

Introduction to Regular Expression (Regex)

USING WILDCARDS AND LITERAL MATCHES



Josh Duffney

DEVOPS ENGINEER

@joshduffney <http://duffney.io/>



Intended
Audience

IT Professionals

Scripters

Automaters

Developers

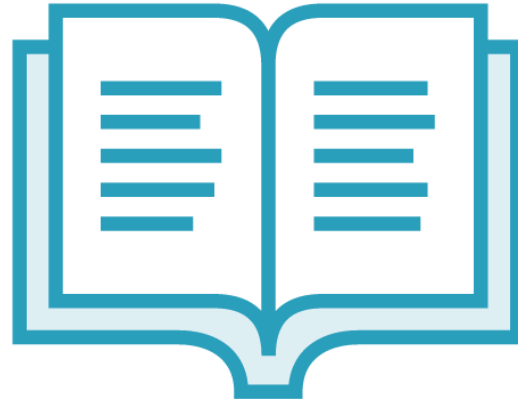


Course Structure



Syntax

Break down of regular expression syntax



Academic

Demonstration of academic syntax example

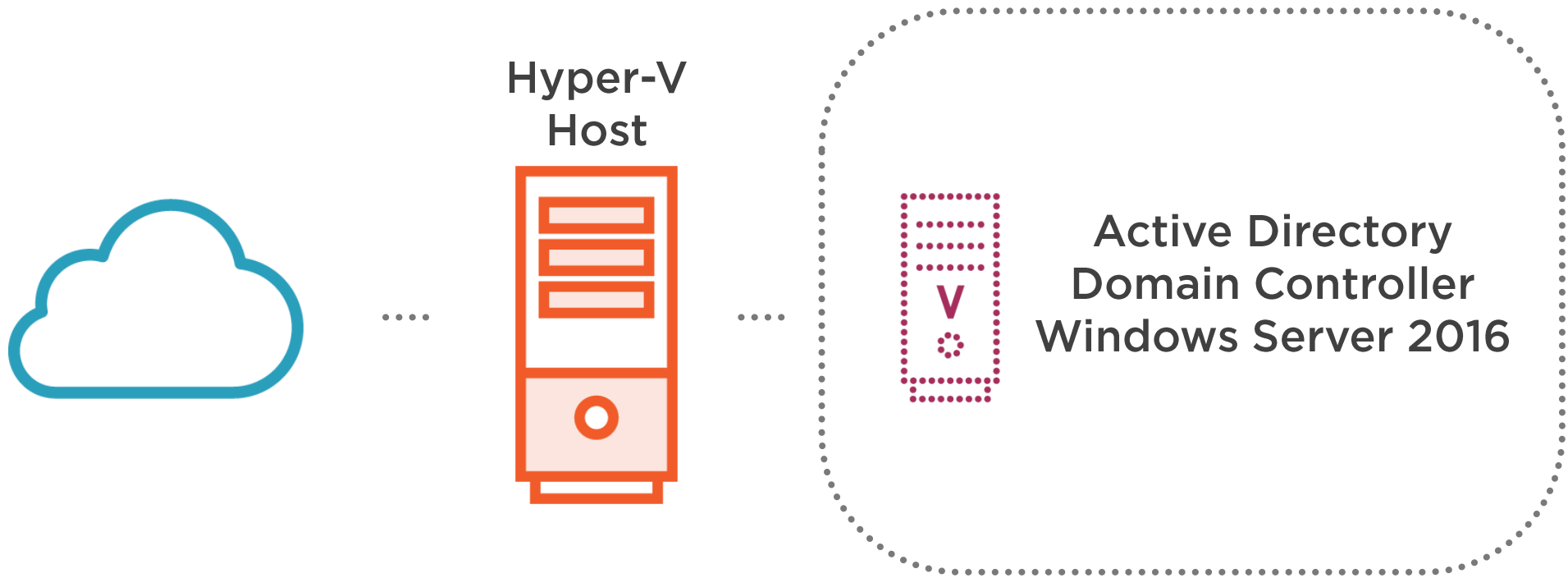


Real World

Practical example and demonstration of regular expression in action



Lab Setup



Regular Expression

Is a special sequence of characters that define a search pattern, mainly for use in pattern matching with strings.

-Wikipedia, The Free Encyclopedia



```
Get-ChildItem *.txt
```

```
gci *demos*
```

```
Get-Command *et-*net*
```

Wildcards

Wildcards can be used if supported by the cmdlet or command. They are used to represent any character any number of times.



Demo



Searching Files

- Search for file extensions
- Look for words in a file name

Finding Services

- Wildcards Parameters
- Where-object

Searching Active Directory

- Look for users
- Find a user's manager



```
' \\GFS-01\Share\regex.ps1' -match 'GFS-01'
```

```
' CN=jgreen,CN=Users,DC=globomantics,DC=com' -match 'jgreen'
```

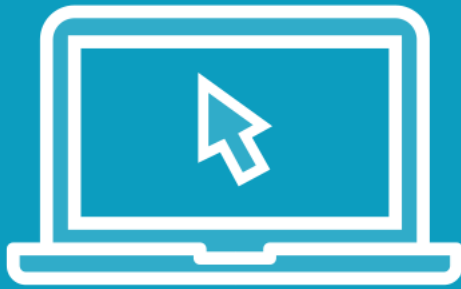
```
'globomantics,dc=com' -replace ',dc=', '.'
```

Literal Character Matches

Literal characters are used within regular expression to match a single character. Case sensitivity is determined by the regular expression engine. If the regular expression engine is case insensitive, a single character represents both the upper and lower case character.



Demo



Basics of Character Matching

- Matching a word within a string

-match Operator

- \$Matches variable

Filtering Active Directory Data

- Extract user name from distinguished name
- Extract domain name from distinguished name



Summary



Defined Regular Expression

Wildcards

Literal Characters

-match operator

-replace operator

