**Redundancy:**

Redundancy is the purposeful duplicate of a system's vital components or operations with the objective of boosting system reliability, typically in the form of a backup or fail-safe, or to boost real system performance.

There are various types of redundancy such as Hardware Redundancy, Software Redundancy, Data Redundancy, Communication Redundancy, Organizational Redundancy, and Information Redundancy.

**For DMS purposes:**

**NGINX:**

* Popular web servers, reverse proxy servers, and load balancers which can be utilized to give web applications and services redundancy and high availability.
* It can divide incoming network traffic among several of them to ensure that the service is still available even if one or more backend servers fail.
* The following are some typical techniques and attributes of NGINX for achieving redundancy:
  + **Failover:** Carry out backend server health checks, and immediately reroute traffic from problematic or down servers
  + **Redundancy Configuration:** In the case of one NGINX server going down, you can achieve redundancy by running multiple servers in parallel with synced configurations.
  + **Reverse Proxy:** Carries out the function of a reverse proxy, acting as a bridge between two servers. which by shielding the backend servers from direct internet exposure can help to increase security and redundancy.

**Django-dbbackup:**

* It offers administration commands to support backing up and restoring the project's media files and database to various storages, including the cloud or local file storage.
* It enables users to archive with compression, deal easily with remote archiving, secure your backup with signature and encryption, and maintain the most recent version of the development database.
* There are several other tools available for managing automated backups which work in tandem with this solution.
* It has two main functions:
  + **dbbackup**: Creates a database backup to the designated storage. By default, all databases will be backed up, and any previous backups won't be deleted. A separate server can be used as an option for the backup.
  + **dbrestore**: Restores a database from the designated storage. By default, this will search for and restore from the most recent backup. It is possible to restore data from a database image made on a different server.

**Dual Stack Storage:**

* This approach is manual and not particular to Django.
* This strategy is storing the media files at two different locations for redundancy and archival purposes.
* This technique is mostly used to backup material for long-term storage needs.
* This approach, when combined with the file handling and encryption security features of Django, gives files enough redundancy and protection.

**vSphere Scheduled Task:**

* It helps to construct scheduled tasks for actions to conduct automatically once or on a regular basis using the vSphere Scheduled Task feature of VMware vSphere.
* For a variety of tasks, including taking a snapshot of a virtual machine and backing up databases and configurations, it can be used to automate these processes on a timely basis.
* To ensure redundancy and continuous availability of virtual machines and in the event of a host failure, for instance, it can be used to schedule a task to automatically migrate virtual machines from one host to another.

**References:**

<https://en.wikipedia.org/wiki/Redundancy_(engineering)>

<https://docs.nginx.com/nginx/admin-guide/high-availability/ha-keepalived-nodes/>

<https://pypi.org/project/django-dbbackup/>

<https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.vcenterhost.doc/GUID-03B90638-6C70-4379-8CAF-D66866D115F4.html>

**Integration:**

Integration is the act of merging two or more pieces of hardware or software so that they operate as a single unit. It can also refer to the connection between data, applications, APIs, and devices in an IT organization to be more efficient, productive, and agile. For instance, you can utilize the Internet Information Services (IIS) web server and a service that can serve as a bridge such as WSGI or FastCGI to combine a Django server with a Windows server.

WSGI (Web Server Gateway Interface) and FastCGI are both protocols that enable communication between web servers and web applications. For creating web applications, WSGI provides a Python-specific interface, but FastCGI is a more universal protocol that may be used with any programming language.

For integration the Django Server with Windows Server, we are using **WFastCGI**. It is based on WSGI and FastCGI-based IIS-Python bridge which is created and maintained by Microsoft themselves. Through IIS, it offers an effective approach to manage requests and process pools.

The following are some advantages it over other solutions:

1. **Effective request management:** Using IIS, WFastCGI offers an effective method for managing requests and process pools.
2. **Compatibility:** Any Python web application or framework that supports WSGI can be used with WFastCGI.
3. **Integration:** WFastCGI was specifically designed for Python (Django) integration with Windows.
4. **Ease of Use:** WFastCGI is simple to install and set up, making it straightforward to use and easier to troubleshoot.
5. **Performance**: When compared to other Python web servers like mod\_python, WFastCGI performs better and processes resources faster.

**References:**

[https://pypi.org/project/WFastCGI/](https://pypi.org/project/wfastcgi/)

<https://stackoverflow.com/questions/3937224/differences-and-uses-between-wsgi-cgi-fastcgi-and-mod-python-in-regards-to-py>

<https://pyquestions.com/differences-and-uses-between-wsgi-cgi-fastcgi-and-mod-python-in-regards-to-python>