Microsoft

AZ-203.5 Module 04: Integrate caching and content delivery

Prashanth Kumar

Microsoft

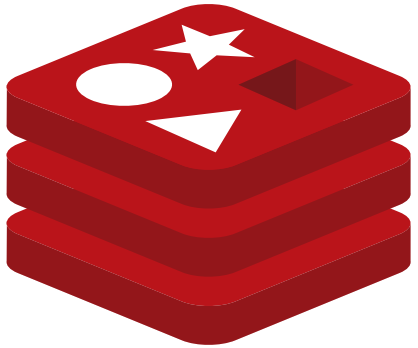
Topics

- Azure Cache for Redis
- Develop for storage on CDNs
- Lab: Monitoring services deployed to Azure

Lesson 01: Azure Cache for Redis

Redis

- Open-source NoSQL storage mechanism that is implemented in the key-value pair pattern common among other NoSQL stores
- Uniquely allows complex data structure for keys and values
- Commonly used as a cache mechanism and is referred to as Redis Cache
- Allows distribution of data in nodes and clusters



Redis data types

`KEY: "greeting"` `VALUE: "Welcome new users!"`

Key	Value
Nil	Test Value
school	School of Fine Art
school:grade:levels	["K", "1", "2", "3", "4", "5"]
school:teachers	**Key** **Value** "2" "Smith"
Binary JSON file	Binary JPEG

Example key schema

- In many Redis applications, you can have a cache server storing hundreds or thousands of key-value pairs
  - At which point it might be difficult to come up with a unique key name for each piece of data that you want to store
- The Redis team recommends that you use a schema to break up your keys into logical groups
  - In most cases, Redis users will use colons to design their schema

Key	Value
application:last_updated_string	January 01, 2016
gradelevels:three:avg_test_score	365
gradelevels:two:avg_test_score	415

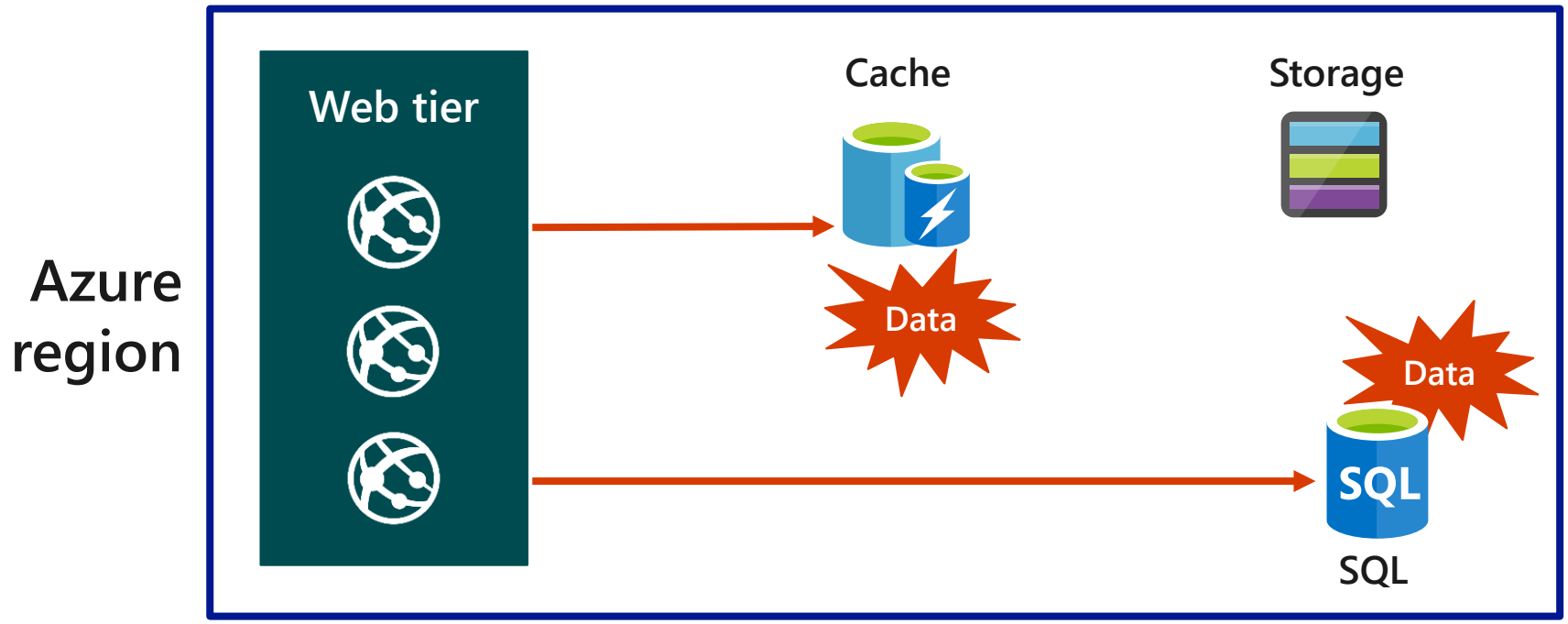
Redis operations

- `GET school:metadata:name`
  - Retrieves the value associated with any key
    - If the key does not exist, the command will return a special nil (null) value
- `SET school:metadata:name "School of Fine Art"`
  - Changes the value associated with the specified key
- `GETSET school:metadata:name "Bellows College"`
  - Changes the value associated with the specified key
  - Returns the previously associated value
- `EXISTS school:metadata:name`
  - Takes one or more keys as parameters
  - Returns an integer indicating the quantity of keys that have associated values

Azure Cache for Redis

- Managed service based on Redis that helps provide secure nodes as a service:
  - High degree of compatibility with existing tools and applications that already integrate with Redis
  - Already well documented with existing community-driven Redis documentation
- Offers three tiers of service:
  - **Basic**
    - Includes a single node
  - **Standard**
    - Includes two nodes in the primary replica configuration, and includes replication support and a Service Level Agreement (SLA)
  - **Premium**
    - Designed for enterprises with scale-out cache support and advanced persistence and clustering features

Azure Cache for Redis usage



Configuration

- You can create a Redis cache by using the Azure portal, the Azure CLI, or Azure PowerShell
- There are several parameters that you will need to decide to configure the cache properly for your purposes:
  - Name
  - Resource Group
  - Location
  - Pricing tier
  - Virtual network support
  - Clustering support

Accessing a Redis cache from a client

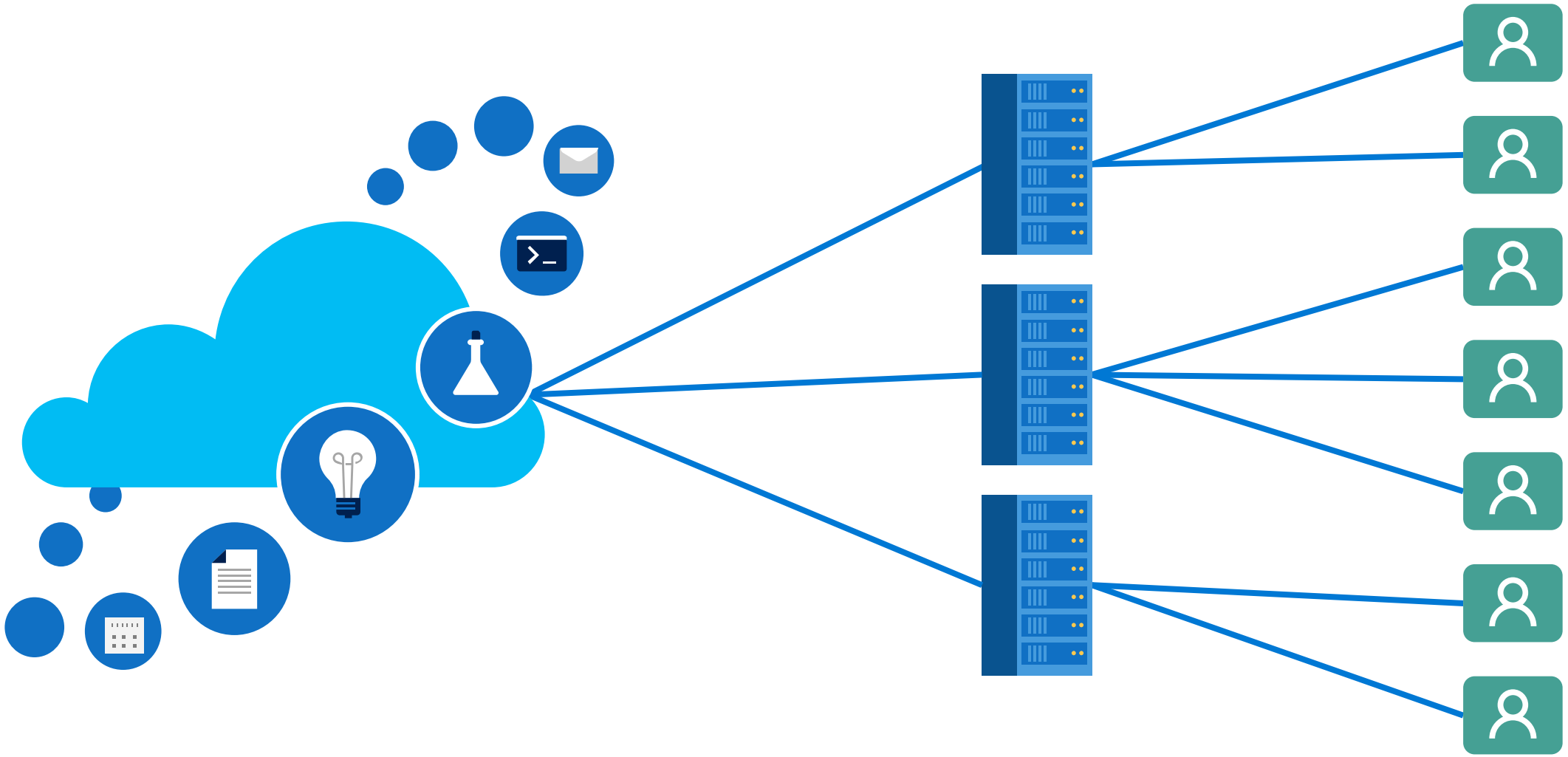
- To connect from a client, you will need:
  - Host name
  - Port
  - Access key
- All this information is available on the Azure portal

Lesson 02: Develop for storage on CDNs

Content delivery networks (CDNs)

- Distributed network of servers that can efficiently deliver web content to users
  - Stores cached content on edge servers that are close to users to minimize latency
  - Edge servers are located in point of presence (POP) locations that are distributed throughout the globe
- Typically used to deliver static content, such as images, style sheets, documents, client-side scripts, and HTML pages

Improving the client experience by using a CDN



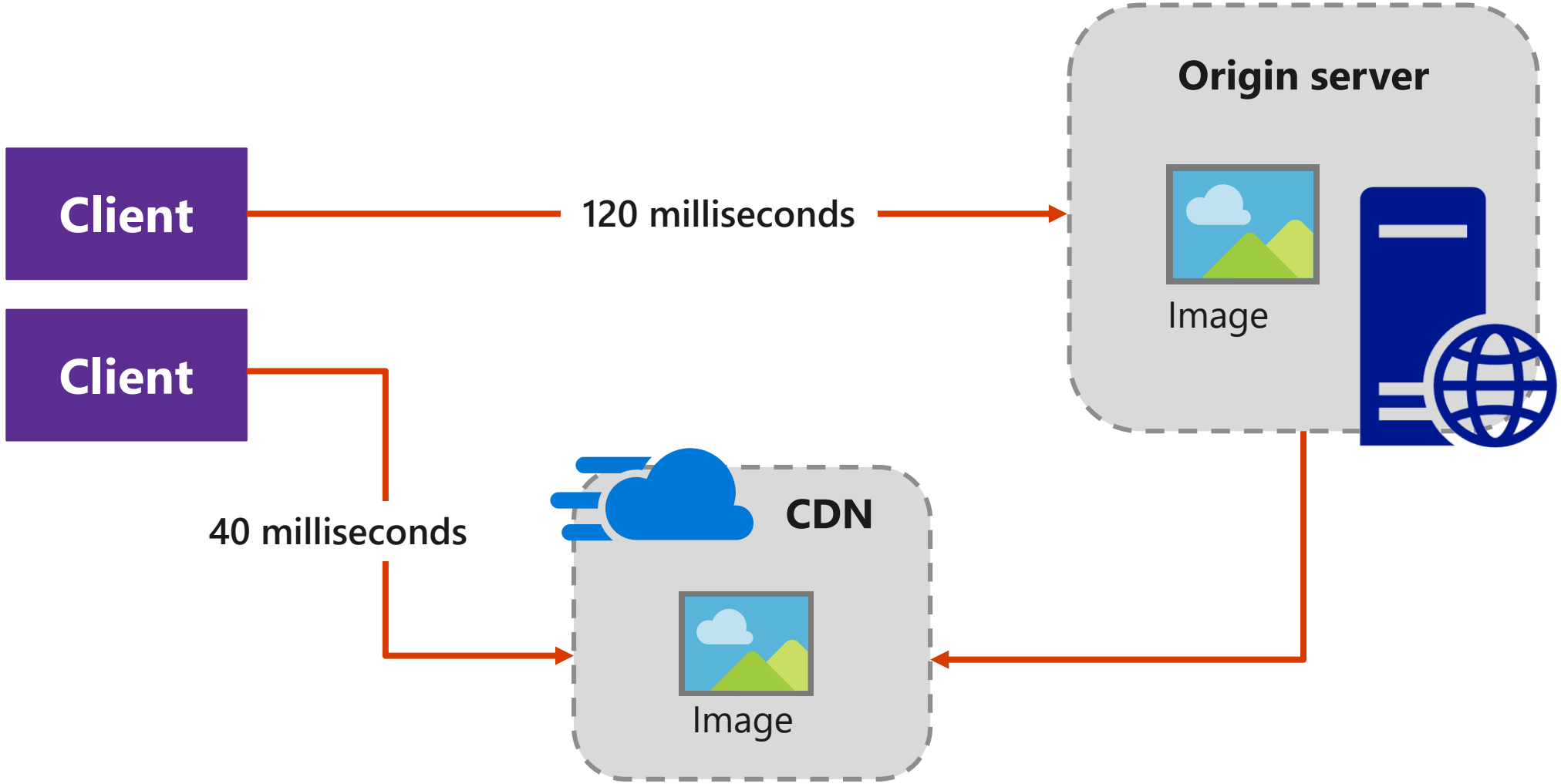
CDN uses

- Delivering static resources, often from a website, for client applications
  - Resources can be images, style sheets, documents, files, client-side scripts, HTML pages, HTML fragments, or any other content that the server does not need to modify for each request
- Delivering public static and shared content to devices such as mobile phones and tablets
- Serving entire websites that consist of only public static content to clients
  - Does not require any dedicated compute resources

CDN uses (continued)

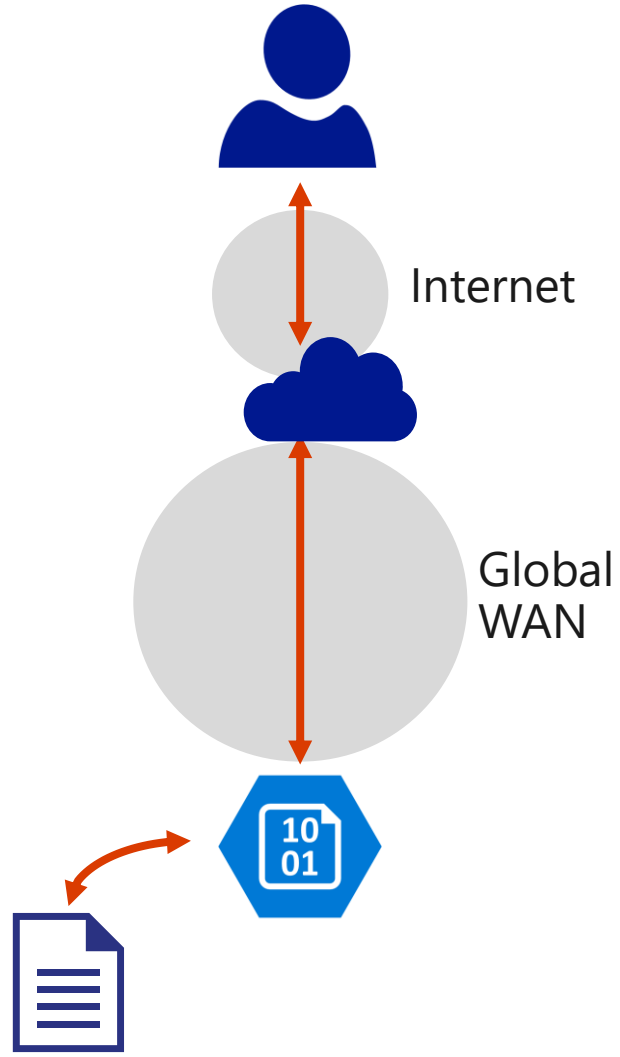
- Streaming video files to client devices on demand
  - Taking advantage of the low latency and reliable connectivity available from the globally located datacenters that offer CDN connections
- Supporting Internet of Things (IoT) solutions
  - The huge numbers of devices and appliances involved in an IoT solution can easily overwhelm an application if it has to distribute firmware updates directly to each device
- Coping with peaks and surges in demand without requiring the application to scale
  - Avoiding the consequent increased running costs associated with scale

CDN usage


Origin server
Client
120 milliseconds
Image
Client
40 milliseconds
CDN
Image

Azure CDN

- Global CDN solution for delivering high-bandwidth content that is hosted in Azure or in any other location
- Cache publicly available objects loaded from Azure Blob Storage, a web application, a virtual machine, or any publicly accessible web server
- Accelerates dynamic content, which cannot be cached, by taking advantage of various network optimizations by using CDN point-of-presence (POP) locations

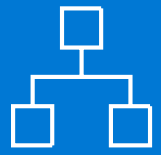


Azure CDN platform

Domain management

Origin load balancing

Caching and streaming



Policy

Service management

Optics and self-service

Locations

Performance

Network

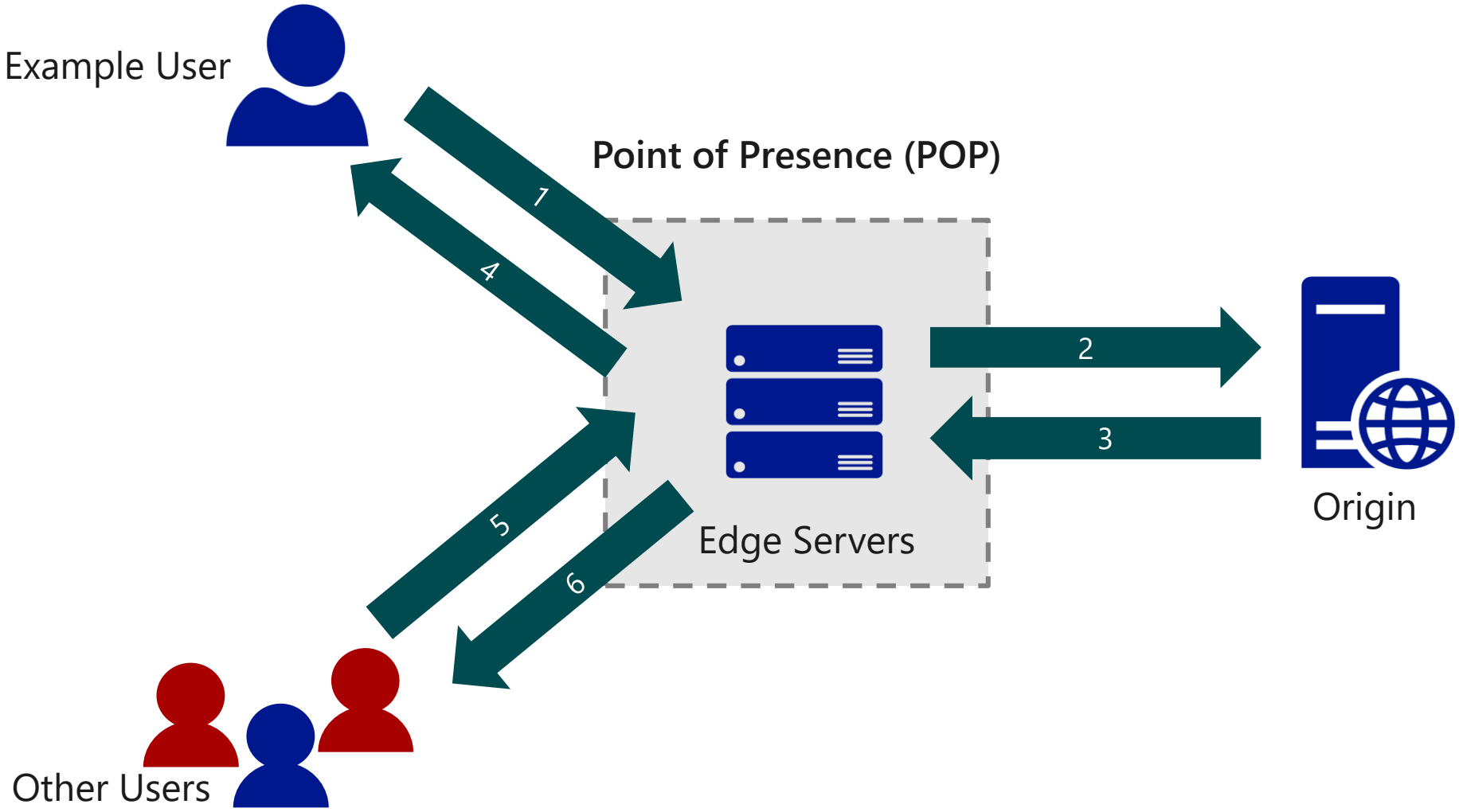
Strengths

verizon Digital Media Services

Akamai

Microsoft

Azure CDN usage



Manage Azure CDN profiles by using Azure CLI

```
az cdn profile list
```

```
az cdn profile list --resource-group ExampleGroup
```

```
az cdn profile create --name DemoProfile --resource-group ExampleGroup –sku
Standard_Akamai
```

You can customize further by using one of the following options:

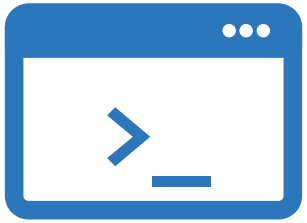
Custom_Verizon

Premium_Verizon

Standard_Akamai

Standard_ChinaCdn

Standard_Verizon



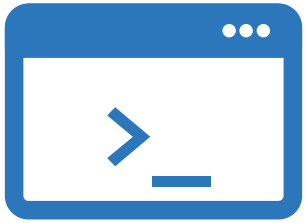
Create Azure CDN endpoints and domains by using Azure CLI

Create a CDN endpoint:

```
az cdn endpoint create --name ContosoEndpoint --origin www.contoso.com --profile-name
DemoProfile --resource-group ExampleGroup
```

Associate a domain with an endpoint:

```
az cdn custom-domain create --name FilesDomain --hostname files.contoso.com --endpoint-
name ContosoEndpoint --profile-name DemoProfile --resource-group ExampleGroup
```



Cache expiration in Azure CDN

- Azure CDN caching rules specify cache expiration behavior both globally and with custom conditions. There are two types of caching rules:
  - **Global caching rules**. You can set one global caching rule for each endpoint in your profile that affects all requests to the endpoint. The global caching rule overrides any HTTP cache-directive headers, if set.
  - **Custom caching rules**. You can set one or more custom caching rules for each endpoint in your profile. Custom caching rules match specific paths and file extensions, are processed in order, and override the global caching rule, if set.
- For global and custom caching rules, you can specify the cache expiration duration in days, hours, minutes, and seconds

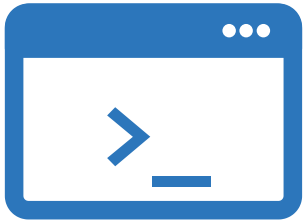
Purging and preloading assets by using Azure CLI

Purge assets from an endpoint:

```
az cdn endpoint purge --content-paths '/css/*' '/js/app.js' --name ContosoEndpoint --
profile-name DemoProfile --resource-group ExampleGroup
```

Preload assets into an endpoint:

```
az cdn custom-domain create --name FilesDomain --hostname files.contoso.com --endpoint-
name ContosoEndpoint --profile-name DemoProfile --resource-group ExampleGroup
```



Microsoft

Review

- Azure Cache for Redis
- Develop for storage on CDNs
- Lab: Monitoring services deployed to Azure