LIVE! 360 TECH CON ORLANDO 2026 · TMHOL01 HANDS-ON LAB

PKI Lab Kit README

Build your lab before Sunday, November 15

> Hands-On Lab: Building Enterprise PKI from Scratch · John O'Neill Sr., Azure Innovators

You build your own lab environment **before Sunday**. On class day you build a production-style two-tier PKI inside it. The build is scripted: about 20 minutes of your attention, then about 75 minutes of waiting in Azure (longer on a laptop, mostly for the 16 GB of downloads).

What you get

```
Your Hyper-V host (Azure VM or your own laptop)
  Hyper-V switch PKI-Lab-NAT 10.201.0.0/24
    PKI-DC01    10.201.0.10   Windows Server 2025 Core      domain controller + DNS (pkilab.internal)
    PKI-ROOT01  10.201.0.11   Windows Server 2025 Desktop   offline root CA host (workgroup, no network, off)
    PKI-CA01    10.201.0.20   Windows Server 2025 Desktop   issuing CA host (domain member, Tier0\Servers)
    PKI-CL01    10.201.0.100  Windows 11 Enterprise         domain client (Workstations)
```

Lab users: `pkiadmin` (CA administrator), `certmgr` (certificate manager), `kraagent` (key recovery agent), `jsmith` and `mgarcia` (employees). No Certificate Services are installed anywhere yet: that is the class.

Everything uses Microsoft **evaluation** media (Windows Server: 180 days, Windows 11 Enterprise: 90 days), downloaded directly from Microsoft by the scripts. No product keys, no licenses to buy.

Choose how to host it

	Option A: Azure (recommended)	Option B: your own laptop
Works on	Any laptop with a modern browser (Windows, Mac, Linux, locked-down corporate)	Windows 11 **Pro, Enterprise or Education** with Hyper-V
You need	An Azure subscription you control: a new Azure free account ($200 credit, needs a credit card and phone for verification), or a pay-as-you-go, Visual Studio, or company sandbox subscription	16 GB RAM or more, ~150 GB free disk, virtualization enabled in firmware, local admin rights
Cost	A few dollars of your credit for the build and the class day (the host stops itself every night at 11 PM Eastern)	None
On class day	Connect from the browser (Azure Bastion, nothing to install)	Bring the laptop, fully charged, with its power adapter

Free account notes: one free account per person; if you have used one before, use a pay-as-you-go or company subscription instead. Free accounts are limited to about 4 vCPUs per region and can't be raised, which is why the host is a 4-vCPU `Standard_E4s_v5`. The $200 credit lasts 30 days from sign-up, so sign up no earlier than about **November 1**. Never use your employer's production subscription.

Use the account you signed up with. The email address you used to create the Azure free account owns the subscription and its billing. Use that same account for the portal, Cloud Shell, Bastion, and

every kit script. A Global Administrator role in Microsoft Entra ID does **not** give an account access to Azure subscriptions or billing, so any other account you create won't see your lab subscription. If you want to explore Entra ID administration in your new tenant, create an extra account for that, and keep using your sign-up account for the lab.

Keep the spending limit on. Don't select **Upgrade** and don't remove the spending limit. With the limit on, a free account stops its resources when the credit runs out instead of charging your card.

Option A: build in Azure

1. Sign in to the Azure portal **with the account you signed up with** and open **Cloud Shell** (the `>_` icon). The first time, choose **PowerShell**, leave **No storage account required** selected, check that your subscription is picked, and click **Apply**.
2. **Upload** `PKI-Lab-Kit.zip` (Manage files > Upload), then:

```
Expand-Archive ./PKI-Lab-Kit.zip -DestinationPath ./PKI-Lab-Kit -Force
cd ./PKI-Lab-Kit
./01-Deploy-AzureHost.ps1
```

Choose a password for the host's admin account (`pkilabadmin`) when asked. The script creates the host (about 10 minutes) and copies the kit to `C:\LabKit` on it. *If East US 2 has no room for the host that day, or your subscription isn't allowed that VM size there, the script tries other 4-vCPU sizes and then the next region on its list by itself (Central US, North Central US, West US, ...), cleaning up as it goes. You'll see yellow "Skipping" or "no room" lines: that's normal. When it finishes, it prints your region, host name and resource group: write them down.*

3. In the portal: **Virtual machines > your host** (the name step 1 printed, usually **PKI-Lab-EastUS2-HV01-vm**) **> Connect > Bastion**, choose the free **Developer** option, and sign in as `pkilabadmin`.
4. On the host, open **Windows PowerShell as Administrator**:

```
cd C:\LabKit
.\02-Prepare-HyperVHost.ps1      # installs Hyper-V and reboots: reconnect through Bastion and run it again
.\Build-Lab.ps1                  # choose ONE lab password, then wait about 75 minutes
```

5. When you see **LAB READY**, take a screenshot. Then in the portal **Stop** the host (it must say *Stopped (deallocated)*). Start it again on class morning.

Option B: build on your laptop

1. Copy the kit to a folder on the drive with the most free space and edit `LabConfig.psd1`:
   - `Hosting = 'Local'`
   - `HyperVHost.DataDriveLetter` and `RootFolder` to that drive, e.g. `'D'` and `'D:\Lab'` (or `'C'` / `'C:\Lab'`)
   - every `BaseImages.*.Iso` path to the same drive (`D:\Lab\ISO\...`)
   - `Network.UpstreamDns` to `'1.1.1.1'` or your router
2. Open **Windows PowerShell as Administrator** in that folder:

```
Set-ExecutionPolicy -Scope Process Bypass
.\02-Prepare-HyperVHost.ps1      # enables Hyper-V and restarts; run it again afterwards
.\Build-Lab.ps1
```

3. Screenshot **LAB READY**. Before class, keep PKI-ROOT01 off and close other apps: the three running VMs use about 11 GB.

Everyday commands (on the host, in the kit folder)

```
. .\LabCheckpoints.ps1          # load the helpers
Get-LabCheckpoint               # list checkpoint sets
Restore-Lab 'Lab-Start'         # start over from the clean starting point
Checkpoint-Lab '01-After-Root-CA'
.\Check-LabReady.ps1            # rerun the readiness check
```

Moving files to and from the **offline root** (it has no network on purpose):

```
$cred = Get-Credential Administrator
Copy-VMFile PKI-ROOT01 -SourcePath C:\Transfer\CA01.req -DestinationPath C:\Transfer\CA01.req `
    -FileSource Host -CreateFullPath
$s = New-PSSession -VMName PKI-ROOT01 -Credential $cred
Copy-Item -FromSession $s C:\Transfer\CA01.crt -Destination C:\Transfer\
```

After class

- Azure: delete the resource group step 1 printed, usually **PKI-Lab-EastUS2-rg** (portal > Resource groups > Delete). That removes everything and stops all charges.
- Laptop: `Get-VM PKI-* | Stop-VM -Force -Passthru | Remove-VM -Force`, then delete the lab folder.

Files

File	What it does
`README.md` / `README.pdf`	This guide (the PDF opens anywhere)
`LabConfig.psd1`	All settings (attendees normally change nothing for Azure)
`01-Deploy-AzureHost.ps1`	Azure only, in Cloud Shell: host VM, network, auto-shutdown, cost alert, copies the kit to the host
`02-Prepare-HyperVHost.ps1`	Hyper-V, data disk, lab switch and NAT
`Build-Lab.ps1`	Runs 03-05, powers off the root, takes the `Lab-Start` checkpoint, runs the readiness check
`03-Build-BaseImages.ps1`	Downloads the evaluation ISOs and builds the base disks
`04-New-LabVMs.ps1`	Creates the four VMs
`05-Configure-Lab.ps1`	Forest, users, groups, joins, offline root
`Check-LabReady.ps1`	PASS/FAIL readiness report
`LabCheckpoints.ps1`	Checkpoint/restore the whole lab as one set
`Complete-Lab.ps1`	Catch-up: `-Lab N` finishes labs 1..N for you and takes the checkpoint (`-Lab 6` repairs the LAB 6 faults)
`Break-Lab.ps1`	LAB 6 only: plants three faults (don't read it before LAB 6)
`06-Update-LabVMs.ps1`	Optional: Windows Update inside the VMs (not needed for class)

Questions before class: JONeillSr@azureinnovators.com