Best Practice to backup Website in Azure-(IIS)_ocTony

Backing up a website in Azure involves following best practices to ensure data integrity, availability, and security. Here are steps and practices to consider:

- 1. **Use Azure Backup Service:** Azure offers a dedicated backup service that integrates seamlessly with various Azure resources, including websites. Utilize Azure Backup to create and manage backup policies for your website.
- Define Backup Schedule: Establish a backup schedule based on your website's update frequency and criticality. Consider daily or weekly backups depending on your needs. Azure Backup allows you to set up automatic backup schedules for convenience.
- 3. **Choose Backup Retention:** Determine how long you want to retain backup data. Azure Backup allows you to define retention policies, such as retaining daily backups for a week, weekly backups for a month, etc. Align retention policies with your data recovery requirements.
- 4. **Implement Incremental Backups:** Opt for incremental backups to reduce storage costs and backup duration. Azure Backup only backs up changes made since the last backup, minimizing data transfer and storage requirements.
- 5. **Enable Backup Alerts:** Configure backup alerts to receive notifications for successful or failed backup operations. Monitor backup job statuses regularly to ensure backups are running as expected.
- 6. **Store Backups Offsite:** Follow the 3-2-1 backup rule by storing backups in multiple locations. In addition to Azure storage, consider storing backups offsite for added redundancy and disaster recovery preparedness.
- 7. **Encrypt Backup Data:** Enable encryption for backup data to protect sensitive information during storage and transit. Azure Backup supports encryption to safeguard your backups against unauthorized access.
- 8. **Test Backup and Recovery:** Regularly test backup and recovery processes to verify data integrity and readiness for restoration. Conduct simulated recovery drills to ensure backups are reliable and accessible when needed.
- 9. **Document Backup Procedures:** Maintain detailed documentation of your backup procedures, including schedules, retention policies, encryption settings, and recovery steps. Documenting backup processes facilitates troubleshooting and ensures consistency across backup operations.
- 10. **Review and Update Backup Strategy:** Periodically review and update your backup strategy to adapt to evolving business needs, regulatory requirements, and technological advancements. Stay informed about Azure Backup features and best practices for continuous improvement.

By following these best practices, you can effectively backup your website in Azure, ensuring data protection, availability, and resilience against potential data loss incidents.

To create a backup of the configuration settings in Internet Information Services (IIS),

you can use the Configuration Editor tool in IIS Manager. Follow these steps:

1. Open IIS Manager:

- Press Win + R, type inetmgr, and hit Enter.
- Alternatively, you can search for "Internet Information Services (IIS) Manager" in the Start menu.

2. Navigate to Configuration Editor:

- In the left-hand Connections pane, expand your server node.
- Click on the server node to highlight it.
- In the main window, you'll see various management options. Look for "Configuration Editor" under the Management section and double-click it.

3. Select Configuration Section:

- In the Configuration Editor, you'll see a **drop-down** menu labeled "Section." Click on this menu and select the configuration section you want to back up.
- For example, you might choose "system.webServer" or "system.applicationHost."

4. Export Configuration:

• After selecting the desired configuration section, **click on the "Export" link located** in the Actions pane on the right side.

5. Specify Backup Location and Name:

- In the Export Configuration window, choose a location to save the backup file.
- Enter a name for the backup file. You can use a meaningful name that reflects the configuration settings being backed up.

6. Save the Backup:

• Click the "OK" button to save the backup file to the specified location.

Create a subsite. <u>http://192.168.29.117/origence/</u> or http://octony.com/Azuresite/

1. Open IIS Manager:

- Press Win + R, type inetmgr, and hit Enter.
- Alternatively, you can search for "Internet Information Services (IIS) Manager" in the Start menu.
- 2. Navigate to the Parent Site:

- In the left-hand Connections pane, expand your server node.
- Expand the "Sites" node to view the list of websites hosted on your server.
- Locate and select the parent site under which you want to create the subsite.

3. Add a New Application:

• Right-click on the parent site and choose "Add Application" from the context menu.

4. Configure the New Application:

- In the Add Application window, you'll need to specify several settings:
 - Alias: Enter a name for your subsite. This will be part of the URL used to access the subsite.
 - Physical path: Specify the physical location where the subsite's files will be stored. You can create a new folder for this purpose if needed.
 - Application pool: Choose an existing application pool or create a new one for the subsite.

5. Configure Other Settings (Optional):

• You can configure additional settings such as the .NET CLR version, managed pipeline mode, and other options based on your requirements. These settings are available in the Add Application window.

6. Save and Test the Subsite:

- Click the "OK" button to create the subsite.
- Once the subsite is created, you can test it by accessing it through a web browser using the URL formed by combining the parent site's URL and the alias you specified for the subsite.

RE-direct subsite to different subsite using web.config page.

To create a redirect in IIS using code, you can use the web.config file of your website or application. The web.config file allows you to configure various settings, including URL redirects. Below is an example of how to create a redirect from one subsite to another subsite using the web.config file:



In this example:

- enabled="true" enables the HTTP redirect.
- **destination="http://example.com/newsubsite**" specifies the destination URL to which the redirect will send users.
- httpResponseStatus="Permanent" indicates that the redirect is permanent (HTTP status code 301). You can use **Temporary** for a temporary redirect (HTTP status code 302) if needed.

Here's a breakdown of the steps to apply this redirect:

- 1. Open your website's or application's web.config file:
 - You can find the web.config file in the root directory of your website or application.
- 2. Add the HTTP redirect code inside the <system.webServer> section:
 - If the <system.webServer> section doesn't exist in your web.config file, you can add it.
 - Insert the <httpRedirect> element with the desired redirection settings as shown in the example above.
- 3. Save the web.config file:
 - After adding the redirect code, save the web.config file in the appropriate location.
- 4. Test the redirect:
 - Access the original subsite URL in a web browser. It should automatically redirect to the specified destination URL.

Keep in mind the following points:

- Make sure to replace http://example.com/newsubsite with the actual URL of the subsite you want to redirect to.
- Adjust the httpResponseStatus attribute based on whether you want a permanent or temporary redirect.

OR CODE index/default page:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="refresh" content="0;url=https://finance.yahoo.com/">
<title>Redirecting...</title>
</head>
```

```
-->
```

Change default setting page IIS

1. Open IIS Manager:

- Press Win + R, type inetmgr, and hit Enter.
- Alternatively, you can search for "Internet Information Services (IIS) Manager" in the Start menu.
- 2. Select Your Website:
 - In the left-hand Connections pane, expand your server node and then expand the "Sites" node.
 - Select the website for which you want to change the default setting page.

3. Open Default Document Settings:

• In the main window, double-click on the "Default Document" icon, or right-click on it and choose "Open Feature" or "Edit Feature Settings."

4. Add or Edit Default Documents:

- In the Default Document settings, you'll see a list of existing default documents.
- To change the default setting page, you can either:
 - Add a new default document by clicking "Add..." in the Actions pane on the right and specifying the filename (e.g., index.html, default.aspx, etc.).
 - Edit an existing default document by selecting it from the list and clicking "Edit..." in the Actions pane.

5. Set Priority (Optional):

• You can change the order of default documents by using the up and down arrows in the Actions pane. The topmost document in the list has the highest priority.

6. Save Changes:

- After adding or editing the default documents, click "OK" to save the changes.
- 7. Test the Default Setting Page:

- Open a web browser and navigate to your website without specifying a particular page in the URL (e.g., http://yourdomain.com/).
- Verify that the new default setting page is served correctly.

set up log file limits to manage the size and retention of log files.

Here's how you can configure log file limits in IIS:

- 1. Open IIS Manager:
 - Press Win + R, type inetmgr, and hit Enter.
 - Alternatively, you can search for "Internet Information Services (IIS) Manager" in the Start menu.
- 2. Select Your Website:
 - In the left-hand Connections pane, expand your server node and then expand the "Sites" node.
 - Select the website for which you want to configure log file limits.

3. Open Logging Settings:

 In the main window, double-click on the "Logging" icon, or right-click on your site and choose "Manage Website" > "Advanced Settings." Then, find the "Logging" section.

4. Configure Log File Limits:

- In the Logging settings, you'll see various options related to logging, including where log files are stored and their format.
- To set up log file limits, look for the following options (the exact wording may vary depending on your IIS version):
 - **Log File Rollover**: This option determines how often IIS creates a new log file. You can choose options like daily, weekly, hourly, or when the log file reaches a certain size.
 - Log File Truncate Size: If you choose the option to roll over logs based on size, you can specify the maximum size (in bytes) before a new log file is created.

5. Save Changes:

- After configuring the log file limits according to your preferences, click "OK" or "Apply" to save the changes.
- 6. Verify Log File Limits:

• To verify that the log file limits are applied correctly, you can check the log file settings again or monitor the log files in the specified directory to ensure they are being managed as expected.

Powershell to backup config IIS.

You can use PowerShell to backup the configuration file of Internet Information Services (IIS). Here's a PowerShell script that demonstrates how to create a backup of the IIS configuration file: **SUCCESS**

| <pre>\$sourcePath = "C:\inetpub\wwwroot\web.config"</pre> |
|--|
| <pre>\$destinationPath = "c:\Backup\Backupconfig.page"</pre> |
| if (Test-Path \$sourcePath) { |
| # Copy the configuration file to the destination path |
| Copy-Item -Path \$sourcePath -Destination \$destinationPath -Force |
| |
| <pre># Check if the backup was successful</pre> |
| if (Test-Path \$destinationPath) { |
| Write-Host "IIS configuration file backed up successfully to |
| \$destinationPath" |
| } else { |
| Write-Host "Failed to create backup. Please check permissions or |
| file paths." |
| } |
| } else { |
| Write-Host "Source configuration file not found. Please check the |
| path: \$sourcePath" |
| |

- \$sourcePath specifies the path to the original configuration file of IIS (applicationHost.config).
- **\$destinationPath** specifies where the backup copy of the configuration file will be saved.
- **Test-Path** is used to check if the source file exists before attempting to create a backup.
- **Copy-Item** is used to copy the configuration file from the source path to the destination path.
- The **-Force** parameter in **Copy-Item** ensures that the backup file is overwritten if it already exists in the destination path.
- The script includes error handling to check if the backup was successful and provides appropriate feedback.

Make sure to replace the **\$sourcePath** and **\$destinationPath** variables with the actual paths you want to use for the source configuration file and the backup destination, respectively. Also, ensure that you have appropriate permissions to read the source file and write to the destination folder.



Powershell to install Notepad++ remotely & 2lines for local

Enable PowerShell Remoting (if not already enabled)

Enable-PSRemoting -Force

Connect to the remote server

Enter-PSSession -ComputerName http://192.168.29.117/ -Credential Administrator

Download Notepad++ installer

Invoke-WebRequest -Uri "https://github.com/notepad-plus-plus/notepad-plusplus/releases/download/v7.7.1/npp.7.7.1.Installer.x64.exe" -OutFile "C:\Temp\npp.7.7.1.Installer.x64.exe"

Install Notepad++ silently

Start-Process -FilePath "C:\Temp\npp.7.7.1.Installer.x64.exe" -ArgumentList "/S" -Wait

Exit remote session

Exit-PSSession
