Vantalect

AWS COST · SAMPLE DELIVERABLE

Spend Baseline and Inventory

Week 1 of 6: Baseline and inventory · prepared for StrataForge Robotics, Inc.

Prepared by Charles Sieg · charles.sieg@vantalect.com · vantalect.com

August 28, 2026 · AWS Cost Analysis & Optimization: The Optimization (six weeks)

SAMPLE DELIVERABLE

This is a sample deliverable. StrataForge Robotics, Inc. is a fictional company: every account, resource, finding, figure, and person in this report is simulated to show the form and depth of a Vantalect engagement. AWS prices are list prices from the AWS Price List API on October 13, 2026.

Contents

Executive summary ... **3**
Where we will look first ... 4
Access and data sources ... **5**
What the numbers come from ... 5
The read-only role ... 6
Cost and Usage Report 2.0 and Compute Optimizer ... 10
What the data cannot tell us ... 11
Twelve months of spend ... **11**
Where the growth came from ... 13
The March 2026 replication spike ... 13
The security-services step-up ... 14
Spend by account ... **14**
Which repository builds what ... 14
The 12 largest accounts ... 16
Spend by category, service, environment, business unit, and Region ... **17**
Category ... 17
Service ... 18
Environment ... 19
Business unit ... 19
Region ... 20
Attribution: who owns the bill ... **20**
Resource inventory ... **22**
Utilization profile ... **24**
Commitments in place ... **25**
The private pricing agreement ... **26**
Forecast with no action ... **27**
What week 2 will do ... **28**

Executive summary

StrataForge's AWS bill is **$4.6 million a month** at the August run-rate, or **$55.7 million a year**, across 48 accounts. Those figures are net: after the 7% private pricing discount, with Savings Plans and reservations spread over their terms, and including $168,726 a month of AWS Enterprise Support.

The bill is growing much faster than the business. July 2026 cost $4.6 million, which is **49% more than August 2025** ($3.1 million). StrataForge's revenue grew 9% over the same period the board used for comparison. If nothing changes, the next twelve months cost $70.6 million and the monthly bill reaches $7.1 million by August 2027 (3.6% compound growth a month, carried forward from the August run-rate).

What you most need to know from week 1:

1. **Growth is concentrated.** The robot-learning platform, the telemetry pipeline, and the 5 accounts that arrived with Kestrel Vision account for most of the increase (Twelve months of spend). Nothing in the history suggests a single bad month except March 2026, when a replication loop in one account added $118,400.
2. **About a quarter of the bill has no owner.** 76% of net spend carries a Team tag. The other 24%, about $1,127,767 a month, cannot be charged to anyone, and 12 accounts hold most of it (Attribution: who owns the bill).
3. **The estate is large and carries old risks.** 2,861 EC2 instances, 4,712 EBS volumes, 65,019 snapshots, and 317 S3 buckets. 63 of 148 database instances and 15 of 16 EKS clusters are on versions that already cost extra in Extended Support (Resource inventory).
4. **About half the general-purpose fleet is lightly used.** Of the 2,366 running instances that are not GPU hosts, 1,169 (49%) never exceeded 20% CPU at the 95th percentile (Utilization profile).
5. **Commitments cover 33.9% of the Linux On-Demand EC2 pool, and the largest plan ends soon.** The one-year Compute Savings Plan ends October 31, 2026. The commitment you renew is the biggest single decision in this engagement, and it should be sized after rightsizing, not before (Commitments in place).
6. **The private pricing agreement has $13.7 million of room today.** Annualized spend is $55.7 million against an annual commitment of $42.0 million in contract years 2 and 3. That room is what any savings program spends first (The private pricing agreement).

The baseline in ten numbers.

Measure	Value
August 2026 run-rate, net, per month	$4,642,925
August 2026 run-rate at list prices, per month	$5,313,307
Annualized, net	$55,715,103
Last twelve months, net (August 2025 through July 2026)	$45,291,313
July 2026 over August 2025	49.2%
Revenue growth, same comparison period	9%
Spend with a Team tag	75.7%
Compute Savings Plans and Linux reservations, share of the Linux On-Demand EC2 pool covered	33.9%
Fleet Cloud cost per connected robot per month (212,000 robots)	$10.18
No-action forecast, next twelve months	$70,640,951

How to read the dollar figures

Every figure is U.S. dollars per month unless it says "a year." "Net" means the amount StrataForge pays after the private pricing discount, with Savings Plan and reservation fees spread across the hours they cover, so a plan's cost lands on the usage it covers. Support is included in totals and shown as its own line. Savings are not ranked in week 1; we rank them in week 2 against list prices and show the arithmetic.

Where we will look first

The baseline points at the places below. We name them now so that you can tell us in the architecture walkthrough if any of them has a reason we do not know. Week 2 sizes each one.

- **Storage, $1.1 million a month.** S3 holds 29.9 PB in 56.5 billion objects, and snapshots hold 1,401 TB billed in 4,445 chains. Week 2 measures how much of it has not been read in 90 days and what the right class is for each data type.
- **Idle and orphaned resources.** EBS has 1,511 TB provisioned; 3,656 still gp2; 1,428 unattached. There are 512 public IPv4 addresses (166 Elastic IPs not associated).
- **Compute that is oversized.** 1,169 instances never exceeded 20% CPU at the 95th percentile. The inventory shows 2,813 running, 48 stopped; 41,426 vCPUs and 710 GPUs running.

- **The EC2 Capacity Reservations line, $76,975 a month.** It is the p5 reservation for Crucible training. We will compare what is reserved with what runs in it.
- **Network paths.** NAT gateways cost $50,992 a month and move traffic through 53 gateways (1,856 TB processed a month). Network Firewall and Transit Gateway sit on the same path.
- **Observability.** CloudWatch costs $228,864 a month. There are 3,261 log groups (268 TB ingested a month; 2,287 never expire) and 848,000 custom metrics (per-robot telemetry metrics in four Regions).
- **Versions in paid Extended Support.** Of 148 database instances, 63 are on engine versions in paid Extended Support. Of 16 EKS clusters, 15 are on Kubernetes versions in extended support. Of 22 ElastiCache clusters, 9 are on Redis OSS 5 (paid Extended Support). AWS charges an Extended Support fee for each, and the fee rises with time.

Access and data sources

We received read access to every in-scope account on August 24, 2026, the first day of the engagement. This section records what we read, so that the numbers in the rest of this report can be reproduced and challenged.

What the numbers come from

The sources behind the baseline.

Source	What it gives us	Window	Limits to keep in mind
Cost and Usage Report 2.0 (Data Exports)	Hourly cost by account, service, resource ID, tag, and discount; Savings Plan and reservation fees spread over the usage they cover	Twelve complete months, August 2025 through July 2026, plus August to date	Tag columns are blank wherever a resource has no tag; costs for the month in progress are still settling
AWS Cost Explorer and Cost Optimization Hub	Cross-check of service totals, and AWS's own rightsizing and idle-resource recommendations	Last twelve months	Hub estimates use your discounts but ignore dependencies between recommendations
AWS Compute Optimizer	Rightsizing and Graviton findings for instances, Auto Scaling groups, volumes, Lambda functions, and databases	Compute Optimizer reads the metrics CloudWatch already holds, with a 14-day lookback by default	Memory appears only where the CloudWatch agent runs; the Hub does not see GPU hosts as candidates

Source	What it gives us	Window	Limits to keep in mind
Amazon CloudWatch metrics	Percentile CPU, memory, network, connections, and IOPS per resource	14 days for rightsizing, 30 days for idle checks	Averages hide bursts, so we use the 95th percentile and show it
Resource inventory	A describe call for every resource type in 48 accounts and the 6 Regions where StrataForge runs resources	Taken on August 24, 2026	17 Regions are enabled in each account; we scanned all of them and found resources in 6
Your four repositories	Which repository builds each resource, so that every change goes through a pull request	Default branch on August 24, 2026	Accounts without a repository (below) are changed by a scripted runbook instead

We tied each month's Cost and Usage Report total to the invoice before using it. Bucket contents, database rows, secrets, and log events were never read; the role cannot read them (next section).

The read-only role

Vantalect assumes one IAM role in each account, deployed from your management account as a CloudFormation StackSet through `platform-landing-zone`. It reads configuration and metrics. It cannot change a resource, and an explicit Deny blocks the calls that read the contents of buckets, tables, secrets, parameters, and logs.

```yaml
AWSTemplateFormatVersion: "2010-09-09"
Description: Read-only analysis role for the Vantalect cost engagement. Delete the stack set
when the engagement ends.
Parameters:
  VantalectAccountId:
    Type: String
    Description: The 12-digit AWS account that Vantalect assumes this role from.
    AllowedPattern: "^[0-9]{12}$"
  ExternalId:
    Type: String
    NoEcho: true
    MinLength: 16
  CurBucketName:
    Type: String
    Description: The S3 bucket that holds the Data Exports files, in the management account.
  AthenaResultsBucketName:
    Type: String
    Description: The S3 bucket that holds Athena query results, in the management account.
  AthenaWorkgroup:
    Type: String
    Default: vantalect-cur
    Description: The Athena workgroup the role may query, with results under the vantalect/
```

```
prefix of the results bucket.
  CurGlueDatabase:
    Type: String
    Default: strataforge_cur
    Description: The Glue database that holds the table over the Data Exports files.
Resources:
  AnalysisRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: VantalectAnalysisReadOnly
      MaxSessionDuration: 3600
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              AWS: !Sub "arn:${AWS::Partition}:iam::${VantalectAccountId}:root"
            Action: sts:AssumeRole
            Condition:
              StringEquals:
                sts:ExternalId: !Ref ExternalId
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/job-function/ViewOnlyAccess"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/ComputeOptimizerReadOnlyAccess"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AWSBillingReadOnlyAccess"
      Policies:
        - PolicyName: MetricsAndInventory
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - cloudwatch:GetMetricData
                  - cloudwatch:GetMetricStatistics
                  - cloudwatch:ListMetrics
                  - cloudwatch:DescribeAlarms
                  - logs:DescribeLogGroups
                  - ce:Get*
                  - ce:List*
                  - bcm-data-exports:Get*
                  - bcm-data-exports:List*
                  - savingsplans:Describe*
                  - pricing:GetProducts
                  - pricing:DescribeServices
                  - organizations:Describe*
                  - organizations:List*
                Resource: "*"
        - PolicyName: CostExportQueries
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Sid: RunQueriesInOneWorkgroup
                Effect: Allow
                Action:
                  - athena:StartQueryExecution
                  - athena:StopQueryExecution
                  - athena:GetQueryExecution
```

```
                  - athena:GetQueryResults
                  - athena:GetWorkGroup
                Resource: !Sub "arn:${AWS::Partition}:athena:us-east-1:$
{AWS::AccountId}:workgroup/${AthenaWorkgroup}"
              - Sid: ReadTableDefinitions
                Effect: Allow
                Action:
                  - glue:GetDatabase
                  - glue:GetTable
                  - glue:GetTables
                  - glue:GetPartition
                  - glue:GetPartitions
                  - glue:BatchGetPartition
                Resource:
                  - !Sub "arn:${AWS::Partition}:glue:us-east-1:${AWS::AccountId}:catalog"
                  - !Sub "arn:${AWS::Partition}:glue:us-east-1:${AWS::AccountId}:database/$
{CurGlueDatabase}"
                  - !Sub "arn:${AWS::Partition}:glue:us-east-1:${AWS::AccountId}:table/$
{CurGlueDatabase}/*"
              - Sid: ReadTheExportFiles
                Effect: Allow
                Action: s3:GetObject
                Resource: !Sub "arn:${AWS::Partition}:s3:::${CurBucketName}/cur2/*"
              - Sid: ReadAndWriteQueryResults
                Effect: Allow
                Action:
                  - s3:GetObject
                  - s3:PutObject
                  - s3:AbortMultipartUpload
                Resource: !Sub "arn:${AWS::Partition}:s3:::${AthenaResultsBucketName}/
vantalect/*"
              - Sid: ListTheTwoBuckets
                Effect: Allow
                Action:
                  - s3:ListBucket
                  - s3:GetBucketLocation
                Resource:
                  - !Sub "arn:${AWS::Partition}:s3:::${CurBucketName}"
                  - !Sub "arn:${AWS::Partition}:s3:::${AthenaResultsBucketName}"
        - PolicyName: NoDataPlaneReads
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Sid: DenyObjectReadsOutsideTheCostExport
                Effect: Deny
                Action:
                  - s3:GetObject
                  - s3:GetObjectVersion
                NotResource:
                  - !Sub "arn:${AWS::Partition}:s3:::${CurBucketName}/cur2/*"
                  - !Sub "arn:${AWS::Partition}:s3:::${AthenaResultsBucketName}/vantalect/*"
              - Sid: DenyAthenaResultsOutsideTheCostWorkgroup
                Effect: Deny
                Action:
                  - athena:GetQueryResults
                  - athena:GetQueryResultsStream
                NotResource: !Sub "arn:${AWS::Partition}:athena:us-east-1:$
```

```
{AWS::AccountId}:workgroup/${AthenaWorkgroup}"
              - Sid: DenyOtherDataReads
                Effect: Deny
                Action:
                  - dynamodb:GetItem
                  - dynamodb:BatchGetItem
                  - dynamodb:Query
                  - dynamodb:Scan
                  - secretsmanager:GetSecretValue
                  - ssm:GetParameter
                  - ssm:GetParameters
                  - ssm:GetParametersByPath
                  - kms:Decrypt
                  - logs:GetLogEvents
                  - logs:FilterLogEvents
                  - logs:StartQuery
                  - logs:GetQueryResults
                  - sqs:ReceiveMessage
                Resource: "*"
Outputs:
  RoleArn:
    Value: !GetAtt AnalysisRole.Arn
```

Deploy it to every member account from the management account, and to the management account itself as an ordinary stack, because service-managed StackSets skip the management account:

```bash
aws cloudformation create-stack-set \
  --stack-set-name vantalect-analysis-readonly \
  --template-body file://vantalect-analysis-role.yaml \
  --parameters ParameterKey=VantalectAccountId,ParameterValue=<vantalect-account-id> \
               ParameterKey=ExternalId,ParameterValue=<external-id> \
               ParameterKey=CurBucketName,ParameterValue=<cur-bucket> \
               ParameterKey=AthenaResultsBucketName,ParameterValue=<athena-results-bucket> \
  --capabilities CAPABILITY_NAMED_IAM \
  --permission-model SERVICE_MANAGED \
  --auto-deployment Enabled=true,RetainStacksOnAccountRemoval=false

aws cloudformation create-stack-instances \
  --stack-set-name vantalect-analysis-readonly \
  --deployment-targets OrganizationalUnitIds=<root-ou-id> \
  --regions us-east-1

aws cloudformation create-stack \
  --stack-name vantalect-analysis-readonly \
  --template-body file://vantalect-analysis-role.yaml \
  --parameters ParameterKey=VantalectAccountId,ParameterValue=<vantalect-account-id> \
               ParameterKey=ExternalId,ParameterValue=<external-id> \
               ParameterKey=CurBucketName,ParameterValue=<cur-bucket> \
               ParameterKey=AthenaResultsBucketName,ParameterValue=<athena-results-bucket> \
  --capabilities CAPABILITY_NAMED_IAM
```

The Cost and Usage Report is the one data-plane read the engagement needs, so the role gets exactly that and nothing more: query access to one Athena workgroup, the table definitions in one Glue database, and the export files and query results in two named S3 prefixes. The Deny statements exempt only those same ARNs (`NotResource`), so the role still cannot read any other object, query result, secret, parameter, or log event. StrataForge's FinOps team creates the `vantalect-cur` workgroup, the results prefix, and the Glue table over the export before the role is deployed. The export bucket uses S3-managed encryption, so the role needs no KMS access; if you encrypt it with a customer-managed key, tell us and we will agree the one key to allow. In the member accounts the two buckets and the workgroup do not exist, so those statements have no effect there.

To remove access at the end of the engagement, delete the stack instances, then the stack set, then the management-account stack. Nothing else was installed.

Cost and Usage Report 2.0 and Compute Optimizer

The report is a Data Exports table in your management account. This is the export definition we use; it is hourly with resource IDs because rightsizing and idle checks need per-resource cost.

```JSON
{
  "Export": {
    "Name": "strataforge-cur2-hourly",
    "DataQuery": {
      "QueryStatement": "SELECT * FROM COST_AND_USAGE_REPORT",
      "TableConfigurations": {
        "COST_AND_USAGE_REPORT": {
          "TIME_GRANULARITY": "HOURLY",
          "INCLUDE_RESOURCES": "TRUE",
          "INCLUDE_SPLIT_COST_ALLOCATION_DATA": "TRUE",
          "INCLUDE_MANUAL_DISCOUNT_COMPATIBILITY": "FALSE",
          "BILLING_VIEW_ARN": "arn:aws:billing::<management-account-id>:billingview/primary"
        }
      }
    },
    "DestinationConfigurations": {
      "S3Destination": {
        "S3Bucket": "<cur-bucket>",
        "S3Prefix": "cur2",
        "S3Region": "us-east-1",
        "S3OutputConfigurations": {
          "OutputType": "CUSTOM",
          "Format": "PARQUET",
          "Compression": "PARQUET",
          "Overwrite": "OVERWRITE_REPORT"
        }
      }
    },
    "RefreshCadence": { "Frequency": "SYNCHRONOUS" }
```

```
  }
}
```

```bash
aws bcm-data-exports create-export --cli-input-json file://cur2-export.json --region us-east-1
aws compute-optimizer update-enrollment-status --status Active --include-member-accounts --
region us-east-1
aws compute-optimizer get-enrollment-statuses-for-organization --region us-east-1
```

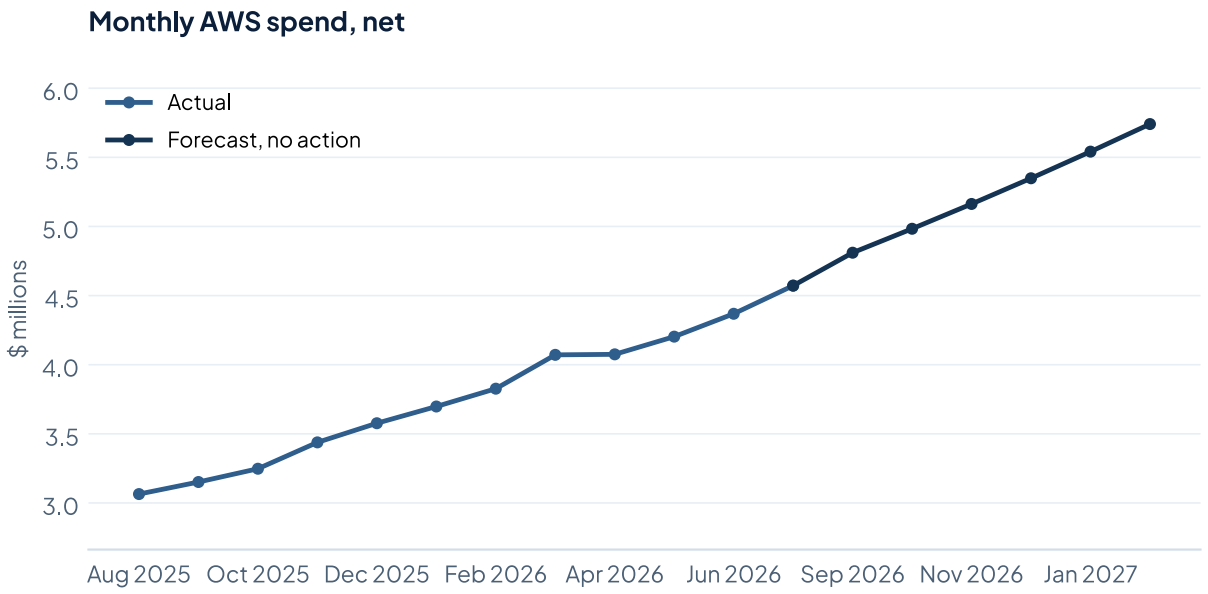
Cost Explorer and the Hub were already enabled. Compute Optimizer was enrolled for the whole organization at kickoff, and its first recommendations arrived within about a day, because it uses metrics CloudWatch already held (a minimum of 30 hours). Where Compute Optimizer and our own measurements disagree in week 2, the report says which one we followed and why.

What the data cannot tell us

- **Memory.** Instances without the CloudWatch agent report CPU but not memory. We size those one step at most and label them "medium confidence."
- **Accounts outside code.** 10 accounts are built by hand and live in no repository: the 5 that came with Kestrel Vision, the 4 sandboxes, and the suspended account. Changes there are made by a scripted runbook with an owner's sign-off.
- **Your GovCloud organization** serves the defense programs and is outside this engagement.
- **Marketplace, tax, and credits** are in the Cost and Usage Report as separate lines and do not change any conclusion in this report.

Twelve months of spend

Spend rose from $3.1 million in August 2025 to $4.6 million in July 2026, a **49% increase** in eleven months. It rose in every month, and April barely rose after the March spike. Revenue grew 9%.



Monthly AWS spend, net, with the no-action forecast.

Net spend by month, including support.

Month	Net spend	Change from prior month
August 2025	$3,064,283	n/a
September 2025	$3,151,424	2.8%
October 2025	$3,247,474	3.0%
November 2025	$3,438,141	5.9%
December 2025	$3,576,595	4.0%
January 2026	$3,697,605	3.4%
February 2026	$3,826,475	3.5%
March 2026	$4,071,248	6.4%
April 2026	$4,074,934	0.1%
May 2026	$4,202,948	3.1%
June 2026	$4,368,351	3.9%
July 2026	$4,571,836	4.7%

Where the growth came from

Account spend (excluding support and the security services shown separately below) grew by $1.4 million a month between August 2025 and July 2026. Eight accounts delivered most of it:

The eight accounts that added the most monthly spend.

Account	Business unit	Owner	August 2025	July 2026	Added per month
crucible-training-prod	Crucible	Hiro Tanaka	$127,062	$416,047	$288,984
crucible-sim-prod	Crucible	Hiro Tanaka	$254,165	$487,183	$233,018
fleet-ingest-prod	Fleet Cloud	Sam Okafor	$369,558	$547,915	$178,357
fleet-api-prod	Fleet Cloud	Sam Okafor	$355,173	$446,248	$91,076
fleet-data-prod	Fleet Cloud	Lucas Ferreira	$250,356	$334,041	$83,685
crucible-data	Crucible	Hiro Tanaka	$109,148	$178,668	$69,520
kestrel-prod	Kestrel Vision	Jonah Reyes	$0	$52,047	$52,047
fleet-perf	Fleet Cloud	Sam Okafor	$59,136	$104,257	$45,121

Three causes sit behind the table. Crucible's robot-learning programs grew GPU and storage use together: the training, simulation, and sensor-log accounts are in the top rows. The telemetry pipeline grew with the connected fleet (212,000 robots today), and cost per robot is now $10.18 a month for Fleet Cloud. And Kestrel Vision's 5 accounts joined in November 2025 and cost $113,778 a month in July, with no repository behind them.

The March 2026 replication spike

crucible-data cost $118,400 more than its trend in March 2026. A replication loop between two buckets in that account copied objects back and forth until it was found and stopped. In Net spend by month, including support, March rose faster than the months around it and April barely rose, which is what a loop that was fully stopped, and not just slowed, looks like.

No alert watched per-account S3 cost, so the loop ran until someone noticed it on the bill. This is the cheapest failure to prevent: week 5 puts Cost Anomaly Detection monitors on every account and on the Team tag, with an alert to the owner above a threshold we set together.

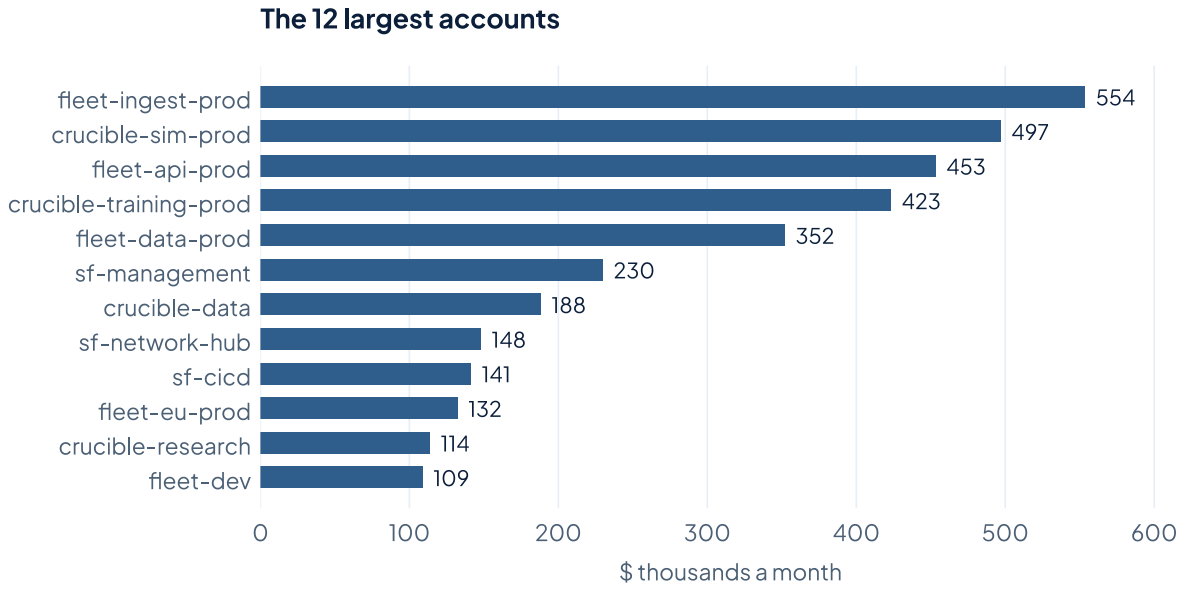
The security-services step-up

Detection services ran at partial coverage until the August hardening. AWS Config, AWS Security Hub CSPM, and Amazon GuardDuty together cost $44,993 a month in August 2025, $58,900 in July 2026 (the last month before detection was enabled everywhere), and **$85,689 a month at the August run-rate**, with every account and Region covered. That is $40,696 a month more than a year ago, and it is spend we recommend keeping.

We do not treat it as waste. Week 2 looks only at how the same detection coverage is recorded: AWS Config recording scope in accounts whose network interfaces live less than an hour, and CloudTrail data-event scope. We will confirm with Marcus Oyelaran, Chief Information Security Officer, that the audit evidence the security engagement mapped stays intact.

Spend by account

Twelve accounts hold 72% of the bill. The other 36 together hold 28%.



The 12 largest accounts, net monthly spend in thousands of dollars.

Which repository builds what

Every change in the Optimization goes through a pull request in one of four repositories, or through a runbook for accounts that live outside code. This table is the map we will use.

Accounts and spend by the repository that builds them.

Repository or method	Accounts	Monthly net	Share of bill
`fleet-infra`	17	$2,149,763	46.3%
`crucible-infra`	6	$1,335,983	28.8%
`platform-landing-zone`	9	$648,896	14.0%
`enterprise-infra`	6	$296,492	6.4%
No repository (scripted runbook)	10	$211,790	4.6%

The accounts outside code hold $211,790 a month. One deserves a look on its own: `suspended-legacy-ci` sits in the Suspended organizational unit, was retired in 2024, and still bills **$32,237 a month**. A suspended account is not an empty account.

The 12 largest accounts

The 12 largest accounts, with July 2026 spend, growth since August 2025, and the repository that builds each one.

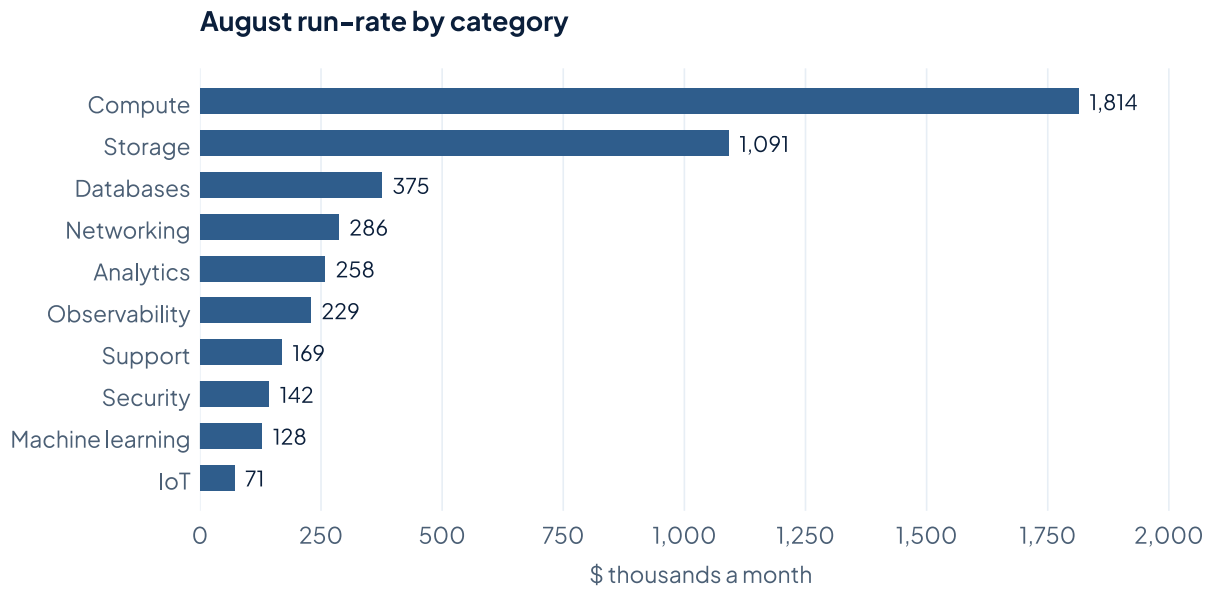
Account	ID	Environment	Business unit	Owner	Built in	August 2025	July 2026	Growth	Monthly net	Share
fleet-ingest-prod	…5012	prod	Fleet Cloud	Sam Okafor	fleet-infra	$369,558	$547,915	48%	$553,574	11.9%
crucible-sim-prod	…7019	prod	Crucible	Hiro Tanaka	crucible-infra	$254,165	$487,183	92%	$496,867	10.7%
fleet-api-prod	…5044	prod	Fleet Cloud	Sam Okafor	fleet-infra	$355,173	$446,248	26%	$453,214	9.8%
crucible-training-prod	…7042	prod	Crucible	Hiro Tanaka	crucible-infra	$127,062	$416,047	227%	$423,241	9.1%
fleet-data-prod	…5071	prod	Fleet Cloud	Lucas Ferreira	fleet-infra	$250,356	$334,041	33%	$352,059	7.6%
sf-management	…0007	management	Platform	Elena Vasquez	platform-landing-zone	$48,746	$61,143	25%	$229,874	5.0%
crucible-data	…7065	prod	Crucible	Hiro Tanaka	crucible-infra	$109,148	$178,668	64%	$188,161	4.0%
sf-network-hub	…1145	shared	Platform	Elena Vasquez	platform-landing-zone	$118,412	$143,238	21%	$147,884	3.2%
sf-cicd	…1356	shared	Platform	Elena Vasquez	platform-landing-zone	$95,718	$129,047	35%	$141,252	3.0%
fleet-eu-prod	…5160	prod	Fleet Cloud	Sam Okafor	fleet-infra	$104,274	$129,433	24%	$132,282	2.9%
crucible-research	…7088	nonprod	Crucible	Hiro Tanaka	crucible-infra	$67,857	$108,555	60%	$113,704	2.5%
fleet-dev	…6037	nonprod	Fleet Cloud	Sam Okafor	fleet-infra	$85,743	$105,745	23%	$108,813	2.3%

The two spend columns and the monthly net column do not measure the same thing. The August 2025 and July 2026 columns are account spend only: they leave out AWS Support and the detection services shown separately, so the 48 accounts add up to $4,346,280 for July 2026 against the $4,571,836 total in Net spend by month, including support, a difference of $225,556. The monthly net column is the August run-rate, which includes Support. AWS bills Support in the management account, so `sf-management` shows $61,143 for July and $229,874 at the run-rate.

Spend by category, service, environment, business unit, and Region

Category

Compute and storage together are 63% of the bill. Everything else is smaller than databases.



August run-rate by category, in thousands of dollars a month.

Net monthly spend by category.

Category	Monthly net	Share
Compute	$1,813,785	39.1%
Storage	$1,091,005	23.5%
Databases	$375,230	8.1%
Networking	$286,234	6.2%
Analytics	$258,026	5.6%
Observability	$228,864	4.9%
Support	$168,726	3.6%
Security	$142,453	3.1%
Machine learning	$128,236	2.8%

Category	Monthly net	Share
IoT	$70,911	1.5%
Other	$62,529	1.4%
End-user computing	$16,926	0.4%

Service

Amazon EC2 is 36% of the bill and Amazon S3 is 15%. Support costs more than any single database service, because it is a percentage of everything else. Several of the smaller lines are the ones to watch: they are the places where architecture, not sizing, sets the price.

The 25 largest services by net monthly spend.

Service	Monthly net	Share
Amazon EC2	$1,658,564	35.7%
Amazon S3	$684,785	14.8%
Amazon CloudWatch	$228,864	4.9%
AWS Support (Enterprise)	$168,726	3.6%
Amazon EBS	$159,524	3.4%
Amazon Aurora	$156,435	3.4%
Amazon SageMaker AI	$128,236	2.8%
Amazon MSK	$96,889	2.1%
Amazon DynamoDB	$91,507	2.0%
Data Transfer	$89,586	1.9%
Amazon FSx for Lustre	$83,995	1.8%
Amazon EC2 Capacity Reservations	$76,975	1.7%
Amazon RDS	$76,351	1.6%
AWS Network Firewall	$73,540	1.6%
AWS Lambda	$72,068	1.6%
AWS IoT Core	$70,911	1.5%
Amazon EBS Snapshots	$66,733	1.4%

Service	Monthly net	Share
Amazon Redshift	$62,472	1.4%
AWS CloudTrail	$56,764	1.2%
NAT Gateway	$50,992	1.1%
Amazon ElastiCache	$50,938	1.1%
Amazon OpenSearch Service	$50,061	1.1%
AWS Backup	$48,734	1.1%
Amazon Elastic Container Registry	$47,235	1.0%
AWS Transit Gateway	$43,812	0.9%

Environment

Production is 72% of the bill, which is what a healthy split looks like. The non-production share, 12%, is where schedules and smaller sizes carry the least risk. Two lines are not environments in any useful sense: the $32,237 a month in suspended accounts, and the $60,301 in sandboxes that no budget watches.

Net monthly spend by environment, from the account's role in the organization.

Environment	Monthly net	Share
prod	$3,359,718	72.4%
nonprod	$541,772	11.7%
shared	$348,779	7.5%
management	$229,874	5.0%
security	$70,243	1.5%
sandbox	$60,301	1.3%
suspended	$32,237	0.7%

Business unit

Fleet Cloud is 46% and Crucible is 30%; together they are 76% of what StrataForge spends on AWS.

Net monthly spend by business unit.

Business unit	Monthly net	Share
Fleet Cloud	$2,158,283	46.5%
Crucible	$1,374,260	29.6%
Platform	$624,396	13.5%
Enterprise IT	$296,492	6.4%
Kestrel Vision	$119,252	2.6%
Security	$70,243	1.5%

Region

us-east-1 carries 74% of spend. "Global" is Support, which AWS bills outside any Region. StrataForge has resources in 6 Regions and 17 enabled in every account; the unused Regions are a security question, covered in the security engagement, and not a cost one.

Net monthly spend by Region.

Region	Monthly net	Share
us-east-1	$3,458,350	74.5%
us-west-2	$494,265	10.7%
eu-central-1	$305,803	6.6%
global	$168,726	3.6%
ap-northeast-1	$140,060	3.0%
ap-southeast-1	$42,294	0.9%
eu-west-1	$33,427	0.7%

Attribution: who owns the bill

75.7% of net spend carries a Team tag. The remaining 24.3%, $1,127,767 a month, belongs to no team that Finance can name. The Savings Plan and reservation fees and Support are counted as attributed, because they are allocated by usage and not by tag.

Twelve accounts hold $814,701 of the untagged spend. The table shows each account's untagged amount and how much of that account it is.

The 12 accounts with the most untagged spend.

Account	Owner	Untagged per month	Share of theBuilt in account
sf-network-hub	Elena Vasquez	$144,692	98%platform-landing-zone
crucible-training-prod	Hiro Tanaka	$108,964	26%crucible-infra
fleet-ingest-prod	Sam Okafor	$92,083	17%fleet-infra
fleet-data-prod	Lucas Ferreira	$88,660	25%fleet-infra
crucible-sim-prod	Hiro Tanaka	$64,271	13%crucible-infra
sf-management	Elena Vasquez	$60,071	26%platform-landing-zone
sf-cicd	Elena Vasquez	$56,606	40%platform-landing-zone
sf-backup	Elena Vasquez	$49,130	100%platform-landing-zone
kestrel-data	Jonah Reyes	$40,036	99%(none)
fleet-api-prod	Sam Okafor	$39,132	9%fleet-infra
corp-data-warehouse	Lucas Ferreira	$35,643	96%enterprise-infra
crucible-data	Hiro Tanaka	$35,413	19%crucible-infra

Two patterns explain most of the gap. Shared network and management accounts (sf-network-hub leads) carry cost that belongs to many teams, and a tag cannot split it; it needs an allocation rule. And the Fleet and Crucible production accounts hold untagged infrastructure created before the tag convention existed.

Week 5 closes the gap with an AWS Organizations tag policy for `Team`, `Application`, `Environment`, and `CostCenter`, enforced for the resource types that carry most of the spend, plus split cost allocation data for the EKS clusters. Until then, the showback report in week 5 shows "untagged" as its own line against each account's owner, so that no team can say the number is not theirs. Check today which of the four keys are active as cost allocation tags, and activate the rest.

Activation is forward-only: a tag shows in the Cost and Usage Report only from the day it is activated, so StrataForge's FinOps team also requests a backfill, which AWS limits to the last twelve months and to one request in 24 hours.

```bash
aws ce list-cost-allocation-tags --status Active --region us-east-1 \
  --query 'CostAllocationTags[].TagKey'
aws ce update-cost-allocation-tags-status --region us-east-1 \
  --cost-allocation-tags-status TagKey=Team,Status=Active TagKey=Application,Status=Active
TagKey=Environment,Status=Active TagKey=CostCenter,Status=Active
aws ce start-cost-allocation-tag-backfill --region us-east-1 \
  --backfill-from 2025-08-01T00:00:00Z
```

Resource inventory

The resource inventory across 48 accounts, taken on August 24, 2026.

Resource type	CountWhat the inventory found
EC2 instances	2,8612,813 running, 48 stopped; 41,426 vCPUs and 710 GPUs running
EBS volumes	4,7121,511 TB provisioned; 3,656 still gp2; 1,428 unattached
EBS snapshots	65,0191,401 TB billed in 4,445 chains
S3 buckets	31729.9 PB in 56.5 billion objects
RDS and Aurora DB instances	14863 on engine versions in paid Extended Support
DynamoDB tables	180120 provisioned
ElastiCache clusters	2286 nodes; 9 on Redis OSS 5 (paid Extended Support)
OpenSearch domains	439 data nodes
MSK clusters	669 brokers
Kinesis data streams	541,971 shards
EKS clusters	1615 on Kubernetes versions in extended support
Lambda functions	1,79019.0 billion invocations a month
Load balancers	17517 Classic
NAT gateways	531,856 TB processed a month

Resource type	CountWhat the inventory found
Public IPv4 addresses	512166 Elastic IPs not associated
CloudWatch log groups	3,261268 TB ingested a month; 2,287 never expire
CloudWatch custom metrics	848,000per-robot telemetry metrics in four Regions
SageMaker notebooks, Studio apps, and endpoints	10022 endpoints

Eight lines in this table carry risk or cost for reasons other than size:

- **Databases: 63 on engine versions in paid Extended Support.** AWS bills Extended Support per vCPU-hour on top of the instance price, and the rate doubles in the third year. Aurora MySQL version 2, RDS for PostgreSQL 11 and 12, and RDS for MySQL 5.7 are already in it.
- **Caches: 86 nodes; 9 on Redis OSS 5 (paid Extended Support).** Redis OSS 5 left standard support on January 31, 2026, and AWS enrolled those clusters in paid Extended Support the next day. ElastiCache for Valkey is the supported path and costs less per node.
- **Kubernetes: 15 on Kubernetes versions in extended support.** Extended support costs $0.60 per cluster-hour against $0.10 in standard support, and the oldest version in the fleet reaches the end of extended support on November 26, 2026, when EKS upgrades the control plane for you with no notice and no rollback.
- **EBS volumes: 1,511 TB provisioned; 3,656 still gp2; 1,428 unattached.** gp3 is cheaper per GB than gp2 and lets you set IOPS and throughput on their own.
- **EBS snapshots: 1,401 TB billed in 4,445 chains.** Snapshot chains outlive the volumes they came from unless a lifecycle policy removes them.
- **Public IPv4: 166 Elastic IPs not associated.** Every public IPv4 address costs $0.005 an hour, attached or not.
- **Logs: 268 TB ingested a month; 2,287 never expire.** A log group with no retention never stops growing.
- **SageMaker: 100 notebooks, Studio apps, and endpoints (22 endpoints).** Notebooks and Studio apps bill by the hour whether or not anyone is typing.

Utilization profile

For every running instance that is not a GPU host, we pulled the 95th percentile of CPU utilization over 14 days. The 95th percentile answers the question a rightsizing decision turns on: how busy does this machine get on a busy day, not on an average one. GPU hosts for simulation and training are excluded, because CPU says nothing about whether a GPU is used.

Non-GPU running EC2 instances by 95th-percentile CPU utilization.

p95 CPU, 14 days	Running instances	Share
Under 10%	684	28.9%
10 to 20%	485	20.5%
20 to 40%	772	32.6%
40 to 60%	383	16.2%
60% or more	42	1.8%
Total	**2,366**	**100%**

684 instances never passed 10% CPU at the 95th percentile, and only 42 (2%) ran hot. CPU alone does not prove an instance is oversized. A database cache can sit at 5% CPU and use all of its memory, and a batch host can idle for 20 hours and need every core in the other four. Week 2 therefore applies CPU, memory (where the agent reports it), and network together, sets different thresholds for production and non-production, and labels every recommendation with a confidence level.

This is the query behind each instance's figure. Run it against any instance ID to check our work. Set `--end-time` exactly 14 days after `--start-time`: the period is the whole window (1,209,600 seconds), so CloudWatch returns one 95th-percentile value instead of one per hour.

```bash
aws cloudwatch get-metric-statistics \
  --namespace AWS/EC2 --metric-name CPUUtilization \
  --dimensions Name=InstanceId,Value=<instance-id> \
  --start-time <start-time> --end-time <end-time> \
  --period 1209600 --extended-statistics p95 --region <region>
```

Commitments in place

StrataForge holds 5 commitments: Savings Plans and reservations that together cost $472,233 a month in fees and save $333,614 a month against On-Demand prices. All of them are fully used today. The two Compute Savings Plans and the Linux EC2 reservations cover **33.9%** of the Linux On-Demand EC2 pool, measured at On-Demand prices. The RDS reservation covers database instances, a different pool, and is outside that percentage.

The commitments in place, in order of expiry.

Commitment	Type and term	Ends	Size	Monthly fee	Monthly savings
`sp-1yr-2025`	Compute Savings Plan, one year, No Upfront	October 31, 2026	$380 an hour	$277,400	$103,065
`ri-r5-8xl`	EC2 Reserved Instances, one year, No Upfront	November 30, 2026	12 x r5.8xlarge	$11,125	$6,535
`ri-aurora`	RDS Reserved Instances, one year, No Upfront	January 14, 2027	6 x db.r5.4xlarge	$6,614	$3,548
`sp-3yr-2024`	Compute Savings Plan, three years, No Upfront	February 28, 2027	$210 an hour	$153,300	$164,362
`ri-hana`	EC2 Reserved Instances, three years, Partial Upfront	May 31, 2027	2 x u-6tb1.112xlarge	$23,794	$56,104

The lowest utilization among them is 100%.

Full utilization looks like good news, and it is. It does not settle what to renew, because a plan renewed this autumn serves the estate that exists after the changes, not the one that exists today. A Compute Savings Plan is a dollars-an-hour commitment, and sizing one before the changes are decided risks buying too little or too much. The order matters: decide the changes first, then size what you renew.

The dates that matter:

- **October 31, 2026: `sp-1yr-2025` ends.** The largest decision of the engagement. It is the one-year plan at $380 an hour, $277,400 a month. Do not renew it as it stands.
- **November 30, 2026: `ri-r5-8xl` ends.** 12 reservations for r5.8xlarge. Check what runs on them before renewing.
- **January 14, 2027: `ri-aurora` ends.** Check what runs on it before renewing.

- **Later expiries** are in the table above. The commitment plan in week 2 puts all of them on one calendar with a purchase date for each.

We will bring a recommendation to the sponsor readout on September 4, 2026, and the purchase decision itself falls in week four, which starts September 14, 2026. Until then, do not renew or buy anything. Vantalect does not purchase commitments; Ravi Menon, Vice President, Finance, approves them and your team purchases them in StrataForge's account.

The private pricing agreement

StrataForge's private pricing agreement with AWS gives a **7% discount** on eligible usage, applied after Savings Plans and reservations, in exchange for an annual spend commitment that steps up: **$36.0 million in contract year 1, $42.0 million in year 2, and $42.0 million in year 3.** The term runs from February 1, 2025 through January 31, 2028, three contract years that begin each February 1.

Today the commitment is not binding. Annualized net spend is $55.7 million, which is $13.7 million (33%) above the $42.0 million of years 2 and 3. That cushion is the reason finance should read the agreement before week 2 ends: every dollar of savings we find comes out of it first. If spend falls below the commitment, the agreement usually charges the difference as a shortfall payment and returns nothing for it.

We have read the summary StrataForge gave us and not the contract. Ravi Menon's team and AWS counsel have to confirm four things:

1. Which spend counts toward the commitment, and whether it is measured before or after the discount. We work from the assumption that it is spend net of the private pricing discount, Savings Plans and reservation fees included.
2. Whether a Savings Plan or reservation purchase counts, and at what amount.
3. How a shortfall is calculated and billed, and whether an overage in one year carries into the next.
4. What the agreement allows if the commitment is renegotiated during the term.

Week 2 models contract year 3 against the forecast and against the roadmap, at today's usage and at the last six months' growth, and the readout puts the result in front of the sponsor with a recommendation.

Forecast with no action

If StrataForge changes nothing, the bill follows the compound monthly growth of the last six months (3.6% a month), carried forward from the August run-rate. The forecast does not include new commitments, new accounts, or any savings.

The no-action forecast, September 2026 through August 2027.

Month	Forecast net spend	Change from prior month
September 2026	$4,810,092	3.6%
October 2026	$4,983,277	3.6%
November 2026	$5,162,698	3.6%
December 2026	$5,348,579	3.6%
January 2027	$5,541,152	3.6%
February 2027	$5,740,659	3.6%
March 2027	$5,947,349	3.6%
April 2027	$6,161,481	3.6%
May 2027	$6,383,323	3.6%
June 2027	$6,613,152	3.6%
July 2027	$6,851,256	3.6%
August 2027	$7,097,932	3.6%
Next twelve months	**$70,640,951**	

The next twelve months cost $70.6 million, 56% more than the last twelve ($45.3 million). By August 2027 the monthly bill is 1.5 times today's. Growth that is faster than revenue growth is not sustainable at any fixed price per robot; the question for the next five weeks is how much of it is waste, how much is waste by design, and how much is the cost of a business that is working.

What week 2 will do

Week 2 runs from August 31, 2026 through the sponsor readout on September 4, 2026. It produces five more deliverables and the readout:

The deliverables of the Analysis, in order.

Deliverable	What it contains
2. The waste register	Every idle and orphaned resource with its ID, owner, monthly cost, and the runbook to remove it
3. The rightsizing plan	Compute, databases, caches, and search sized to measured use, with Graviton where it fits
4. The commitment plan	Compute, Database, and SageMaker Savings Plans and reservations sized to what is left after changes, tested at smaller sizes and against a one-year alternative, with a purchase calendar
5. The architecture review	Data transfer, NAT, storage tiers, logging, and metrics, with the code for each change
6. The savings roadmap	Every opportunity ranked by annual savings, effort, and risk, in four waves
7. The sponsor readout	The identified and realizable savings, the planning range, the top ten, the decisions, and the risks, in one short document

We need five things from you this week:

1. **Owners for every account** in The 12 largest accounts, with July 2026 spend, growth since August 2025, and the repository that builds each one and Accounts and spend by the repository that builds them, with a name for each of the 10 accounts that live outside code. Jonah Reyes has the 5 Kestrel accounts, and Elena Vasquez has the sandboxes and the suspended account.
2. **The private pricing agreement** and the most recent true-up statement, from Ravi Menon's team.
3. **A walkthrough of the Crucible training schedule** with Hiro Tanaka, so the GPU findings reflect how the platform works.
4. **Which Elastic IP addresses customers have allow-listed**, from Sam Okafor's team. Those stay.

5. **A date for the data-governance council**, so that retention changes to the data lakes have a forum before anyone is asked to approve them.

The architecture walkthrough, the finance session, and office hours are on the calendar in Pavilion. Tom Becker remains the engagement's day-to-day contact, and Priya Raman receives the readout.

© 2026 Vantalect, LLC. Sample deliverable; StrataForge Robotics is fictional.

AWS, Amazon Web Services, AWS Graviton, and the AWS service names are trademarks of Amazon.com, Inc. or its affiliates. Kubernetes is a registered trademark of The Linux Foundation. GitHub is a trademark of GitHub, Inc. NVIDIA and Isaac Sim are trademarks of NVIDIA Corporation. SAP and SAP S/4HANA are trademarks of SAP SE. Vantalect is not affiliated with or endorsed by any of them. StrataForge Robotics is fictional.