nic

Future Edition

The OMS Solutions Bakery

How to build a custom OMS solution in 60 minutes

Marcel Zehner | Corporate Ambassador itnetX

Microsoft Regional Director (RD)

Microsoft Most Valuable Professional (MVP)

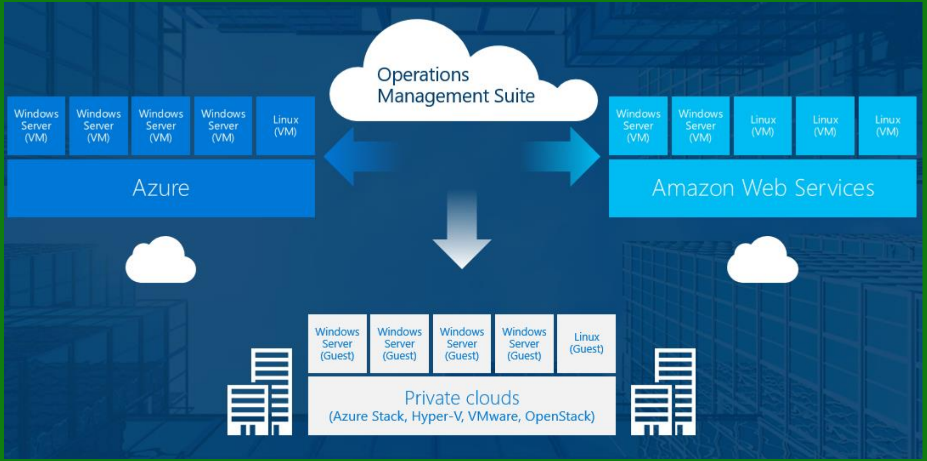
marcelzehner | @marcelzehner

About Me

- itnetX Global Alliance
- Switzerland
- Microsoft MVP & RD
- Microsoft Azure addict
- IT & tech geek
- Speaker & blogger
- Community supporter
- World traveller
- Group fitness fanatic

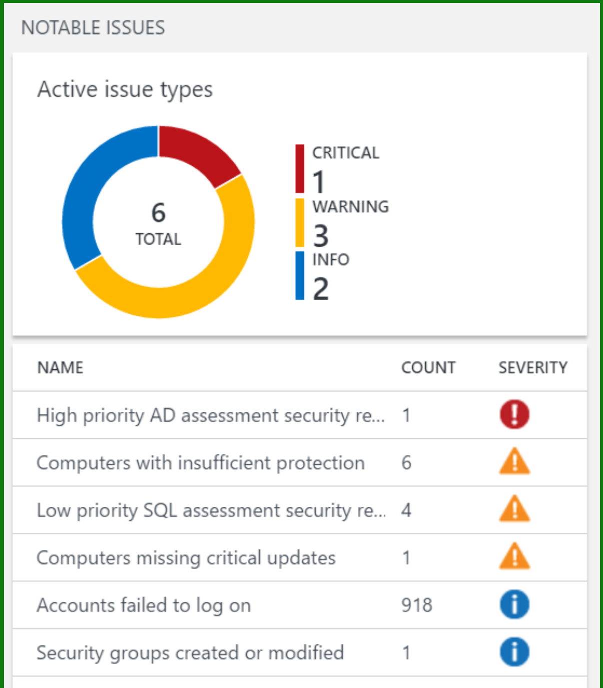
Quick OMS overview

- Hybrid Cloud Management
  - Azure Log Analytics
  - Azure Automation
  - Azure Security Center
  - Azure Recovery Services
- Extensibility
  - Can be extended by adding solutions to the log analytics workspace



What's in a solution?

- Bundled, grouped resources to extend the functionality
  - Gallery/Marketplace
  - Custom
- A solution CAN contain ...
  - a mechanism to collect specific data
  - queries to extract information from the collected data
  - views to aggregate and visualize data
  - alerts for notifications and remediation



NOTABLE ISSUES

Active issue types

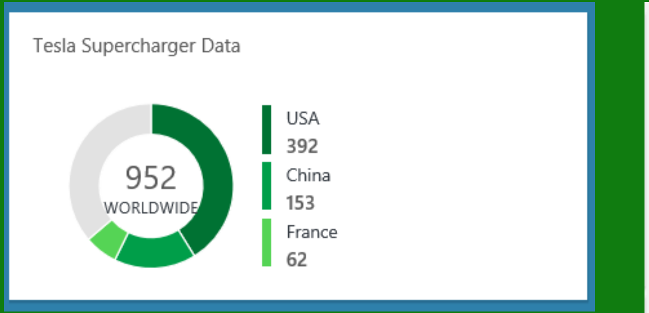
CRITICAL
1

WARNING
3

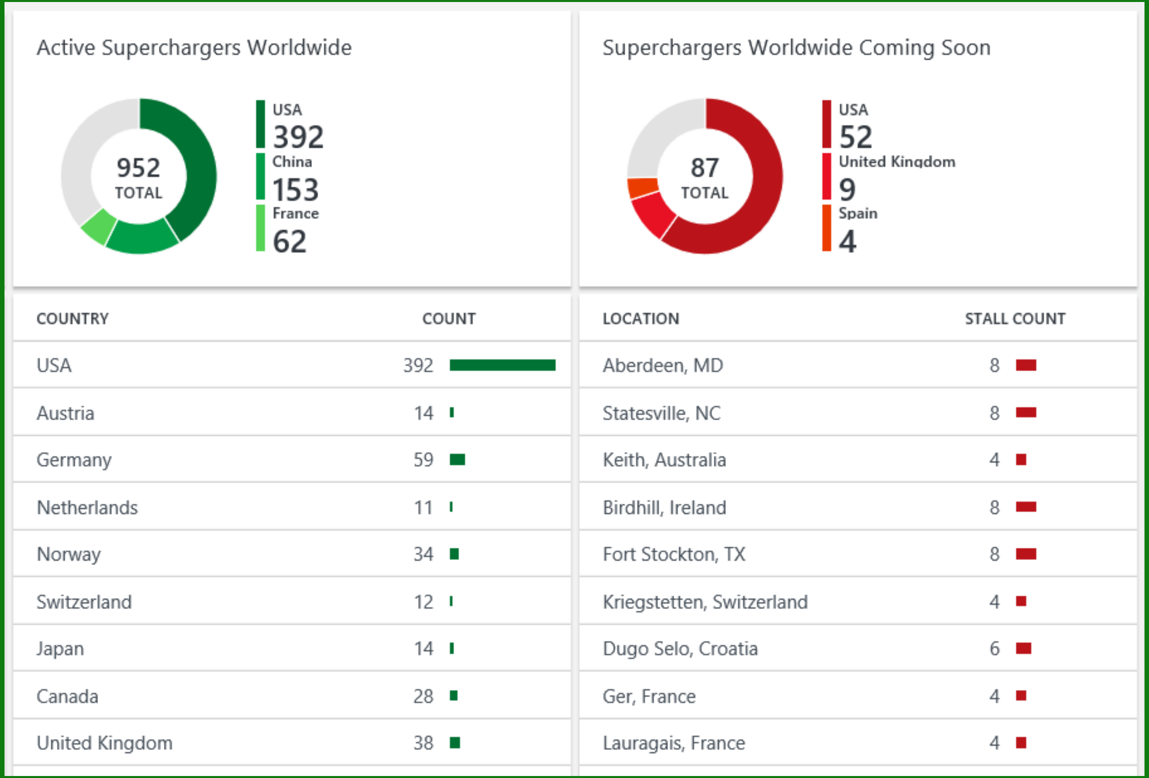
INFO
2

NAME	COUNT	SEVERITY
High priority AD assessment security re...	1	❗
Computers with insufficient protection	6	⚠
Low priority SQL assessment security re...	4	⚠
Computers missing critical updates	1	⚠
Accounts failed to log on	918	ℹ
Security groups created or modified	1	ℹ

No solution for your workload?



Tesla Supercharger Data



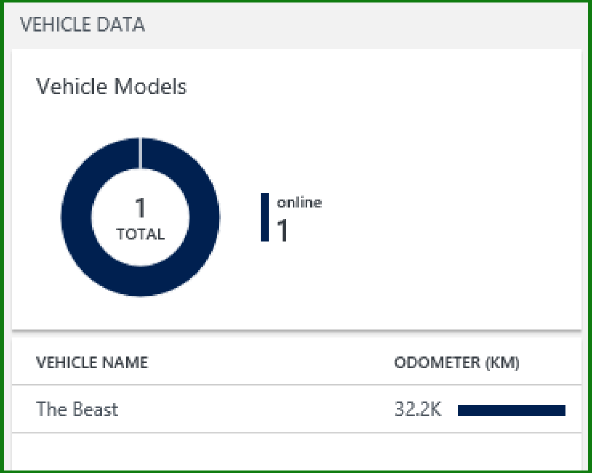
Active Superchargers Worldwide

COUNTRY	COUNT
USA	392
Austria	14
Germany	59
Netherlands	11
Norway	34
Switzerland	12
Japan	14
Canada	28
United Kingdom	38

Superchargers Worldwide Coming Soon

LOCATION	STALL COUNT
Aberdeen, MD	8
Statesville, NC	8
Keith, Australia	4
Birdhill, Ireland	8
Fort Stockton, TX	8
Kriegstetten, Switzerland	4
Dugo Selo, Croatia	6
Ger, France	4
Lauragais, France	4

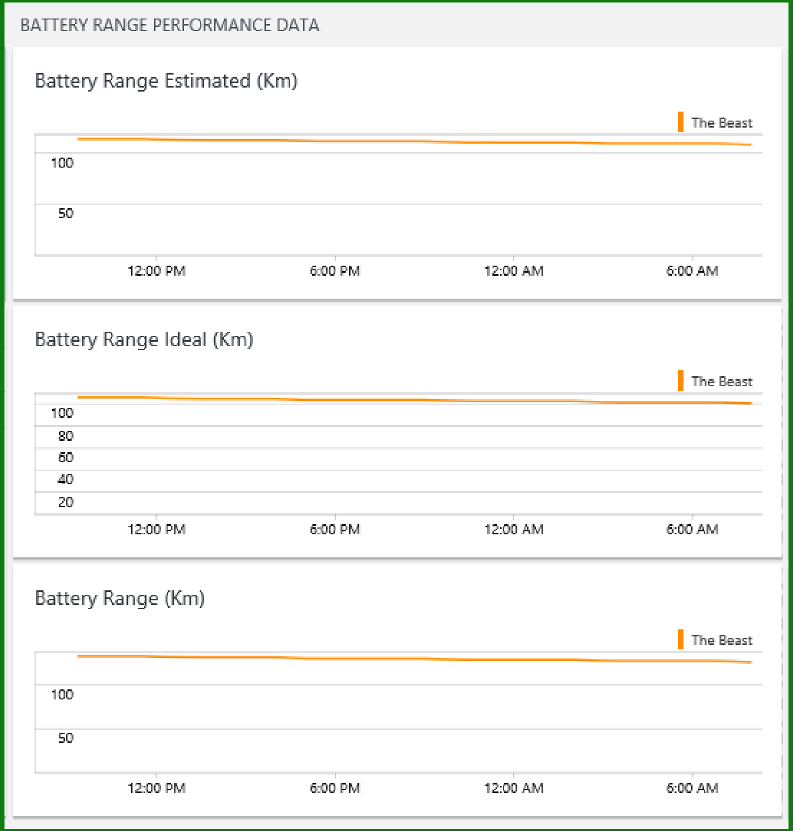
No solution for your workload?



VEHICLE DATA

Vehicle Models

VEHICLE NAME	ODOMETER (KM)
The Beast	32.2K



BATTERY RANGE PERFORMANCE DATA

Battery Range Estimated (Km)

Battery Range Ideal (Km)

Battery Range (Km)

Recipe

What is the goal of the solution?

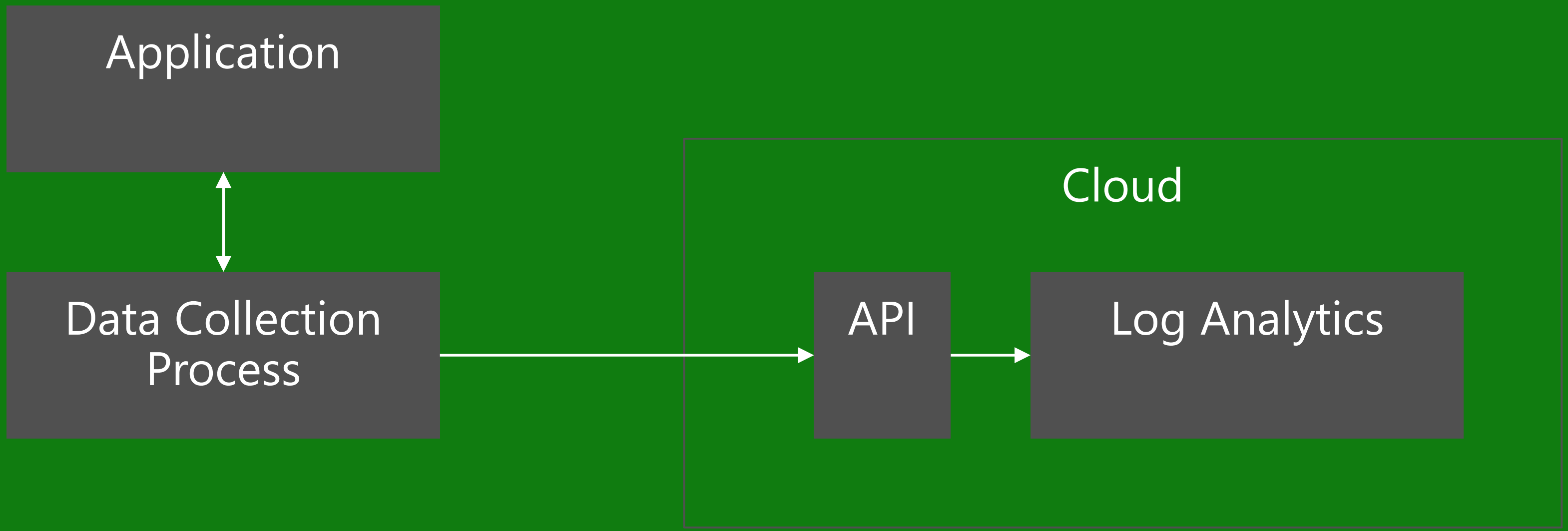
- Ask yourself …
  - Who are the consumers of the solution?
  - What benefit do they expect from the solution?
  - What data needs to be collected?
  - How can the data be presented in an easy-to-understand way?

How will the data be collected?

- Depends on the source system
- Possible collection methods
  - Web service
  - API/SDK
  - Log files
  - Etc.

How is the data ingested into OMS?

- Submit data to the HTTP Data Collector API
- Data needs to be in JSON format



Application

Data Collection Process

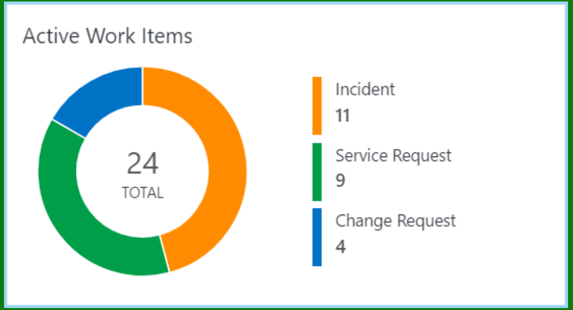
Cloud

API

Log Analytics

My recipe for this demo 1/2

- Visualize tickets from System Center Service Manager
- Stakeholders
  - Operations team, team leaders and managers
- Goals
  - Get immediate overview of active tickets together with other operations insights
  - See most important tickets immediately
  - Escalated tickets
  - SLO warnings/breaches



My recipe for this demo 2/2

- Data will be collected from Service Manager CMDB
  - PowerShell collection script
  - Schedules with Windows Task Scheduler to run every hour
- Collection script will run on a regular basis (every hour)
- View to visualize the collected data

Bake

Collect data

- Create the collection script
  - PowerShell or equivalent
- Schedule
  - Scheduled task, SMA, Azure Automation, Windows Service etc.
  - Triggered based on a specific event or happening in the app
- If properties in the source system has no value, they are not created on the log analytics event
  - Problem when grouping events by field value
  - Example: “Classification” of an Incident could be empty
  - Workaround: Check during collection and use “not set” value (or similar)

Prepare data for ingestion

- Example with PowerShell
  - Collect Data
  - Create a PowerShell object for every active ticket
  - Add ticket property values
  - Pipe to “ConvertTo-Json”

```
[
    {
    “Title”: “Server 2 has a physical disk failure”,
    “Escalated”: true,
    “Classification”: ”Server Hardware”,
    “AffectedUser”: “Jack Jones”
    },
    {
    “Title”: "Cannot print from desktop”,
    “Escalated”: false,
    “Classification”: “Not Set”,
    “AffectedUser”: “Megan Masterson”
    }
]
```

Ingest data

- Use 2 functions provided by Microsoft
  - Build authorization signature
  - Ingest data
- Option: Use PowerShell module «OMSIngestionAPI»

https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-data-collector-api

Query data and create alerts

- Create queries
  - Easy-to-learn query language
  - Log Search or Advanced Analytics portal
- Throw alerts
  - When specific events are collected/discovered
  - Send mail, trigger webhooks or runbooks, use ITSM connector

Visualize data

- Create views
  - Use view designer together with your queries
  - Main tile
  - Specific views (drill down)
- Use data from Power BI
  - Create Power Query (M language)
  - Use Power BI Desktop to access data
  - Design report, then publish to Power BI Service
  - Create Power BI reports/dashboards

Solution development best practices

- Use versioning and source control
  - Visual Studio Online
  - TFS
  - Git
  - Etc.
- Test your collection script inside out
  - Include error handling
- Check if the ingested data is correct
- Check views (special focus: selected object queries)

My recipe for this demo

- Create a PowerShell script that collects tickets from System Center Service Manager
- Implement PowerShell script as a scheduled task
- Create queries
- Create view with the view designer
- Create alerts
- Create query for Power BI

Live Demo «Bake»

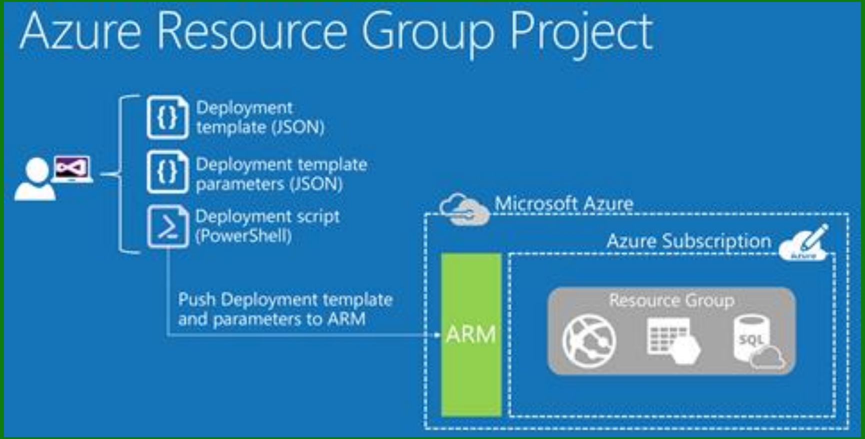
Deliver

Deployment

- The full solution has multiple components
  - OMS Log Analytics workspace
  - OMS Dashboard view
  - Collection script
  - Alerts, queries and more …
- Use Azure Resource Manager (ARM) for deployment
  - Declarative approach
  - Deliver all components of the solution together (bundled)

Azure Resource Manager (ARM)

- Set of resource providers
- Accessed through APIs to manage Azure resources
- Declarative provisioning & lifecycle management
- Use appropriate tool to create JSON file
  - Visual Studio with Azure SDK
  - Visual Studio Code
- Use Azure quick start templates on Github to start your journey


Azure Resource Group Project
Deployment template (JSON)
Deployment template parameters (JSON)
Deployment script (PowerShell)
Push Deployment template and parameters to ARM
Microsoft Azure
Azure Subscription
ARM
Resource Group
SQL

Deploy ARM Templates

- Deploy using Visual Studio (with Azure SDK)
- Deploy using PowerShell
  - New-AzureRMResourceGroupDeployment
- Azure CLI
  - azure group deployment create
- Use template deployment in Azure Portal
- Publish to Azure Marketplace

My recipe for this demo

- Create an ARM template
  - Log analytics workspace
  - 1 view
  - 2 queries
  - Create solution
  - Deploy whole solution

Live Demo «Deliver»

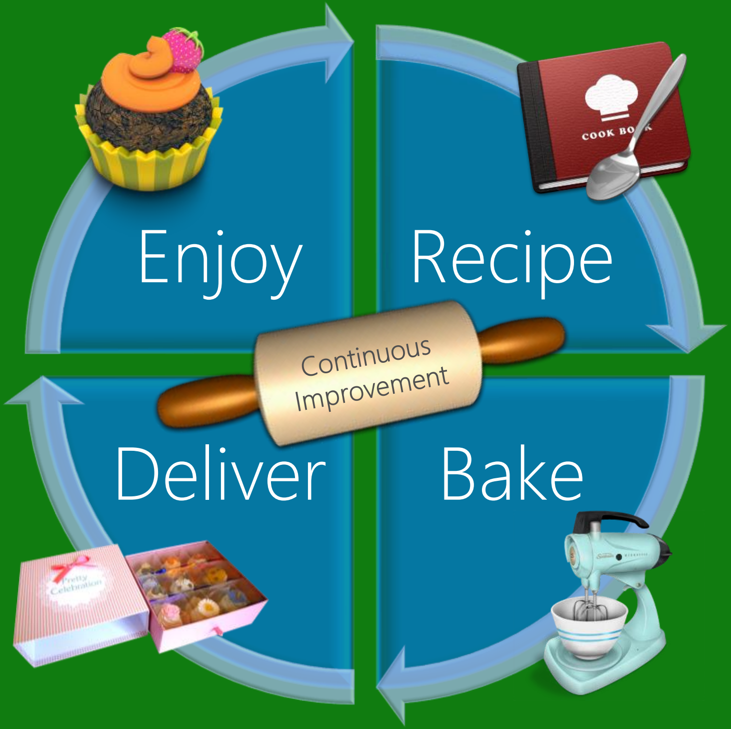
Enjoy

Enjoy

- Enjoy your solution
- Ask consumers for feedback
- Improve the solution over and over again
  - Continuous improvement process

Recap

OMS Solution Lifecycle



Recipe

Bake

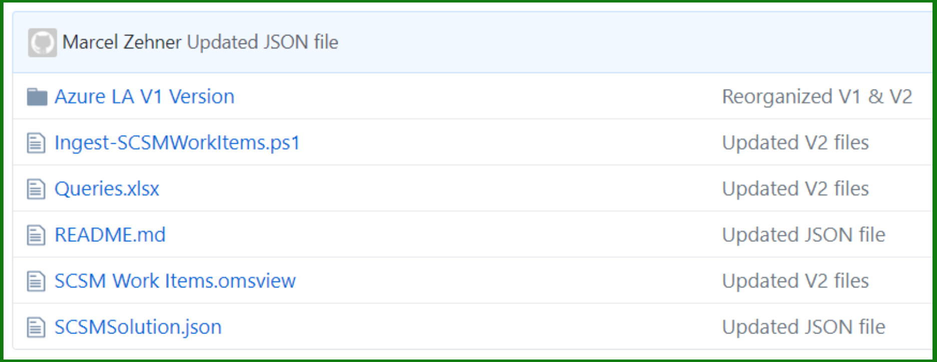
Deliver

Enjoy

Continuous Improvement

Download Example Files

- github.com/MarcelZehner/OMSServiceManagerSolution
  - Collection script
  - Excel file with queries
  - Exported OMS view
  - ARM template to deploy the solution



Marcel Zehner Updated JSON file

Azure LA V1 Version	Reorganized V1 & V2
Ingest-SCSMWorkItems.ps1	Updated V2 files
Queries.xlsx	Updated V2 files
README.md	Updated JSON file
SCSM Work Items.omsview	Updated V2 files
SCSMSolution.json	Updated JSON file

The OMS Solutions Bakery

How to build a custom OMS solution in 60 minutes

Marcel Zehner | Corporate Ambassador itnetX

Microsoft Regional Director (RD)

Microsoft Most Valuable Professional (MVP)

marcelzehner | @marcelzehner