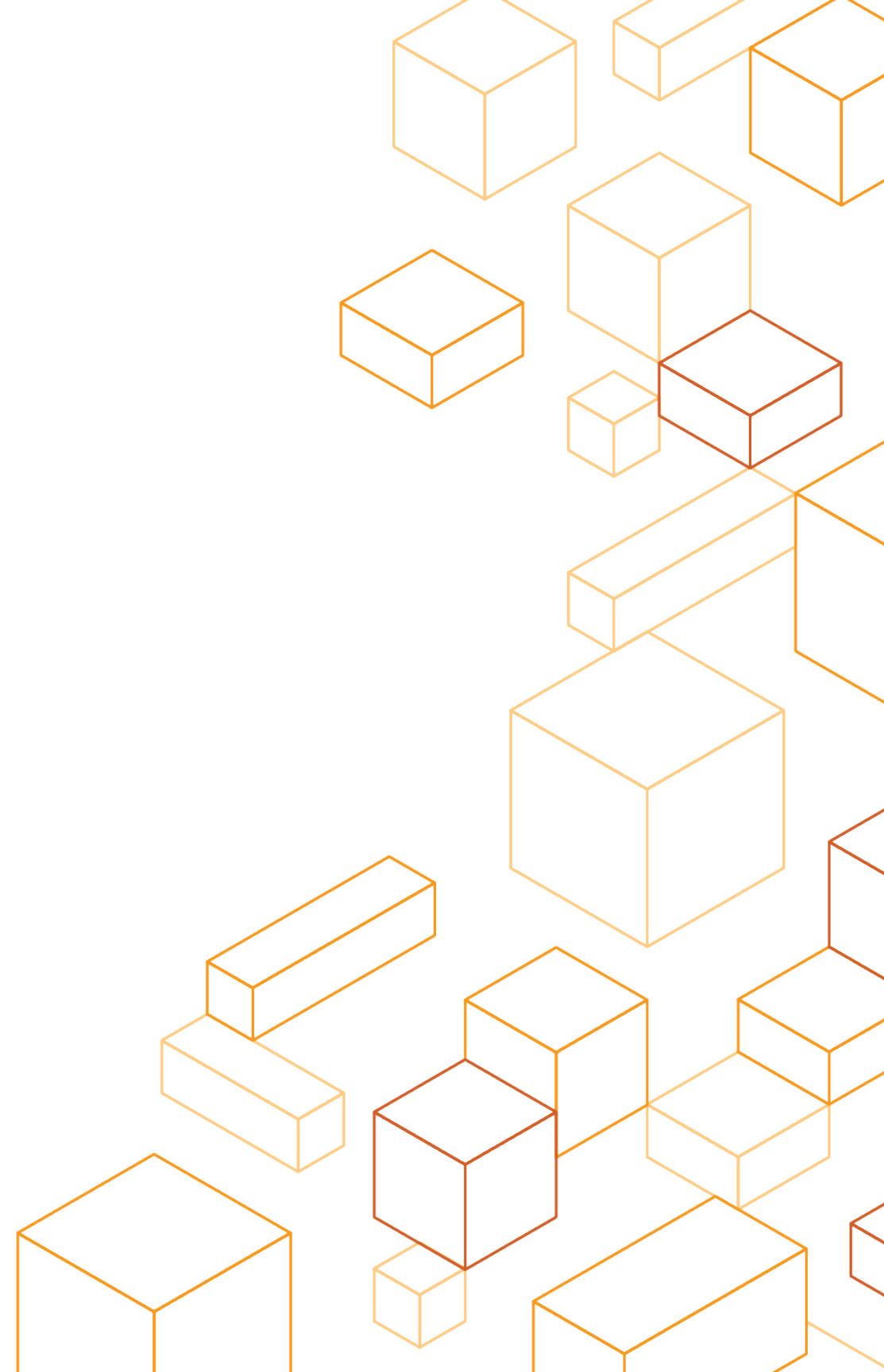




# Six ways to reduce your AWS bill

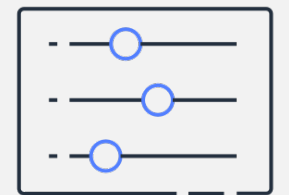
AWS Cloud Economics



# We hear you

- We've talked with startup customers like you around the globe
- This is not business as usual
- Usage patterns are changing

What you spend on AWS should always be optimized.

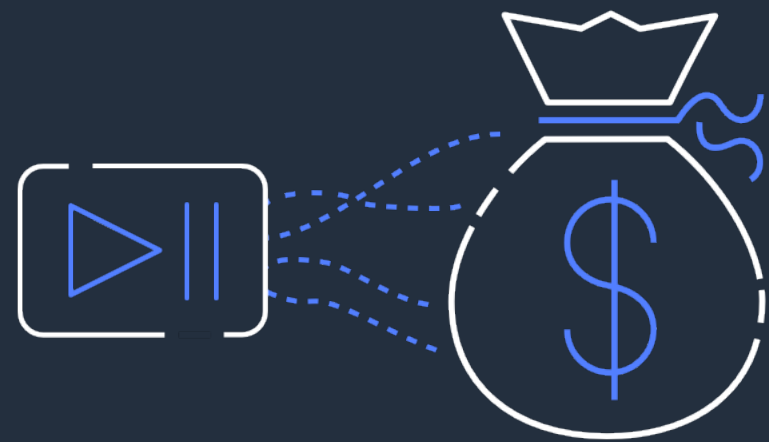


Today's focus

Tools and approaches  
that startups can use  
to optimize AWS costs



# Before we start...



# 6 the six ways

1

Enable S3  
Intelligent-  
Tiering

2

Stop paying  
for idle EC2  
and RDS  
instances

3

Choose  
Amazon EC2  
Spot for  
containers

4

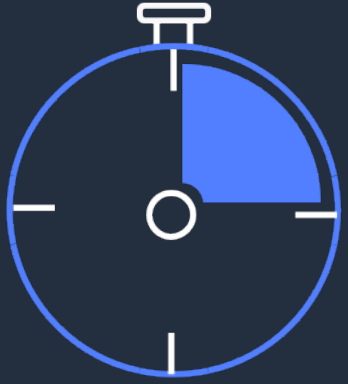
Use AWS  
Compute  
Savings Plans

5

Use AWS  
Reserved  
Instances

6

Cleanup  
underutilized  
resources



Implementation time: *Minutes*

# Enable S3 Intelligent-Tiering

# Scenario

- You are using S3 standard storage class
- You might be paying for S3 storage you don't use



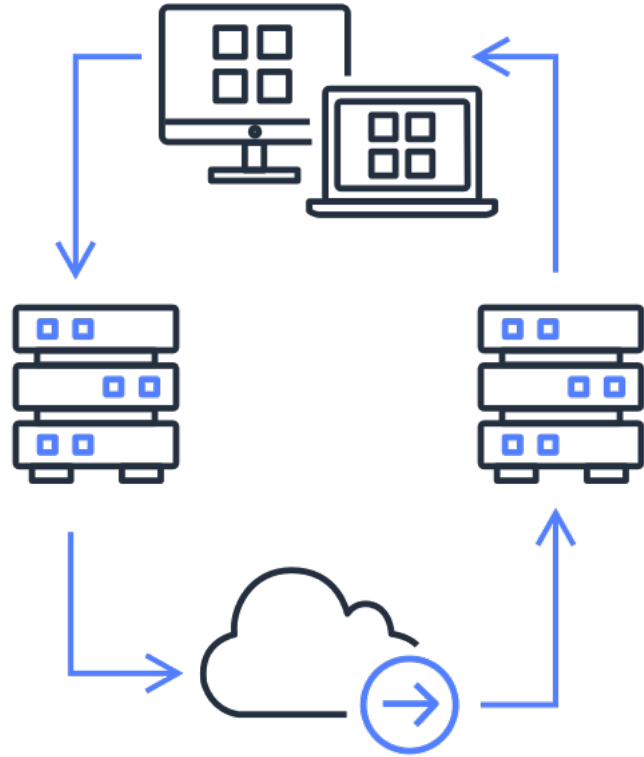
Solution: S3 Intelligent-Tiering

# Enable S3 Intelligent-Tiering for infrequently accessed objects

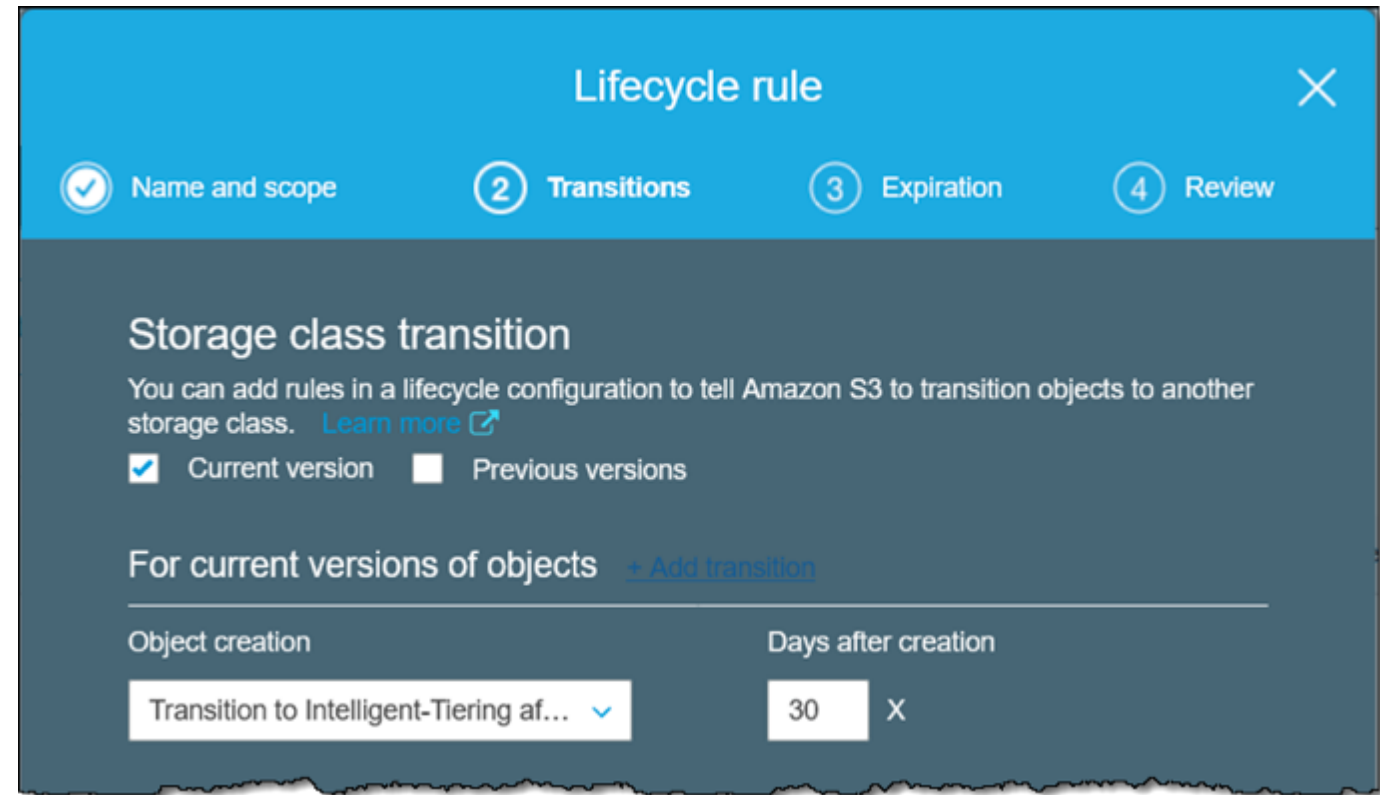
Implementation time	Savings potential	Time to realize savings	Commitment required
Minutes	20% – 30% (for S3 Standard objects transitioned to S3 Intelligent-Tier)	30 days	None



# Get started



Upload objects directly into  
S3 Intelligent-Tier (API)



Create Lifecycle Rules that make  
use of Intelligent-Tiering (UI)

<https://aws.amazon.com/blogs/aws/new-automatic-cost-optimization-for-amazon-s3-via-intelligent-tiering/>



Implementation time: *Minutes to hours*

# Stop paying for idle EC2 and RDS instances

# Scenario

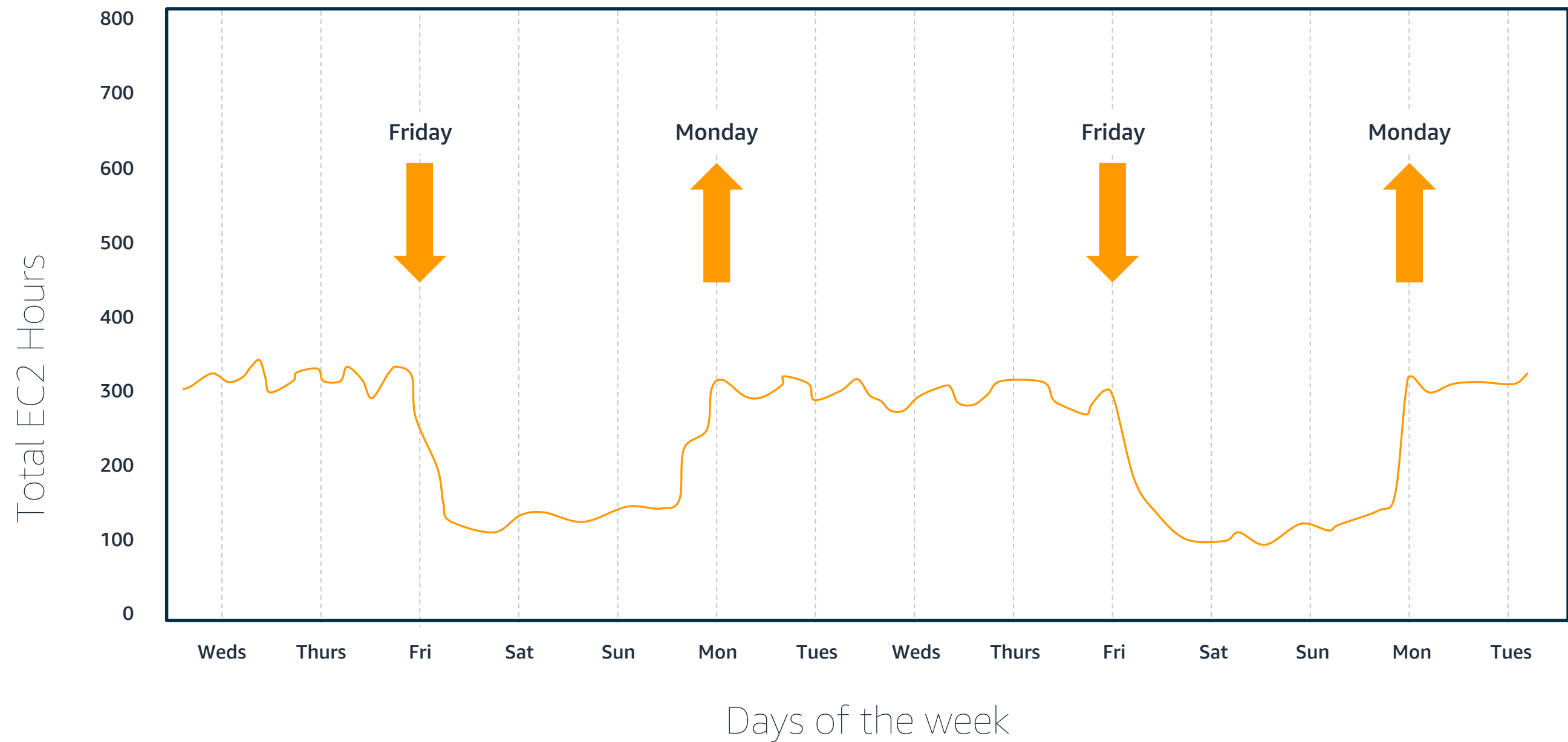
- You leave instances running during evenings, weekends, and holidays
- You might be paying for EC2 and RDS instances even when they are idle



Solution: AWS Instance Scheduler

# Pay for what you need

EC2 Hours vs Time



# Schedule EC2 and RDS instances in non-production environments

Implementation time	Savings potential	Time to realize savings	Commitment required
Minutes to hours	Reduce On-Demand costs by up to 35%*	Minutes	None

*\* Instance scheduling starts on Friday at 6pm and ends Monday at 6am*

# Get started

- 1 Install the AWS Instance Scheduler
- 2 Create schedule based on business requirements
- 3 Tag non-production EC2 and RDS instances to be scheduled

<https://aws.amazon.com/solutions/instance-scheduler/>

## AWS Instance Scheduler

Version 1.3.1

Last updated: 03/2020

Author: AWS

Estimated deployment time: 5 min

[Source code](#)

[CloudFormation template](#)

[View deployment guide](#)

[Launch solution in the AWS Console](#)

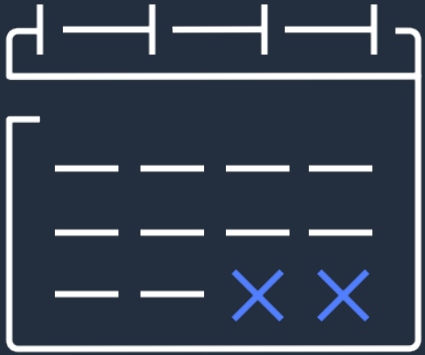
[Deploy with an AWS IQ expert](#)

## Deployment resources

[Download deployment guide](#)

[AWS Solution resources »](#)

[Contact us »](#)



Implementation time: *Hours/days to weeks*

# Choose Amazon EC2 Spot for containers

# Scenario

- You are running containerized workloads on EC2, or using managed services such as ECS, EKS, and Fargate
- You are paying the default On-Demand pricing



Solution: Amazon EC2 Spot



# Choose Spot for containerized workloads that are stateless, fault-tolerant, and loosely-coupled

Implementation time	Savings potential	Time to realize savings	Commitment required
Hours/days to weeks	Up to 90% cheaper than On-Demand	Hours/days to weeks	None

# Getting started

## Self-service container references

1

### ECS on Spot

<https://aws.amazon.com/ec2/spot/containers-for-less/get-started/>

2

### EKS on Spot

<https://aws.amazon.com/blogs/compute/run-your-kubernetes-workloads-on-amazon-ec2-spot-instances-with-amazon-eks/>

3

### Fargate on Spot

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/fargate-capacity-providers.html>

4

### AWS Spot workshops for other workloads

<https://ec2spotworkshops.com/>



Implementation time: *Hours*

# Use AWS Compute Savings Plans

# Scenario

- You have EC2 or Fargate workloads that are always on
- You are leveraging Lambda in your architecture
- You are paying the default On-Demand pricing



Solution: AWS Compute Savings Plans

# AWS Compute Savings Plans

Provides the most flexibility across...

- **Instance family:** e.g. Move from C5 to M5
- **Region:** e.g. change from EU (Ireland) to EU (London)
- **OS:** e.g. Windows to Linux
- **Tenancy:** e.g. switch Dedicated tenancy to Default tenancy
- **Compute options:** e.g. move from EC2 to Fargate or Lambda



# Choose 1 year, No Upfront Compute Savings Plans

Implementation time	Savings potential	Time to realize savings	Commitment required
Hours	Up to 54% (EC2), 20% (Fargate), 12% (Lambda) cheaper than On-Demand	Hours	1 year, No Upfront costs

# Get started

## AWS Cost Explorer will provide you with Savings Plans recommendations

- 1 Review your Savings Plans recommendations in the AWS Cost Explorer
- 2 Customize recommendations based on your needs (Term Length: 1 Year, Payment Option: No Upfront)
- 3 Add preferred Savings Plans amount to cart and purchase

Savings Plans > Purchase Recommendations

Recommendation options

Savings Plans type: ☒ Compute, ☐ EC2 Instance

Savings Plans term: ☒ 1-year, ☐ 3-year

Payment option: ☒ All upfront, ☐ Partial upfront, ☐ No upfront

Based on the past: ☐ 7 days, ☒ 30 days, ☐ 60 days

Recommendation: Purchase a Compute Savings Plan at a commitment of \$0.30/hour

You could save an estimated \$48 monthly by purchasing the recommended Compute Savings Plan.

Based on your past 30 days of usage, we recommend purchasing a Savings Plan with a commitment of \$0.30/hour for a 1-year term. With this commitment, we project that you could save an average of \$0.07/hour - representing a 14% savings compared to On-Demand. To account for variable usage patterns, this recommendation maximizes your savings by leaving an average \$0.09/hour of On-Demand spend.

Before recommended purchase	After recommended purchase (based on your past 30 days of usage)	
Monthly On-Demand spend ⓘ <b>\$329</b> (\$0.45/hour) <small>Your estimated On-Demand spend based on your usage over the past 30 days (including all active Savings Plans)</small>	Estimated monthly spend ⓘ <b>\$281</b> (\$0.39/hour) <small>Your recommended \$0.30/hour Savings Plans commitment + an average \$0.09/hour of On-Demand spend</small>	Estimated monthly savings ⓘ <b>\$48</b> (\$0.07/hour) <small>14% monthly savings over On-Demand \$329 - \$281 = \$48</small>

This recommendation examines your usage over the past 30 days (including your existing Savings Plans and EC2 Reserved Instances) and calculates what your costs would have been had you purchased the recommended Savings Plans. See applicable rates for Savings Plans [here](#). To generate this recommendation, AWS simulates your bill for different commitment amounts and recommends the commitment amount that provides the greatest estimated savings. [Learn more](#)

Recommended Compute Savings Plans

Download CSV Add selected Savings Plan(s) to cart

Term	Payment option	Recommended commitment	Estimated hourly savings ⓘ
<input checked="" type="checkbox"/> 1-year	All upfront	\$0.30/hour	\$0.07 (14%)

Spend and minimum hourly spend based on your current on-demand spend for the given instance family.

<https://docs.aws.amazon.com/savingsplans/latest/userguide/get-started.html>



Implementation time: *Hours*

# Use AWS Reserved Instances



# Scenario

- You have RDS, Redshift, ElastiCache and Elasticsearch workloads that are always on
- You are paying the default On-Demand pricing





Solution: AWS Reserved Instances

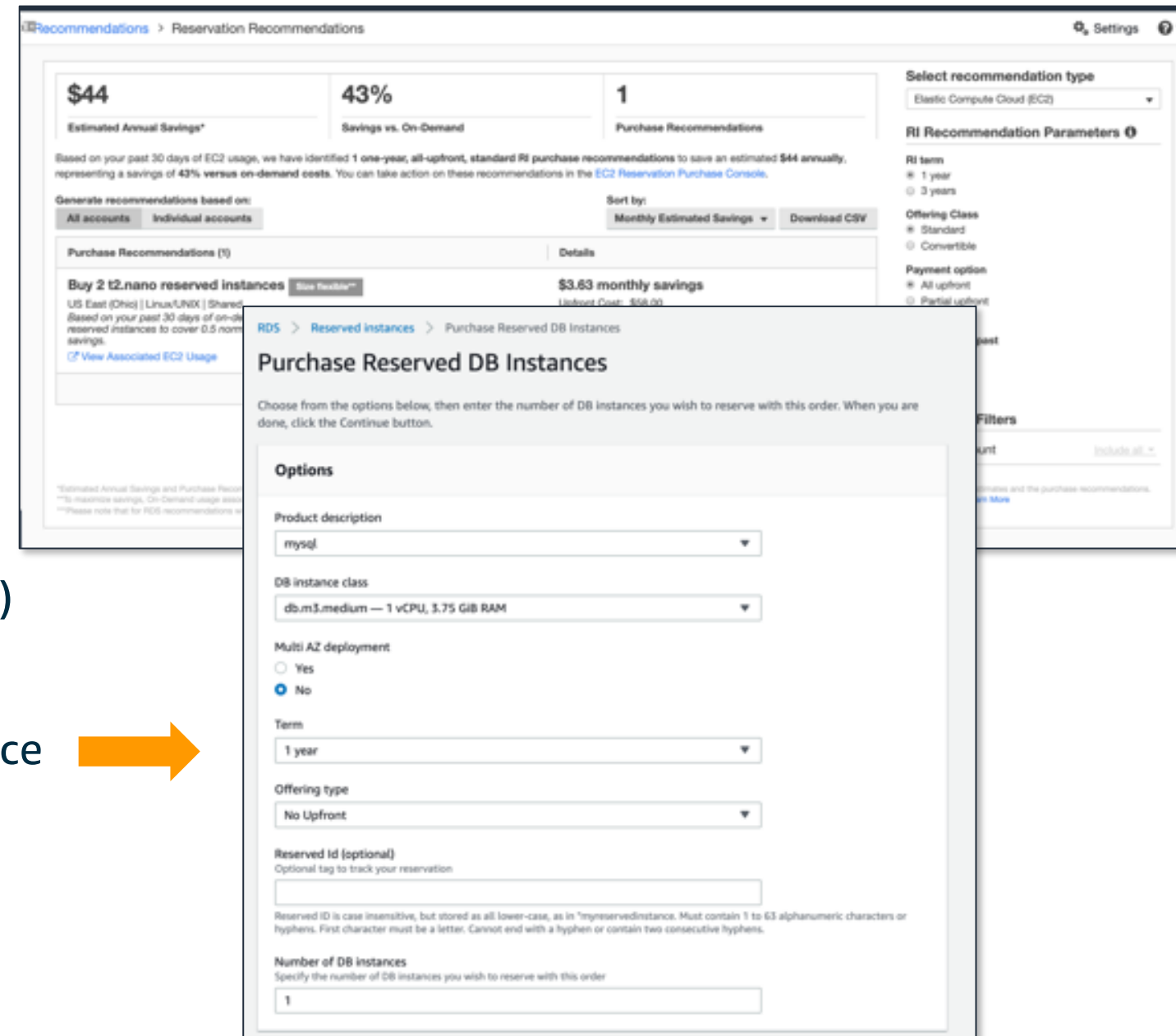
# Choose 1 year, No Upfront Reserved Instances

Implementation time	Savings potential	Time to realize savings	Commitment required
Hours	Up to 42% (RDS), 32% (ElastiCache), 31% (Elasticsearch), 30% (Redshift) cheaper than On-Demand	Hours	1 year, No Upfront costs

# Get started

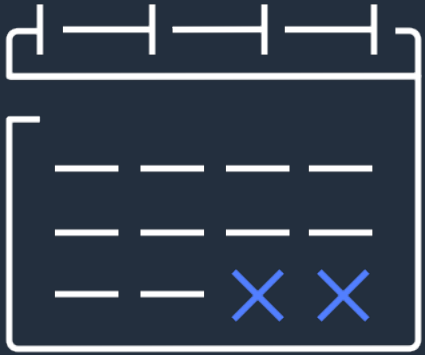
## AWS Cost Explorer will provide you with Reserved Instance recommendations

- 1 Review your Reserved Instance recommendations in the AWS Cost Explorer 
- 2 Customize recommendations based on your needs (Term Length: 1 Year, Payment Option: No Upfront)
- 3 Purchase Reserved Instances from the specific service page in the AWS console 



The image shows two screenshots from the AWS console. The top screenshot is from the 'Recommendations' page, specifically 'Reservation Recommendations'. It displays a summary: \$44 Estimated Annual Savings\*, 43% Savings vs. On-Demand, and 1 Purchase Recommendation. Below this, it lists a recommendation for 'Buy 2 t2.nano reserved instances' with a monthly savings of \$3.63. The bottom screenshot is the 'Purchase Reserved DB Instances' page. It shows a form with the following fields: Product description (mysql), DB instance class (db.m3.medium — 1 vCPU, 3.75 GiB RAM), Multi-AZ deployment (No), Term (1 year), Offering type (No Upfront), Reserved ID (optional), and Number of DB instances (1).

<https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/ri-recommendations.html>



Implementation time: *Hours/days*

# Cleanup underutilized resources

# Scenario

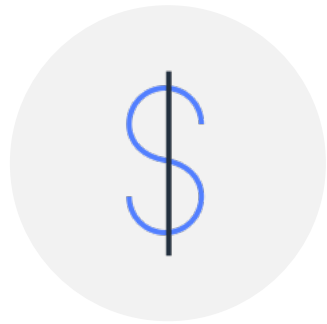
- You created RDS, Redshift, Route 53, ELBs, EIPs, and EBS resources that are now underutilized
- You are paying for the resources as if they are still being fully used



Solution: AWS Trusted Advisor

# AWS Trusted Advisor

For Business Support Plan subscribers



Cost Optimization

**\$1,092.56**  
Potential monthly savings



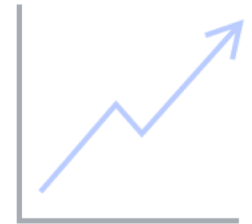
Performance



Security



Fault Tolerance



Service Limits






# Cleanup underutilized resources identified by AWS Trusted Advisor

Implementation time	Savings potential	Time to realize savings	Commitment required
Hours to days	\$100s to \$1000s	Hours to days	None

# Getting started

- 1 Refresh all of your Cost Optimization checks
- 2 Review recommended actions and potential savings provided by the checks
- 3 Prioritize resource cleanup based on greatest savings

## Cost Optimization Checks

- ▶  **Underutilized Amazon Redshift Clusters**  
Checks your Amazon Redshift configuration for clusters that appear to be underutilized.  
1 of 1 Amazon Redshift clusters appear to be idle. Monthly savings of up to \$612.00 are available by shutting down the clusters if they are billed at the hourly rate.
- ▶  **Amazon Route 53 Latency Resource Record Sets**  
Checks for Amazon Route 53 latency record sets that are configured inefficiently.  
0 of 0 domain names have only one latency resource record set.
- ▶  **Idle Load Balancers**  
Checks your Elastic Load Balancing configuration for load balancers that are not actively used.  
0 of 0 load balancers appear to be idle. Monthly savings of up to \$0 are available by minimizing unused load balancers.
- ▶  **Unassociated Elastic IP Addresses**  
Checks for Elastic IP addresses (EIPs) that are not associated with a running Amazon Elastic Compute Cloud (Amazon EC2) instance.  
0 of 0 Elastic IP addresses are not associated with a running instance.
- ▶  **Underutilized Amazon EBS Volumes**  
Checks Amazon Elastic Block Store (Amazon EBS) volume configurations and warns when volumes appear to be underused.  
0 of 3 EBS volumes appear to be underutilized. Monthly savings of up to \$0 are available by minimizing underused EBS volumes.

<https://aws.amazon.com/premiumsupport/technology/trusted-advisor/>  
<https://aws.amazon.com/premiumsupport/pricing/>



# Final thoughts

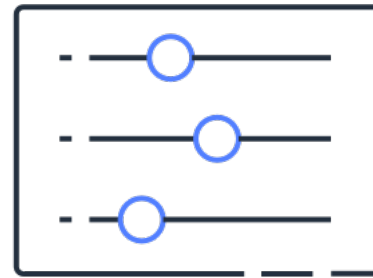


To understand  
your costs

use



AWS  
Cost Explorer

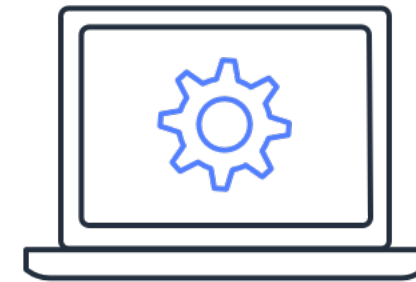


To control  
your costs

use



AWS Budgets



To optimize  
your costs

use

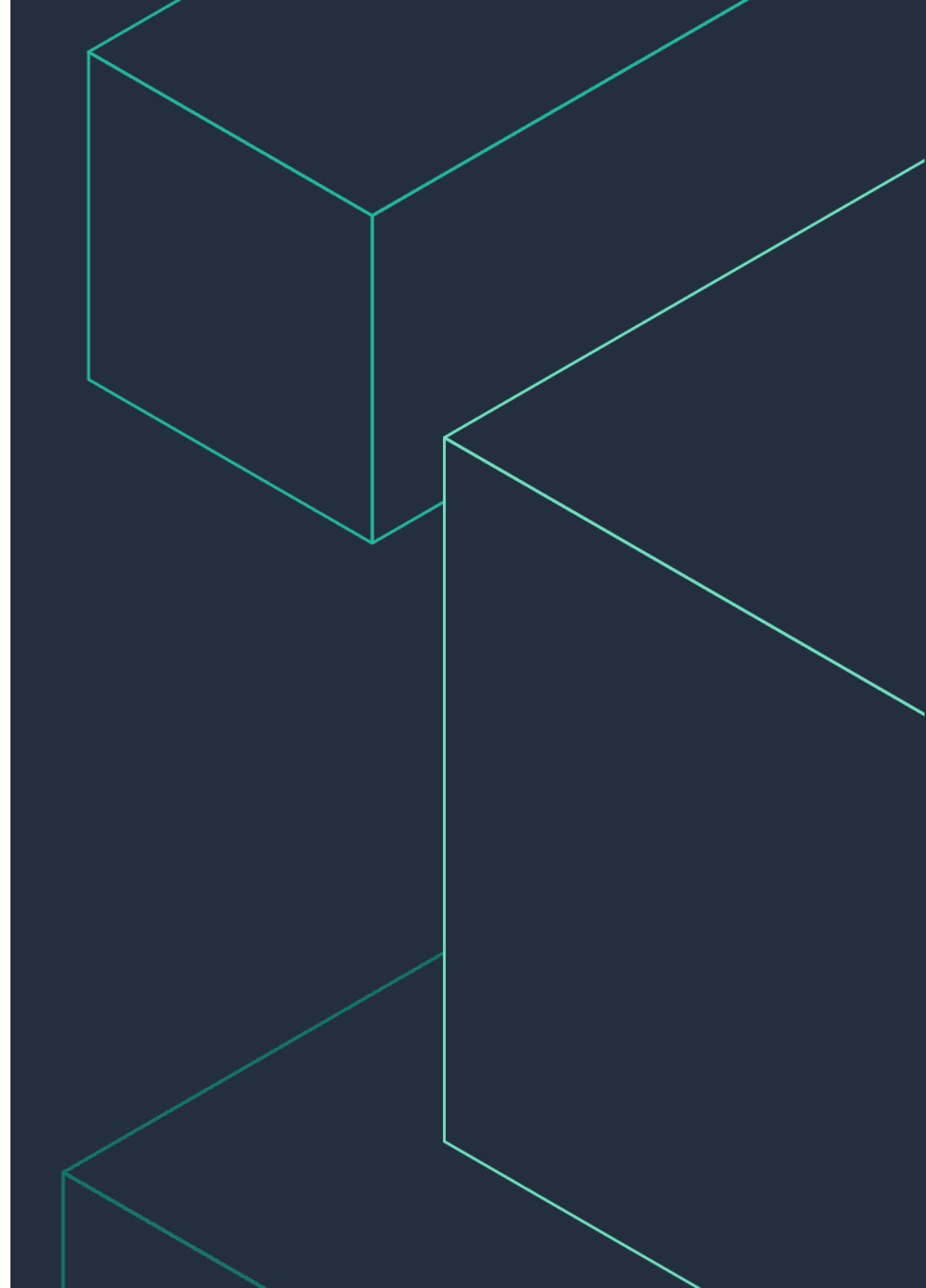


AWS  
Recommendations

# Every startup is different

You may not have seen your particular scenario in this presentation. If you didn't, there are other AWS resources available.

We're here to help.



# Thank You