

# Appeon PowerServer

Berlin 29.04.2025  
Meoni Marco, Appeon MVP



# DISCLAIMER

This presentation was authored by volunteer(s) in the Appeon community. This is not a work for hire by Appeon. The views and opinions expressed in this presentation are those of the author(s).

Its contents are protected by US copyright law and may not be reproduced, distributed, transmitted, displayed, published or broadcast without the prior written permission of Appeon. All rights belong to their respective owners.

Any reference to third-party materials, including but not limited to Websites, content, services, or software, has not been reviewed or endorsed by Appeon. YOUR USE OF THIRD-PARTY MATERIALS SHALL BE AT YOUR OWN RISK.

Appeon makes no warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Appeon assumes no responsibility for errors or omissions.



# Profile



marco.meoni@gmail.com

Appeon MVP



meonimarco

## Recent Projects

- PowerServer addict
- Training in Appeon products
- WebApp driven by .NET Datastore
- Maintenance of legacy PB apps
- Consultant for Appeon, Novalys, SPI, Power People-IT...

## Key Skills

- PowerBuilder
- PowerServer
- SnapDevelop
- .NET DataStore
- SnapObject ORM
- Vue.JS
- Big Data
- Machine Learning

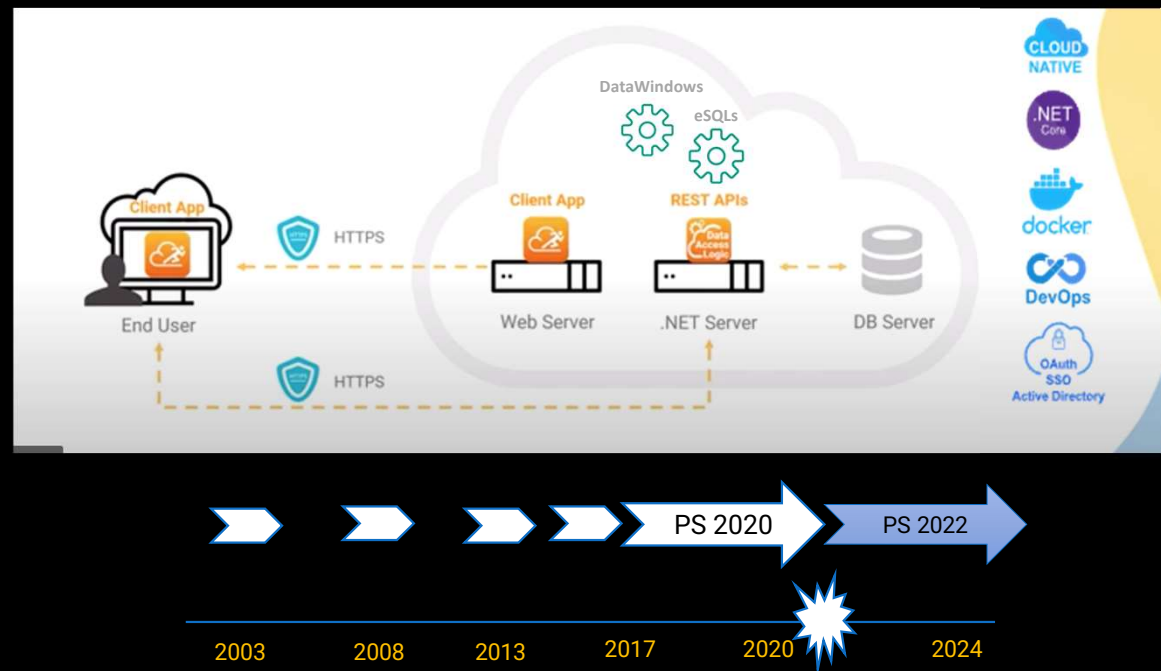
# Session Agenda

- Quick Start
  - Architecture
  - Demo
  - SW Components
  - PB Project
  - App Launcher
  - Database Connection
- Advanced
  - Dynamic DB mapping
  - Long-running vs short-lived
  - Merge server calls
- PS2025 new features



# What is PowerServer

- PB project that converts c/s App into Installable Cloud App (ICA)



# Why PowerServer

- Internet Access
- Minimal Effort
- Leverage PB skills only
- Different from RDP/VDI
- .NET8-based API and PB Native Frontend
- Web Security



# PB (Frontend + Backend)

The screenshot displays the PowerBuilder 2022 R3 IDE running a Sales CRM Demo. The main window, titled "Sales CRM Demo", shows a table of addresses. The table has columns for ID, Address 1, Address 2, and City. The data is as follows:

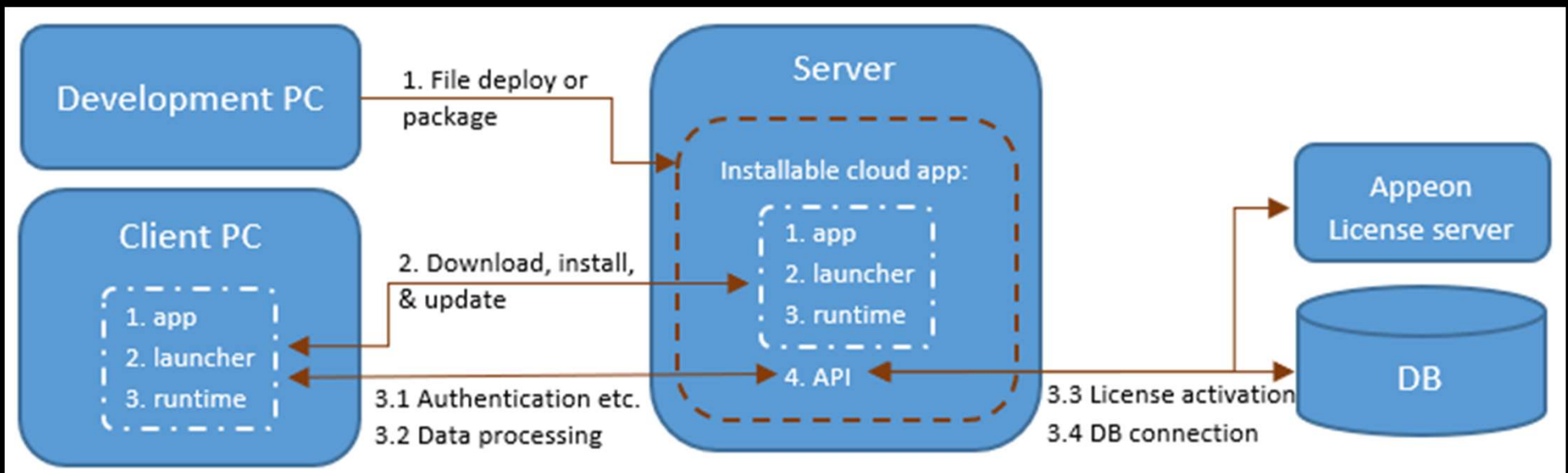
ID	Address 1	Address 2	City
466	655-4th Ave S.W.		Calgary
467	25900-700-9th Ave S.W.		Calgary
475	2512-410th Avenue S.W.		Calgary
487	9259 - 1110th Avenue Sw		Calgary
502	No. 25400 10665 Jasper Avenue		Edmonton
504	Suite 99320 255 - 510th Avenue S.W.		Calgary
505	6400, 888 - 3rd Avenue		Calgary
519	2500-622 5th Ave Sw		Calgary
526	251420a 170th Street		Edmonton

The "Sales CRM Demo (For Development Use Only) - [Address]" window is also visible, showing a "Home" tab and a "Sales CRM Demo" button. The "localhost:5099/salesdemo\_cloud/" browser window shows a "Start the Application" button. The PowerBuilder IDE in the background shows the "Deploy" button highlighted in the "Basic" tab of the "Salesdemo\_cloud" project.

The background is a solid blue color. It features several white starburst or lens flare effects scattered across the surface. A prominent, lighter blue diagonal band or streak runs from the bottom left towards the upper right. In the bottom right corner, there is a small cluster of overlapping colored shapes in yellow, green, and blue.

**Demo**

# PS Components



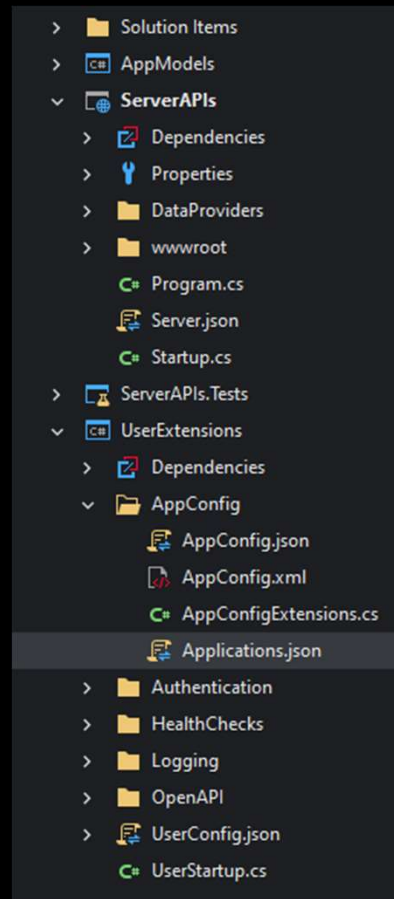
# Cloud App Launcher

- Why is it needed?
- How is it installed?
- What types exist?
- Are admin rights needed?
- Can multiple versions coexist?



# App Web API

- SnapDevelop Solution
  - AppModels
  - ServerApi
  - ServerApi.Tests
  - UserExtensions



## Publish

Where do you want to publish?



### Folder

Publish your application to a local folder or file share



### Web Server (IIS)

Publish your application to IIS using Web Deploy



### Docker Container Registry

Publish your application to a supported container registry that works with Docker images



### Kubernetes Cluster


Publish your application to the Kubernetes cluster.

# OpenApi

















## PowerServer API

swagger/v1/swagger.json





PowerServer management APIs

Authorize 

### Application

GET	/api/Application/LoadOne/{appName}	 
GET	/api/Application/LoadAll	 
POST	/api/Application/Add	 
POST	/api/Application/Edit	 
POST	/api/Application/Remove/{appName}	 
POST	/api/Application/AddTransactionMapping/{appName}/{transName}	 
POST	/api/Application/EditTransactionMapping/{appName}/{transName}	 
POST	/api/Application/RemoveTransactionMapping/{appName}/{transName}	 

### Connection

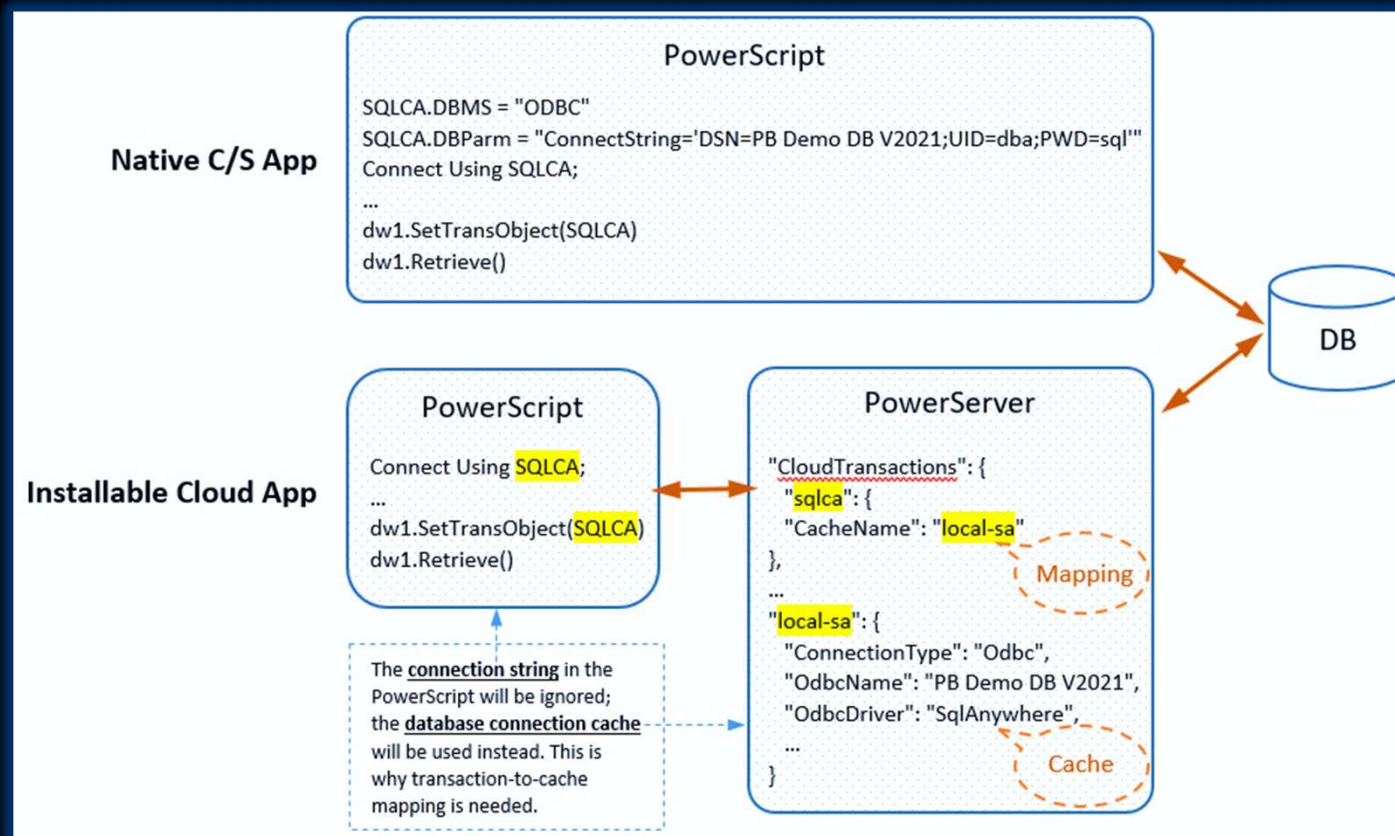
GET	/api/Connection/LoadOne/{cacheGroup}/{cacheName}	 
GET	/api/Connection/LoadGroup/{cacheGroup}	 

# Database Support

- IBM DB2 UDB 11.x
- Informix 12.x or 14.x
- MySQL 5.6, 5.7, or 8.0
- Oracle 12c, 18c, 19c, 21c, or 23c
- PostgreSQL 12, 13, 14, or 15
- MSSQL 16, 17, 19, 22, or Azure SQL
- ASE (ODBC) 16.0
- SQL Anywhere (ODBC) 16, or 17



# Transaction-to-Cache Mapping



# Debugging

- Debug a PS project directly in PB IDE
- PS Debugger has same toolset as PB Debugger
- PS project is debugged against PS runtime
- See return values from PS Web API



# PBAutoBuild

```
{
  "BuildPlan": {
    "SourceControl": {
      "Git": [
        {"SrcPath": "https://my.azure.com/pbdemo/_git/pbdemo", "User": "user@gmail.com, ..."}
      ],
      "Merging": [
        {"Target": "..\\PowerBuilder 22.0\\pbdemo\\pbdemo.pbt", ...}
      ],
      "PostCommand": ""
    },
    ...
  },
  "Projects": {
    "pbdemo": {
      "ProjectType": 3,
      "Application": {
        ...
      },
      "Deploy": {
        "External": {
          "IIS": {
            "ServerProfileName": "DEV",
            "Name": "myserver",
            "SiteName": "PS2022",
            "Username": "workgroup\\marmeo",
            "Password": "MV58ljE7KXk9c3doMnU+XTohIVg=",
            "FrameworkMode": 0,
            "AppURL": "https://myserver.cloud.com",
            "WebAPISiteURL": "https://myserver.cloud.com:8080"
          }
        },
        ...
      }
    }
  }
}
```

# Advanced Topics

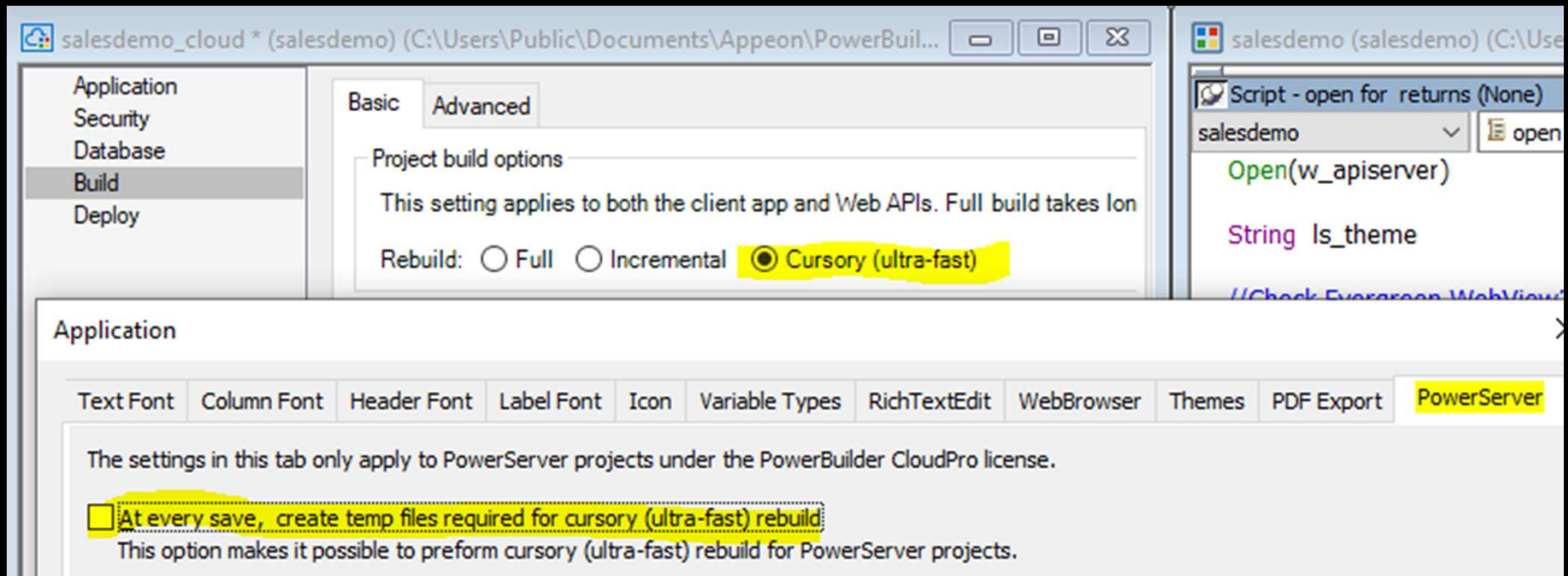


# Topics List

- Ultra-fast build option
- Database Handling
  - Connection types
  - Long Running Connection
- Call Merging
- Web API URL



# Ultra-Fast Build



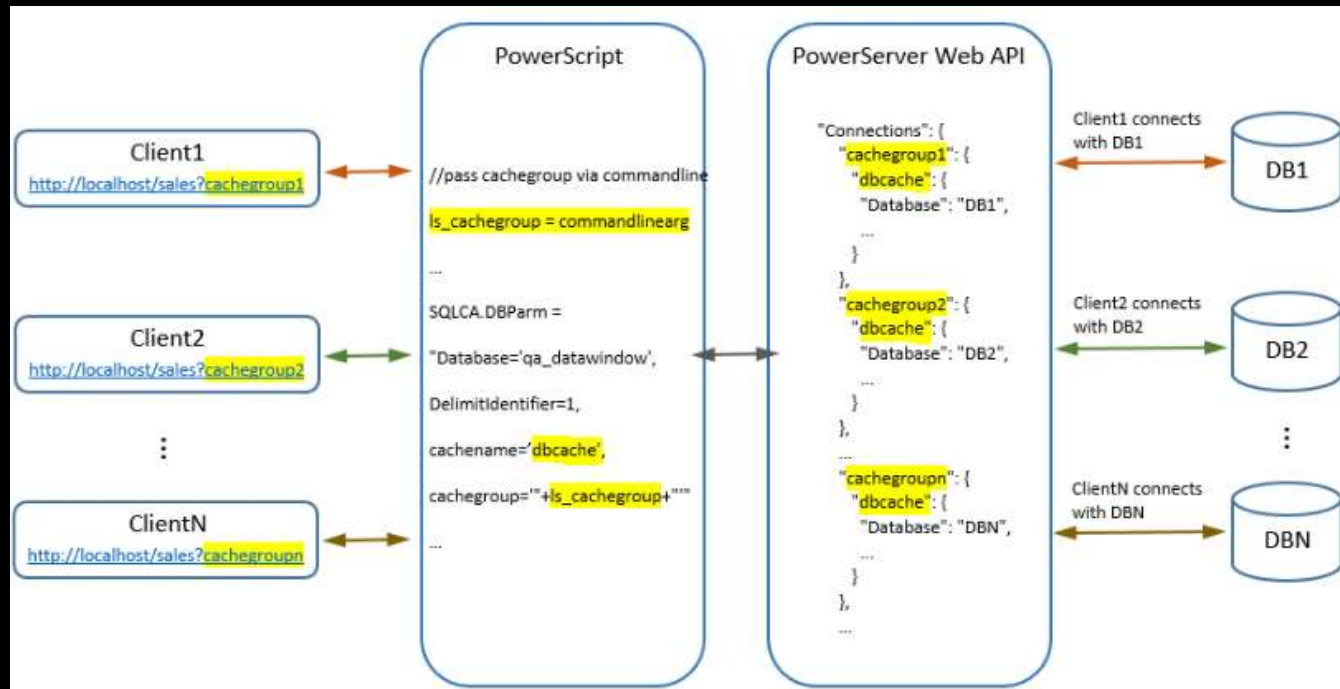
# DataBase Handling

- Connection Types
  - Transaction-to-Cache (static)
  - Cache-names (dynamic)
- Long-Running Connection



# Dynamic Cache

`sqlca.dbparm="cachegroup='prod', cachename='pbdemo2022' "`



ServerAPI > AppConfig > Applications.json

## Long-Running vs Short-Lived

- Client/Server app (PB)
  - *Long-running* DB connection
  - Always the same one for a given client
- Cloud App (PS)
  - *Short-lived* DB connection
  - Made from PS ServerAPI
  - Cached in a pool when idle and used by any client when needed
  - Can change during client session
  - Can be shared across multiple client sessions



# Pitfalls

- Client/Server app
  - Temp tables are destroyed when DISCONNECT is executed
  - Context variables are removed when DISCONNECT is executed
- Cloud app
  - Results of DB operations are confined to current connection, which is *short-lived*
  - Temp tables or context variables are lost when connection is returned to the pool



## LongConnection Option

- LongConnection=1 in DBparm enables long-lived connection
  - Recommended increase the transaction timeout value
  - Keep in mind server overhead
  - Use two transaction objects: one for long-running and one for short-lived connections



# Performance Tuning

- Tuning excessive server calls
- Why?
  - Network-intensive scripts highlight poor network connection
- How?
  - Remove recursive eSQL
  - Remove DW computed fields calling user functions w/ eSQL
  - Move non-visual logic to REST APIs
  - Group multiple server calls with PowerServerLabel



# Call Merging

```
PowerServerLabel lnv_queue  
lnv_queue.StartMerge()  
    lds.Retrieve(ll_id)  
    dw_1.Retrieve(ll_id)  
    SELECT id INTO :ll_id FROM TableA;  
    Execute Immediate :ls_sql;  
lnv_queue.EndMerge()  
COMMIT;
```

```
PowerServerLabel lnv_queue  
lnv_queue.StartMerge(1)  
    dw_1.Update()  
    dw_2.Update()  
lnv_queue.EndMerge()  
If lnv_queue.Results[1].SQLCode = 0 And  
    lnv_queue.Results[2].SQLCode = 0 Then  
    COMMIT;  
Else  
    ROLLBACK;  
End If
```



# Web API URL

- apprun.json (*static* URL change)

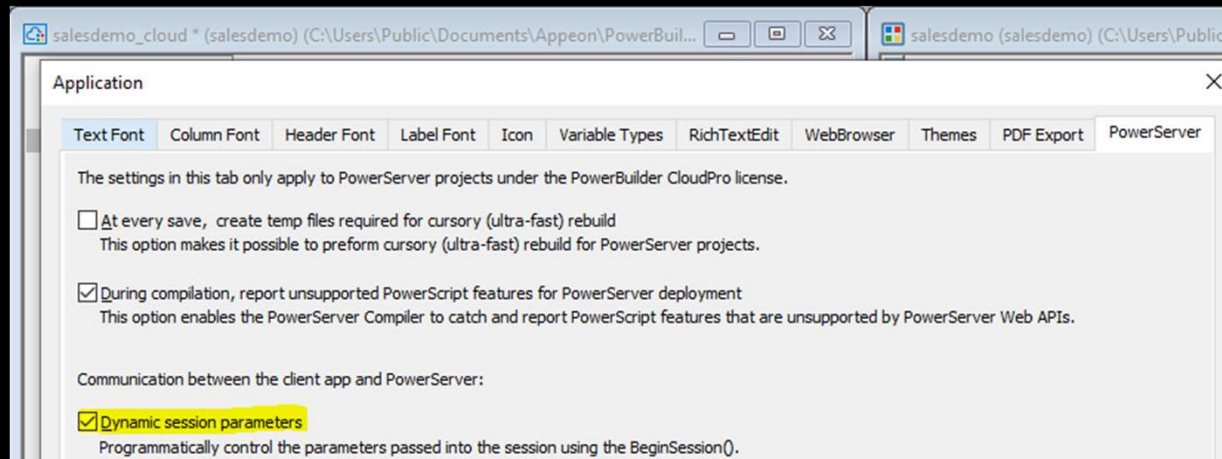
```
{ "deployment_urls": {  
    "launcher": "CloudAppPublisher/CloudAppInstall 22.0",  
    "runtime": "CloudAppPublisher/runtime/22.2.0.3356",  
    "web_api": "http://myserver.cloud.com:5099" }  
}
```



## Web API URL (cont)

- SetPowerServerURL / BeginSession (*dynamic* API URL change)

```
GetApplication().SetPowerServerURL(ls_new_url)  
GetApplication().BeginSession()
```



## Integration & Security

- PS 2022 solution can target .NET 8.0 framework
- HTTP/2 & TLS 1.3 support
- Multiple authentication templates and services

The screenshot displays the 'Basic' tab of the 'Web API generation' configuration window. The left sidebar shows a tree view with 'Instructions', 'Client App', 'Server Setup' (expanded), 'Web Server', '.NET Server' (selected), 'Database', and 'Startup'. The main area is titled 'Web API generation' and contains the instruction: 'Specify how to create the Web API solution that will contain the Web API projects, namely, the AppModels'. Below this, there are four configuration fields: '\* Solution name:' with a dropdown set to 'PowerServer\_salesdemo'; '\* Auth Template:' with a dropdown set to 'Do not use auth service'; '\* Namespace:' with a dropdown set to 'Do not use auth service'; and '\* Target framework:' with a dropdown set to 'Use external Azure Active Directory service'. At the bottom, there is a checkbox labeled 'Overwrite server settings (DB connection, .NET server listening port, and license)' which is checked.

# PowerServer 2025

## New Features



# Apprun.json

- Configure the application shortcut name

```
{
  "deployment_urls": {
    "launcher": "CloudAppPublisher/CloudAppInstall 25.0",
    "runtime": "CloudAppPublisher/runtime/25.0.0.3018",
    "web_api": "http://localhost:5099"
  },
  "deployment_settings": {
    "shortcut_name": "testshortcutname"
  }
}
```



## Loading images via URL

- CloudAppSet ("picturecachepath", "/picturecache");
- CloudAppSet ("clearpicturecacheonclose", "true");
- CloudAppSet ("checkpictureforupdate", "true");

```
dw_1.Object.pb_1.filename = http://172.16.5.175/button2.jpg  
dw_1.Object.compute_1.Expression = "bitmap('http://172.16.5.175/computed2.jpg')"
```

## Misc

DataWindow SQLPreview

- CloudAppSet ("enablesqlpreview", "true")

Request Header for large data

- CloudAppSet ("multipartformdataminsize", "128");



The background is a solid blue color with several thin, white, curved lines and starburst patterns scattered across it. A prominent white curved line runs from the bottom left towards the center. There are three starburst patterns: one in the upper left, one in the upper center, and one in the middle right. In the bottom right corner, there is a small cluster of overlapping colored shapes in orange, yellow, green, and blue.

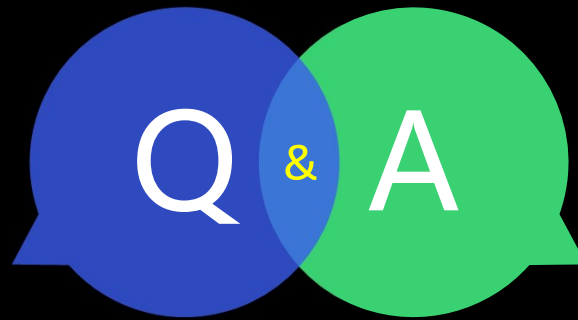
**Demo**

# Conclusions

- PowerServer
  - Seamlessly enables PB apps to the Cloud
  - Support of PB features is ~100%
  - Few days conversion effort
  - Open REST/JSON design
  - DB-centric framework



# Q&A Time



The background is a solid blue color. It features several thin, white, curved lines that sweep across the frame. There are also four starburst or lens flare effects, each consisting of a small white square with four radiating lines, positioned at various points on the background.

**Thank You**