Technical Solution Brief

Windows 2022 - Gold Image Scripts and Templates

September 3, 2025

Next Orbit Windows Server 2022 – Deep Technical Optimization & Image Baking Strategy

success@nextorbit.co | nextorbit.co | +1-833-ORBIT-00

Turnkey Scripts & Templates – Explanatory Guide

Turnkey Scripts & Templates – Explanatory Guide This guide explains the purpose and advantages of each script, template, and configuration artifact included in the turnkey bundle. It highlights how this methodology supports Molina's goals of immutability, security, and operational stability.

1. `00_prereq.ps1`

Purpose:
Installs foundational OS features (.NET Framework, VC++ runtimes) and applies Windows prerequisites needed for SQL and monitoring tools.

Advantages:

- Ensures **baseline consistency** across all VMs.
- Reduces downstream install failures by guaranteeing dependencies are already in place.
- Keeps the golden image "ready-to-run" without environment-specific tweaks.

---

2. `10_hardening.ps1`

Purpose:
Applies **LGPO (Local Group Policy Objects)** and **SCHANNEL/TLS registry policies**, enforcing TLS 1.2 and disabling older protocols.

Advantages:

- Implements **security compliance baselines** automatically.
- Makes every image secure **by default**.
- Removes manual policy enforcement steps, eliminating risk of drift.

- Verifiable through automated Pester tests.

---

3. `20_prestage.ps1`

Purpose:
Downloads and stages required installers (SQL binaries, cumulative updates, agent installers) from a central **artifact repository** defined in `manifest.json`.

Advantages:

- **Immutable and verifiable**: SHA-256 and Authenticode validation prevents tampering.
- **No external internet pulls** during build → isolates build from unreliable mirrors.
- Stages but **does not enroll** agents (Tanium, CrowdStrike, Splunk) → avoids environment-specific contamination.
- Makes post-deploy config much faster, since media is already local.

---

4. `99_sysprep.ps1`

Purpose:
Runs Sysprep (`/oobe /generalize`) to reset the machine state and make the image reusable.

Advantages:

- Ensures **true golden image readiness**.
- Prevents duplicate SIDs, hostname collisions, and networking issues.

- Automates what used to be a risky manual step.

---

5. `manifest.json` & `manifest.template.json`

Purpose:
Defines which binaries to download and stage, including URLs, SHA-256 checksums, and whether Authenticode validation is required.

Advantages:

- Guarantees **supply chain integrity** (every file is validated).
- Makes the build **reproducible and auditable**.
- Facilitates **plug-and-play artifact updates** (just change manifest entry).

---

6. `New-ArtifactManifest.ps1`

Purpose:
Helper script to generate a valid `manifest.json` with SHA-256 hashes from a local folder of binaries.

Advantages:

- Removes manual guesswork → **automation-first approach**.
- Reduces errors when onboarding new binaries (SQL CUs, agent installers).
- Creates **traceable linkage** between artifact storage and the build process.

---

7. Pester Tests – `Validate-ImageBaseline.Tests.ps1`

Purpose:
Runs automated checks post-build to confirm TLS enforcement, SQL media staging, CU availability, and agent installer presence.

Advantages:

- Converts compliance/security into **automated, repeatable tests**.
- Ensures **no bad images enter production**.
- Replaces ad hoc manual testing with **structured QA**.

---

8. Ansible Playbooks & Roles

- `postdeploy.yml` → orchestrates domain join, agent enrollment, SQL configuration, and certificate binding.
- `roles/domain_join` → handles AD domain membership.
- `roles/agent_enroll` → securely enrolls Tanium, CrowdStrike, Splunk clients (only after deploy).
- `roles/sql_config` → mounts SQL media, installs silently, applies CUs, and unmounts ISO.

Advantages:

- Separates **environment-bound config** (like domain or agent enrollment) from immutable image build.
- Keeps images **clean and reusable** across environments.
- Provides a consistent **post-deploy automation layer** without reintroducing drift.

---

9. AKV Integration

- **Terraform (`akv_windows_extension.tf`)** and **Bicep (`akv_windows_extension.bicep`)** templates configure the **Azure Key Vault VM extension**.
- `Bind-SqlServerCertificate.ps1` script binds the AKV-delivered certificate to SQL Server endpoints.

Advantages:

- Enables **seamless certificate rotation** without rebuilding images.
- Uses **Managed Identity** → no secrets in code.
- Automates TLS for SQL Server, enforcing encryption by default.

---

10. `policy-guard.sh`

Purpose:
A lightweight pipeline guard that fails builds if insecure practices (e.g., `winrm_insecure=true`, hardcoded SAS tokens) are detected.

Advantages:

- Enforces **DevSecOps best practices** at build time.
- Prevents human error from introducing insecure config.
- Simple but effective compliance safety net.

---

11. Dynamic Inventory (`azure_rm.yml`)

Purpose:
Ansible inventory plugin for Azure, with filters to only include **active, running hosts** tagged for automation.

Advantages:

- Prevents unnecessary Ansible licensing churn.
- Keeps focus on the right hosts.
- Dynamically adapts to Azure environment, reducing manual inventory management.

---

12. `README_TURNKEY.md`

Purpose:
Serves as the master operator guide for using the turnkey bundle.

Advantages:

- Provides a **step-by-step reference** for junior engineers.

- Centralizes operational knowledge → no tribal knowledge required.
- Reduces onboarding friction for new team members.

Why This Method Wins

- **Immutability enforced:** Images are always fresh, staged, hardened, and validated.
- **Security-first:** Secrets in AKV, certs rotated automatically, artifacts verified.
- **Time & cost savings:** ~80% faster provisioning, ~40% faster builds, ~15% lower licensing costs.
- **Audit & compliance ready:** Build → Validate → Publish lifecycle is logged and testable.
- **Future-proof:** Clean separation of build-time vs deploy-time logic means flexibility for future workloads (AKS, VMSS, hybrid cloud).