drives with redundancy: - Navigate to **Storage > Pools > Add**. Choose to create a new pool. Select all the data drives. Our standard configuration for, say, 8 drives might be RAID-Z2 (which can tolerate two drive failures). For 6 drives or fewer, RAID-Z2 is still good; if high performance is needed and capacity trade-off is acceptable, use mirrored pairs. The guide suggests an optimal layout based on drive count (and lists pros/cons).
- **Name the pool** (e.g., `tank1` or `mainpool`). Enable disk encryption if required by client policy (TrueNAS offers dataset-level encryption which we can turn on later; full pool encryption is also possible).
- **Advanced pool options:** We recommend enabling **SSD cache (L2ARC)** if the system has an NVMe or SSD spare to use for read cache and maybe a **ZFS Log device (SLOG)** if we expect a lot of synchronous writes (like for VMs or databases). The guide details: if using an SLOG, it should be a high-endurance NVMe. Also, set **ashift=12** (default for modern drives) to optimize 4K sector alignment. After confirming, TrueNAS will create the pool and all underlying vdevs.

3. Datasets and Organization:

Under the new pool, create **datasets** to organize data and apply specific settings: - We recommend separate datasets for different data types. For example: `tank1/media`, `tank1/backups`, `tank1/vm-disks`, `tank1/config`, etc. Each dataset can have its own properties. For instance, enable compression (lz4, on by default) on all datasets – it's lightweight and saves space. For a dataset that will store large media (videos, etc.), you might turn off deduplication (to save RAM and because media won't dedupe well) and set recordsize to 1M for sequential reads. For a dataset for VMs or databases, you might use a smaller recordsize (16K or 32K) for better I/O matching. The guide provides a table of recommended dataset settings: e.g., - Backups dataset: compression on, recordsize 128K, dedup off. - VM images dataset: compression on, recordsize 16K, sync=standard (or sync=always if using SLOG for safety), dedup off (unless lots of identical VMs). - General documents dataset: compression on, recordsize 128K, maybe enable dedup if the client stores a lot of similar files (use with caution due to high RAM needs). - **Snapshots:** TrueNAS allows easy snapshots. We outline a snapshot policy: e.g., take nightly snapshots of important datasets (and keep, say, last 7 daily, 4 weekly, 12 monthly). The guide shows how to configure periodic snapshot tasks in **Data Protection > Periodic Snapshot Tasks**. These provide on-site versioning (to recover from accidental deletions or ransomware). We explain naming schemes and how to prune old snapshots.

4. Network Shares Setup:

Most clients will need to access the NAS data from other computers. Depending on the environment, set up SMB (for Windows/Mac) or NFS (for Linux or ESXi, etc) shares: - Go to **Sharing**. For a Windows-heavy environment or mixed environment, create an **SMB share** for each dataset that should be shared. The wizard will prompt to enable the SMB service (do so). Provide a name for the share (e.g., "MediaDrive"). Set permissions accordingly: often, we create a group (like "family" or "staff") and assign read/write to that group on the dataset. TrueNAS can set the dataset permissions in the process (choose Windows ACL if using SMB). We emphasize to disable SMB1 (TrueNAS does by default now) and to enable the **recycle bin** option on shares if users want deleted files to be recoverable easily. - For more technical use (like hypervisor storage), set up **NFS shares**. E.g., an NFS share of `tank1/vm-disks` to a Proxmox or VMware host. Add an NFS share, restrict it by network or host (like only allow the Proxmox host's IP to mount it), and map all users to root or nobody as needed depending on use case. - If applicable, mention **AFP/Time Machine** for Mac backups (TrueNAS SCALE doesn't have AFP, but Time Machine can use SMB with the tmutil capability; we guide how to mark a share as a time machine backup target). - We note that all share access will be over only.

5. User Accounts and Access Control:

Under **Accounts**, create user accounts or import directory services: - In a small/home deployment, we manually create users for each person who will access (matching their Windows/Mac logins possibly) and assign them to groups (like a group per share or a general “nasusers” group). Set strong passwords or encourage use of their system credentials (with perhaps Active Directory integration). - In an enterprise setting, TrueNAS can join an Active Directory domain. The guide covers how to do that (under **Credentials > Directory Services**). If the client has AD, join it so that domain users can be granted access to shares without separate NAS credentials. We cover the required info: AD domain name, credentials, setting up an ID mapping, etc. After joining, one can assign domain users/groups permissions on datasets. - Ensure services (SMB especially) are configured to use the directory service for authentication if needed. We mention enabling SMB “fruit” settings if Mac devices for better compatibility, and enabling NFS v4 with Kerberos if using AD for NFS auth (advanced topics given as optional steps). - TrueNAS SCALE can also enable **two-factor auth** for the web UI logins – we suggest enabling that for the root/admin account for extra safety.

6. Apps and Services (TrueNAS SCALE Apps):

One of the powerful features of SCALE is the ability to run containerized apps (through Kubernetes under the hood). We leverage this for additional services: - **Apps Catalog:** Go to **Apps** section. TrueNAS ships with an official catalog and one can add the community TrueCharts catalog. The guide instructs how to add TrueCharts (providing the URL, etc.), which offers many ready-to-deploy apps. - We recommend certain apps as part of our deployment: - **Nextcloud** (or similar) if the client wants a private cloud storage frontend. - **Plex** or **Jellyfin** for media serving (if media is part of the setup). - **Home Automation controllers** (if applicable) like HomeAssistant (though we might use a separate VM, but TrueNAS could run it too). - **Bitwarden (Vaultwarden)** for the password manager (if not running it in Proxmox, TrueNAS app could host it since it’s lightweight). - **Syncthing** for syncing files between devices, etc. - Example: Deploy **Vaultwarden (Bitwarden)** from TrueCharts. The guide shows the steps: select the app, set a dataset for it to store data (probably on an SSD pool or a dataset with small recordsize for database), set admin user, etc. Once deployed, the service is available on the network (we’d integrate it with our DNS so e.g. vault.local resolves to it). - Another example: If a client wants to use the NAS for virtualization, TrueNAS SCALE itself can host VMs or Kubernetes pods. However, since we already have Proxmox for heavy VMs, we tend to reserve TrueNAS’s VM feature for lightweight utility VMs or to not use it at all. We mention it for completeness but our standard is to let Proxmox handle VMs and TrueNAS handle storage and lightweight containers. - We stress on setting resource limits for apps where appropriate to avoid them consuming too much memory/disk.

7. Performance Optimization:

ZFS tuning and network tuning: - Enable **auto-snapshot pruning** and schedule scrubs monthly (TrueNAS usually auto-schedules scrubs every ~35 days; adjust to ~1 month). Scrubs help detect and heal bit rot with parity, but they are heavy – schedule them for a low usage time. - Check **SMART tests** schedule for disks (we set short tests weekly, long tests monthly). - If the NAS has 10GbE or faster, ensure MTU (jumbo frames) is consistent if using them. Often for simplicity, we stick to 1500 MTU unless a dedicated storage network vdevs: tune the **ARC size** (ZFS read cache in RAM) via TrueNAS advanced settings if the server also needs to run apps – maybe cap ARC to leave RAM for apps. By default TrueNAS will use a lot of RAM for ARC which is good for pure storage performance. - **Alerts:** TrueNAS has an alert system (disk failures, pool capacity >80%, etc.). Ensure email alert settings are configured (via root email or Alert Services – perhaps sending to our support email or the client's IT). We absolutely want to catch a degraded pool early to swap a disk. - Advise on **capacity management:** Through experience, ZFS pools shouldn't be over 80-90% full for best performance. We mention to monitor usage and plan upgrades if needed. Also, for expansion, we note that ZFS requires adding vdevs (not just single disks easily) so plan for future growth (e.g., leave some drive bays free).

8. Backup and Replication:

We include a section on backup strategies: - If the client has a second TrueNAS or an off-site NAS, we can configure **replication tasks**. TrueNAS can send ZFS snapshots to another TrueNAS over SSH. The guide would show how to set up a periodic replication of critical datasets to an off-site box or to cloud storage (TrueNAS can also sync to cloud, but since we focus on privacy, maybe replication to a physically secure device is preferred). - Additionally, remind about the rotated USB hard drives (if used): If the blueprint included rotating external backups, then outline the procedure: TrueNAS can periodically back up certain datasets to an external USB drive (via Rsync or ZFS send). When a drive is plugged in, an auto-rsync could kick off. This can be configured using autofs and a script or manual process – details beyond scope but concept mentioned.

By meticulously following the **TrueNAS Scale Configuration** guide, our team ensures that the client's NAS is not just a storage box, but a highly reliable data hub: data is organized and protected (ZFS snapshots and parity), accessible but secure (proper shares and permissions), and extensible (apps providing additional functionality on the same hardware). Given that data is the core of many operations, this guide emphasizes both **performance optimization (ZFS tuning)** and **data safety (redundancy, backups)**, aligning with the overall mission of privacy and resilience in our solutions.

WireGuard VPN Automation

This section describes our automated setup for **WireGuard VPN** and the accompanying scripts for client provisioning. WireGuard is a modern, lightweight VPN protocol known for its speed and robust encryption. We use it to provide secure remote access into the client's network (for themselves or for our maintenance). The automation scripts streamline the process of adding new VPN users/clients, generating their configs, and even producing QR codes for easy mobile device setup.

WireGuard Overview: WireGuard operates on the principle of public/private key cryptography. Each peer (server or client) has a key pair. The server (at the client's site, e.g. running on the firewall or a dedicated Raspberry Pi, etc.) listens on a UDP port (usually 51820) and has a list of allowed clients (identified by their public keys). It's ideal for our needs since it's fast (built into the Linux kernel) and has minimal attack surface. We configure WireGuard such that a connected client can access the secure internal network as if they were on-site, which is invaluable for remote monitoring and support, and for clients to connect to their "digital estate" while traveling.