Cloud Cost Optimization

AWS community builder

#whoami: yogesh sharma

AWS community builder

- Cloud Architect @ Tata Consultancy Services
- Overall 10+ years of experience, 5+ years of experience in Cloud, DevOps
- ML and Data Enthusiast
- AWS Community Builder Program (First Batch)
- 3x AWS Certifications
- Technical Speaker and Blogger
- Owner of https://multicloudwarrior.com/

aws certified Solutions Architect Professional

aws certified Solutions Architect Associate

aws certified Developer Associate

https://www.linkedin.com/in/sharma-yogesh/

What to expect

- Cloud Cost Optimization- What, Why
- Pillars for Cost Optimization
- Cost Optimization Process
- Checklist
- Latest Trends
- Q&A

We all love Cloud and its Services…

AWS is for builders

You build it,

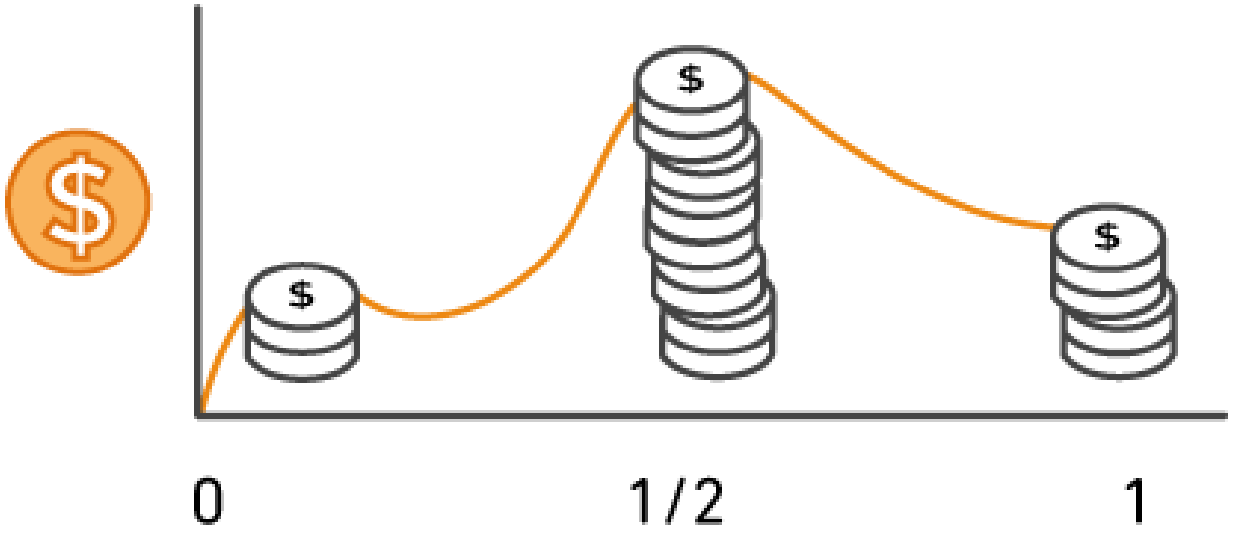
You run it,

You optimize it

But are we fully aware about Cloud Financials

Are we spending too much on unnecessary stuff knowingly/unknowingly?

Pay for what you use



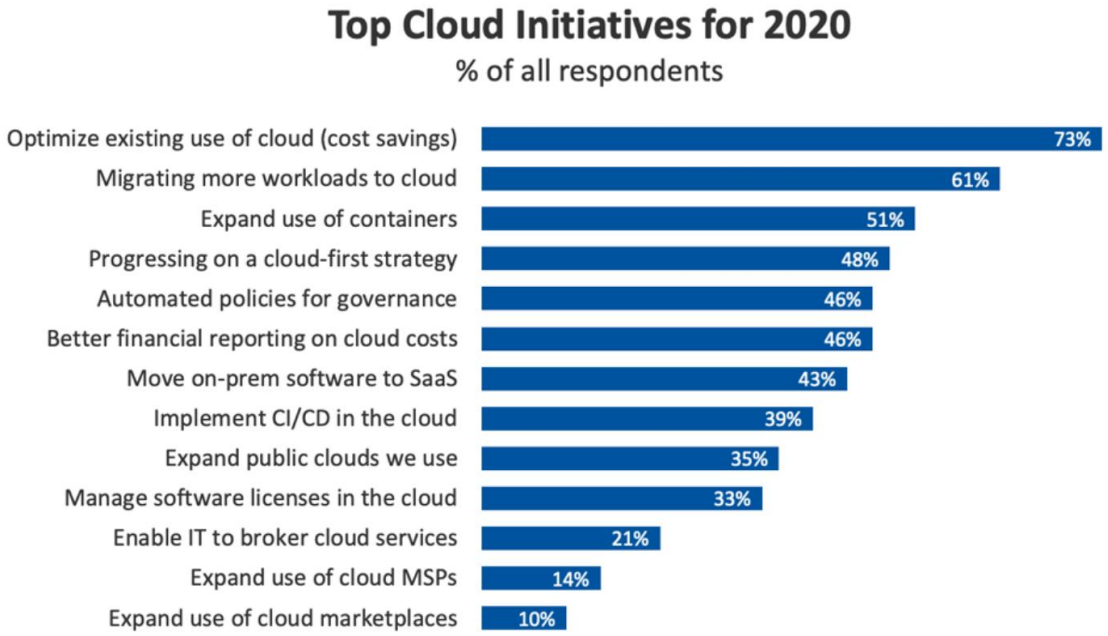
Bill Formula:

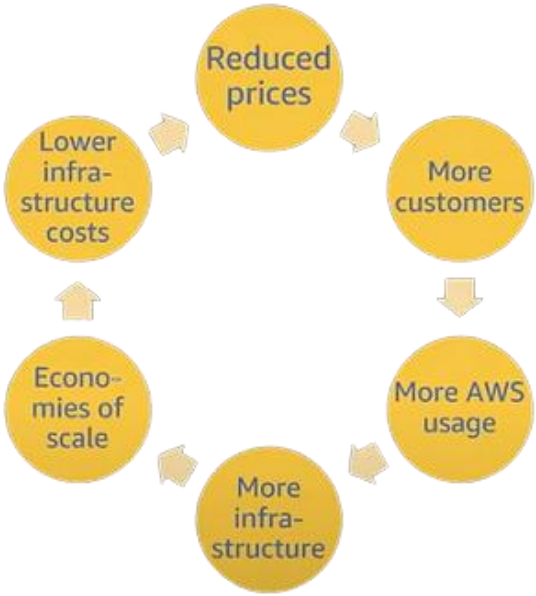
Spend = Usage * Rate

But how would you ensure that you pay what you actually need?

Cloud Cost Optimization

Managing costs is a challenge for organizations using public cloud services but also an opportunity to drive efficient consumption of IT.





Pillars for Cost Optimization

Pillars for Cost Optimization

Right-sizing

Elasticity

Pricing Models

Storage Classes

Measure and Mechanisms

Designing for Cost

Right-sizing

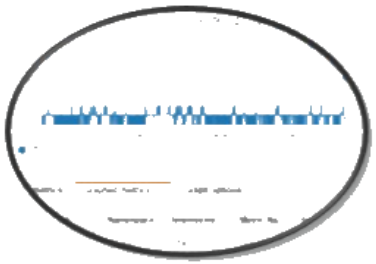
m4.4xlarge
$1.72 per hour

1. Provision

2. Check Metrics
(CPU, RAM, Network)

m4.large
$0.215 per
hour

3. Right-Size



4. Review Performance

86%
Saving

5. Save

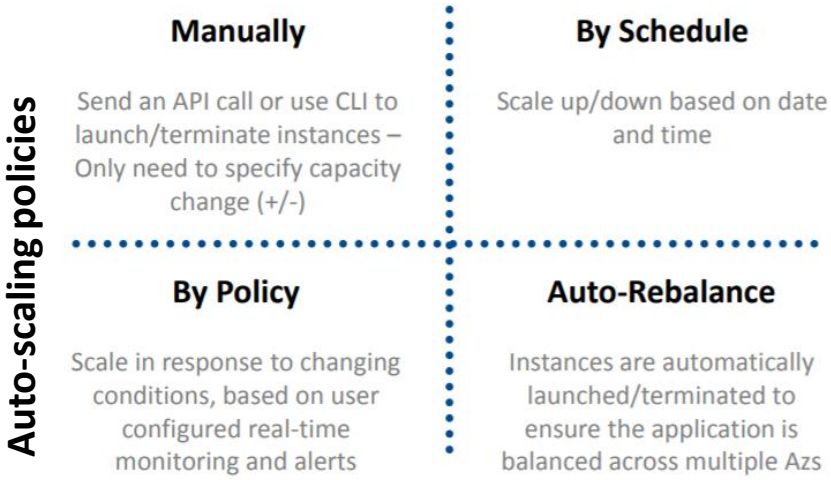
Start	Tune	Spread
Choose instance that meets your basic requirements best	Change instance size up or down based upon monitoring	Run instances across multiple availability zones
Match memory & virtual cores	Use trusted advisor to assess	Smaller sizes equals greater granularity

Elasticity

- Scale up and down to meet capacity requirements based upon CPU, RAM, network etc.
- Automatically turn nonproduction off outside of working hours

Think about:

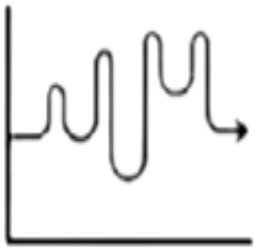
- AWS Auto Scaling and EC2 Fleet
- AWS Instance Scheduler
- AWS CloudWatch



Pricing Models

On- Demand

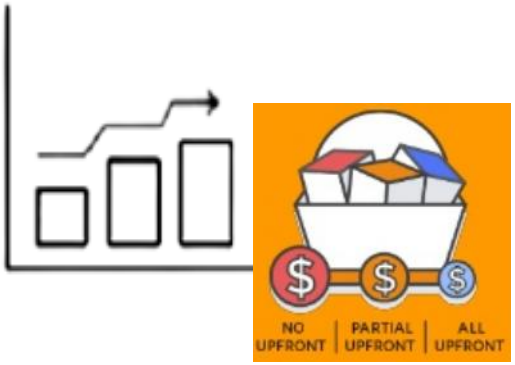
Pay-for-compute capacity by the second with no long-term commitments



Spiky workloads to Define needs

Savings Plans & Reserved Instances

Make a commitment and receive a significant discount off compute



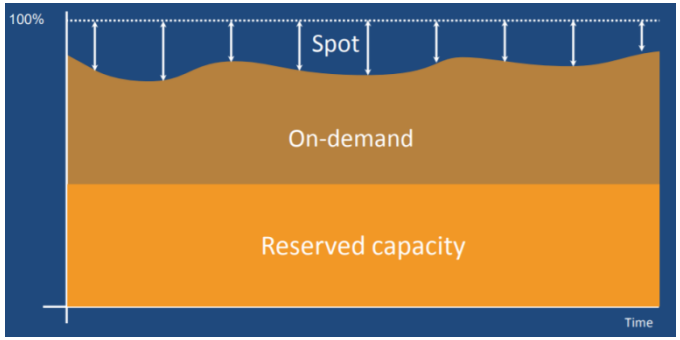
Committed & steady-state usage

Spot Instances

Spare AWS EC2 Capacity at saving up to 90% off On-Demand prices

If your bid > spot price → get an instance
If your bid < spot price → instance is terminated

Fault-tolerant, flexible, stateless workloads



Storage Classes

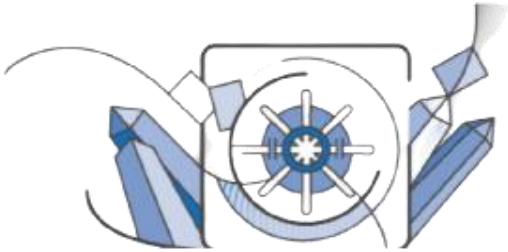
HOT

Amazon S3

WARM

Infrequent Access

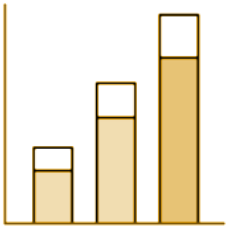
COLD



Amazon Glacier

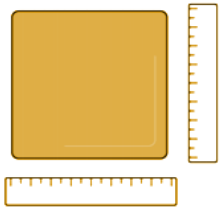
$0.125/GB-month	$0.10/GB-month (20% cheaper)	$0.045/GB-month (64% cheaper)	$0.025/GB-month (80% cheaper)
Provisioned IOPS SSD (io1)	General Purpose SSD (gp2)	Throughput Optimized HDD (st1)	Cold HDD (io1)

Measure and Mechanisms



Analyze spend by BU, account, service, etc.

What has changed and why?



Instance utilization metrics.

Are the instances of the right size and well utilized?

Peak vs. off-peak cost and usage.

Are instances turned off when they are not needed?

RI coverage and utilization, Spot usage.

Are RIs and Spot being used?

Designing for Cost

Consolidated Billing
Using AWS Organization

Go Serverless

Unused Elastic IPs

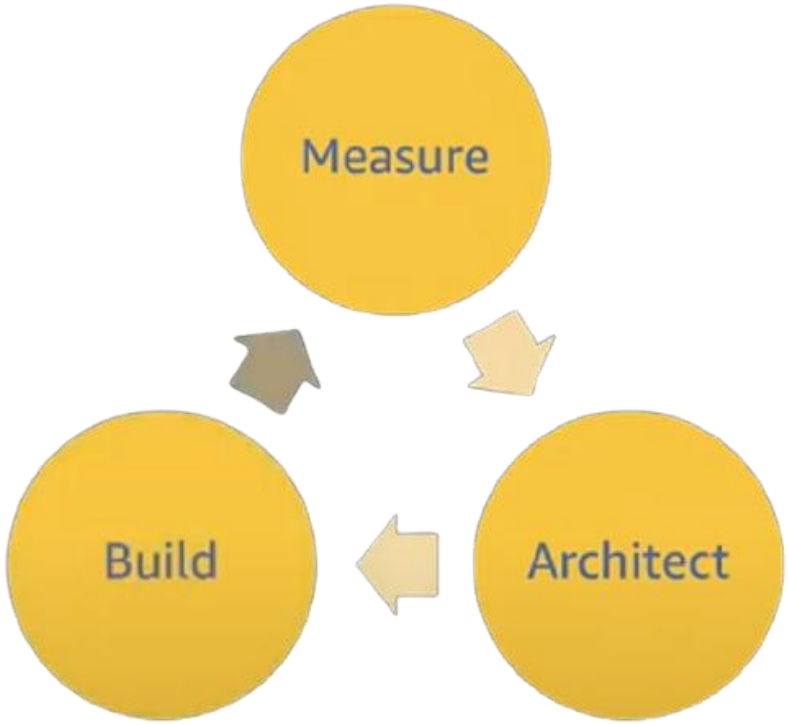
CloudFront

Containerization

Managed RDS

Cost Optimization Process

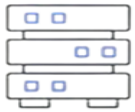
Cost Optimization Process



Biggest Strength- Architecture Flexibility

Main goal- Avoid unnecessary stuff

Unnecessary Resources



Idling Resources

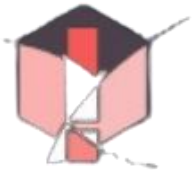
Repetitive Work

Measure: Monitor your workloads

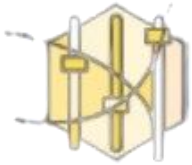
Set up metrics to defines success and track progress
"What KPI makes sense for this workload?"

% Instances turned off daily
%instances right sized
%RI Utilization

Monitor AWS Resources

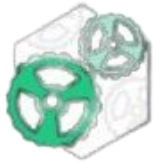


Set Alarms



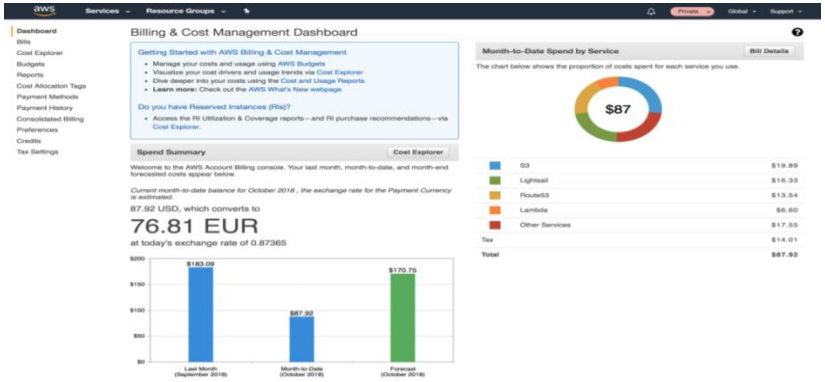
Monitor Custom Metrics

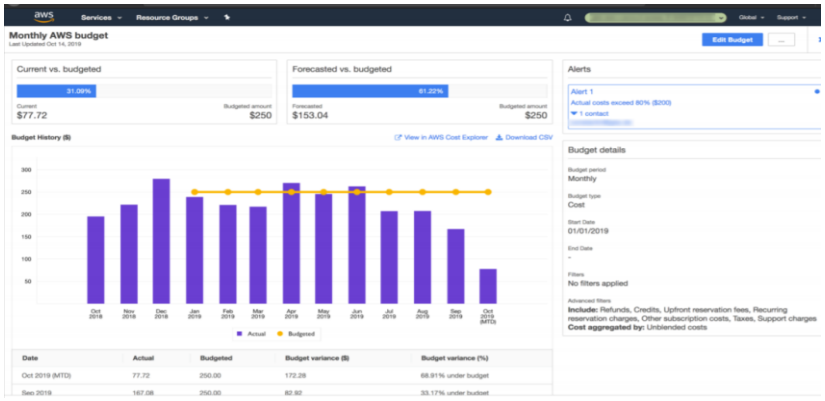
View Graphs and Statistics

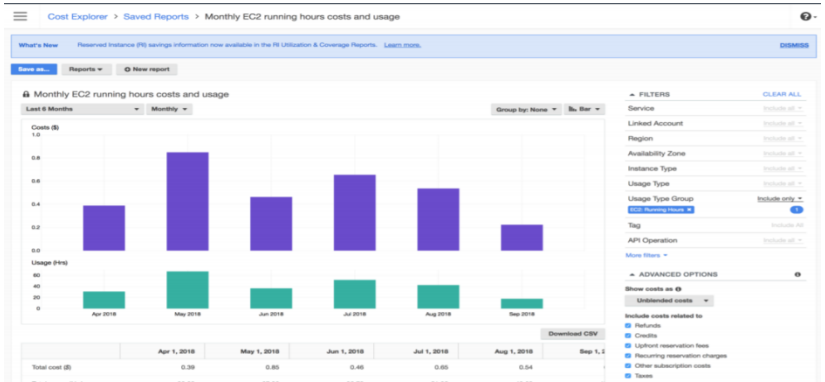


Monitor & React to
Resource Changes

Measure- AWS Billing/Cost Dashboard







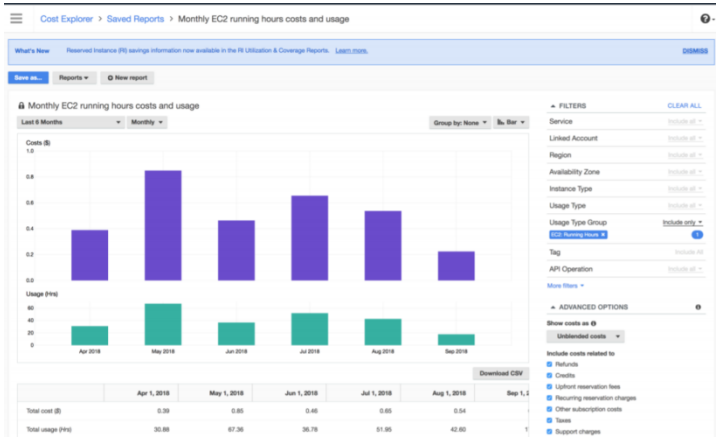
- Use AWS Billing and Cost Management Dashboard
- Set up your monthly AWS budget
- Dive deep with AWS Cost Explorer

Also checkout-

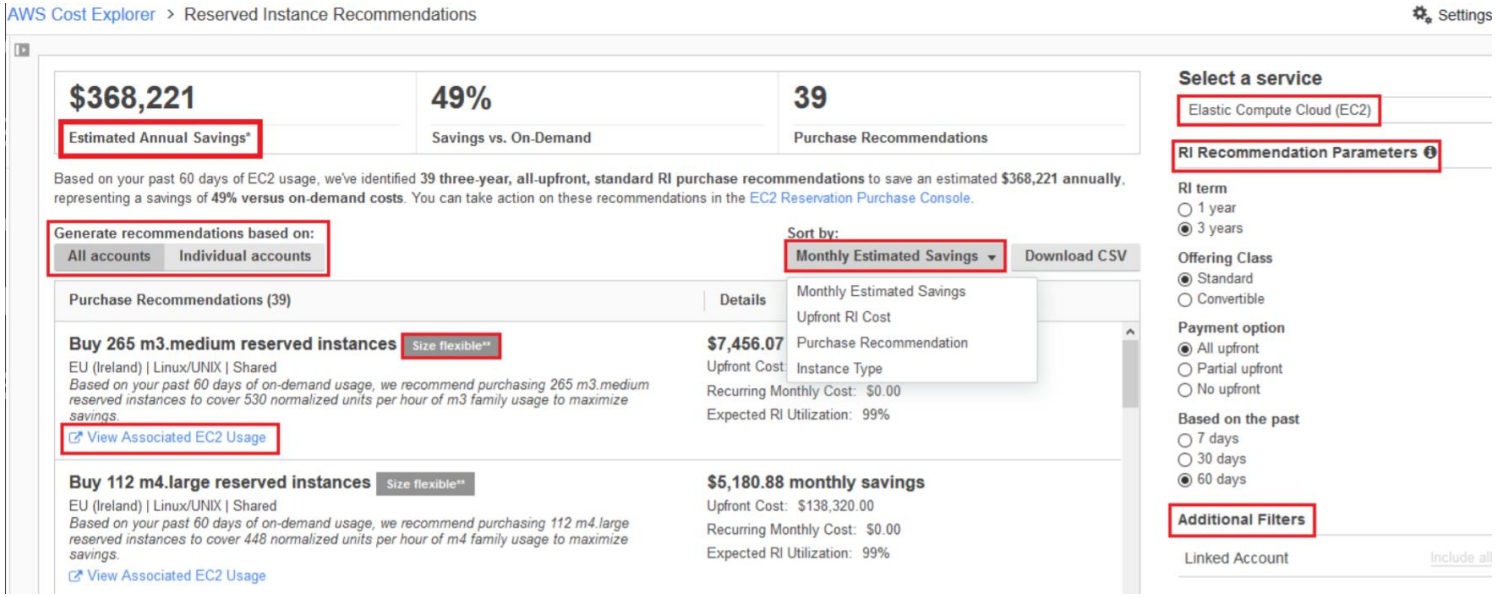
https://calculator.s3.amazonaws.com/index.html
https://calculator.aws/
https://aws.amazon.com/economics/

AWS community builder

Measure: Cost Explorer



Use Cost Explorer Personalized Recommendations and decide to go with Reserved Instances Category- Standard, Convertible and Scheduled RIs



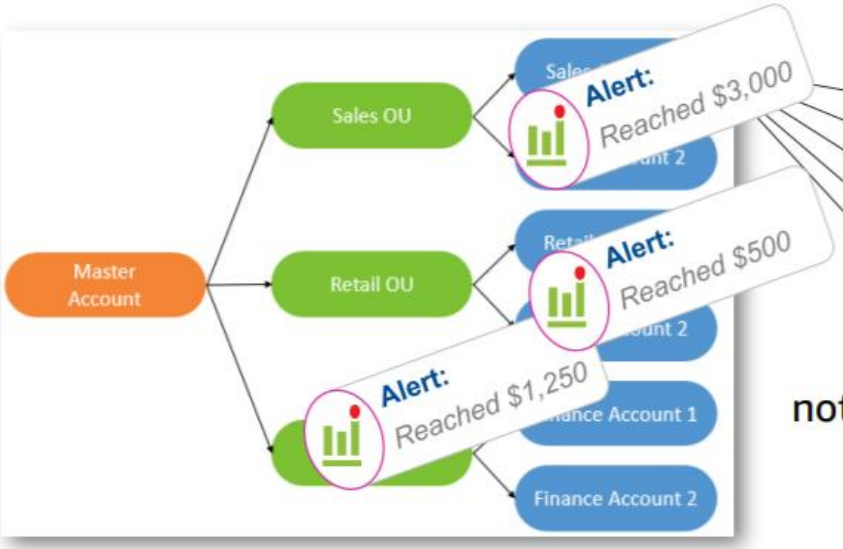
Architecture- Using AWS Organizations Consolidated Billing

Control AWS service use across accounts

Automate account creation

Consolidate billing and usage reporting

- **One Bill** for multiple accounts
- **Easy Tracking** of account charges (e.g., download CSV of cost data)
- **Volume Discounts** can be reached faster with combined usage
- **Reserved Instances** are shared across accounts (including RDS Reserved DBs)



Cost allocation tags

- Environment
- Project
- Team
- Application ID
- Cost center

Alerts can provide notifications when budget is reached

Architecture- Establish Tagging Strategy

What's in a Tag

Resource Allocation
Search, filter & categorize

Automation
Autoscaling & Scheduling

Cost Allocation
Reports & chargebacks

Control & Compliance
Restricted resource access through IAM

Using tags to establish visibility & chargebacks

Stakeholders	Examples
• Finance	• Cost center
• Engineering	• Application or workload
• Line of business owners	• User
• IT	• Expiration date
• Security	• Automation support

Checklist

- Turn off unused instances- Use instance start and stop
- Delete unused/orphan volumes
- Stop paying for Idle EC2 and RDS instances and Redshift Clusters
- Understand and forecast your costs using AWS Cost Explorer and AWS Budget
- Use consolidated billing to avail volume discount across accounts
- Turn on/off whole architecture setups Using AWS CloudFormation, Terraform
- Automate Everything- Automate your instance start/stop cycle
- Choose EC2 Spot Instances
- Use Compute Savings Plan
- Use Reserved Instances
- Embrace latest trends and learn from Industry
  - Adopt Micro-service architecture
  - Modernize applications to make use of provider managed services when these are more cost-effective.
  - Consider Serverless architecture- no idle capacity
  - Adopt Containerization
    - Containers with EC2-Spot, Fargate with Spot
  - Enable S3-Intelligent Tiering- Apply Life cycle transitioning on S3 Objects

Latest Trends- Launched in re:Invent 2020

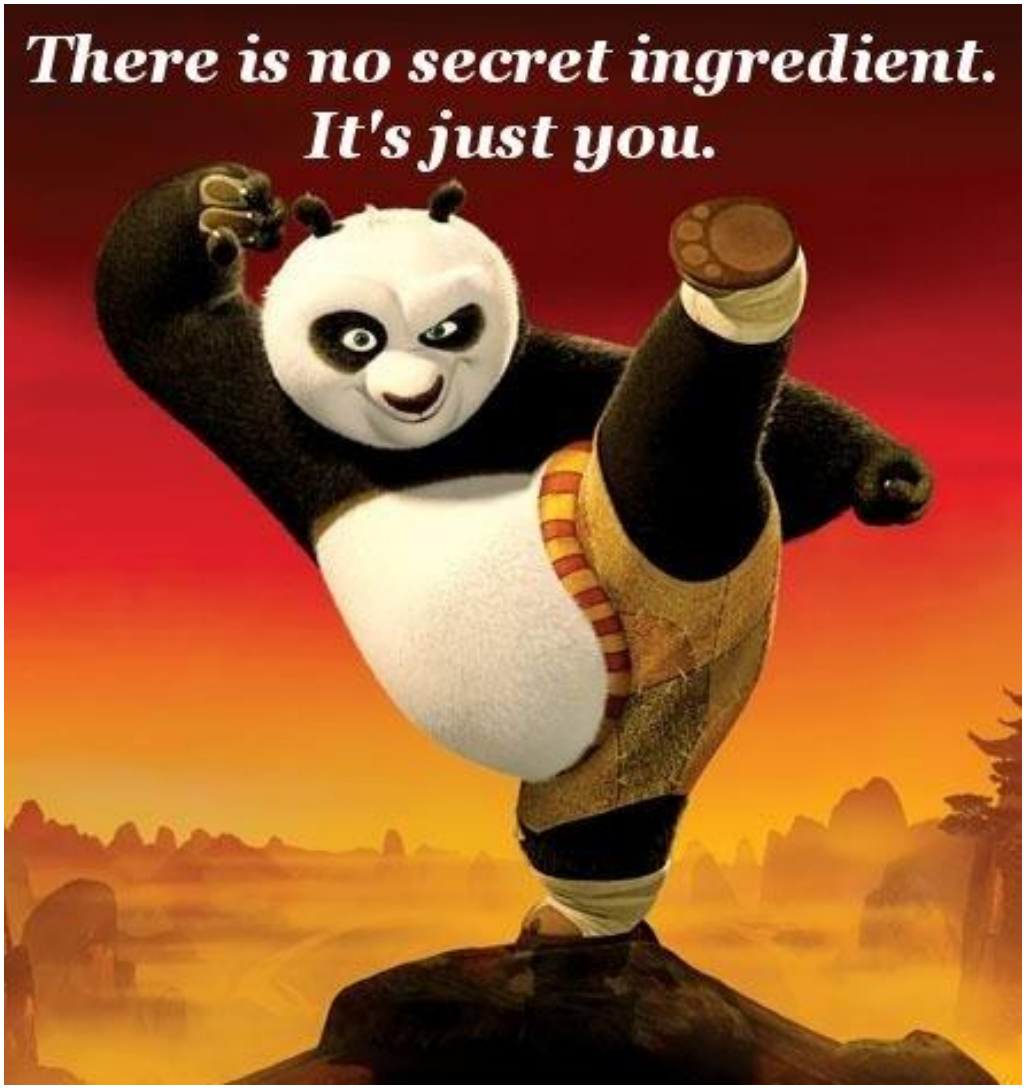
- New for AWS Lambda – 1ms Billing Granularity Adds Cost Savings

  https://aws.amazon.com/blogs/aws/new-for-aws-lambda-1ms-billing-granularity-adds-cost-savings/

- New – Amazon EBS gp3 Volume Lets You Provision Performance Apart From Capacity

  https://aws.amazon.com/blogs/aws/new-amazon-ebs-gp3-volume-lets-you-provision-performance-separate-from-capacity-and-offers-20-lower-price/

  It lets customers independently increase IOPS and throughput without having to provision additional block storage capacity, paying only for the resources they need.



Thank you!