



AWS
re:Invent

ARC310

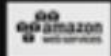
Cost Optimizing Your Architecture

Practical Design Steps To Get Big Savings

Simon Elisha, Head of Solution Architecture, ANZ Public Sector, AWS.

December 2016

© 2016, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



What to Expect from the Session

We are going to dive **RIGHT IN!**

Assumed knowledge... You know a fair bit about AWS

Cover a LOT in a **hurry!**

Operational optimization

Infrastructure optimization

Architectural optimization

Plenty of **FOOD FOR THOUGHT**

All pricing shown will be for Asia Pacific (Sydney) Region

SHOW ME THE MONEY!!!

32%

53%

87%

42%

72%

Quick Wins (The Obvious Stuff)

Use Reserved Instances For Stable Workloads

Standard Reserved Instances

- ✓ 1 or 3 year term
- ✓ Up to 75% cheaper than On-Demand price

Convertible Reserved Instances

- ✓ 3-year term
- ✓ Up to 45% cheaper than On-Demand price
- ✓ Change instance families, OS, or tenancy

Run your system for at least 3 months, get an understanding of what your workloads actually look like, spend some time tuning and optimizing your systems, then make your ROI commitment using reserved instances for your stable workloads.

Use Consolidated Billing For Volume Discounts

Payer account combines usage from all sub-accounts

Many services offer volume pricing tiers across certain usage dimensions

E.g., EC2, S3, & CloudFront

Gives you a lower price overall whenever possible

A New Domain for Architecture



Availability



Performance



Security



Function



Economy

You also need to look at the economy of your architecture because you have a lot of control over it

Economy of Architecture



Economy

- ✓ Same or better outcome for the lowest cost
- ✓ Trial and change **during** the lifetime of the system
- ✓ Radical changes are possible – driven by economics
- ✓ Changes in services that can be consumed
- ✓ Transaction cost + operational cost

What is Your Per-Transaction cost?



Cost per query?

Cost per user?

Cost per processing unit?

Do you know it real time?

Do you track it?

Do you do this things for your favorite application or systems?

Actionable Steps



Beginner

Calculate by hand (Cost Explorer with tags)

Intermediate

Instrument your system to gather transaction volume

Calculate based on tags & billing data periodically

Advanced

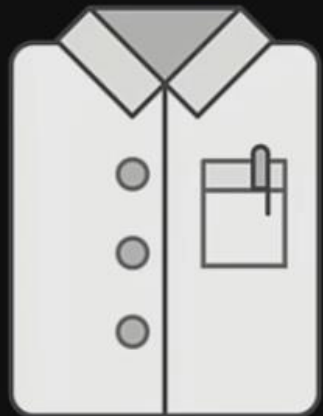
Monitor in real time

Display somewhere **public**

Drive the development team to optimize

You can automate the data using a service like Kinesis firehose, put the data on a display somewhere with updates every hour, then focus on how you can optimize and reduce the cost prices.

Operational Optimization



Operational Cost – By Country (median)

Skill	Country	Cost per Hour (USD)
Systems Administrator	USA	\$30
	Australia	\$26
	UK	\$18
	Japan	\$47
	Germany	\$24
	Brazil	\$32
	Singapore	\$17

*Source: [payscale.com](https://www.payscale.com) & [indeed.com](https://www.indeed.com)

Operational Cost – By Country (median)

Skill	Country	Cost per Hour (USD)
Database Administrator	USA	\$35
	Australia	\$25
	UK	\$19
	Japan	\$49
	Germany	\$31
	Brazil	\$45
	Singapore	\$19

*Source: [payscale.com](https://www.payscale.com) & [indeed.com](https://www.indeed.com)

What's the Admin Effort for a Minor DB Version Update?

Self-managed

- Back up primary
- Back up secondary
- Back up server OS
- Assemble upgrade binaries
- Create change record
- Create rollback plan
- Rehearse in development
- Run against staging
- Run against production standby
- Verify
- Failover
- Run in production
- Verify

8 Hours
(minimum)

RDS – admin effort

- Verify Update Windows
- Create “notification” change record
- Verify success in staging
- Verify success in production

1 Hour

Granularity 60 second(s)

☒ I authorize RDS to create the IAM role rds-monitoring-role.

Maintenance

Auto Minor Version Upgrade Yes

Maintenance Window Select Window

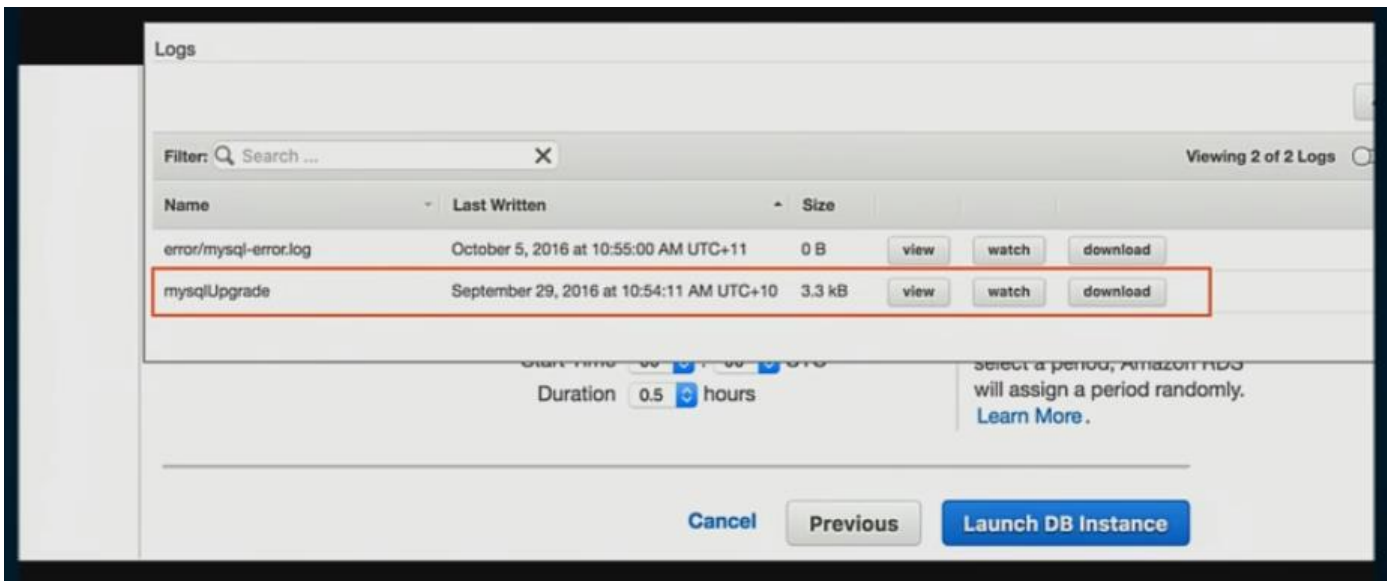
Start Day Monday

Start Time 00 : 00 UTC

Duration 0.5 hours

Select the period in which you want pending modifications (such as changing the DB instance class) or patches applied to the DB instance by Amazon RDS. Any such maintenance should be started and completed within the selected period. If you do not select a period, Amazon RDS will assign a period randomly. [Learn More.](#)

Cancel Previous **Launch DB Instance**



Control – Without Operational Cost

Country	Self Managed @ 8 Hours	RDS @ 1 Hour	Saving
USA	\$280	\$35	\$245
Australia	\$200	\$25	\$175
UK	\$152	\$19	\$133
Japan	\$392	\$49	\$343
Germany	\$248	\$31	\$217
Brazil	\$360	\$45	\$315
Singapore	\$152	\$19	\$133

*Median Salary - USD

Infrastructure Optimization



Use Amazon S3 Storage Tiers



Take advantage of Infrequent Access (IA)
No code changes – immediate savings

Example

- ✓ Our app stores 100 TB of photos
- ✓ Each photo is about 10 MB
- ✓ Using Standard S3, we pay \$3,292.78 per month (SYD)

Let us see some ways to save money on your storage without any code change at all once you understand your usage patterns

Activate a Policy

Move data after 30 days

Assume we can move 80% of the data

Of which 20% is still accessed regularly < twice per month

The screenshot shows the 'Lifecycle Rules' configuration page in the AWS console. It is divided into three steps: Step 1: Choose Rule Target, Step 2: Configure Rule, and Step 3: Review and Name. The 'Action on Objects' section is active, showing three options: 'Transition to the Standard - Infrequent Access Storage Class' (selected), 'Archive to the Glacier Storage Class', and 'Permanently Delete'. The 'Transition' option is configured to occur 30 days after the object's creation date. Below this, an 'EXAMPLE' diagram illustrates the process: an object is uploaded on September 27, 2016 (Day 0), and after 30 days (October 27, 2016), the rule transitions it to the Standard - Infrequent Access storage class.

You can create a policy that applies to your S3 bucket and moves stuff into more longer term, cheaper storage option like Glacier.

S3 – IA – Result!



Same 100 TB (20 TB Standard/80 TB IA)
costs only \$2,220.65 per month (SYD)

Revisit & Right-Size EC2



Revisit your EC2 instance choice **at least every 6 months**

- ✓ Families change
- ✓ Workloads change

Cost savings available between EC2 families depending on your workload

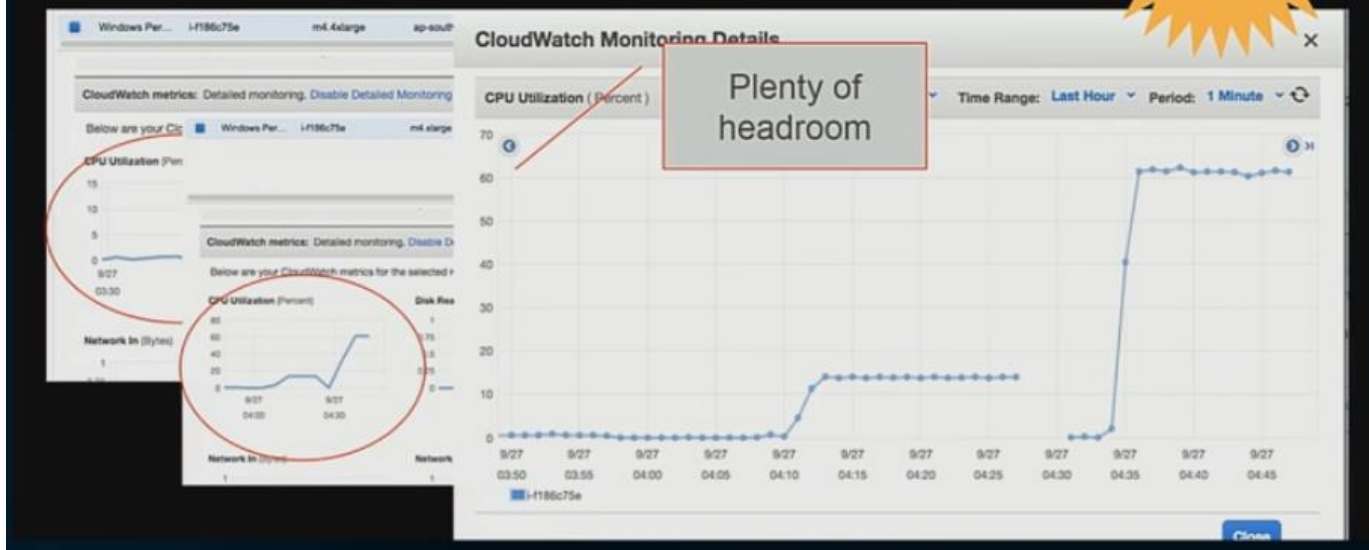
Use Trusted Advisor for hints

Windows Instance Example

m4.4xlarge -> \$2.353/hr (SYD)

m4.xlarge -> \$0.294/hr (SYD)

87%
Saving



Trusted Advisor - Hints

Cost Optimization



7 2 0

\$3.04

Potential monthly savings

Filter by tag

Tag Key

Tag Value

Apply filter

Reset

View

All checks

Cost Optimization Checks



Unassociated Elastic IP Addresses

Refreshed: a minute ago



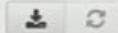
Checks for Elastic IP addresses (EIPs) that are not associated with a running Amazon Elastic Compute Cloud (Amazon EC2) instance.

1 of 1 Elastic IP addresses are not associated with a running instance.



Underutilized Amazon EBS Volumes

Refreshed: a few seconds ago



Checks Amazon Elastic Block Store (Amazon EBS) volume configurations and warns when volumes appear to be underused.

4 of 4 EBS volumes appear to be underutilized. Monthly savings of up to \$3.04 are available by minimizing underused EBS volumes.

Trusted Advisor - Hints

Cost Optimization



7 2 0



Low Utilization Amazon EC2 Instances

Refreshed: a minute ago



Checks the Amazon Elastic Compute Cloud (Amazon EC2) instances that were running at any time during the last 14 days and alerts you if the daily CPU utilization was 10% or less and network I/O was 5 MB or less on 4 or more days. Running instances generate hourly usage charges. Although some scenarios can result in low utilization by design, you can often lower your costs by managing the number and size of your instances.

Estimated monthly savings are calculated by using the current usage rate for On-Demand Instances and the estimated number of days the instance might be underutilized. Actual savings will vary if you are using Reserved Instances or Spot Instances, or if the instance is not running for a full day. To get daily utilization data, download the report for this check.

Alert Criteria

Yellow: An instance had 10% or less daily average CPU utilization and 5 MB or less network I/O on at least 4 of the previous 14 days.

Recommended Action

Consider stopping or terminating instances that have low utilization, or scale the number of instances by using Auto Scaling. For more information, see [Stop and Start Your Instance](#), [Terminate Your Instance](#), and [What is Auto Scaling?](#)

Additional Resources

[Monitoring Amazon EC2](#)

[Instance Metadata and User Data](#)

[Amazon CloudWatch Developer Guide](#)

[Auto Scaling Developer Guide](#)

0 of 0 Amazon EC2 instances have low average daily utilization. Monthly savings of up to \$0 might be available by minimizing underutilized instances.

Filter by tag

Tag Key

Cost Optimiz



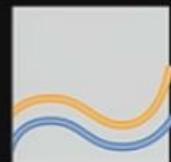
Use Spot Instances

Incredible value for the right workloads

Build it into your Auto Scaling groups

Can balance price & availability

Reserved + On-Demand + Spot



Auto Scaling Group Launch Configuration

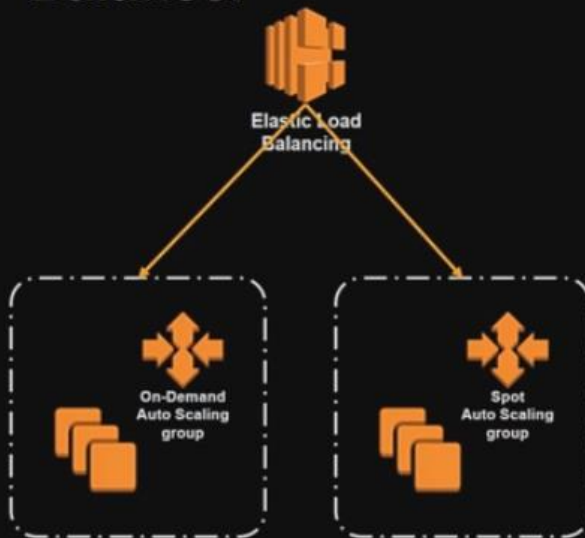
```
},
"Properties": {
  "ImageId": { "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" },
    { "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" }, "Arch" ] } ] },
  "InstanceType": { "Ref": "InstanceType" },
  "SecurityGroups": [ { "Ref": "WebServerSecurityGroup" } ],
  "KeyName": { "Ref": "KeyName" },
  "UserData": { "Fn::Base64": { "Fn::Join": [ "", [
    "#!/bin/bash -xe\n",
    "yum update -y aws-cfn-bootstrap\n",
    "\n",
    "/opt/aws/bin/cfn-init -v ",
    "  --stack ", { "Ref": "AWS::StackId" },
    "  --resource LaunchConfig ",
    "  --configsets full_install ",
    "  --region ", { "Ref": "AWS::Region" }, "\n",
    "\n",
    "/opt/aws/bin/cfn-signal -e $? ",
    "  --stack ", { "Ref": "AWS::StackId" },
    "  --resource WebServerGroup ",
    "  --region ", { "Ref": "AWS::Region" }, "\n"
  ] ] } }
  ] ] }
}
```

Duplicate Auto Scaling Group - Add One Extra Line

```
},
"Properties": {
  "ImageId": { "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" },
    { "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" }, "Arch" ] } ] },
  "InstanceType": { "Ref": "InstanceType" },
  "SecurityGroups": [ { "Ref": "WebServerSecurityGroup" } ],
  "KeyName": { "Ref": "KeyName" },
  "SpotPrice": { "Ref": "SpotPrice" },
  "UserData": { "Fn::Base64": { "Fn::Join": [ "", [
    "#!/bin/bash -xe\n",
    "yum update -y aws-cfn-bootstrap\n",
    "\n",
    "/opt/aws/bin/cfn-init -v ",
    "  --stack ", { "Ref": "AWS::StackId" },
    "  --resource LaunchConfig ",
    "  --configsets full_install ",
    "  --region ", { "Ref": "AWS::Region" }, "\n",
    "\n",
    "/opt/aws/bin/cfn-signal -e $? ",
    "  --stack ", { "Ref": "AWS::StackId" },
    "  --resource WebServerGroup ",
    "  --region ", { "Ref": "AWS::Region" }, "\n"
  ] ] } }
  ] ] }
}
```

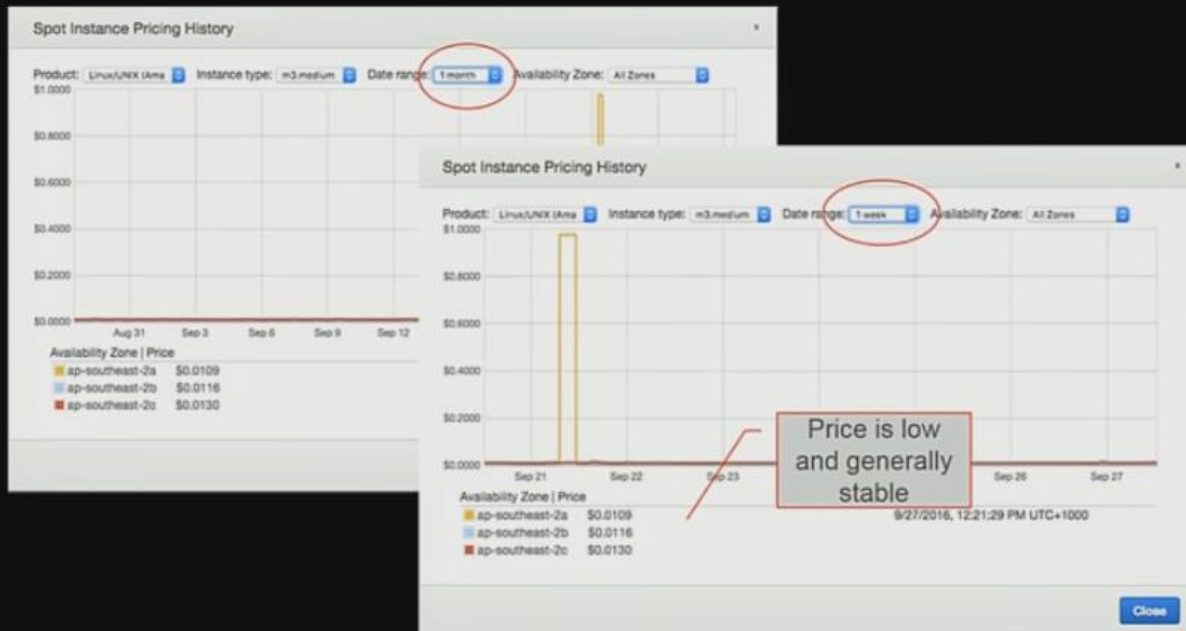
Magic happens here...

Both Groups Registered to the Same Load Balancer



- ✓ Set your Spot price > Market < On-Demand
 - ✓ Will always give you the market rate at or under your bid
- ✓ CloudWatch alarm on number of Spot Instances in group
 - ✓ `GroupInServiceInstances`
- ✓ Scale On-Demand group if #Instances in Spot group drops below threshold

Understand Price History



Spot Price vs On-Demand

Region: Asia Pacific (Sydney)

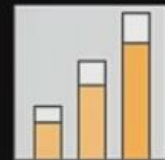
	vCPU	ECU	Memory (GiB)	Instance Storage (GiB)	Linux/UNIX Usage
General Purpose - Current Generation					
t2.nano	1	Variable	0.5	EBS Only	\$0.01 per Hour
t2.micro	1	Variable	1	EBS Only	\$0.02 per Hour
t2.small	1	Variable	2	EBS Only	\$0.04 per Hour
t2.medium	2	Variable	4	EBS Only	\$0.08 per Hour
t2.large	2	Variable	8	EBS Only	\$0.16 per Hour
m4.large	2	6.5	8	EBS Only	\$0.168 per Hour
m4.xlarge	4	13	16	EBS Only	\$0.336 per Hour
m4.2xlarge	8	26	32	EBS Only	\$0.673 per Hour
m4.4xlarge	16	53.5	64	EBS Only	\$1.345 per Hour
m4.10xlarge	40	124.5	160	EBS Only	\$3.363 per Hour
m3.medium	1	3	3.75	1 x 4 SSD	\$0.093 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	\$0.186 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	\$0.372 per Hour
m3.2xlarge	8	26	30	2 x 80 SSD	\$0.745 per Hour

Type	Servers	Cost per Hour
On-Demand	12	\$0.093
Spot	0	\$0.013
TOTAL	12	\$1.17

Type	Servers	Cost per Hour
On-Demand	2	\$0.093
Spot	10	\$0.013
TOTAL	12	\$0.32



Anatomy of the Developer Work Week



“Your mileage may vary!”

Monday->Friday 10 hours per day = 50 Hours

✓ “Our devs work hard!”

Total hours available in the week = 168 Hours

Turning things off = immediate **70% saving**

Automate Dev/Test Elasticity

A number of approaches (choose one that works for you):

- ✓ Tagging with script to shut down/start up
- ✓ Lambda to drive the shutdown/startup
- ✓ Start up based on time
- ✓ Start up based on request
- ✓ Scheduled Auto Scaling groups
- ✓ Tagging with different rules based on tags (e.g., run over weekends)

Be sure to use **something**...

An Incomplete List of Tools to Help You

Use AWS Lambda:

<https://aws.amazon.com/blogs/compute/creating-an-enterprise-scheduler-using-aws-lambda-and-tagging/>

Use AWS Data Pipeline:

<https://aws.amazon.com/premiumsupport/knowledge-center/stop-start-ec2-instances/>

Use Amazon CloudWatch:

<https://aws.amazon.com/about-aws/whats-new/2013/01/08/use-amazon-cloudwatch-to-detect-and-shut-down-unused-amazon-ec2-instances/>

Cache for Savings



Read-Intensive workloads in particular

Handy whitepaper:

<https://d0.awsstatic.com/whitepapers/performance-at-scale-with-amazon-elasticache.pdf>

Example 30,000 reads per second

db.m3.large – 30,000 PIOPS – Single-AZ: \$3,889.68

cache.m3.large – 30,000 GETS per second: \$355.02

90%
saving

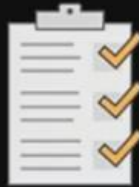
Architectural Optimization



New Levers to Pull



Code changes



Architectural
tradeoffs



Immediate
benefits

Eliminate Your Web Server Tier



How much work is your web server tier actually doing?

How much time do you spend on maintenance?

How secure is your web tier?

S3 Static Website Hosting

Not so "static"

JavaScript
(includes AWS SDK)
CORS

Plus

No servers
No patching
No scaling rules

The screenshot shows the AWS Management Console interface for an S3 bucket named 'sa-video-assignments'. On the left, a table lists the bucket's contents:

Name	Storage Class	Size	Last Modified
favicon.ico	Standard	161.2 KB	Sun Jun 05 18:37:39 GMT+1000 2016
index.html	Standard	2.7 KB	Sun Jun 05 18:37:41 GMT+1000 2016
logo.png	Standard	4.8 KB	Sun Jun 05 18:37:41 GMT+1000 2016

On the right, the 'Static Website Hosting' configuration page is shown. It includes a description of static website hosting, an endpoint URL, and options to enable or disable website hosting. The 'Enable website hosting' option is selected, and the 'Index Document' is set to 'index.html'. The 'Error Document' field is empty. There is also a section for 'Edit Redirection Rules' with a checkbox for 'Redirect all requests to another host name'.

S3 - Cost Optimized Web Serving - Example

Average objects per page: 150

Average size per page: 2 MB

100,000 page views per day =

450,000,000 GETs per month

5.859 TB data transfer out per month

Total cost **PER MONTH**: \$1,139.43 (SYD)

S3 - Cost Optimized Web Serving

Cost avoidance

- ✓ Web server patching
- ✓ Capacity planning
- ✓ Security scanning
- ✓ Content rollouts/updates

Hours per month...

Queues – Not Just for Decoupling

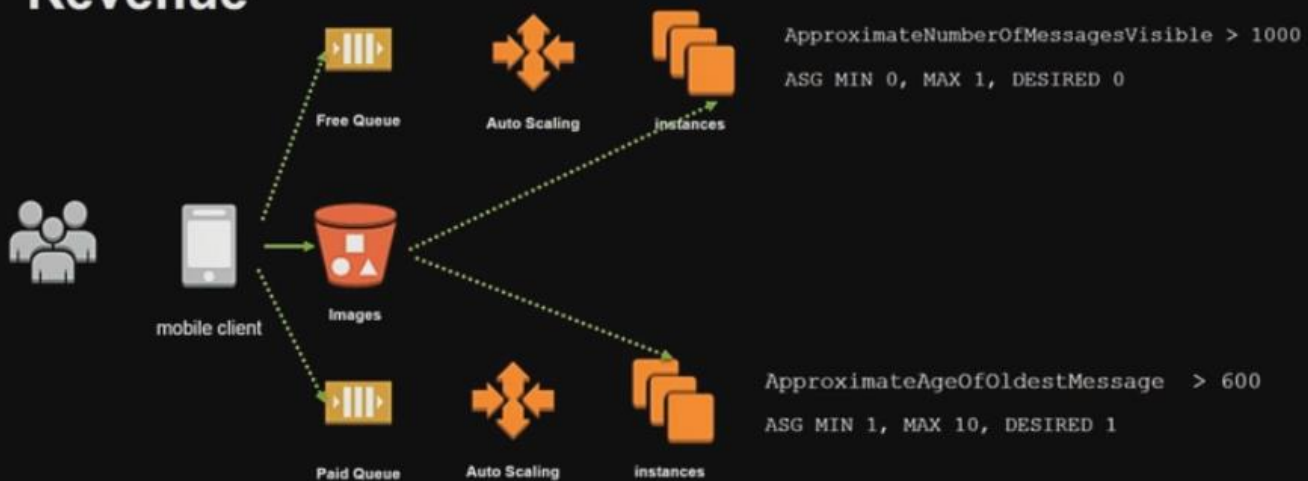


SQS gives you tremendous power to decouple your architecture

However, resilience is only one part of the story

Use queues to manage **cost**

SQS Trigger Auto Scaling Groups Based on Revenue



In our example, a single `m4.large` can process 1,000 images in an hour. We don't want paid customers to wait more than 10 minutes for their result.

The auto-scaling group for the free tier queue only spins up to process the images when we have at least 1000 images in that free queue. For paid, customers, the queue uses the age of the oldest object in the queue that should not be more than 10 minutes.

SQS Trigger Auto Scaling Groups Based on Revenue



For example
 2,000 TXNs
 50/50 free/premium

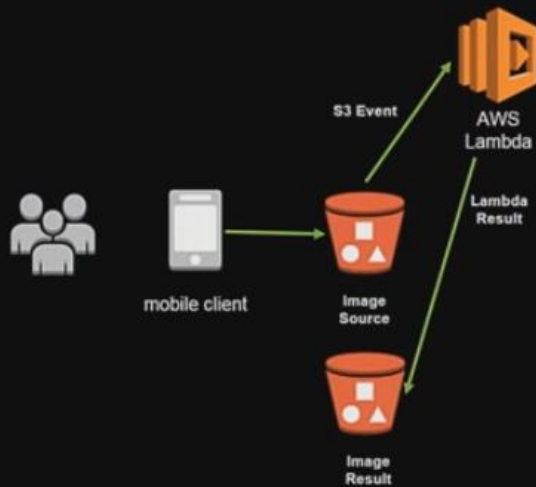
Free queue TXN cost:
 $\$0.336/1,000 = \0.000336
 *One instance

Paid queue TXN cost:
 $(\$0.336*6)/1,000 = \0.002016
 *Six instances to meet 10 minute SLA

In our example, a single `m4.large` can process 1,000 images in an hour. We don't want paid customers to wait more than 10 minutes for their result.

Can we make this even cheaper?

Simplify. Less Components. Less Cost.



For example

Each TXN takes 3750ms

Assuming largest Lambda memory size (1536 MB)

@ \$ 0.000002501 per 100ms

Each TXN now costs \$0.000937875

Premium queue TXN was \$0.002016

PLUS The ongoing free tier would give us over 71K free TXNS per month
We also eliminated SQS cost (negligible as that is)

53%
saving

We instead move to a lambda based model, where we have 2 separate buckets. We receive an image in the receive bucket and do an S3 event that triggers a lambda function that does the work of processing the image and stores the result into the 2nd result bucket. You also get a free tier with lambda

Simple to Configure

Event Notifications enable you to send alerts or trigger workflows. Notifications can be sent via [Amazon Simple Notification Service \(SNS\)](#) or [Amazon Simple Queue Service \(SQS\)](#) or to a [Lambda function](#) (depending on the bucket location).

Name	<input type="text" value="ImageProcessor"/>	i
Events	<input type="text" value="Put"/>	i
Prefix	<input type="text" value="e.g. images/"/>	i
Suffix	<input type="text" value=".jpg"/>	i
Send To	<input type="radio"/> SNS topic <input type="radio"/> SQS queue <input checked="" type="radio"/> Lambda function	i
Lambda function	<input type="text" value="CRWriteReport"/>	

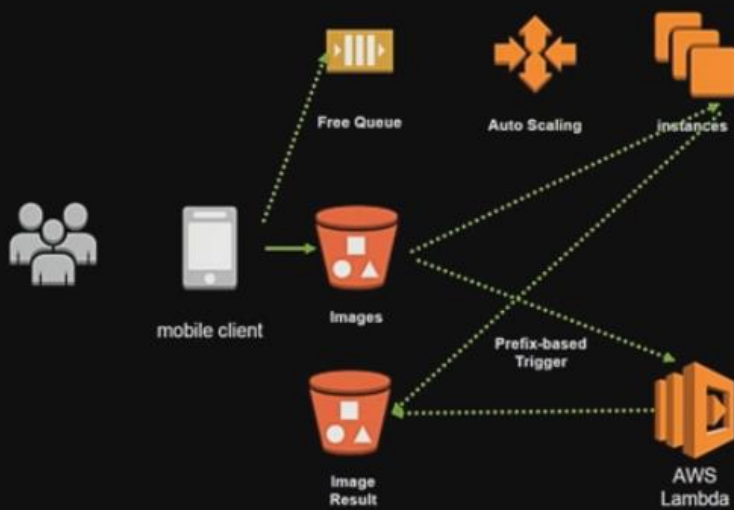
S3 will add the necessary permissions to invoke your Lambda function from this source bucket. See the [Developer Guide](#).

Save

Cancel

To set up an S3 notification, we do the above. We are creating an event notification function called **ImageProcessor** that responds to a **PUT** request in an S3 bucket and then triggers a lambda function called **CRWriteReport**.

Can Still Use SQS To Keep Free Tier Cheap



More optimization

Make free tier even cheaper using EC2 Spot Instances

Paid tier now gets instant response (remove the 10 minute delay/SLA) at low cost

We can now further optimize by having our Lambda function for our paid user tier and apply spot pricing to it also as above

Database Economic Architecture “Rules of Thumb”



Not everything is just relational or just NoSQL

Caches are usually more efficient and cost effective

✓ But you might have to do a little-more work up-front!

Do not conflate transactional DBs with analytical DBs

You will typically have “hot spot” tables or datasets – look to put these in NoSQL or Cache storage

Replace “Hot Spots” in RDS With DynamoDB

Example

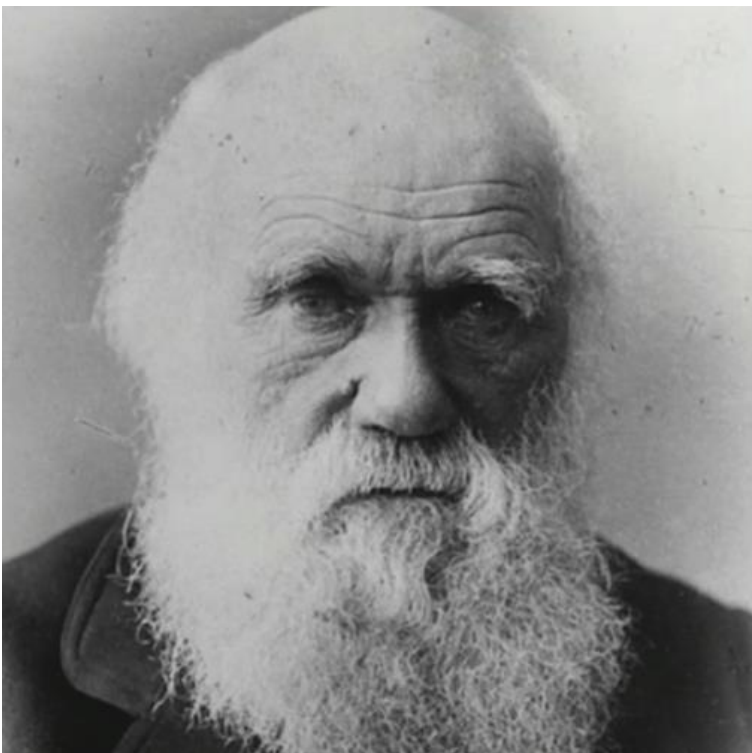
10,000 IOPS (80% Read/20% Write)

3TB Capacity

Multi-AZ RDS for MySQL

RDS: \$4,689.64

DynamoDB: \$2,887.70



Architectures
can evolve...

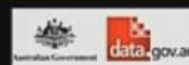
Example “Restroom Finder” API Microservice

Find me all the toilets in a particular postcode (zip code) in Australia!

Started with Python/Bottle/MySQL

Lookup table with 18,408 records

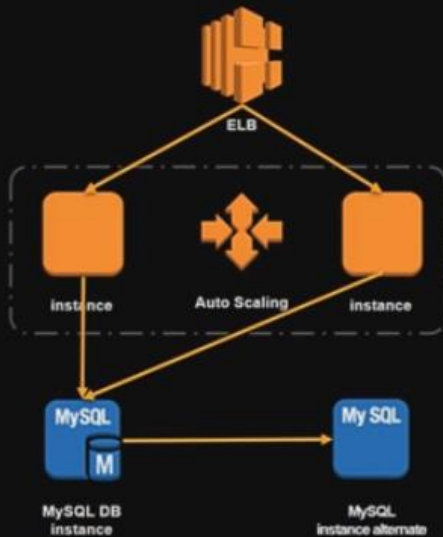
<http://data.gov.au/dataset/national-public-toilet-map>



```
17 @route('/postcode/<postcode>')
18 def postcode(postcode):
19
20     returnData = "INVALID input"
21     #Validate input - can only be 4 numerics
22     if len(postcode) == 4 and postcode.isdigit():
23
24         #MySQL Connection
25         cnx = mysql.connector.connect(user='[redacted]', password='[redacted]',
26                                     host='toiletdata.[redacted].ap-southeast-2.rds.amazonaws.com',
27                                     database='toiletdata')
28
29         #Query
30         cursor = cnx.cursor()
31
32         query = ("SELECT name, address1, town FROM toilets "
33                "WHERE postcode=" + postcode)
34
35         cursor.execute(query)
36
37         resultRow = ""
38
39         for (name, address1, town) in cursor:
40             resultRow = resultRow + json.dumps({'name': name, 'address1': address1, 'town': town}) + ","
41
42         resultRow = string.rstrip(resultRow, ",")
43
44         cursor.close()
45
46         cnx.close()
47         returnData = "[" + resultRow + "]"
48
49     return returnData
```

This microservice simply takes in a postcode value, validates it, then runs a query against our RDS database to get the list of public data in that postcode, and returns the list back as JSON data.

Typical EC2 Deployment - ElasticBeanstalk



Pros

- ✓ Scalable
- ✓ Familiar
- ✓ Reliable

Cons

- ✓ Need to manage API layer
- ✓ Throttling, security, etc.
- ✓ Storage layer heavy to ensure availability
- ✓ Need to patch/manage servers

We then spin up ElasticBeanStalk EBS to deploy our new microservice into as above

```
(vmv)[RestroomFinder] eb create
Enter Environment Name
(default is application-dev2): restroomfinder-prod
Enter DNS CNAME prefix
(default is restroomfinder-prod): restroomfinder-prod

Select a load balancer type
1) classic
2) application
(default is 1): 1
Creating application version archive "app-2ba8-161811_083816".
uploading: [#####] 100% Done...
Environment details for: restroomfinder-prod
Application name: application
Region: ap-southeast-2
Deployed Version: app-2ba8-161811_083816
Environment ID: e-6prn2akc2
Platform: 64bit Amazon Linux 2016.03 v2.1.6 running Python 2.7
Tier: WebServer-Standard
CNAME: restroomfinder-prod.ap-southeast-2.elasticbeanstalk.com
Updated: 2016-10-10 21:38:03.098000+00:00

Printing Status:
INFO: createEnvironment is starting.
INFO: Using elasticbeanstalk-ap-southeast-2-581371153492 as Amazon S3 storage bucket for environment data.
INFO: Created load balancer named: aseb-e-6-AR5E8Loo-190WQLD84G0Y
INFO: Environment health has transitioned to Pending. Initialization in progress (running for 22 seconds). There are no instances.
INFO: Created security group named: aseb-e-6prn2akc2-stack-AR5E8SecurityGroup-1C399GQ4H6ZP
INFO: Created Auto Scaling launch configuration named: aseb-e-6prn2akc2-stack-AR5E8AutoScalingLaunchConfiguration-1E38AUX0NE89
INFO: Added instance [i-0eb8c471] to your environment.
INFO: Created Auto Scaling group named: aseb-e-6prn2akc2-stack-AR5E8AutoScalingGroup-1M0J0PQ21D5E
INFO: Waiting for EC2 instances to launch. This may take a few minutes.
INFO: Created Auto Scaling group policy named: arn:aws:autoscaling:ap-southeast-2:581371153492:scalingPolicy:7372611f-f18e-4ca1-ab96-229f8b5c31b:autoScalingGro
upName/aseb-e-6prn2akc2-stack-AR5E8AutoScalingGroup-1M0J0PQ21D5E:policyName/aseb-e-6prn2akc2-stack-AR5E8AutoScalingScaleDownPolicy-CY0X21Q59H0
INFO: Created Auto Scaling group policy named: arn:aws:autoscaling:ap-southeast-2:581371153492:scalingPolicy:4b1ac8b-f2d1-41d9-9656-8b8dc77e3e2:autoScalingGro
upName/aseb-e-6prn2akc2-stack-AR5E8AutoScalingGroup-1M0J0PQ21D5E:policyName/aseb-e-6prn2akc2-stack-AR5E8AutoScalingScaleUpPolicy-1GCSA6U3350U1
INFO: Created CloudWatch alarm named: aseb-e-6prn2akc2-stack-AR5E8CloudWatchAlarm-1454126025C7
INFO: Environment health has transitioned from Pending to OK. Initialization completed 10 seconds ago and took 4 minutes.
INFO: Successfully launched environment: restroomfinder-prod

(vmv)[RestroomFinder] |
```

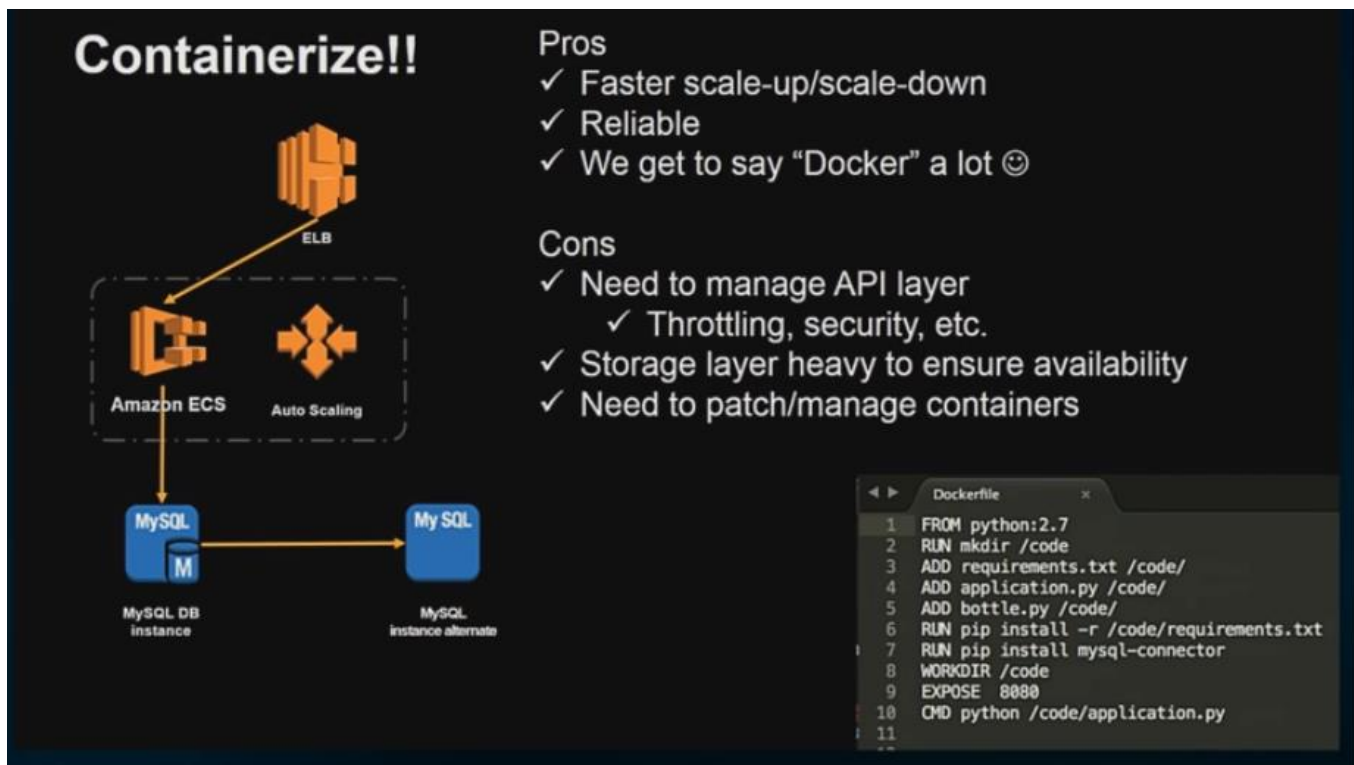
Use the **eb create** command to get the stack created in a few minutes

```

4. seah@145c8b2a2eb: ~/Projects/RestroomFinder (2af0)
(master)
(cvm) [restroomfinder] curl http://restroomfinder-prod.ap-southeast-2.elasticbeanstalk.com/postcode/2000
[[{"town": "Millers Point", "address": "Watson Road", "name": "Watson Road"}, {"town": "The Rocks", "address": "George Street", "name": "Cumberland Street"}, {"town": "Sydney", "address": "Elizabeth Street", "name": "Elizabeth Street"}, {"town": "Sydney", "address": "James Road", "name": "Hyde Park - North 1"}, {"town": "Sydney", "address": "James Road", "name": "Hyde Park - North 2"}, {"town": "Sydney", "address": "College Street", "name": "Hyde Park - North 3"}, {"town": "Sydney", "address": "Liverpool Street", "name": "Hyde Park South"}, {"town": "Haymarket", "address": "Hay Street", "name": "Belmore Park"}, {"town": "Sydney", "address": "York Street", "name": "Wynyard Park"}, {"town": "Sydney", "address": "Grassmore Street", "name": "Long Park"}, {"town": "Sydney", "address": "Mrs Macquaries Road", "name": "Mrs Macquaries Road 1"}, {"town": "Sydney", "address": "Mrs Macquaries Road", "name": "Mrs Macquaries Road", "name": "Royal Botanic Gardens Visitor Centre"}, {"town": "Sydney", "address": "Mrs Macquaries Road", "name": "The Gaiety", "name": "Sydney", "address": "Art Gallery Road", "name": "Art Gallery of NSW"}, {"town": "Sydney", "address": "Circular Quay", "name": "Circular Quay Train Station"}, {"town": "Sydney", "address": "Macquarie Street", "name": "Martin Place Train Station"}, {"town": "Sydney", "address": "Elizabeth Street", "name": "Museum Train Station"}, {"town": "Sydney", "address": "Prince Albert Road", "name": "St James Train Station"}, {"town": "Sydney", "address": "George Street", "name": "Town Hall Train Station"}, {"town": "Sydney", "address": "York Street", "name": "Wynyard Train Station"}, {"town": "Sydney", "address": "32 Martin Place", "name": "Colonial Centre"}, {"town": "Darling Harbour", "address": "Harbour Street", "name": "Darling Walk"}, {"town": "Sydney", "address": "Corner George and Margaret Streets", "name": "Metcentre"}, {"town": "Sydney", "address": "588 George Street", "name": "Pavilion Plaza"}, {"town": "Sydney", "address": "100 Convent Street", "name": "The Wintergarden"}, {"town": "Sydney", "address": "Corner of Bo thurst & Kent Streets", "name": "Town Hall Square"}, {"town": "Haymarket", "address": "Hay Street", "name": "Market City - Paddy's Market"}, {"town": "Haymarket", "address": "Hay Street", "name": "Market City - Level 1"}, {"town": "Haymarket", "address": "Hay Street", "name": "Market City - Level 2 Fashion"}, {"town": "Haymarket", "address": "Hay Street", "name": "Market City - Level 3 Foodcourt"}, {"town": "Haymarket", "address": "Edy Avenue", "name": "Central Train Station"}, {"town": "Darling Harbour", "address": "Wheel Road", "name": "Darling Harbour - Harbourside East"}, {"town": "Darling Harbour", "address": "Pier Street", "name": "Darling Harbour - Pier Street"}, {"town": "Darling Harbour", "address": "Liverpool Street", "name": "Darling Harbour - Tumbalong Park West"}, {"town": "Darling Harbour", "address": "Corner Harbour & Liverpool Streets", "name": "Darling Harbour - Tumbalong Park East"}, {"town": "Darling Harbour", "address": "Off Darling Drive", "name": "Sydney Convention & Exhibition Centre Car Park"}, {"town": "Circular Quay", "address": "Alfred Street", "name": "Circular Quay - Alfred Street"}, {"town": "Sydney", "address": "31 Alfred Street", "name": "Customs House Library"}, {"town": "Haymarket", "address": "774 George Street", "name": "Haymarket Library"}, {"town": "Sydney", "address": "86-108 Castlereagh Street", "name": "David Jones - Castlereagh Street"}, {"town": "Sydney", "address": "67-77 Market Street", "name": "David Jones - Market Street"}, {"town": "Sydney", "address": "227 Smithfield Road (Cnr The Horsely Drive)", "name": "Giles & Iress Smithfield"}, {"town": "Sydney", "address": "456 Kent Street", "name": "Town Hall House Public Toilet"}, {"town": "Sydney", "address": "2 Huntley Street", "name": "The Grounds Alexandria"}, {"town": "Haymarket", "address": "Pitt Street", "name": "Central Station Platform 1"}, {"town": "Sydney", "address": "Wharf Road", "name": "Cockle Bay Restaurant Precinct"}, {"town": "Sydney", "address": "Cockle Bay Wharf", "name": "Darling Harbour - Harbourside West"}, {"town": "Sydney", "address": "137 Pitt Street", "name": "Mid City Centre"}, {"town": "Sydney", "address": "455 George Street", "name": "Queen Victoria Building"}, {"town": "Sydney", "address": "210 Pitt Street", "name": "Piccadilly Centre"}, {"town": "Sydney", "address": "412-414 George Street", "name": "Strand Arcade"}, {"town": "Sydney", "address": "44 Hunter Street", "name": "Slater & Gordon"}, {"town": "Sydney", "address": "Cor Elizabeth & Market Streets", "name": "David Jones Store"}, {"town": "Sydney", "address": "Pitt Street Mall", "name": "Mid City Shopping Centre"}, {"town": "Sydney", "address": "436 George Street", "name": "Myer"}, {"town": "Sydney", "address": "436 George Street", "name": "Myer"}, {"town": "Haymarket", "address": "Pitt Street", "name": "Central Station Grand Concourse"}, {"town": "Millers Point", "address": "Hickson Road", "name": "Roslyn Packer Theatre Ground Floor"}, {"town": "Millers Point", "address": "22 Hickson Road", "name": "Roslyn Packer Theatre Level 2"}]]
(cvm) [restroomfinder] curl http://restroomfinder-prod.ap-southeast-2.elasticbeanstalk.com/postcode/1
(master)

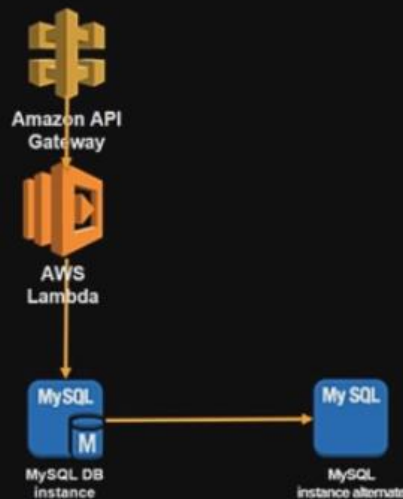
```

We can now query the service as above



We can take the code, convert it into a Docker container, put it into ECS, and deploy using the new architecture above. The instances now spin up faster but it is basically still the same

Lambda-ize!!



Pros

- ✓ Compute scaling perfectly matched to load
- ✓ Reliable
- ✓ Ability to manage API
- ✓ No server patching

Cons

- ✓ Storage layer heavy to ensure availability

We are now serverless and only spin up compute only when we need to do some real work. But we still have the storage layer unchanged yet!

NoSQL-ize!!



Pros

- ✓ Compute scaling perfectly matched to load
- ✓ Reliable
- ✓ Ability to manage API
- ✓ No server patching
- ✓ No DB Patching
- ✓ DB tuned to performance workload
- ✓ We can say "serverless" a lot 😊

Cons

- ✓ Less familiar deployment and operational model

Now we use DynamoDB to go fully serverless.


```

26
27 @app.route('/postcode/<postcode>')
28 def postcode(postcode):
29
30     returnData = "INVALID input"
31     resultRow = ""
32     #Validate input - can only be 4 numerics
33     if len(postcode) == 4 and postcode.isdigit():
34
35         dynamodb = boto3.resource('dynamodb', region_name='ap-southeast-2')
36
37         table = dynamodb.Table('toiletData')
38
39         response = table.query(
40             KeyConditionExpression=Key('Postcode').eq(postcode)
41         )
42
43         for i in response['Items']:
44             resultRow = resultRow + "{name:" + str(i['Name']) + ",address1:" + i['Address1'] + ",town:" + i['Town'] + "},"
45
46         resultRow = string.rstrip(resultRow, ",")
47
48         returnData = "[" + resultRow + "]"
49
50     return returnData

```

This is the new code. We have replaced the MySQL call with a DynamoDB call. We are connecting into our DynamoDB table using the AWS Python SDK (or AWS NodeJS SDK), run a query get the items we need, and return the data back as JSON.



Zappa (like serverless) is an OSS deployment tool that can deploy to lambda using a config file

```

(venv) [RestroomfinderZappa] zappa deploy prod
Packaging project as zip...
Uploading restroomfinderz-prod-1476135873.zip (6.1MiB)...
100%|#####| 6.43M/6.43M [00:01:00:00, 4.00M/s]
Scheduling..
Scheduled restroomfinderz-prod-zappa-keep-warm-handler.keep_warm_callback!
Uploading restroomfinderz-prod-template-1476135881.json (85.7KiB)...
100%|#####| 87.7K/87.7K [00:00:00:00, 172K/s]
Waiting for stack restroomfinderz-prod to create (this can take a bit)...
94%|#####| 15/16 [00:31:00:02, 2.18s/res]
Deploying API Gateway..
Deployment complete! https://y3fgkvwcs.execute-api.ap-southeast-2.amazonaws.com/prod
(venv) [RestroomfinderZappa]

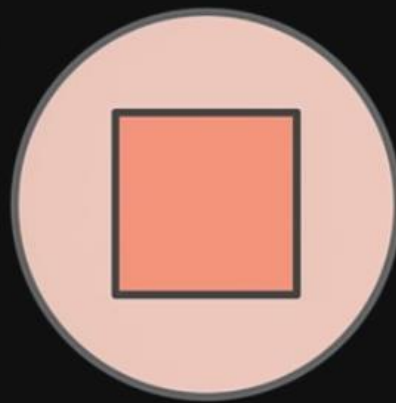
```

We have the microservice deployed

```
(venv) [RestroomFunderZappa] curl https://y3fgvlewa.execute-api.ap-southeast-2.amazonaws.com/prod/postcode/2000
[({"name": "Watson Road", "address": "Watson Road", "town": "Millers Point"}, {"name": "Cumberland Street", "address": "Geroge Street", "town": "The Rocks"}, {"name": "Hyde Park - North", "address": "Elizabeth Street", "town": "Sydney"}, {"name": "Elizabeth Street", "address": "Elizabeth Street", "town": "Sydney"}, {"name": "Hyde Park - North 1", "address": "James Road", "town": "Sydney"}, {"name": "Hyde Park - North 2", "address": "James Road", "town": "Sydney"}, {"name": "Hyde Park - North 3", "address": "College Street", "town": "Sydney"}, {"name": "Hyde Park South", "address": "Liverpool Street", "town": "Sydney"}, {"name": "Belmore Park", "address": "Hay Street", "town": "Haymarket"}, {"name": "Wynyard Park", "address": "York Street", "town": "Sydney"}, {"name": "Long Park", "address": "Crosvenor Street", "town": "Sydney"}, {"name": "Mrs Macquaries Road 1", "address": "Mrs Macquaries Road", "town": "Sydney"}, {"name": "Mare & Foal Statue", "address": "Mrs Macquaries Road", "town": "Sydney"}, {"name": "Royal Botanic Gardens Visitor Centre", "address": "Mrs Macquaries Road", "town": "Sydney"}, {"name": "The Collyer", "address": "Mrs Macquaries Road", "town": "Sydney"}, {"name": "Art Gallery of NSW", "address": "Art Gallery Road", "town": "Sydney"}, {"name": "Circular Quay Train Station", "address": "Alfred Street", "town": "Circular Quay"}, {"name": "Martin Place Train Station", "address": "Macquarie Street", "town": "Sydney"}, {"name": "Museum Train Station", "address": "Elizabeth Street", "town": "Sydney"}, {"name": "St James Train Station", "address": "Prince Albert Road", "town": "Sydney"}, {"name": "Town Hall Train Station", "address": "George Street", "town": "Sydney"}, {"name": "Wynyard Train Station", "address": "York Street", "town": "Sydney"}, {"name": "Colonial Centre", "address": "52 Martin Place", "town": "Sydney"}, {"name": "Darling Walk", "address": "Harbour Street", "town": "Darling Harbour"}, {"name": "Metcentre", "address": "Corner George and Margaret Streets", "town": "Sydney"}, {"name": "Pavilion Plaza", "address": "588 George Street", "town": "Sydney"}, {"name": "The Wintergarden", "address": "11 O'Connell Street", "town": "Sydney"}, {"name": "Town Hall Square", "address": "Corner of Bathurst & Kent Streets", "town": "Sydney"}, {"name": "Market City - Paddy's Market", "address": "Hay Street", "town": "Haymarket"}, {"name": "Market City - Level 1", "address": "Hay Street", "town": "Haymarket"}, {"name": "Market City - Level 2 Fashion", "address": "Hay Street", "town": "Haymarket"}, {"name": "Market City - Level 3 Foodcourt", "address": "Hay Street", "town": "Haymarket"}, {"name": "Central Train Station", "address": "Eddy Avenue", "town": "Haymarket"}, {"name": "Darling Harbour - Harbourside East", "address": "Wheat Road", "town": "Darling Harbour"}, {"name": "Darling Harbour - Pier Street", "address": "Pier Street", "town": "Darling Harbour"}, {"name": "Darling Harbour - Tumbalong Park West", "address": "Liverpool Street", "town": "Darling Harbour"}, {"name": "Darling Harbour - Tumbalong Park East", "address": "Liverpool Street", "town": "Darling Harbour"}, {"name": "Sydney Convention & Exhibition Centre Car Park", "address": "Off Darling Drive", "town": "Darling Harbour"}, {"name": "Circular Quay - Alfred Street", "address": "Alfred Street", "town": "Circular Quay"}, {"name": "Customs House Library", "address": "31 Alfred Street", "town": "Sydney"}, {"name": "Haymarket Library", "address": "774 George Street", "town": "Haymarket"}, {"name": "David Jones - Castlereagh Street", "address": "86-188 Castlereagh Street", "town": "Sydney"}, {"name": "David Jones - Market Street", "address": "67-77 Market Street", "town": "Sydney"}, {"name": "Coles Express Smithfield", "address": "227 Smithfield Road (Cor The Horsely Drive)", "town": "Sydney"}, {"name": "Town Hall House Public Toilet", "address": "456 Kent Street", "town": "Sydney"}, {"name": "The Grounds Alexandria", "address": "2 Huntley Street", "town": "Sydney"}, {"name": "Central Station Platform 1", "address": "Pitt Street", "town": "Haymarket"}, {"name": "Cockle Bay Restaurant Precinct", "address": "Wheat Road", "town": "Sydney"}, {"name": "Darling Harbour - Harbourside West", "address": "Cockle Bay Wharf", "town": "Sydney"}, {"name": "Mid City Centre", "address": "197 Pitt Street", "town": "Sydney"}, {"name": "Queen Victoria Building", "address": "455 George Street", "town": "Sydney"}, {"name": "Piccadilly Centre", "address": "218 Pitt Street", "town": "Sydney"}, {"name": "Strand Arcade", "address": "412-414 George Street", "town": "Sydney"}, {"name": "Slaters & Gordon", "address": "44 Market Street", "town": "Sydney"}, {"name": "David Jones Store", "address": "Cor Elizabeth & Market Streets", "town": "Sydney"}, {"name": "Mid City Shopping Centre", "address": "Pitt Street Mall", "town": "Sydney"}, {"name": "Myer", "address": "436 George Street", "town": "Sydney"}, {"name": "Myer", "address": "436 George Street", "town": "Sydney"}, {"name": "Central Station Grand Concourse", "address": "Pitt Street", "town": "Haymarket"}, {"name": "Ruslyn Packer Theatre Ground Floor", "address": "Hickson Road", "town": "Millers Point"}, {"name": "Ruslyn Packer Theatre Level 2", "address": "22 Hickson Road", "town": "Millers Point"}])
(venv) [RestroomFunderZappa]
```

We can now use curl to call the API Gateway endpoint as above

Common Anti-Patterns



HA vs. Cost Optimization

Product owners often push for cost optimization

- Ignoring their availability requirements
- Until something fails

FIRST availability, **THEN** cost!

Economical Mitigation For AZ-Level Disruption

Example Always maintain 12 instances even if one AZ is disrupted



24
Instances



18
Instances

Always deploy in multi AZs for true HA configuration like above

Economical Mitigation For AZ-Level Disruption

Example Always maintain 12 instances even if one AZ is disrupted



24
Instances



18
Instances

25%
saving

We can remove the one shown to maintain HA by spending 25% more. This is a simple config change in your auto-scaling group to get this layout.

The Cost of Using Your DB as a BLOB Store

Databases (particularly RDBMSs) **make poor BLOB stores!**

Negative in terms of performance

Negative in terms of management

Negative in terms of cost

Storing Your BLOB in the RDBMS – Example

```
CREATE OR REPLACE TABLE  
user_images  
(userid int,  
imageguid char(36),  
image BLOB);
```

For example

Each Image = 2MB

2,049K per row

3 million rows = 5.862 TB

RDS MySQL cost = \$3,279.56 (SYD)
per month (multi-AZ, Standard
Storage)

Assuming we are storing our BLOBs in a database table as above,

Store BLOB in S3, Metadata in RDS

```
CREATE OR REPLACE  
TABLE user_images  
(userid int,  
imageguid char(36),  
imageURL  
VARCHAR(1000));
```

Per row we now store 2K in the RDBMS

3 Million rows = 5.722 GB

S3 Storage = 5.862TB = \$194.94 (storage)
+ 3MM Puts, 3MM Gets = \$212.93
(inclusive, GETs, PUTs, Storage)

RDS SQL cost = \$1,663.30 (SYD) per
Month

Total: \$1,876.23

42%
saving

You then just point your code to point to where the BLOB is in S3 to get it from there.

A New Domain for Architecture



Availability



Performance



Security



Function



Economy

Economy of Architecture



Economy

- ✓ Same or better outcome for the lowest cost
- ✓ Trial and change **during** the lifetime of the system
- ✓ Radical changes are possible – driven by economics
- ✓ Changes in services that can be consumed
- ✓ Transaction cost + operational cost

One More Thing! Free Trial of Trusted Advisor

For attendees of this session!

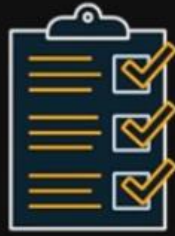
Free trial begins on 6 December 2016

Runs for 30 days

Makes the full suite of checks and best practice recommendations available for customers not already on business/enterprise support plans

No action required; just log in and start using it!

<https://console.aws.amazon.com/trustedadvisor>



**Remember to complete
your evaluations!**