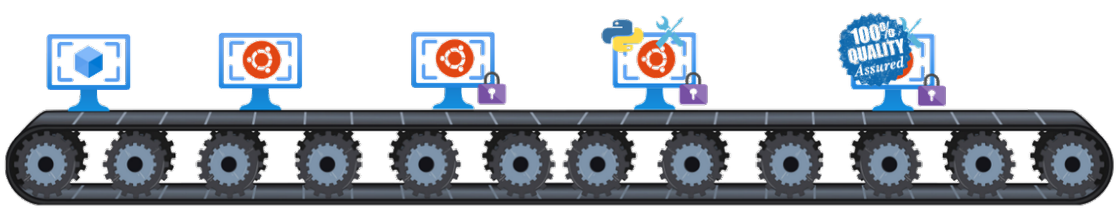
CONTINO

# AUTOMATION

## Azure Image Factory

### Define, Build, Deliver & Auto-Update CIS Hardened OS




**Azure Image Factory: Secured/Assured OS Images**

- using GittHub repositories, GitHub actions, Terraform, Packer, Azure and Ansible to auto-build OS images from the latest Azure base - including OS updates, Essential Tooling and CIS Role
- Deliver images consumable either directly or as the base for further customisation

Requirement

**Business Policy**
**Legal Requirement**
**Governance**

Policies

**Business Requirement**

While various elements of the business will require all manner of capability and customisation foer their development and delivery, a core facet will remain

*All the platforms will require the organmisational level of hardening and assurance - becuase not adhering to compliance invites a HIGH/Critical risk*

**Secure Foundations**

Given the current business standards. SecOps can focus on a policy based on CIS rules that when defined can be applied across any OS within it's class

The application of CIS roles should be an "end user experience" for security stajkeholders

OS Choice

**OS Choices**

Using either the microsoft tools or a site such as https://az-vm-image.info/ we set the OS Paramaters for defining and delivering a consumable hardened image

example **Publisher** *Canonical* **Offer** *UbuntuServer* ***SKU 18.04-LTS***

CIS automation

Security

**IaC Repeatable & Automated**

Using an array of online resources that define the CIS and other standards in code, the SecOps team can pour effort into creating an internal repository of IaC ansible roles - applicable for agreed OS choices **producing a baseline fpr each OS choice**

example : https://github.com/ansible-lockdown

Tooling

**Standardisaton**
Organisation standard tools and applications can be applied during the base OS image build
Part of standardisation could also be set validation of images and continual scanning

Requirement

**Business Policy**
**Legal Requirement**
**Governance**

Policies

using the **Business Defined OS Requirement** matched with approved **SecOps CIS Defintition** paramaters, we can initiate a custom defintion and hardening of this OS version

```
focal = {
  input_publisher = "Canonical"
  input_offer     = "0001-com-ubuntu-minimal-focal"
  input_sku       = "minimal-20_04-lts-gen2"
  cis_role_url    = "https://github.com/ansible-lockdown/UBUNTU20-CIS"
}
```

**Azure OS definition**

**Using the supplied OS and CIS roles, we define a unique consumable image in a shared image gallery**

**Azure OS update**

**version updates during periodic polling triggers new build process, incrementing & assuring a version**

1.2.3-
Major Minor Patch

**Dynamic Files**

To build each OS. we use parameters supplied, the OS and role and build files that are always up to date

**Packer Build**

**Using a standard packer manifest and files from terraform, we can opt to build the image via various options***


**Packer Version**

Packer delivers a version into the gallery for consumption - also triggering notifacation channels that can be used to trigger other workflows

**Module Initialisation**

- load the OS defintion fro the supplied Publisher, Offer, SKU
- load the OS defintinon "version" & store the value to indicate current base platform
- create an Image Definition in the gallery should one not exist

**Module Build Process**

- for the confirgurations defined in the repository
- if [[ local.version_file [[ <> [[ azure_os.get_latest_version ]]
  - buld manifest files using latest version
  - deploy VM using latest OS version
  - apply OS updates
  - apply standard software tool installations
  - apply the CIS role for the OS flavour
  - Generalise Image
- Create managed Image version (for validation)
  - alternatively images can be delivered into a "to be verified" gallery

Once the build is complete, the custom image repository can then process

- update the base OS version so we dont keep rebuilding
- update any release notes and version
- automatically branch/PR/Tag and release depending on requirement