Cloud Monitoring

CHECKMK CONFERENCE #5 – MUNICH, APRIL 29, 2019

tribe29

CLOUD MONITORING

Agenda

1. PARADIGM SHIFTS AND HOW WE ADDRESS THEM
2. OUR APPROACH TO CLOUD MONITORING
3. AWS
4. AZURE

Paradigm Shifts in the Cloud

Highly dynamic environments

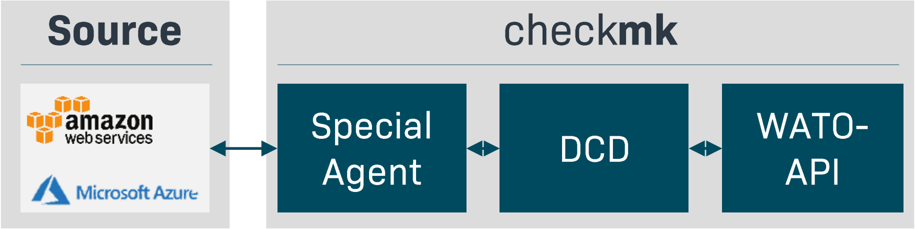
“Labels“ as central concept to manage infrastructure

Rate and resource limits are externally imposed

Costs are pay-as-you-grow – but grow quickly

ADDRESSING THE CHALLENGES

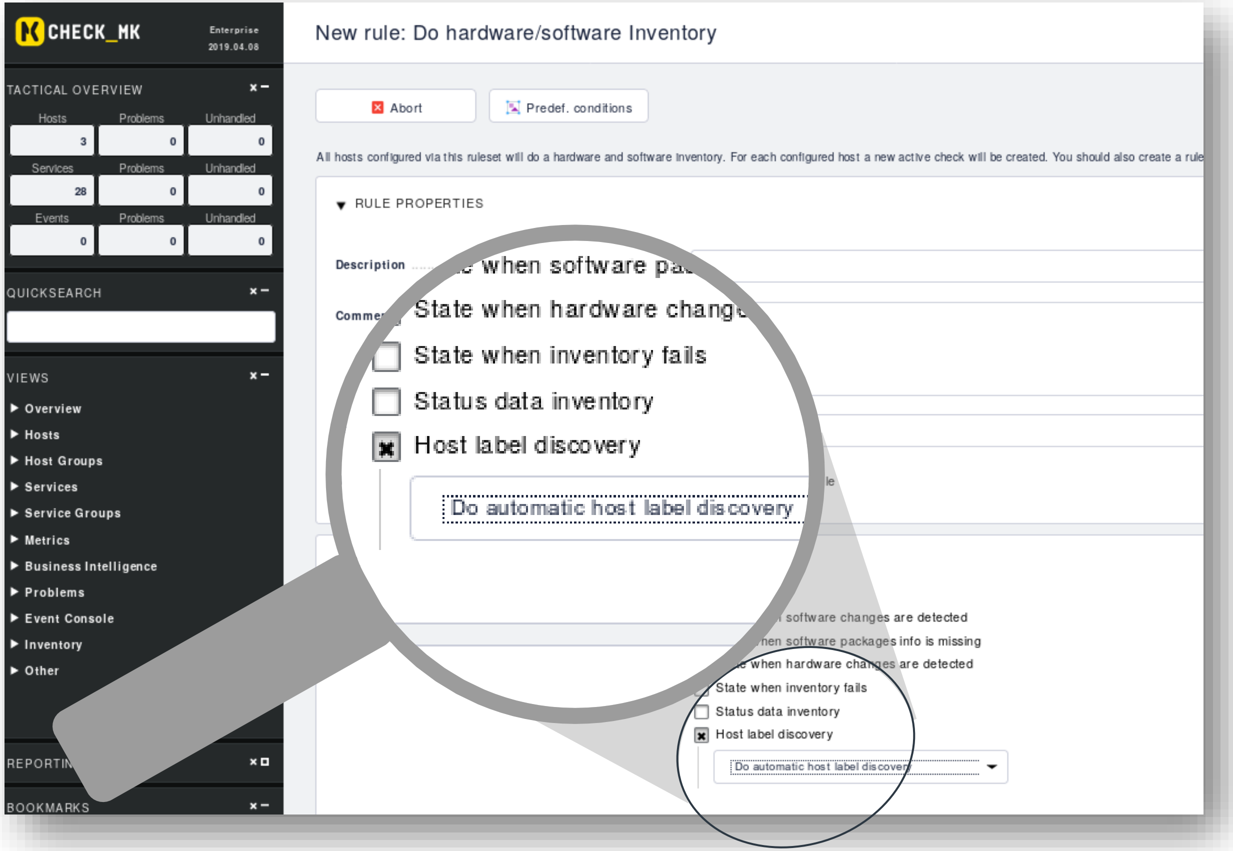
How: Dynamic environments



- Dynamic Configuration Deamon (DCD) was built for these more dynamic cloud environments
- For AWS and Azure, DCD performs two primary tasks
  - Dynamic creation of Piggyback Hosts
  - Automatic Service Discovery

ADDRESSING THE CHALLENGES

How: Handling of ‘Service Labels’



New rule: Do hardware/software Inventory

All hosts configured via this ruleset will do a hardware and software inventory. For each configured host a new active check will be created. You should also create a rule

…software changes are detected

…hen software packages info is missing

…e when hardware changes are detected

☐ State when inventory fails

☐ Status data inventory

☒ Host label discovery

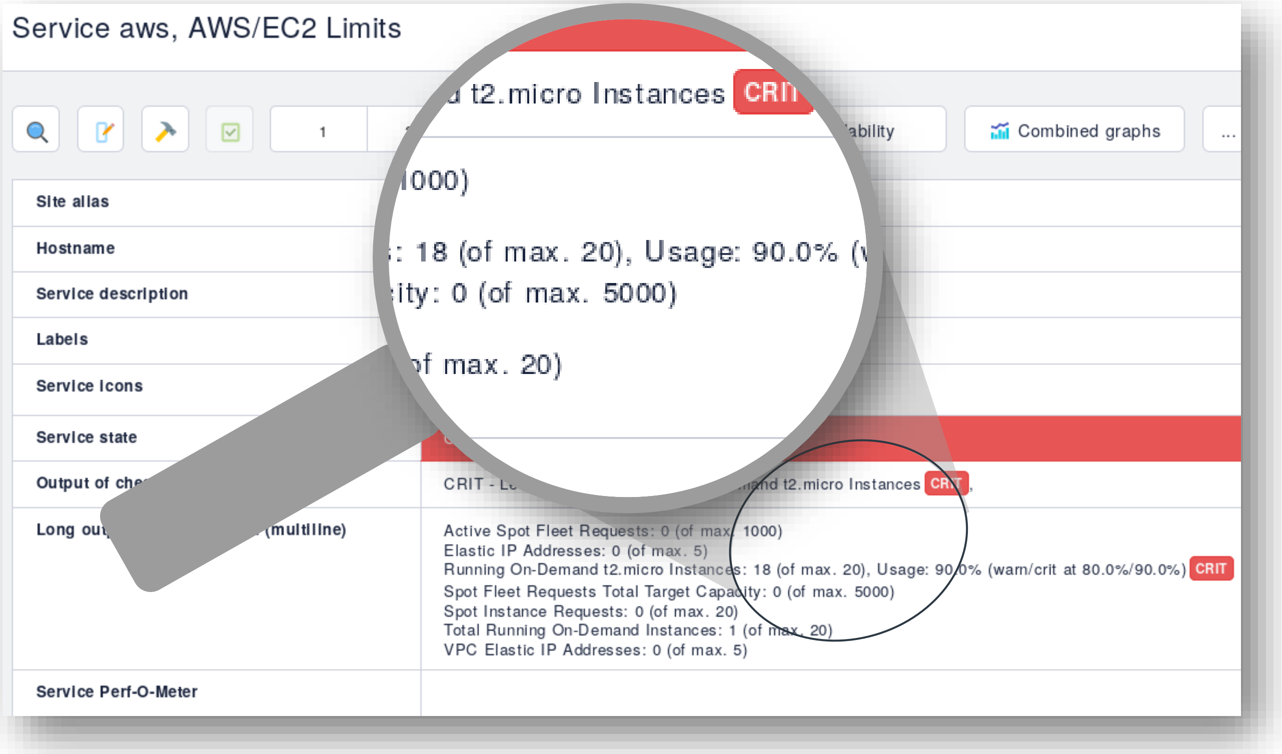
Do automatic host label discovery

- Automatic discovery of labels through HW/SW inventory („Host Label Discovery“)
- Two use cases:
  - Searching hosts and services
  - Special Agent configuration: limiting retrieved data to certain labels

tribe29

ADDRESSING THE CHALLENGES

How: Handling of Rate & Resource Limits



Service aws, AWS/EC2 Limits

Site alias

Hostname

Service description

Labels

Service icons

Service state

Output of che… CRIT - Le… …and t2.micro Instances CRIT,

Long out… (multiline)

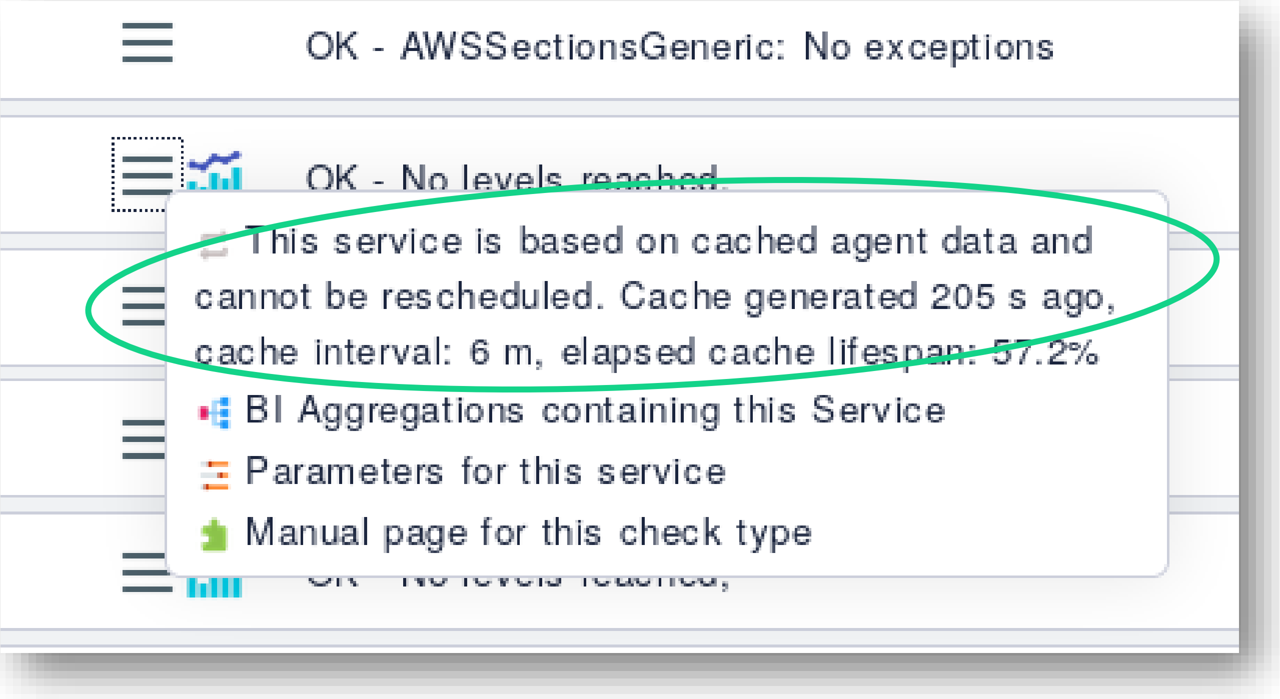
```
Active Spot Fleet Requests: 0 (of max. 1000)
Elastic IP Addresses: 0 (of max. 5)
Running On-Demand t2.micro Instances: 18 (of max. 20), Usage: 90.0% (warn/crit at 80.0%/90.0%) CRIT
Spot Fleet Requests Total Target Capacity: 0 (of max. 5000)
Spot Instance Requests: 0 (of max. 20)
Total Running On-Demand Instances: 1 (of max. 20)
VPC Elastic IP Addresses: 0 (of max. 5)
```

Service Perf-O-Meter

- Resource limits
  - check**mk** monitors account resource limits
  - Limits provided by API (usually), individual limits (i.e. from custom contracts) can be edited
  - Limits are monitored at the account level (e.g. max 20 EC2 instances per region)
  - Monitoring at a resource level where sensible

ADDRESSING THE CHALLENGES

How: Handling of Rate & Resource Limits



OK - AWSSectionsGeneric: No exceptions

OK - No levels reached

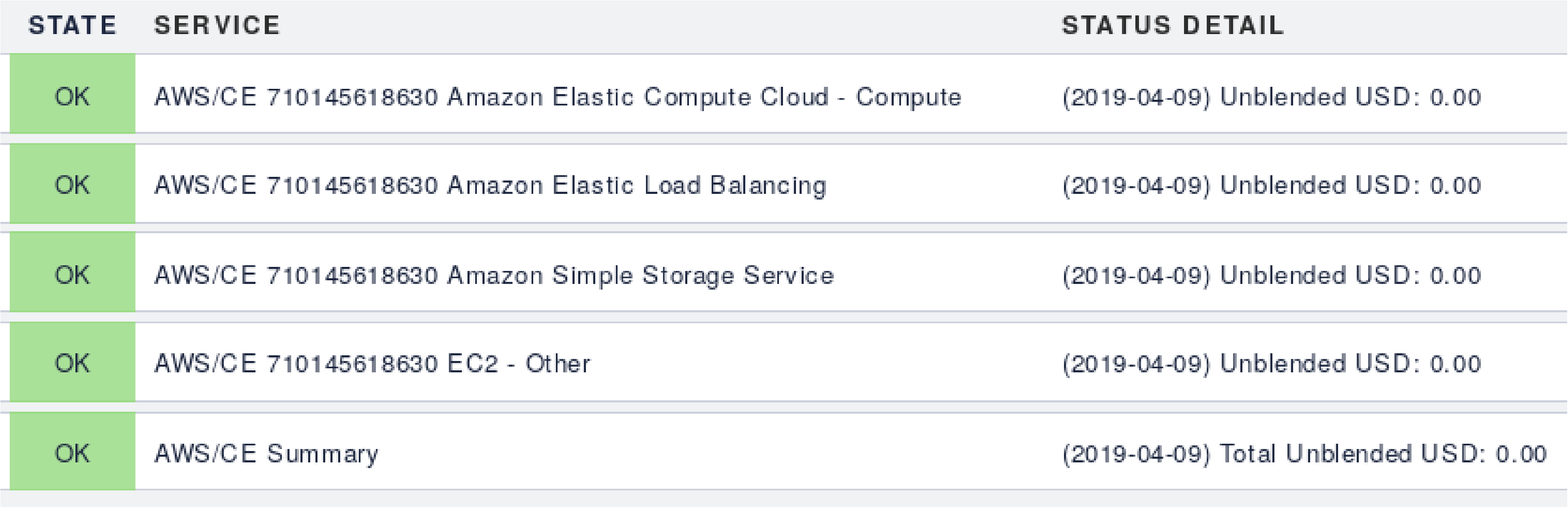
This service is based on cached agent data and cannot be rescheduled. Cache generated 205 s ago, cache interval: 6 m, elapsed cache lifespan: 57.2%

BI Aggregations containing this Service
Parameters for this service
Manual page for this check type
OK - No levels reached,

- Rate limits
  - Azure Agent monitors rate limits for Azure API
  - To limit use, agent bundles requests and internally caches data
  - Can be further optimized through explicit config

ADDRESSING THE CHALLENGES

How: Handling of Cost Monitoring



STATE	SERVICE	STATUS DETAIL
OK	AWS/CE 710145618630 Amazon Elastic Compute Cloud - Compute	(2019-04-09) Unblended USD: 0.00
OK	AWS/CE 710145618630 Amazon Elastic Load Balancing	(2019-04-09) Unblended USD: 0.00
OK	AWS/CE 710145618630 Amazon Simple Storage Service	(2019-04-09) Unblended USD: 0.00
OK	AWS/CE 710145618630 EC2 - Other	(2019-04-09) Unblended USD: 0.00
OK	AWS/CE Summary	(2019-04-09) Total Unblended USD: 0.00

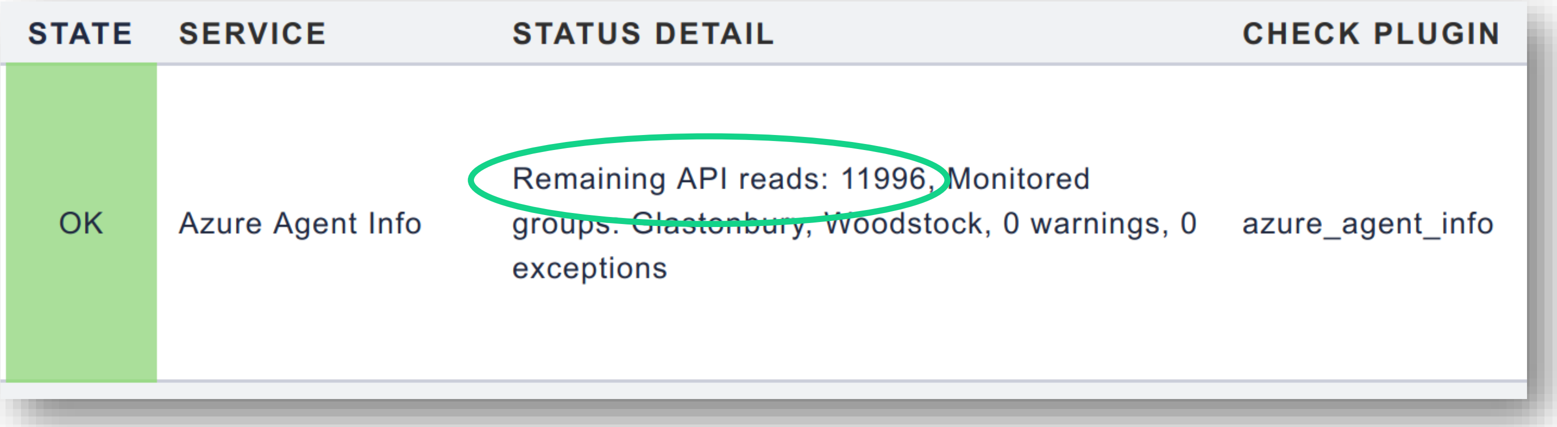
Cost Monitoring is the very next development for Azure

ADDRESSING THE CHALLENGES

How much: Speaking of costs… monitoring costs?

- Unfortunately, monitoring cloud services is not free (at least with Amazon)
  - AWS charges 0.01 USD / 1000 API calls
  - Example: Cost for monitoring 300 AWS Cloudwatch metrics: ≈ 2 USD/day

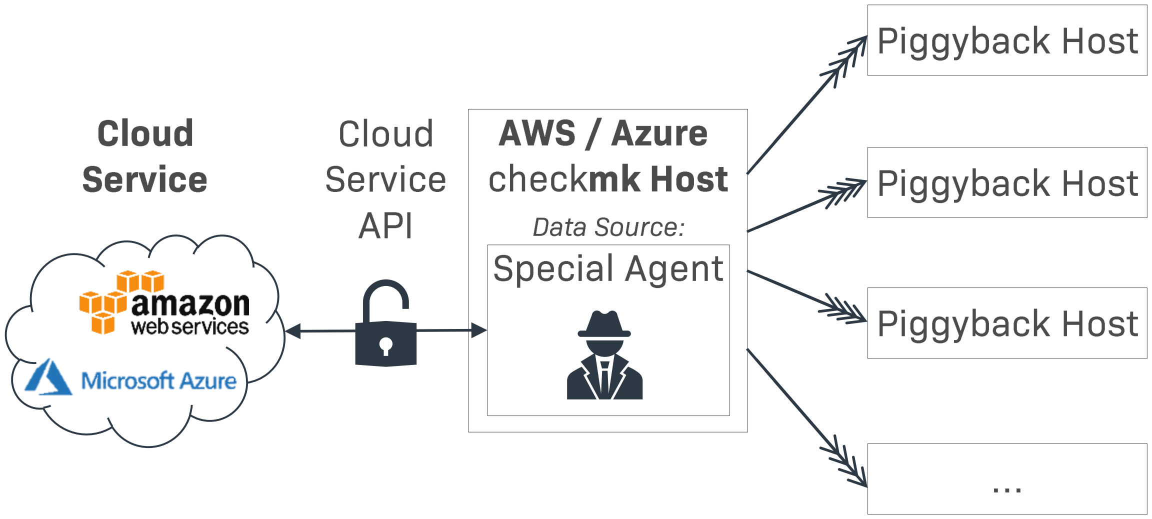
- API Calls for Azure are not charged, but have a relatively strict rate limit



STATE	SERVICE	STATUS DETAIL	CHECK PLUGIN
OK	Azure Agent Info	Remaining API reads: 11996, Monitored groups: Glastonbury, Woodstock, 0 warnings, 0 exceptions	azure_agent_info

OUR APPROACH

Technical Concept



- Technical concept for monitoring cloud services is well-established
- check**mk** dynamically creates *Piggyback Hosts*
- Data is piggybacked by AWS/Azure Host to these hosts

OUR APPROACH

Multiple data sources used

checkmk
by tribe29

AWS* Special Agent

Amazon Cloudwatch

Internal Monitoring API
AWS Cloudwatch

Service APIs
Directly from resource / service (e.g. EC2 instance)

Global Services
Log / Event services
Cost Explorer

** similar for Azure*

Amazon Web Serices

AWS MONITORING

Working on checks for the most important AWS services

Selection of AWS services

Compute	Storage	Database	Networking	Serverless	Mgmt Tools
🟩 EC2	🟩 S3	🟩 RDS	🟩 Elastic Load Balancing	🟦 Lambda	🟩 Cost Explorer
□ Elastic Beanstalk	🟩 EBS	🟦 DynamoDB	□ CloudFront	□ Elastic Container Service	🟦 CloudTrail
□ ...	🟦 Glacier	□ Redshift	□ ...	🟦 SNS	🟦 CloudWatch
	□ ...	□ ...		□ Fargate	🟦 API Gateway
				□ ...	□ ...

🟩 Existing 🟦 Planned □ Future

How it works: AWS Monitoring

AWS Resources...

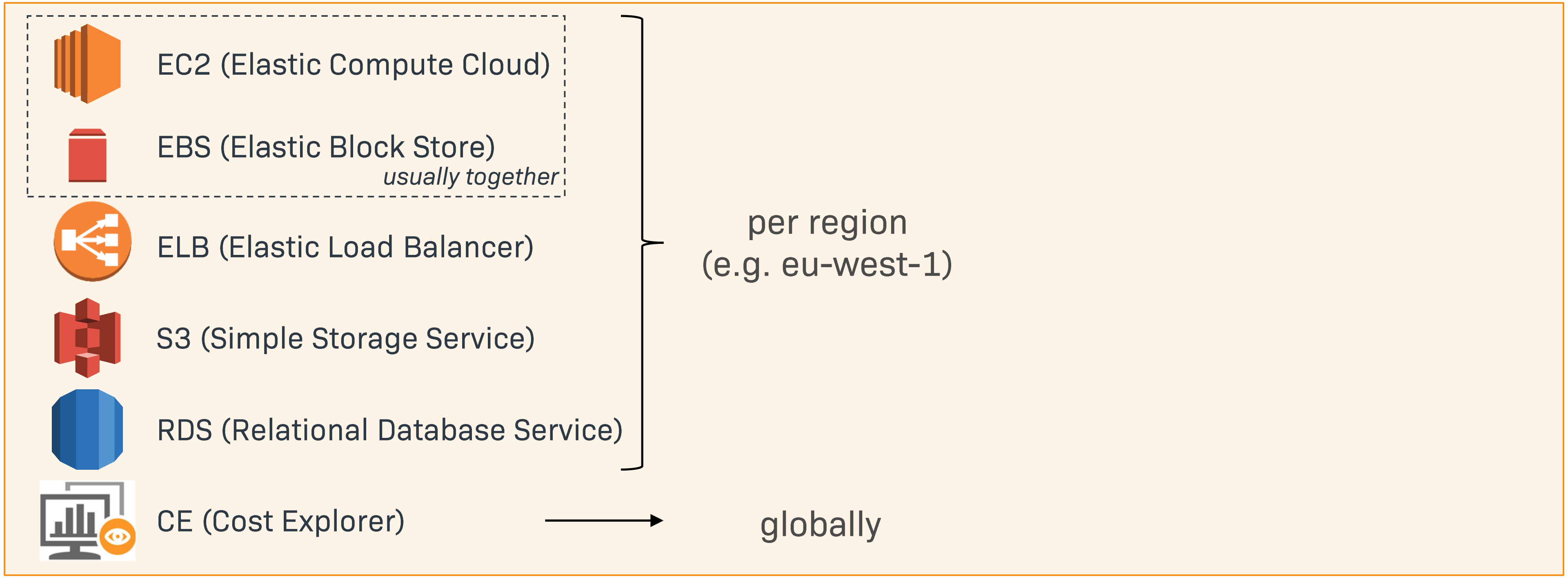
- EC2 (Elastic Compute Cloud)
- EBS (Elastic Block Store)

usually together

- ELB (Elastic Load Balancer)
- S3 (Simple Storage Service)
- RDS (Relational Database Service)
- CE (Cost Explorer)

How it works: AWS Monitoring

AWS Resources... ... are available by region or globally...



EC2 (Elastic Compute Cloud)

EBS (Elastic Block Store)

usually together

ELB (Elastic Load Balancer)

S3 (Simple Storage Service)

RDS (Relational Database Service)

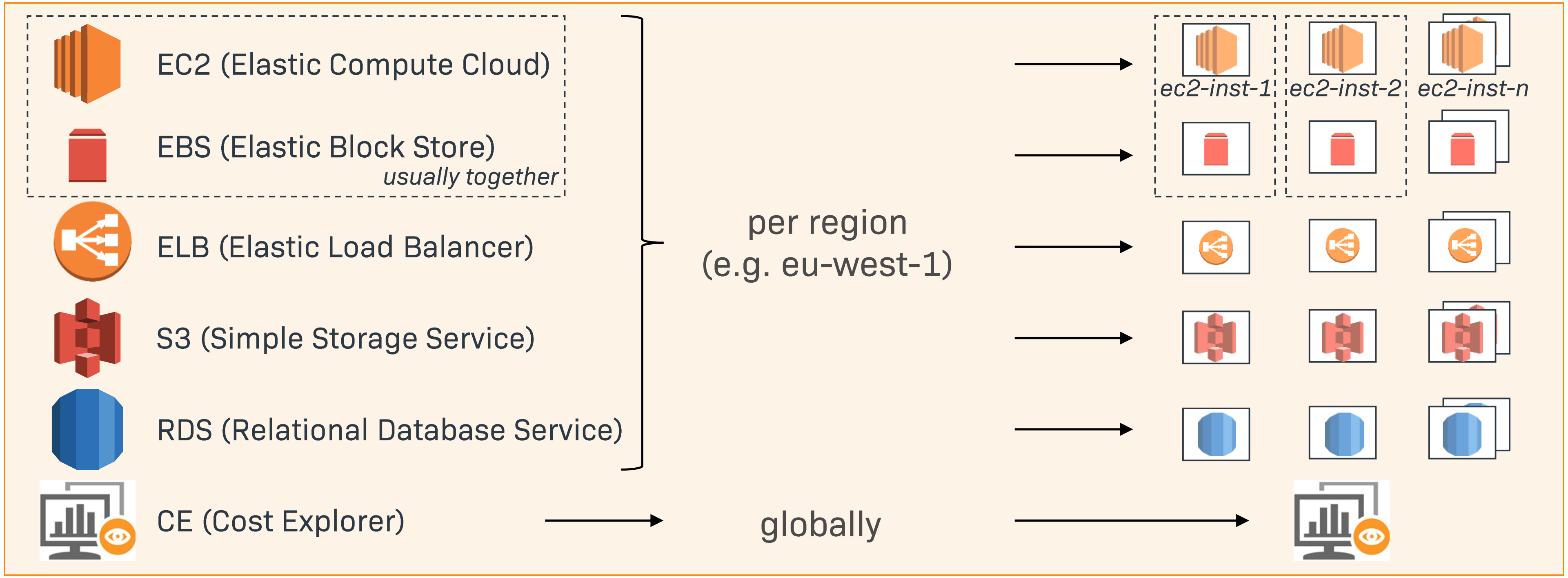
per region
(e.g. eu-west-1)

CE (Cost Explorer)

globally

How it works: AWS Monitoring

AWS Resources… … are available by region or globally… … and can exist in multiple instances



EC2 (Elastic Compute Cloud)

EBS (Elastic Block Store)

usually together

ELB (Elastic Load Balancer)

S3 (Simple Storage Service)

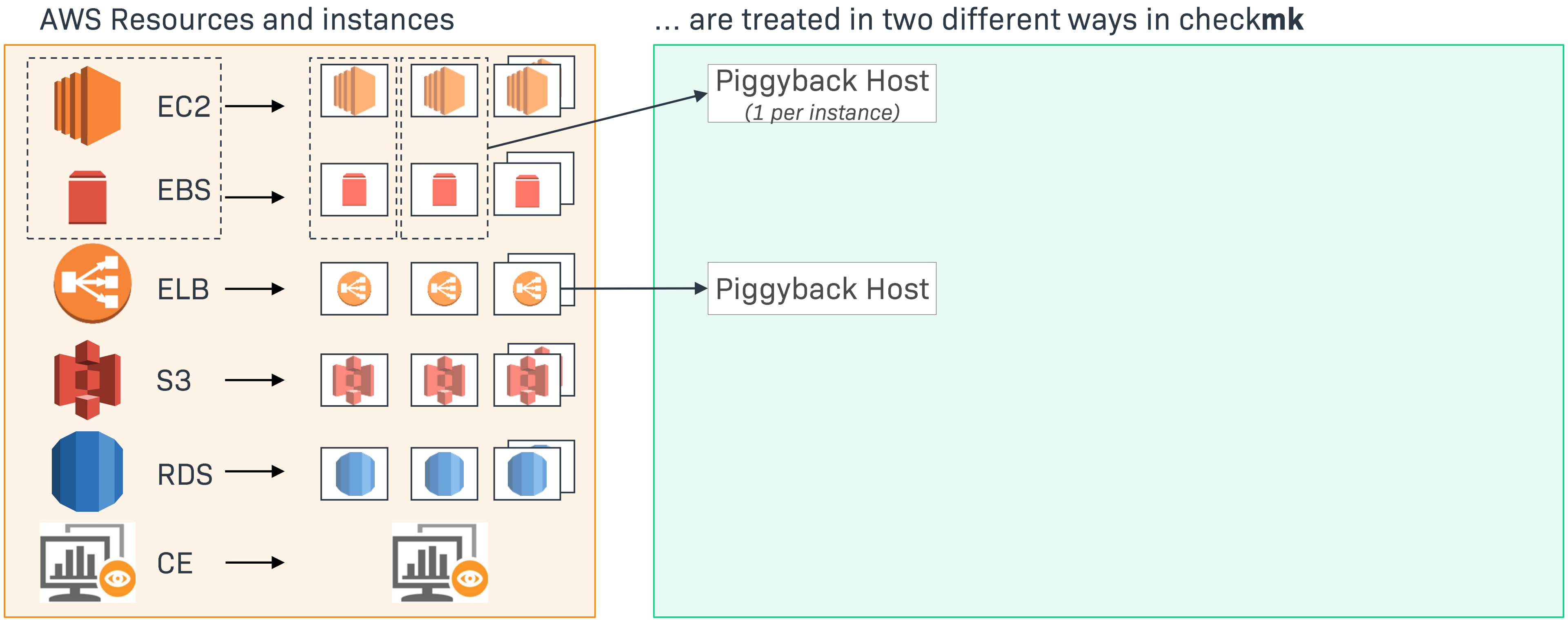
RDS (Relational Database Service)

CE (Cost Explorer)

per region
(e.g. eu-west-1)

globally

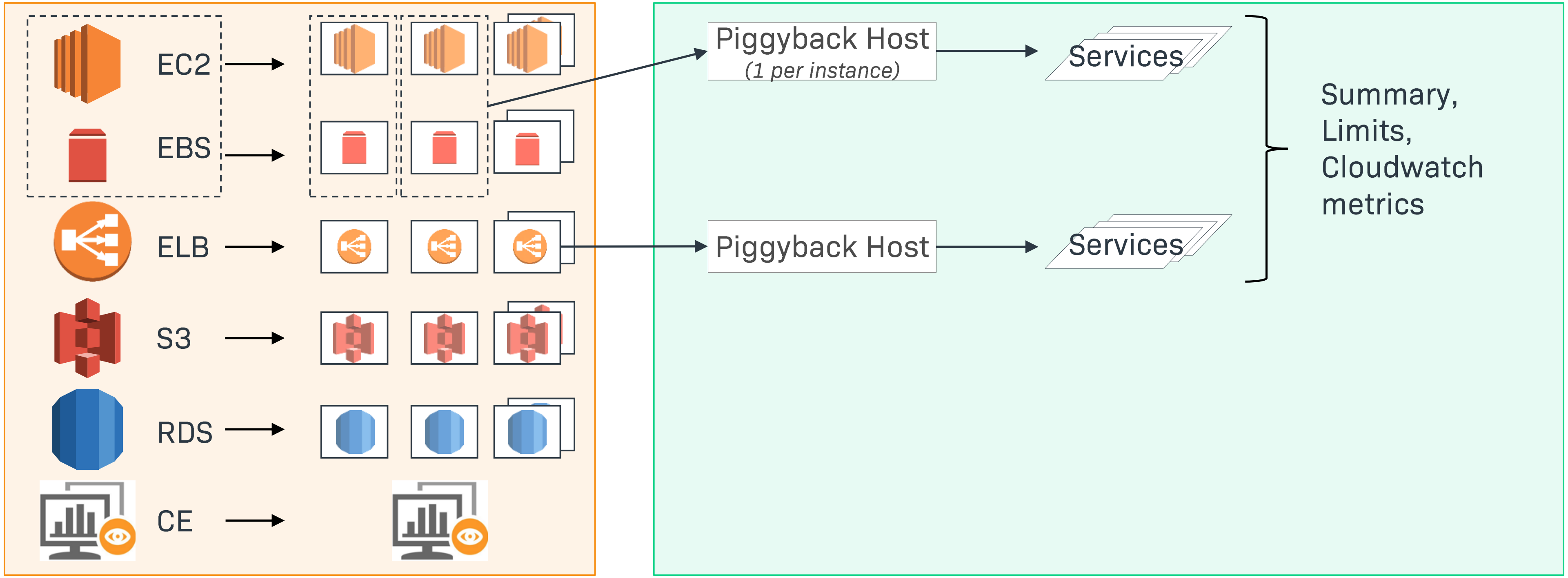
How it works: AWS Monitoring


AWS Resources and instances
... are treated in two different ways in checkmk
EC2
EBS
ELB
S3
RDS
CE
Piggyback Host
(1 per instance)
Piggyback Host

How it works: AWS Monitoring

AWS Resources and instances

… are treated in two different ways in check**mk**



EC2

EBS

ELB

S3

RDS

CE

Piggyback Host
(1 per instance)

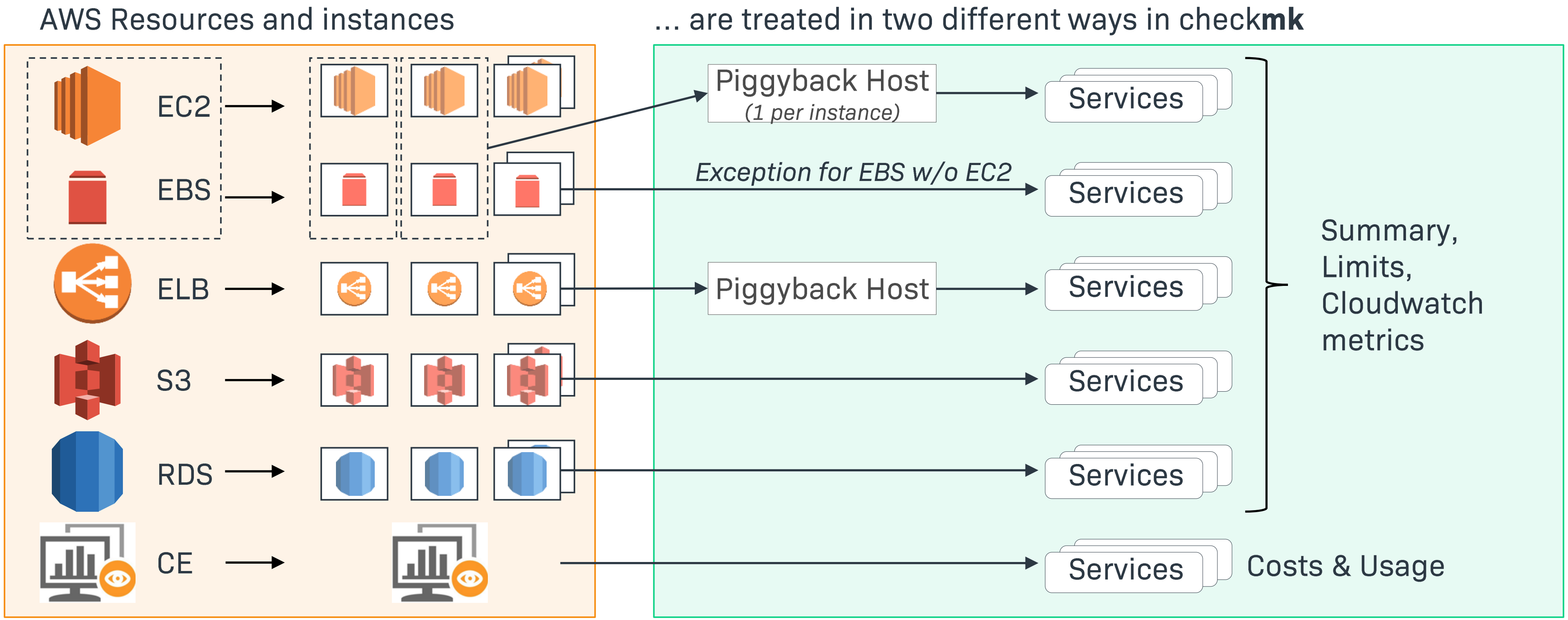
Services

Piggyback Host

Services

Summary,
Limits,
Cloudwatch
metrics

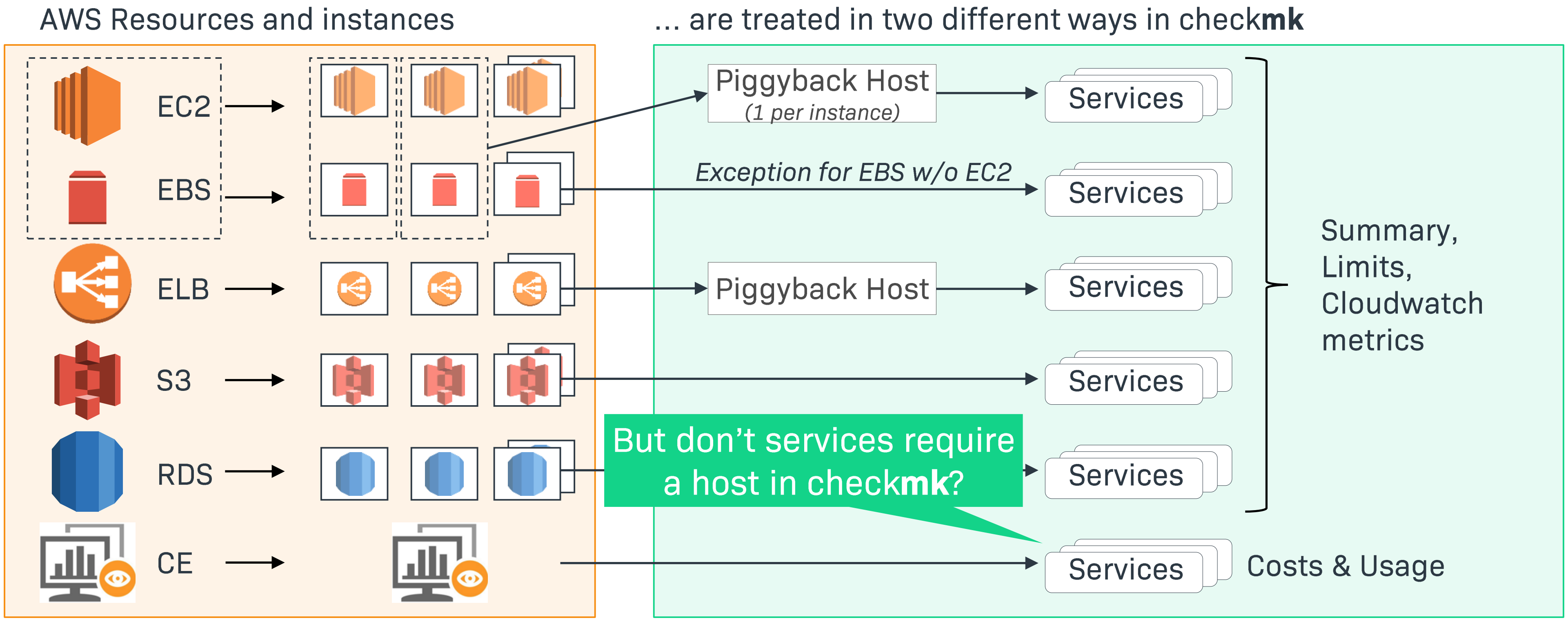
How it works: AWS Monitoring



tribe29

AWS MONITORING

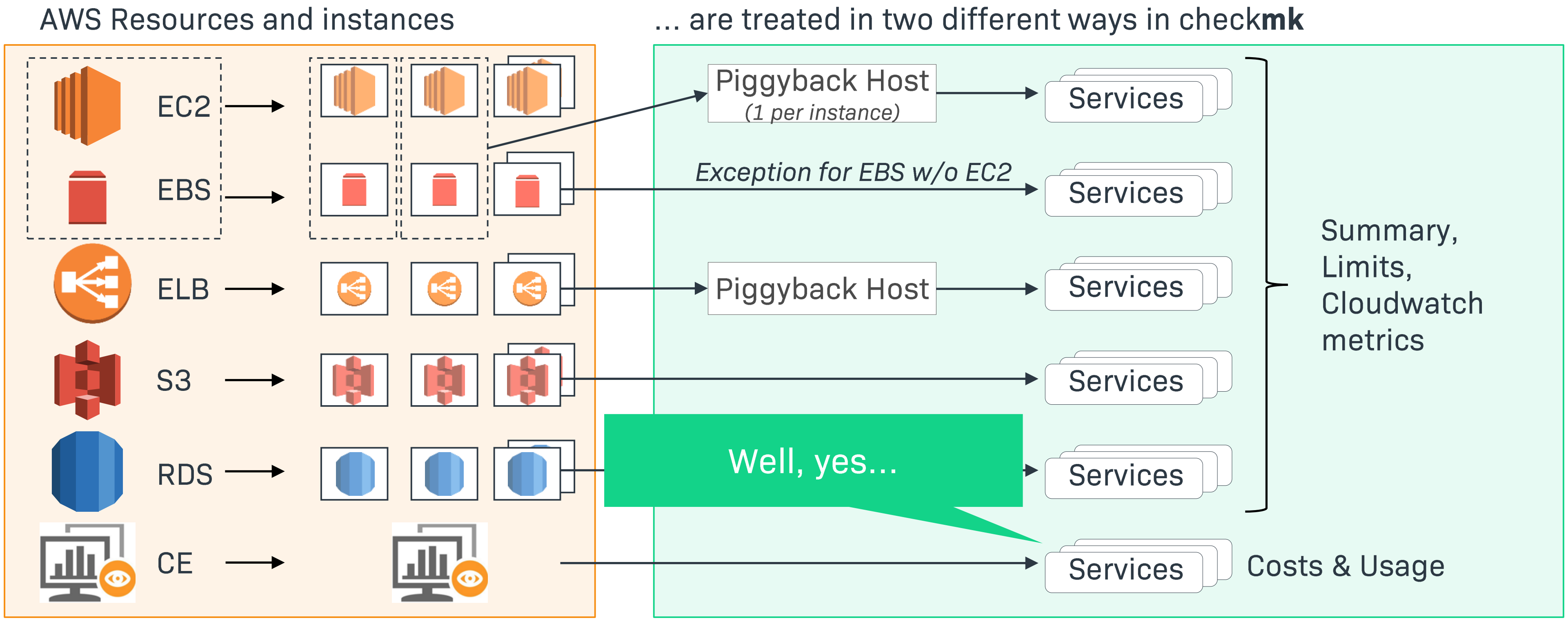
How it works: AWS Monitoring



But don’t services require a host in checkmk?

tribe29

How it works: AWS Monitoring

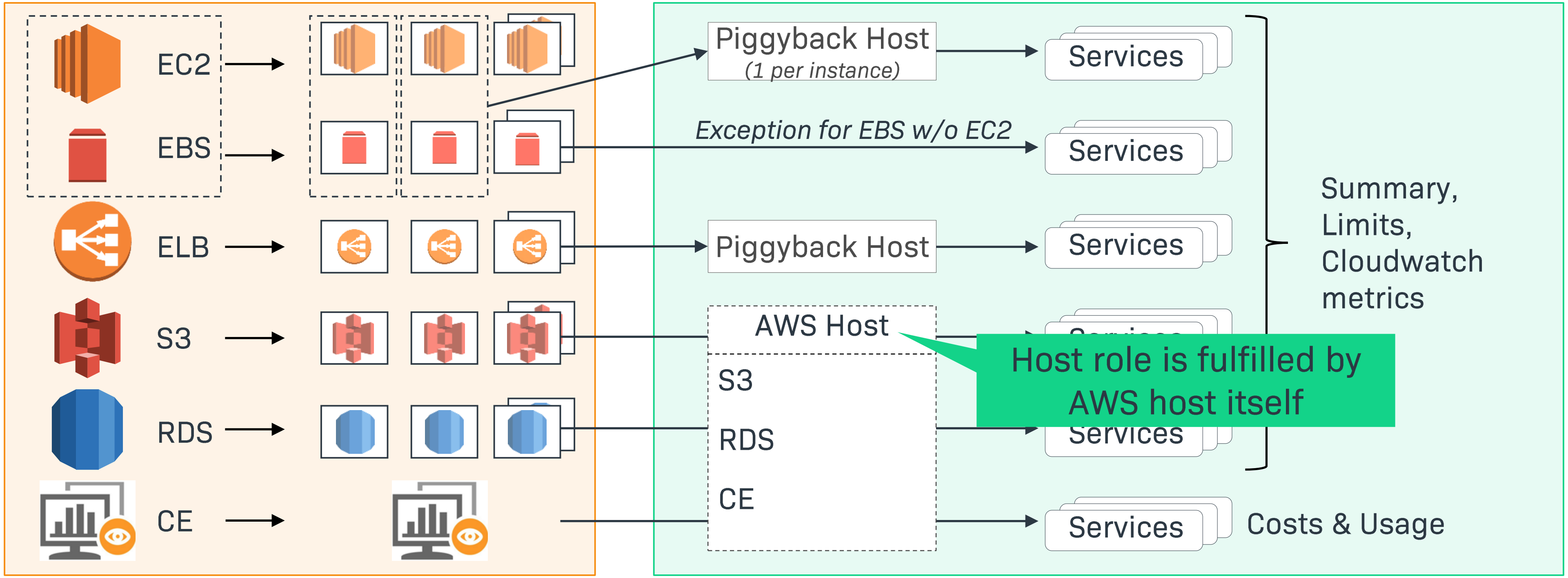

AWS Resources and instances
... are treated in two different ways in checkmk
EC2
EBS
ELB
S3
RDS
CE
Piggyback Host
(1 per instance)
Services
Exception for EBS w/o EC2
Services
Piggyback Host
Services
Services
Services
Services
Summary, Limits, Cloudwatch metrics
Costs & Usage
Well, yes...

AWS MONITORING

How it works: AWS Monitoring

AWS Resources and instances

... are treated in two different ways in check**mk**



EC2

EBS

ELB

S3

RDS

CE

Piggyback Host

(1 per instance)

Exception for EBS w/o EC2

Piggyback Host

AWS Host

S3

RDS

CE

Services

Services

Services

Services

Services

Services

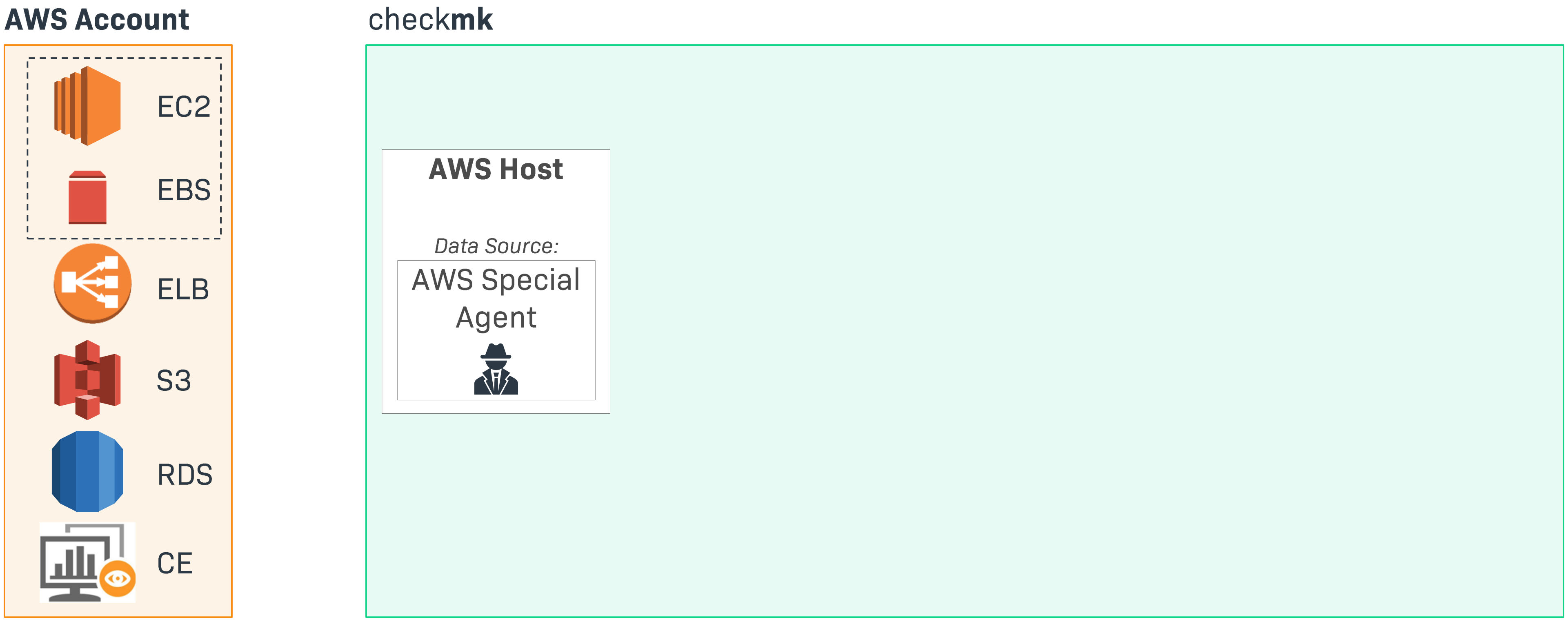
Summary, Limits, Cloudwatch metrics

Costs & Usage

Host role is fulfilled by AWS host itself

tribe29

How it works: AWS Monitoring



AWS Account

EC2

EBS

ELB

S3

RDS

CE

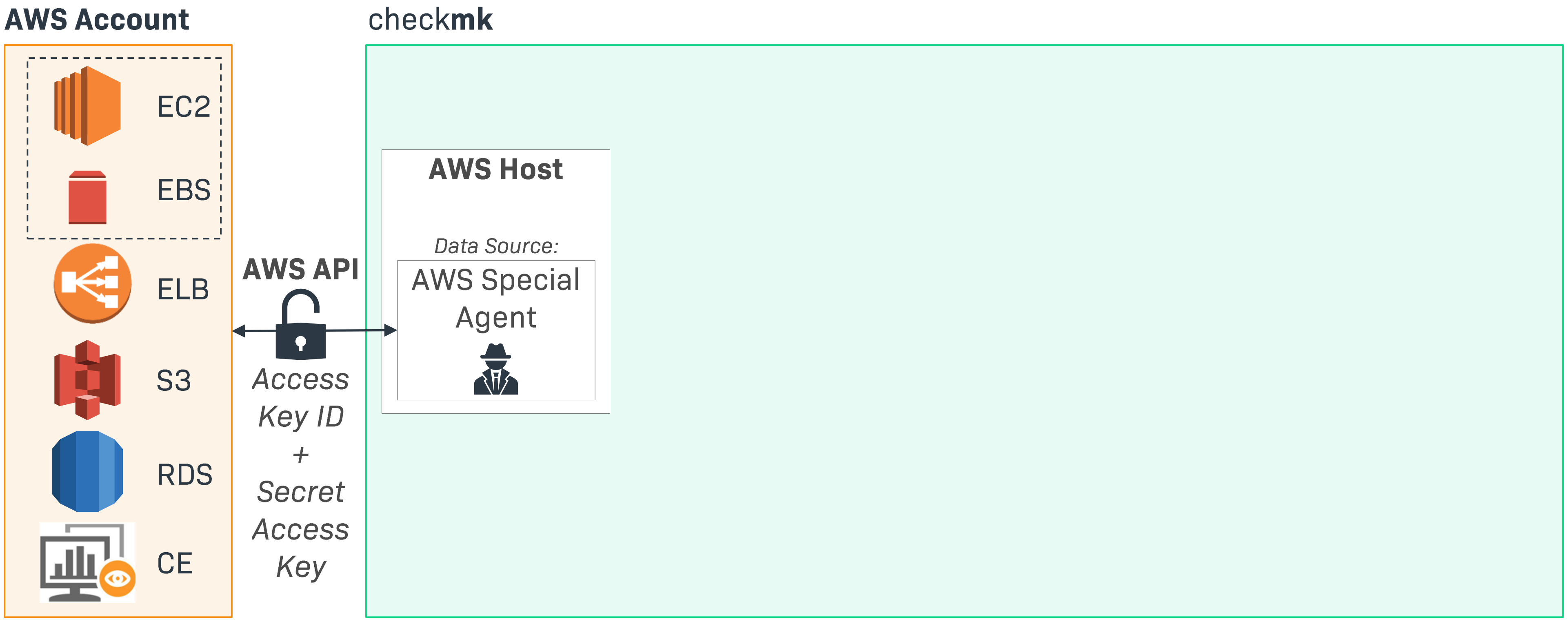
check**mk**

AWS Host

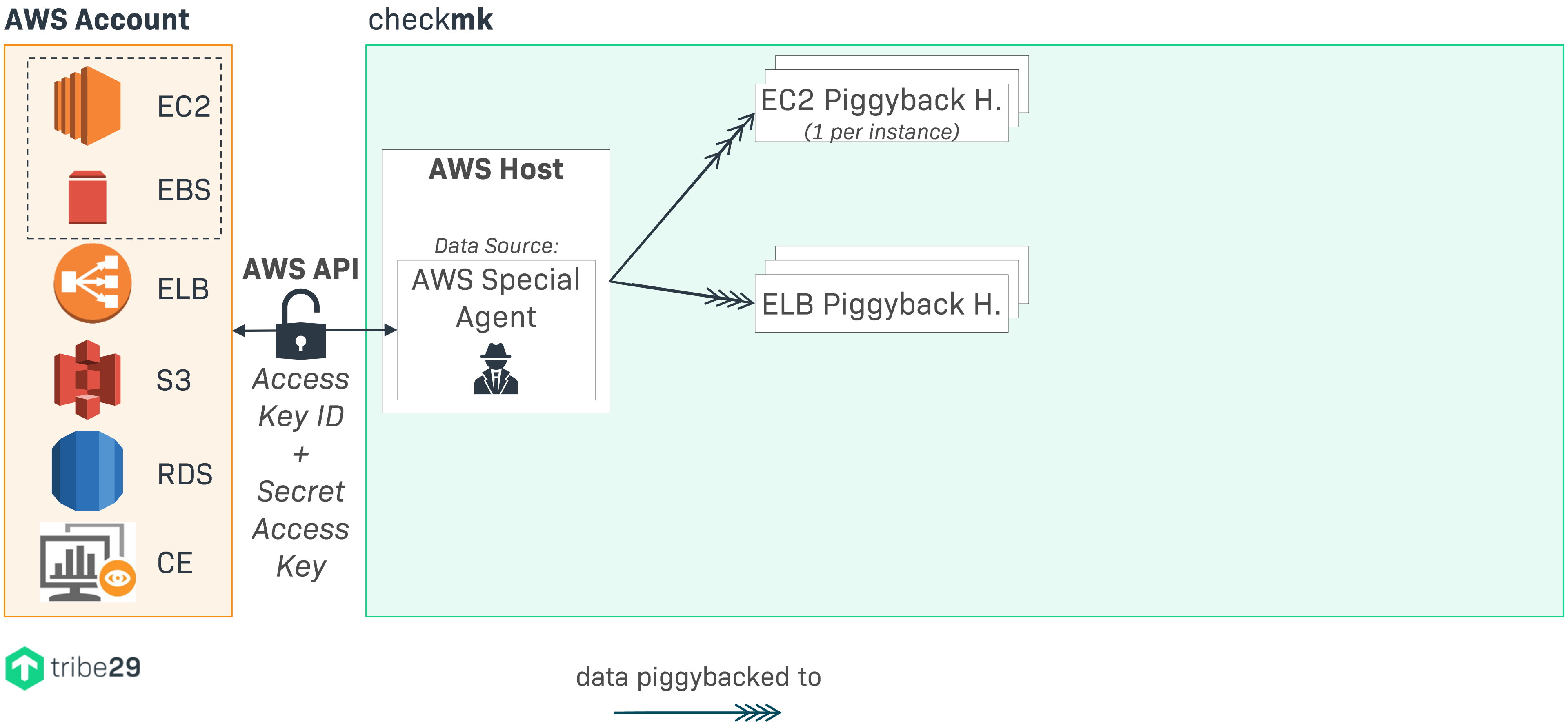
Data Source:

AWS Special Agent

How it works: AWS Monitoring



How it works: AWS Monitoring



AWS Account

EC2

EBS

ELB

S3

RDS

CE

AWS API

Access Key ID + Secret Access Key

checkmk

AWS Host

Data Source:

AWS Special Agent

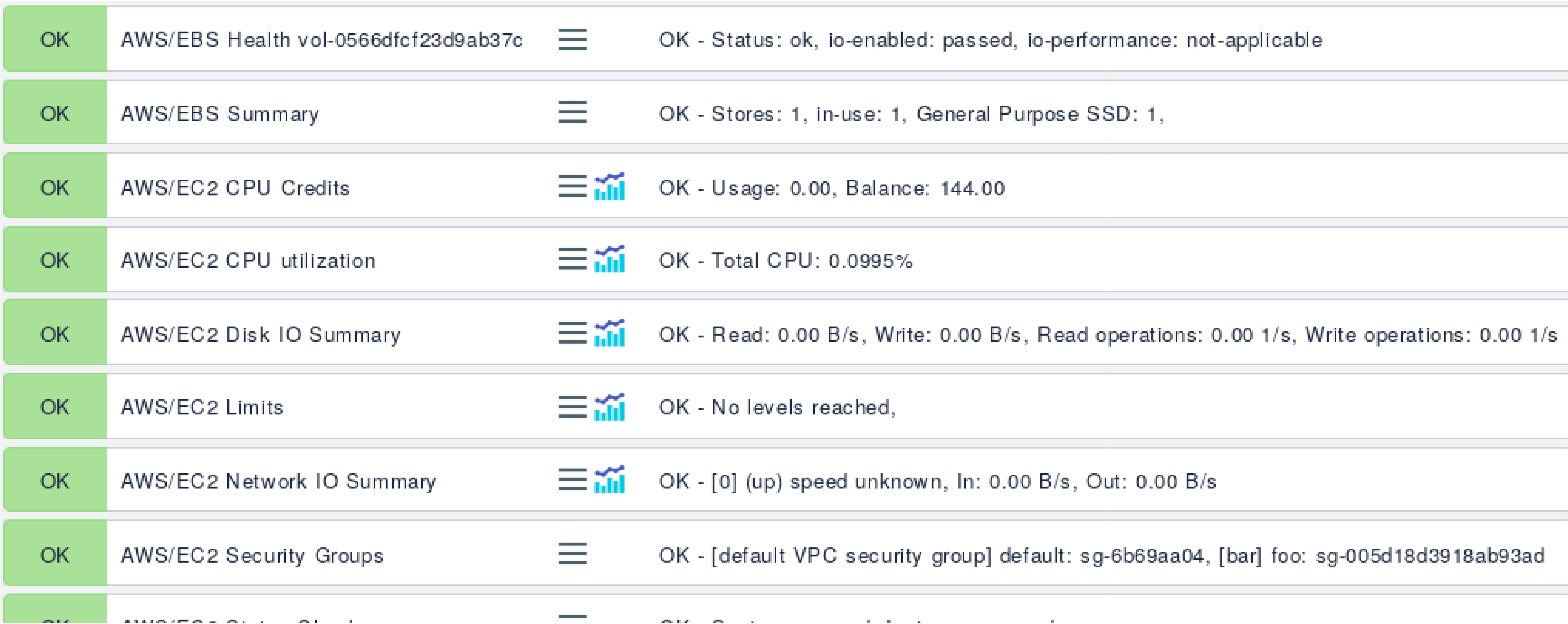
EC2 Piggyback H.

(1 per instance)

ELB Piggyback H.

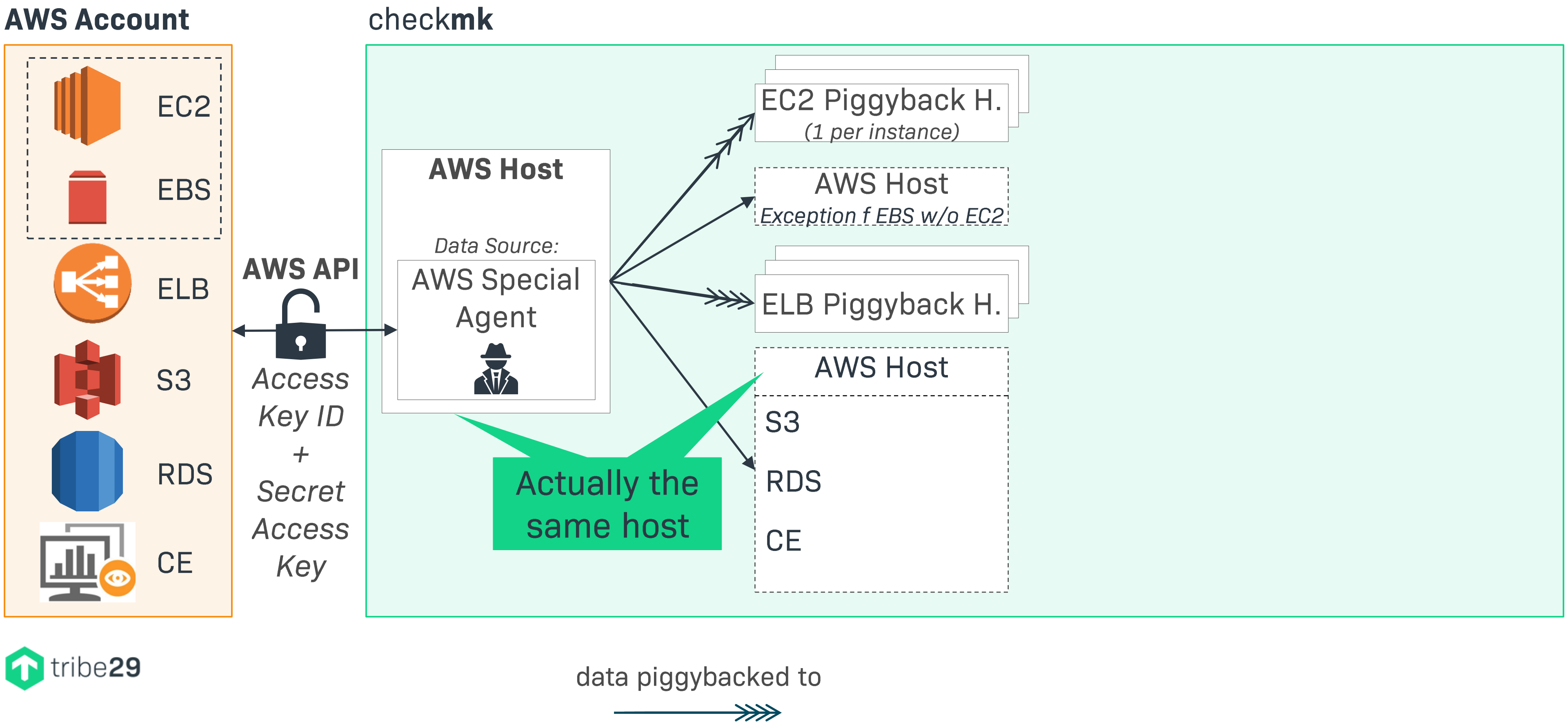
data piggybacked to

What it could look like: EC2 (+EBS) Piggyback Host

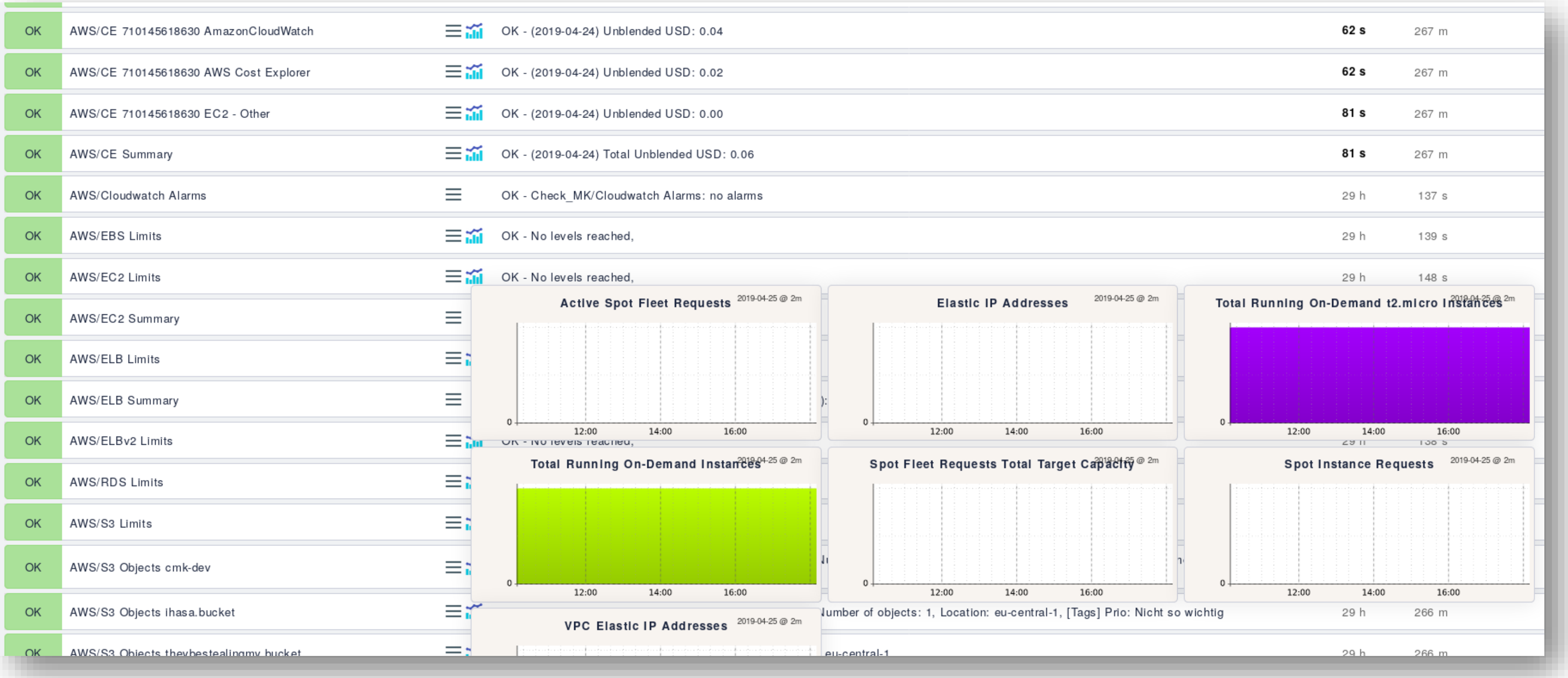


OK	AWS/EBS Health vol-0566dfcf23d9ab37c	☰	OK - Status: ok, io-enabled: passed, io-performance: not-applicable
OK	AWS/EBS Summary	☰	OK - Stores: 1, in-use: 1, General Purpose SSD: 1,
OK	AWS/EC2 CPU Credits	☰ 📊	OK - Usage: 0.00, Balance: 144.00
OK	AWS/EC2 CPU utilization	☰ 📊	OK - Total CPU: 0.0995%
OK	AWS/EC2 Disk IO Summary	☰ 📊	OK - Read: 0.00 B/s, Write: 0.00 B/s, Read operations: 0.00 1/s, Write operations: 0.00 1/s
OK	AWS/EC2 Limits	☰ 📊	OK - No levels reached,
OK	AWS/EC2 Network IO Summary	☰ 📊	OK - [0] (up) speed unknown, In: 0.00 B/s, Out: 0.00 B/s
OK	AWS/EC2 Security Groups	☰	OK - [default VPC security group] default: sg-6b69aa04, [bar] foo: sg-005d18d3918ab93ad

How it works: AWS Monitoring

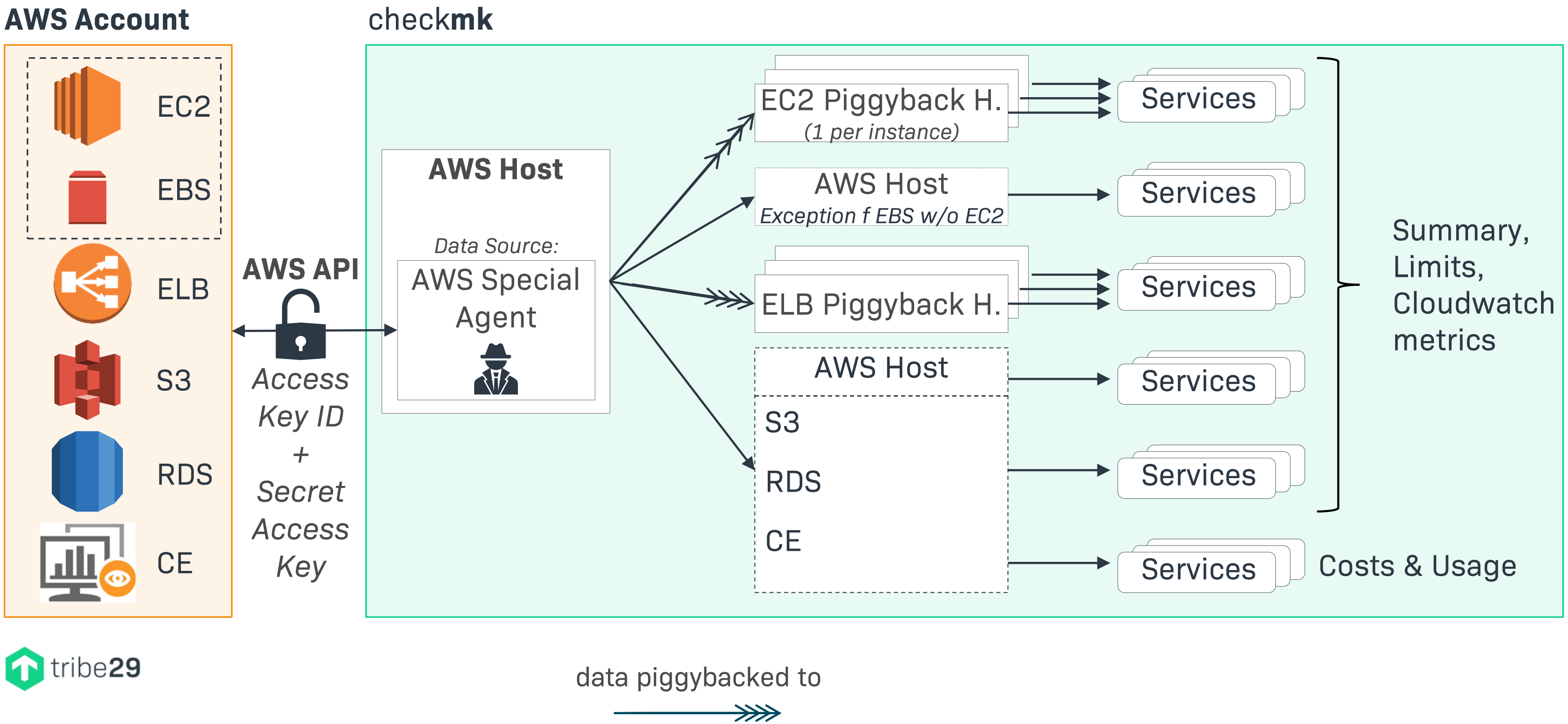


What it looks like: AWS Host



OK	AWS/CE 710145618630 AmazonCloudWatch	OK - (2019-04-24) Unblended USD: 0.04	**62 s**	267 m
OK	AWS/CE 710145618630 AWS Cost Explorer	OK - (2019-04-24) Unblended USD: 0.02	**62 s**	267 m
OK	AWS/CE 710145618630 EC2 - Other	OK - (2019-04-24) Unblended USD: 0.00	**81 s**	267 m
OK	AWS/CE Summary	OK - (2019-04-24) Total Unblended USD: 0.06	**81 s**	267 m
OK	AWS/Cloudwatch Alarms	OK - Check_MK/Cloudwatch Alarms: no alarms	29 h	137 s
OK	AWS/EBS Limits	OK - No levels reached,	29 h	139 s
OK	AWS/EC2 Limits	OK - No levels reached,	29 h	148 s
OK	AWS/EC2 Summary			
OK	AWS/ELB Limits			
OK	AWS/ELB Summary	):		
OK	AWS/ELBv2 Limits	OK - No levels reached,	29 h	138 s
OK	AWS/RDS Limits			
OK	AWS/S3 Limits			
OK	AWS/S3 Objects cmk-dev	Nu		
OK	AWS/S3 Objects ihasa.bucket	Number of objects: 1, Location: eu-central-1, [Tags] Prio: Nicht so wichtig	29 h	266 m
OK	AWS/S3 Objects theybestealingmy.bucket	eu-central-1	29 h	266 m

How it works: AWS Monitoring



Microsoft Azure

AZURE MONITORING

We already developed several checks for Azure

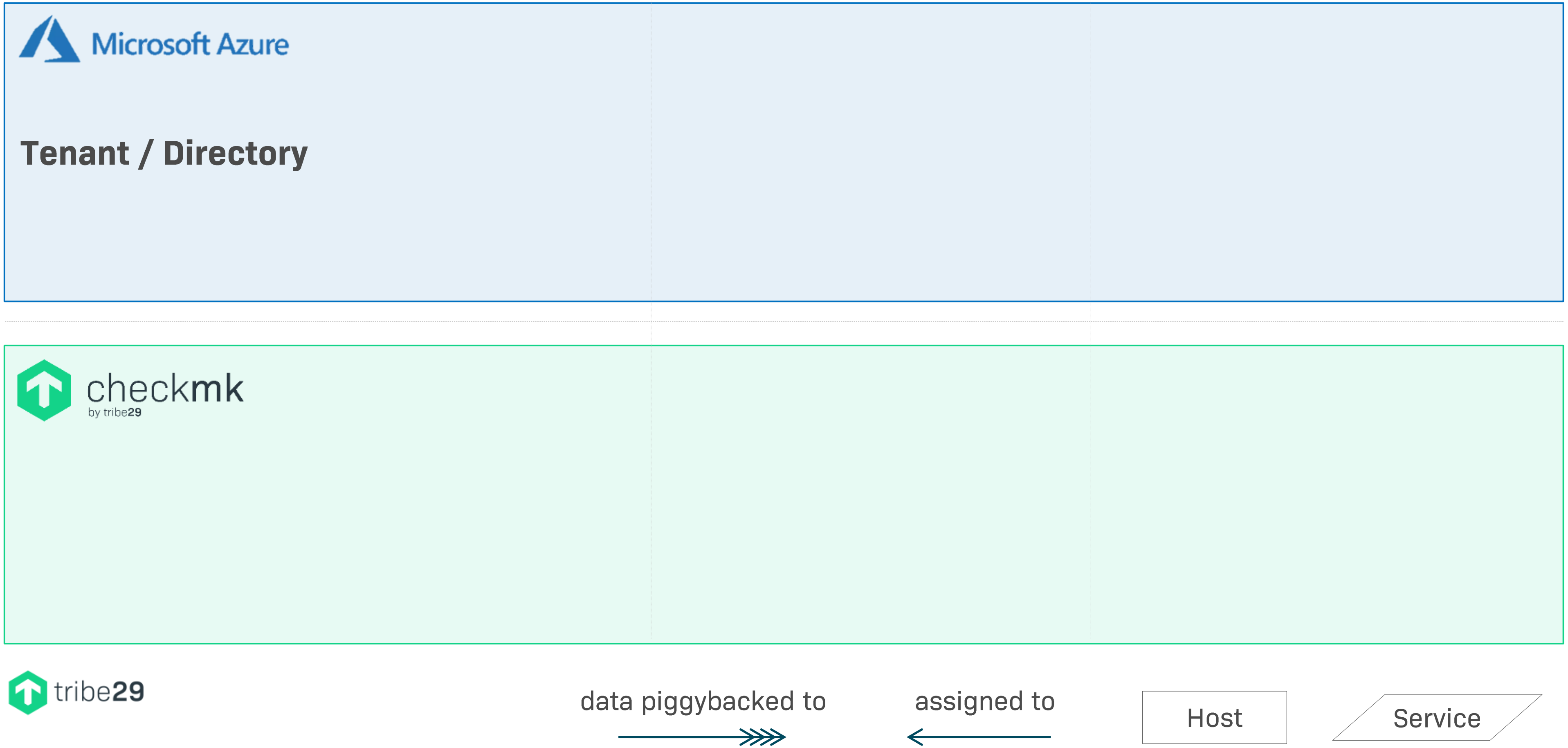
Microsoft Azure

Selection of Azure services

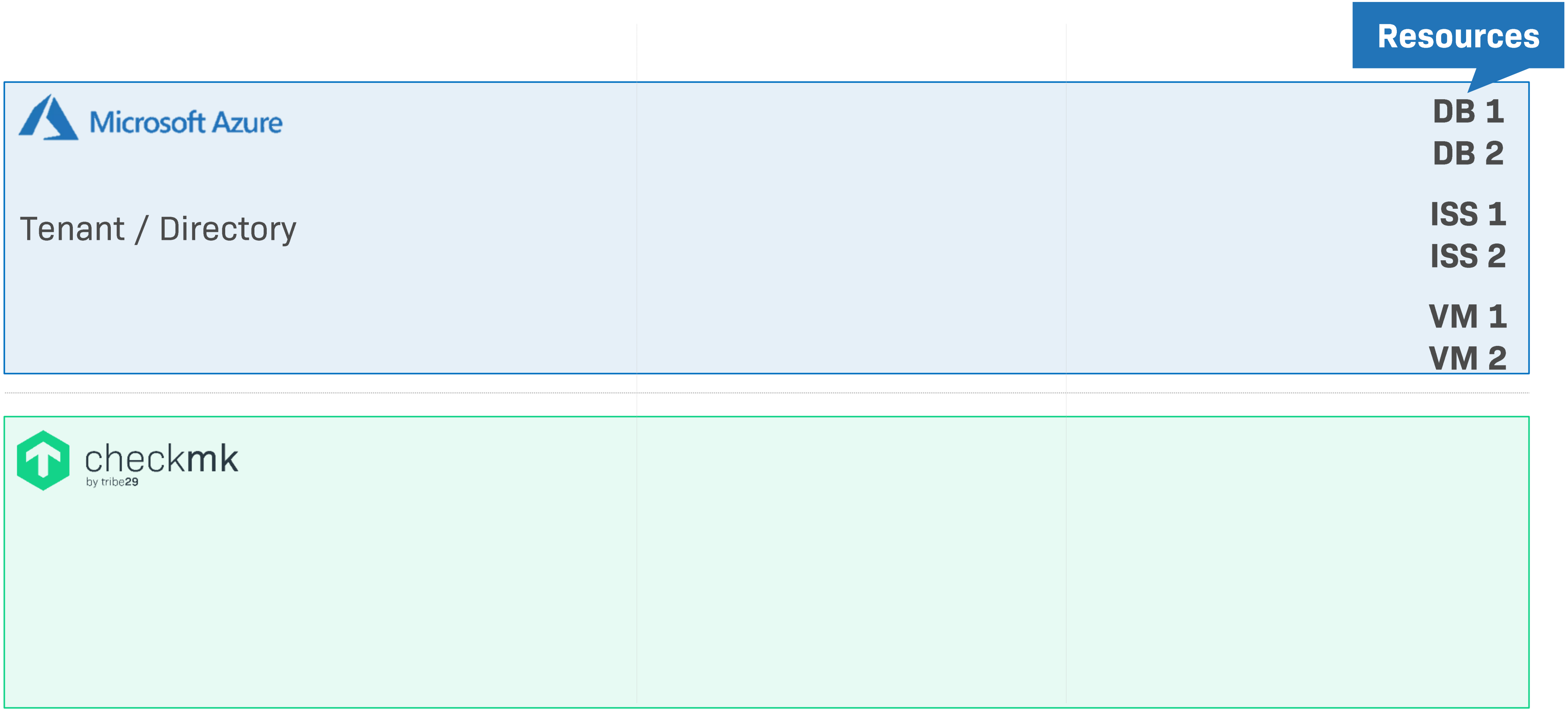
Compute & Storage	**Databases**	**Analytics & IoT**	**Networking**	**Identity & Security**	**Generic / Mgmt**
🟩 **Virtual Machines**	🟩 **Azure SQL DB**	⬜ Event Hub	🟦 Load Balancer	⬜ Active Directory	🟩 **Web Apps (Sites)**
🟩 **Blob Storage**	⬜ SQL Data Warehouse	⬜ Data Factory	🟩 **Virtual network**	⬜ ...	🟦 Backup
⬜ Functions	⬜ Azure Cosmos DB	⬜ IoT Hub	🟦 Azure DNS		🟦 Cost Management
⬜ Container Instances	⬜ Azure Cache	⬜ Stream Analytics	⬜ Network Watcher		⬜ Service Bus
⬜ ...	⬜ ...	⬜ ...	⬜ ...		⬜ ...

🟩 **Existing** 🟦 Planned ⬜ Future

How it works: MS Azure Monitoring



How it works: MS Azure Monitoring



Resources

Microsoft Azure

Tenant / Directory

DB 1
DB 2

ISS 1
ISS 2

VM 1
VM 2

checkmk by tribe29

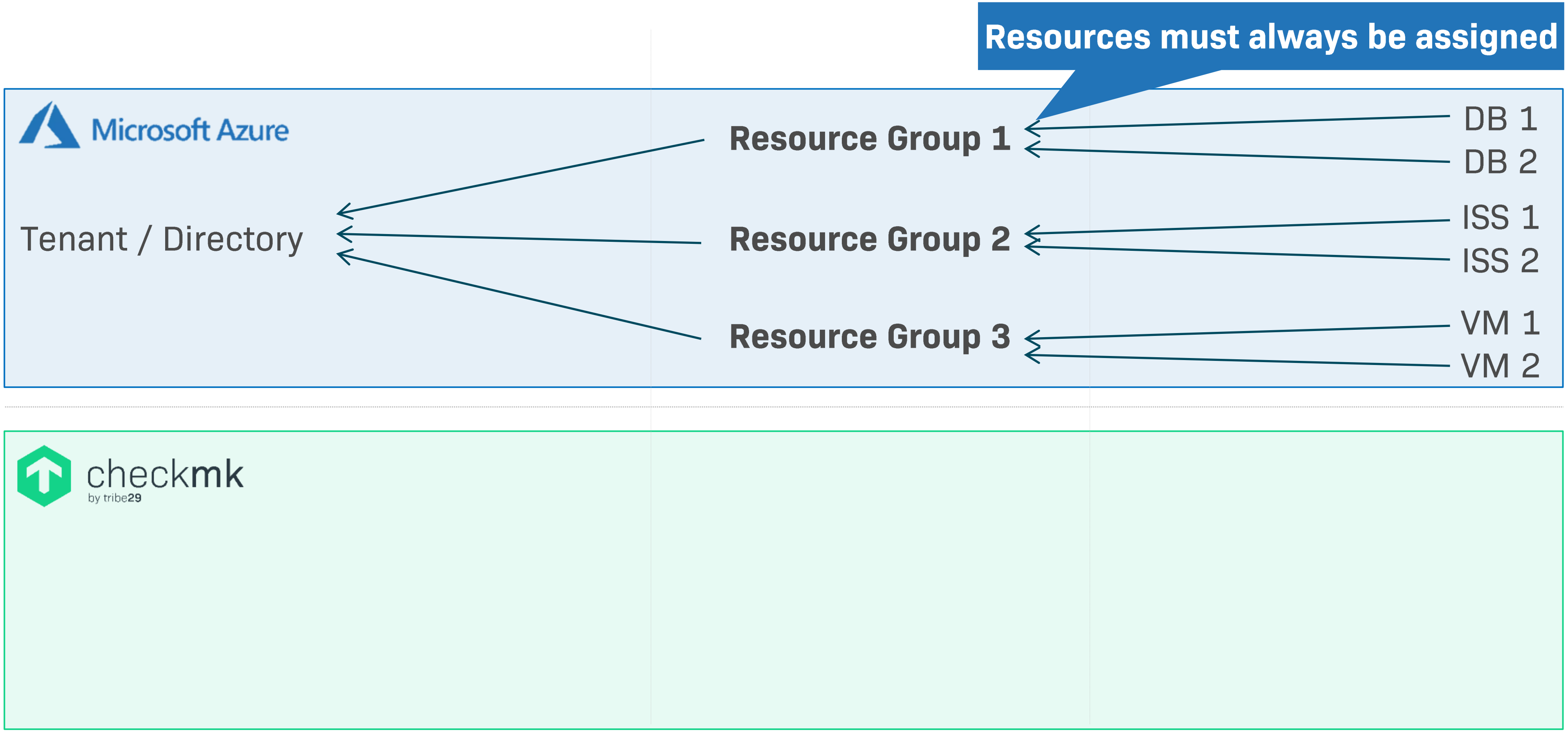
data piggybacked to ⋙

assigned to ←

Host

Service

How it works: MS Azure Monitoring



tribe29

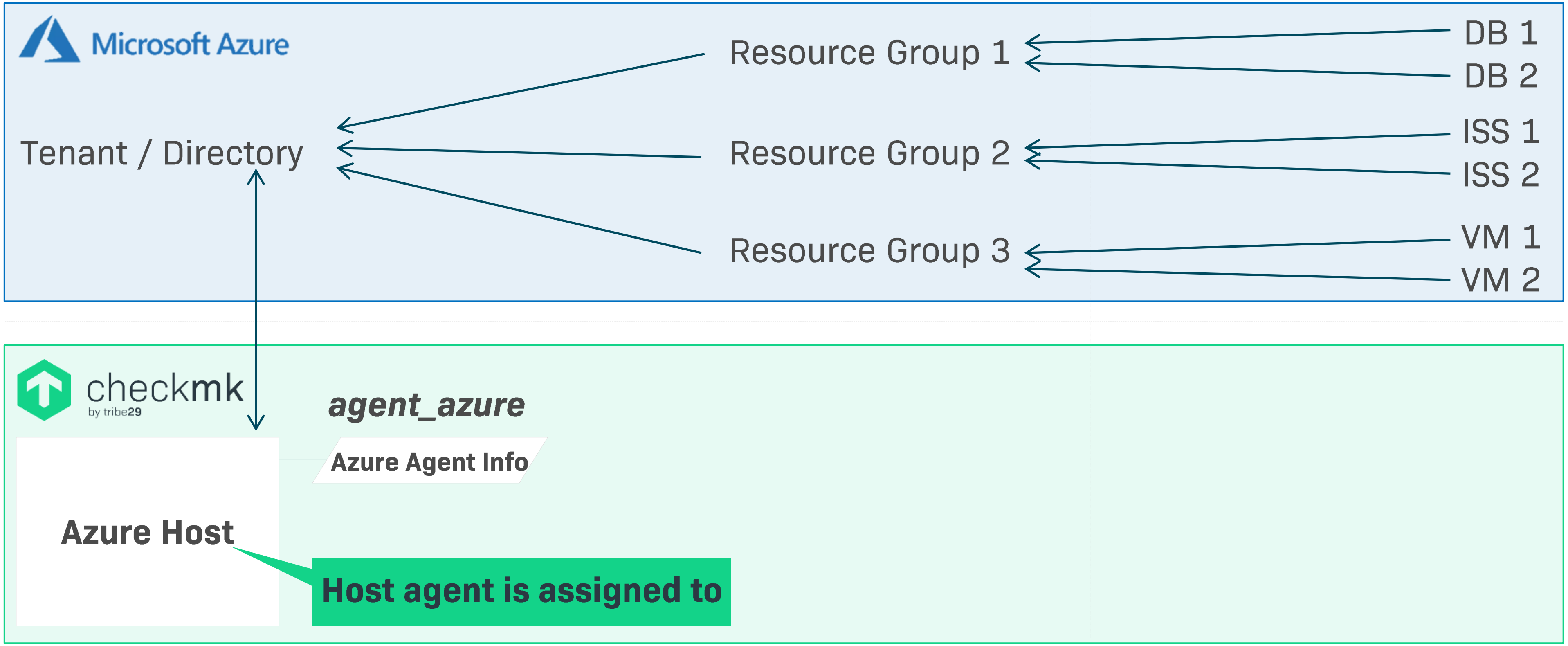
data piggybacked to ⋙

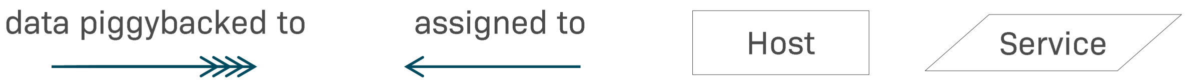
assigned to ←

Host

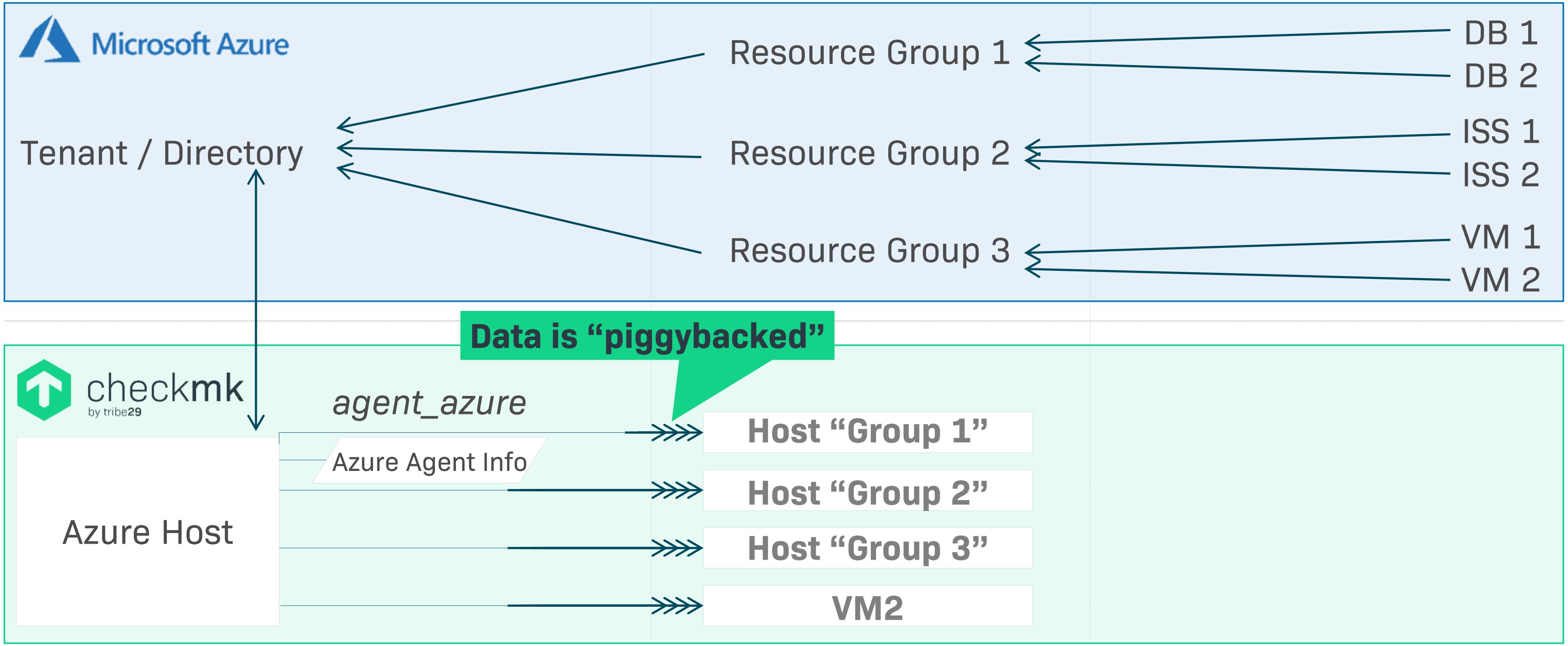
Service

How it works: MS Azure Monitoring





How it works: MS Azure Monitoring



Data is "piggybacked"

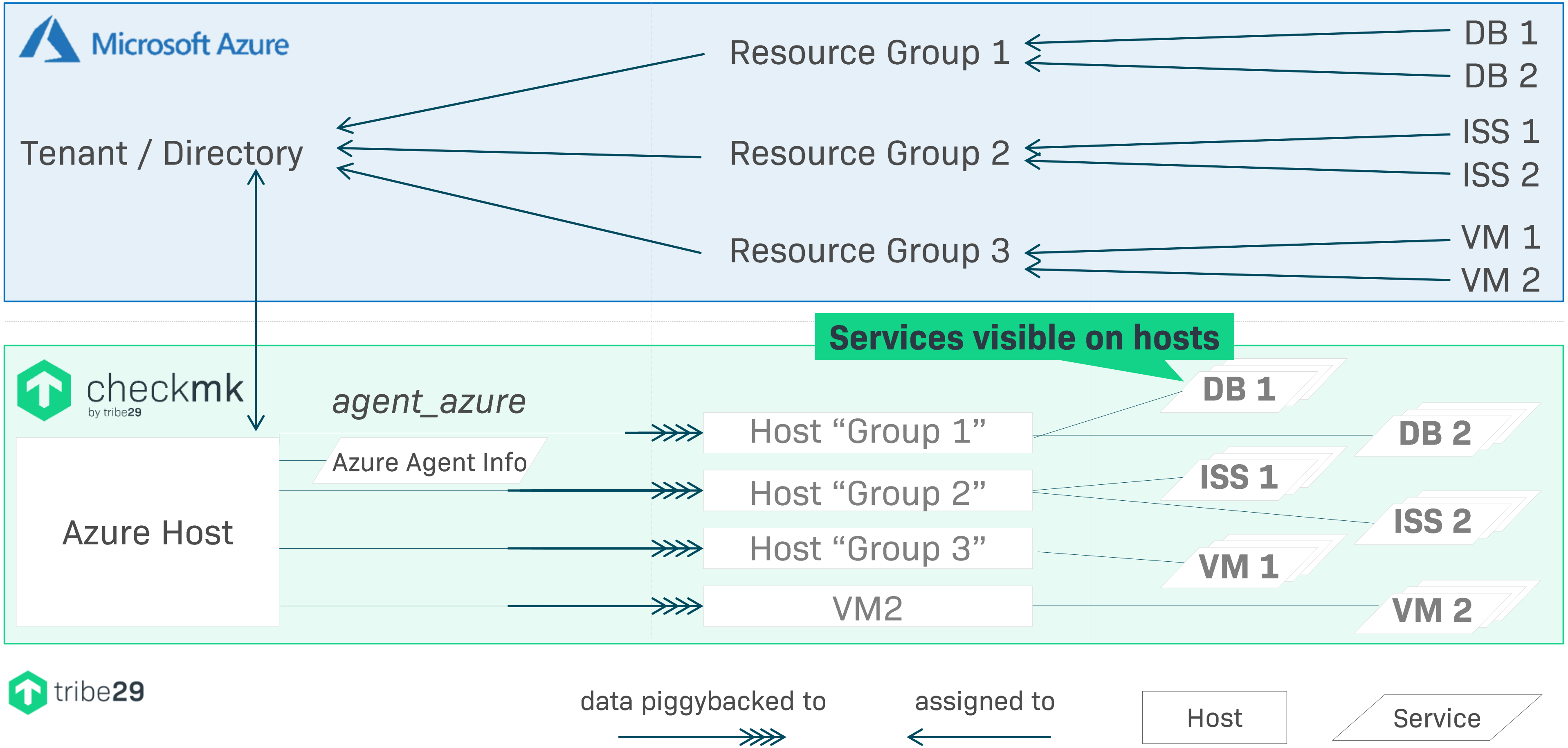
data piggybacked to

assigned to

Host

Service

How it works: MS Azure Monitoring



ROADMAP

What we‘re thinking about for the future

MORE

- More checks for more services
- Adding resource types based on customer demand & popularity

BETTER

- Improve simplicity & convenience, e.g.
  - Single Sign On
  - Pre-packaged monitoring config through Amazon Machine Image
  - One-click deployment of Azure monitoring with Microsoft Extension Manager

Thank you!

tribe29

tribe29 GmbH
Kellerstraße 29
81667 München
Deutschland

Web — tribe29.com
E-Mail — mail@tribe29.com