What are we going to see in this session?

Let’s do some admin activity with AWS

- Introduction to Conditionals
- Built-in-functions
- Count
- for_each expression

Introduction to conditionals

- Conditionals can be used in many cases like
  - ✓ Use simple math functions
  - ✓ Refer to other variables
  - ✓ Use conditional like (if-else)
  - ✓ Using conditional in count

- We were using conditionals in many cases so far like
  - ✓ Refer the variables from different file.
  - ✓ Provide the path information’s to fetch the modules
  - ✓ Output the attributes.
  - ✓ Using proper resource types.
  - ✓ Generating data source and passing to resource block

These are all some common inbuild conditionals done by terraform itself for you.

Built-In Functions

- There are some built-in functions which can be used in terraform resources.
- Some common built-in functions are.

  - ✓ file(filename) – used to read the content of file, which we have used in many cases before. [To read .pem file]
  - ✓ length – helps to determine the length of given list.
  - ✓ element(list, index) – Returns the single element from a list at the given index.
  - ✓ lookup – used to lookup 2 values provided.
  - ✓ index(list, value) – Used to find the element index for the given value.

You can just go through this link to know more Functions

Count

Let's see how to make use of count efficiently.

- Example 1 :
  - ✓ Let’s try creating list of users using count.
  - ✓ We may combine count with terraform built in functions such as [Length & Element] to achieve this.

Code : git-repo [file path : Terraform/Codes/E19_count_iam]

- Example 2 :
  - ✓ Let’s try creating list of EC2 instance with different machine names using count increment.

Code : git-repo [file path : Terraform/Codes/E20_count_ec2_increment]

- Example 3 :
  - ✓ Let’s try creating list of EC2 instance by defining the tags in vars file.

Code : git-repo [file path : Terraform/Codes/E21_count_ec2_element]

for_each expression

- For_each expression allows you to loop over lists, sets and maps to create either
- (a) multiple copies of an entire resource (b) multiple copies of inline block with a resource.
- Example : Create uses using for_each

  Code : git-repo [file path : Terraform/Codes/E19_count_iam]
  - ✓ Note the use of ”toset” is to convert the ”var.users” list into a set, because “for_each” only supports sets and maps when used on a resource.
  - ✓ When ”for_each” loops over this set, it will make each username available in ”each.value”
- (b) multiple copies of inline block with a resource.

  Code : git-repo [file path : Terraform/Codes/E19_count_iam]

End of this topic !

Any questions?

TERRAFORM