You

Let’s Explore All Real Time Projects based QnA for all kinds of MICROSOFT 365 WEB PARTS, APPS, SERVICES USED TO DEPLOY ON SHAREPOINT ONLINE VS ON PREM FROM 2003,3007,2010,2013,2016,2019 BOTH CLASSIC VS MODERN at all Levels: Admin, Development, Designing, Migration, Branding level for Super Senior Architect Consultant level IN A TABULAR REPRESENTATION MENTIONING ALL KINDS OF POINTS SUMMARY !!

Copilot

Certainly! Let’s delve into the comparison between SharePoint Online and SharePoint On-Premises, focusing on various aspects. I’ll provide a concise summary in a tabular format:

Aspect	SharePoint Online	SharePoint On-Premises
Deployment Model	Cloud-based, hosted on Microsoft’s servers.	On-premises, requires dedicated hardware (Windows Server, SQL Server).
Customization	Limited customization due to cloud constraints.	High level of control and customization options.
Scalability	Scalable, with automatic updates and scalability features.	Scalability depends on your infrastructure and capacity planning.
Security	Advanced security measures, integrated with Microsoft 365.	Customizable security, but requires internal IT management.
Connectivity	Accessible from anywhere with an internet connection.	Requires local network access; may have limitations for remote users.
Maintenance	Automatic updates and maintenance by Microsoft.	Requires manual maintenance, patching, and upgrades by your IT team.
Cost	Subscription-based model; predictable costs.	Upfront investment in hardware and software; ongoing maintenance costs.

Remember that both options have their pros and cons, so choosing the right one depends on your organization’s specific needs and priorities . If you have any further questions or need additional details, feel free to ask! 😊

You

What are the key differences in development between the two?

Copilot

Certainly! Let’s explore the key differences in development between SharePoint Online and SharePoint On-Premises:

1. **Environment**:
   - **SharePoint Online**: Developers work in a cloud-based environment hosted by Microsoft.
   - **SharePoint On-Premises**: Developers work within their organization’s local network.
2. **Customization and Extensibility**:

   - **SharePoint Online**:
      - Limited customization due to cloud constraints.
      - Customizations are done using SharePoint Framework (SPFx) solutions.
      - No server-side code (e.g., no farm solutions).
   - **SharePoint On-Premises**:
      - High level of control and customization options.
      - Supports server-side code (e.g., farm solutions, event receivers).

3. **Development Tools**:

   - **SharePoint Online**:
      - Use Visual Studio Code with SPFx extensions.
      - Leverage SharePoint Workbench for testing.
   - **SharePoint On-Premises**:
      - Use Visual Studio or SharePoint Designer.
      - Directly deploy to the on-premises farm.

4. **Deployment and Testing**:

   - **SharePoint Online**:
      - Deploy to SharePoint Online directly from development environment.
      - Test in SharePoint Workbench or a developer site.
   - **SharePoint On-Premises**:
      - Deploy to a staging environment first.
      - Test on an isolated on-premises farm.

5. **Access and Permissions**:

   - **SharePoint Online**:
      - Developers need appropriate permissions in the Microsoft 365 tenant.
      - Accessible from anywhere with an internet connection.
   - **SharePoint On-Premises**:
      - Developers need access to the local network.
      - Permissions managed within the organization.

6. **Lifecycle Management**:

   - **SharePoint Online**:
      - Automatic updates and maintenance by Microsoft.
      - No need to worry about patching or upgrades.
   - **SharePoint On-Premises**:
      - Requires manual maintenance, patching, and upgrades.
      - IT team manages the lifecycle.

Remember that the choice between SharePoint Online and On-Premises depends on your organization’s specific requirements, compliance needs, and infrastructure preferences. Feel free to ask if you need further details! 😊 🚀

You

What are the limitations of SPFx in SharePoint Online?

Copilot

Certainly! While the SharePoint Framework (SPFx) offers several advantages for client-side development, it does have some limitations:

1. **iFrame Integration**:

   - SPFx works within a separate iFrame, which can impact performance.

    - It requires additional authentication, calls to SharePoint data, and loading JavaScript files, making it slower than script editor web parts.

2. **Responsive Designs and Theming**:

    - Creating responsive designs and inheriting CSS and theming information can be more challenging with SPFx.

Despite these limitations, SPFx remains a powerful customization and extensibility model for SharePoint Online, allowing developers to build modern, responsive experiences. If you have any more questions, feel free to ask! 😊 🚀

You

What are the best practices for SPFx development?

Copilot

Certainly! When developing with SharePoint Framework (SPFx), consider the following best practices:

1. **Set Up Your Development Environment**:

    - Install Node.js (v18 LTS is recommended for SPFx).
    - Choose a code editor (e.g., Visual Studio Code or WebStorm) for building your web parts.

2. **Optimize Performance**:

    - Always use release builds in production.
    - Avoid including third-party libraries in the bundle; load them from a CDN or an organization-owned hosting location.

3. **Leverage Modern JavaScript Frameworks**:

    - Use React, Angular, or other modern JavaScript libraries to build interactive and dynamic SPFx web parts.

4. **Separation of Concerns**:

    - Organize your solution structure to share common code and components.
    - Follow the “Separation of Concern” concept when dealing with multiple web parts.

Remember that these practices enhance productivity, maintainability, and performance in your SPFx development. If you need further guidance, feel free to ask! 😊 🚀

You

Can you explain SharePoint Framework extensions?

Copilot

Certainly! SharePoint Framework (SPFx) Extensions allow you to **extend** the SharePoint user experience in various ways. Here’s a brief overview:

1. **Application Customizers**:

    - Add custom scripts to pages.
    - Extend well-known HTML element placeholders with custom renderings.

2. **Field Customizers**:

    - Modify views for fields within lists.

3. **Command Sets**:
    - Add new actions to SharePoint command surfaces.
    - Implement client-side behaviors.

4. **Form Customizers**:
    - Associate and override default new, edit, and view forms for lists and libraries with custom forms.

SPFx Extensions work within modern pages and document libraries, using familiar SPFx tools and libraries. You can build them with modern JavaScript frameworks like React or plain JavaScript. Remember, avoid dependencies on page DOM structure or CSS styles, as they may change . If you need more details, feel free to ask! 😊 🚀