Amazon Web Services – RIs vs. Savings Plans

- Reserved Instances (RIs)
  - What are they?
  - Benefits / Drawbacks
- Savings Plans
  - What are they?
  - Benefits / Drawbacks
  - No-brainer? 99% of the time, invest in Compute Savings Plans! – *but not so fast!*

Reserved Instances

- What are they?
  - Discounted pricing (compared to On-Demand) in exchange for commitment to a utilization level for 1 or 3 year
  - Two types: Standard & Convertible
    - 1-yr, No Upfront, Standard, t3.medium, Linux, regional scope, us-east-1, shared tenancy @ $0.0261/hr (37% discount)
    - 3-yr, All Upfront, Convertible, m5.xlarge, Windows, zonal scope, us-east-1a, shared tenancy @ $0.272/hr (28% discount)

Features	Limitations
• Get up to 72% discount pricing from OnDemand	• Pay for RIs even if they are underutilized
• Reserve capacity with Zonal RIs (AZ)	• Restricted to AZ; no instance size flexibility
• Regional RIs (AZ) & instance size flexibility for systems running Linux OS	• Constrained to reduce costs even if instances are oversized (e.g. wrong family, too much resource)
• Standard RIs can be sold in AWS marketplace	• Governed by market prices; sold at less than purchase value
• Convertible RIs offer more flexibility (Instance family, OS, tenancy)	• Not automated; manage exchange process & interpret complex exchange rules • Minimize true up costs during exchange

Savings Plans

- What are they?
  - AWS Introduced Savings Plans in Fall 2019 to simplify long-term purchase commitments
  - Discounted pricing (compared to On-Demand) in exchange for commitment to a dollar spend for 1 or 3 year
  - Two types: EC2 Savings Plan & Compute Savings Plan
    - 1-yr, No Upfront, EC2 SP, us-east-1, t3 @ $0.50/hr
    - 3-yr, All Upfront, Compute SP @ $10/hr

Comparing Reserved Instances & Savings Plans

Comparing Reserved Instances & Savings Plans

- Advantages of Savings Plans over Reserved Instances
  - Discounts are automatically applied. No management overhead
  - Regional Flexibility – Compute SPs can be applied to any region
  - Service Flexibility – Compute SPs can be shared amongst all compute services EC2, EKS, Fargate, Lambda etc…

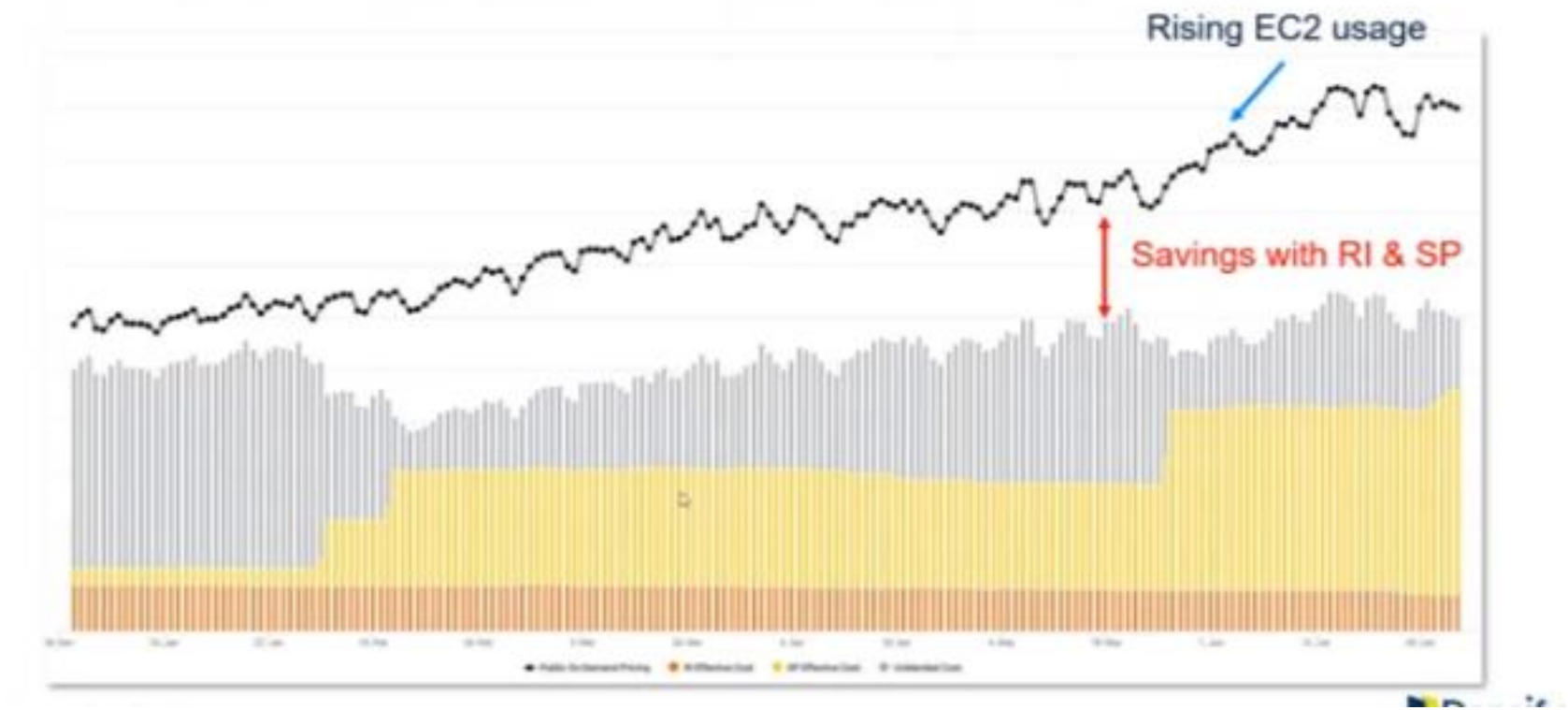
- Advantages of Reserved Instances over Savings Plans
  - Standard RIs can be sold in marketplace
  - RIs can be purchased for RDS, Redshift, Elasticache
  - Capacity reservations can be made with Zonal RIs
  - For SUSE EC2 instances, much higher discounts are available via RIs vs. SPs

- It seems like a no-brainer to move to Compute Savings Plans for 99%, right? – *but not so fast!*

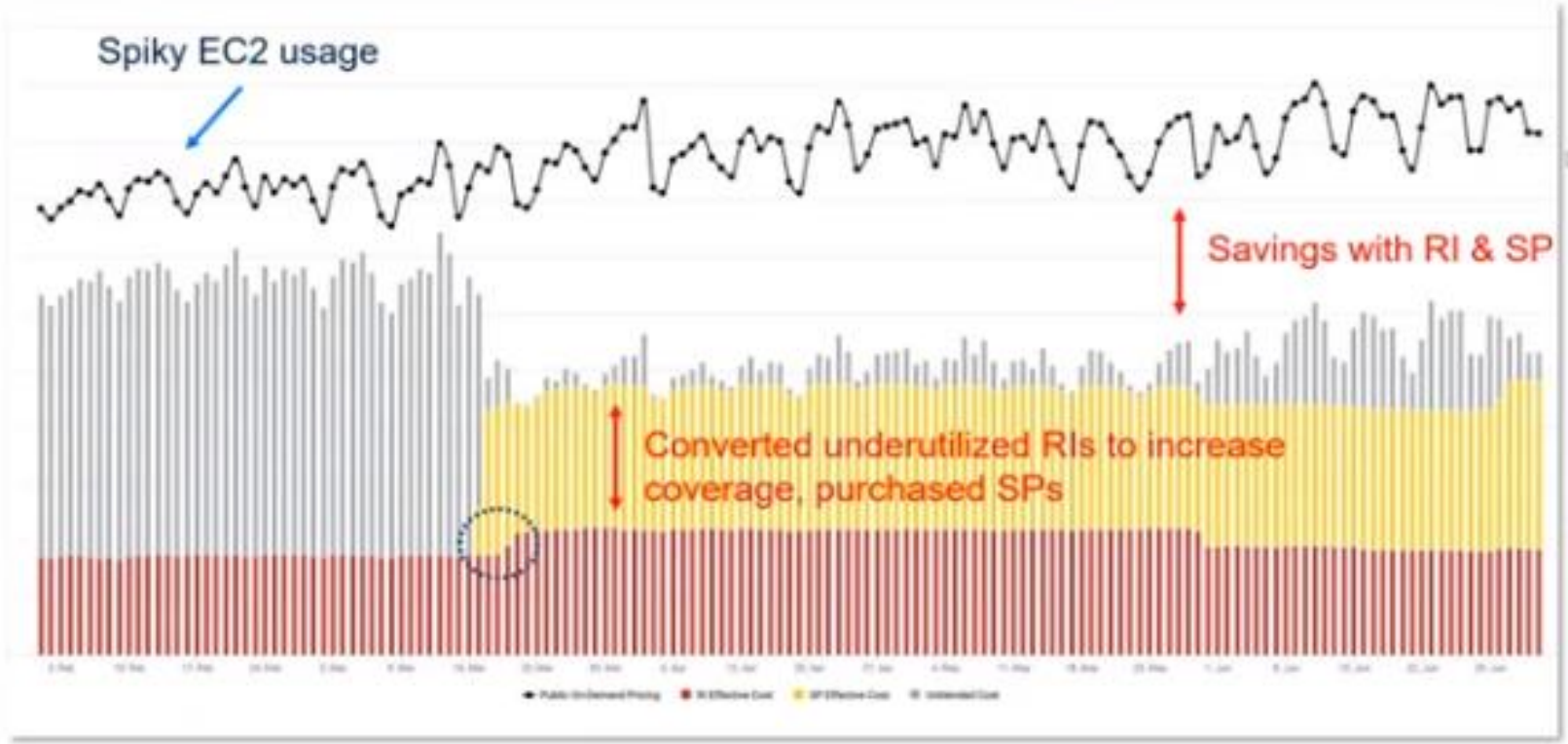
Key Considerations when purchasing Savings Plans

- Centralized Purchasing vs. "Departmental" based decision making
- Are your instances and Auto Scaling Groups optimized / right-sized?
  - Purchase Savings Plans based on estimated "Right-sized" cost, not current state
- Do you have underutilized Convertible Reserved Instances?
  - If you have RIs that are convertible but are not fully utilized, you should convert to instances that provide better coverage
  - Purchase Savings Plans based on fully utilized RIs
- Cloud provider / bill reader recommendations – Beware!
  - "Buy this much" but are they optimized?
  - "Free tools / assessments" limitations

Example 1 – Increasing Savings Plan Coverage



Example 2 – Convertible Ris with Savings Plans



Optimizing Purchases with Densify

- Constantly leverage the right RI portfolio
- Minimize true up costs
- Manage out of existing RI's where optimal
- Savings plan selection and optimal commits

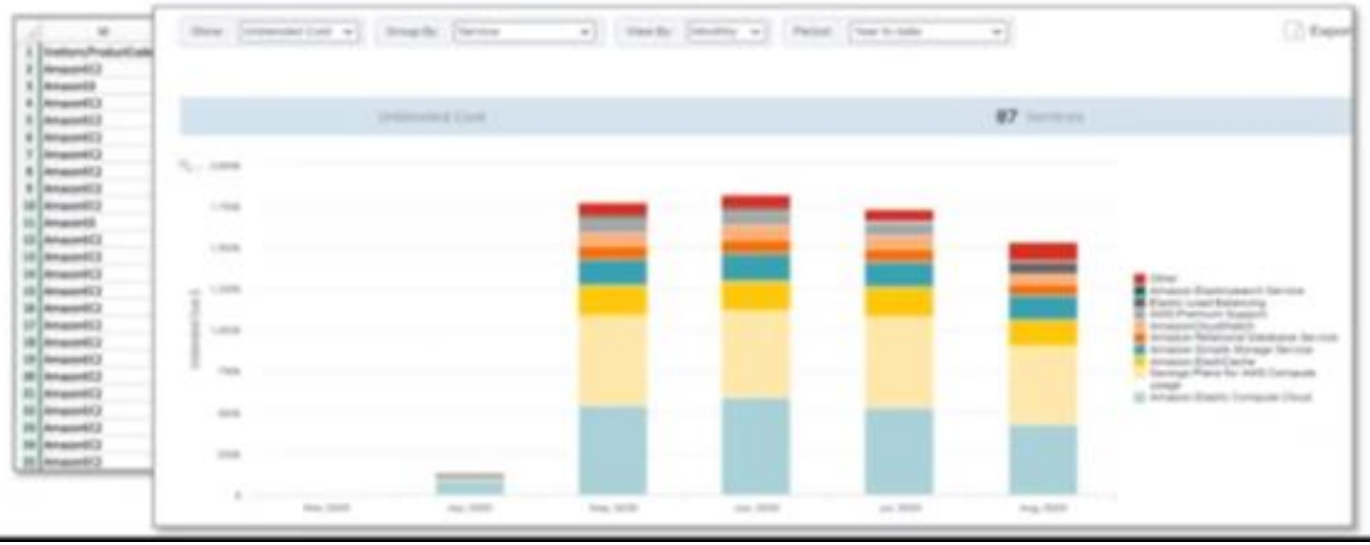
- Full monitoring of your RI Utilization & Coverage to maximize efficiency
- RI Aware Recommendations
  - Factor in existing standard RIs to ensure they are fully utilized
- Instance Optimization
  - Know your potential monthly savings through predictive rightsizing before oversubscribing
- Granular & fully transparent Convertible RI exchange process to minimize True Up Costs
- Savings Plans recommendations that are aware of your existing Reserved Instances
  - Exchange idle Convertible RIs before oversubscribing Savings Plans
- Managed Service
  - Optimization & Financial Visibility analysis from Densify Cloud Experts

Agenda

- Examining your AWS Cost and Usage Reports
  - Hidden costs in your CUR files affects your bottom line without you even knowing
- Top 10 Strategies to Controlling your Cloud Costs
- New Cloud Operating Model & Need for Cost Governance (FinOps)
  - What is FinOps?
  - Building a culture of cost awareness across your organization

Examining your AWS Cost and Usage Report

- AWS provides detailed list of items that are attributing to your Costs
  - *LineItems, BillingPeriod, Costs/Rate, Amortized Costs, ProductCode* etc..
  - Too overwhelming and free tools may not be enough
- Densify – Cloud Cost Intelligence makes it simpler to read your Cloud Bill
  - Understand what is making up the "cost" of your service
  - Cloud Experts provide expert guidance



1. Select the correct cloud instance type

- Typically EC2 represents a significant portion of cloud bills
- Wide selection of instance families & types targeting different workloads
  - Compute, Memory, General purpose, High Performance Computing, Storage Optimized
  - Chipsets: Intel or AMD or Gravitron

- Example: OnDemand Monthly Prices listed for various instance types in US-East (Ohio) for Linux AMIs

t3.large	m5.large	c5.large	r5.large	i3.large	z1d.large
$60.74	$70.08	$62.05	$91.98	$113.88	$135.78

- Procurement Options: OnDemand, Reserved Instances or Savings Plans

Densify – Continuous Compute Optimization

- Hidden items that you are getting charged for: Oversized EC2 instances
  - Challenge: Engineers/Developers unaware of ideal resource selection
- Continuous Rightsizing – effort to align workloads to the optimal EC2 instance
  - Identify CPU/Memory/Disk/Network performance & match them to best EC2 instance
  - Do not compromise performance for cost savings, via Policy

2 & 3. Selecting lower cost AMI and chip sets

- Hidden items that you are getting charged for:
  - Utilize Free/Open Source AMIs where applicable to lower compute costs
    - Example: *c5.xlarge instance in US-East Ohio (OnDemand)*

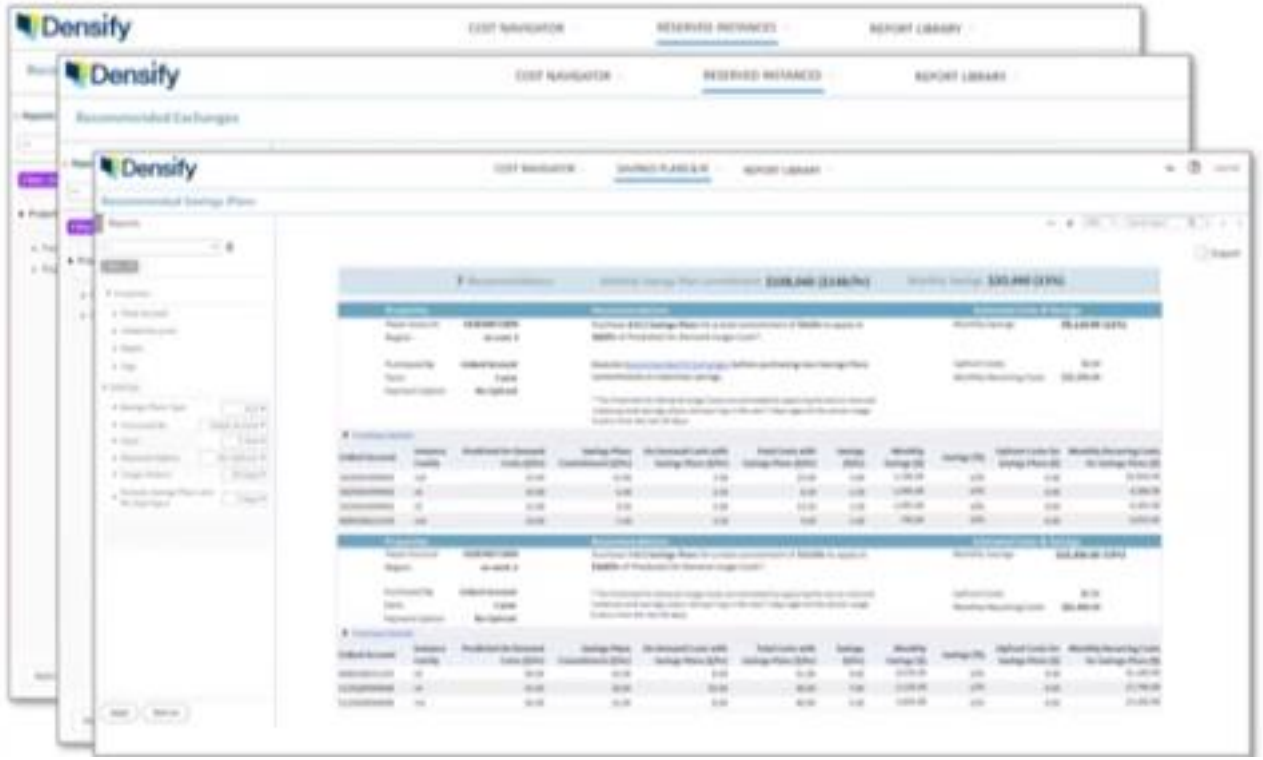
Linux	Windows	RHEL	SUSE	Windows SQL Standard	Linux SQL Server Standard	Windows SQL Server Enterprise
$124.10	$258.42	$167.90	$197.10	$608.82	$474.50	$1,353.42

  - Switch to AMD instance types, if you haven't done so already
    - Consider geographic region
    - Example: *r5.2xlarge instance with Linux OS (OnDemand)*

	US-East1 (Ohio)	US-West (N. California)	AP (Mumbai)	AP (Tokyo)	EU (London)	EU (Stockholm)	ME (Bahrain)	South America (Sau Paulo)
r5.2xlarge	$367.92	$408.80	$379.60	$443.84	$432.16	$391.28	$452.60	$586.92
r5a.2xlarge	$329.96	$367.92	$208.78	$400.04	$388.36	n/a	n/a	$528.52

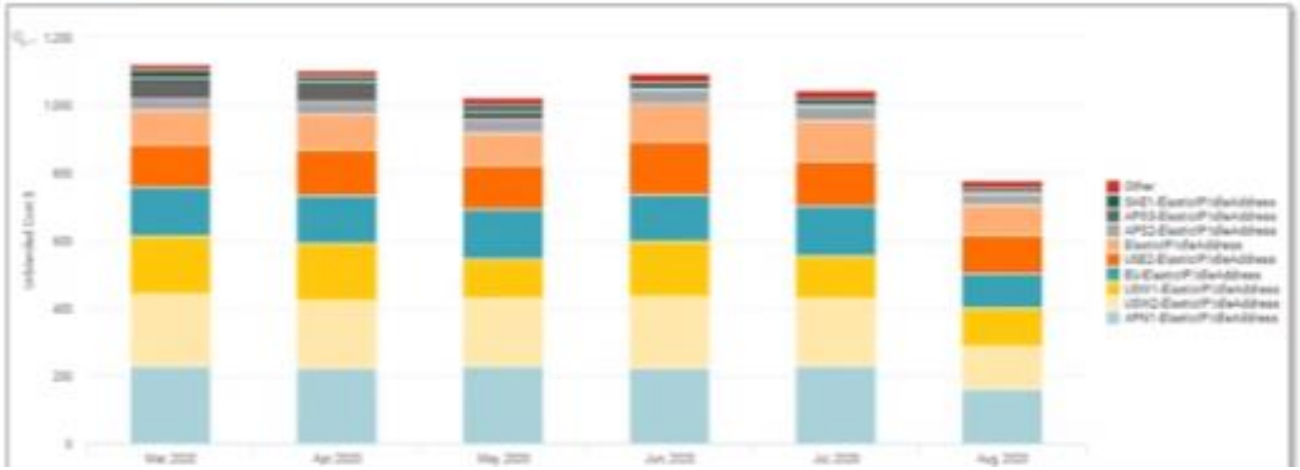
4. Make better RIs & Savings Plans commitments

- Understand your current RI/SP utilization & coverage
- Hidden Costs:
  - True up costs for Convertible RIs
  - Manage out of existing RI's where optimal
  - Savings plan selection and optimal commits



5. Release idle Elastic IP addresses

- Elastic IP address is a static IPv4 address associated with an AWS Account
  - Attached to a Network interface
- Hidden items that you are getting charged for:
  - Idle Elastic IP addresses
    - AWS imposes a small hourly charge if an Elastic IP address is not associated with a running instance



6. Identify Cost Anomalies & Outliers

- Closely inspect your AWS Cost and Usage Reports regularly to identify any spend anomalies (Outliers)
  - For example: Cloudwatch Log Delivery

7. Select optimal S3 storage

- Factors attributing to S3 costs:
  - Storage costs varies by Class
  - Data Access/APIs costs varies by Class
  - Data Transfer costs varies by Region
- Use LifeCycle Manager to manage objects storage based on customizable policies

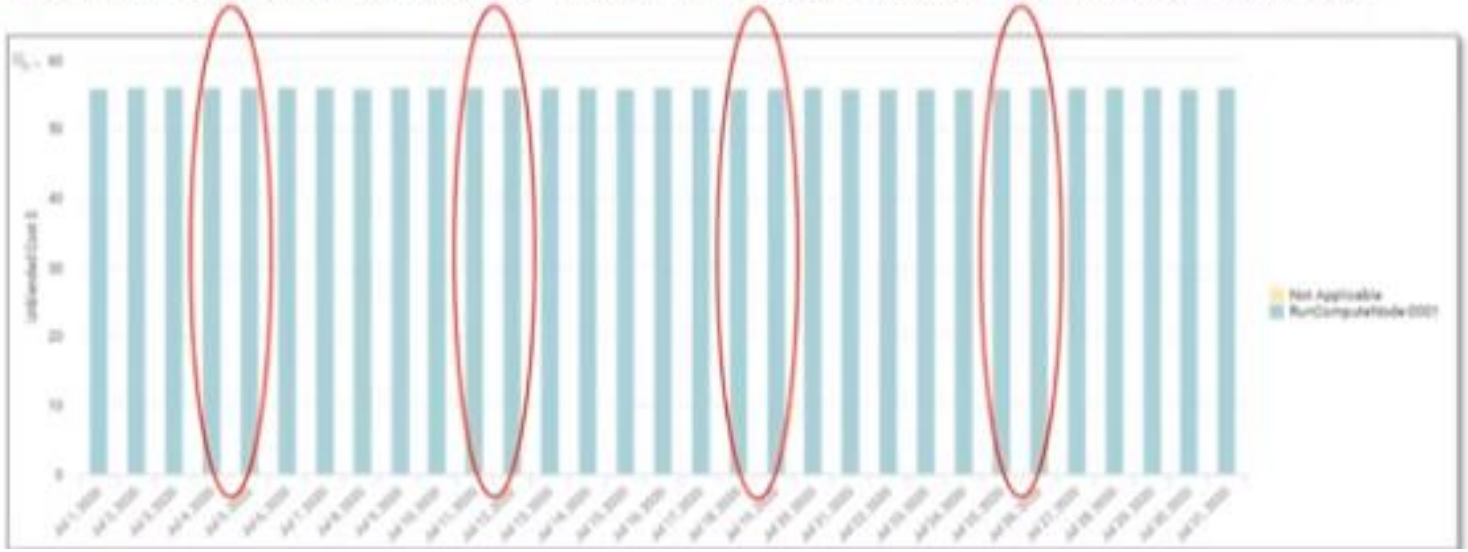
Strategies for S3

- Hidden items that you are getting charged for:
  - Incorrect Storage Class Selection
  - Data API costs
  - Data Transfer
- Example: 500TB of storage in N. Virginia
  - 250TB of that is infrequent access*
  - 50M PUT, 100M GET requests
  - 2000 TB Data Transfer to Internet

	S3 standard	S3 Intelligent Tiering	S3 Standard – IA	S3 One Zone – IA	S3 Glacier	S3 – Glacier Archive
orage cost (FA)	$11,050	$11,050	$6,250	$5,000	$2,000	$495
orage cost (IA)*	$11,050	$8,675	$6,250	$5,000	$2,000	$495
ta API costs (N.Virginia)	$290	$290	$600	$600	$2,540	$2,540
ta API costs (S.Paulo)	*$406*	*$406*	*$600*	*$600*	*$3,556*	*$5,056*
ta Transfer (Internet)	$92,500	$92,500	$92,500	$92,500	$92,500	$92,500
ta Transfer (S.Paulo)	*$40,000*	*$40,000*	*$40,000*	*$40,000*	*$40,000*	*$40,000*

9. Pause idle RedShift Clusters

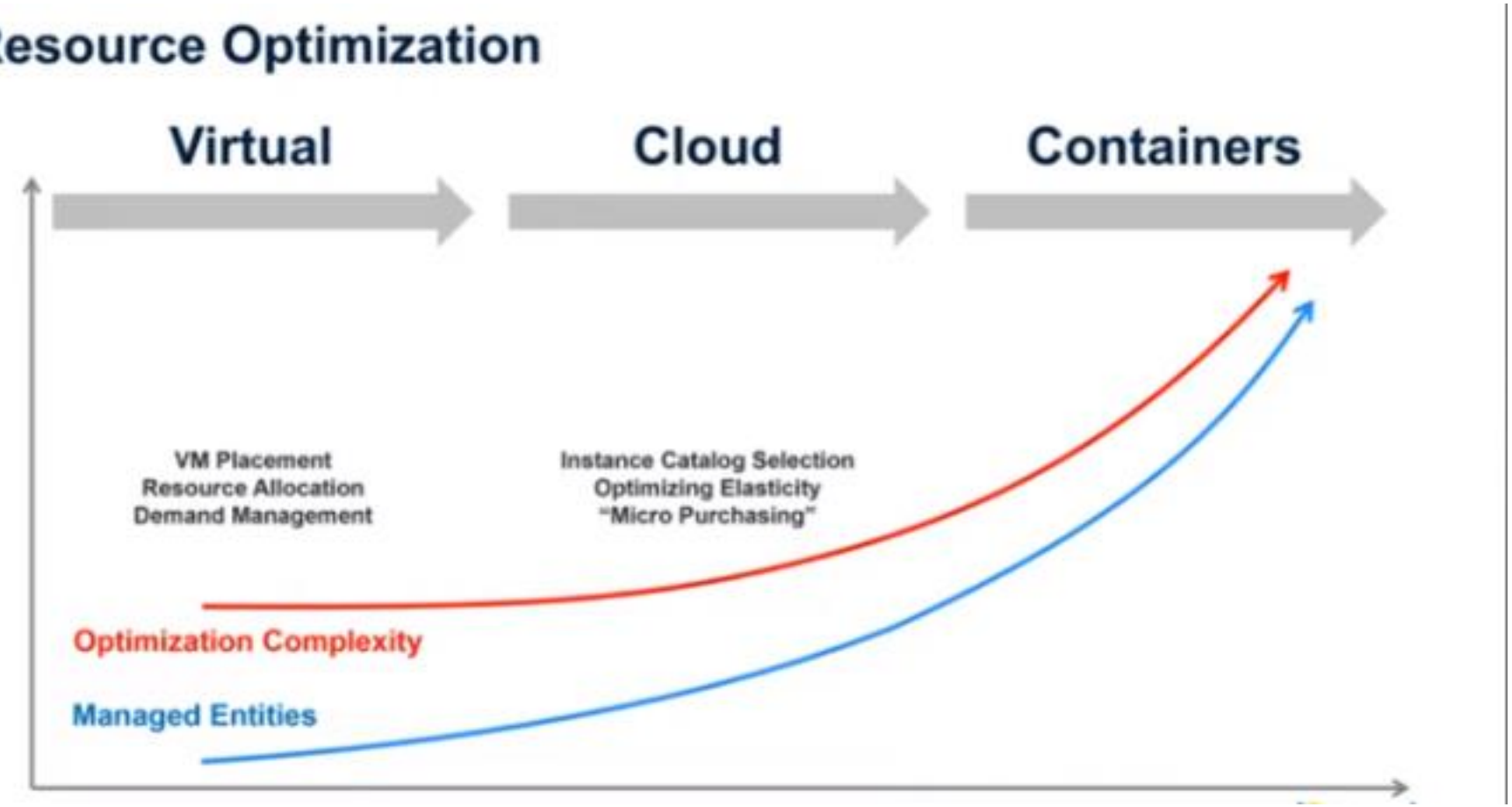
- Redshift Clusters consist of specialized Compute Nodes used for Massively Parallel Processing
- Hidden items that you are getting charged for:
  - Running Redshift cluster nodes during offhours (i.e. weekends)
  - Utilize new "Pause & Resume" feature on Redshift Cluster nodes when not in use



10. Automate Infrastructure selection & tagging

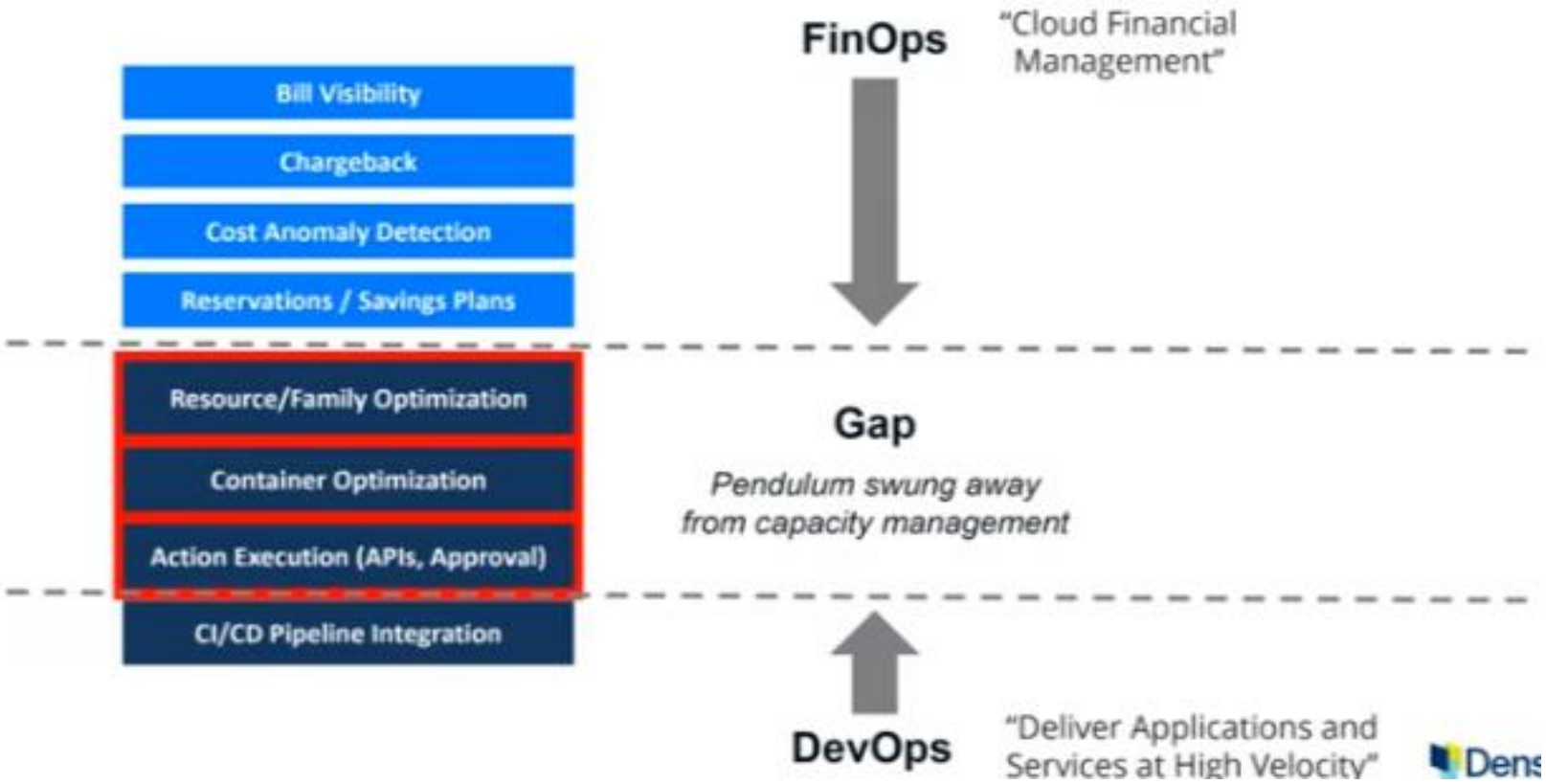
- Automate rightsizing as part of infrastructure provisioning
  - Integrate into the CI/CD framework
- Automate tagging best enforcement

Resource Optimization

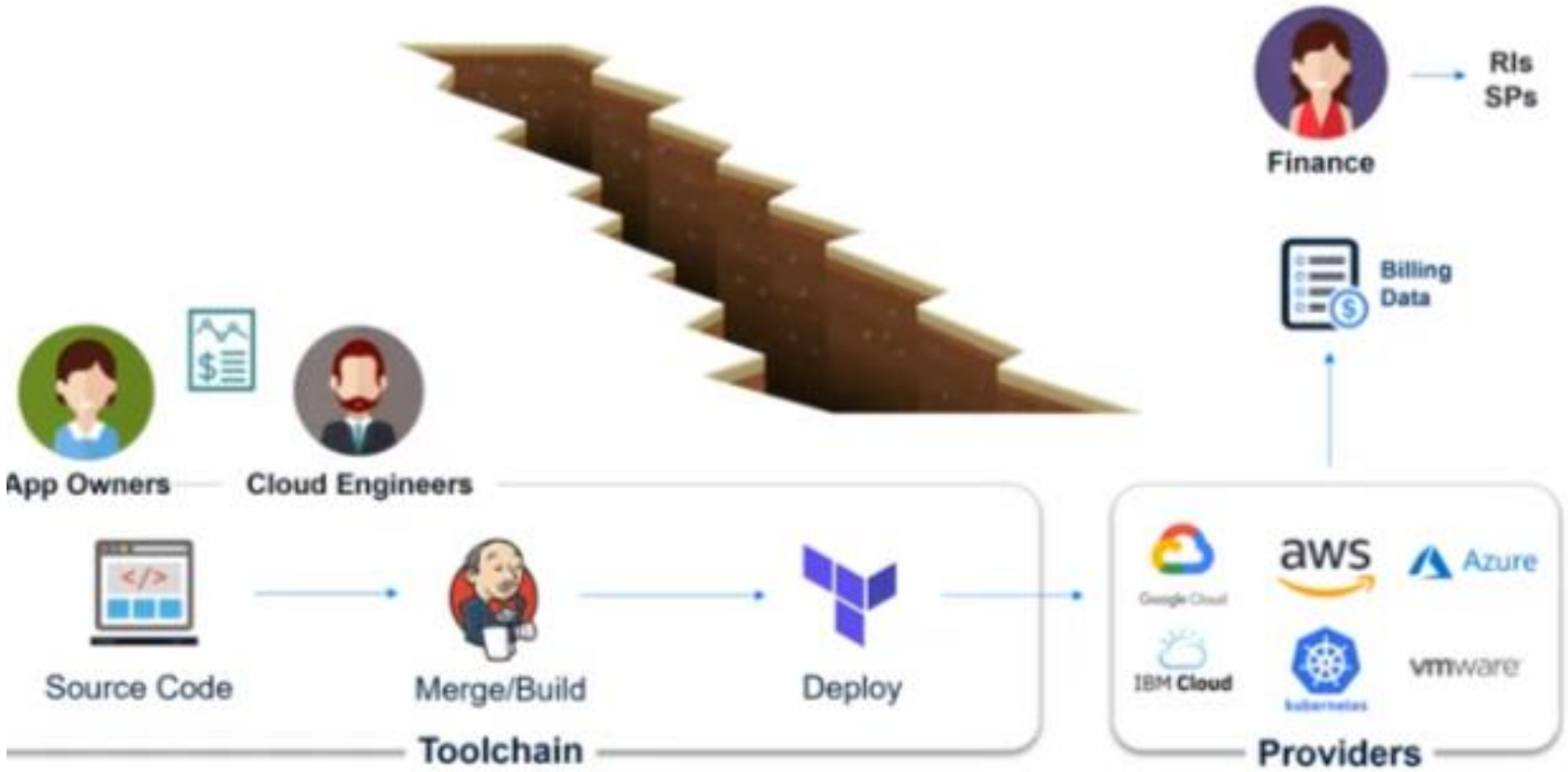


Cloud & Container Resource Optimization

Cloud & Container Resource Optimization



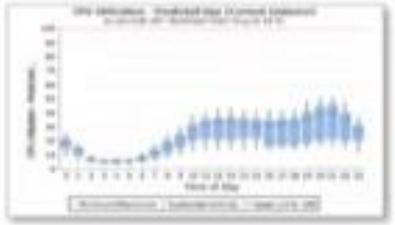
What this Looks Like in Practice



Why Don't Engineers take Action?

- The real question: why don't engineers take *these* actions
- In order to act on a recommendation there are several things that are required:

Precision



Actions need to be correct

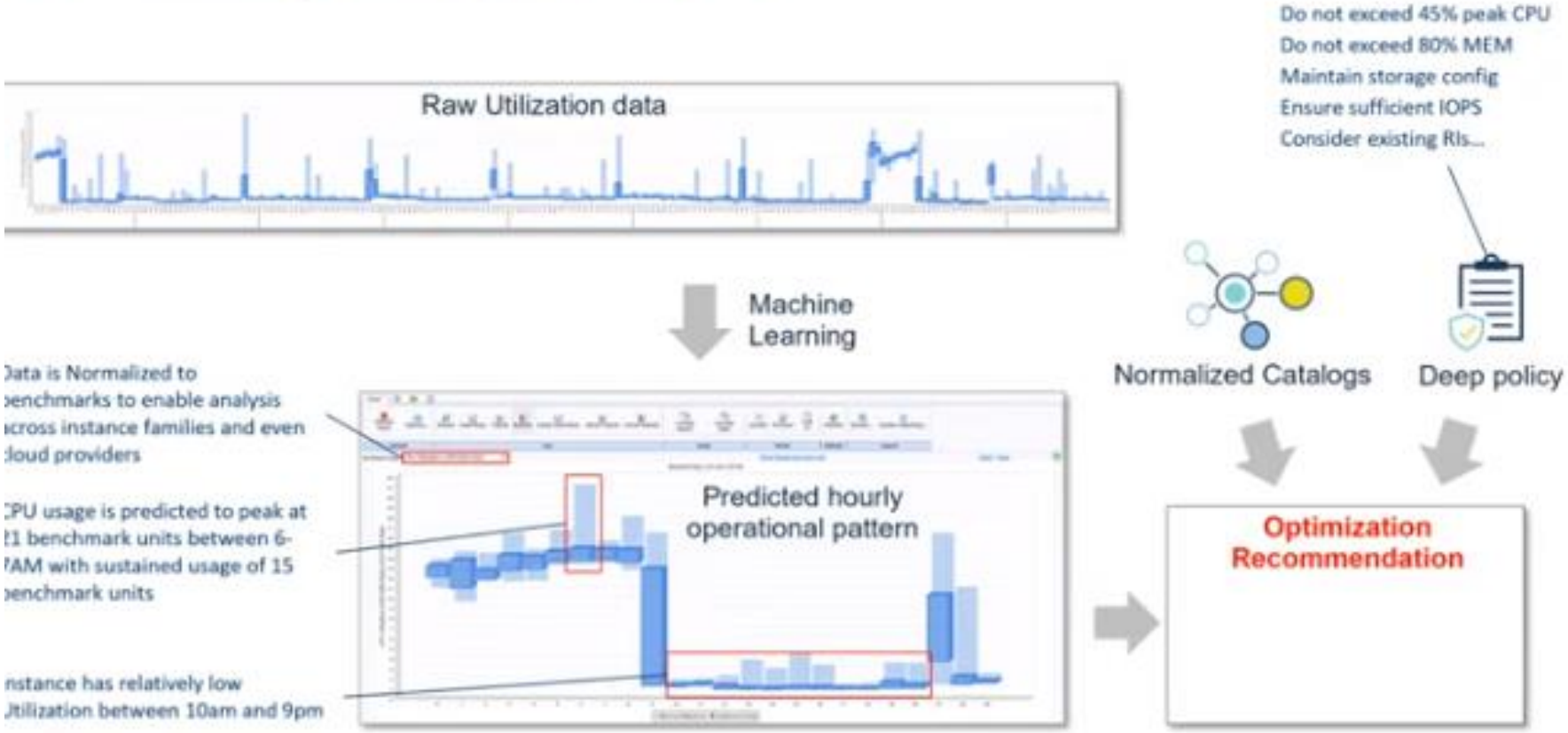
Transparency

Stakeholders need to be able to understand and approve them

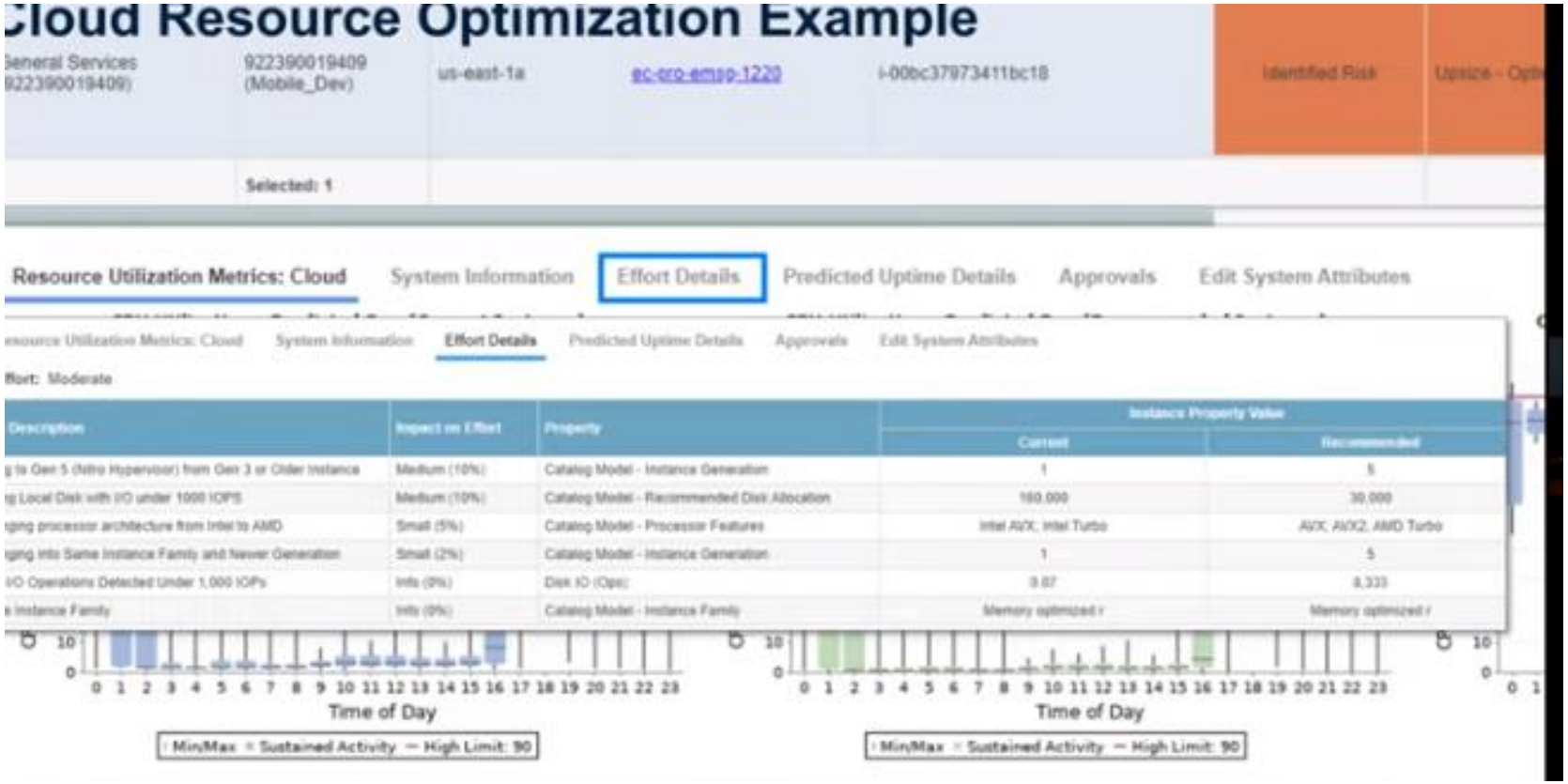
Integration

They need to go to the right tools and pipelines

Optimizing Cloud Resources



Precision = Correct & Actionable


Cloud Resource Optimization Example
General Services
922390019409)
922390019409
(Mobile_Dev)
us-east-1a
ec-ero-emgp-1220
i-00bc37973411bc18
Identified Risk
Selected: 1
Resource Utilization Metrics: Cloud
System Information
Effort Details
Predicted Uptime Details
Approvals
Edit System Attributes
Effort: Moderate
Description
Impact on Effort
Property
Instance Property Value
Current
Recommended
Catalog Model - Instance Generation
Catalog Model - Recommended Disk Allocation
Catalog Model - Processor Features
Catalog Model - Instance Generation
Disk IO (Ops)
Catalog Model - Instance Family
Time of Day
Min/Max
Sustained Activity
High Limit: 90

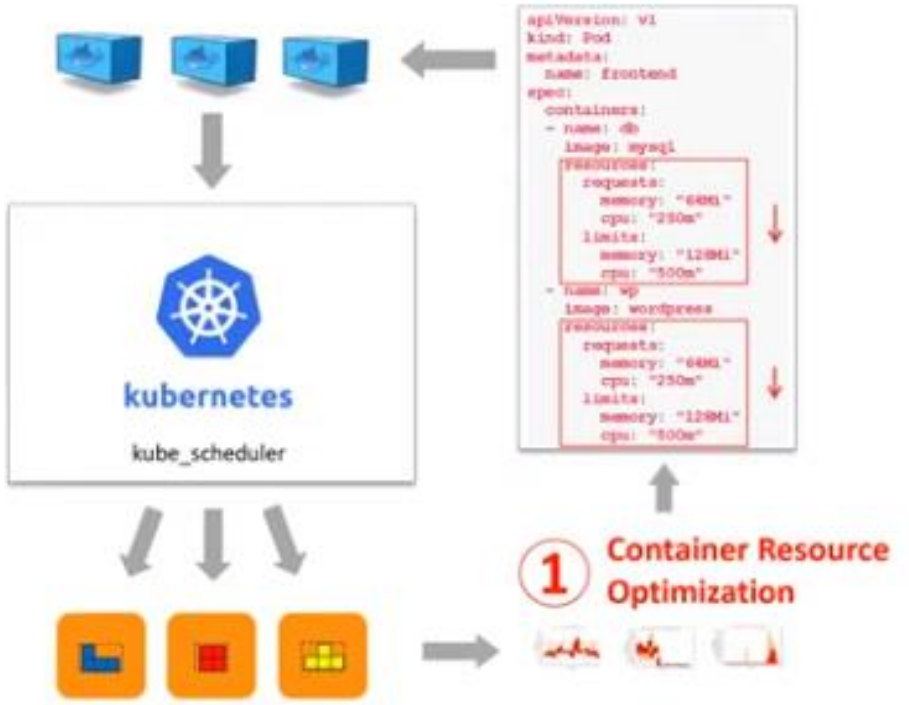
Communicating with App Owners – Plainly & Clearly

Automating via Infrastructure as Code


```
provider "aws" {
  region = "${var.aws_region}"
}

resource "aws_instance" "web" {
    name = "Web Server"

    instance_type = "m4.large"

    ami = "${lookup(var.aws_amis, var.aws_region)}"
}
```

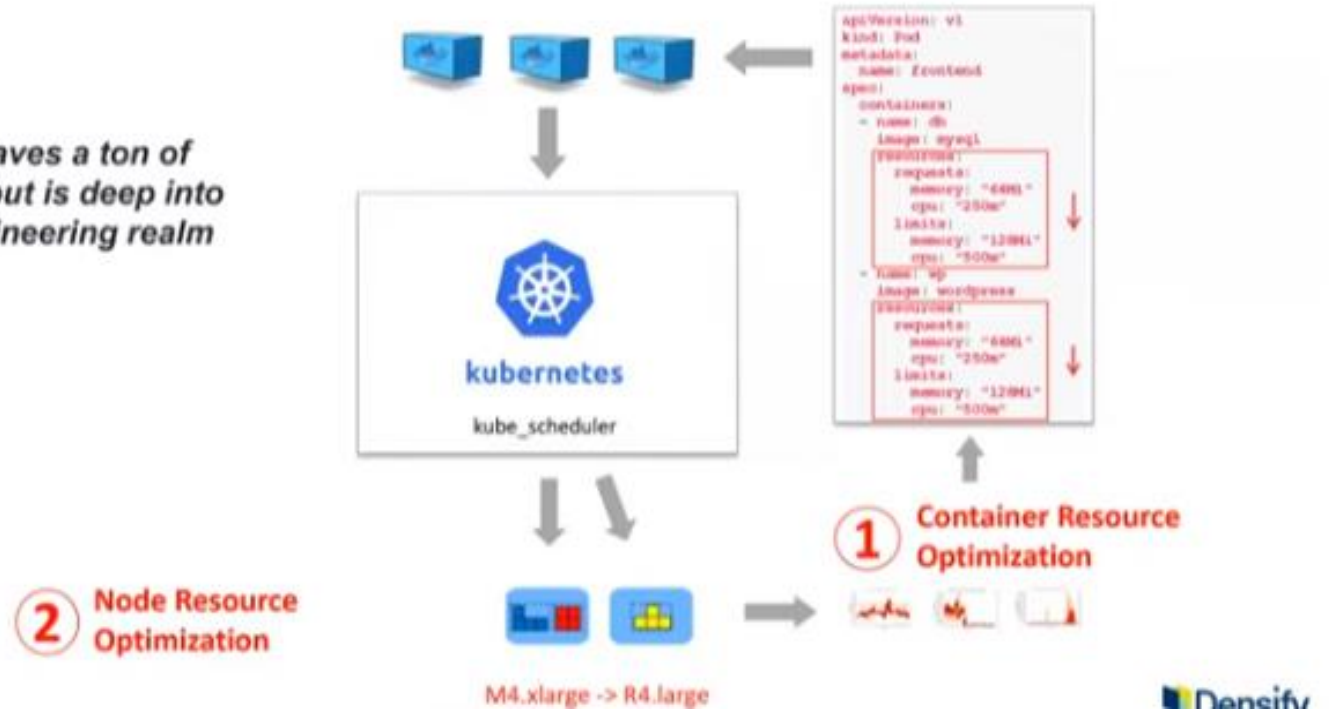
- Hard-Coded
- Rough Estimate
- Huge problem in scale

Container Optimization – Multi-Step Process



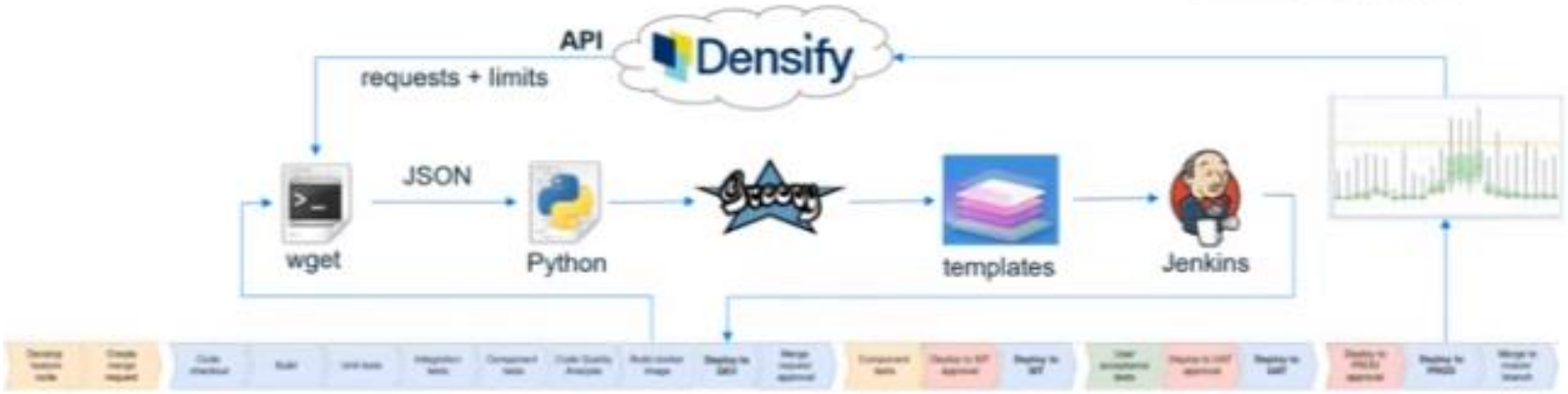
Container Optimization – Multi-Step Process

This saves a ton of money, but is deep into the Engineering realm



Container Automation Example

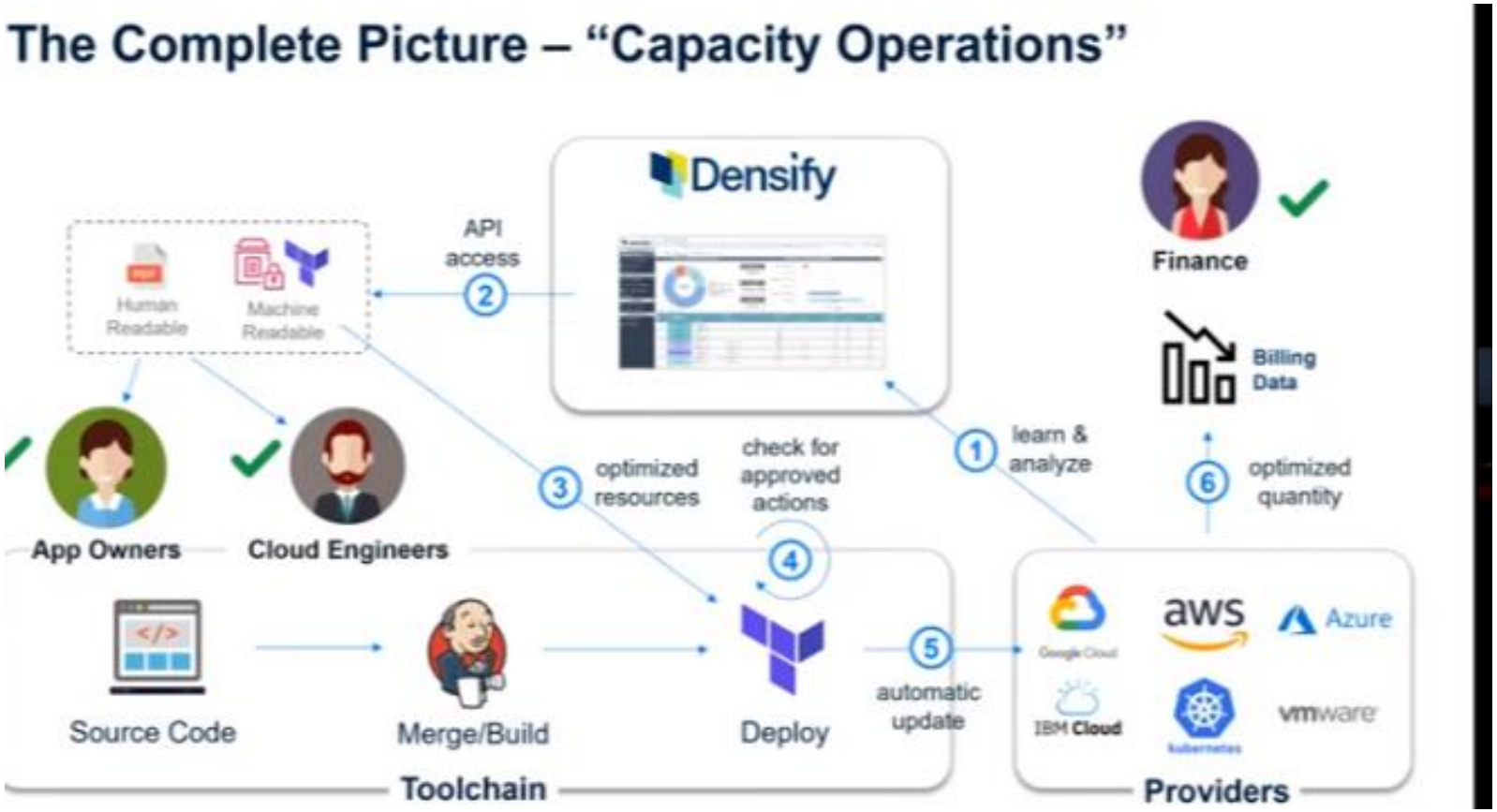
Raiffeisen BANK



1. Shell script uses wget to call the Densify Recommendation API
2. Densify returns optimized container request and limit settings in JSON format
3. JSON is parsed Python
4. Groovy used to inject new values into container template
5. Jenkins pipeline redeploys container with optimized settings

https://www.densify.com/resources/optimizing-openshift-resources

The Complete Picture – “Capacity Operations”



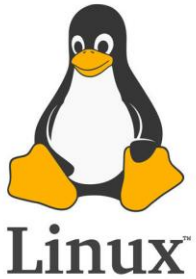

CKA – DAY 2


K.I.S.S.
Keep It Simple, Stupid!

TECH STACK LEARNINGS

- Vagrant
- Virtualbox
- Ubuntu OS
- BaseImages
- Docker /Docker Hub
- Application dependency
- OS package manager
- Linux kernel
- NoSQL
- Building/pushing/Running container's
- Servers/VM
- PAAS
- 12factor apps – cloud native
- Much more …..

FSF , GNU/Linux , Linux Foundation



- https://www.fsf.org/
- https://www.kernel.org/
- https://kernelnewbies.org/
- https://lwn.net/ - linux kernel news
- https://www.linuxfoundation.org/