Leverage from Pester to automate Azure Storage Account testing

cc BY SA

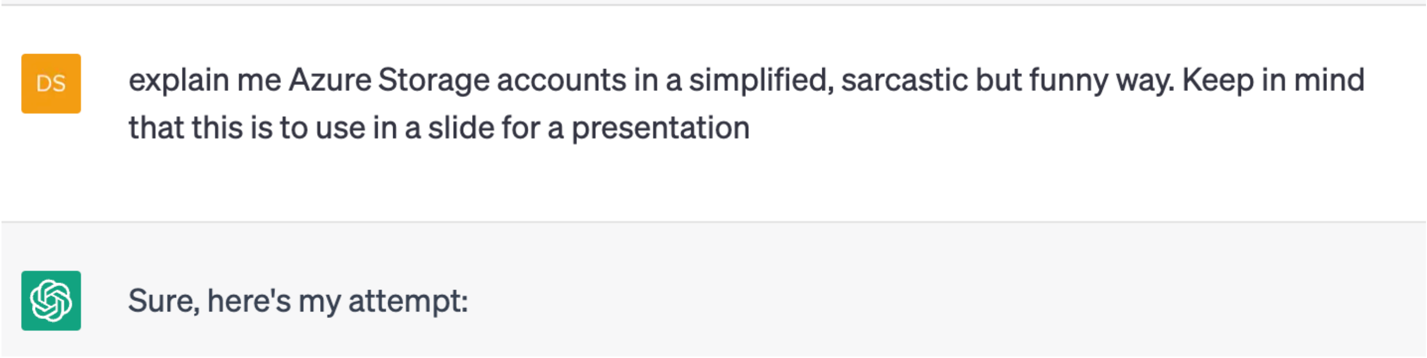
Get-UserInfo

- Daniel Silva
- Senior DevOps Engineer @ RELEX
- Focused in Azure & Terraform
- 1st time attendee
- Owner of 2 lovely cats
  - That love sleeping in weird positions

https://www.danielssilva.dev/ @DanielSilv9 in/danielssilva @DanielSSilva@techhub.social

cc BY SA

Hey ChatGPT…



DS

explain me Azure Storage accounts in a simplified, sarcastic but funny way. Keep in mind that this is to use in a slide for a presentation

Sure, here's my attempt:

cc BY SA

Azure Storage account

- “(…) It's like having an infinite closet that not only holds your files but can also be configured to let your friends (or foes) take a peek!”
- “You can set up a virtual bouncer to allow or deny access, like giving VIP treatment to your favourite IP addresses or forcing others to show a special pass (SAS tokens, anyone?)”

BY SA

The catch?

BY SA

Azure Storage account

- “Manually testing all these configurations can be a real nightmare. It's like playing a never-ending game of "Where's Waldo?" with your data and security settings.”

- “But don't worry, we've got a secret weapon called Pester that's going to make your life a whole lot easier!”

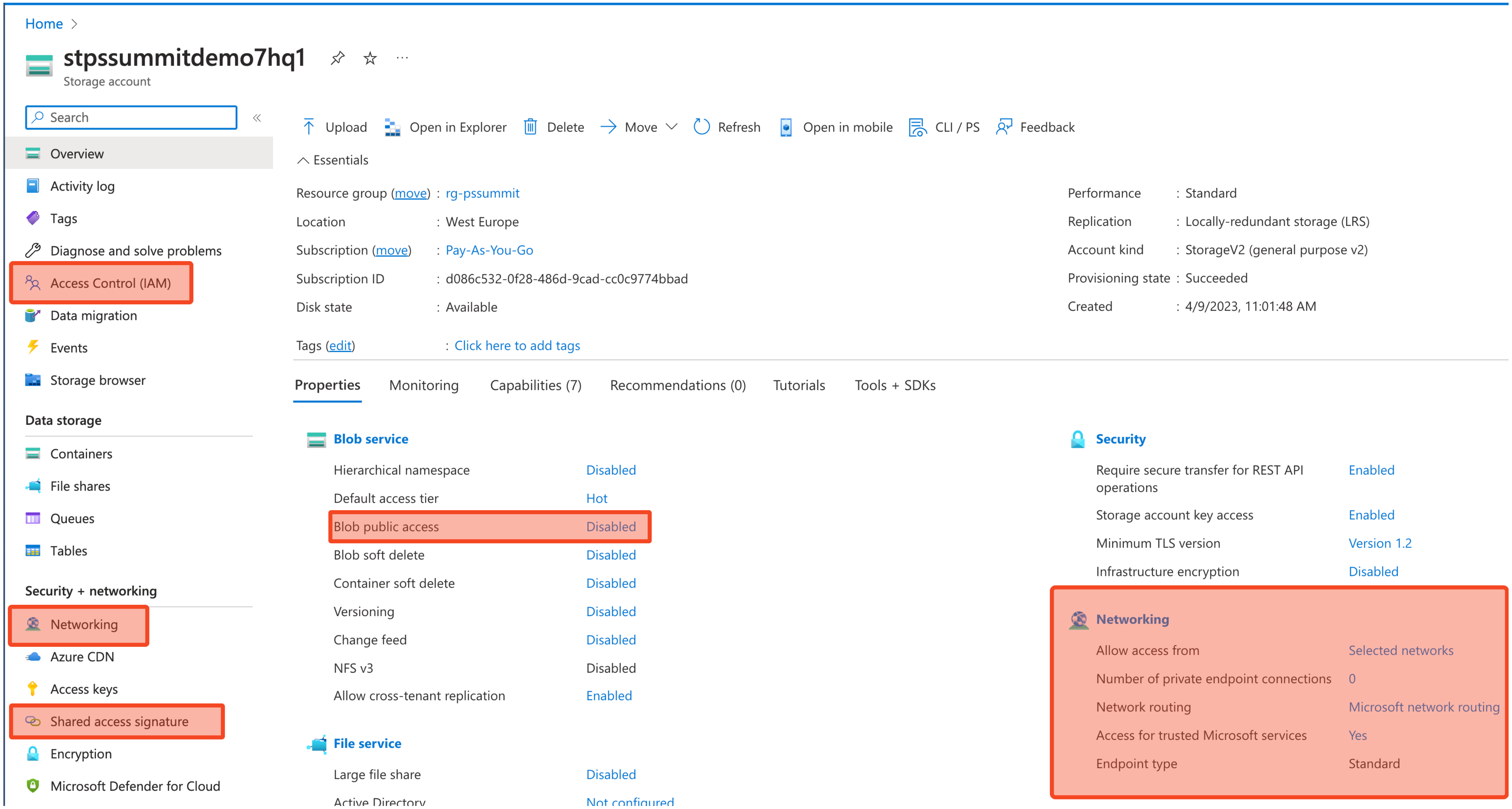
BY SA

Azure Storage account access challenges

- Network
  - Public access
  - Anonymous access
  - Firewall rules

- User
  - SAS Token
  - Service Principal
  - AAD Access

CC BY SA



stpssummitdemo7hq1
Storage account

Essentials

Resource group (move) : rg-pssummit
Location : West Europe
Subscription (move) : Pay-As-You-Go
Subscription ID : d086c532-0f28-486d-9cad-cc0c9774bbad
Disk state : Available
Tags (edit) : Click here to add tags

Performance : Standard
Replication : Locally-redundant storage (LRS)
Account kind : StorageV2 (general purpose v2)
Provisioning state : Succeeded
Created : 4/9/2023, 11:01:48 AM

Blob service

Hierarchical namespace	Disabled
Default access tier	Hot
Blob public access	Disabled
Blob soft delete	Disabled
Container soft delete	Disabled
Versioning	Disabled
Change feed	Disabled
NFS v3	Disabled
Allow cross-tenant replication	Enabled

File service

Large file share	Disabled
Active Directory	Not configured

Security

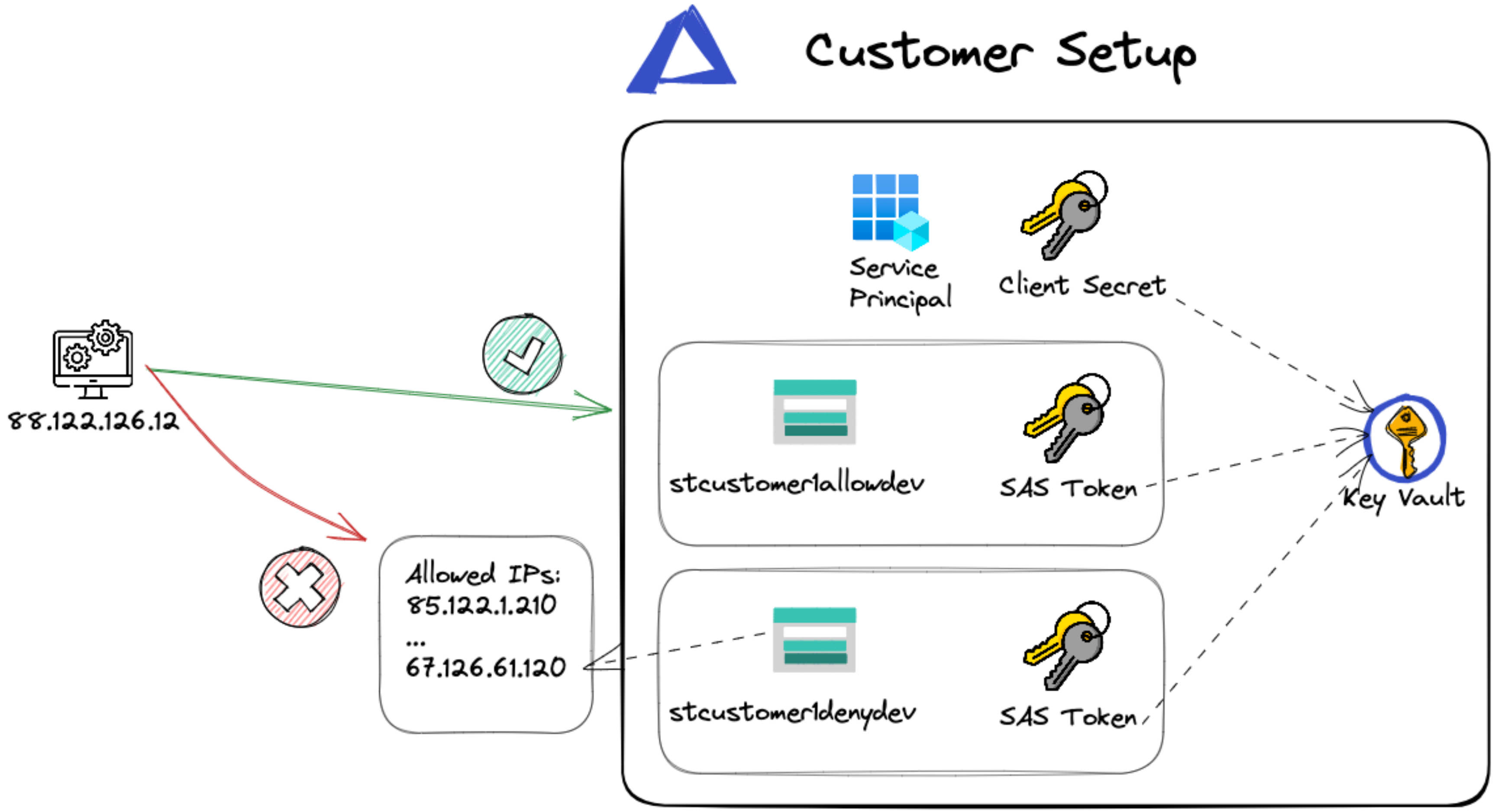
Require secure transfer for REST API operations	Enabled
Storage account key access	Enabled
Minimum TLS version	Version 1.2
Infrastructure encryption	Disabled

Networking

Allow access from	Selected networks
Number of private endpoint connections	0
Network routing	Microsoft network routing
Access for trusted Microsoft services	Yes
Endpoint type	Standard

BY SA

Example of a customer setup



CC BY SA

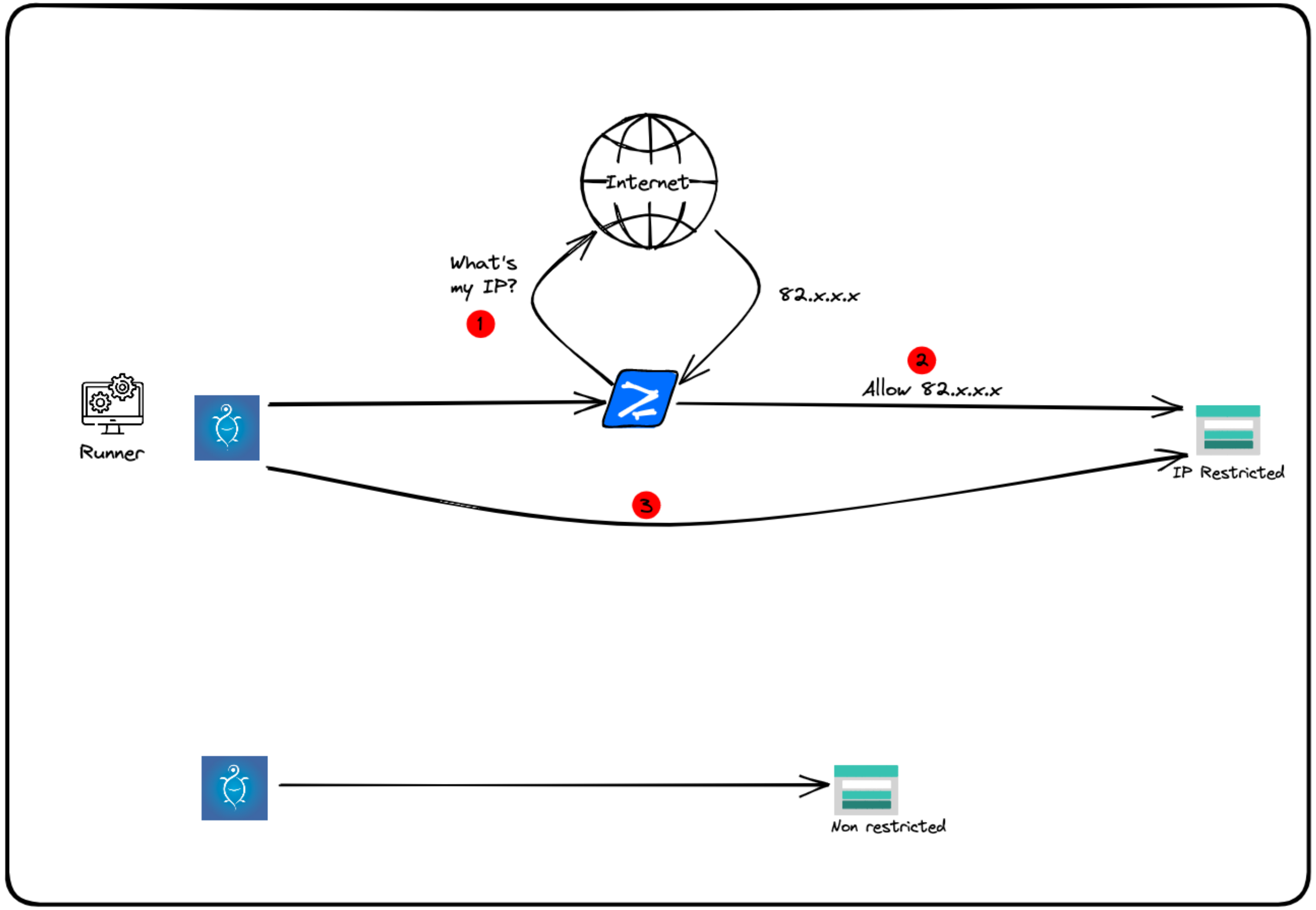
Storage accounts with Network rules

Storage accounts with Deny Network rules

- IP Whitelisting is only applied to Data Storage
  - Containers
  - Queues
  - …
- Only given IPs are allowed to access containers
- Our CI/CD runner shouldn't be allowed by default
  - But it still needs to have access so that terraform can get the state

cc BY SA

Storage accounts with Deny Network rules



1. Get current IP
2. Run Add-AzStorageAccountNetworkRule
3. Execute the remaining test flow
   - Run Remove-AzStorageAccountNetworkRule

cc BY SA

Setup 101

Terraform will be used to provision the infrastructure

A Service Principal is the identity for such provision

Naming Conventions

<resource abbreviation>-<customer name>-<environment>

Storage account: st<customer name><allow/deny><environment>

Secret names in Key Vault: <Service Principal name>-secret

cc BY SA

Testing scenarios

Scenario 1 – Test all storage accounts after a release, or test a single storage account

Upload a file

Using SAS Token

Using Service Principal

→

Download previously uploaded file

Using SAS Token

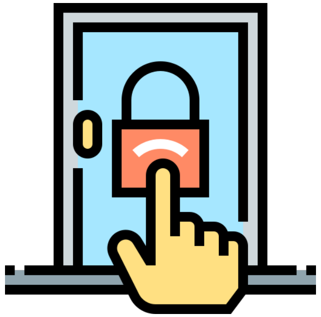
Using Service Principal

CC BY SA

Scenario 2 – Storage account report

For each Storage Account check if:

Only HTTPS traffic is allowed



Public access is disabled

Keys are not older than X days

cc BY SA

Thank you!

BY SA