

SYSTEM PROVISIONING AND CONFIGURATION MANAGEMENT

LAB FILE

NAME: SMRITI RAI

SAP ID: 500096396

BATCH: B3

SUBMITTED TO: Dr. Hitesh Kumar Sharma

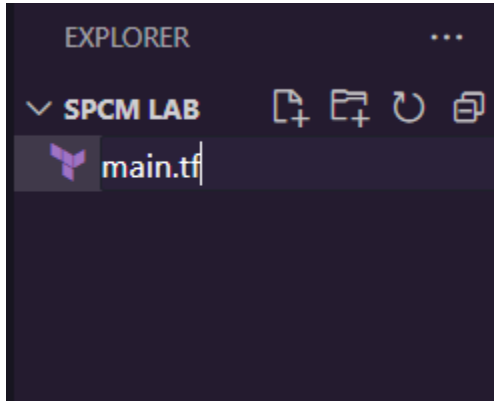
SEMESTER: VI

ENROLLMENT NO.: R2142211212

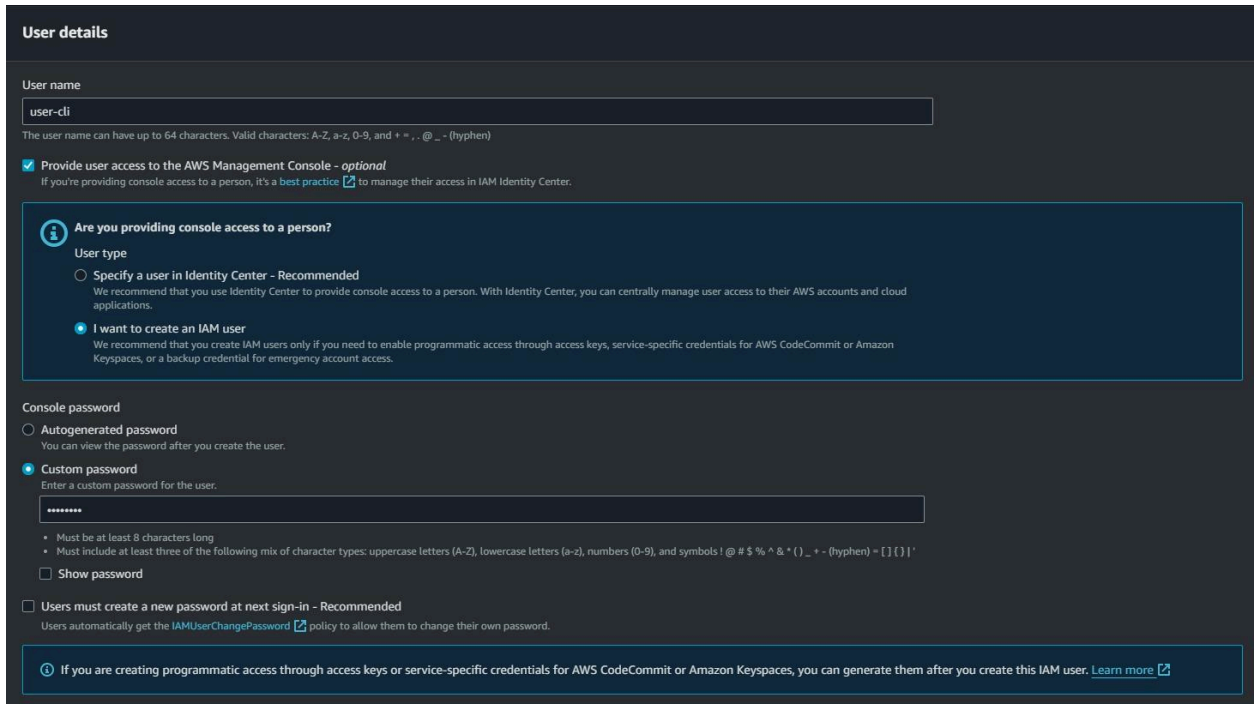
EXPERIMENT 2:

Terraform AWS Provider and IAM user Settings

1. Create a new folder for your terraform configuration.
2. Add a file named 'main.tf'.



3. Now make a new IAM account in your AWS console. Also set the custom password.

A screenshot of the AWS IAM console 'User details' page. The page has a dark theme. At the top, it says 'User details'. Below that, there's a section for 'User name' with a text input field containing 'user-cli'. A note below the field says 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , @ _ - (hyphen)'. Below this, there's a checkbox labeled 'Provide user access to the AWS Management Console - optional' which is checked. A note below it says 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' Below this is a section titled 'Are you providing console access to a person?' with an information icon. It has two radio button options: 'Specify a user in Identity Center - Recommended' and 'I want to create an IAM user'. The second option is selected. Below this is a section for 'Console password' with two radio button options: 'Autogenerated password' and 'Custom password'. The second option is selected. Below this is a text input field for the custom password, which is currently empty. Below the field, there are two bullet points: 'Must be at least 8 characters long' and 'Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } | ' '. Below the field is a checkbox labeled 'Show password' which is unchecked. Below this is a checkbox labeled 'Users must create a new password at next sign-in - Recommended' which is unchecked. A note below it says 'Users automatically get the IAMUserChangePassword policy to allow them to change their own password.' At the bottom, there's a blue box with an information icon and text: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more'.

4. Set appropriate permissions.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1171)

Choose one or more policies to attach to your new user.

Filter by Type

All types

	Policy name	Type	Attached entities
<input type="checkbox"/>	AccessAnalyzerServiceRolePolicy	AWS managed	0
<input checked="" type="checkbox"/>	AdministratorAccess	AWS managed - job function	1

- Select create Access Key and note down the access key and secret key.

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

☒ Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

☐ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

☐ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

☐ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

☐ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

☐ Other
Your use case is not listed here.

Alternatives recommended

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

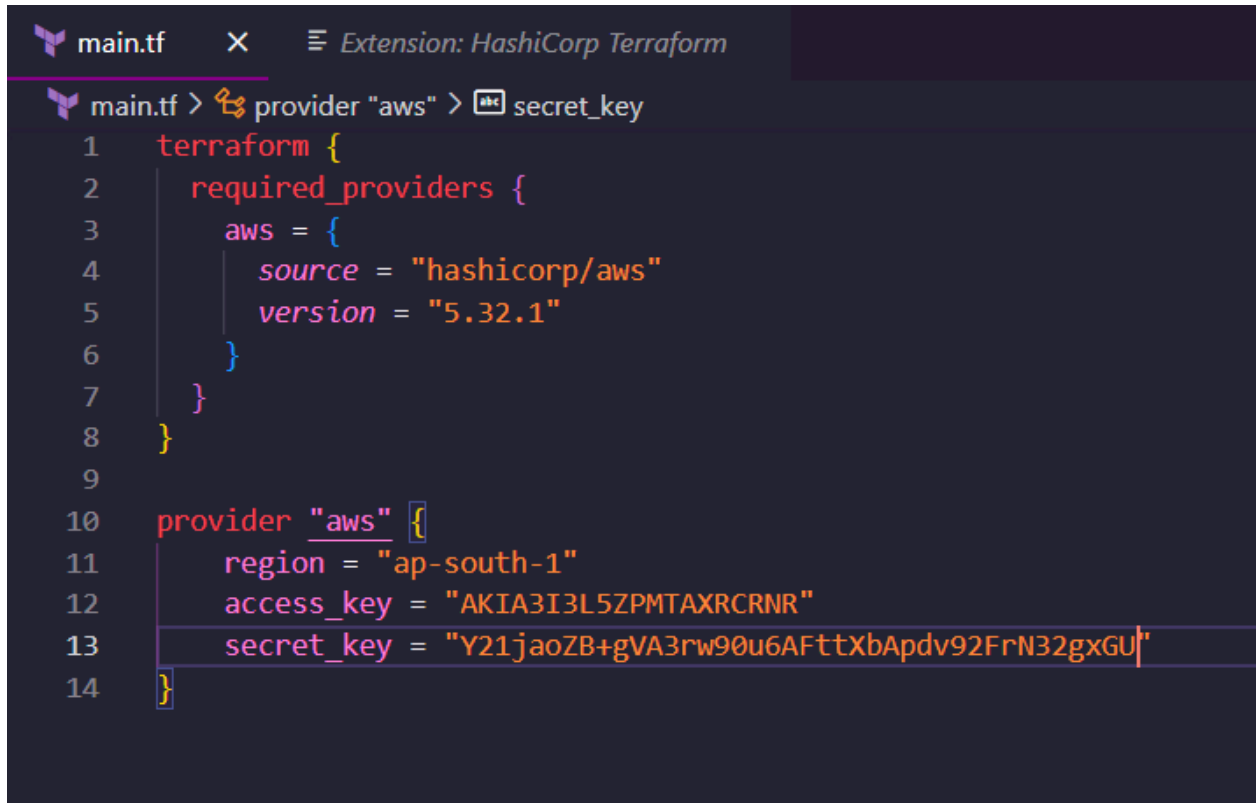
Confirmation

☒ I understand the above recommendation and want to proceed to create an access key.

Cancel

Next

6. Add the following content.



The screenshot shows a code editor with a dark theme. At the top, there's a tab labeled 'main.tf' and a status bar indicating the 'HashiCorp Terraform' extension is active. The editor content shows a Terraform configuration file with the following structure:

```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.32.1"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "ap-south-1"
12   access_key = "AKIA3I3L5ZPMTAXRCNR"
13   secret_key = "Y21jaoZB+gVA3rw90u6AFttXbApdv92FrN32gxGU"
14 }
```

7. Run 'terraform init' command to initialise the working directory.

```
C:\Windows\System32\cmd.e  X + v
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.

D:\docss\UPES\sem 6\SPCM Lab>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.32.1"...
- Installing hashicorp/aws v5.32.1...
- Installed hashicorp/aws v5.32.1 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

D:\docss\UPES\sem 6\SPCM Lab>x
```

