Automated Serverless Security Testing

Tal Melamed | Senior Director
Cloud Native Security Research

CONTRAST SECURITY

```
user@host:~
```


4ppsec

@ 4ppsec

talmelamed

tal.melamed@qu.edu

tal@appsec.it

cloudessence

Co-Founder & CTO

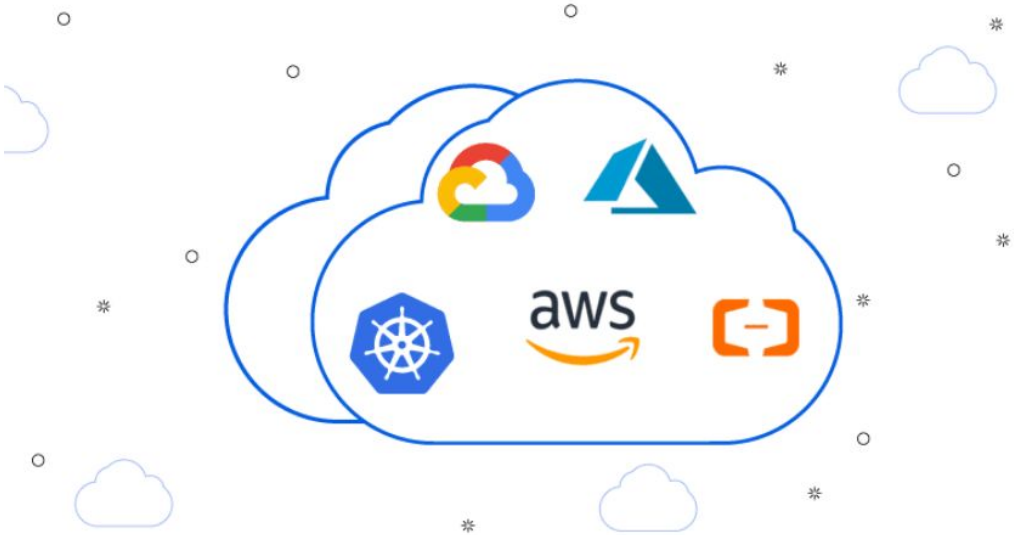
Acquired by Contrast Security, 2020

Protego

Head of Security Research

Acquired by CheckPoint, 2019

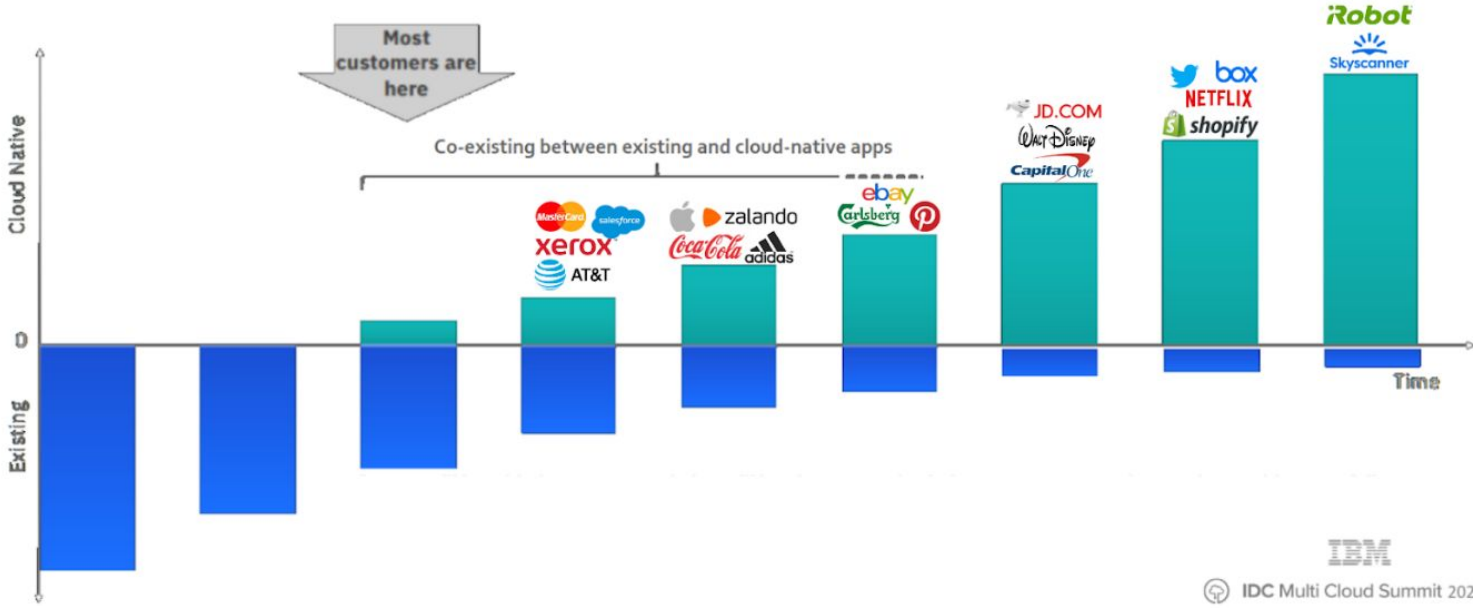
Cloud Native is the **Future** of Application Development



“25% of developers will use serverless regularly by the end of 2021”

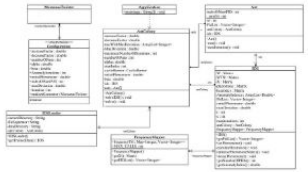
FORRESTER®

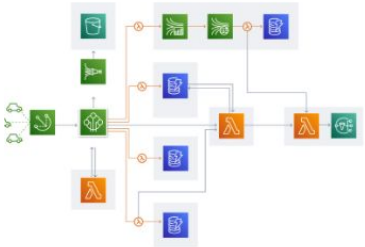
The **Cloud Native Transformation** Has Begun



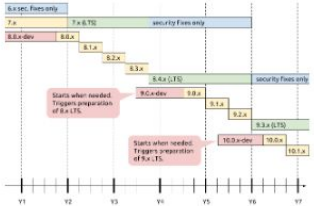
It not just a development pattern

Architecture

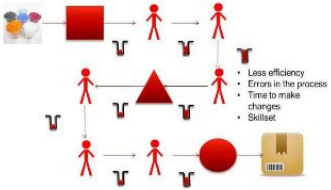




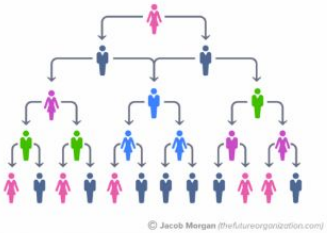
Cycles

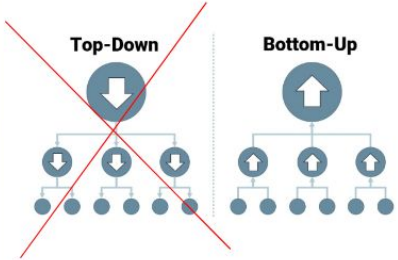


Process

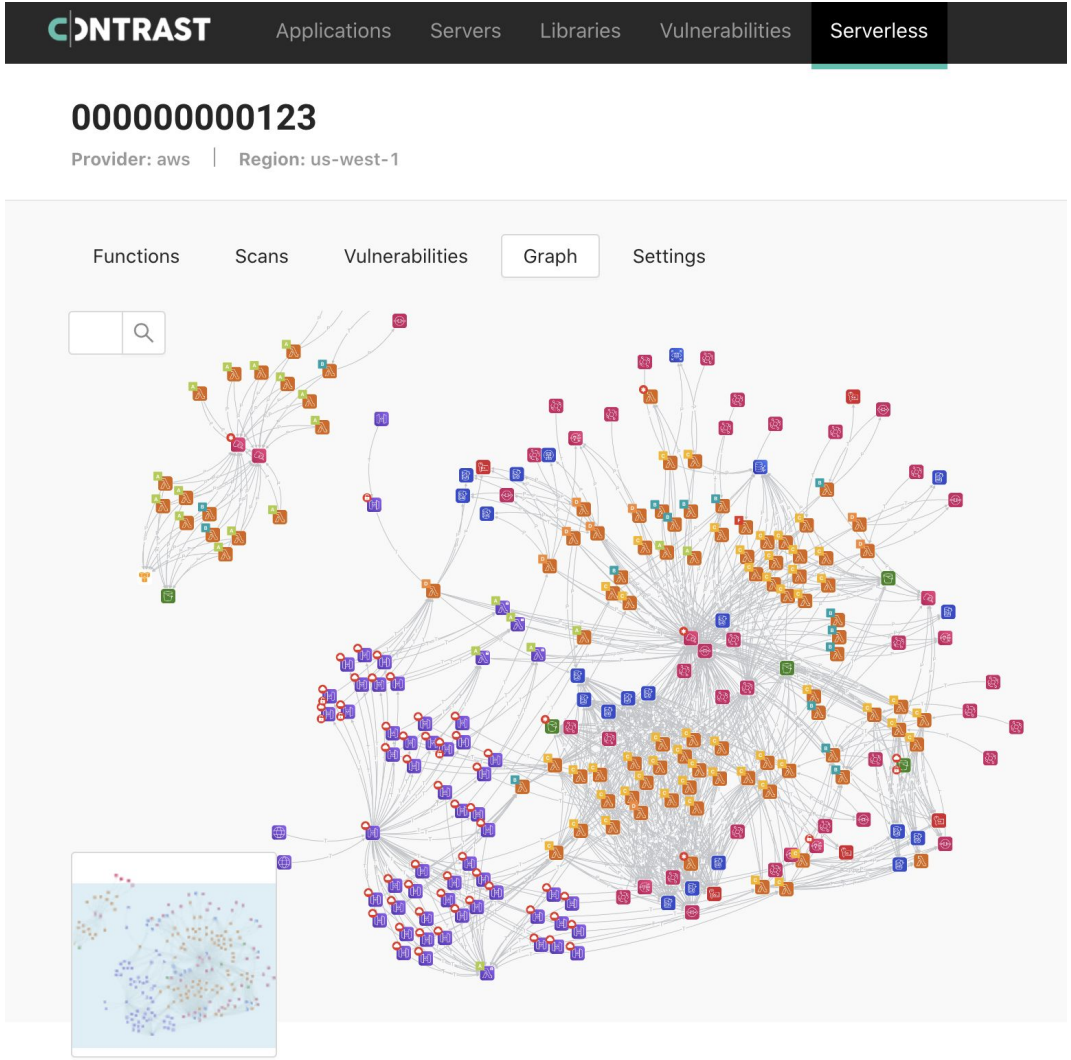


Decision

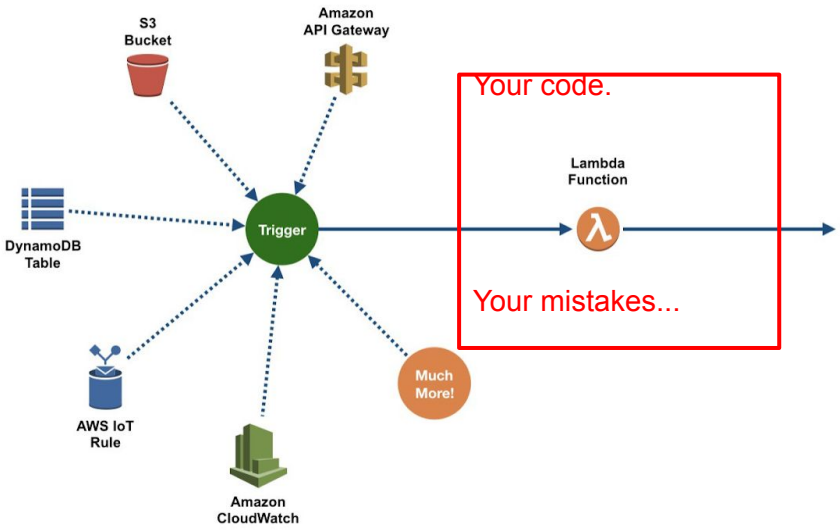


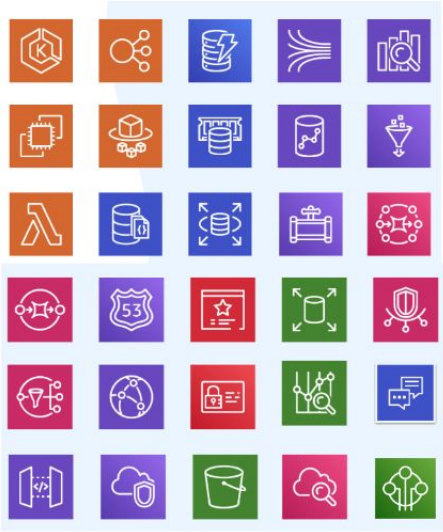


Serverless Architecture



Event-Driven Architecture



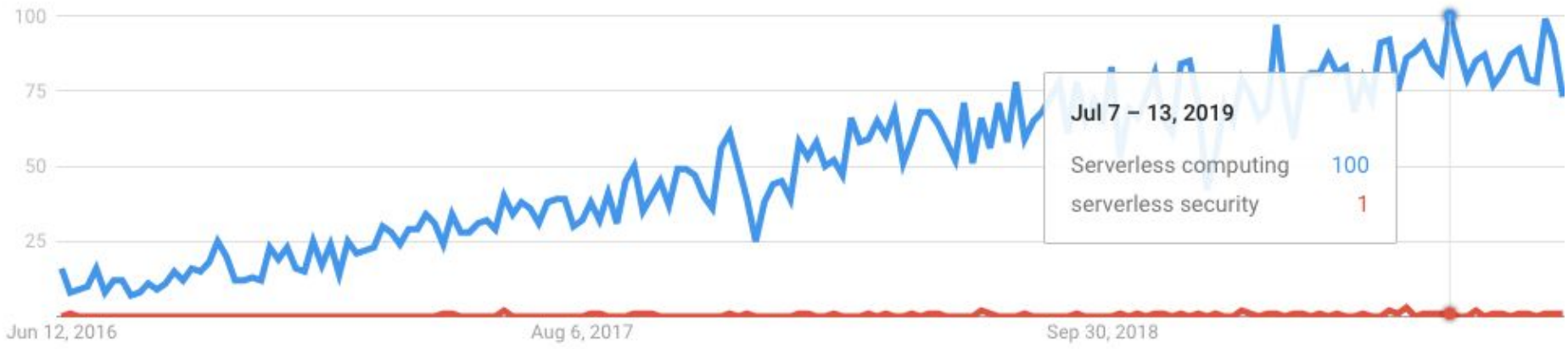


- Triggered via events
- Container spins up when required
- Terminates when code execution

AWS Lambda - Security Aspects

- Read-only environment, except for */tmp*
- Not wired to the internet*
- Data is temporary**
- Code reside in environment
- Keys are available as environment variables

Is Serverless Security a thing?



Can we apply
traditional AppSec
to Serverless?

Resource-Based IAM

DVSA-ORDER-NEW

```
def lambda_handler(event, context):
    orderId = str(uuid.uuid4())
    itemList = event["items"]
    status = 100

    userId = event["user"]
    address = "{}"
    ts = int(time.time())
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table(os.environ["ORDERS_T
    response = table.put_item(
        Item={...}
    )

    if response['ResponseMetadata']['HTTPStatus
        res = {"status": "ok", "msg": "order cr
    else:
        res = {"status": "err", "msg": uld n

    return res
```

Execution role

Role name

serverlessrepo-DVSA-OrderNewFunctionRole-N65M2RQ1B6QS

Resource summary

Amazon DynamoDB
1 action, 2 resources

To view the resources and actions that your function has permission to access, choose a service.

By action | By resource

Action	Resources
dynamodb:PutItem	Allow: arn:aws:dynamodb:us-east-1:402181209224:table/*

Trust no one!

Lambda and DynamoDB : is not authorized to perform: dynamodb:Scan

Asked 2 Months ago Answers: 4 Viewed 63 times

I've created my API with serverless, after I deployed my API into lambda, and we I try to test the endpoint via the "Test" button in the GatewayAPI, I get the error:

> *"User: arn:aws:sts::245912153055:assumed-role/pets-service-dev-us-east-1-lambdaRole/pets-service-dev-listPets is not authorized to perform: dynamodb:Scan on resource: arn:aws:dynamodb:us-east-1:245912153055:table/Pets"*

I should probably need to give the permission to Lambda, but I'm a little bit lost ...

25

Worked with an Amazon engineer and it turns out the problem was in the policy configuration:

should be

```
"dynamodb:*"
```

Trust no one!

2

I solved this by adding the `AWSLambdaFullAccess` permissions to the Lambda

1. Go to the Lambda IAM Role
2. Select "Attach existing policies directly"
3. Search for `AWSLambdaFullAccess`, select it and click `next:review` at the bottom of the page.
4. Click `Add Permissions`

And that should do it.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:*",
        "cognito-identity:ListIdentityPools",
        "cognito-sync:GetCognitoEvents",
        "cognito-sync:SetCognitoEvents",
        "dynamodb:*",
        "events:*",
        "iam:ListAttachedRolePolicies",
        "iam:ListRolePolicies",
        "iam:ListRoles",
        "iam:PassRole",
        "kinesis:DescribeStream",
        "kinesis:ListStreams",
        "kinesis:PutRecord",
        "lambda:*",
        "logs:*",
        "s3:*",
        "sns:ListSubscriptions",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics",
        "sns:Subscribe",
        "sns:Unsubscribe"
      ],
      "Resource": "*"
    }
  ]
}
```

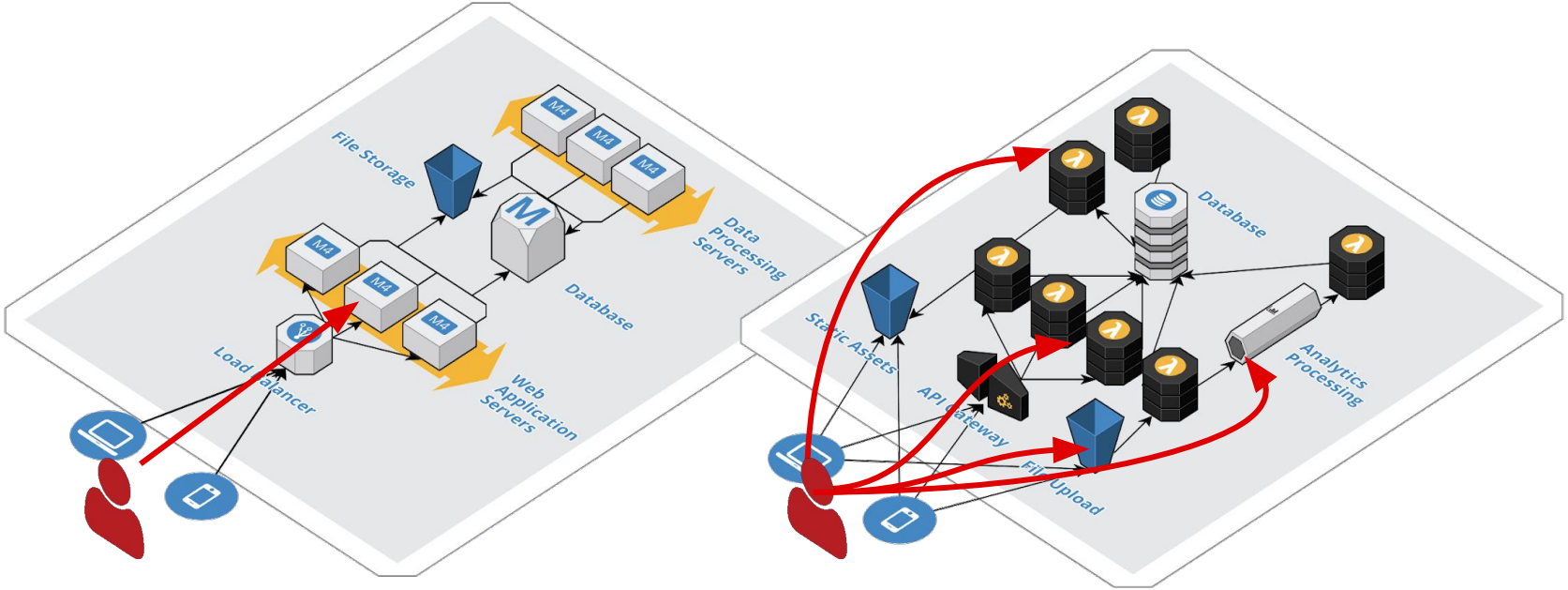
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:*",
        "cognito-identity:ListIdentityPools",
        "cognito-sync:GetCognitoEvents",
        "cognito-sync:SetCognitoEvents",
        "dynamodb:*",
        "events:*",
```

Note

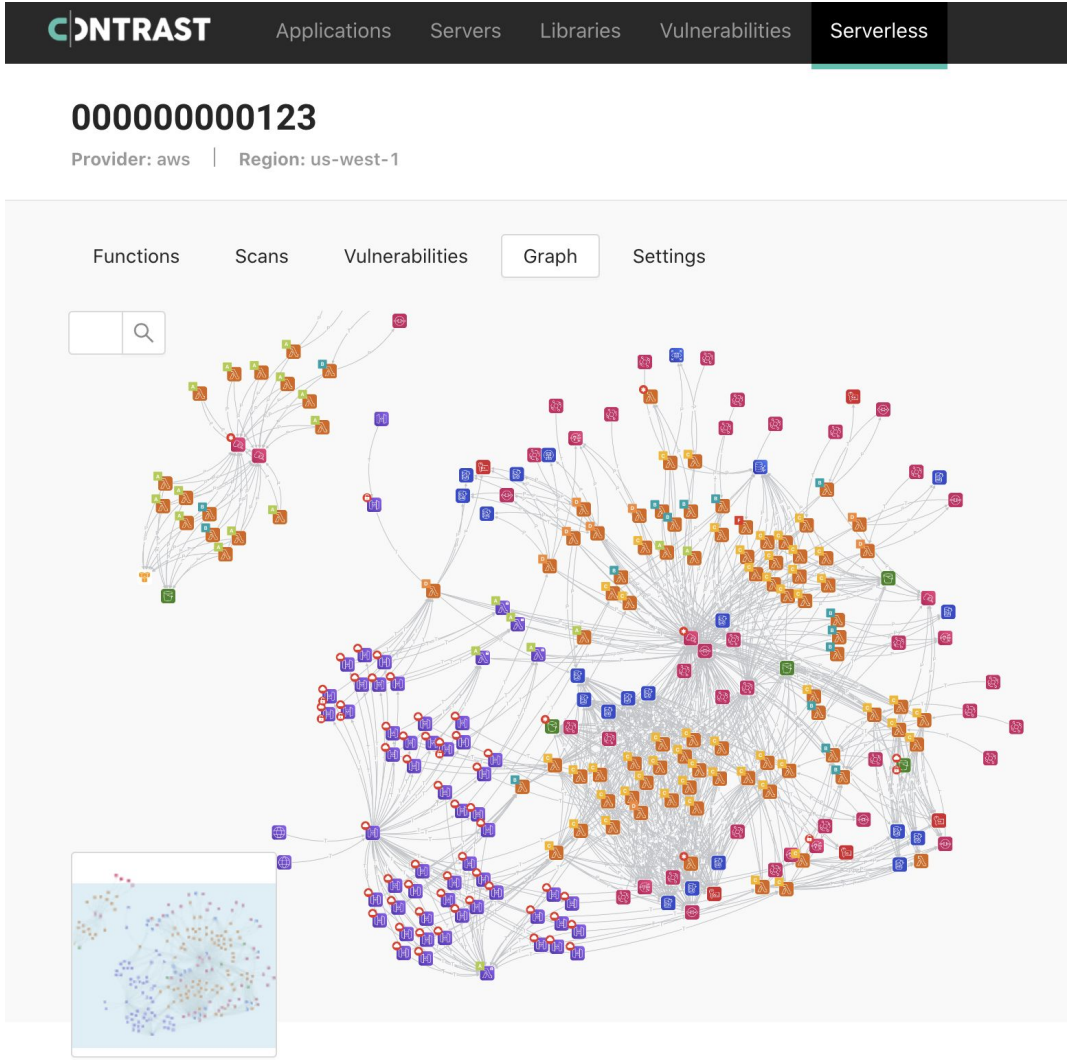
The AWS managed policies **AWSLambdaFullAccess** and **AWSLambdaReadOnlyAccess** will be deprecated on March 1, 2021. After this date, you cannot attach these policies to new IAM users. For more information, see the related troubleshooting topic.

```
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics",
        "sns:Subscribe",
        "sns:Unsubscribe"
      ],
      "Resource": "*"
    }
  ]
}
```

Loss of Perimeter



Serverless Architecture



Serverless Risks

Event Injection

Broken Authentication

Sensitive Data Exposure

Over-Privileged Functions

Vulnerable Dependencies

Insufficient Logging & Monitoring

Open Resources

DoW / DoS

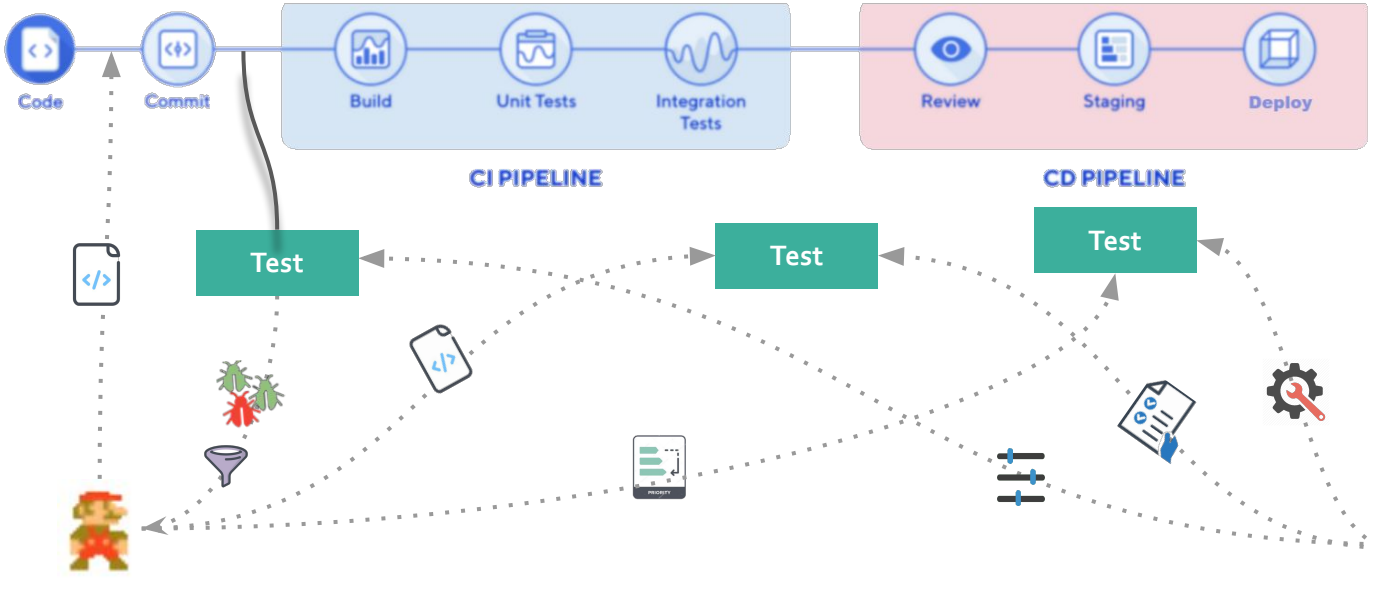
Insecure Shared Space

Insecure Secret Management

Can Security Scale?

- Lots of services
- Frequent deployments
- What is connected to what?
- Many developers /Few AppSec
- What's important?
- Is security even the same?
- Who takes care of Infra?

Traditional Testing in **Modern** CI/CD Pipelines

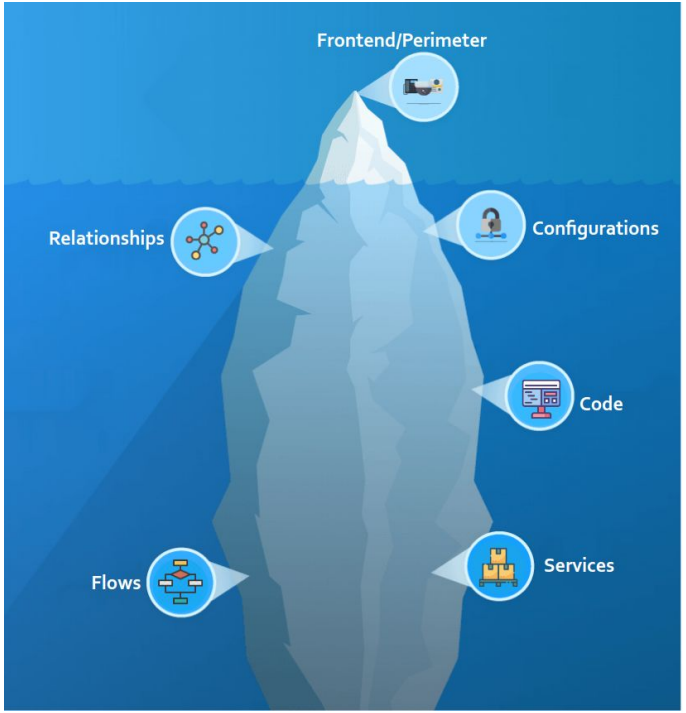


Mario,
Sr. Developer, Toadstool Inc.

Luigi,
AppSec Team

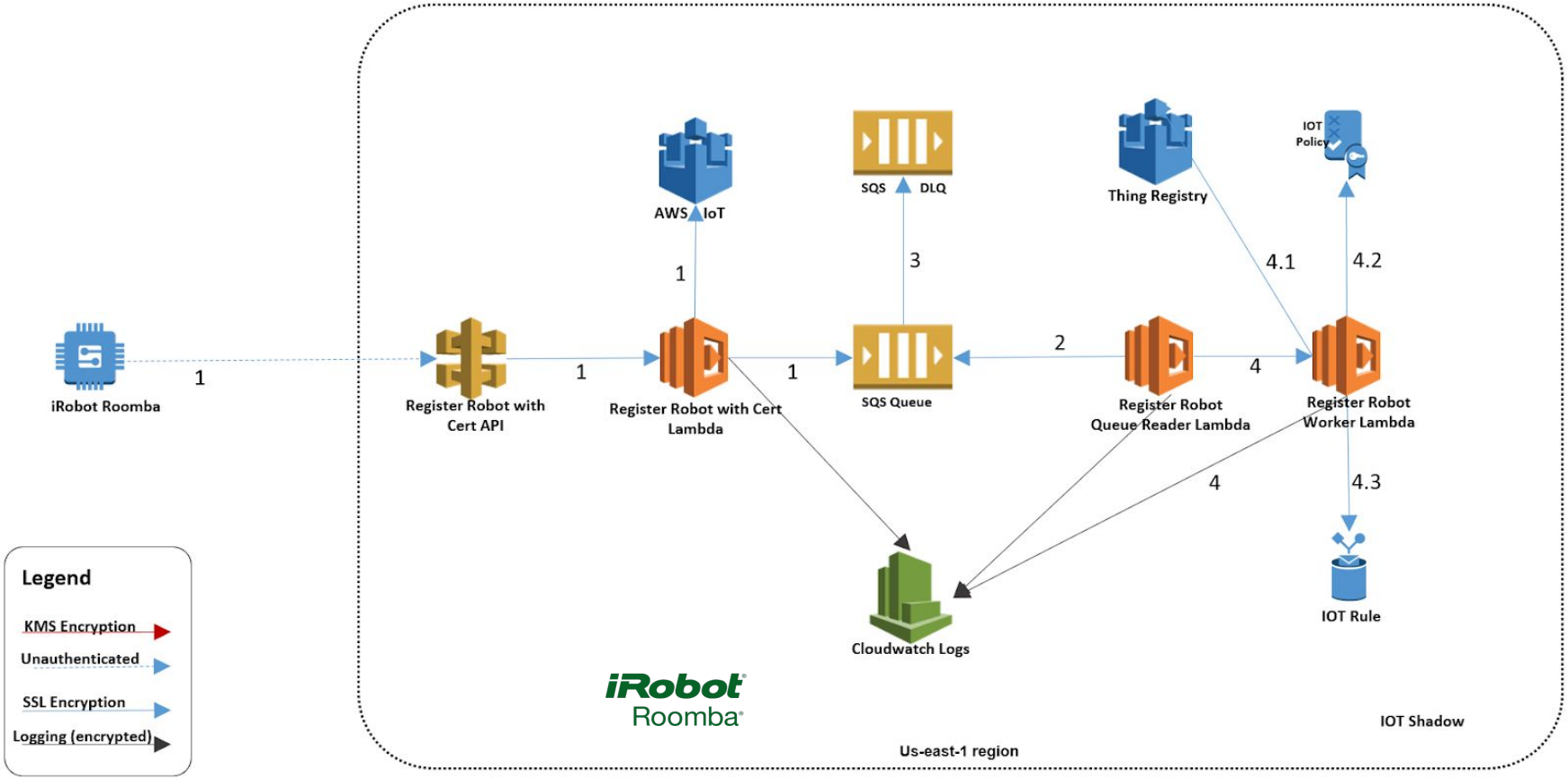
Traditional AppSec Testing for Cloud Native

- Ignorant to the environment and context
- Completely blind to no-edge services
- Block developers, disruptive to CI/CD
- Hard to scale



How can we do **Security Testing** on Serverless Apps?

iRobot Serverless App

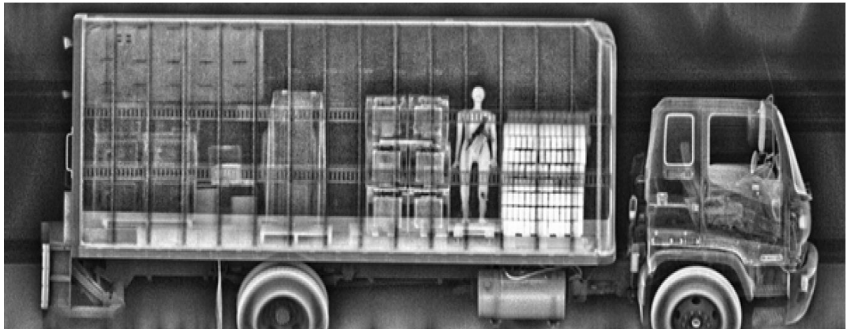


@ 4ppsec

Source: https://aws.amazon.com/solutions/case-studies/irobot/

Easy... **SCA & Image Scanning**

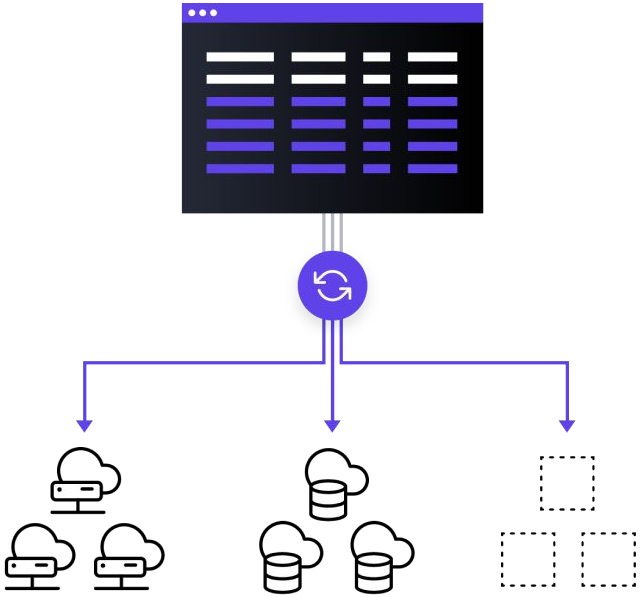
- Covers ~10% of your app
- Just fixing problems you imported
- What about your **CODE**, your services & your configurations?
- Provided by the cloud provider and many OS Project



@ 4ppsec

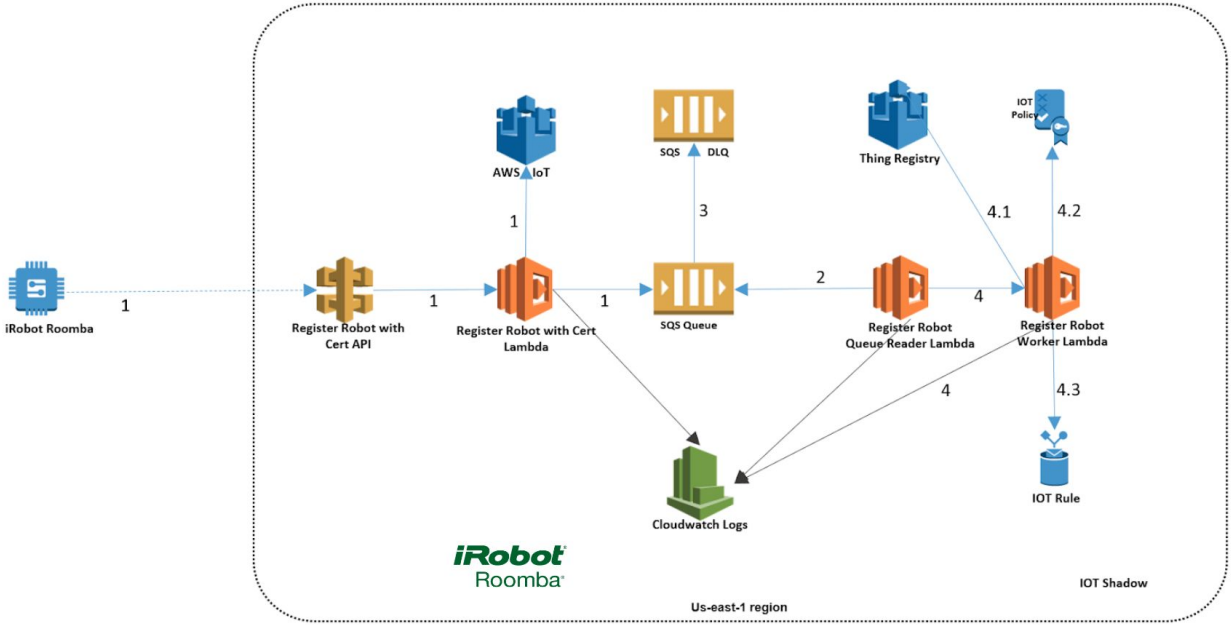
I know… **IaC**

- Really?! Blaming it on the DevOps?
- As left as it can get, but…
  - Limited visibility
  - **Zero code coverage**
  - No logic, no prioritization
  - IaC dependant



IAST!

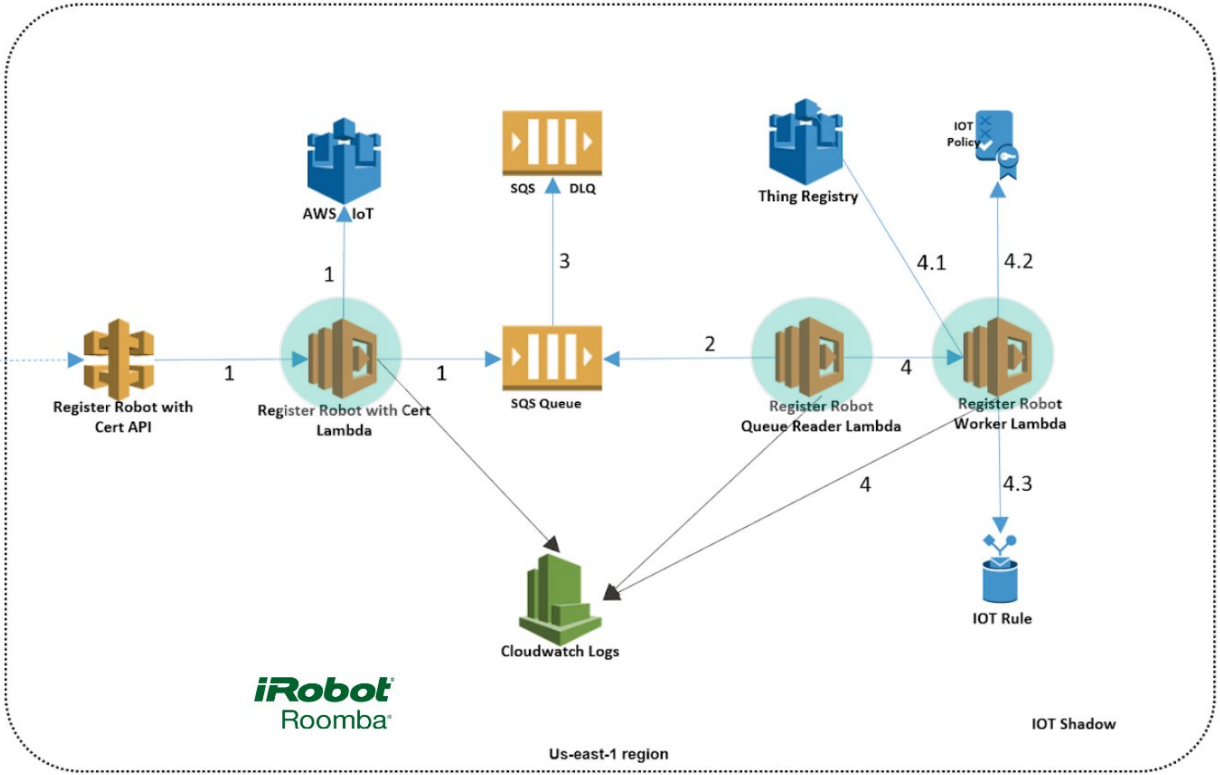
- Modern AppSec
- Accurate & Reliable
- Enables DevOps



But! No <u>Server</u> to instrument...

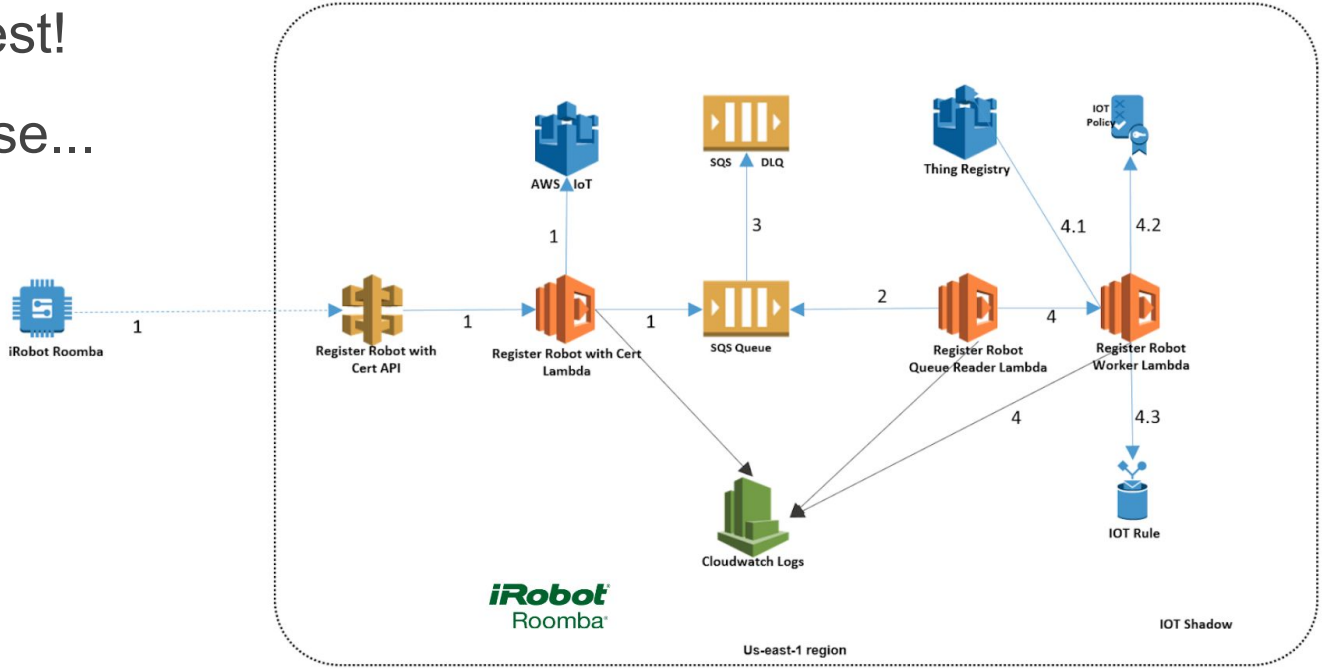
SAST

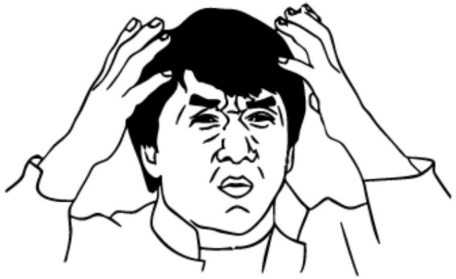
- So many apps??
- No source (http/input)?
- No sink (output)?
- No DBs?
- Who wrote this?
- What's going on???



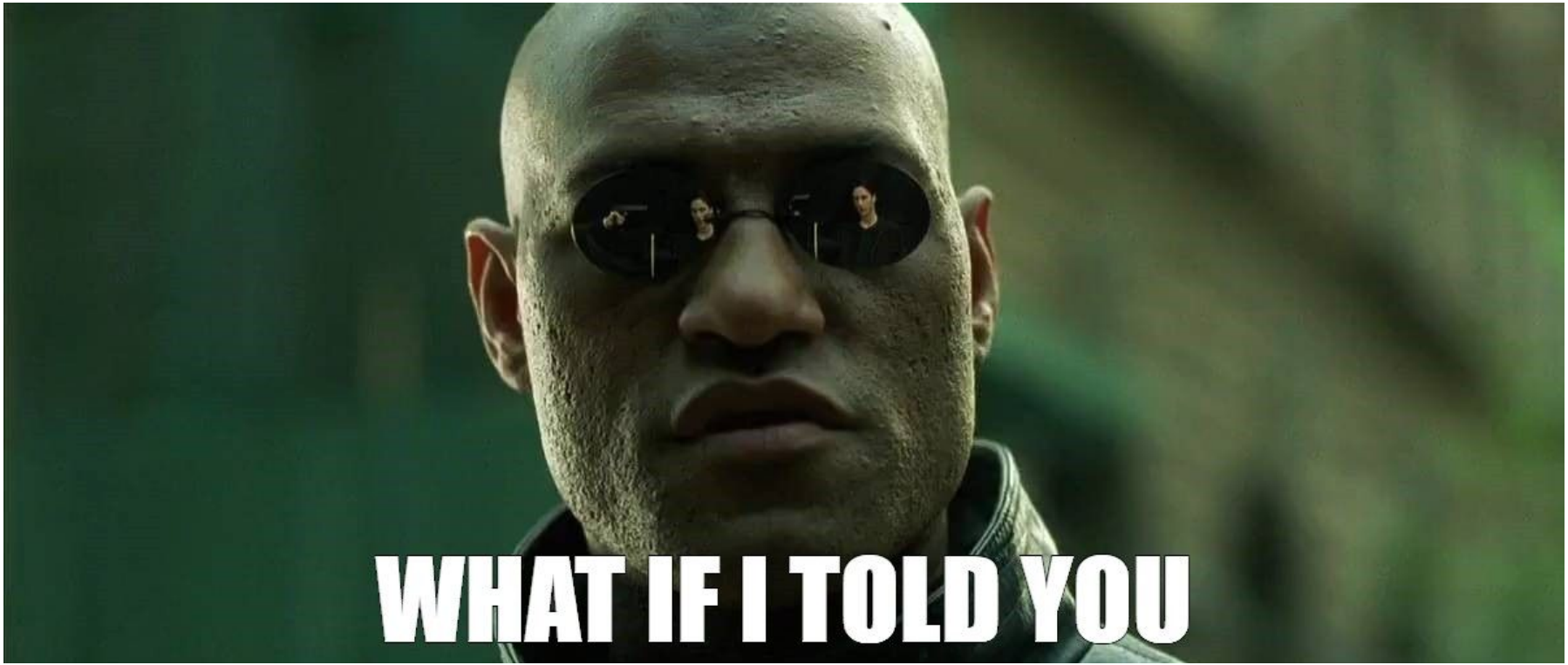
DAST

- No HTTP/S Request!
- No (sync) Response...
- Nothing to test?

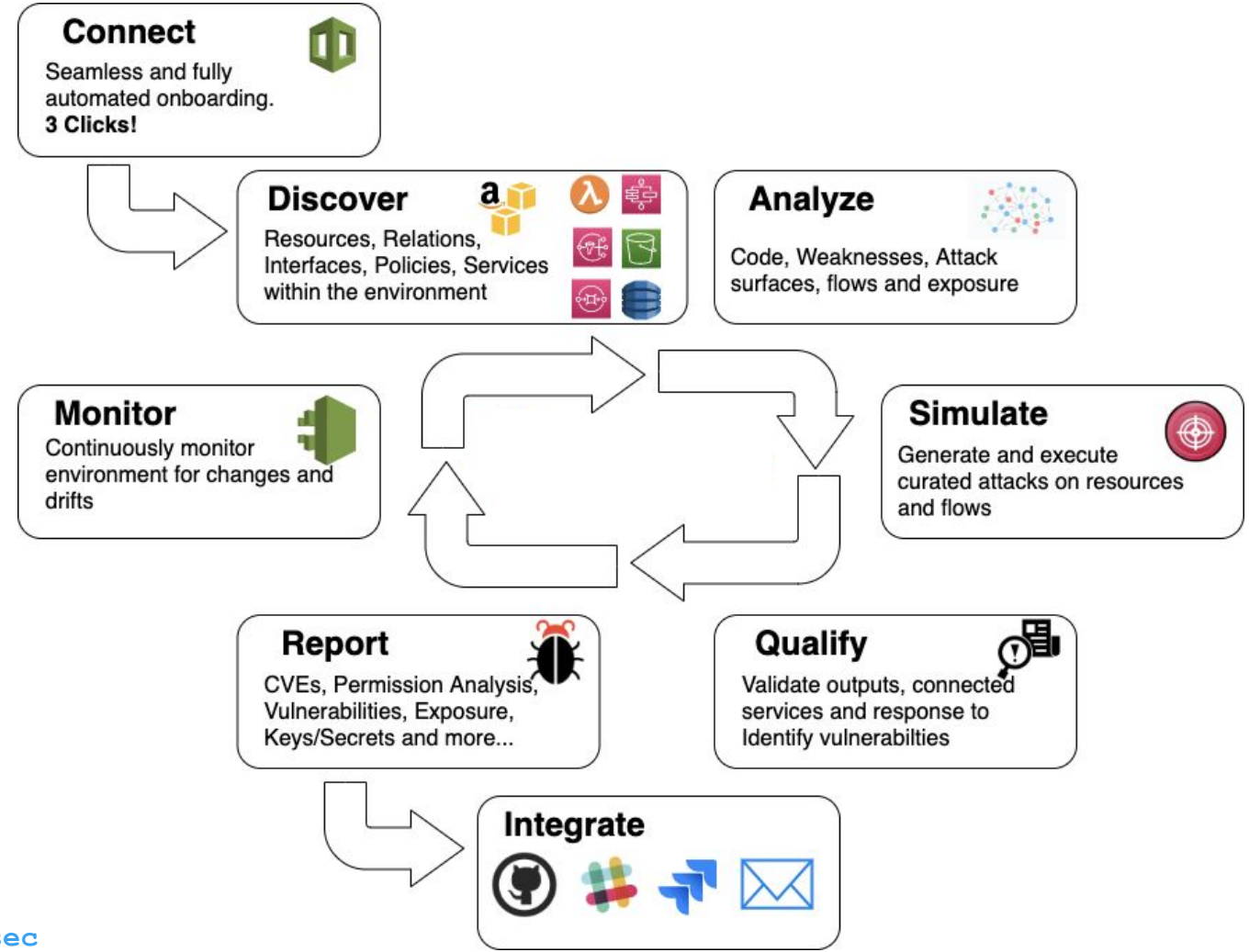




Where do I even start?!

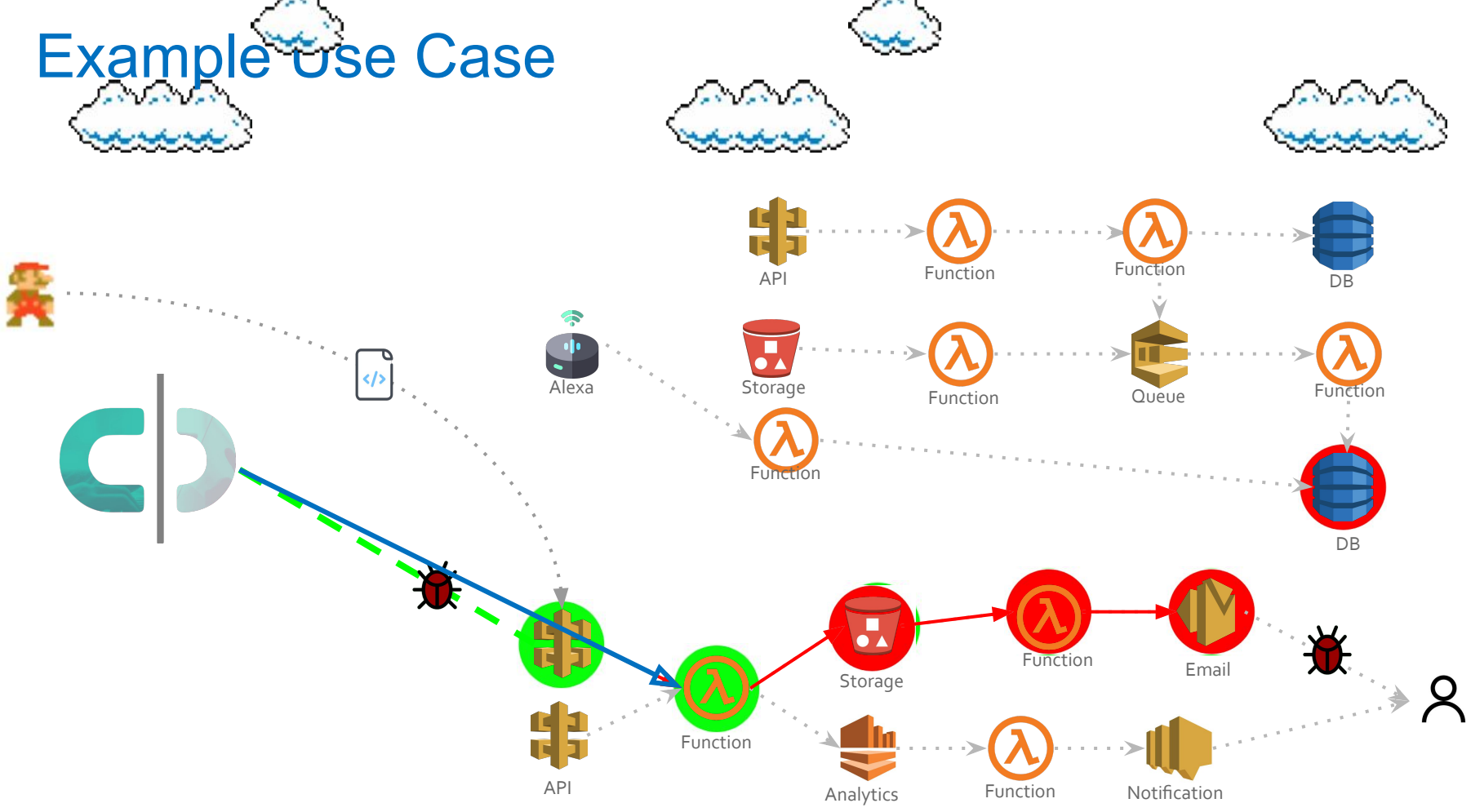


@ 4ppsec


Connect
Seamless and fully automated onboarding.
3 Clicks!
Discover
Resources, Relations, Interfaces, Policies, Services within the environment
Analyze
Code, Weaknesses, Attack surfaces, flows and exposure
Monitor
Continuously monitor environment for changes and drifts
Simulate
Generate and execute curated attacks on resources and flows
Report
CVEs, Permission Analysis, Vulnerabilities, Exposure, Keys/Secrets and more...
Qualify
Validate outputs, connected services and response to Identify vulnerabilties
Integrate

@ 4ppsec

Example Use Case



Automated Serverless Security Testing

OS Command Injection

HIGH | **Category:** Exploits | **Function:** arn:aws:lambda:us-east-1:733980656228:function:contras

Description

Command injection (CWE-77, CWE-78) is an attack in which the goal is execution of arbitrary comman
vulnerable application. Command injection attacks are possible when an application passes unsafe data
attacker-supplied commands is executed with the privileges of the vulnerable function. Command injec
insufficient input validation. This attack differs from Code Injection, in that code injection allows the atta
executed by the application. In Command Injection, the attacker extends the default functionality of the
commands, without the necessity of injecting code.

What happened

cloudessence has identified evidence in the function logs (CWL) that indicates of was successfull explo

```
logGroup: /aws/lambda/contrast-vulnerable-function-via-S3
logStream: 2021/11/02/[$LATEST]c8c20827d5dc49ab9c2523206e6525f2
payload: |echo $((4*9537971237249))
requestId: 49d03709-b176-43fb-8e9b-7bd03ccd66a9
type: Log
vector: arn:aws:s3:::contrast-vulnerable-bucket-733980656228-us-east-1
```

Remediation

If at all possible, use library calls rather than external processes to recreate the desired functionality. Ac
sanitize all inputs from untrusted sources.

Impact

```
{
    "PolicyName": "DVSA-AdminRolePolicy-dev",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "1",
                "Effect": "Allow",
                "Action": [
                    "logs:CreateLogGroup",
                    "logs:CreateLogStream",
                    "logs:PutLogEvents"
                ],
                "Resource": [
                    "arn:aws:logs:us-east-1:733980656228:log-group:/aws/lambda/*:*:*"
                ]
            },
            {
                "Sid": "2",
                "Effect": "Allow",
                "Action": [
                    "dynamodb:Scan"
                ],
                "Resource": [
                    "arn:aws:dynamodb:us-east-1:733980656228:table/*"
                ]
            }
        ]
    }
}
```

OWASP Serverless Top 10

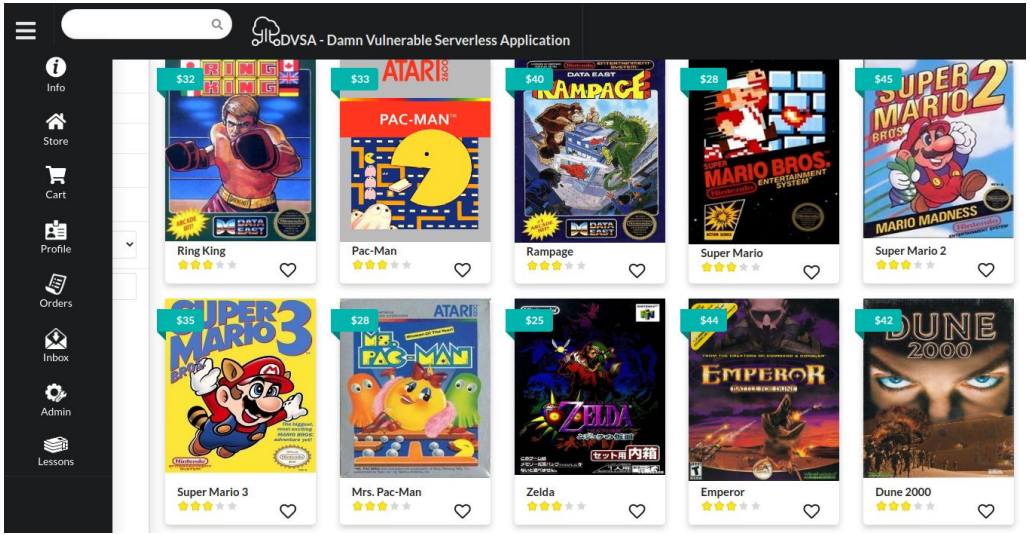
- Current project state:
  - Interpretation of Top 10
  - Open Data Call: https://appsec.it/serverless-call
- Goal: Serverless-tailored Top 10

https://owasp.org/www-project-serverless-top-10/

github.com/owasp/dvsa

@DVSAowasp

! NOT in PRODUCTION !



https://owasp.org/www-project-dvsa/

Thank you!

Tal.Melamed@ContrastSecurity.com

CONF42CAST