Connecting to Redis Cache

Warner Chaves
MS DATA PLATFORM MVP

@warchav sqlturbo.com

What's in this module?

Redis Cache

Network Security Options

Authentication and Authorization

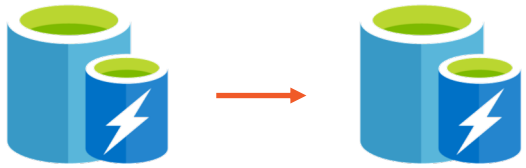
Secure Connection with Redis CLI

Redis Cache

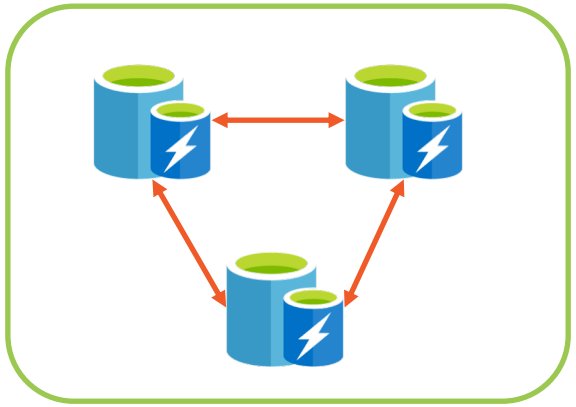
A high-throughput, low-latency, secure managed service of this open-source in-memory cache

Tiers

Basic



Standard



Premium

Service Name

Protocol: TCP

Port: 6379 or 6380 (SSL)

Server name:
*****.redis.cache.windows.net

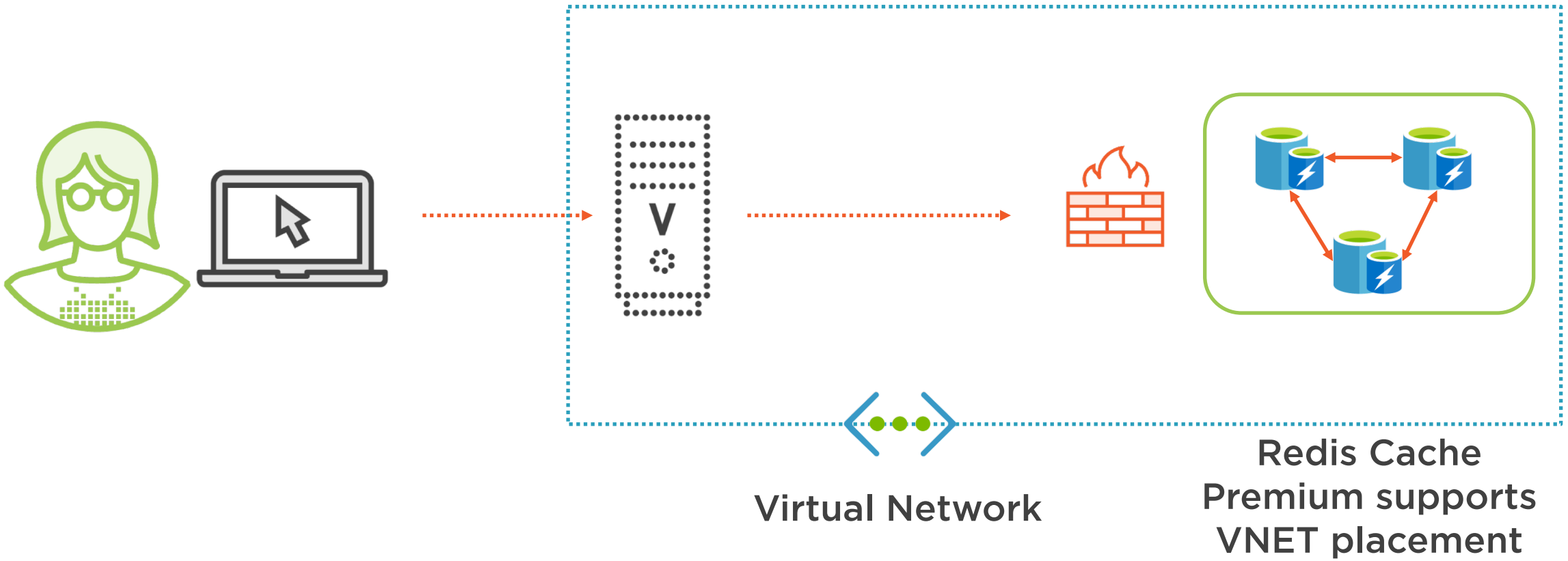
As a security best practice, do not enable non-SSL port connections

Network Security Options

Public Endpoint

Internet-facing endpoint

Resource Inside the VNET



Redis Cache does not currently offer Service Endpoints

Proper firewall configuration is critical, especially for public endpoint caches

Authentication and Authorization

Authorizing Access

Shared Keys

Root password of the cache

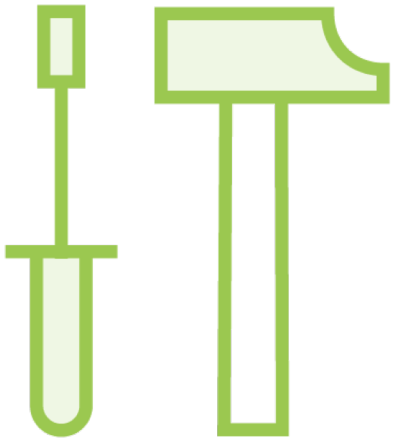
There are always 2 active keys

No identity authentication

Store Shared Keys securely as they provide full access to your cache

Secure Connection with Redis CLI

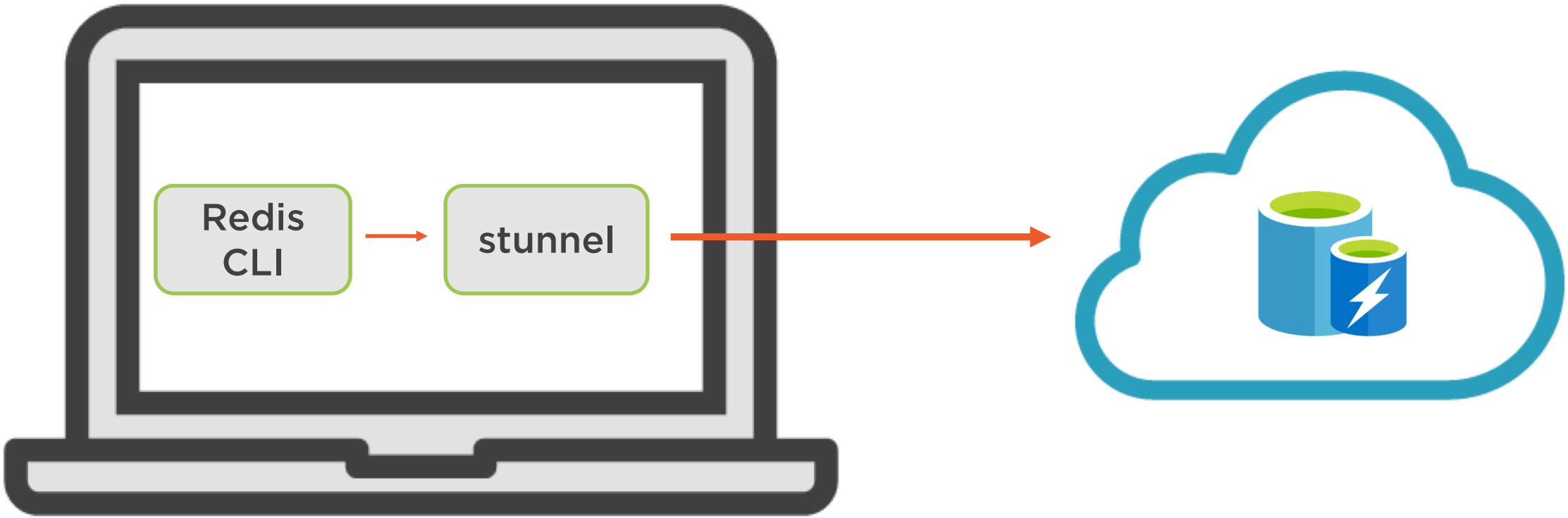
Tools



Redis CLI

stunnel

Connection Process



Putting Everything Together

It has a public endpoint and our client is on the Firewall

We have the server name

We are going to get the keys

We will connect with a Redis utility

Demo

Connecting Redis CLI to Azure Redis Cache

Summary

The service includes public endpoint or VNET placement

Access authorization is done through Shared Keys

Redis CLI is a popular utility to interact with a Redis Cache

Put it all together to connect Redis CLI to Azure Redis Cache

Continue on your Azure Developer Journey!