AWS re:Invent

IOT306

IoT Visualization & Analytics Workshop

You have the Data, Now Let’s Do Something with It

Asha Chakrabarty, Senior Solutions Architect, AWS

Greg Urban, Director, Partner Engineering, PTC

November 30, 2016

amazon web services

Workshop Prerequisites

- Download prerequisites information:
    - http://bit.ly/2gxOMAM

What will we cover today?

- Overview of AWS IoT and Amazon Kinesis
- Understand the benefits of Real-time Data Analysis with AWS IoT and Amazon Kinesis
- Learn how the AWS platform can help turn data into insights & actions
- Review what we will build today

Expectations

- This is not a hackathon or bootcamp.
- No devices and no coding. Yay!
- No C, Java, etc. – It’s all node.js today!
- We assume good knowledge of AWS in general; we won’t cover basics like AWS 101, AWS Identity & Access Management (IAM) roles, etc.
- You have reviewed and set up the prerequisites.
- Collaborate, learn, and have fun!

“If you can’t measure it, you can’t improve it”

-Lord Kelvin

One of the big challenges with the IoT is to

Collect

Analyze

Act on

data from devices to generate insights.

AWS IoT

“Securely connect one or one-billion devices to AWS,
so they can interact with applications and other devices”

1

Securely connect and manage any physical device across multiple networks and protocols

2

Extract and filter data from your devices and take action with custom rules

3

Create web and mobile applications that interact with devices reliably at any time

Device SDK

Device Security and Policy Management

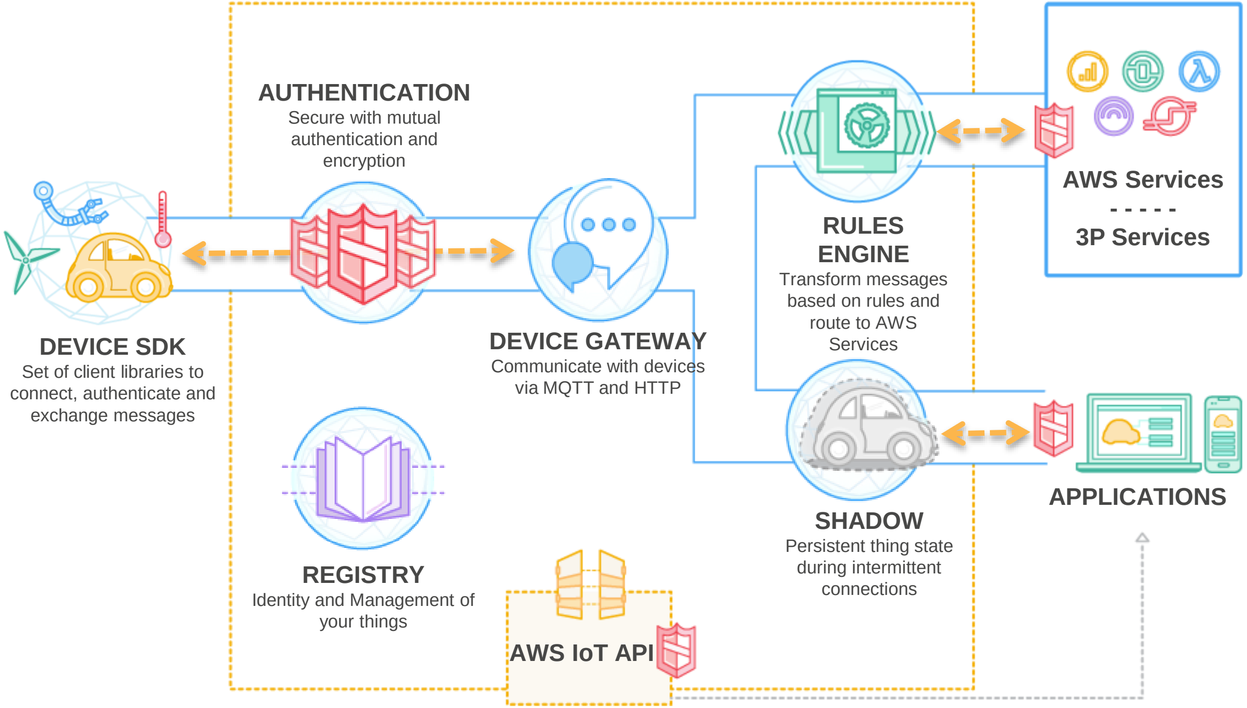
Device Gateway

Registry

Rules Engine

Shadow

AWS IoT: How it Works


AUTHENTICATION
Secure with mutual authentication and encryption
DEVICE SDK
Set of client libraries to connect, authenticate and exchange messages
DEVICE GATEWAY
Communicate with devices via MQTT and HTTP
RULES ENGINE
Transform messages based on rules and route to AWS Services
AWS Services
- - - - -
3P Services
SHADOW
Persistent thing state during intermittent connections
APPLICATIONS
REGISTRY
Identity and Management of your things
AWS IoT API

AWS IoT: Front Door to AWS

Registry

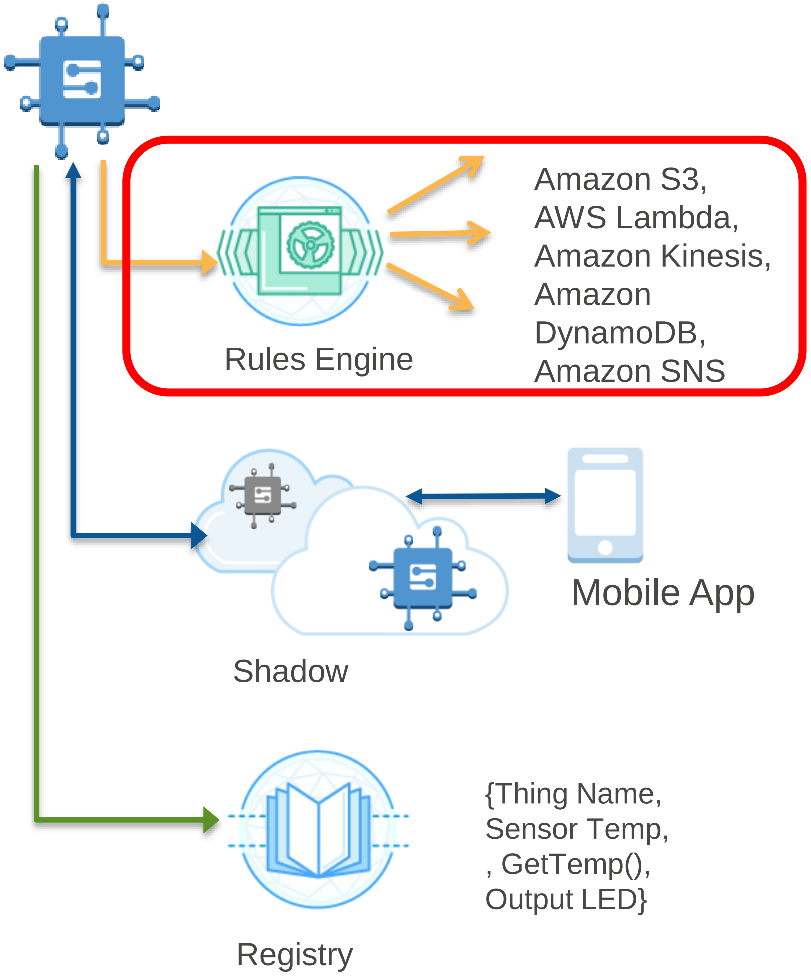
Establishes an identity for devices and manages metadata such as the devices' attributes and capabilities

Shadows

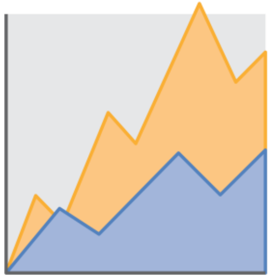
Apps and devices can access "RESTful" Shadow (Thing's State) that is in sync with the device

Rules and Actions

Match patterns and take actions to send data to other AWS services or republish

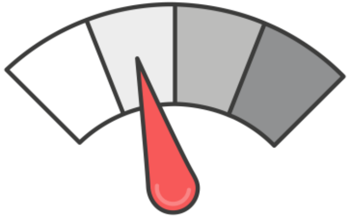


Three Ways to Analyze Data



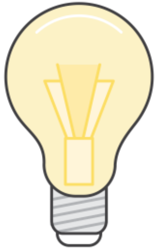
Retrospective
analysis and
reporting

Past Data



Here-and-now
real-time processing
and dashboards

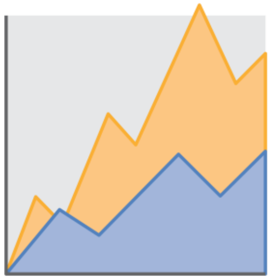
Present Data



Predictions
to enable smart
applications

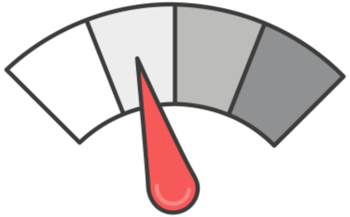
“Future Data”

Three Ways to Analyze Data



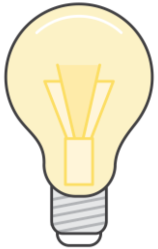
Retrospective
analysis and
reporting

Amazon Redshift
Amazon RDS
Amazon S3
Amazon EMR



Here-and-now
real-time processing
and dashboards

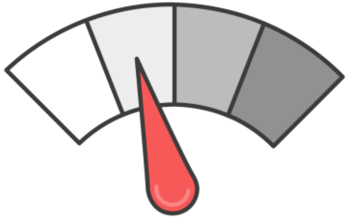
Amazon Kinesis
AWS Lambda
Amazon DynamoDB
Amazon EC2



Predictions
to enable smart
applications

Amazon Machine
Learning

Real Time Requires Quick Processing



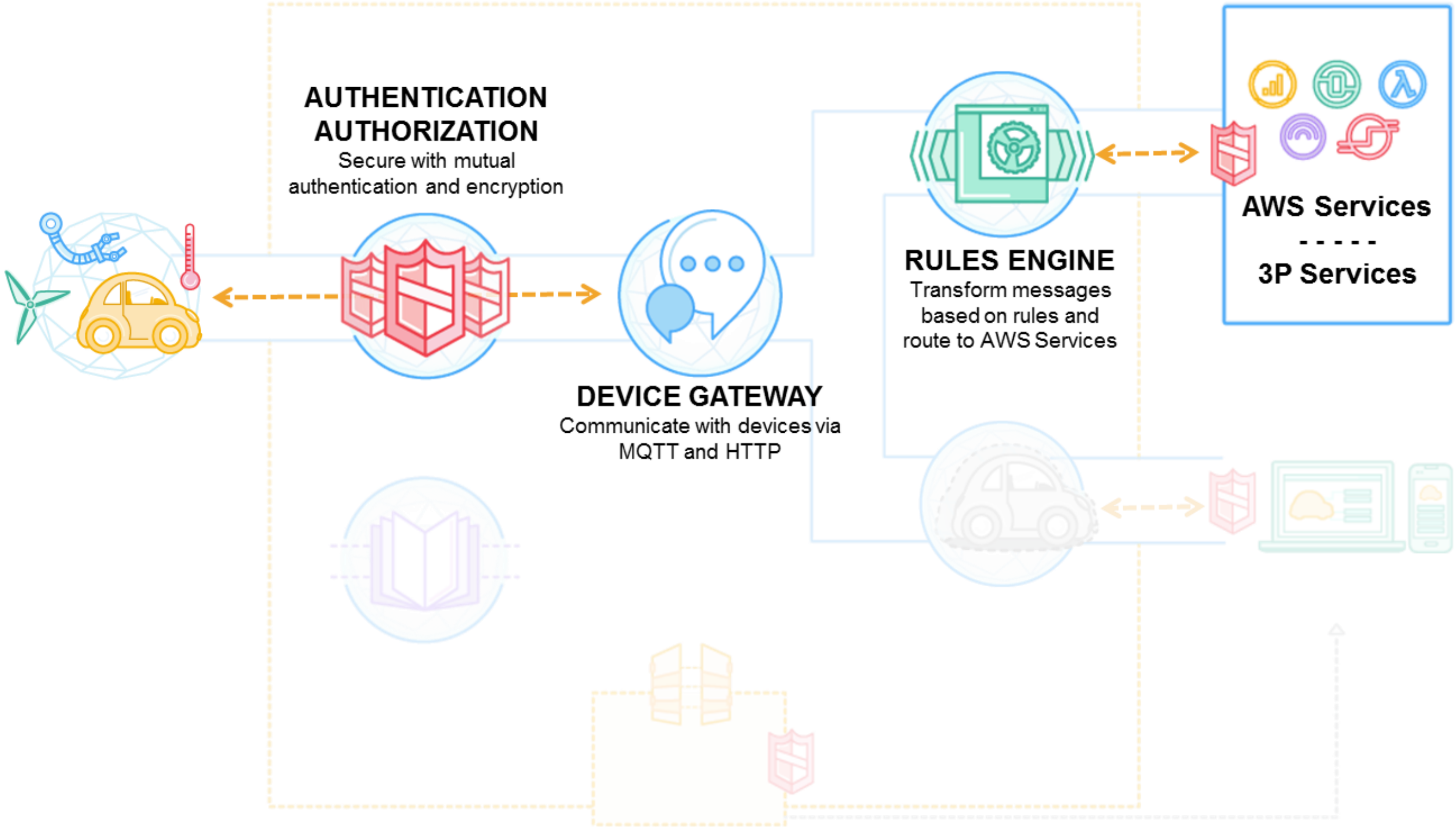
Here-and-now
real-time processing
and dashboards

- Discover patterns in live sensor data
- Correlate events as they happen
- Enrich live data with additional info

Why?

- Trigger quick reactions
- Adapt to usage of Things
- Users want quick reaction & feedback

AWS IoT Telemetry



AUTHENTICATION AUTHORIZATION

Secure with mutual authentication and encryption

DEVICE GATEWAY

Communicate with devices via MQTT and HTTP

RULES ENGINE

Transform messages based on rules and route to AWS Services

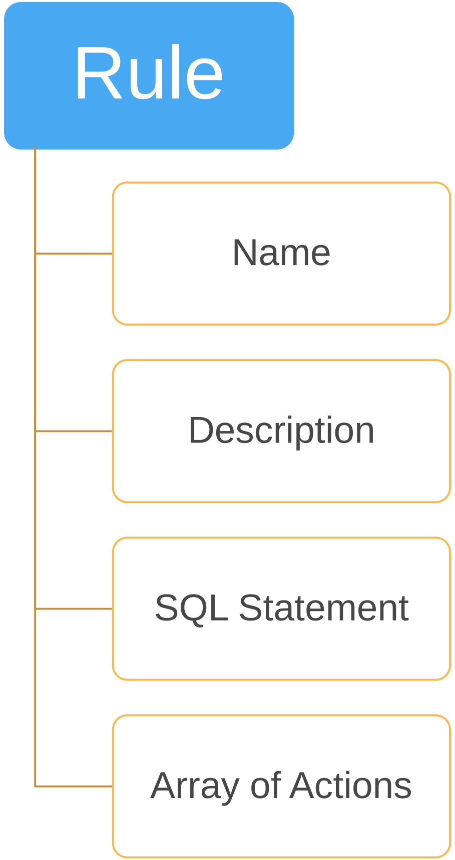
The Lean, Mean Data Analysis Machine

Better Together: AWS IoT & Amazon Kinesis

- Relevant Data Collection (IoT Rules Engine)
- Data Ingestion & Analysis (Amazon Kinesis, AWS Lambda)
- Visualize (ELK, Custom Visualizations)

Relevant Data Collection:
The IoT Rules Engine

AWS IoT Rules Engine Basics



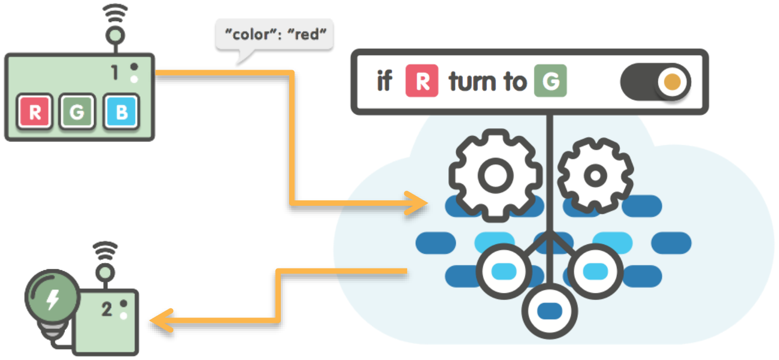
Rule

Name

Description

SQL Statement

Array of Actions



SELECT * FROM ‘things/thing-2/color’
WHERE color = ‘red’

AWS Services, Native

AWS IoT Rules Engine – Format

```
{
    "sql":"SELECT 'IDLE' AS status FROM 'vacuum/+/events' WHERE  event =
'COMPLETE'",
    "actions": [
        {
            "dynamoDB": {
                "tableName":"vaccum-status",
                "hashKeyField":"vacuum_id",
                "hashKeyValue":"${topic(2)}",
                "payloadField":"statusDocument",
                "roleArn":"arn:aws:iam::77777:role/rules_action_ddb"
            }
        }
    ]
}
```

AWS IoT Rules Engine

SELECT DATA FROM TOPIC WHERE FILTER

THEN ACTION

AWS IoT – SQL Reference

SELECT DATA FROM **TOPIC** WHERE FILTER

- Like scanning a database table
- Default source is an MQTT topic

EXAMPLES:

- FROM mqtt(‘my/topic’)
- FROM mqtt(‘my/wildcard/+/topic’)
- FROM (‘my/topic’)

AWS IoT – SQL Reference

SELECT **DATA** FROM…

- SELECT *
- SELECT deviceid, temp
- SELECT coords.latitude
- SELECT a.another_level.b
  - Returns {“b” : 3}
- SELECT a..b
  - Returns {“b” : 3}

SAMPLE PAYLOAD

```
{
        “deviceid” : “iot123”,
        “temp” : 54,
        “humidity” : 32,
        “coords” : {
                “latitude” : 47.615694,
                “longitude” : -122.3359976
        },
        “a” : {
                “another_level” : {
                        {“b” : 3},
                        {“b” : 5}
        }}
}
```

AWS IoT – SQL Reference

SELECT DATA FROM TOPIC WHERE **FILTER**

Token	Meaning	Example
=	Equal, comparison	color = 'red'
<>	Not Equal, comparison	color <> 'red'
AND	Logical AND	color = 'red' AND siren = 'on'
OR	Logical OR	color = 'red' OR siren = 'on'
()	Parenthesis, grouping	color = 'red' AND (siren = 'on' OR isTest)
+	Addition, arithmetic	5 + 3
-	Substitution, arithmetic	5 - 4
/	Division, arithmetic	8 / 2

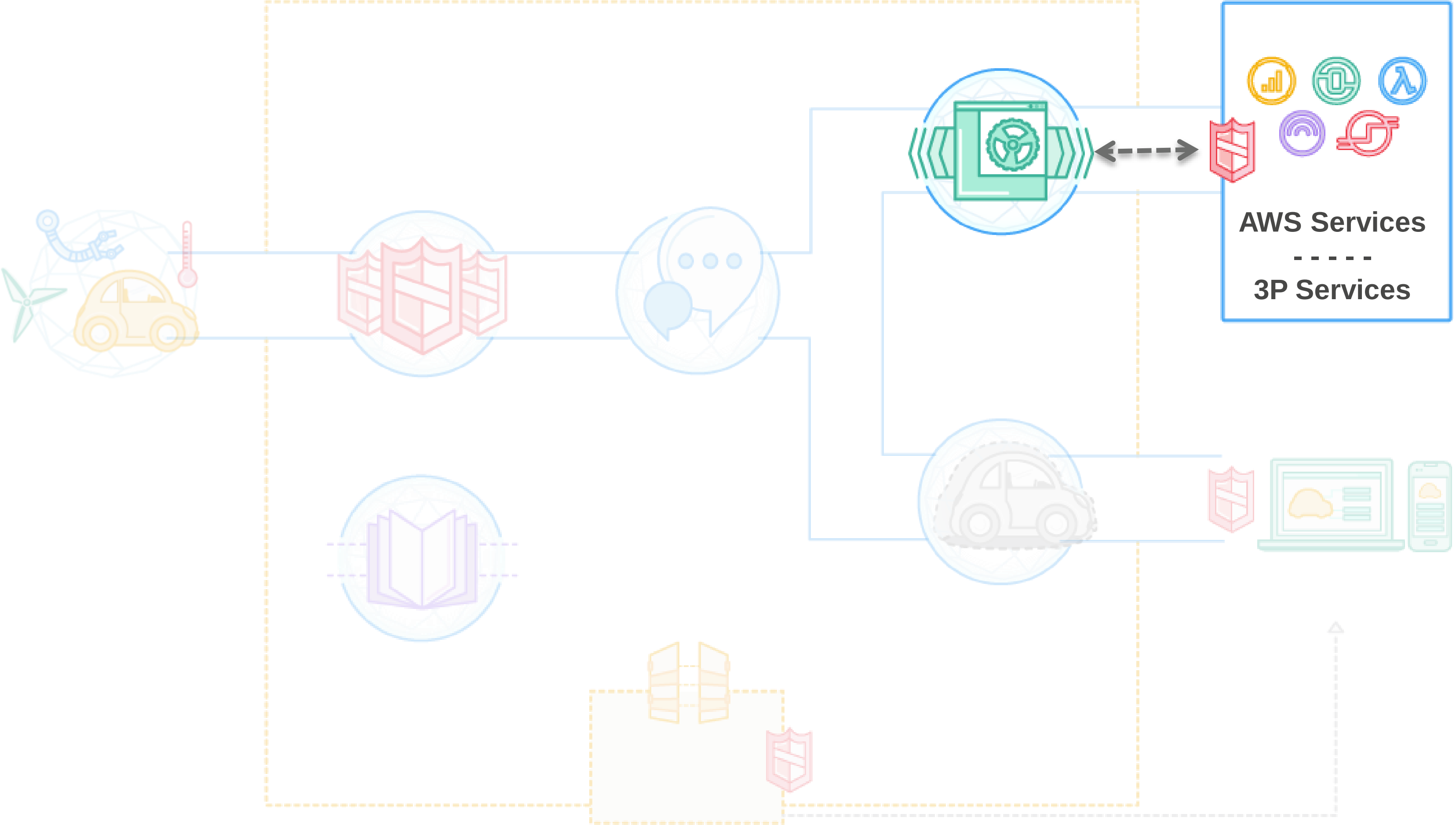
AWS IoT – SQL Reference

SELECT DATA FROM TOPIC WHERE **FILTER**

Token	Meaning	Example
<	Less than, comparison	color = 'red'
<=	Less than or equal	color <> 'red'
>	Greater than, comparison	color = 'red' AND siren = 'on'
>=	Greater than or equal	color = 'red' OR siren = 'on'

CASE …
WHEN …
THEN …
ELSE …
END
	Case statement	CASE location
		WHEN 'home’
		THEN 'off'
		WHEN 'work’
		THEN 'on' ELSE 'silent' END

AWS IoT Rules Engine Actions



AWS IoT Rules Engine

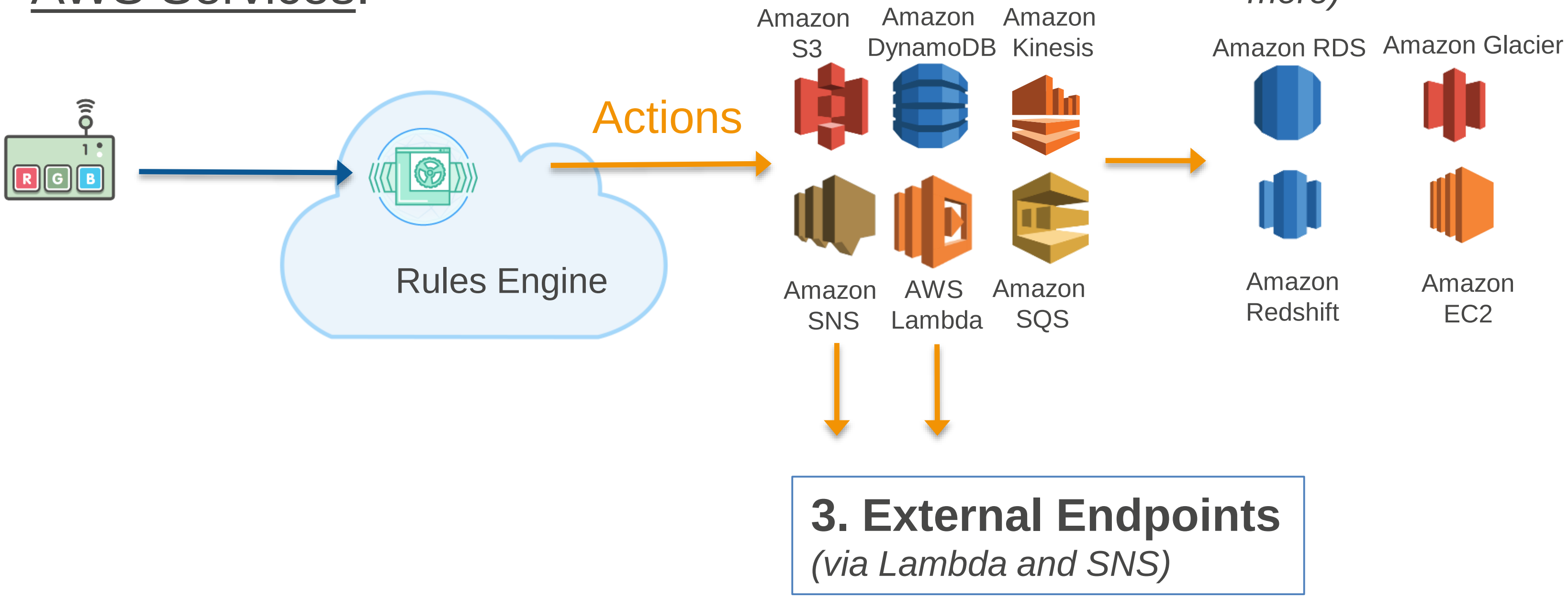
Rules Engine connects AWS IoT to External Endpoints and AWS Services.

1. AWS Services

(Direct Integration)

2. Rest of AWS

(via Amazon Kinesis, AWS Lambda, Amazon S3, and more)



Actions

Rules Engine

Amazon S3

Amazon DynamoDB

Amazon Kinesis

Amazon SNS

AWS Lambda

Amazon SQS

Amazon RDS

Amazon Glacier

Amazon Redshift

Amazon EC2

3. External Endpoints

(via Lambda and SNS)

AWS IoT Rules Engine

Rules Engine evaluates inbound messages published into AWS IoT, transforms and delivers to the appropriate endpoint based on business rules.

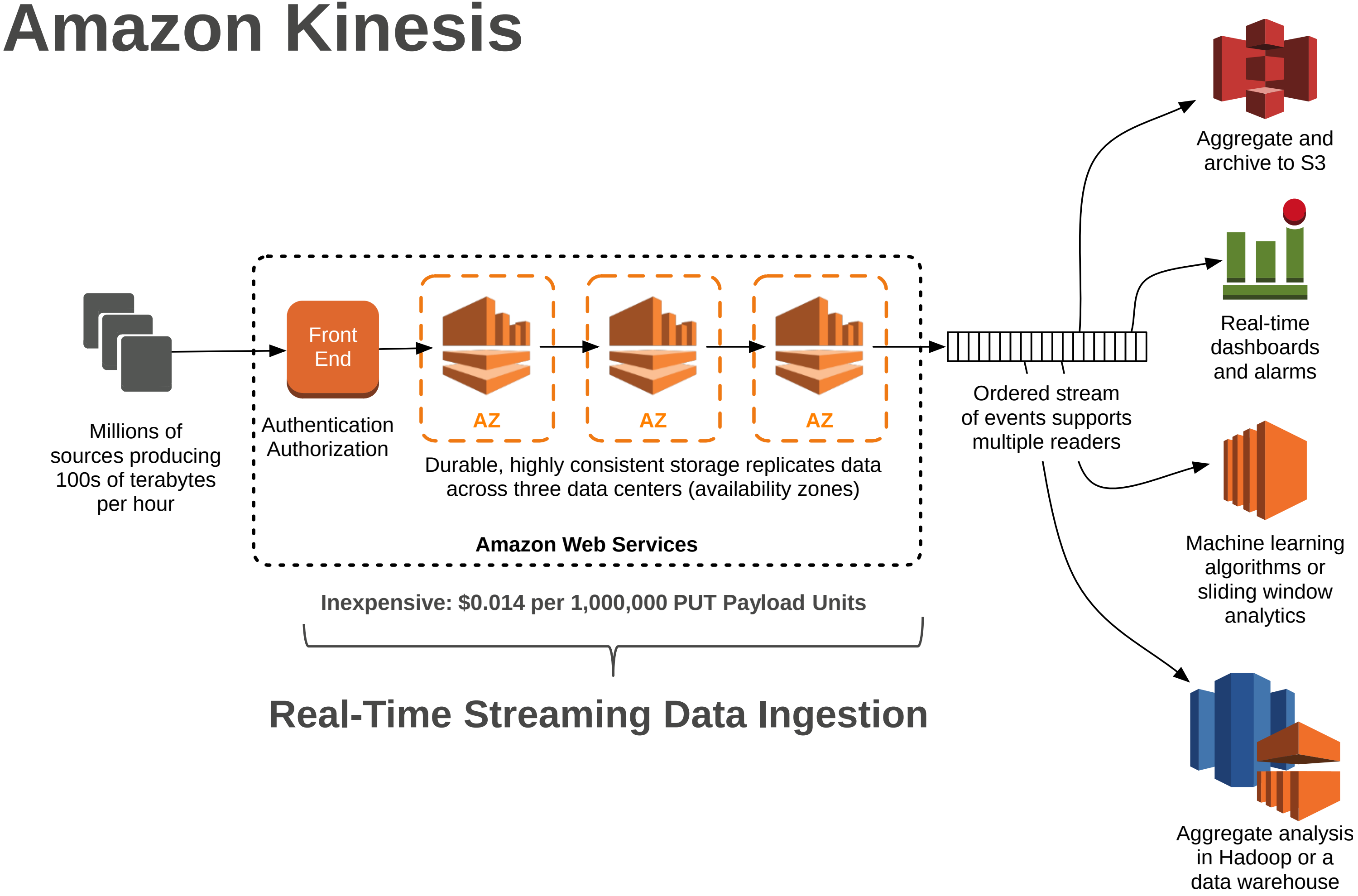
External endpoints can be reached via Lambda and Amazon Simple Notification Service (Amazon SNS).

Actions

- Invoke a Lambda function
- Put object in an S3 bucket
- Insert, Update, Read from a DynamoDB table
- Publish to an SNS Topic or Endpoint
- Publish to an Amazon Kinesis stream
- Publish to Amazon Kinesis Firehose
- Republish to AWS IoT

Data Ingestion:
Amazon Kinesis

Amazon Kinesis



Millions of sources producing 100s of terabytes per hour

Front End

Authentication Authorization

AZ

AZ

AZ

Durable, highly consistent storage replicates data across three data centers (availability zones)

Amazon Web Services

Ordered stream of events supports multiple readers

Aggregate and archive to S3

Real-time dashboards and alarms

Machine learning algorithms or sliding window analytics

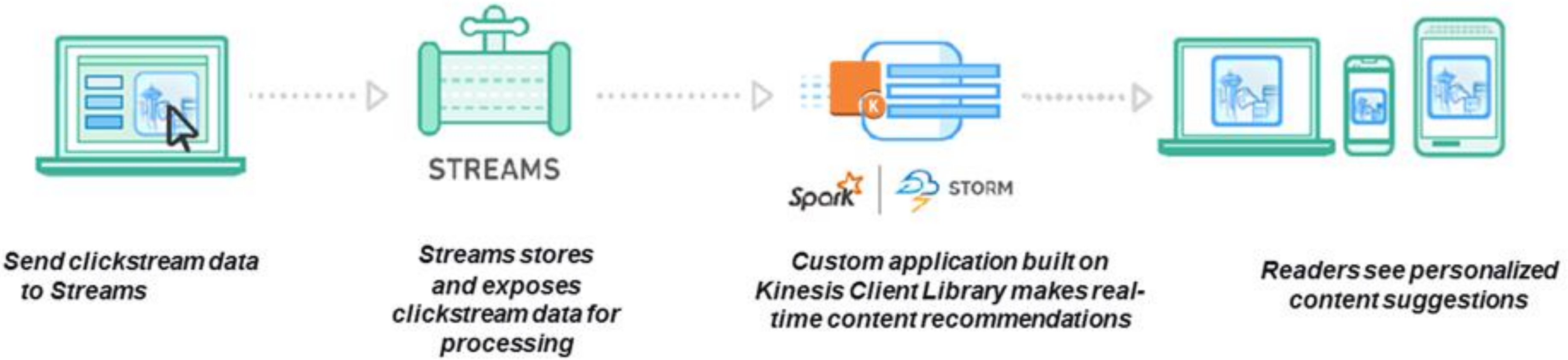
Aggregate analysis in Hadoop or a data warehouse

Inexpensive: $0.014 per 1,000,000 PUT Payload Units

Real-Time Streaming Data Ingestion

Amazon Kinesis Streams

Store data as a continuous stream



STREAMS
Spark
STORM
Send clickstream data to Streams
Streams stores and exposes clickstream data for processing
Custom application built on Kinesis Client Library makes real-time content recommendations
Readers see personalized content suggestions

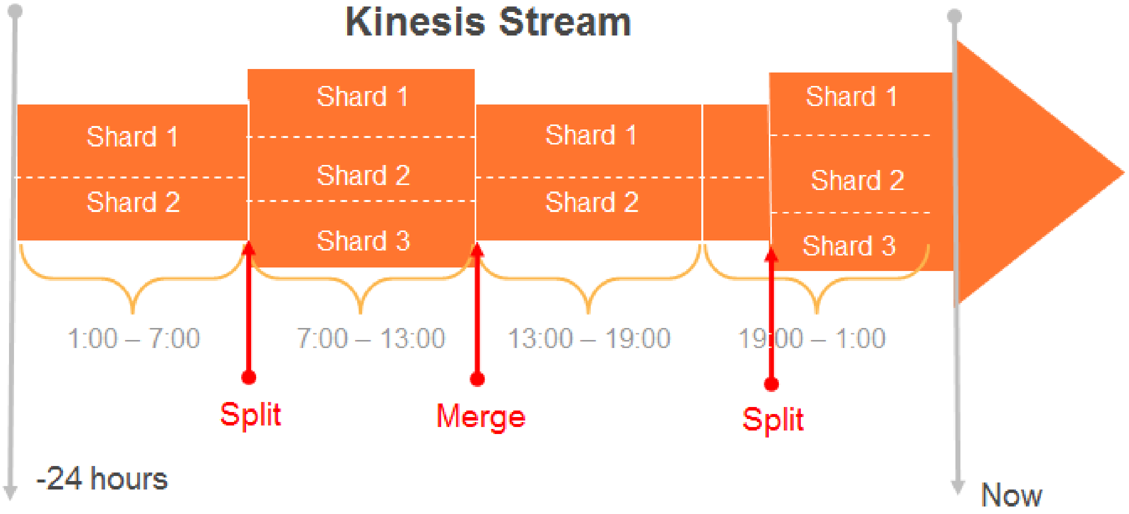
Easy administration: Simply create a new stream and set the desired level of capacity with shards. Scale to match your data throughput rate and volume.

Build real-time applications: Perform continual processing on streaming big data using Amazon Kinesis Client Library (KCL), Apache Spark/Storm, AWS Lambda, and more.

Low cost: Cost-efficient for workloads of any scale.

Amazon Kinesis Stream

Managed Ability to capture and store Data

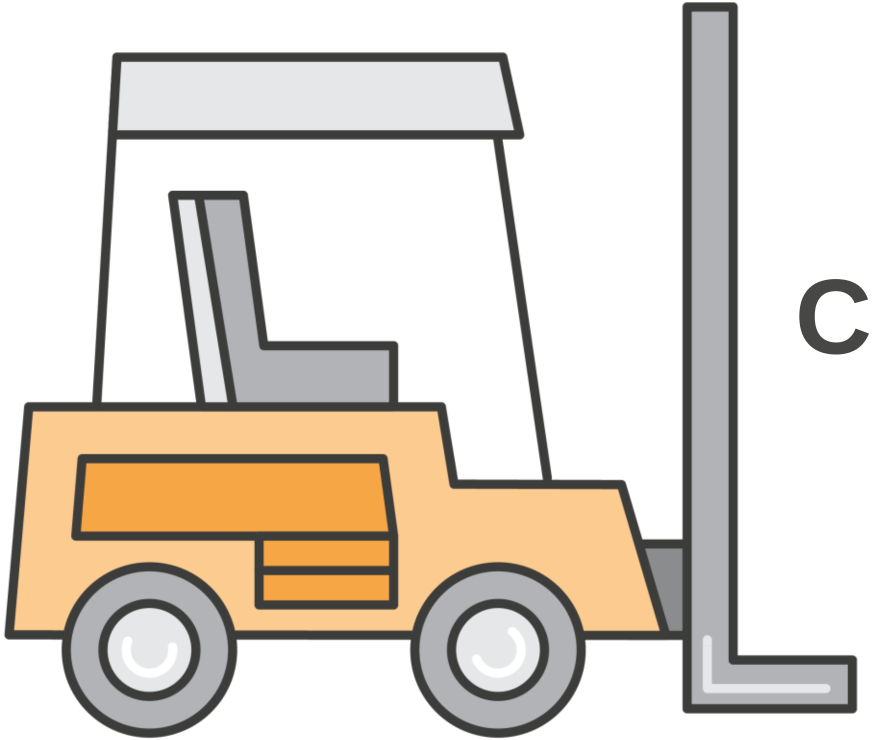


- Streams are made of **Shards**
- Each Shard ingests data up to 1MB/sec, and up to 1000 TPS
- Each Shard emits up to 2 MB/sec
- All data is stored for **24 hours – 7 days**
- **Scale** Kinesis streams by splitting or merging Shards
- **Replay** data inside of 24Hr -7days Window

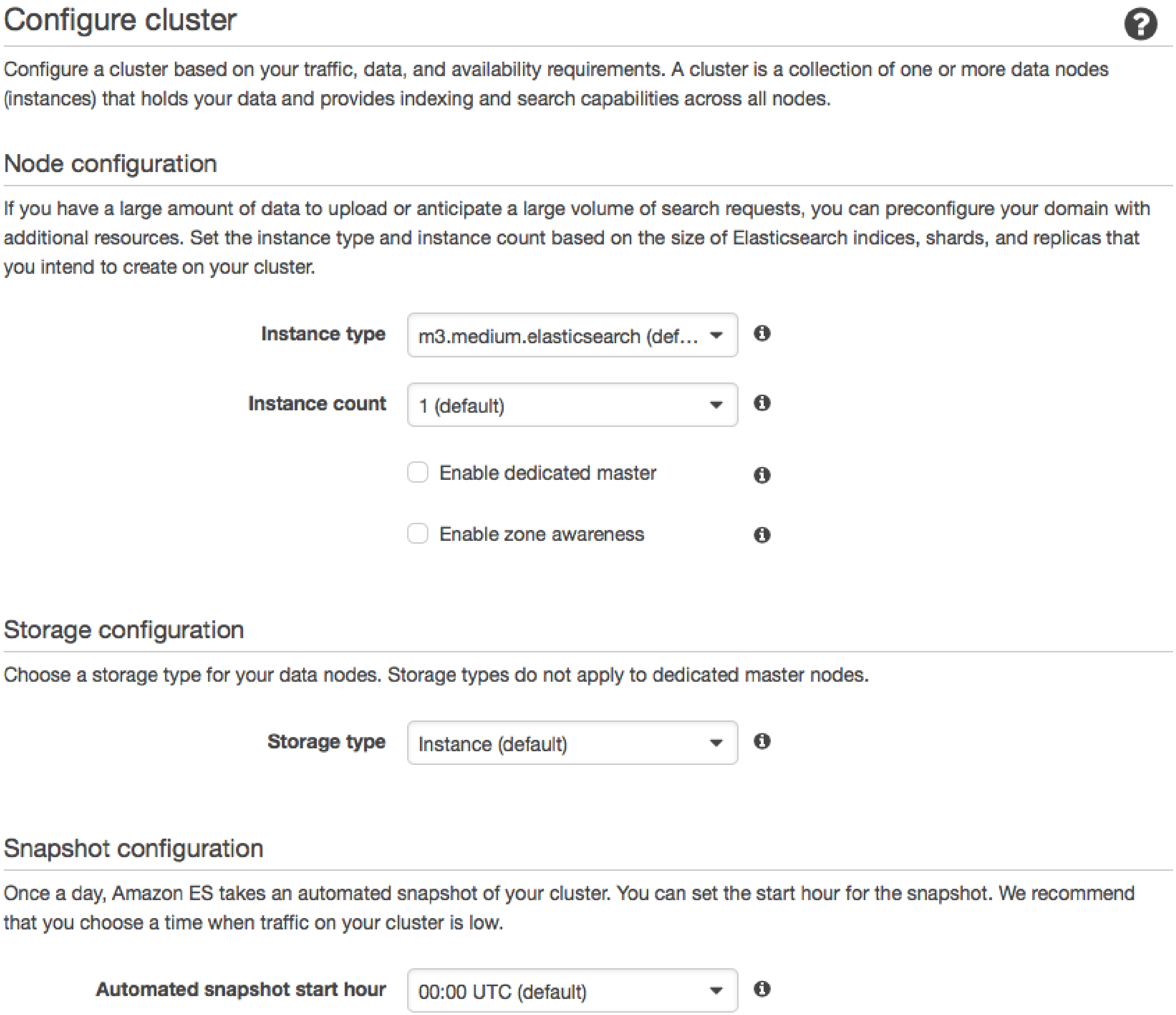
Visualization:
Amazon Elasticsearch Service

Amazon Elasticsearch Service

Amazon Elasticsearch Service is a managed service from AWS that makes it easy to set up, operate, and scale Elasticsearch clusters in the cloud.



Create the cluster



Configure cluster

Configure a cluster based on your traffic, data, and availability requirements. A cluster is a collection of one or more data nodes (instances) that holds your data and provides indexing and search capabilities across all nodes.

Node configuration

If you have a large amount of data to upload or anticipate a large volume of search requests, you can preconfigure your domain with additional resources. Set the instance type and instance count based on the size of Elasticsearch indices, shards, and replicas that you intend to create on your cluster.

Instance type m3.medium.elasticsearch (def…

Instance count 1 (default)

□ Enable dedicated master

□ Enable zone awareness

Storage configuration

Choose a storage type for your data nodes. Storage types do not apply to dedicated master nodes.

Storage type Instance (default)

Snapshot configuration

Once a day, Amazon ES takes an automated snapshot of your cluster. You can set the start hour for the snapshot. We recommend that you choose a time when traffic on your cluster is low.

Automated snapshot start hour 00:00 UTC (default)

AWS CLI commands

add-tags

create-elasticsearch-domain

delete-elasticsearch-domain

describe-elasticsearch-domain

describe-elasticsearch-domain-config

describe-elasticsearch-domains

list-domain-names

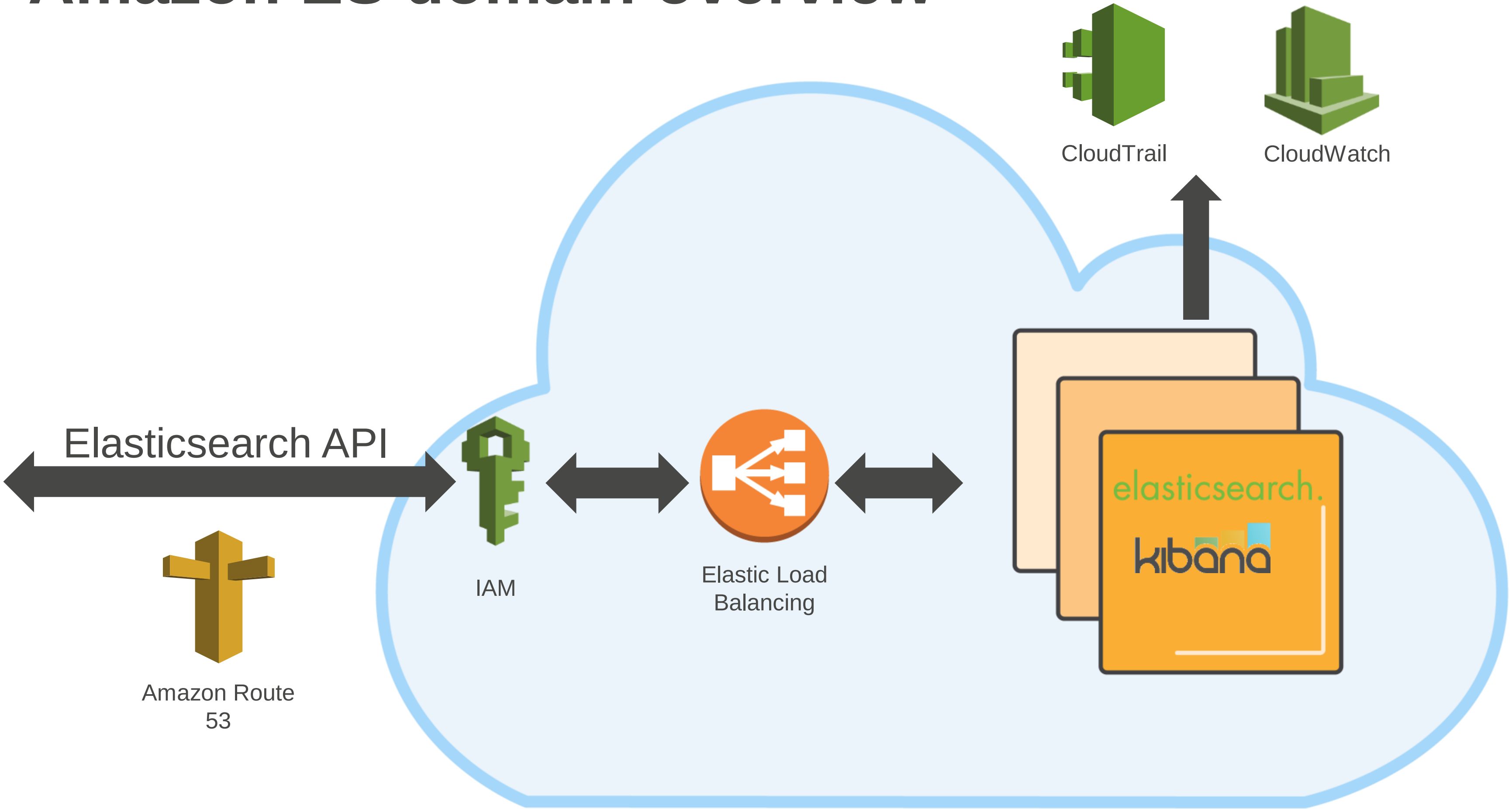
list-tags

remove-tags

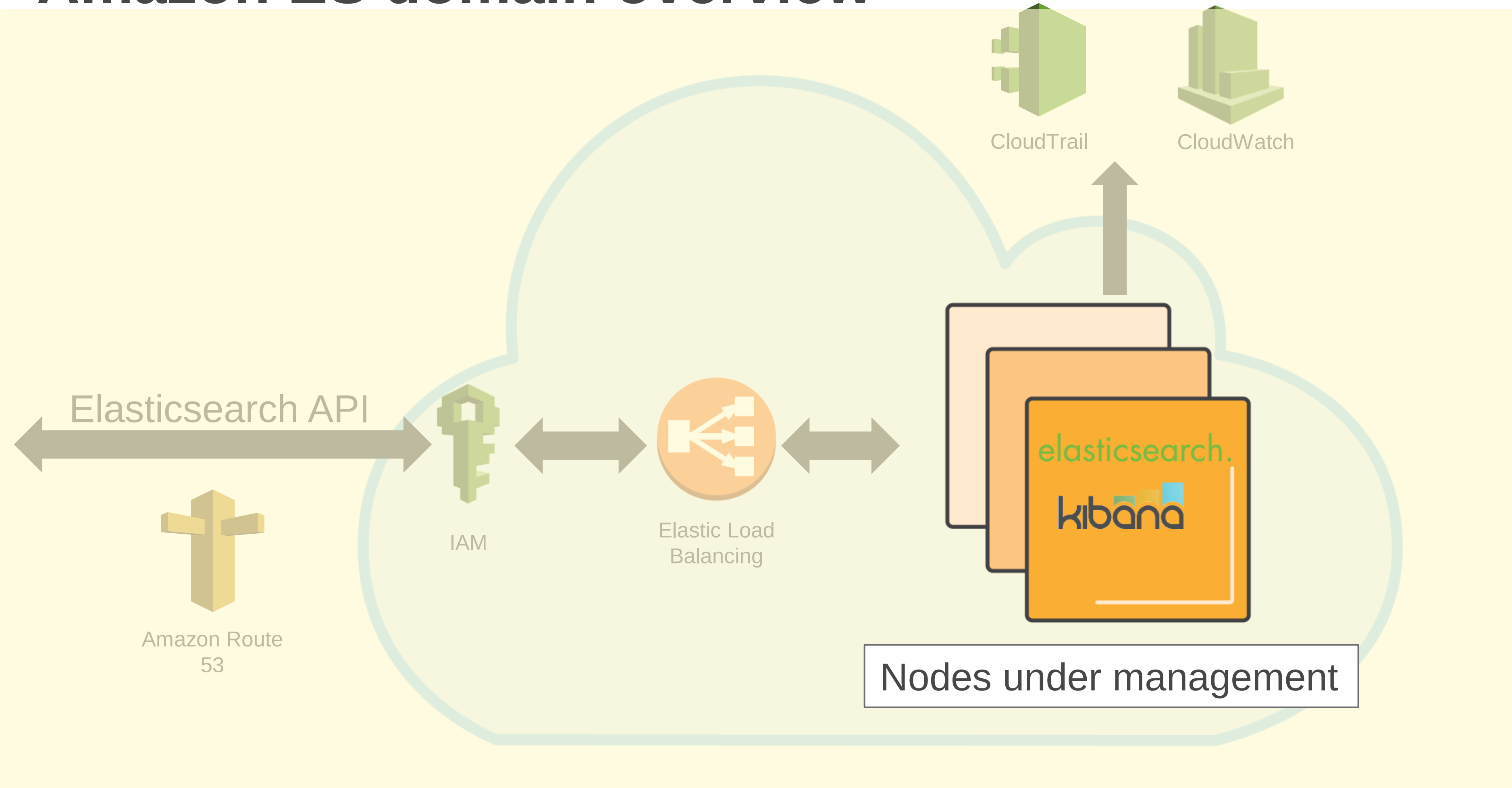
update-elasticsearch-domain-config

```
aws es create-elasticsearch-domain --domain-name my-domain
 --elasticsearch-cluster-config
   InstanceType=m3.xlarge.elasticsearch,InstanceCount=3
 --ebs-options
   EBSEnabled=true,VolumeType=gp2,VolumeSize=512
```

Amazon ES domain overview

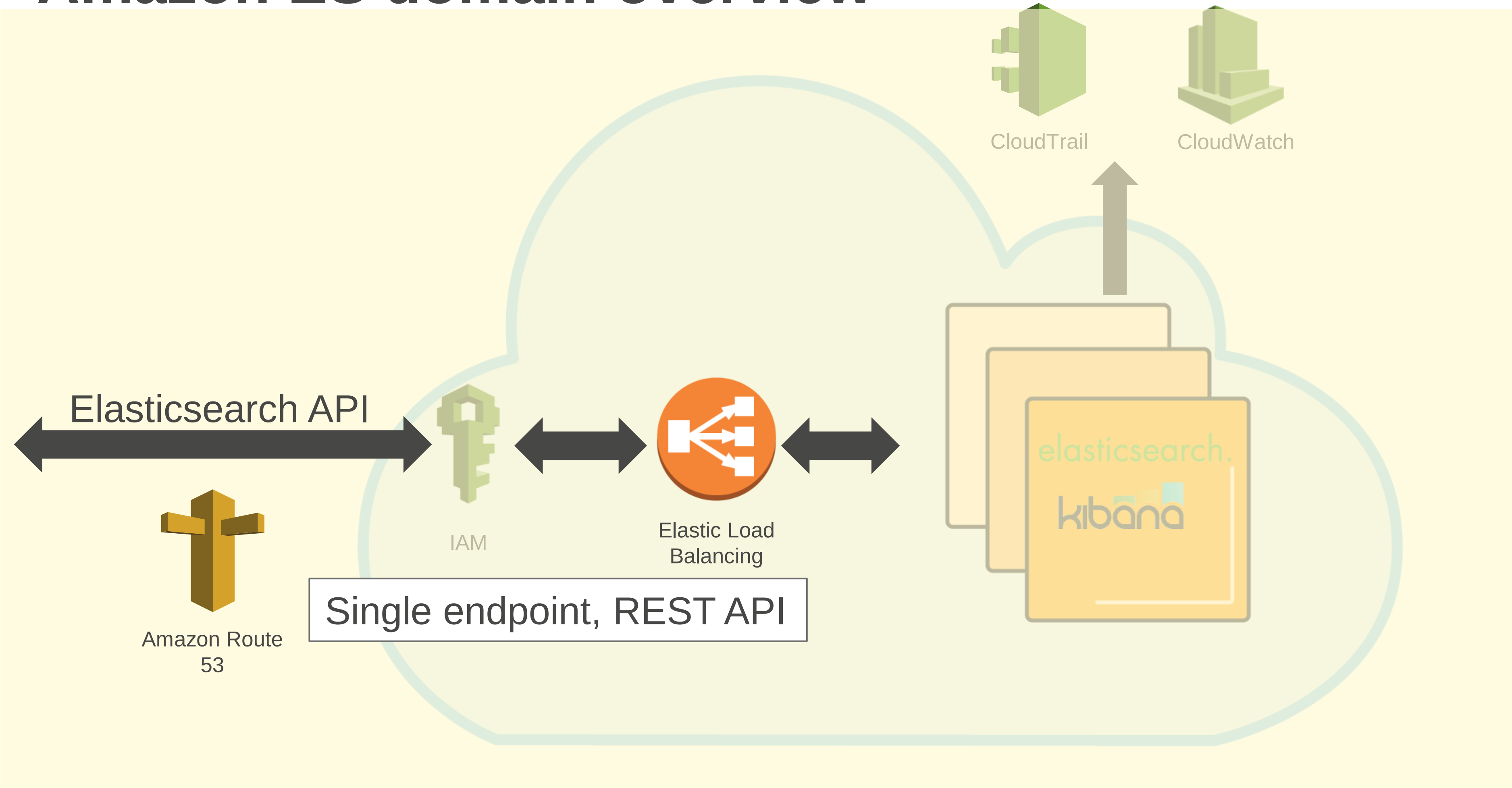


Amazon ES domain overview



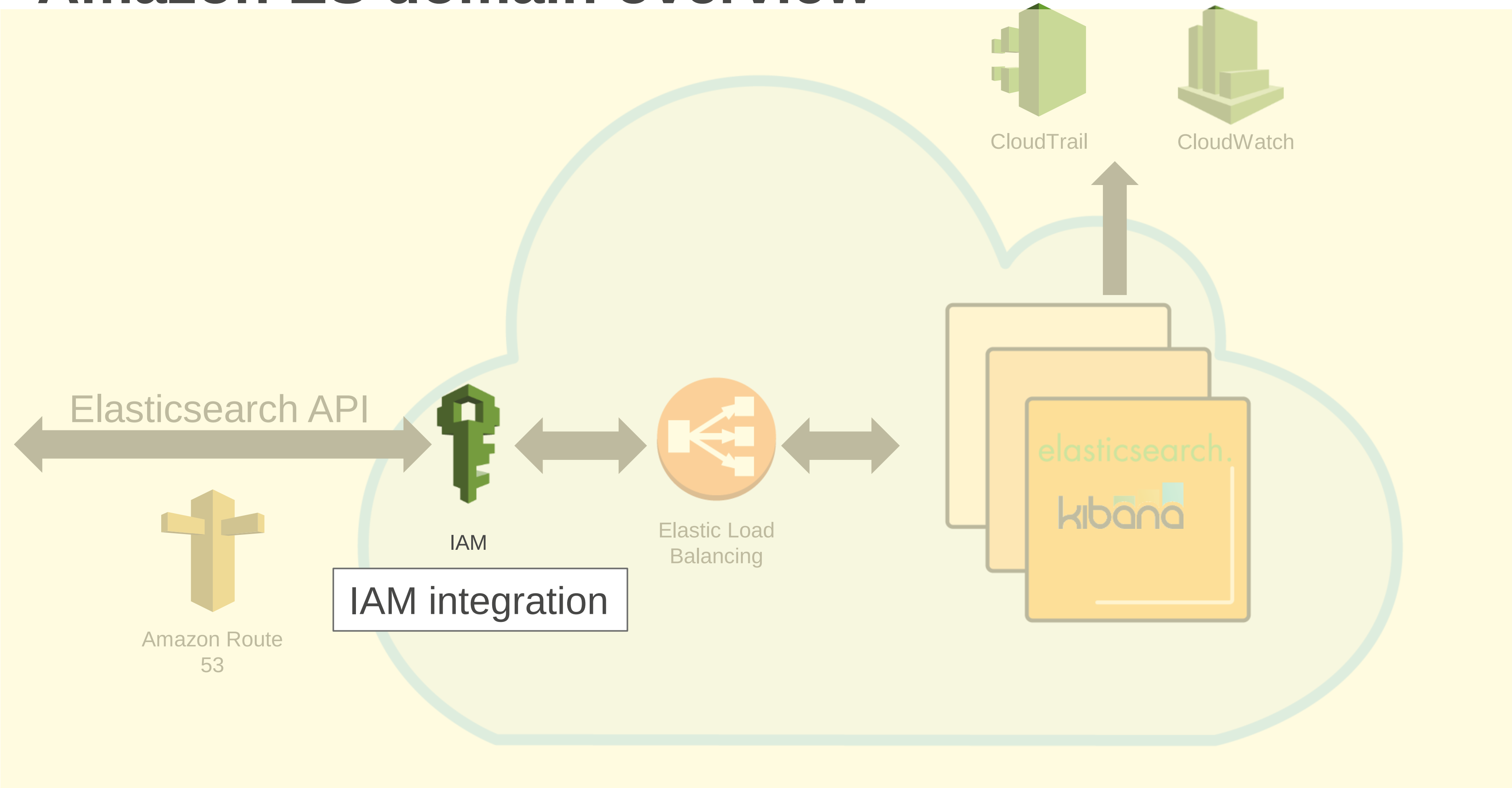
Nodes under management

Amazon ES domain overview



Single endpoint, REST API

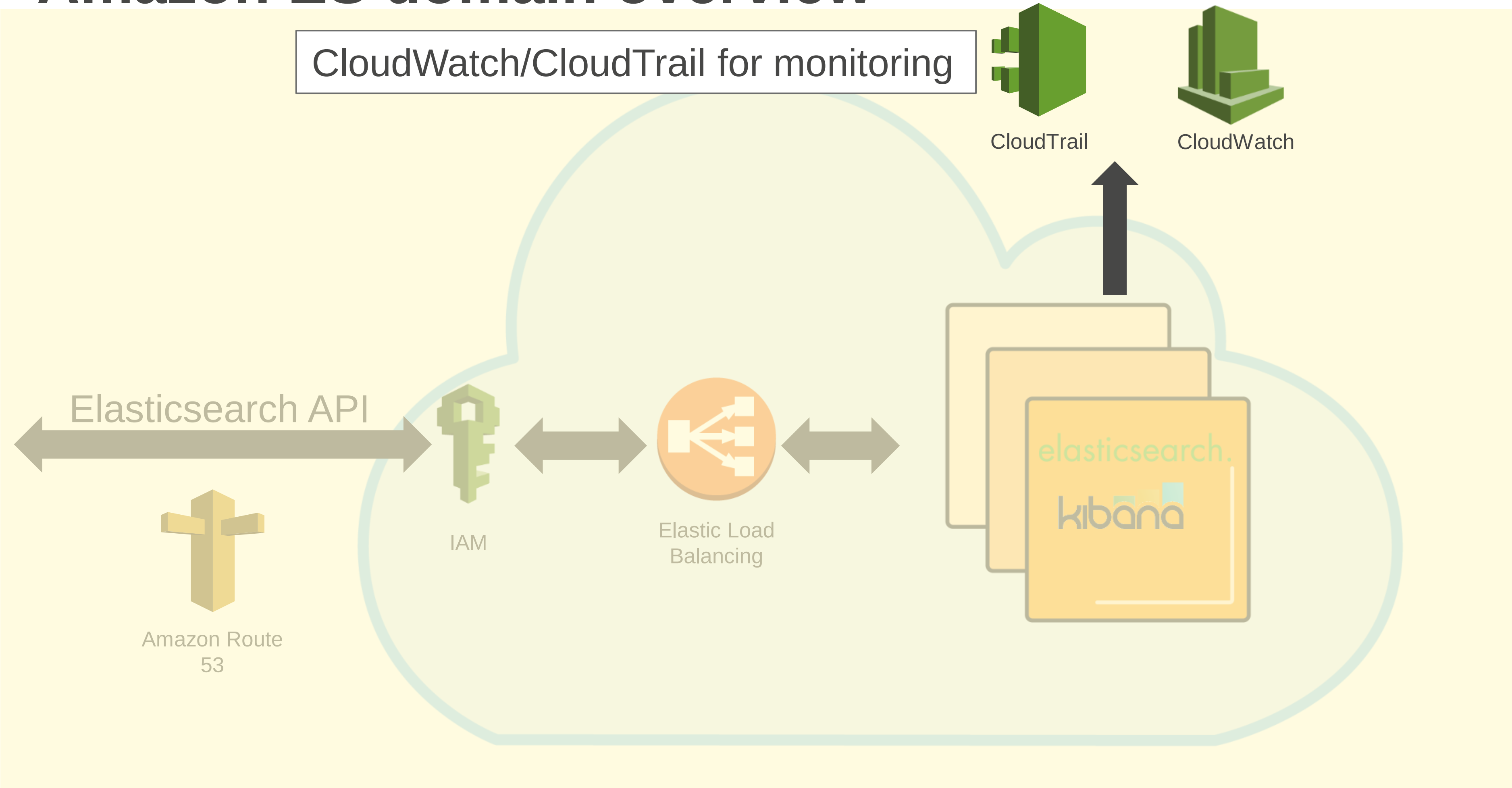
Amazon ES domain overview



IAM integration

Amazon ES domain overview

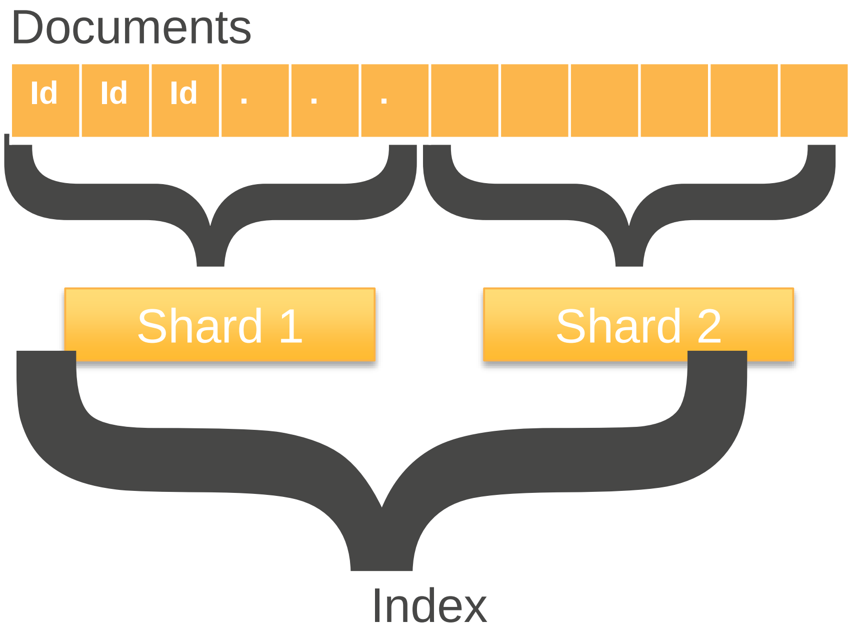
CloudWatch/CloudTrail for monitoring



Scale for your workload

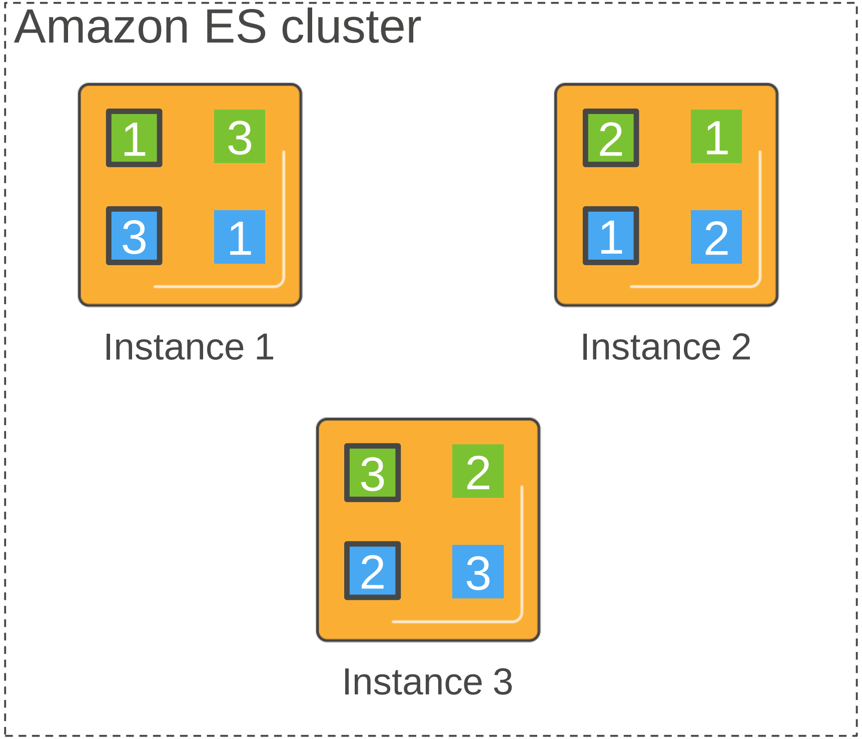
Data partitioning for search

- Document: The unit of search
- ID: Unique identifier, one per document
- Field: Documents comprise a collection of fields
- Shard: An instance of Lucene with a portion of an index
- Index: A collection of data



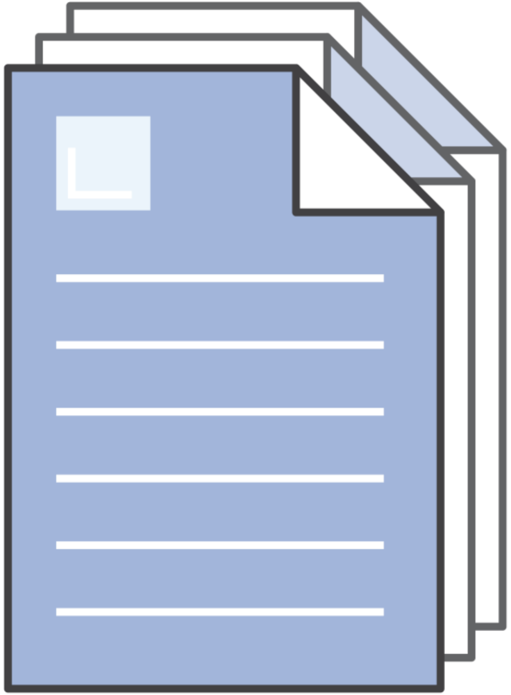
Deployment of indices to a cluster

	Primary	Replica
• Index 1		
• Shard 1	1	1
• Shard 2	2	2
• Shard 3	3	3
• Index 2		
• Shard 1	1	1
• Shard 2	2	2
• Shard 3	3	3



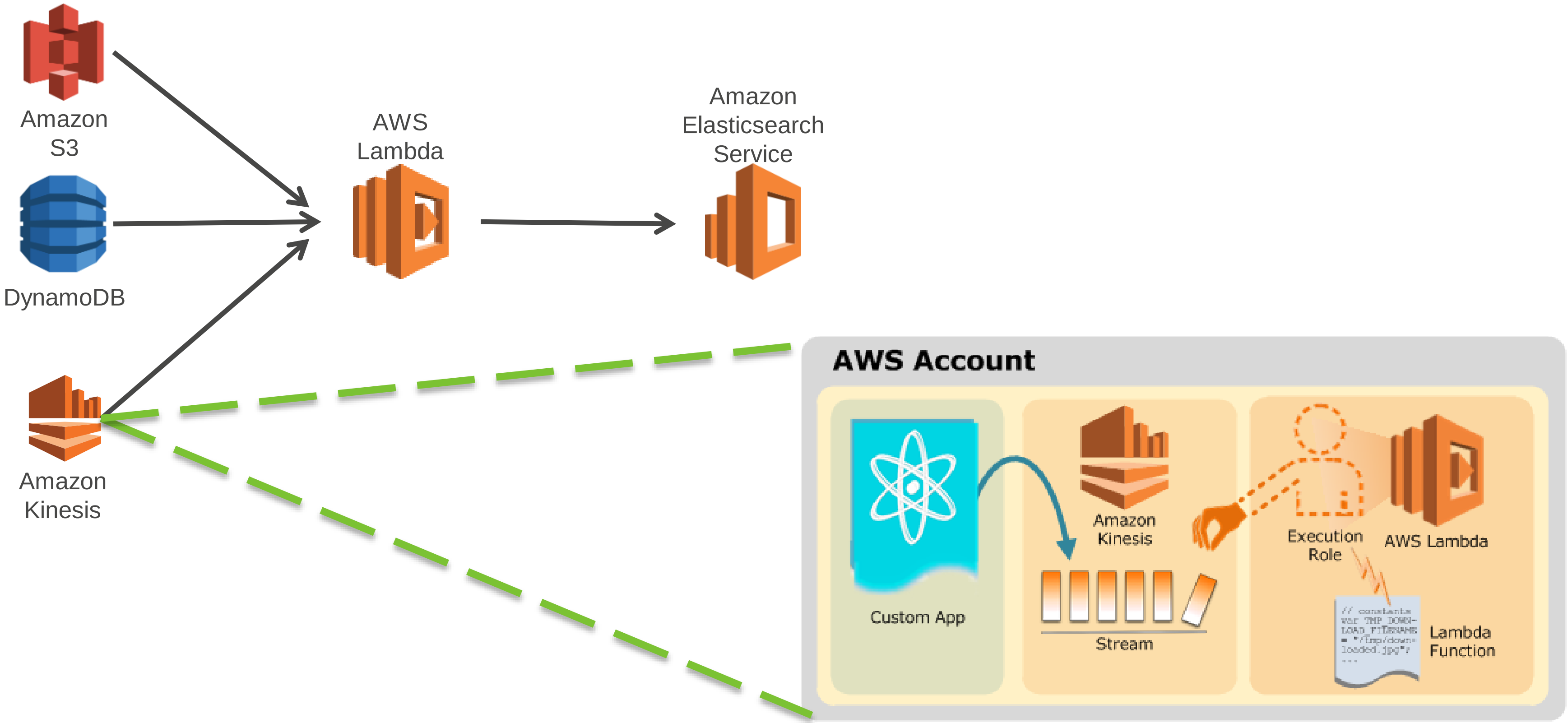
Instance type recommendations

Instance	Workload
T2	Entry point. Dev and test. OK for dedicated masters.
M3	Equal read and write volumes. Up to 5 TB of storage with EBS.
R3	Read-heavy or workloads with high query demands (e.g., aggregations).
I2	Up to 16 TB of SSD instance storage.



Load data

Loading data using Lambda



AWS Lambda Programming Model

Bring your own code

- Node.js, Java, Python
- Bring your own libraries (even native ones)

Simple resource model

- Select power rating from 128 MB to 1.5 GB
- CPU and network allocated proportionately
- Reports actual usage

Programming model

- AWS SDK built in (Python and Node.js)
- Lambda is the “webserver”
- Use processes, threads, /tmp, sockets normally

Stateless

- Persist data using Amazon DynamoDB, S3, or Amazon ElastiCache
- No affinity to infrastructure (can’t “log in to the box”)

Using AWS Lambda

Flexible use

- Call or send events
- Integrated with other AWS services
- Build whole serverless ecosystems

Flexible authorization

- Securely grant access to resources, including VPCs
- Fine-grained control over who can call your functions

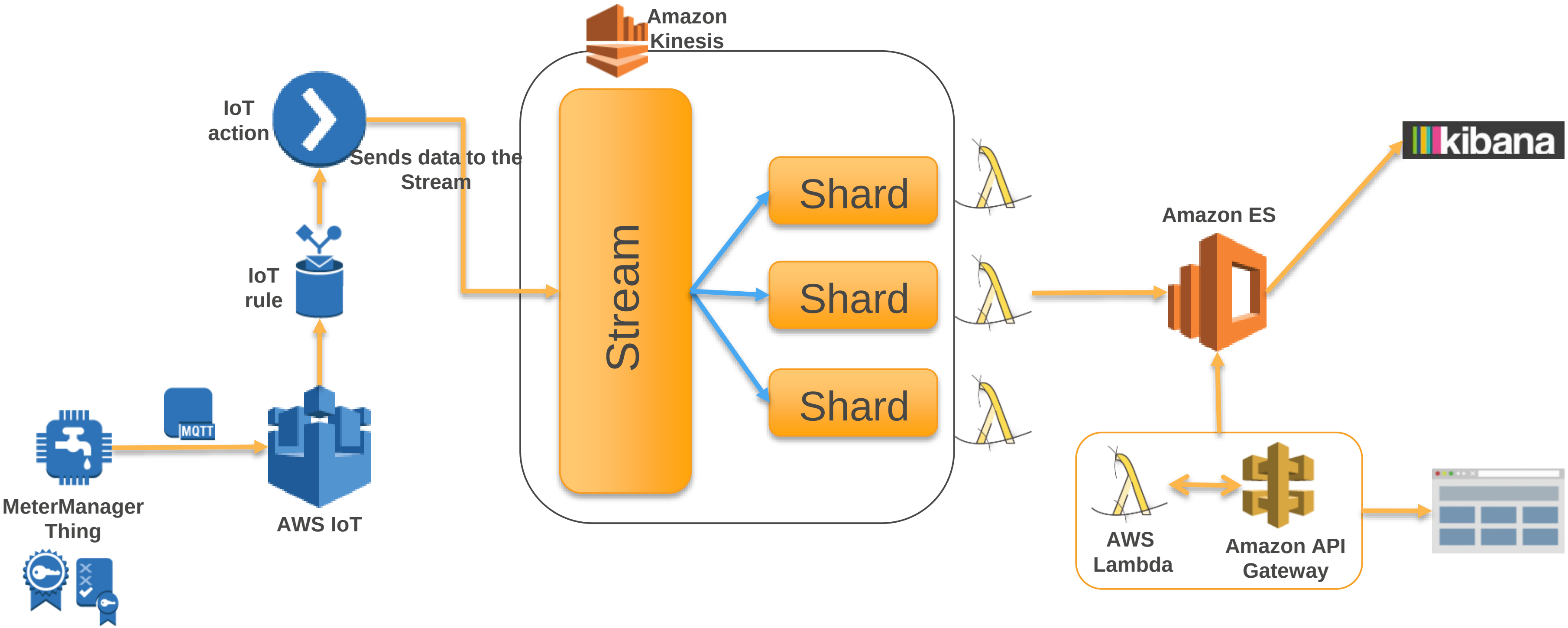
Authoring functions

- Author directly using the console WYSIWYG editor
- Package code as a .zip and upload to Lambda or S3
- Plugins for Eclipse and Visual Studio
- Command line tools

Monitoring and logging

- Built-in metrics for requests, errors, latency, and throttles
- Built-in logs in Amazon CloudWatch Logs

Zero Infrastructure, Real Time Data Collection and Analytics



AWS IoT with ThingWorx Analytics

thingworx®

What will we cover today?

1. Brief Overview of ThingWorx Platform

2. ThingWorx and AWS IoT Connector & Demo

3. ThingWorx Analytics Visualization Example & Demo

About the Speaker – Greg Urban

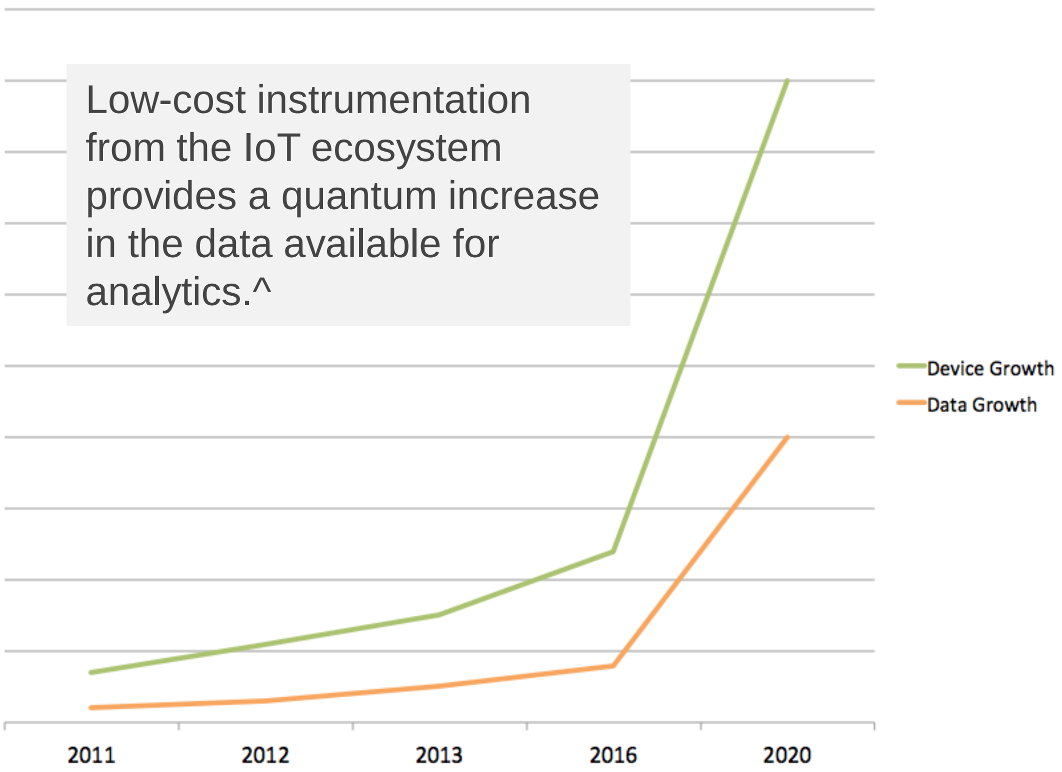
Director, Partner Engineering
Technical Platform Group
PTC

Greg leads a highly-talented team of engineers who work with partners and customers to develop effective, right-time analytics solutions for the Internet of Things (IoT).

He brings over a decade of experience in applied research and operational transformation when developing bespoke analytics solutions across multiple industry verticals including manufacturing, healthcare, marketing, energy, consumer products, telecom, transportation, etc.

Greg holds Masters degrees from Cranfield University and Villanova University, where he has also guest lectured on analytics.

IoT Device and Data Growth



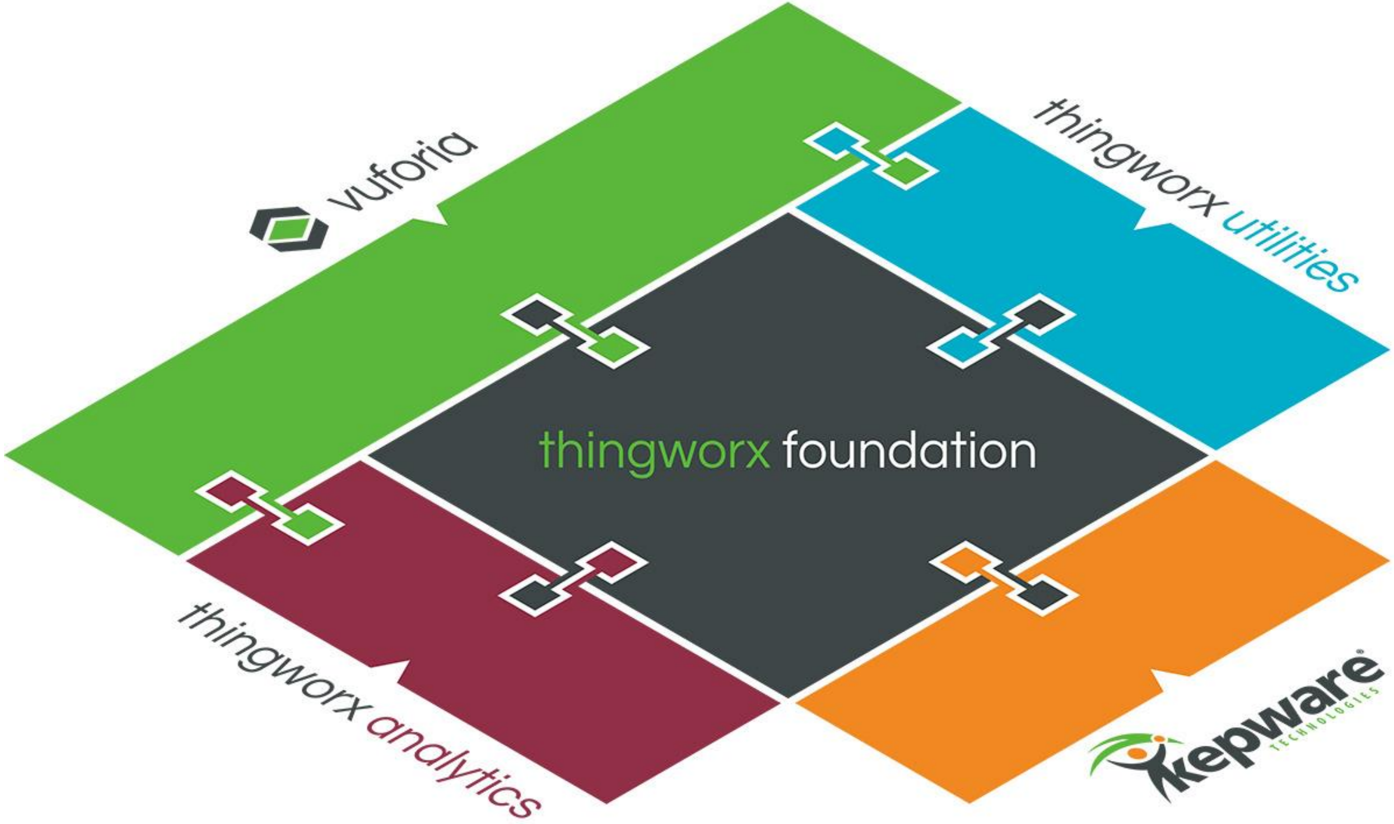
Low-cost instrumentation from the IoT ecosystem provides a quantum increase in the data available for analytics.^

50B Devices by 2020*

40 ZB of Data Created in 2020*

*Gartner & Iron Paper
^Practical Analytics

ThingWorx Platform



ThingWorx and AWS IoT Joint Solution

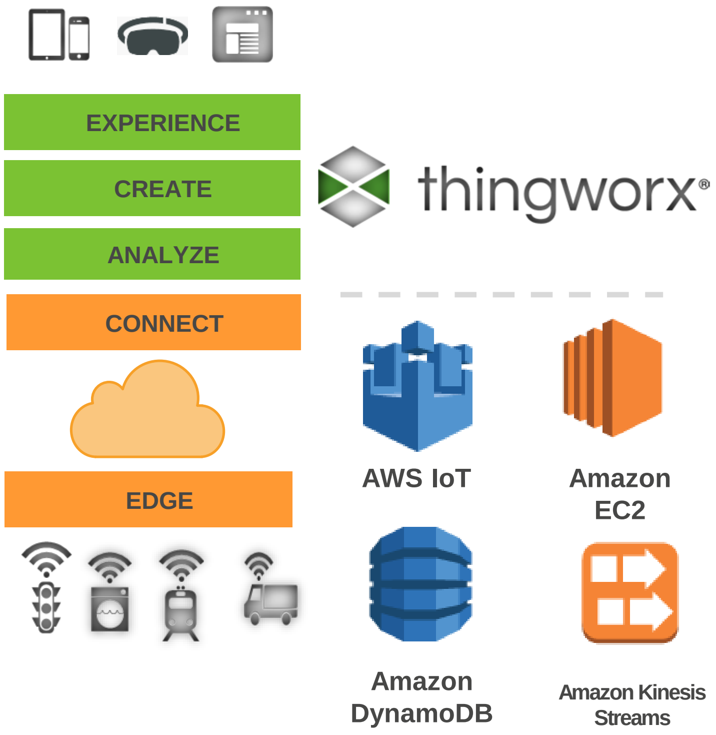
Solves two fundamental IoT business problems

1. Collect and Connect

- AWS IoT collects data from the edge into the cloud securely, at scale, and at a low cost
- AWS Cloud Services provides compute, storage, and security of your data

2. Interact

- ThingWorx uses data to analyze, create, and experience the IoT in a meaningful way.
- Contextually see and experience the digital data in the physical world through the power of Augmented Reality



EXPERIENCE

CREATE

ANALYZE

CONNECT

EDGE

thingworx®

AWS IoT

Amazon EC2

Amazon DynamoDB

Amazon Kinesis Streams

ThingWorx - AWS IoT Connector

AWS IoT

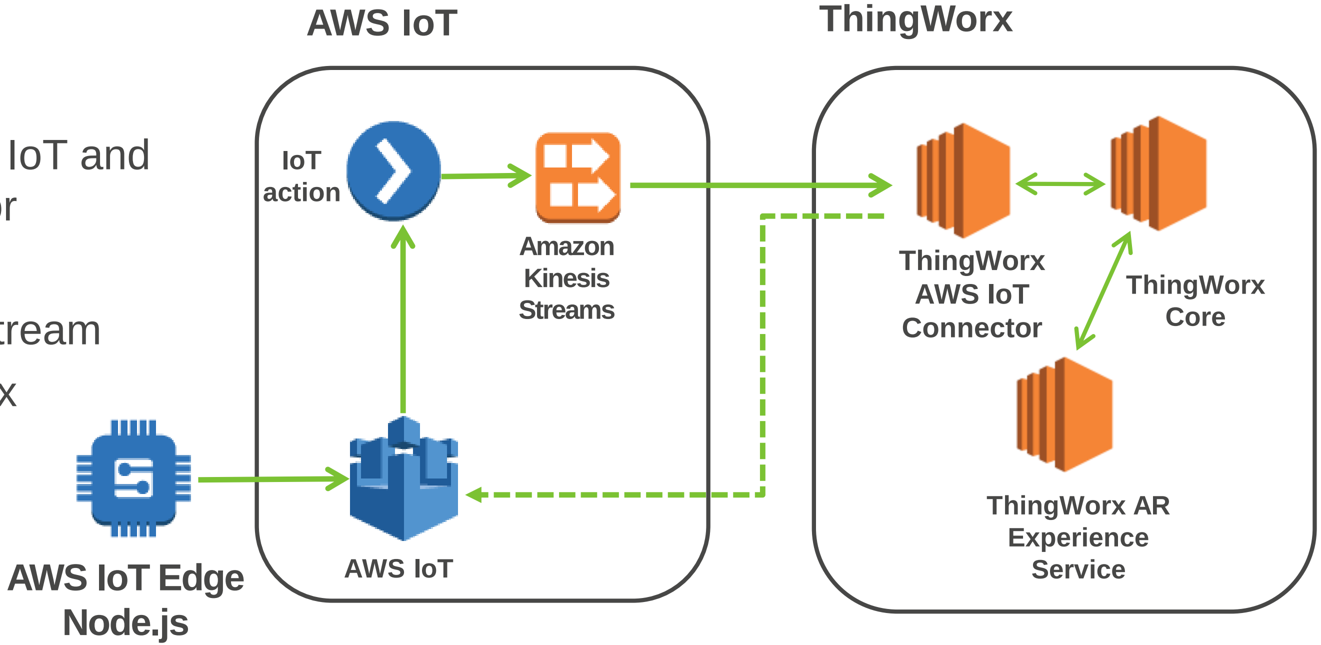
- Ingestion Layer
- Rule that forwards data to Amazon Kinesis

Amazon Kinesis

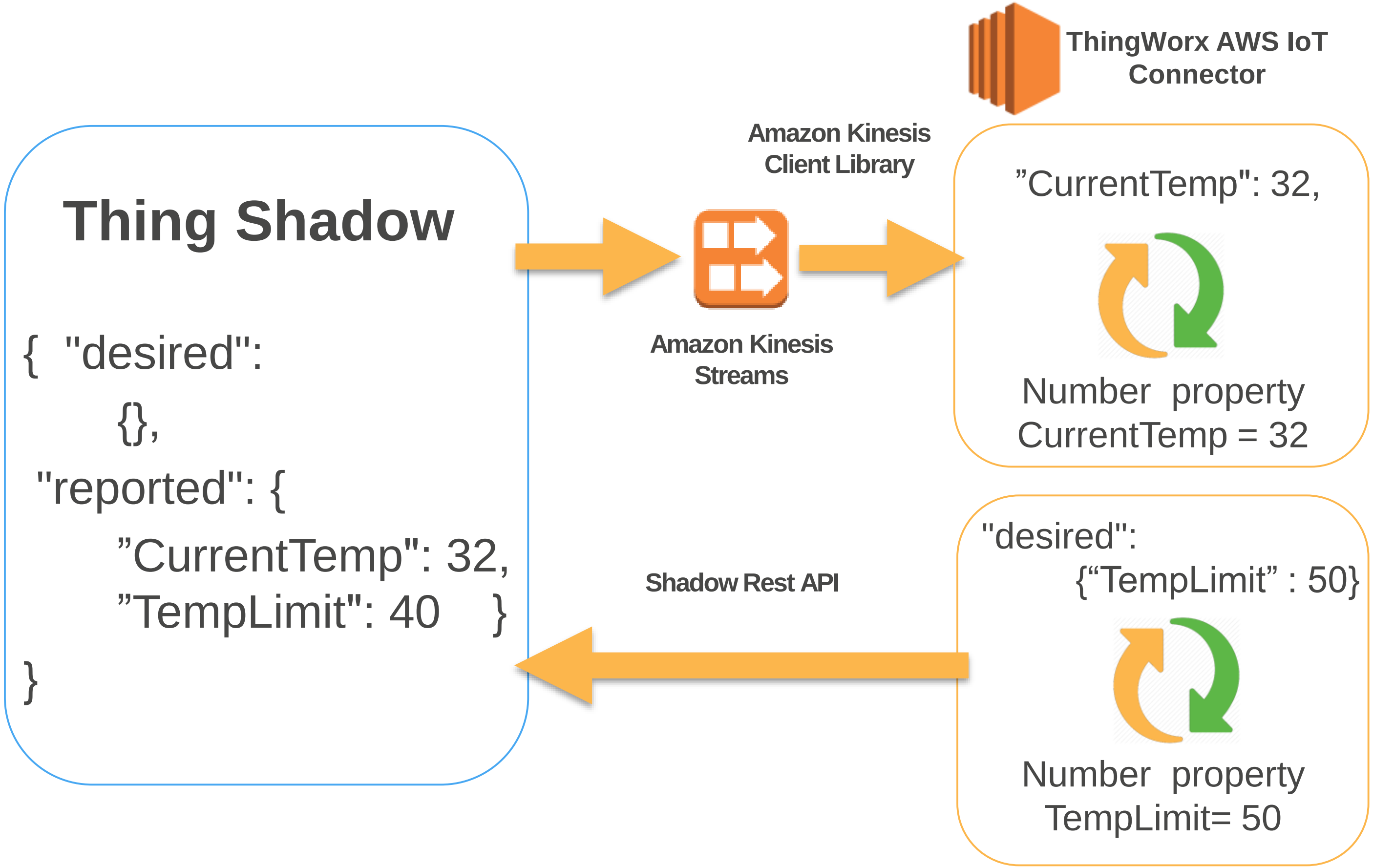
- Buffer between AWS IoT and ThingWorx Connector

ThingWorx Connector

- Pulls data from the stream
- Ingest into ThingWorx platform



Thing Shadow to ThingModel



ThingWorx AWS IoT
Connector

Thing Shadow

{ "desired":
{},
"reported": {
”CurrentTemp": 32,
”TempLimit": 40 }
}

Amazon Kinesis
Client Library

Amazon Kinesis
Streams

”CurrentTemp": 32,

Number property
CurrentTemp = 32

Shadow Rest API

"desired":
{“TempLimit” : 50}

Number property
TempLimit= 50

ThingModel

Properties:

Owner: John Smith

Warranty ID - 4352352

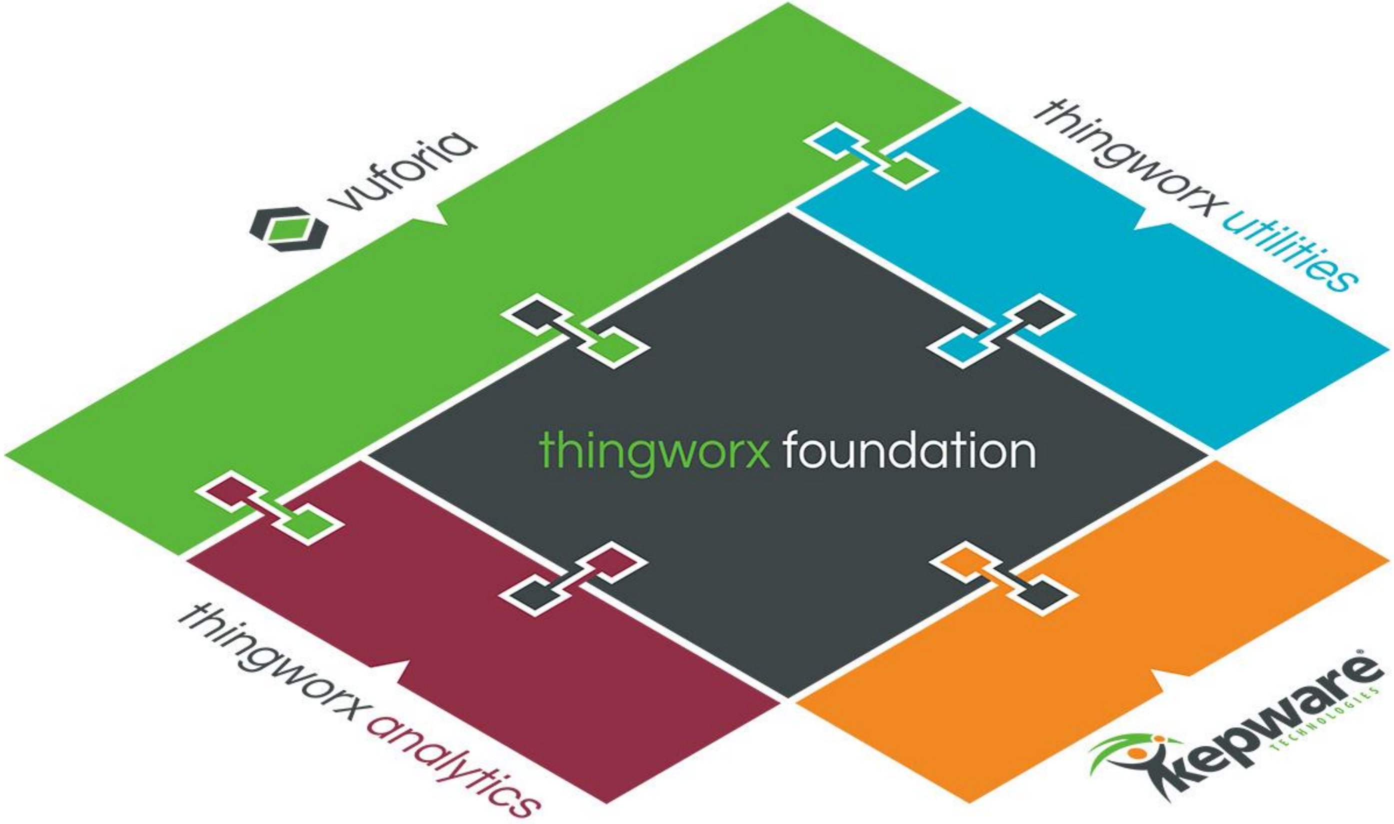
CurrentTemp - 32

TempLimit - 50

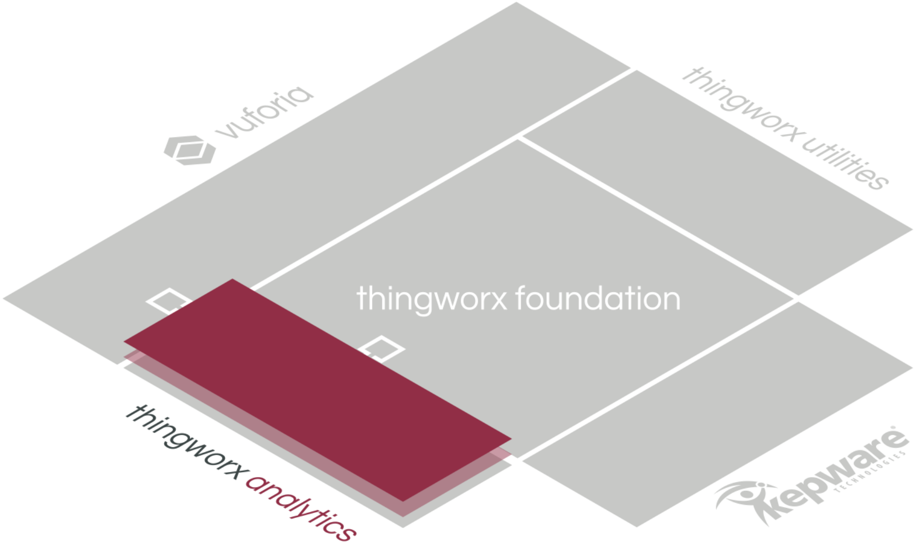
Demo time!

ThingWorx AWS IoT Connector Demo

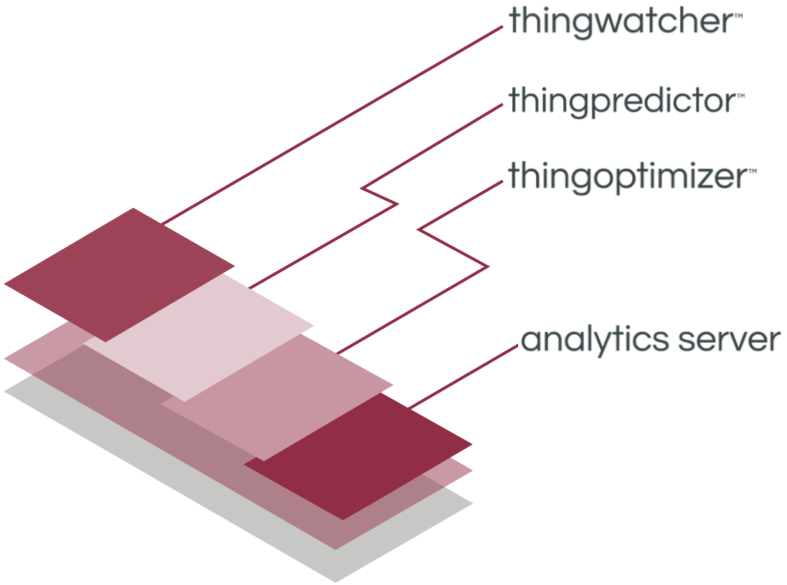
ThingWorx Platform



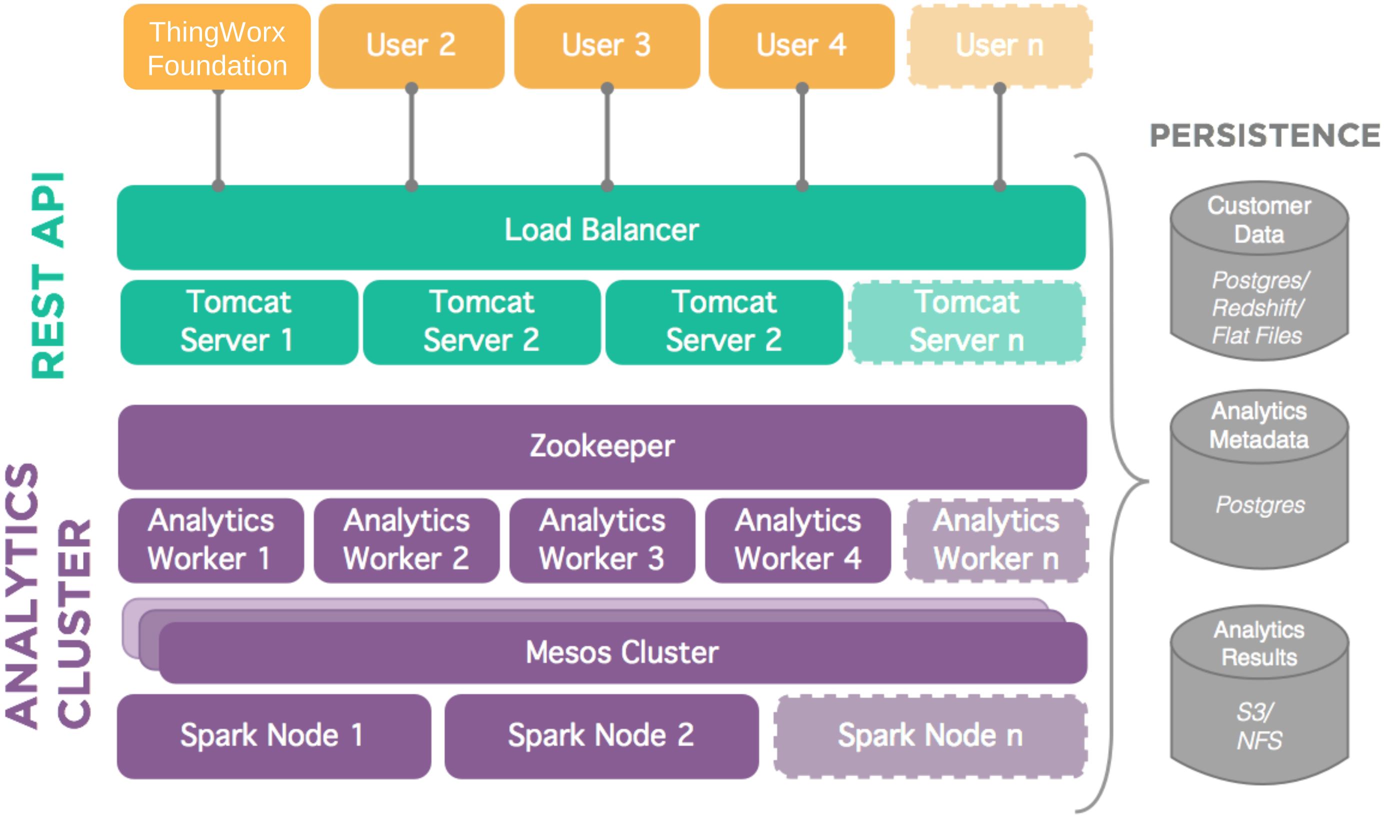
ThingWorx Analytics – built for IoT data



thingworx analytics



ThingWorx Analytics Server Architecture

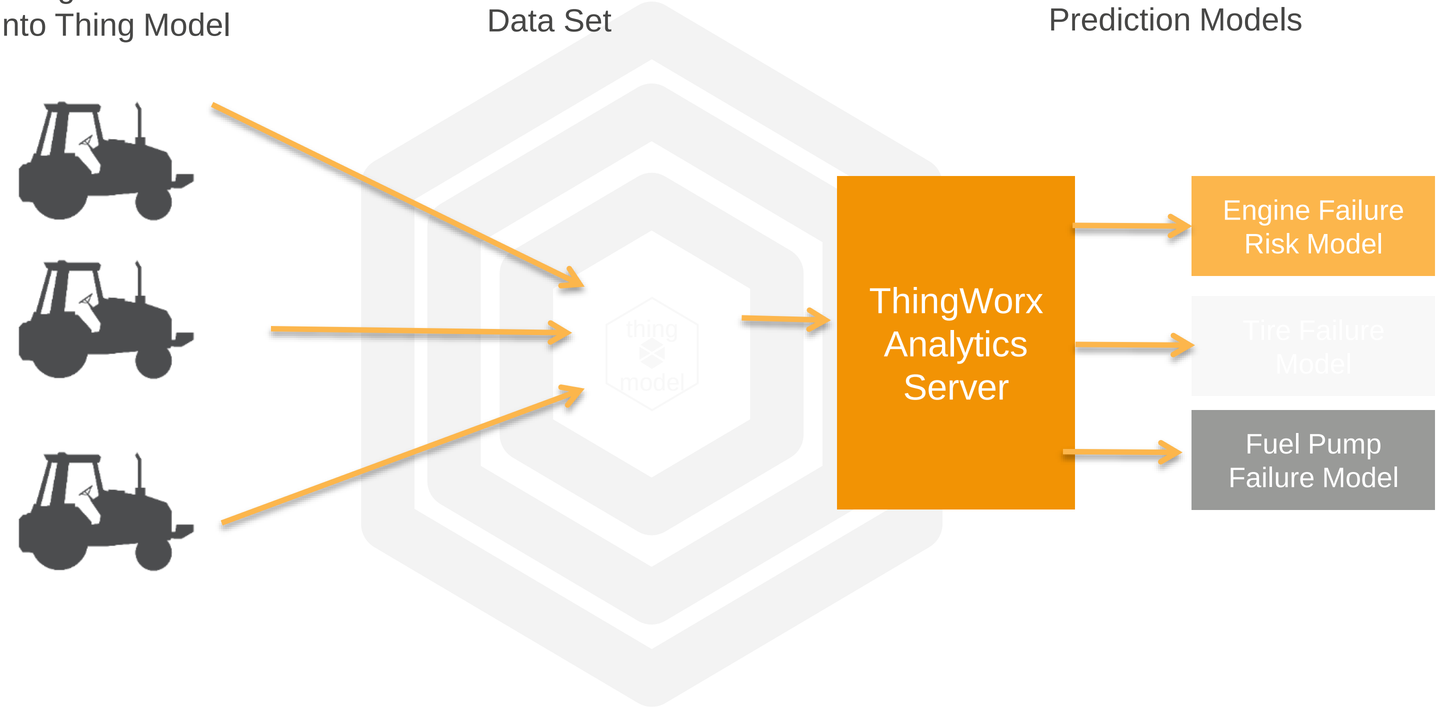


ThingModel integration to ThingWorx Analytics

Data Collected from Thing Sensors sent into Thing Model

Ingests Data from ThingModel into a Machine Learning Ready Data Set

ThingWorx Analytics Server Generates and Validates Prediction Models



ThingPredictor

Automatically build and validate predictive models without assistance from a statistician, using your Thing data as a learning source

Subscribe your “things” to one or more predicted outcomes (time to failure, future efficiency, etc.)

Real time or batch predictions (“scoring”)

Uses prediction models generated by ThingWorx Analytics Server or equivalent PMML-compliant prediction model generation tool

Things Subscribe to Outcome Prediction Models

	Engine Failure Risk Model	Tire Failure Model	Fuel Pump Failure Model
ID = 9090	5%	2%	7%
ID = 0773	**82%**	**82%**	**82%**
ID = 4242	12%	2%	**72%**
ID = 1101	32%	13%	6%
ID = 9993	5%	**72%**	6%

Each ‘Thing’ gets a customized and “personal” set of predictions based on its individual sensor readings and environmental conditions data.

Demo - Bean Pro Espresso

About the Company:

- Manufacturer of connected custom espresso machines.
- Customers include chains, medium-sized shops, and storefront operations.
- Bean Pro Espresso sells and services their equipment directly.
- Key differentiator – constant connectivity of their machines theoretically limit downtime for operators and therefore minimize the risk of lost revenue due downtime due to malfunction or extensive repairs.

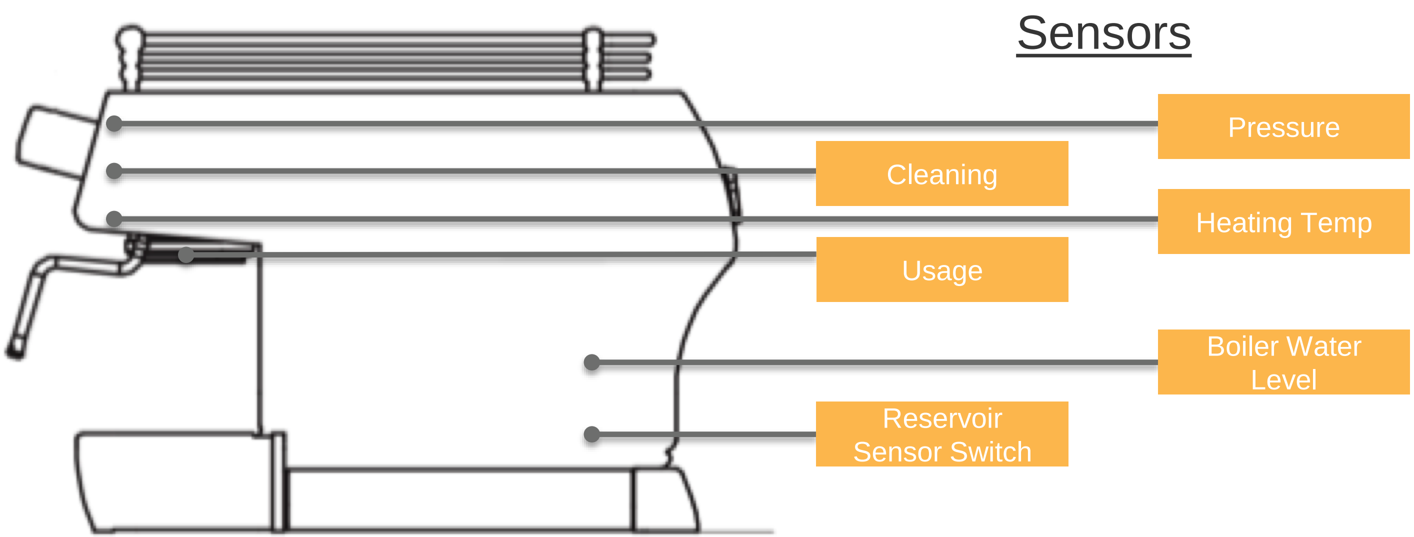
Challenge:

- Machines are experiencing downtime causing operator customer service issues.
- Operators always desire to avoid or minimize downtime as it directly impacts their revenue and customer satisfaction.
- While connected data is being monitored, it isn’t being used for predictive analysis.
- Service managers and technicians need quicker ways to implement fixes for both current and future issues.

Bean Pro’s machines

bean PRO ESPRESSO

Sensors



Other Data

- Machine Characteristics
- Fault Codes
- Service Requests
- Alert Codes & Urgency
- Technician Data
- Repair Hours
- Location

Demo time!

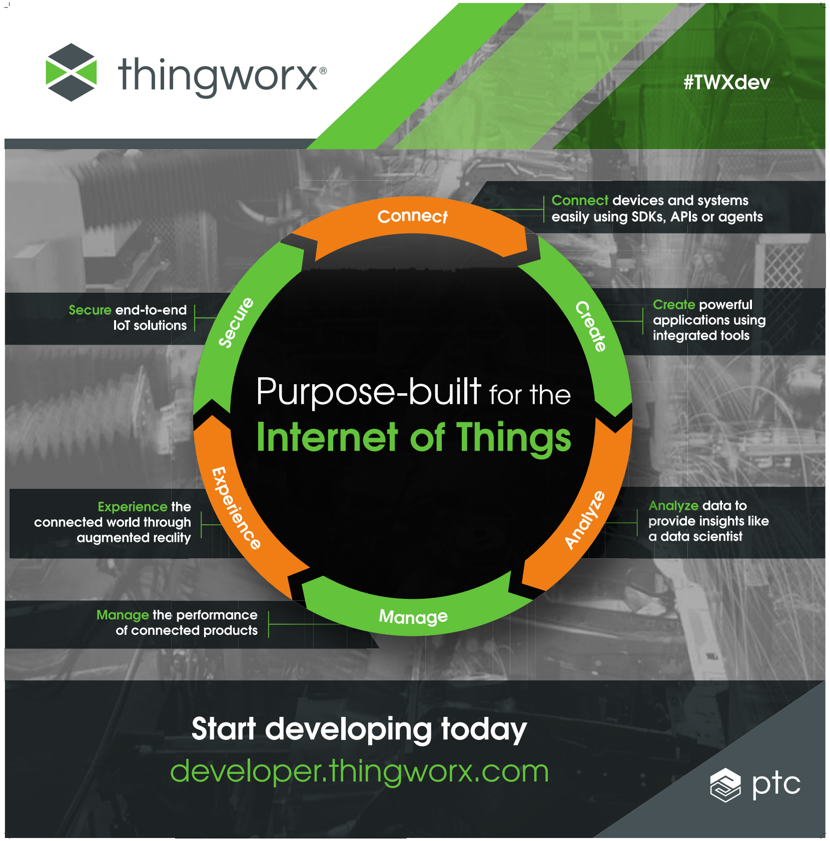
Bean Pro Espresso Demo

Bean Pro Results

By using the ThingWorx platform, smart connected product manufacturers and operators are able to:

- Understand critical predictors of various machine failures to improve service plans and future products.
- Shift their machine service strategy to be proactive and keep operator facilities running smoothly.
- Enhance the manufacturing processes to improve upon faulty processes and parts from suppliers.
- Educate technicians and operators to understand how to better service each individual machine to prevent predicted failures.
- Share services best practices amongst the operator community based on usage conditions, real time monitoring and other dynamic factors

GO TO Developer.ThingWorx.com



thingworx®

#TWXdev

Connect devices and systems easily using SDKs, APIs or agents

Create powerful applications using integrated tools

Analyze data to provide insights like a data scientist

Manage the performance of connected products

Experience the connected world through augmented reality

Secure end-to-end IoT solutions

Purpose-built for the Internet of Things

Start developing today

developer.thingworx.com

ptc

Now let’s see this in action!

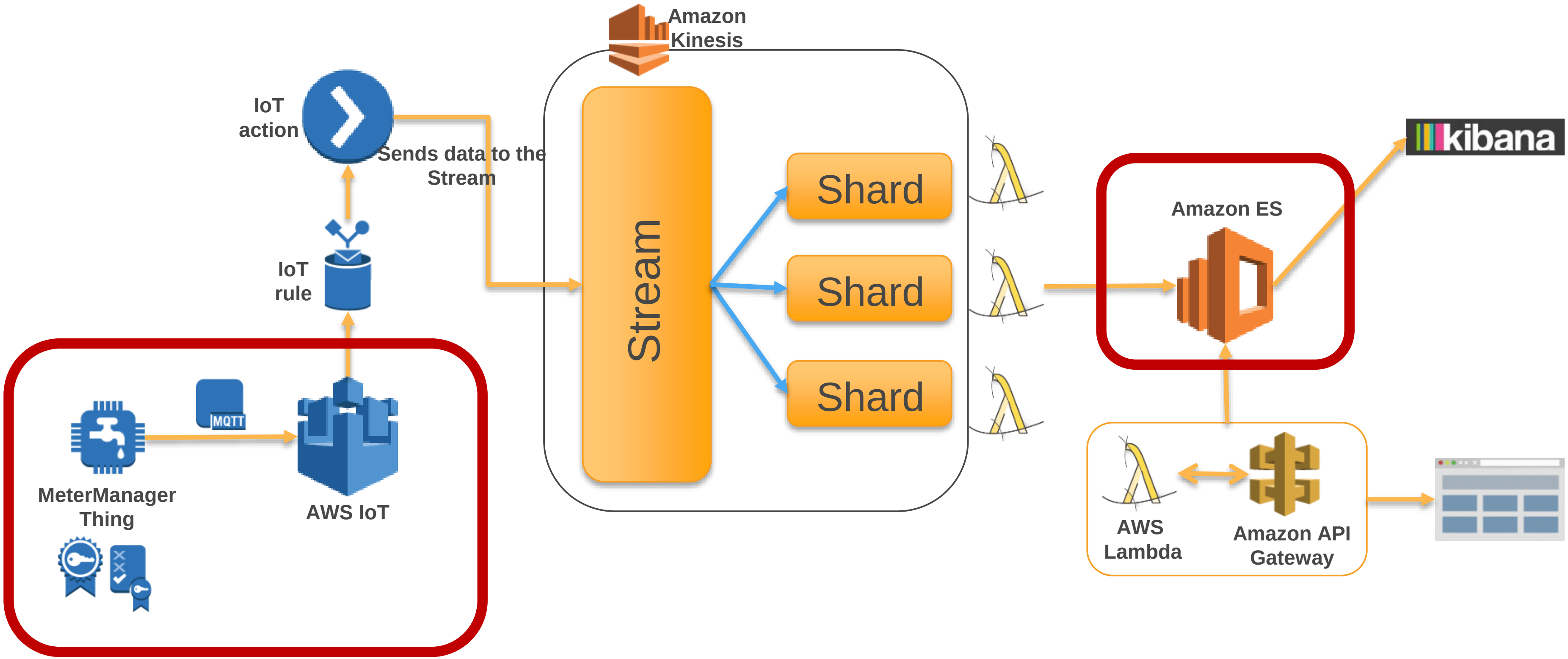
Workshop Prerequisites

- AWS Account
- AWS CLI installed on your machine
- Familiarity using the AWS Management Console, AWS Services, and the AWS CLI
- Download prerequisites from S3:
    - http://bit.ly/2gxOMAM

Part I: Device Data -> AWS IoT

- Create the Elasticsearch cluster
- Create a thing in AWS IoT
- Create and associate security credentials for the thing
- Send data to AWS IoT

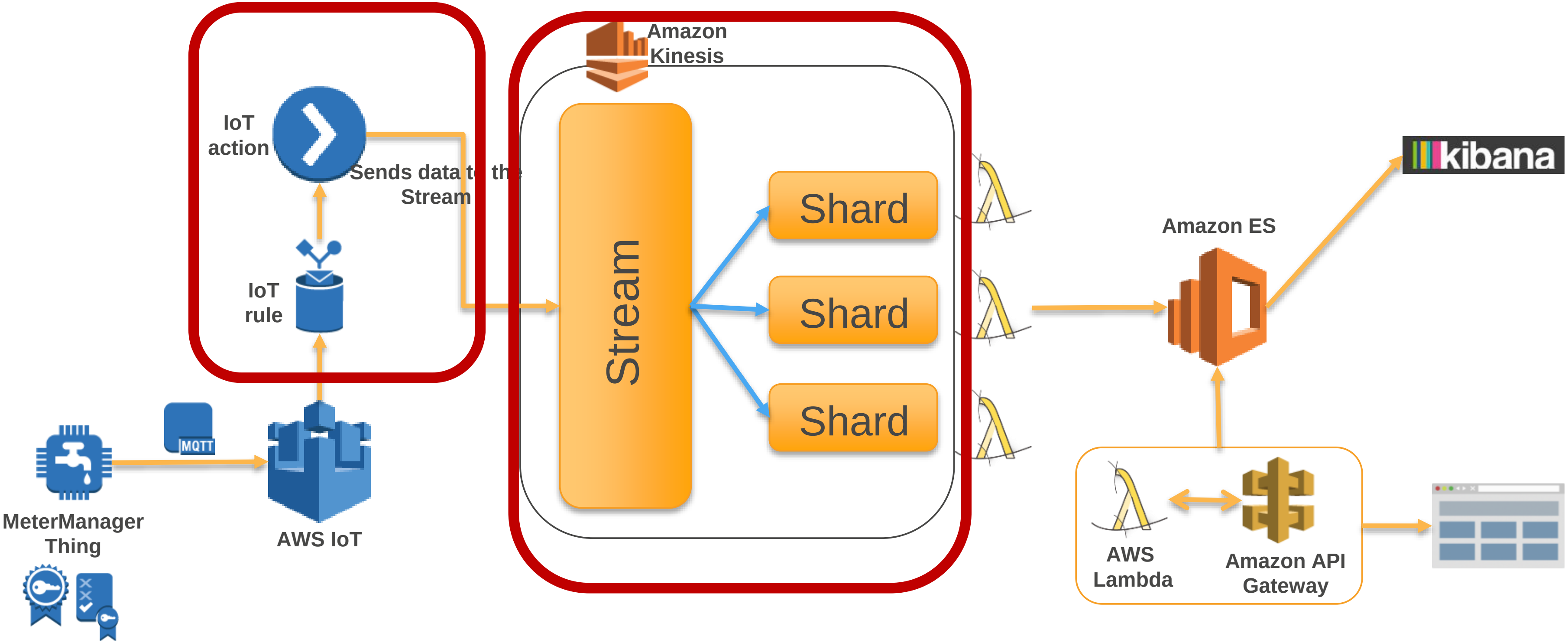
Part I: Device Data -> AWS IoT



Part II: AWS IoT -> Amazon Kinesis

- Create the Amazon Kinesis Stream
- Create an IAM role in IAM to allow AWS IoT to put records onto the Stream
- Create an IoT rule in AWS IoT
- Send data to AWS IoT and observe data PUTs on the Stream

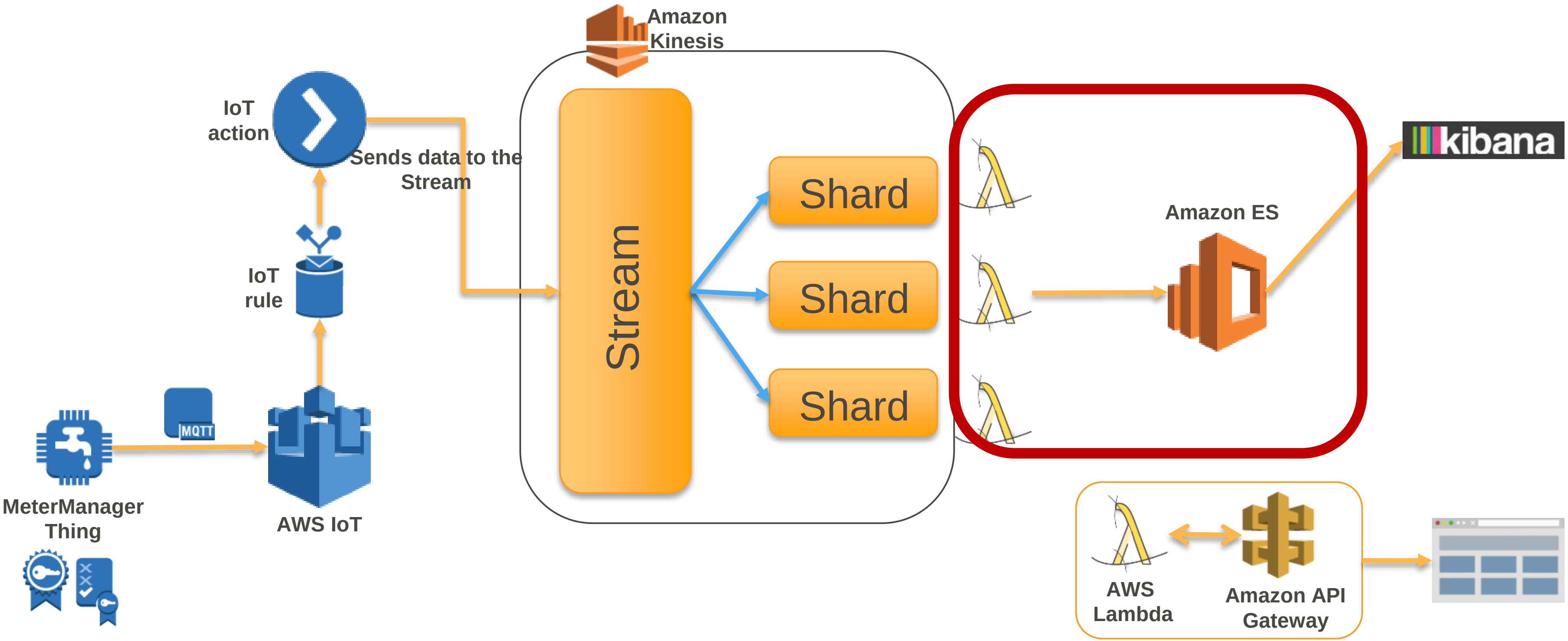
Part II: AWS IoT -> Amazon Kinesis



Part III: Amazon Kinesis -> Amazon ES (via AWS Lambda)

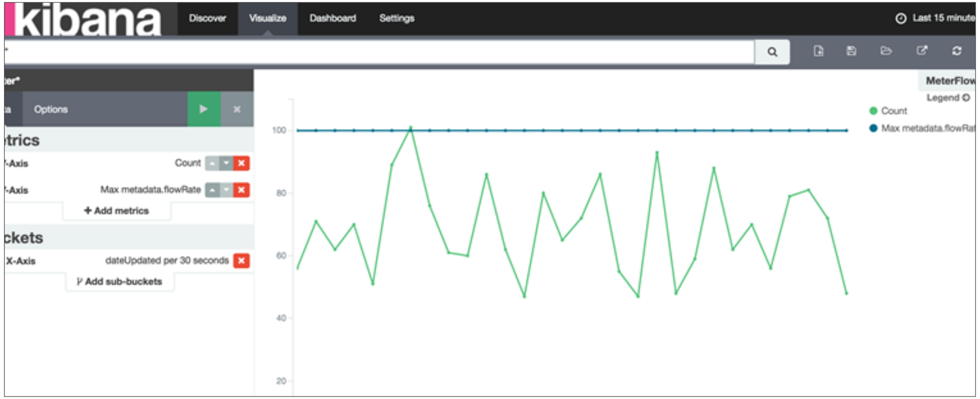
- Configuration for Lambda function
- Create the Lambda function
- Create an index in Elasticsearch
- View records appearing in Elasticsearch

Part III: Amazon Kinesis -> Amazon Elasticsearch (via AWS Lambda)



Part IV: Data Visualization with Kibana

- Click on the Kibana link from the Amazon ES console
- Create graphs for visualization of meter data



trics

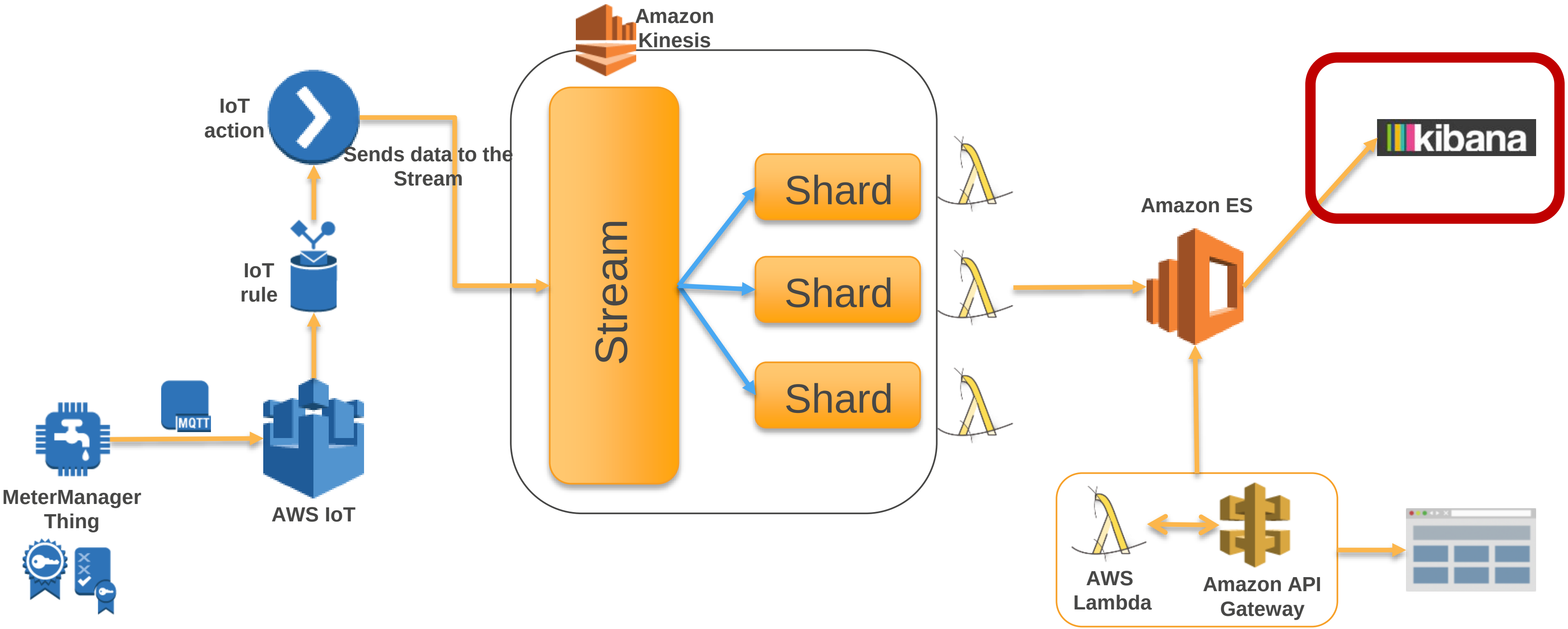
-Axis Count

-Axis Max metadata.flowRate

ckets

X-Axis dateUpdated per 30 seconds

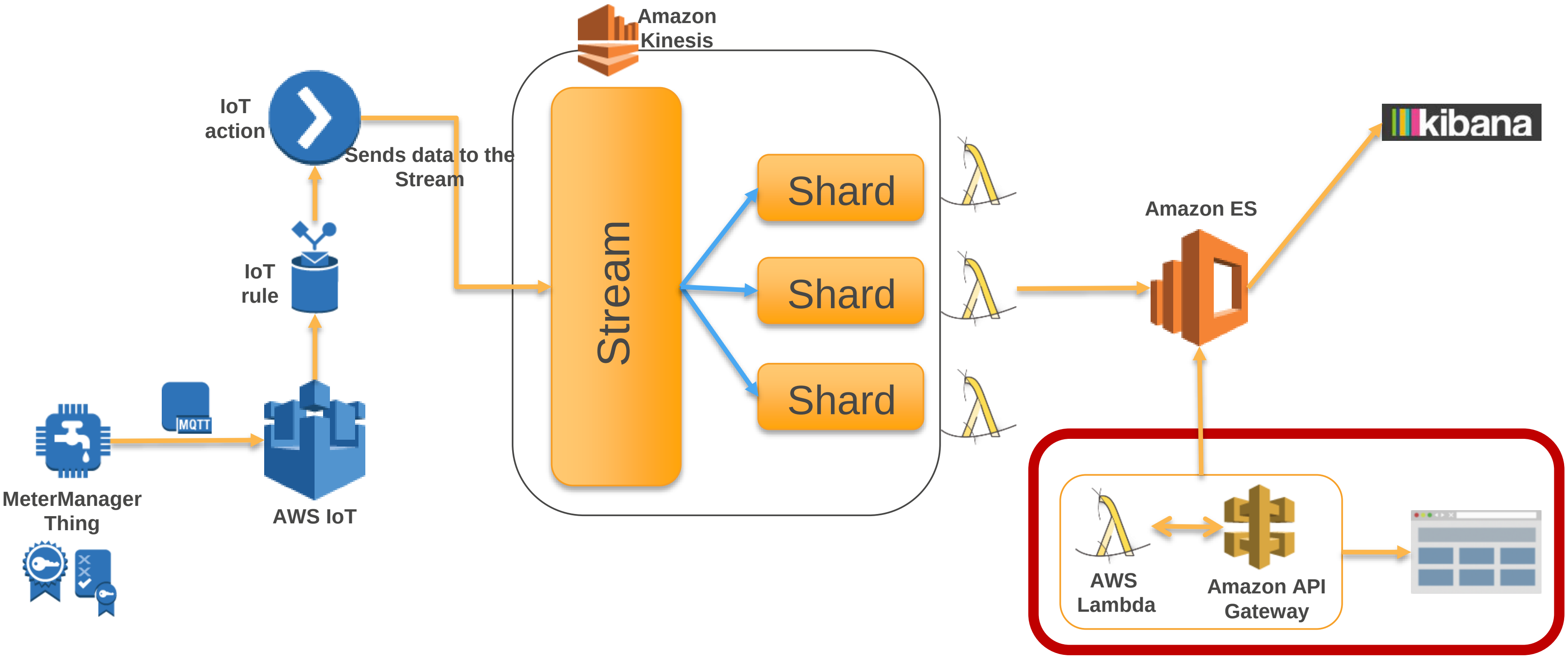
Part IV: Data Visualization with Kibana



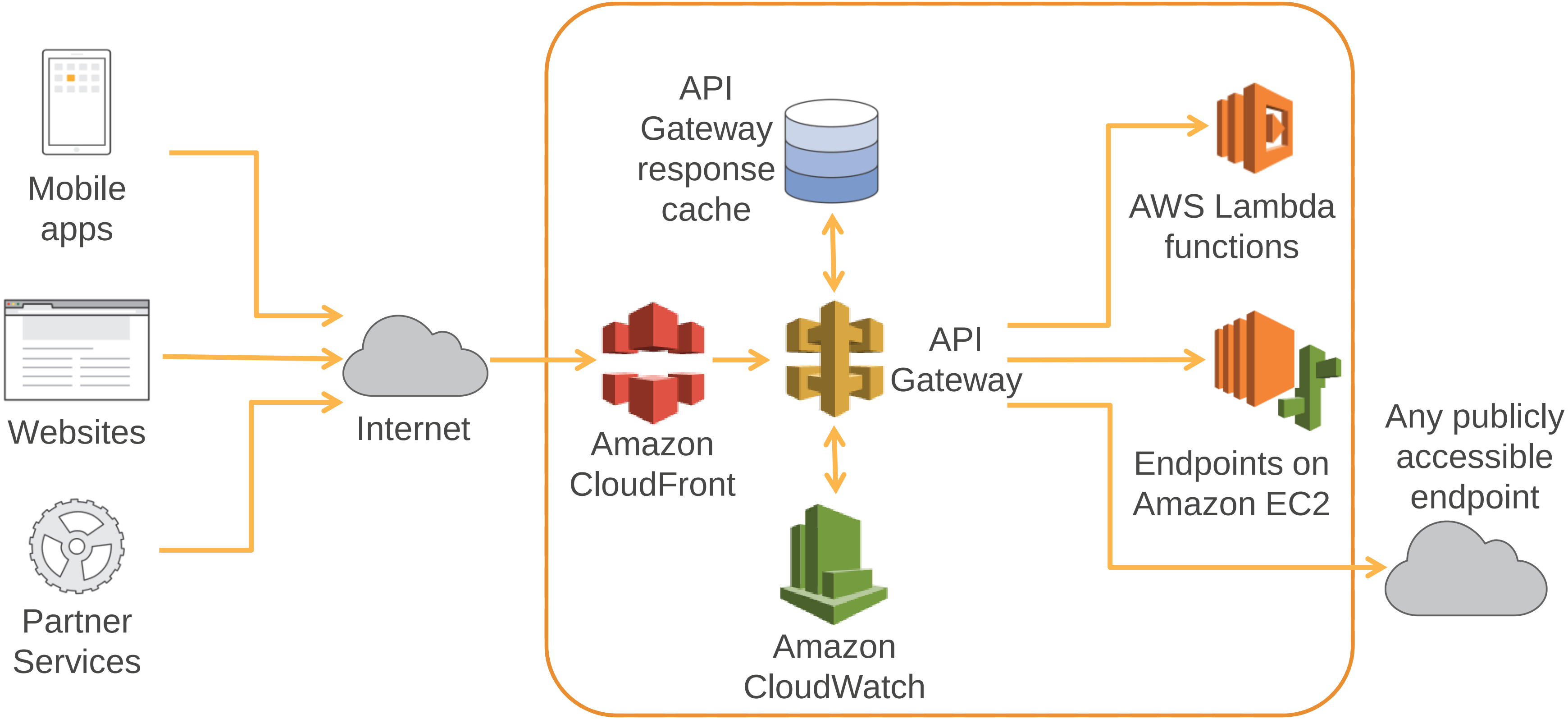
Part V: Custom Visualization with Amazon API Gateway and AWS Lambda

- Configuration for Lambda function
- Create the Lambda function
- Create API Gateway endpoint
- Test and view results (browse to localhost:3000 after running node app)

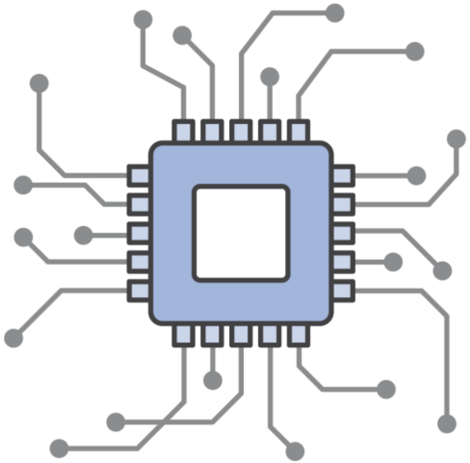
Part V: Custom Visualization with Amazon API Gateway and AWS Lambda



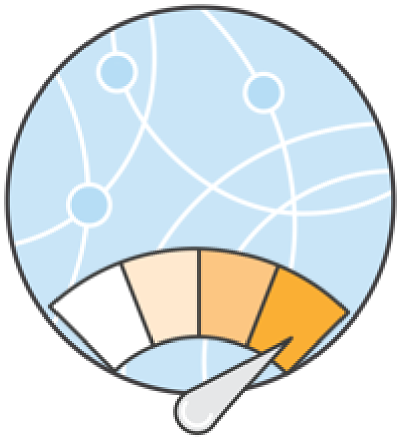
Amazon API Gateway: Serverless APIs



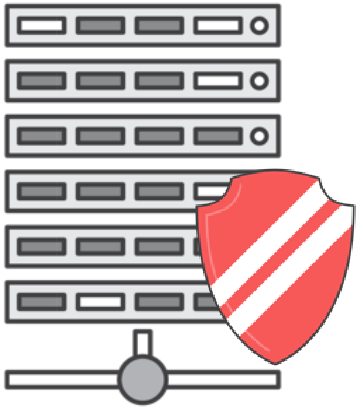
Benefits of Amazon API Gateway



Create a unified API front end for multiple microservices



DDoS protection and throttling for back-end systems



Authenticate and authorize requests

Summary

- In less than 2 hours, you built a near-real time data analytics and visualization workflow!
- Six services used:
  - AWS IoT
  - Amazon Kinesis
  - Amazon Elasticsearch
  - AWS Lambda
  - API Gateway
  - AWS IAM
- All code is yours to take and build upon.

AWS re:Invent

Thank you!

amazon web services

Remember to complete your evaluations!