



Hands-On Demo

Azure PaaS App

App Url: <https://salespoc.azurewebsites.net>

GitHub Url: <https://github.com/csdmichael/SalesPOC.UI>

Read Me: [Frontend UI for Sales POC App](#)

Prepared by:

Michael Yaacoub
Sr. Solution Engineer – AI Applications
Microsoft



Steps for building Sales POC

- 0. Walk-through Final Product**
- 1. Identify functional Requirements and non-functional Requirements**
- 2. Generate Architecture Diagram to satisfy requirements**
- 3. Build APIs and publish to Azure Container App**
- 4. Create new API Management Instance in Azure and Import Sales-API**
- 5. Create MCP Server in APIM**
- 6. Go to API Center -> Register APIM Instances for Dev, Testing, Prod for API discovery & Linting**
- 7. Go to AI Foundry -> Build an agent that uses MCP Server (exposed in APIM) as a tool**
- 8. Build Frontend Application and Publish to Azure App Service**
- 9. Test the Application & Sales Chat AI Assistant**



0. Walkthrough-Final Product

Sales POC

Customers

Customers

Products

Sales Reps

Sales Orders

Sales Facts

Customers

Total: 1000 records

SEARCH NAME
Search customers...

TYPE
All Types

INDUSTRY
All Industries

COUNTRY
All Countries

Apply Filters

ID	NAME	TYPE	INDUSTRY	COUNTRY	STATE	CITY	ANNUAL REVENUE (USD)
1	Customer 1	Distributor	Semiconductors	USA	CA	San Jose	\$476,385,208.00
2	Customer 2	Foundry	Semiconductors	USA	CA	San Jose	\$400,317,597.00
3	Customer 3	Fabless	Semiconductors	USA	CA	San Jose	\$274,165,439.00
4	Customer 4	Research	Semiconductors	USA	CA	San Jose	\$64,610,403.00
5	Customer 5	OEM	Semiconductors	USA	CA	San Jose	\$12,672,534.00
6	Customer 6	Distributor	Semiconductors	USA	CA	San Jose	\$268,436,184.00
7	Customer 7	Foundry	Semiconductors	USA	CA	San Jose	\$17,983,267.00
8	Customer 8	Fabless	Semiconductors	USA	CA	San Jose	\$501,041,045.00
9	Customer 9	Research	Semiconductors	USA	CA	San Jose	\$348,010,605.00
10	Customer 10	OEM	Semiconductors	USA	CA	San Jose	\$288,451,090.00
11	Customer 11	Distributor	Semiconductors	USA	CA	San Jose	\$214,094,691.00
12	Customer 12	Foundry	Semiconductors	USA	CA	San Jose	\$412,678,643.00
13	Customer 13	Fabless	Semiconductors	USA	CA	San Jose	\$246,942,484.00
14	Customer 14	Research	Semiconductors	USA	CA	San Jose	\$329,872,100.00
15	Customer 15	OEM	Semiconductors	USA	CA	San Jose	\$412,947,454.00

Sales POC

Customers

Products

Sales Reps

Sales Orders

Sales Facts

Sales Facts

Total: 30054 records

\$19,507,633,864.00
Total Revenue (USD)

76,477,299
Total Quantity Sold

30,054
Line Items

CUSTOMER
Search customer...

PRODUCT
Search product...

CATEGORY
All Categories

REGION
All Regions

SALES REP
All Reps

Apply Filters

ORDER ID	ORDER DATE	CUSTOMER	TYPE	PRODUCT	CATEGORY	QTY	UNIT PRICE	LINE TOTAL	REP	REGION
1	2025-10-10	Customer 250	Research	Chip-29	RF	2487	\$267.00	\$664,029.00	Rep 19	APAC
1	2025-10-10	Customer 250	Research	Chip-192	MCU	2168	\$232.00	\$502,976.00	Rep 19	APAC
1	2025-10-10	Customer 250	Research	Chip-134	FPGA	744	\$138.00	\$101,184.00	Rep 19	APAC
1	2025-10-10	Customer 250	Research	Chip-97	ASIC	1423	\$201.00	\$286,023.00	Rep 19	APAC
2	2024-09-01	Customer 218	Fabless	Chip-66	MCU	607	\$178.00	\$108,832.00	Rep 12	West
2	2024-09-01	Customer 218	Fabless	Chip-13	ASIC	1952	\$464.00	\$905,728.00	Rep 12	West
2	2024-09-01	Customer 218	Fabless	Chip-100	Sensor	176	\$166.00	\$29,216.00	Rep 12	West
2	2024-09-01	Customer 218	Fabless	Chip-18	MCU	2956	\$178.00	\$526,168.00	Rep 12	West
2	2024-09-01	Customer 218	Fabless	Chip-150	MCU	4817	\$328.00	\$1,570,342.00	Rep 12	West
3	2024-09-14	Customer 519	Research	Chip-70	Sensor	4337	\$67.00	\$290,579.00	Rep 15	APAC
3	2024-09-14	Customer 519	Research	Chip-19	ASIC	2891	\$359.00	\$1,037,869.00	Rep 15	APAC
3	2024-09-14	Customer 519	Research	Chip-109	ASIC	2429	\$392.00	\$952,168.00	Rep 15	APAC

Sales POC

Customers

Products

Sales Reps

Sales Orders

Sales Facts

Products

Total: 200 records

SEARCH NAME
Search products...

CATEGORY
All Categories

LIFECYCLE STATUS
All Statuses

Apply Filters

ID	NAME	CATEGORY	PROCESS NODE (NM)	PACKAGE TYPE	UNIT PRICE (USD)	LIFECYCLE STATUS
1	Chip-1	ASIC	7	QFN	\$455.00	Active
2	Chip-2	FPGA	14	WL CSP	\$324.00	Active

Product Description

Chip-2 is a FPGA semiconductor device built on a 14nm process node. It is optimized for IoT applications requiring high performance, power efficiency, and long-term reliability. Chip-2 supports industrial-grade temperature ranges and is designed for scalable system architectures and long lifecycle deployments.

Product Documents

ENGINEERING DATASHEET

Engineering-Datasheets/Chip-2_Engineering_Datasheet.docx

Download

MARKETING BRIEF

Marketing-Brief/Chip-2_Marketing_Brief.docx

Download

SALES ONE PAGER

Sales-One-Pager/Chip-2_Sales_One_Pager.docx

Download

3	Chip-3	Power IC	28	DIP	\$84.00	Active
4	Chip-4	Sensor	40	BGA	\$356.00	Active
5	Chip-5	RF	65	QFN	\$308.00	Active

Sales POC

Customers

Products

Sales Reps

Sales Orders

Sales Facts

Sales Orders

Total: 9 records

CUSTOMER *
Customer 10

STATUS
All Statuses

SALES REP
All Reps

Clear Filters

ORDER ID	CUSTOMER	SALES REP
226	Customer #10	Rep #23
374	Customer #10	Rep #16
3500	Customer #10	Rep #4
4923	Customer #10	Rep #3
5013	Customer #10	Rep #18
5412	Customer #10	Rep #12
8353	Customer #10	Rep #15
8480	Customer #10	Rep #20
9176	Customer #10	Rep #17

Showing 1-9 of 9 records

Sales Assistant

Powered by Azure AI Foundry

Hi! I'm the Sales Assistant. Ask me anything about sales data, customers, products, or orders.

8:00 PM

What is product description for Chip-50? Also retrieve Sales One Pager, Datasheet and market brief documentation. Finally check its unit price and status of its life cycle

8:08 PM

Product Description for Chip-50
Chip-50 is a FPGA semiconductor device built on a 14nm process node. It is optimized for automotive applications requiring high performance, power efficiency, and long-term reliability. Chip-50 supports industrial-grade temperature ranges and is designed for scalable system architectures and long lifecycle deployments.

Key Product Documents for Chip-50
• Sales One Pager:
Download Chip-50 Sales One Pager
• Datasheet:

Ask a question about sales data...

© Copyright Microsoft Corporation. All rights reserved.

1. Identify functional Requirements and non-functional Requirements



Functional Requirements

What the system does for the user.

1. Reports to display and filter data

Provide interactive reports and dashboards for

Customers (segmentation), **Products** (performance), and **Sales Reps** (quota attainment)

2. Real-Time Order Tracking

Real-time visibility into order status, fulfillment pipelines, and historical transaction logs

3. AI-Powered Natural Language Query (NLQ)

An integrated chatbot that allows users to ask questions like “What are the top 10 products by price?” and receive instant data visualizations.

Next Phase

4. Advanced Sales Analytics

5. Role-Based Access Control (RBAC)

Non-Functional Requirements

How the system operates and its quality constraints.

1. Intuitive UX/UI

A "zero-training" interface designed for mobile and desktop, ensuring high adoption rates among non-technical sales staff.

2. Low Latency Performance

Chatbot responses and dashboard renders must occur in **under 5 seconds** to maintain user engagement.

3. Scalability

Handle a **10x increase in data volume** or concurrent users without degradation in speed.

4. High Availability

Ensure **99.9% uptime** to support global sales teams operating across different time zones.

5. Data Security & Compliance

2. Generate Architecture Diagram to satisfy requirements



- Account for big Picture & Integrations & Reusability in other systems
- Estimate the system capacity in short and long term to determine the proper Services, Service Tiers, SKUs
- Choose relevant Azure Services to implement a reliable / scalable / cost effective Architecture

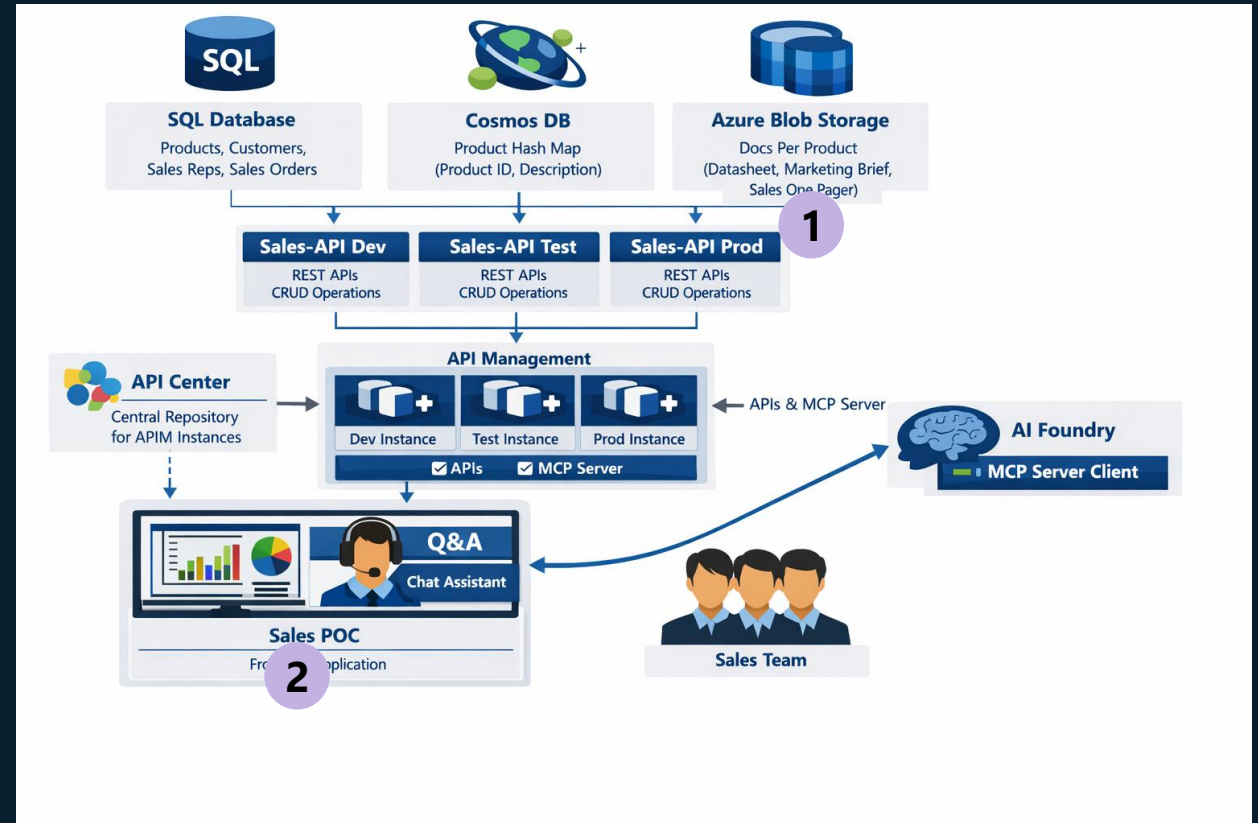
Decisions to make

1. Which Service to use for Sales-API?

- We need to review big picture

2. Which Service to use for Frontend Internal Reporting Web App?

- Azure App Service (Internal Sales Team – Small number of users)



2. Account for big picture - Sales Transactional System

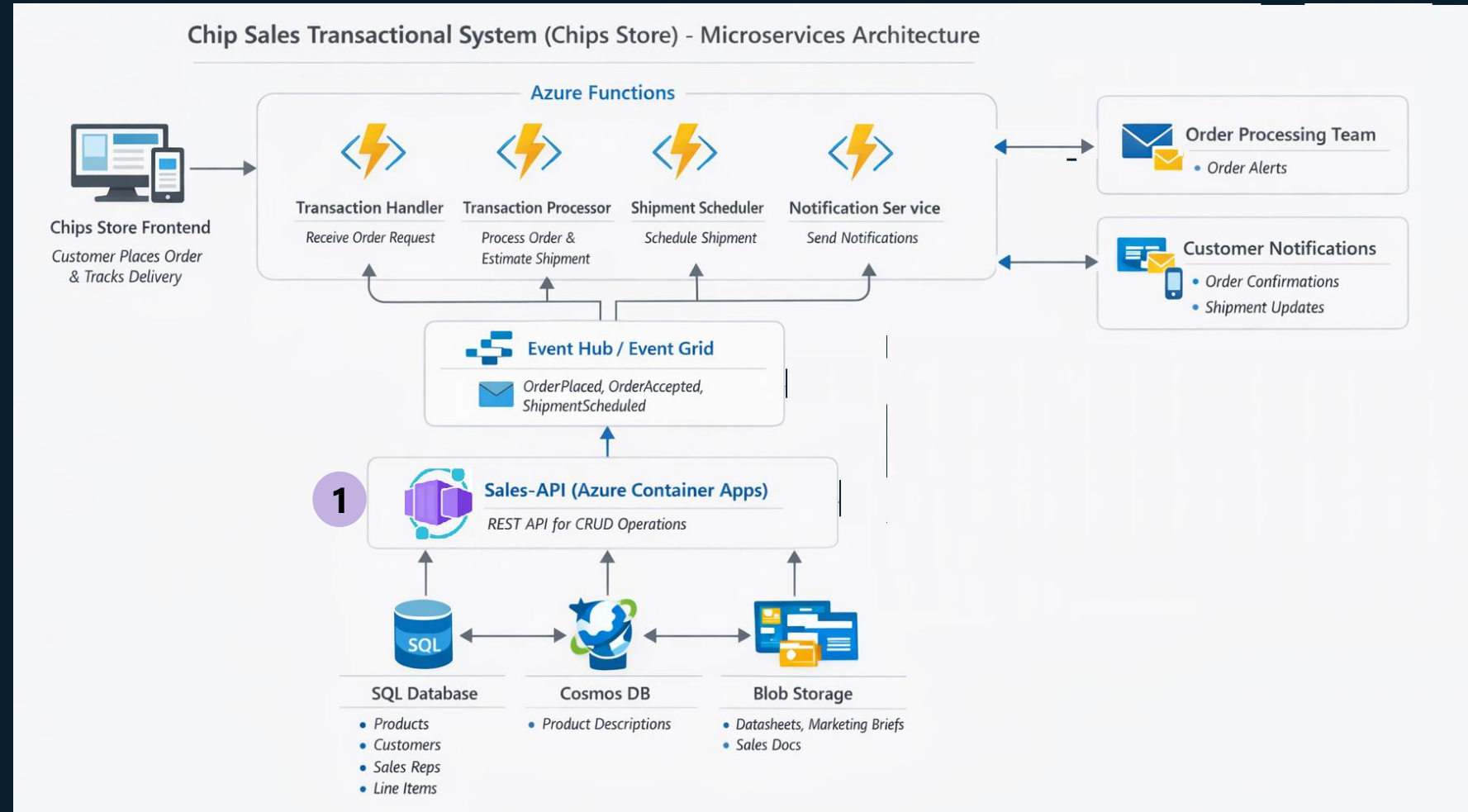


Big Picture helps understand scale

Sales-API
can use
“**Azure Container Apps**”

Why?

**Shared service in
internal Report System
and external Sales
Transactional System**



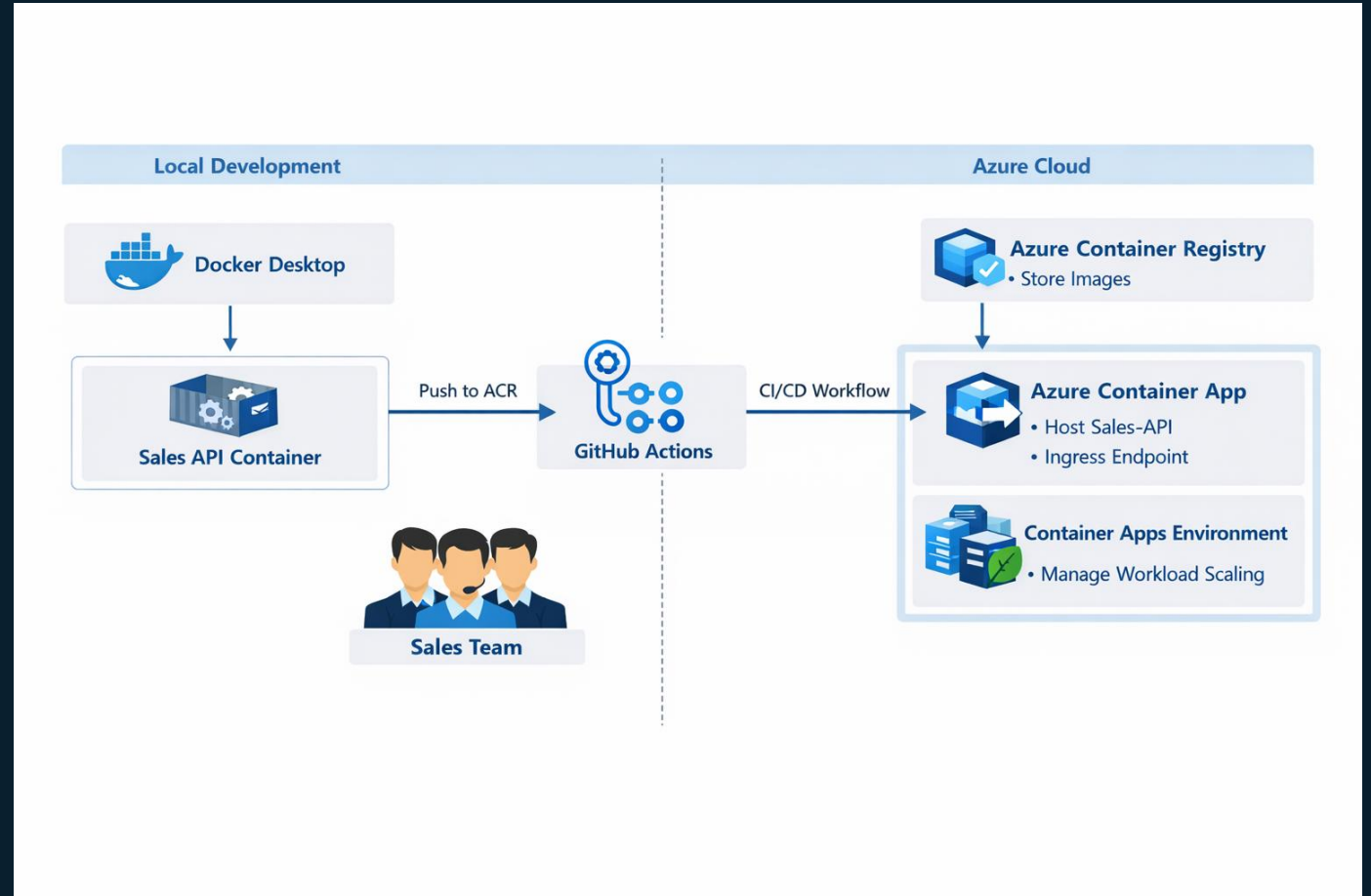
3. Build APIs – Azure Container Apps



Flow to create Azure Container App

Components Needed

- **Docker Desktop**
- **GitHub Action** to push docker container (e.g. Sales-API) to **ACR**
- **ACR** stores Container Images
- Deploy Sales-API from **ACR** to **ACA** (Azure Container App)
- **ACA** hosts Sales-API
- **Container Apps Environment** manages Workload Scaling



- **ACR:** Azure Container Registry
- **ACA:** Azure Container Apps

3. Build APIs – Azure Container Apps



1. Go to Azure Portal
 - a) Create "**Azure Container Registry**" and fetch Uri (Identifier)
 - b) Create "**Azure Container App**" and fetch Uri (Identifier)
2. Fetch Connection Strings for all Data Sources that you need to connect to in your App Service
3. Create a GitHub Repository (e.g. SalesPOC.API) and clone it to your computer
4. Open Visual Studio Code (or Visual Studio) and leverage "Copilot Coding Agent" to generate the source code.
Example Prompt: "Generate a .NET 10 REST APIs to implement CRUD operations for the following data sources: {Paste list of data source names and connection strings / access keys} and generate GitHub actions workflow file to deploy the code to Azure Container Registry and Azure Container App {Paste Uri / identifier for ACR, ACA}. Add an ingress https endpoint for Sales-API in Azure Container App".
5. Run Code locally in browser to ensure that it works as expected or re-prompt Copilot to fix issues
6. Ask Copilot Coding Agent to generate a swagger documentation and / or OpenAI Specs to import in APIM later
7. Go to GitHub Repo online and ensure that GitHub Actions ran successfully and deployed App

4. Create new API Management Instance in Azure and Import Sales-API



Microsoft Azure

Home > apim-poc-my

apim-poc-my | APIs

Search resources, services, and docs (G+I)

Search

Developer portal Send us your feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

Events

APIs

Workspaces

APIs

MCP Servers

Products

Subscriptions

Named values

Backends

Policy fragments

API Tags

Schemas

Credential manager

API Center

Power Platform

Developer portal

Portal overview

Settings

Users

Groups

Identities

Delegation

OAuth 2.0 + OpenID Connect

Monitoring

Analytics

Analytics (classic)

Application Insights

Alerts

Metrics

Diagnostic settings

Logs

Advisor

Create an AI API

Language Model API
Create an API for an OpenAI API-compatible model.

Microsoft Foundry
Connect API Management services to Microsoft Foundry.

Define a new API

HTTP
Manually define an HTTP API

WebSocket
Streaming, full-duplex communication with a WebSocket server

GraphQL
Access the full capabilities of your data from a single endpoint.

gRPC
High performance, universal Remote Procedure Call framework

Create from definition

OpenAPI
Standard, language-agnostic interface to REST APIs

WADL
Standard XML representation of your RESTful API

WSDL
Standard XML representation of your SOAP API

OData
Standard XML representation of your OData API

Create from Azure resource

App Service
API hosted on App Service.

Function App
Serverless, event driven experience on App Service.

Container App
Serverless containers for microservices.

Logic App
Scalable hybrid integrations and workflows.

Microsoft Azure

Home > apim-poc-my

apim-poc-my | APIs

Search resources, services, and docs (G+I)

Search

Developer portal Send us your feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

Events

APIs

Workspaces

APIs

MCP Servers

Products

Subscriptions

Named values

Backends

Policy fragments

API Tags

Schemas

Credential manager

API Center

Power Platform

Developer portal

Portal overview

Settings

Users

Groups

Identities

Delegation

OAuth 2.0 + OpenID Connect

Monitoring

Analytics

Analytics (classic)

Application Insights

Alerts

Metrics

Diagnostic settings

Logs

Advisor

REVISION 3

CREATED Feb 10, 2026, 4:10:00 PM

Design Settings Test Revisions (3) Change log

POST Create an order item

DEL Delete a customer

DEL Delete a product

DEL Delete a sales order

DEL Delete a sales representative

DEL Delete an order item

GET Get a customer by ID

GET Get a product by ID

GET Get a sales order by ID

GET Get a sales rep by ID

GET Get all customers

GET Get all customers

GET Get all order items

GET Get all order items

GET Get all products

GET Get all products

GET Get all sales facts

GET Get all sales facts

GET Get all sales orders

GET Get all sales orders

GET Get all sales representatives

GET Get all sales reps

GET Get an order item by ID

SalesAPI > Get all products > Console

Headers

+ Add header

NAME	VALUE	TYPE	DESCRIPTION
------	-------	------	-------------

Apply product scope

Product

None

Request URL

https://apim-poc-my.azure-api.net/v1/api/Products

HTTP request

GET https://apim-poc-my.azure-api.net/v1/api/Products HTTP/1.1
Host: apim-poc-my.azure-api.net

HTTP response

Message Trace

HTTP/1.1 200 OK
content-encoding: gzip
content-type: application/json; charset=utf-8
date: Wed, 11 Feb 2026 23:48:52 GMT
request-context: appId=cid-v1:604f8a5d-e448-413e-93cc-0099076cc3ec
transfer-encoding: chunked
vary: Accept-Encoding, Origin
x-powered-by: ASP.NET

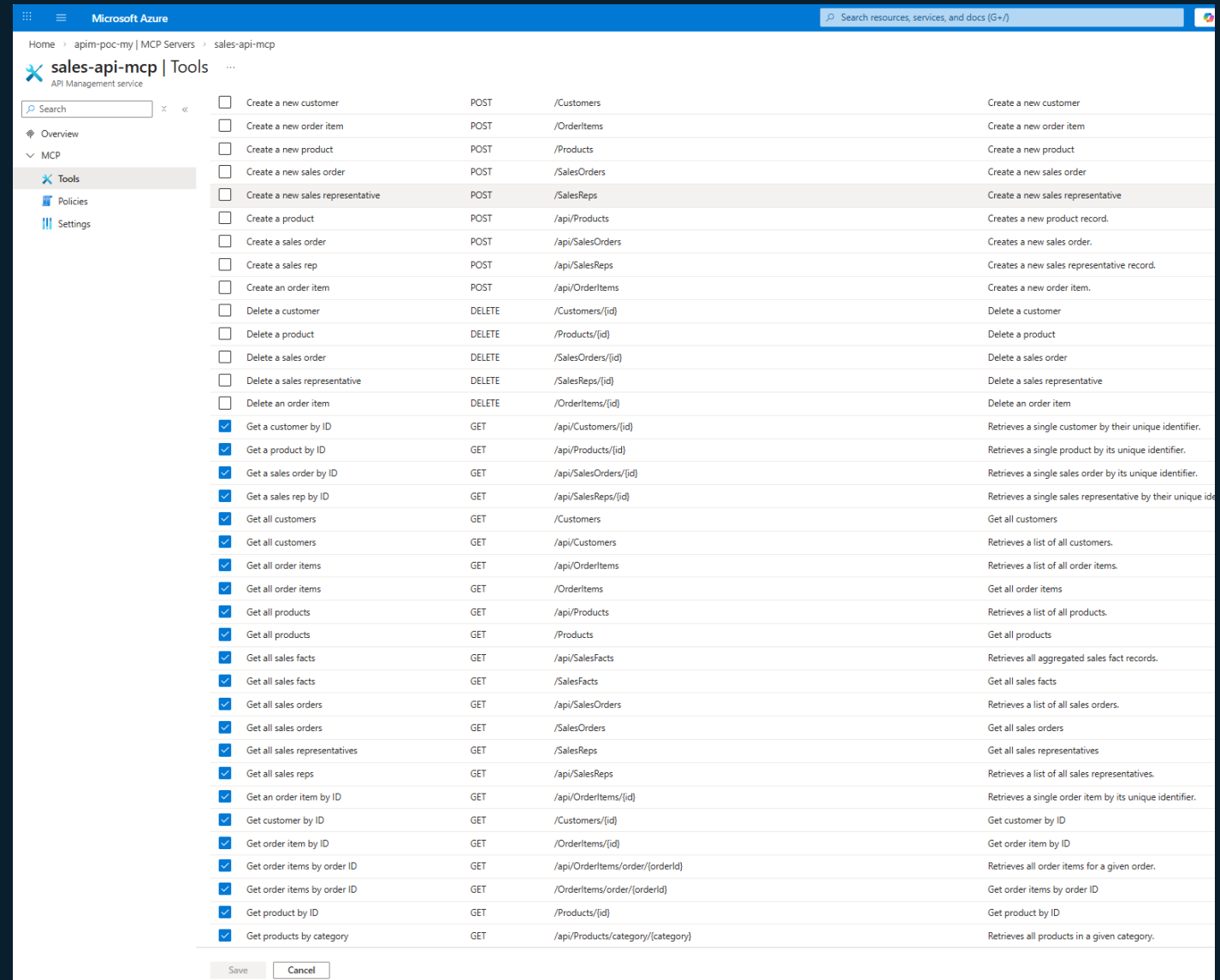
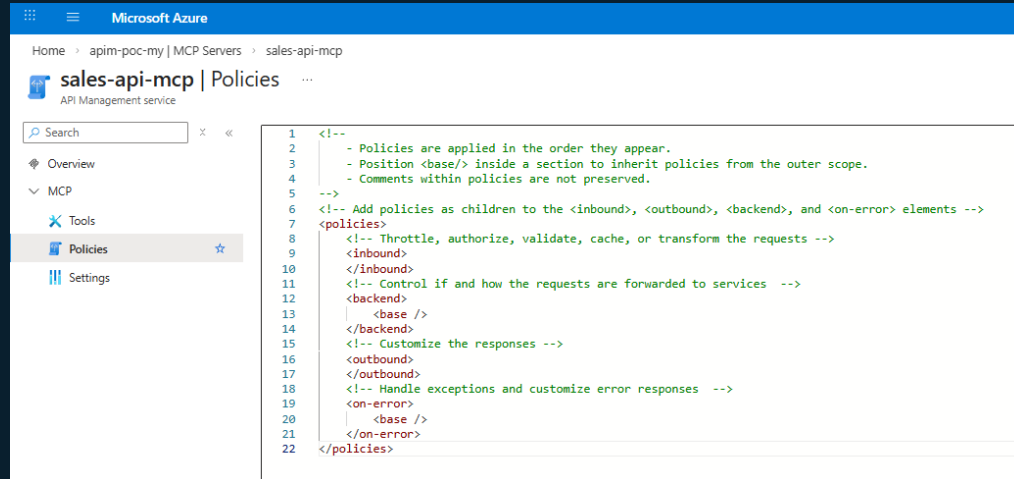
```
[{"productId": 1, "productName": "Chip-1", "productCategory": "ASIC", "processNode": 7, "packageType": "QFN", "unitPriceusd": 455.00, "lifecycleStatus": "Active", "createdDate": "2026-01-26T23:56:53.1254873", "orderItems": []}, {"productId": 2, "productName": "Chip-2", "productCategory": "FPGA", "processNode": 14, "packageType": "MLCSP", "unitPriceusd": 324.00, "lifecycleStatus": "Active", "createdDate": "2026-01-26T23:56:53.1254873", "orderItems": []}, {"productId": 3,
```

Send Trace Bypass CORS proxy 60 Timeout (in second)



5. Create MCP Server in APIM

- Expose Tools (API operations)
- Define policies for AI Agents to consume MCP Server



6. API Center – Register APIM Instance



- Register APIM Instances for Dev, Test, Prod
- Review Portal to discover APIs & MCP Servers

- API Analysis / Linting

Microsoft Azure

Home > api-center-poc-my

api-center-poc-my | Environments ☆ ...

API Center

Search

An environment represents a location where an API runtime is deployed, such as an API management platform or a Kubernetes cluster. You can

+ New environment Refresh

Search Type: All Server type: All

Environment title	Environment type	Server type
APIM Prod	Production	Azure API Management
APIM Dev	Development	Azure API Management

Microsoft Azure

Home > api-center-poc-my | API Analysis

API Analysis Report - Default ...

Title: SalesAPI

API version: v1

Definition: Default

Lint issue summary: 190 total issues

Error (77): Action required. The error must be fixed for successful operation.

Warning (113): Recommendation. Consider revising for optimal performance and readability.

Information (0): Contains suggestions for improvement and notes on code practices not classified as errors or warnings.

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "SalesAPI",
    "description": "Sales API for managing customers, products, orders, and sales representatives.",
    "version": "v1"
  },
  "servers": [
    {
      "url": "http://apim-poc-my.azure-api.net/v1"
    },
    {
      "url": "https://apim-poc-my.azure-api.net/v1"
    }
  ],
  "paths": {
    "/customers": {
      "get": {
        "tags": [
          "customers"
        ],
        "summary": "Get all customers",
        "description": "Get all customers",
        "operationId": "Customers_GetAll",
        "responses": {
          "200": {
            "description": "Success",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/CustomerArray"
                },
                "example": [
                  {
                    "customerId": 0,
                    "customerName": "string",
                    "customerType": "string",
                    "industry": "string",
                    "country": "string",
                    "state": "string",
                    "city": "string",
                    "annualRevenueUsd": 0,
                    "createdDate": "string"
                  }
                ]
              }
            }
          }
        }
      }
    }
  }
}
```

Description	Line	Rule	Analyzer
Info object must have "contact" object.	3	info-contact	Spectral
Info object must have "license" object.	3	info-license	Spectral
License object must include "url".	3	license-url	Spectral
"createdDate" property must match format "date-time"	43	oas3-valid-media-example	Spectral
"createdDate" property must match format "date-time"	73	oas3-valid-media-example	Spectral
"createdDate" property must match format "date-time"	95	oas3-valid-media-example	Spectral
Parameter objects must have "description".	112	oas3-parameter-description	Spectral
"createdDate" property must match format "date-time"	139	oas3-valid-media-example	Spectral
Parameter objects must have "description".	160	oas3-parameter-description	Spectral

7. AI Foundry – Build Sales Agent using MCP Server



- Create a new agent
- Add MCP Server as a Tool under Tools
- Provide clear instructions
- Add Guardrails (Responsible AI)

The screenshot displays the Microsoft Foundry interface for configuring an AI agent. The left sidebar shows navigation options: Agents, Workflows, Models, Fine-tune, Tools, Knowledge, Data, Evaluations, and Guardrails. The main area is titled 'arrow-sales-agent' and includes tabs for Playground, Traces, Monitor, and Evaluation. The 'Playground' tab is active, showing the agent's configuration.

Instructions:

You are an expert sales analytics assistant. You have access to a SQL Server database containing sales data including customers, products, sales orders, order items, sales representatives, and a consolidated sales fact view.

Tools:

Connect tools to your agent for faster access to key information and improved efficiency. [Learn more](#) about best practices.

sales-mcp (https://apim-poc-my.azure-api.net/sales-api-mcp/mcp)

Knowledge:

Memory:

Guardrails:

Chat:

Draw a bar chart for top 10 products by unit price.

Here are the top 10 products by unit price (in USD):

Rank	Product Name	Unit Price (USD)
1	Chip-70	\$503.00
2	Chip-127	\$501.00
3	Chip-11	\$496.00
4	Chip-193	\$494.00
5	Chip-60	\$493.00
6	Chip-108	\$493.00
7	Chip-74	\$491.00
8	Chip-90	\$486.00
9	Chip-39	\$485.00
10	Chip-21	\$482.00

Bar Chart (text representation)

Product Name	Unit Price (USD)
Chip-70	\$503.00
Chip-127	\$501.00
Chip-11	\$496.00
Chip-193	\$494.00
Chip-60	\$493.00
Chip-108	\$493.00
Chip-74	\$491.00
Chip-90	\$486.00
Chip-39	\$485.00
Chip-21	\$482.00

If you need a visual chart image or want this exported for use in Excel or another tool, let me know!

gpt-4.1 8.4s 15987t sales_mcp Debug

Message the agent...

AI-generated content may be incorrect

8. Build Frontend web App – Azure App Service



1. Go to Azure Portal -> Create "Web App" Service and fetch Uri (Identifier)
2. Fetch API URL from APIM that will be used in the Frontend Web App
3. Create a GitHub Repository (e.g. SalesPOC.UI) and clone it to your computer
4. Open Visual Studio Code (or Visual Studio) and leverage "Copilot Coding Agent" to generate the source code.

Example Prompt:

"Generate a web application using Angular and Typescript to display and filter data using this API End-Point {Paste Sales-API URL from APIM}.

Create a separate screen for Customers, Products, Sales Reps, Sales Orders, Sales Facts.

For each product in Products screen, add a button to show the Product description and Product documentation which includes 3 documents per product (Datasheet, Marketing Brief, Sales One Pager).

Finally, add a Sales Chat Assistant that uses sales-agent from AI Foundry.

AI Foundry Project Endpoint is {Paste Foundry Project Endpoint} and API Key is {Paste API Key}"

5. Run Code locally in browser to ensure that it works as expected or re-prompt Copilot to fix issues
6. Go to GitHub Repo online and ensure that GitHub Actions ran successfully and deployed App

Sales POC Alternative Architecture – Data Platform

