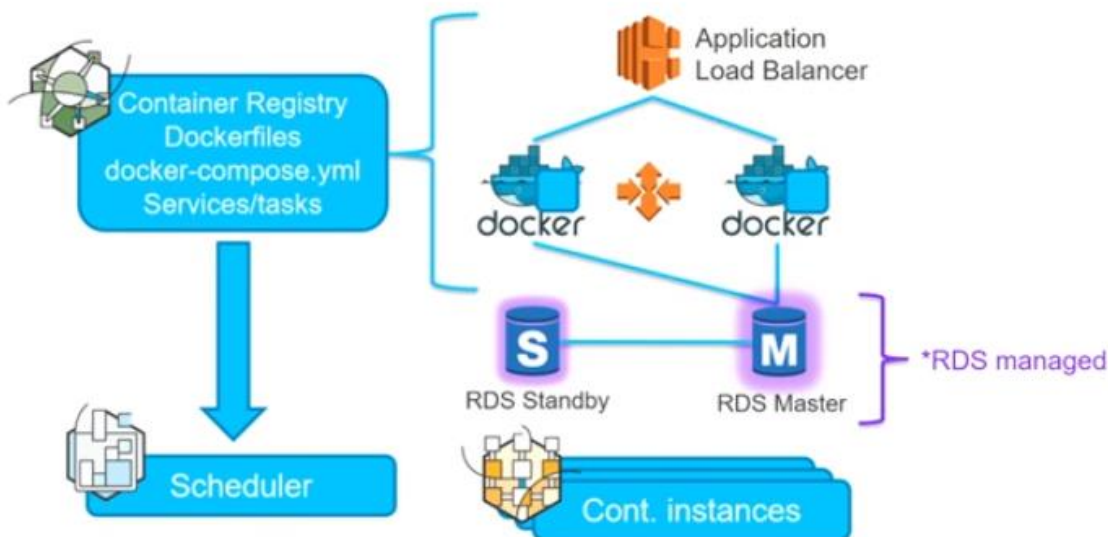
```
## set up
aws ecs get-login
docker build -t <tagName> .
docker tag <tagName>:latest <repoUrl>/<tagName>:latest

ecs-cli configure --region us-west-2 --cluster <clusterName>
ecs-cli up --keypair <keyPairID> --capability-iam --size 2 --type ...

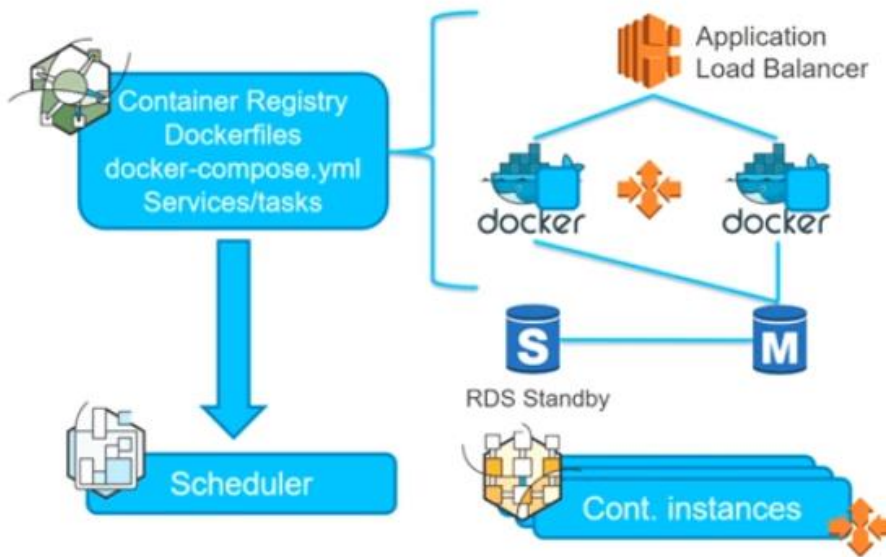
## auto-generate service and task definition, no ALB, no ASG
ecs-cli compose service create --file docker-compose.yml
ecs-cli compose service start
## instead, use ECS to define more sophisticated services
aws ecs create-service --service-name <serviceName> --cli-input-json
file://sophisticated-service-def.json
```

We can use the ECS CLI to do the following to set up our stack.

## What does ECS give us?



# But what about?



-  Service discovery
-  Container instance scaling
-  Container-level logging

**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



## What does this cost?

“Production-grade” EC2 stack  
(40% utilization)

Tier	Spec	Monthly cost
Load Balancer	1x	\$18.30
Application server	2 x m4.large	\$121.18
Database server	2 x m4.large 100 GB	\$198.93
Total monthly		\$320.11

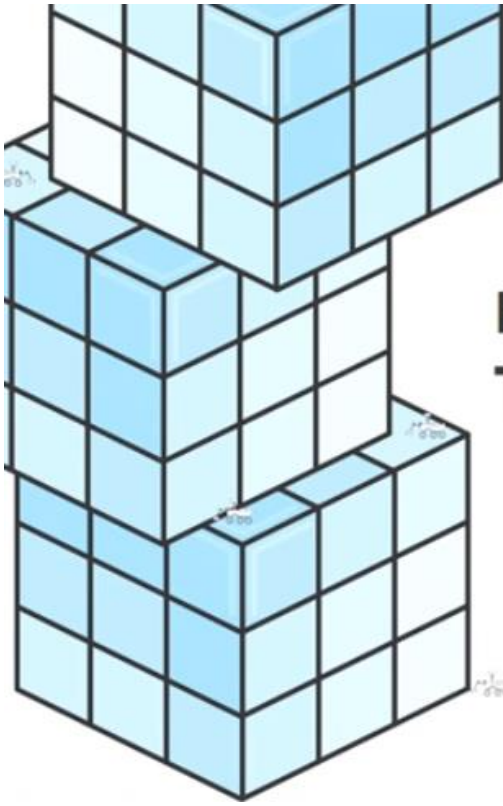
“Production-grade” ECS stack  
(80% utilization)

Tier	Spec	Monthly cost
Load Balancer	1x	\$18.30
Container instances	2 x m4.large	\$121.18
Database server	2 x m4.large 100 GB	\$198.93
Total monthly		\$320.11

**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.





## N-tier/Container DevOps The Stack Challenge

Opinion Time:  
Traditional VM and Container  
architectures are rooted in  
emulating classic physical  
servers...



... and therefore inherit the Stack Challenge

### Who's responsibility is?

- Server-level configuration
  - Packages/dependencies
  - Users/groups
  - Build sources
  - Files
  - Bootstrapping commands
  - Services
  - Security
- Cluster-level configuration
  - Container instances
  - Supporting core services

**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



## The Stack Challenge

### Who's responsibility is?

- Server-level configuration
  - Packages/dependencies
  - Users/groups
  - Build sources
  - Files
  - Bootstrapping commands
  - Services
  - Security
- Cluster-level configuration
  - Container instances
  - Supporting core services

**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Tooling can get you so far...

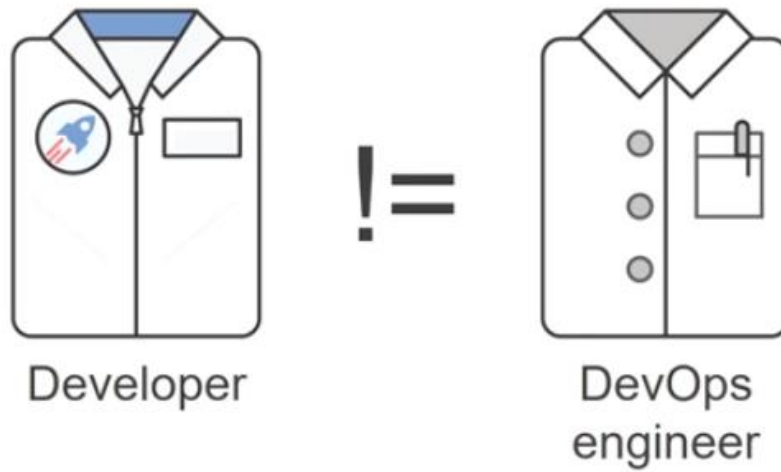


Eventually, you need devops staff





# Why?

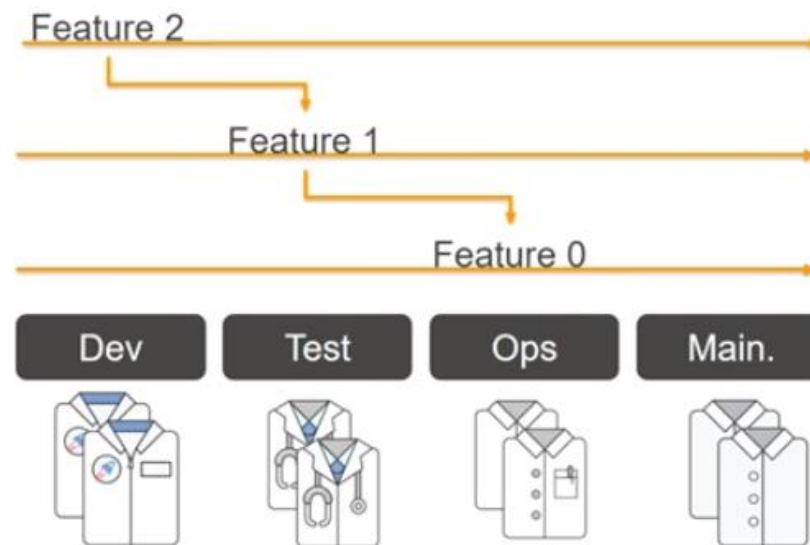


**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



## What is that rushing sound?



**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



# Meanwhile, back at the lab...



Thinking big, inventing, simplifying  
Traditional VM and container architectures are rooted in  
emulating classic physical servers.

Why should anyone care about servers?

Feature development is far more valuable than solving  
server-centric stack challenges.

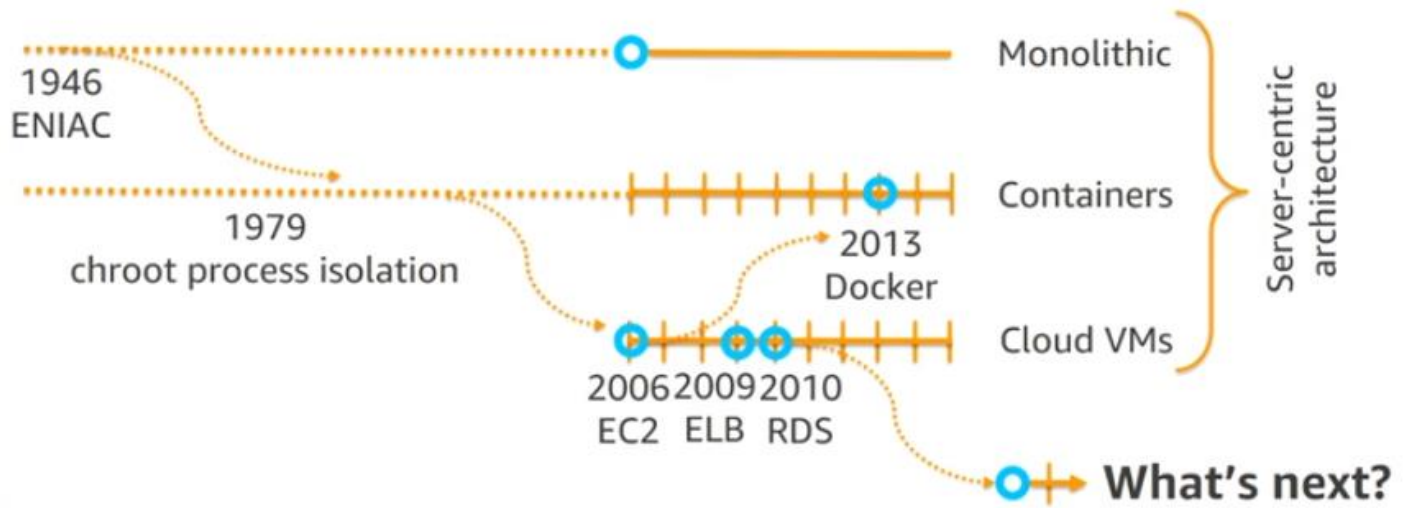
Why can't things just scale automatically?

**AWS**  
**re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



# Historical perspective



● Tipping point for "modern" startup adoption

AWS  
re:Invent

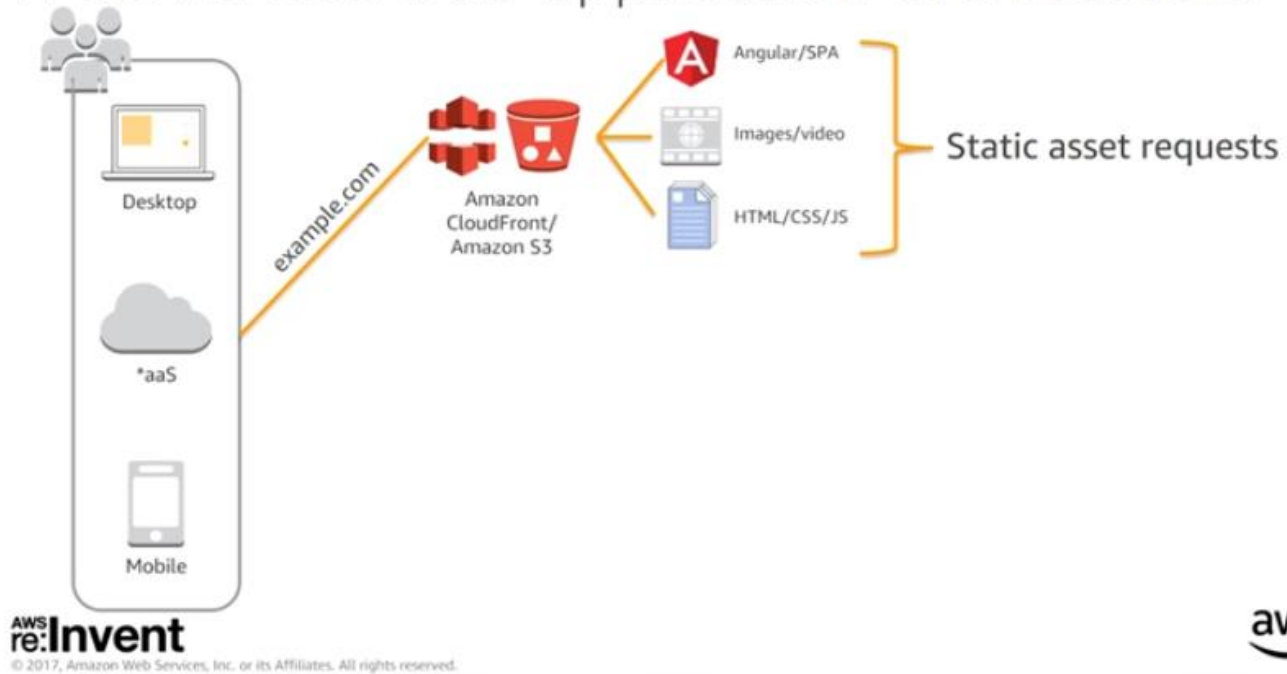
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



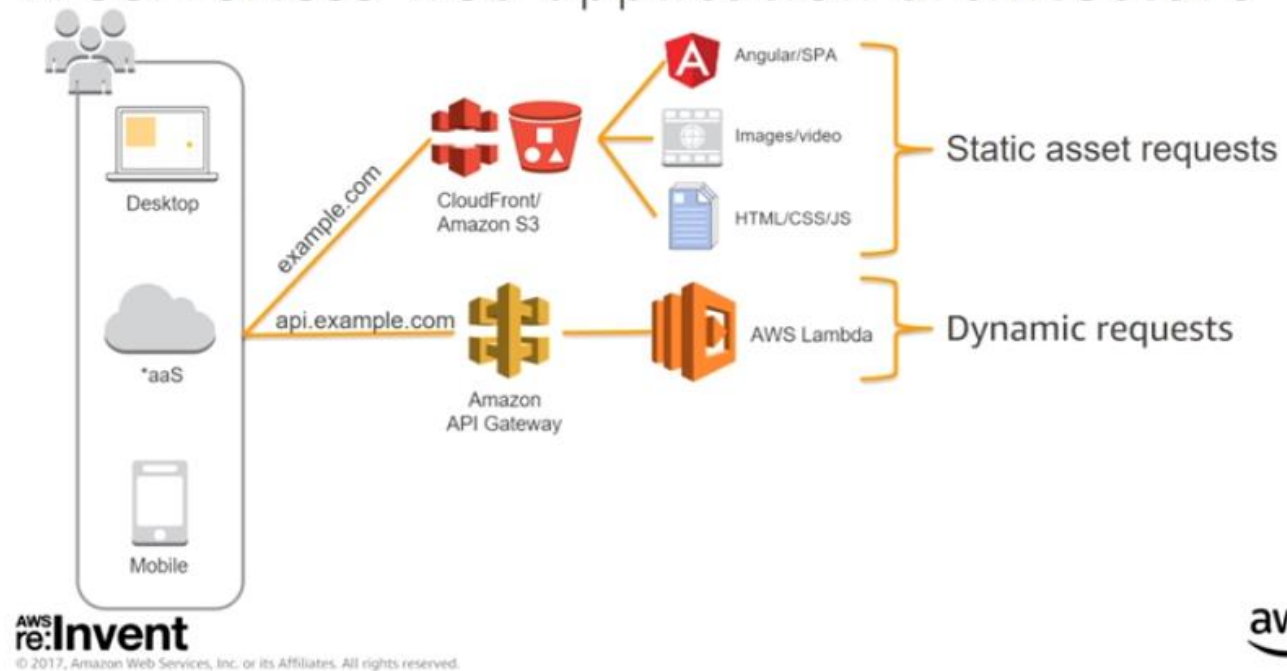
## Serverless



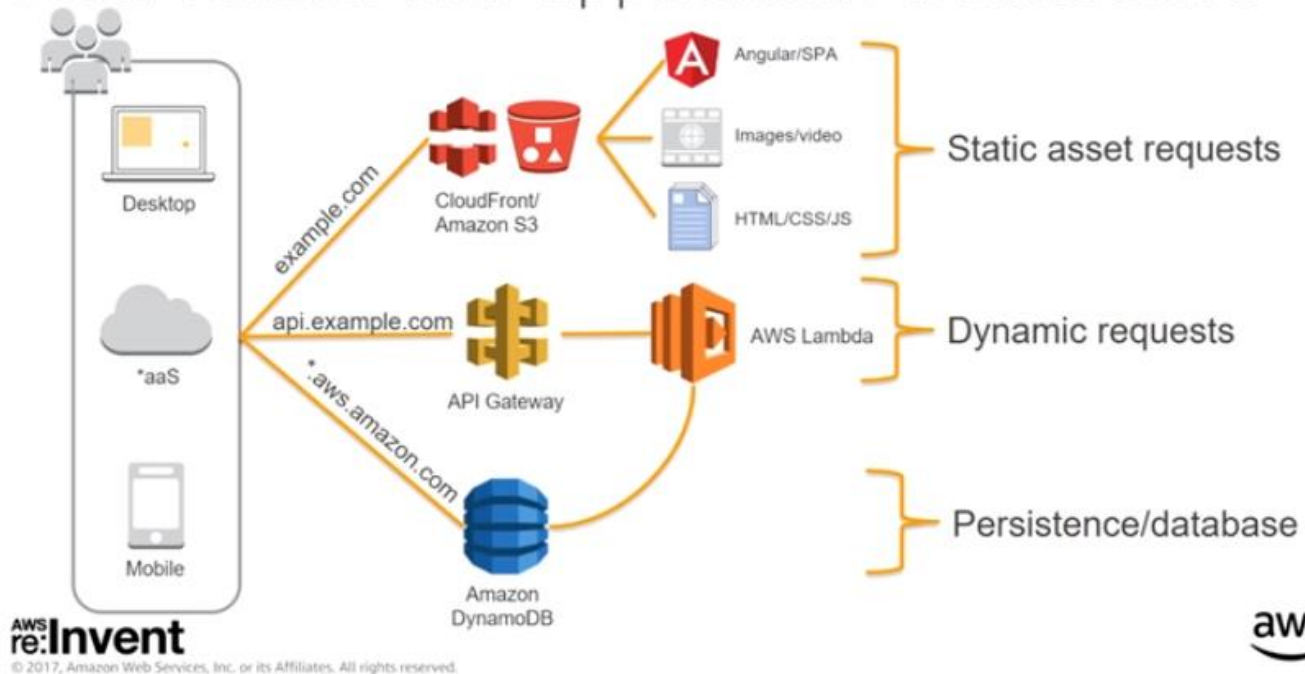
# A serverless web application architecture



# A serverless web application architecture

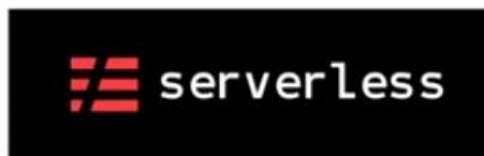


# A serverless web application architecture



## How are startups building serverless architecture?

### Native AWS: Serverless Application Modules



AWS re:Invent  
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



**AWS's SAM** allows you to define everything in your stack within a single cloudFormation template, it allows you to define API Gateways, Lambda functions, DynamoDB tables, etc. another popular approach is to use tools like **serverless** or **stdlib**.



# What does this cost?

## Cost per user/month:

Imagine the following daily customer usage pattern:

Assumption	Unit
Total pages/day	10
Avg. size of page	200 kb
API requests/page	5
Avg. size of API req	4 kb
DB Ops per API req	2 1r/1w
Storage (per month)	500 kb

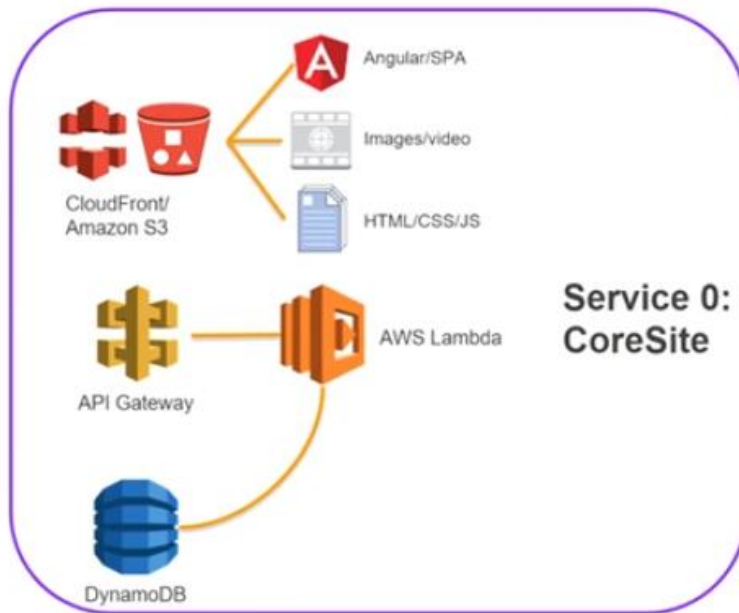
Charge	Monthly cost
CloudFront Data transfer	0.0051
CloudFront request pricing	0.0003
S3 request pricing (15% cache-hit)	0.00102
S3 data transfer	0.004335
API Gateway data transfer	0.00054
API Gateway request pricing	0.00525
Lambda request pricing	0.0003
Lambda duration cost	0.000312
DynamoDB IO pricing	0
DynamoDB storage	0.000125
<b>Total monthly cost/user</b>	<b>\$0.017282</b>

**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



# Through a microservice lens on day 0



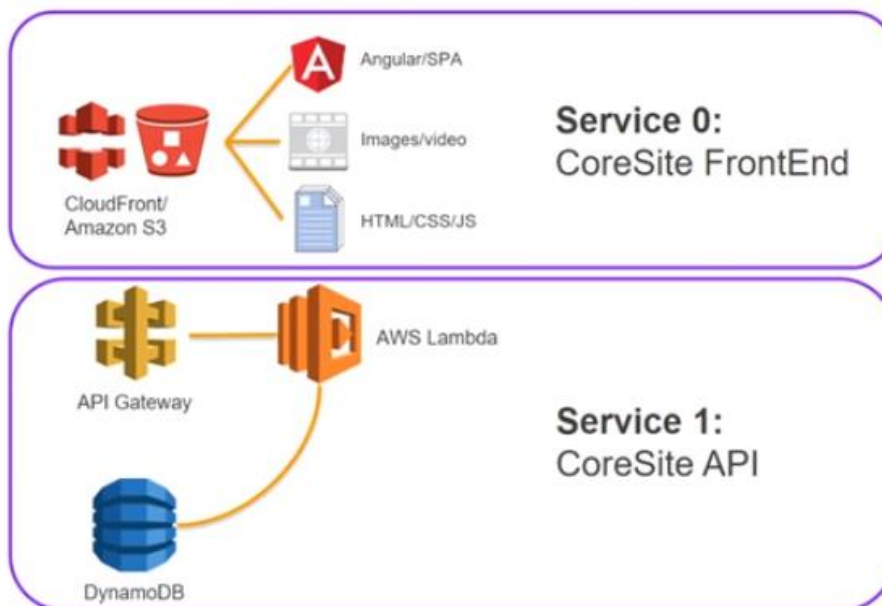
**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



This is what a microservices architecture might look like for a startup with one developer on day 1 building everything.

## As microservice complexity scales...



**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



You might have another microservice for the API separately as above

...and so on...

Let your teams pick the right tools for the job



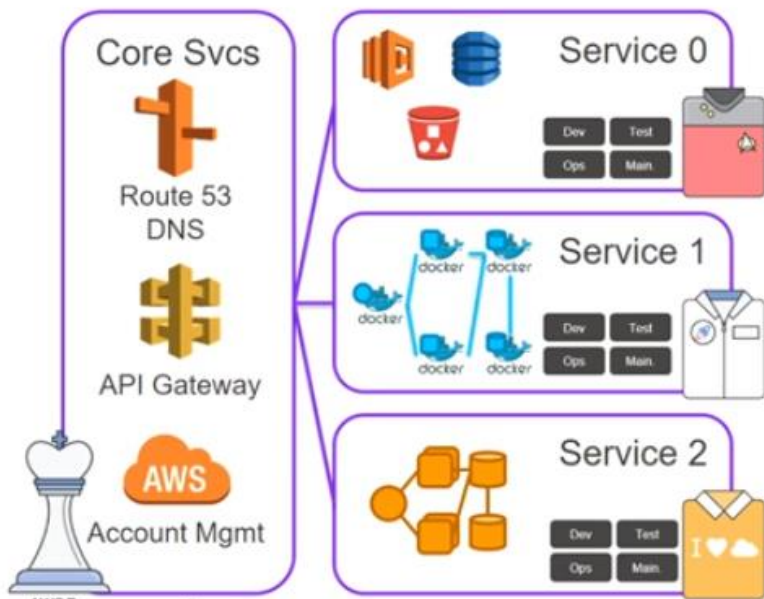
**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Now we have teams aligned around their separate services using the tools they want

Let your teams pick the right tools for the job



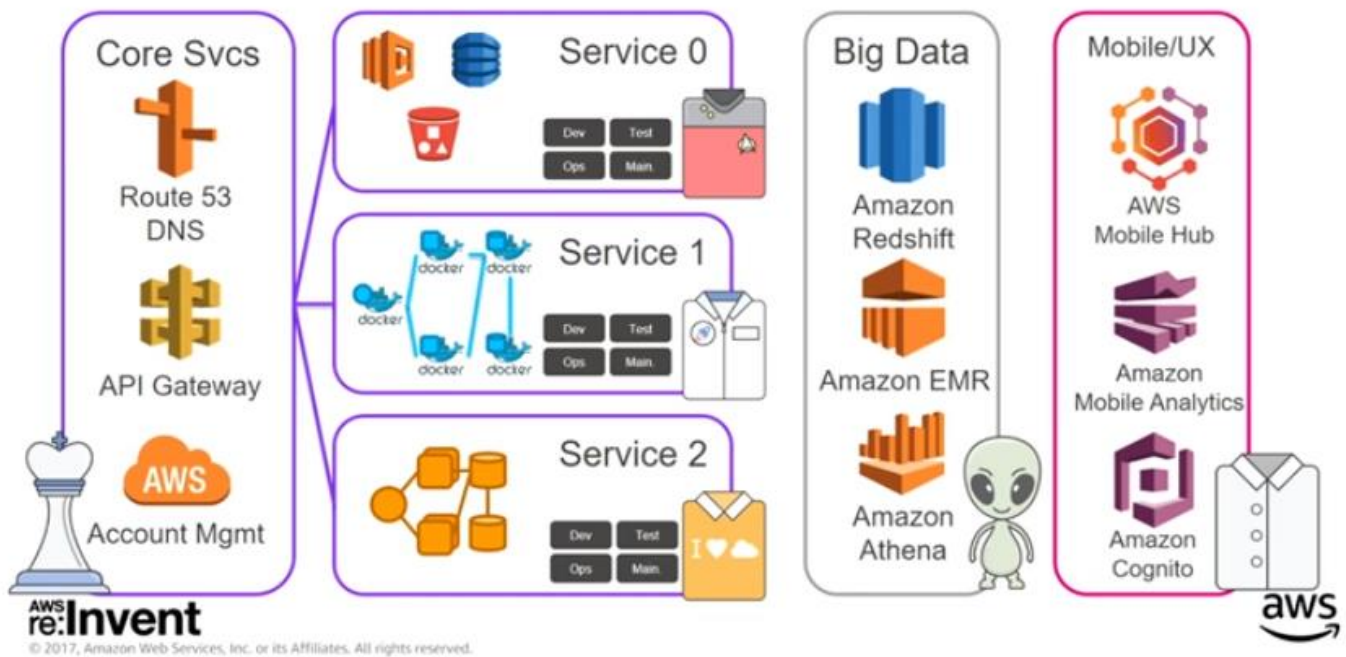
**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Now you can let your enterprise architecture take care of common things on the left for the entire teams you have like defining the interfaces while letting the teams implement those interfaces

Let your teams pick the right tools for the job



You can then let your new big data guy and the mobile analytics guy choose whatever services they want





## THIRD ROCK VENTURES INTRODUCTION

- Founded in 2007 with the goal of building a premier life sciences firm
  - *Innovation in academia, science/medicine exploding*
  - *Venture not interested in building early-stage companies*
  - *Large life science companies strategically looking externally to augment pipelines*
- Today, we are executing on this goal
  - *Created a firm based on a model of discovering and building innovative new companies that make a difference for patients*
- Differentiated Discover, Launch, Build strategy is delivering value to our key stakeholders



© 2017 Third Rock Ventures

## THIRD ROCK VENTURES—WHO WE ARE/WHAT WE DO

### DISRUPTIVE SCIENCE



We seek bold ideas and new frontiers to make a dramatic difference for patients

### BEST PEOPLE



We work with the best at TRV and through a highly integrated and leveraged network

### DISCOVER LAUNCH BUILD



We have a unique process of ideating, incubating, and growing companies

### COLLABORATIVE COMPANY BUILDING

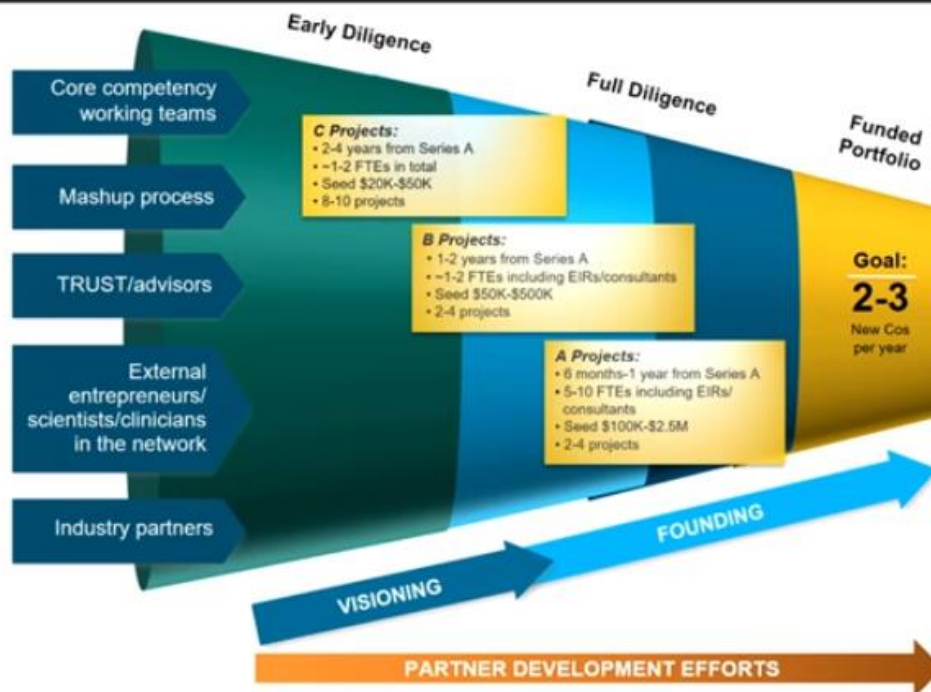


Be the preferred partner in pursuing innovative deal making

© 2017 Third Rock Ventures



## DISCOVERY PROCESS: C → B → A



## TRV PORTFOLIO COMPANIES HAVE PROVEN TO BE VALUABLE LONG-TERM INVESTMENTS

<b>\$616M</b> (2016)	<b>Fund IV</b>				
<b>\$516M</b> (2013)	<b>Fund III</b>				
<b>\$426M</b> (2010)	<b>Fund II</b>				
<b>\$378 M</b> (2007)	<b>Fund I</b>				

● Public/Exited ● Acquired ● Alliances ● Shut Down

© 2017 Third Rock Ventures

## TRV'S CONTINUOUS FLOW OF NEW COMPANIES ENABLES STANDARDIZATION OF AN ADVANCED INFRASTRUCTURE BUILD

---

- Heavy reliance on cloud infrastructure
  - *Common processes ensure efficiency, quality, and maintainability*
- COTS solutions over custom solutions
  - *Limited software engineering resources exist in early biotech companies*
  - *Maintain very high bar for vendors*
  - *Strong preference for companies with strong support of IaaS/SaaS options*
  - *Data standards are critical*
- Early software development focused on data integration efforts
  - *System integration, warehousing, services*
- Strong, cross-functional TRV platform team provides early companies with critical expertise and experience
  - *IT, software engineering, computational sciences, project management, automation*
  - *Common "playbook" utilized (and continuously improved) by the team*
- Common platform enables internal data needs and external data sharing
  - *Collaborators and corporate partners are integral to company success*

© 2017 Third Rock Ventures

## UNIQUE APPROACH TO COMPANY CREATION ENABLES A CORE PLATFORM APPROACH

---

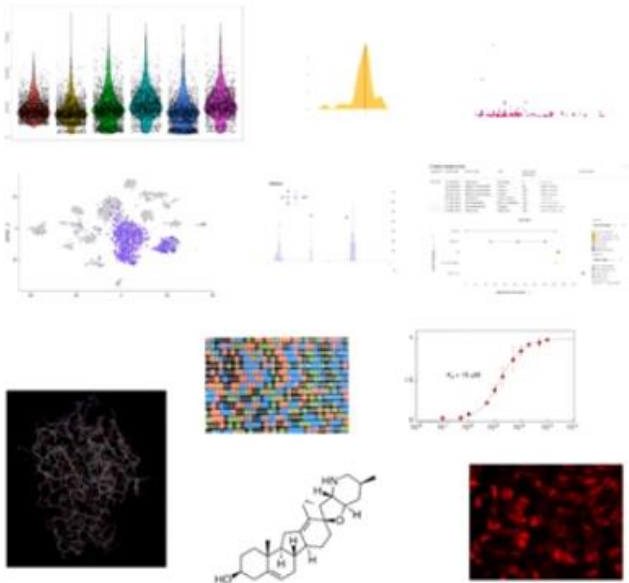
- VISION: Platform team increases the efficiency of research and development and decreases startup time by *helping* existing and emerging TRV companies share expertise, access best-in-class technologies, and encourage focused, practical leadership in the portfolio companies.
- We are a science, technology, information science, and systems resource.
- The Platform team can be leveraged to initiate or augment your research and development effort.

- Enable non-proprietary, cross company discussions
  - Make Contact Forums on topics of portfolio-wide interest
  - Make Contact Website supports ongoing conversations and provides access to forum content and other resources
- Drive CRO and tech resource sharing
- Validate breakthrough tech solutions
- Help identify key hires or contractors
- Strategic input: computational methods, infrastructure, and bench science
- Access CIO and CTO level insight

© 2017 Third Rock Ventures

## SMALL BIOTECH'S STRUGGLE TO ADDRESS COMMON DATA CHALLENGES

## Data Sources and Types Evolve Rapidly

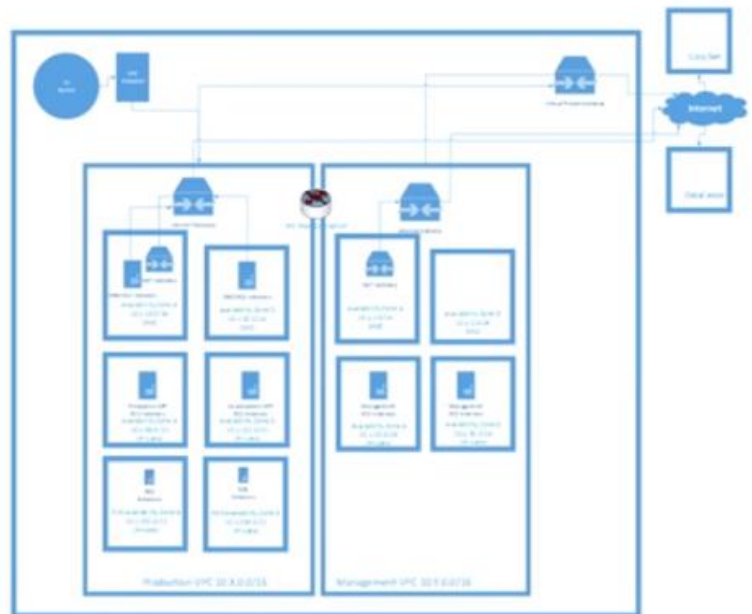


© 2017 Third Rock Ventures

- Solutions are generally tactical
  - *Appropriate in-house technical expertise does not generally exist*
  - *Lack of extensibility in infrastructure*
- Time to generate, analyze, and interpret data is the priority
  - *Reductionist approaches are common*
  - *Lack of focus on long-term value of data integration*

## 'BIOTECH BLUEPRINT' ENABLES RAPID DEPLOYMENT OF AN EXTENSIBLE INFRASTRUCTURE

- Initial focus on VPC and identity management
- Creation and expansion of TRV catalog
  - Contribution from TRV, vendors, and TRV PortCos*
  - Strong appetite from (some) vendors to enable native cloud deployment*
- Individual companies extend the framework to suit their specific needs
  - COTS deployment*
  - Custom solutions*



© 2017 Third Rock Ventures

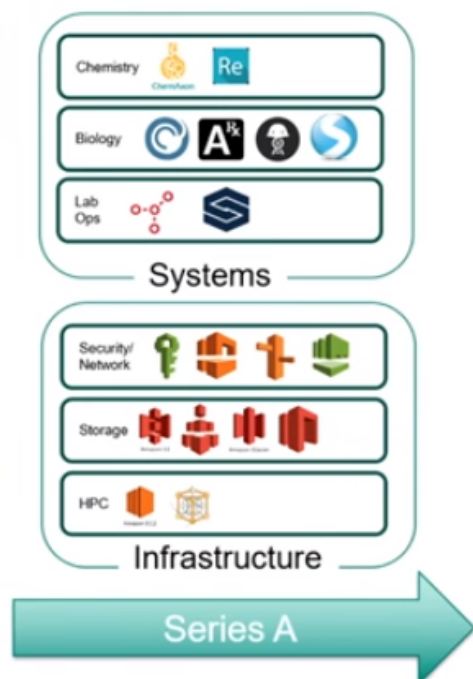
For a biotech startup, we first have to create the underlying technological infrastructure which is usually a typical 2 VPC setup with a development VPC and a production VPC setup with their IAM setups. This is now done in minutes and we can start up companies even without a space yet. We stand up the infra in the cloud and start pulling in data from the public APIs and getting access to 3<sup>rd</sup> party tools needed.

We have also started working on building out our TRV catalog which includes a mix of templates for things like the VPC, lambda scripts, and several other Amazon scripts from several vendors too.



After initial infra setup, each company can then go on and do their unique things. A computational chemistry company can create its stack different from a hard-core genomics company, or a traditional screening company.

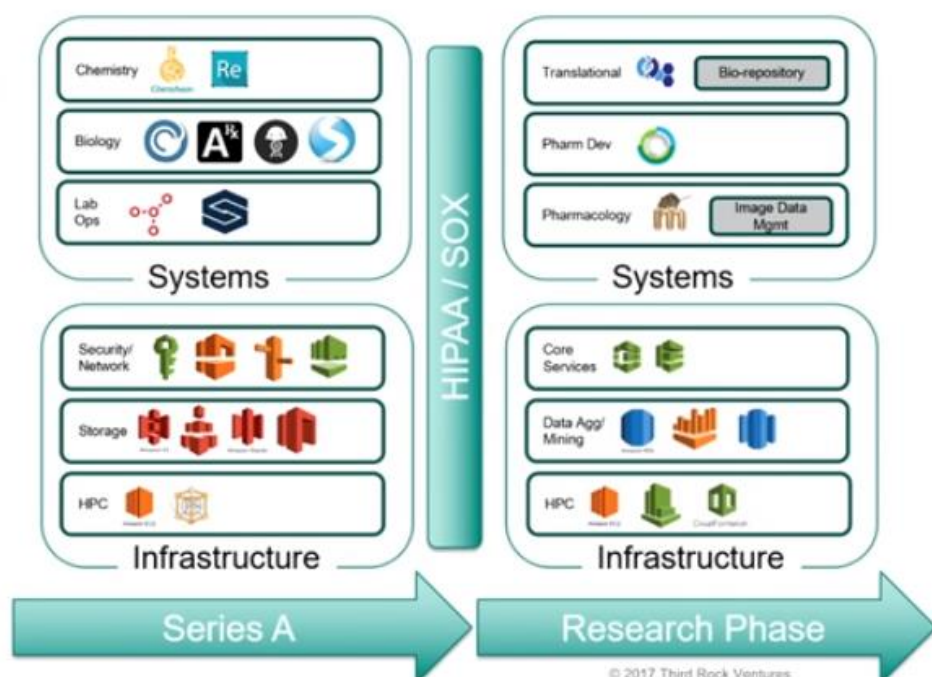
#### THERE IS A COMMON PATTERN TO COMPANY GROWTH AND INFRASTRUCTURE NEEDS



© 2017 Third Rock Ventures

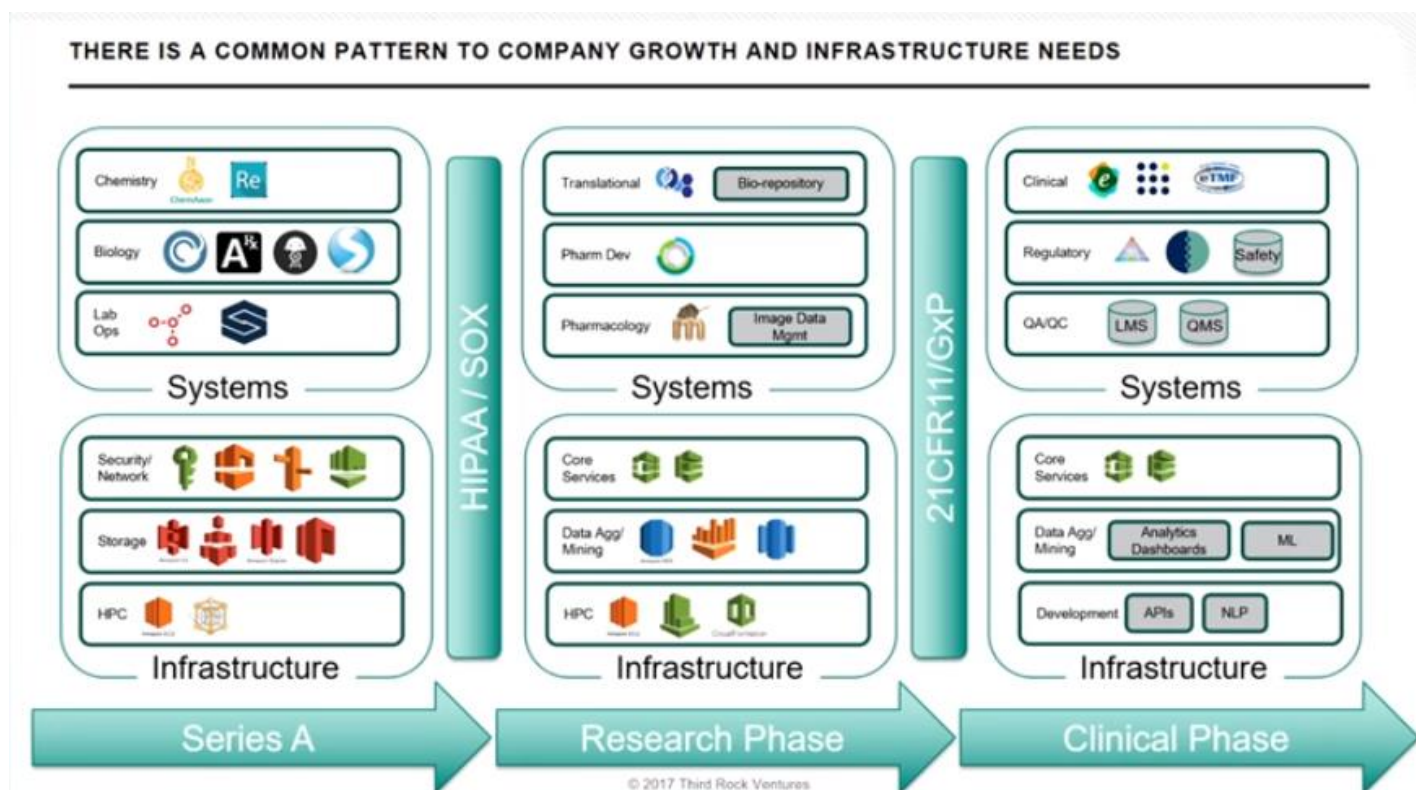
What is interesting is that we can sort of predict what is going to come next, there is a common pattern that emerges when we look at how the companies spend their different stages money. HPC needs becomes a major need for most of our early stage companies during initial startup, then we layer on top of these commercial off the shelf technologies that should be plug and play for things like chemical registration for chemistry related technology companies.

#### THERE IS A COMMON PATTERN TO COMPANY GROWTH AND INFRASTRUCTURE NEEDS

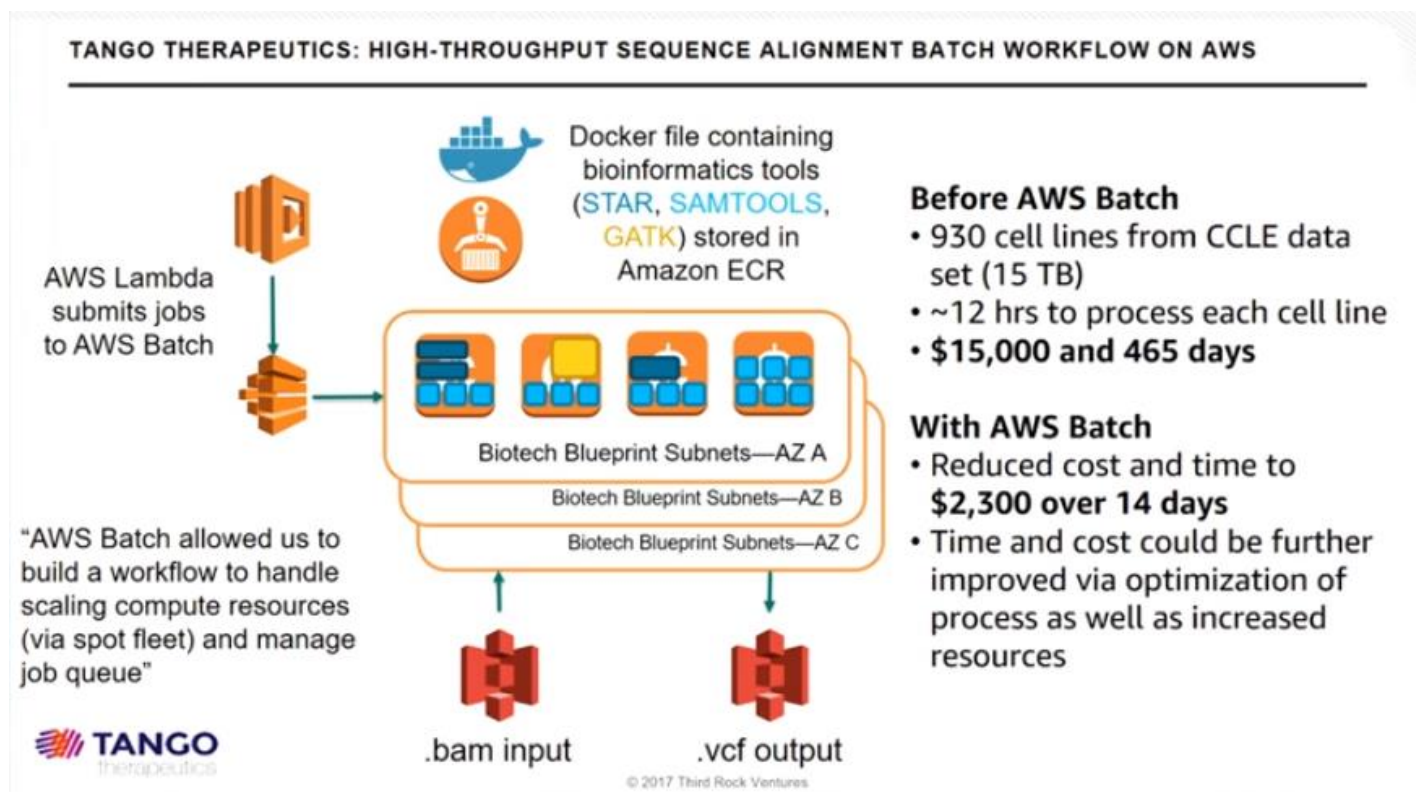


© 2017 Third Rock Ventures

During the hardcore research phase, we are typically adding more functionalities to the company like Pharmacology to the already setup things like LabOps to generate more data. This is where we build out the infra to support these new functionalities using more HPC for things like HIPAA/SOX compliance, data aggregation and mining.

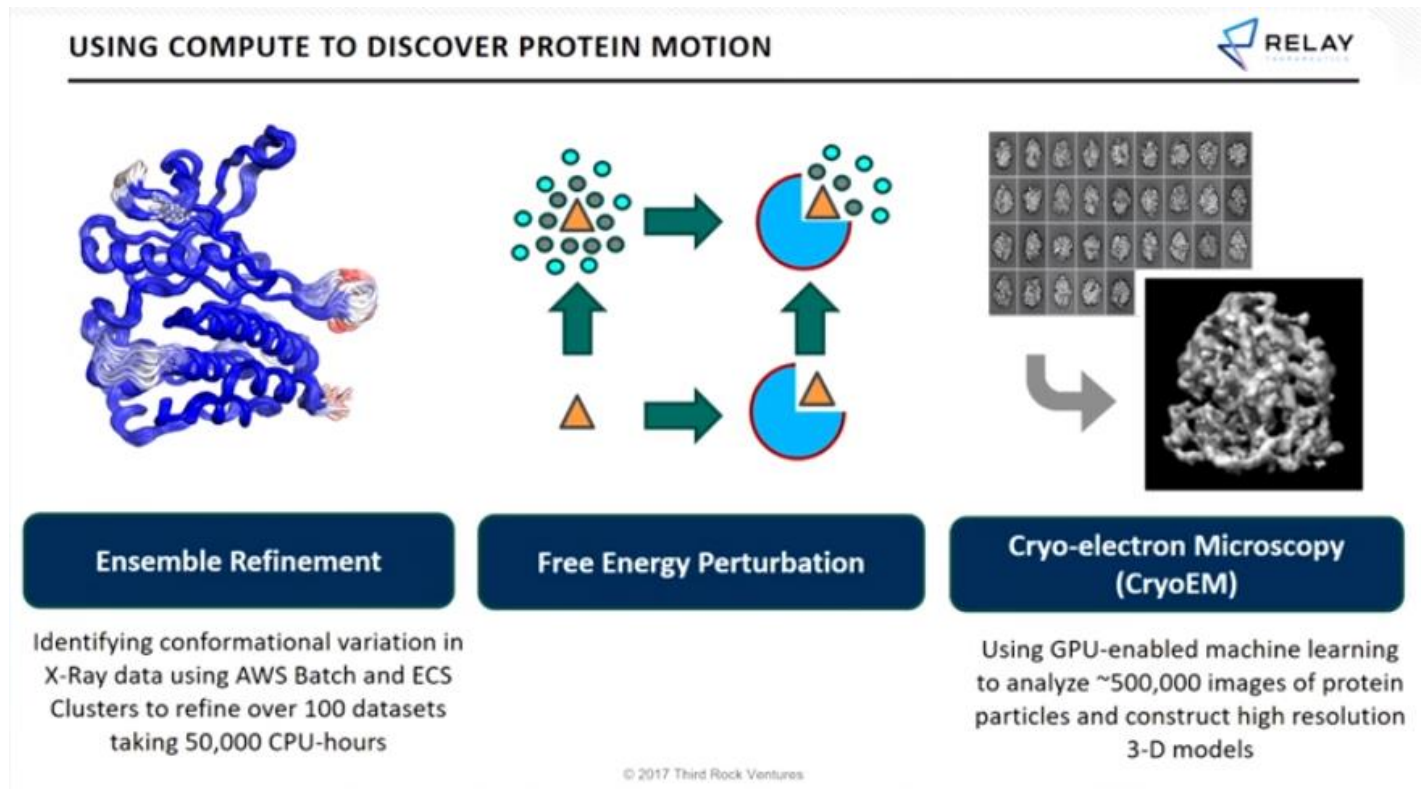


Eventually we get to the clinical phase where we start dealing with a lot of compliance issues like 21CFR11/GxP regulations in our systems. You also start building out new functionalities for clinical, regulatory issues like how to submit things to the FDA, how do we take generated data to back up our FDA approval processes.



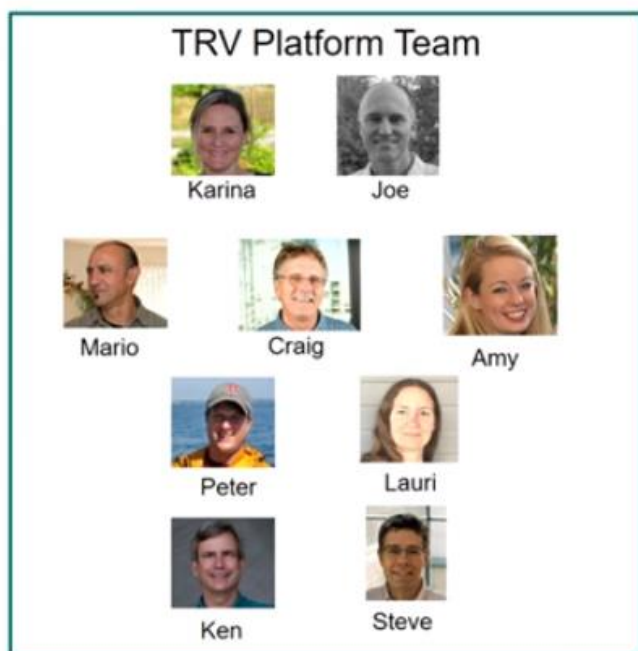


This is one of our companies for developing cancer therapies, they are creating cutting edge genomics to identify novel targets and also be able to pick patient population via targeting. This allows them to develop technologies for matching the right patient to the right cancer therapy. The data being generated by this company comes from both internal efforts via screening, and also from external datasets. Now we have a system that leverages AWS Lambda and AWS Batch to reduce a process that takes over a year (465 days) to process the available cell lines to now take just 14 days.



Another of our companies called Relay Therapeutics is developing small molecules therapy based on understanding the **protein structures** such that we can find unique binding spot targets on these proteins. This is important for new types of drug therapies. Above are 3 ways that the computational team does this. There are huge computational needs for doing this using the right platform and framework for analyzing these datasets using P3 instances on AWS and the available ML services to build 3D images of the new structures.

## ACKNOWLEDGEMENTS



© 2017 Third Rock Ventures

69

So many of the best ideas actually come from our 44 portfolio companies that contribute back to the mothership

## Final thought

### Think like a startup



**AWS re:Invent**

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

