Page
Date / /

Q1- SSO (single sign-on) is an authentication method that enables users to securely authenticate with multiple applications & websites by using just one set of credentials. An SSO service does not necessarily ~~[illegible]~~ remember who a user is, since it does not store user identities. Most SSO services work by checking user credentials against a separate identity management service.

Adv of SSO:-

(1) Stronger password
(2) No repeated password
(3) Better password policy enforcement
(4) Multi-factor authentication
(5) Single point for enforcing password re-entry.
(6) Less time wasted on password recovery.

⇒ steps of implementing of SSO:-

① Navigate to the AWS SSO Console & choose AWS acc. from the navigation pane. Choose any acc. Then choose Assign users

Page
Date / /

② Choose Users, Start typing to search for users & then choose Search connected directory.

③ To select ~~pers~~ permission sets, you first have to create one. Choose create new ~~per~~ permission set.

④ You can use an existing job function policy to create a ~~per~~ permission set. This type of policy allows you to apply predefined AWS managed policies to a permission set that are based on common job function in the IT industry. Alternatively, you can create a ~~cu~~ custom permission ~~set~~ set based on custom policies.

⑤ Choose the policy & then choose create. As a result this permission set will be available to pick on the next screen.

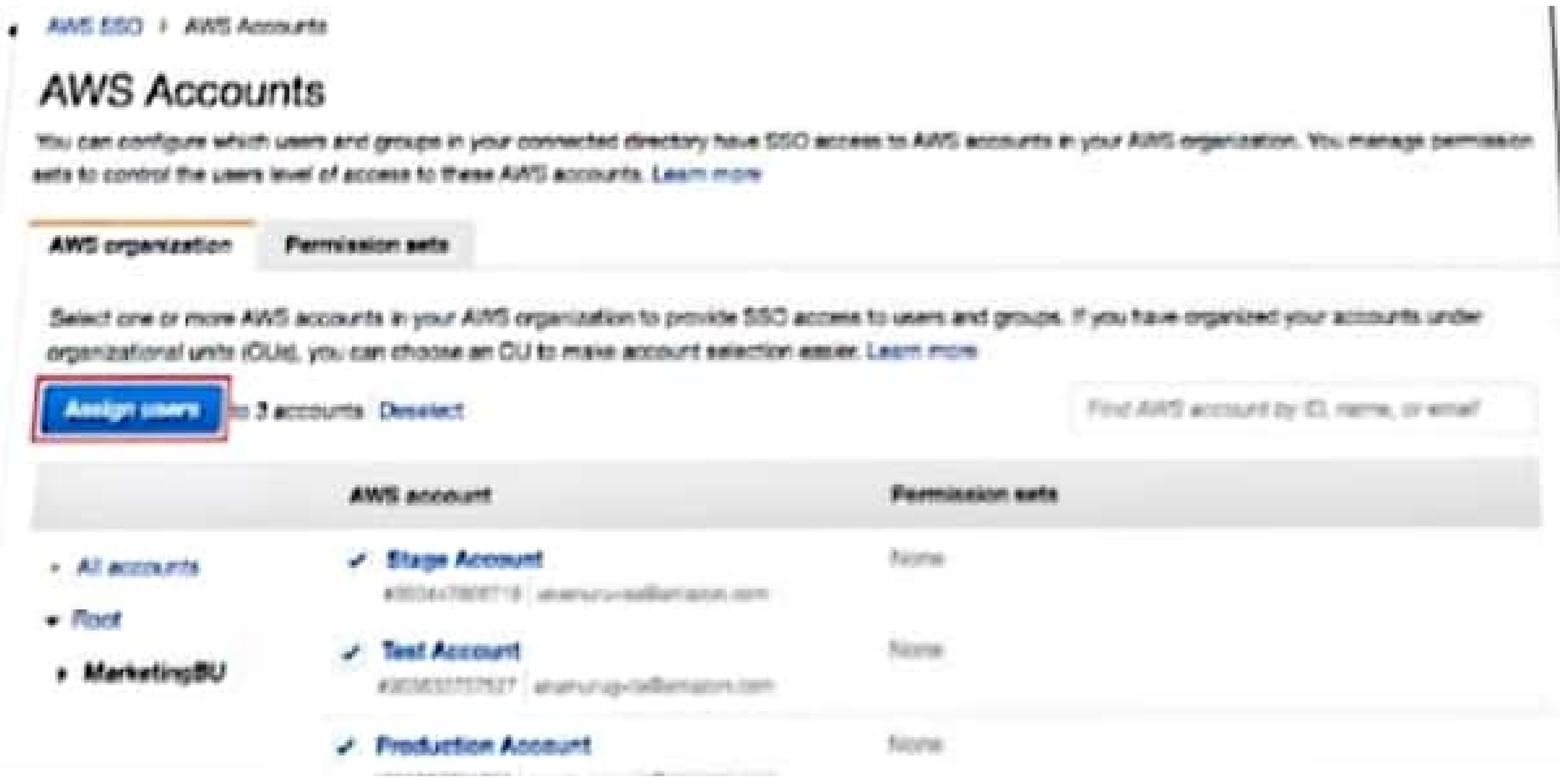
⑥ Choose the ~~pe~~ permission set to indicate ~~to~~ what level of access you want to grant ~~to~~ your users.

⑦ User can sign in to the user portal & access the acc. to which you give them access. AWS SSO automatically sets up the ~~ne~~ necessary trust b/w acc. to enable SSO.

Page
Date / /

(8) The user can choose an acc. & a permission set to sign in to that acc. without needing to provide a password again.

~~Q~~ ~~(1) Create new repository on Github~~



AWS SSO › AWS Accounts

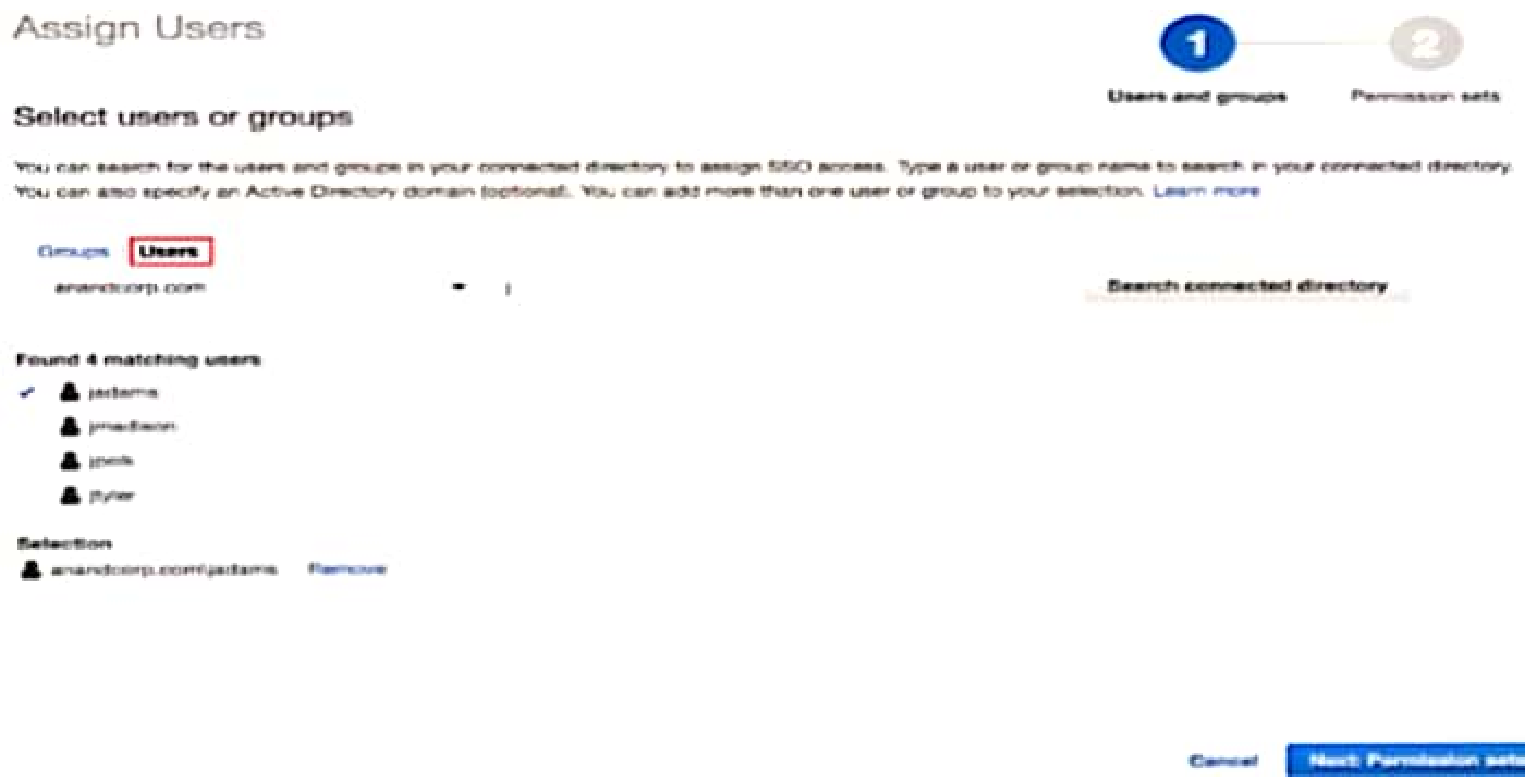
AWS Accounts

You can configure which users and groups in your connected directory have SSO access to AWS accounts in your AWS organization. You manage permission sets to control the users level of access to these AWS accounts. Learn more

Select one or more AWS accounts in your AWS organization to provide SSO access to users and groups. If you have organized your accounts under organizational units (OUs), you can choose an OU to make account selection easier. Learn more

to 3 accounts Deselect

AWS account	Permission sets
✓ Stage Account [illegible] | [illegible]	None
✓ Test Account [illegible] | [illegible]	None
✓ Production Account	None



Assign Users

Select users or groups

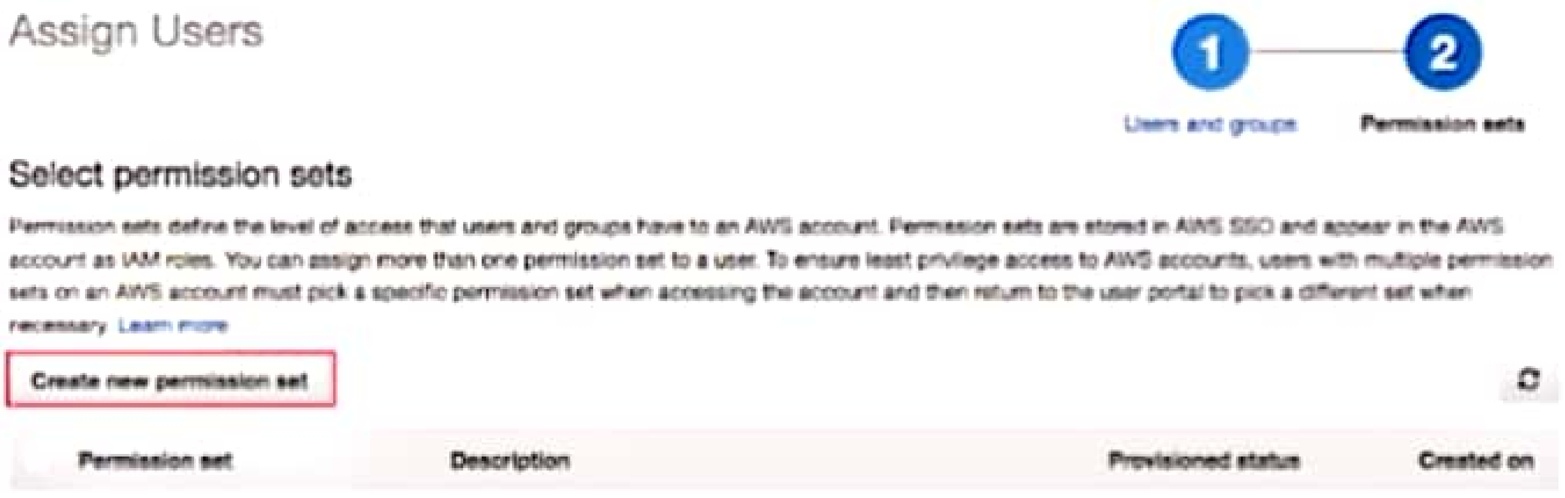
You can search for the users and groups in your connected directory to assign SSO access. Type a user or group name to search in your connected directory. You can also specify an Active Directory domain (optional). You can add more than one user or group to your selection. Learn more

Found 4 matching users

- ✓ jadams
- jmadison
- jpolk
- jtyler

Selection

anandcorp.com\jadams Remove

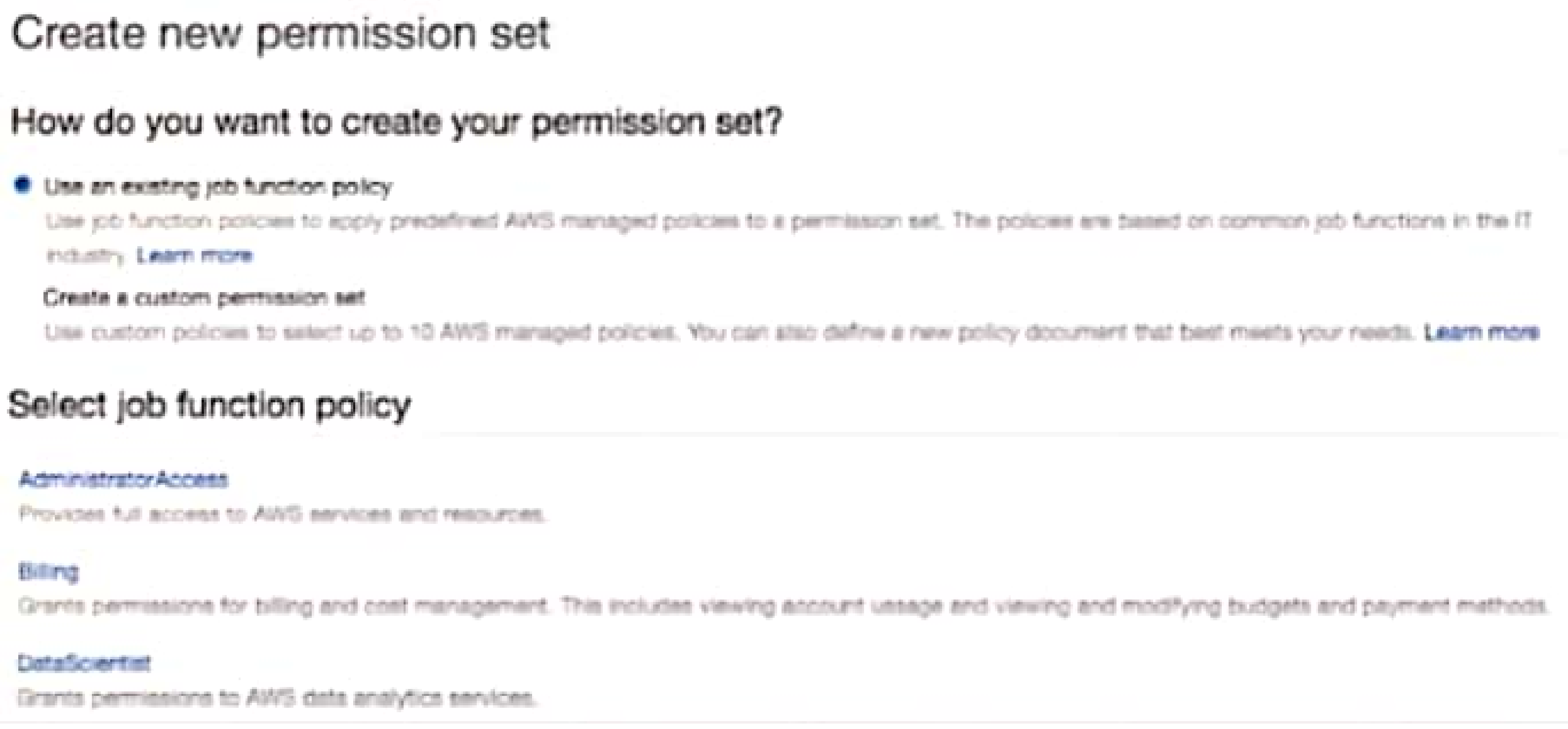


Assign Users

Select permission sets

Permission sets define the level of access that users and groups have to an AWS account. Permission sets are stored in AWS SSO and appear in the AWS account as IAM roles. You can assign more than one permission set to a user. To ensure least privilege access to AWS accounts, users with multiple permission sets on an AWS account must pick a specific permission set when accessing the account and then return to the user portal to pick a different set when necessary. Learn more

Permission set	Description	Provisioned status	Created on



Create new permission set

How do you want to create your permission set?

● Use an existing job function policy
Use job function policies to apply predefined AWS managed policies to a permission set. The policies are based on common job functions in the IT industry. Learn more

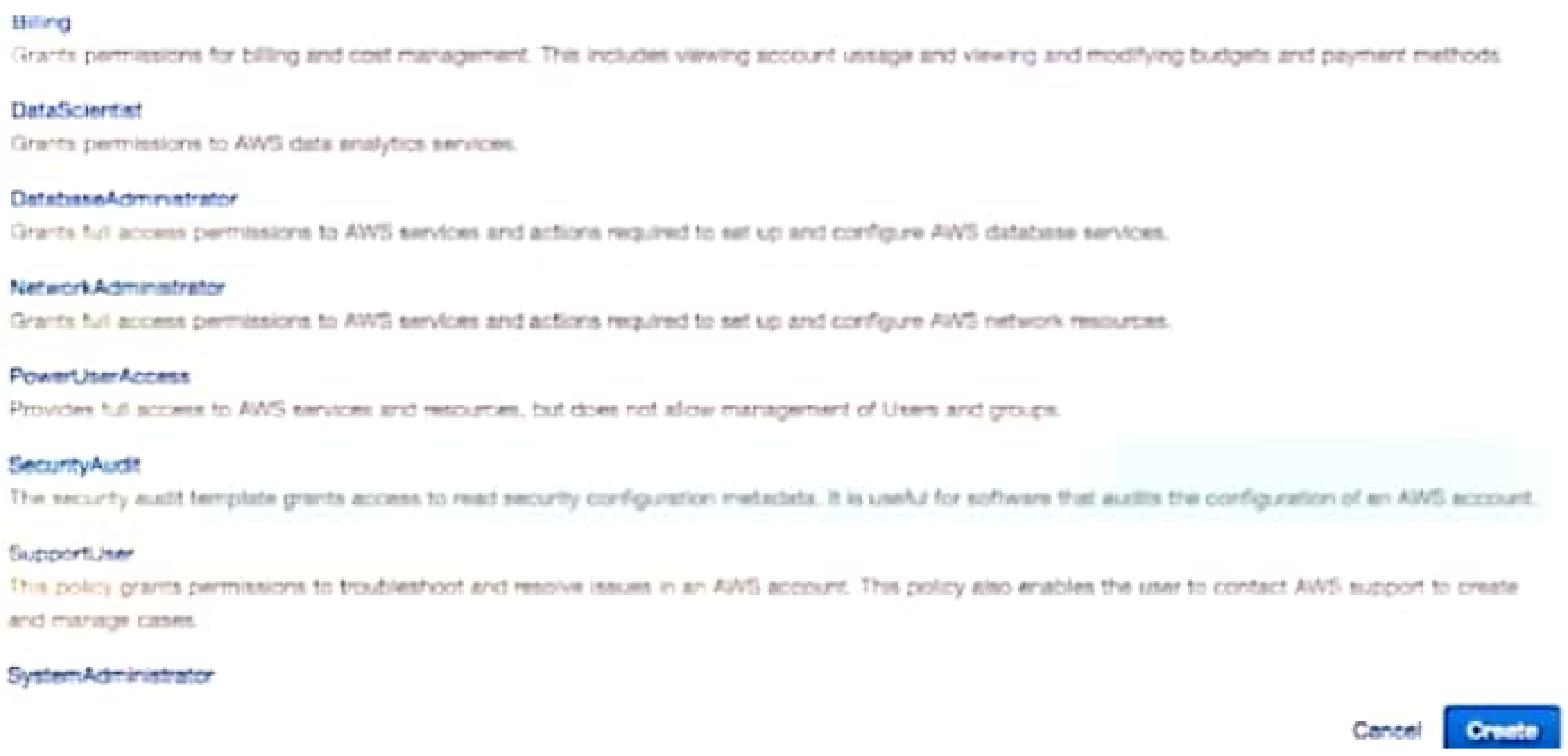
Create a custom permission set
Use custom policies to select up to 10 AWS managed policies. You can also define a new policy document that best meets your needs. Learn more

Select job function policy

AdministratorAccess
Provides full access to AWS services and resources.

Billing
Grants permissions for billing and cost management. This includes viewing account usage and viewing and modifying budgets and payment methods.

DataScientist
Grants permissions to AWS data analytics services.



Billing

Grants permissions for billing and cost management. This includes viewing account usage and viewing and modifying budgets and payment methods.

DataScientist

Grants permissions to AWS data analytics services.

DatabaseAdministrator

Grants full access permissions to AWS services and actions required to set up and configure AWS database services.

NetworkAdministrator

Grants full access permissions to AWS services and actions required to set up and configure AWS network resources.

PowerUserAccess

Provides full access to AWS services and resources, but does not allow management of Users and groups.

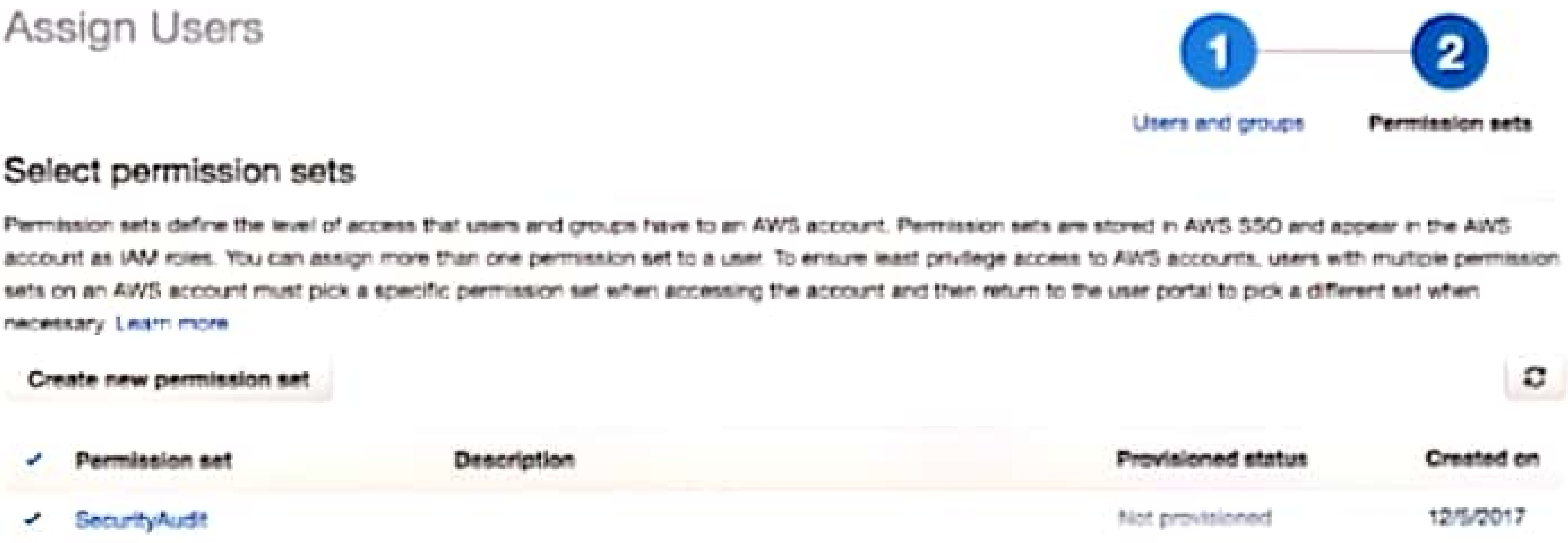
SecurityAudit

The security audit template grants access to read security configuration metadata. It is useful for software that audits the configuration of an AWS account.

SupportUser

This policy grants permissions to troubleshoot and resolve issues in an AWS account. This policy also enables the user to contact AWS support to create and manage cases.

SystemAdministrator



Assign Users

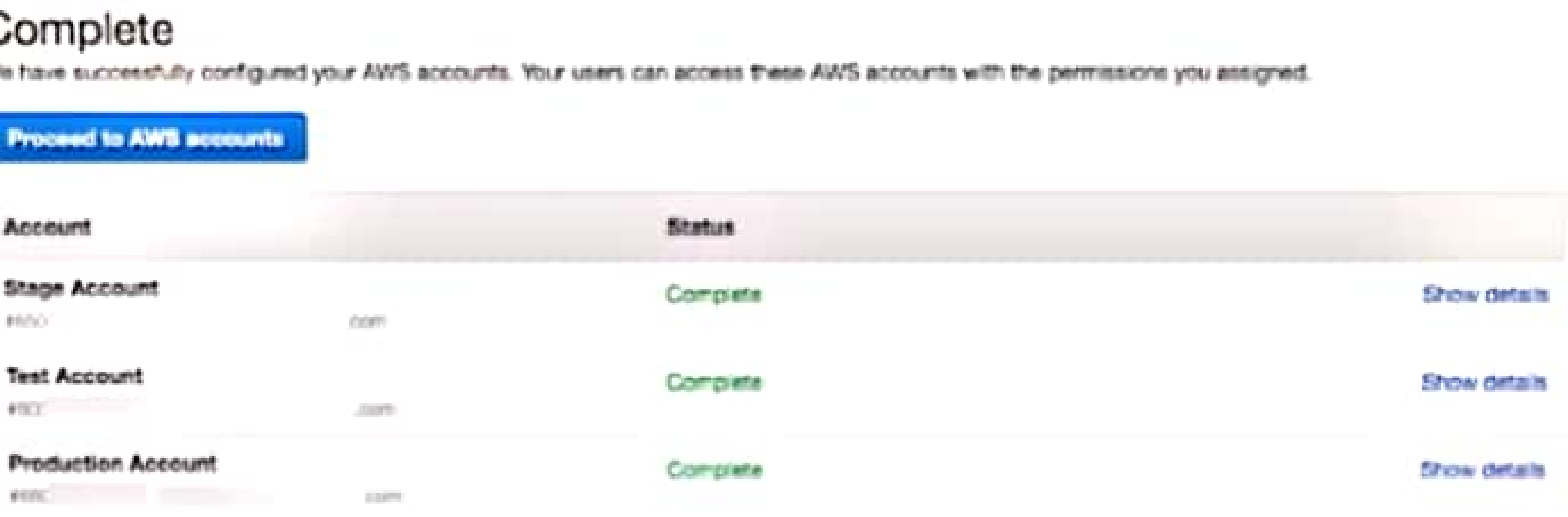
1 Users and groups — 2 Permission sets

Select permission sets

Permission sets define the level of access that users and groups have to an AWS account. Permission sets are stored in AWS SSO and appear in the AWS account as IAM roles. You can assign more than one permission set to a user. To ensure least privilege access to AWS accounts, users with multiple permission sets on an AWS account must pick a specific permission set when accessing the account and then return to the user portal to pick a different set when necessary. Learn more

Create new permission set

✓	Permission set	Description	Provisioned status	Created on
✓	SecurityAudit		Not provisioned	12/5/2017

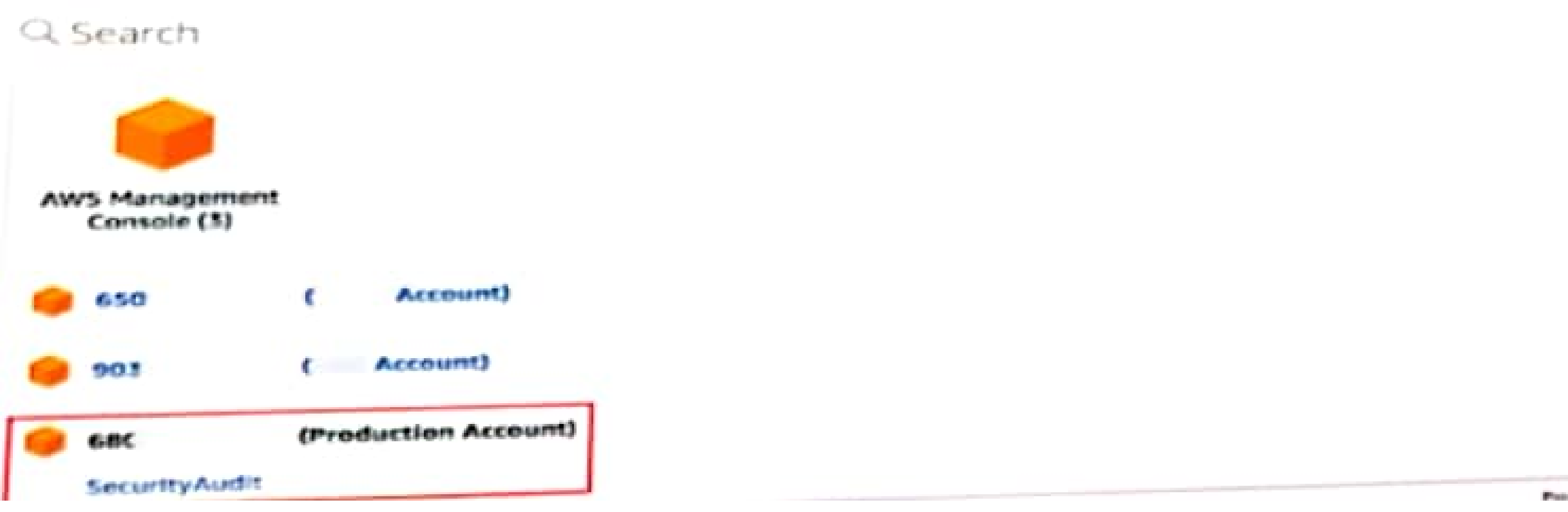


Complete

e have successfully configured your AWS accounts. Your users can access these AWS accounts with the permissions you assigned.

Proceed to AWS accounts

Account	Status	
Stage Account #ISO com	Complete	Show details
Test Account #SC .com	Complete	Show details
Production Account #ISSC .com	Complete	Show details



Search

AWS Management Console (3)

650 (Account)

903 (Account)

68C (Production Account)

SecurityAudit

Page
Date / /

⑧ Pull repository from github
Command → git pull origin master.

<u>All the commands</u> –

Cd / your directory
① git init
② git add . or git add ('filename')
③ git commit -m "my first file"
④ git remote origin https:// github.com/your name / your - repo - name.git
⑤ git pull origin master
⑥ git push origin master

Scanned with CamScanner

Page
Date / /

Q2 — ① Create new repository on github.
② Now open cmd use cd command to the local files directory that you publish on github.
③ Initialise local directory
Command → git init

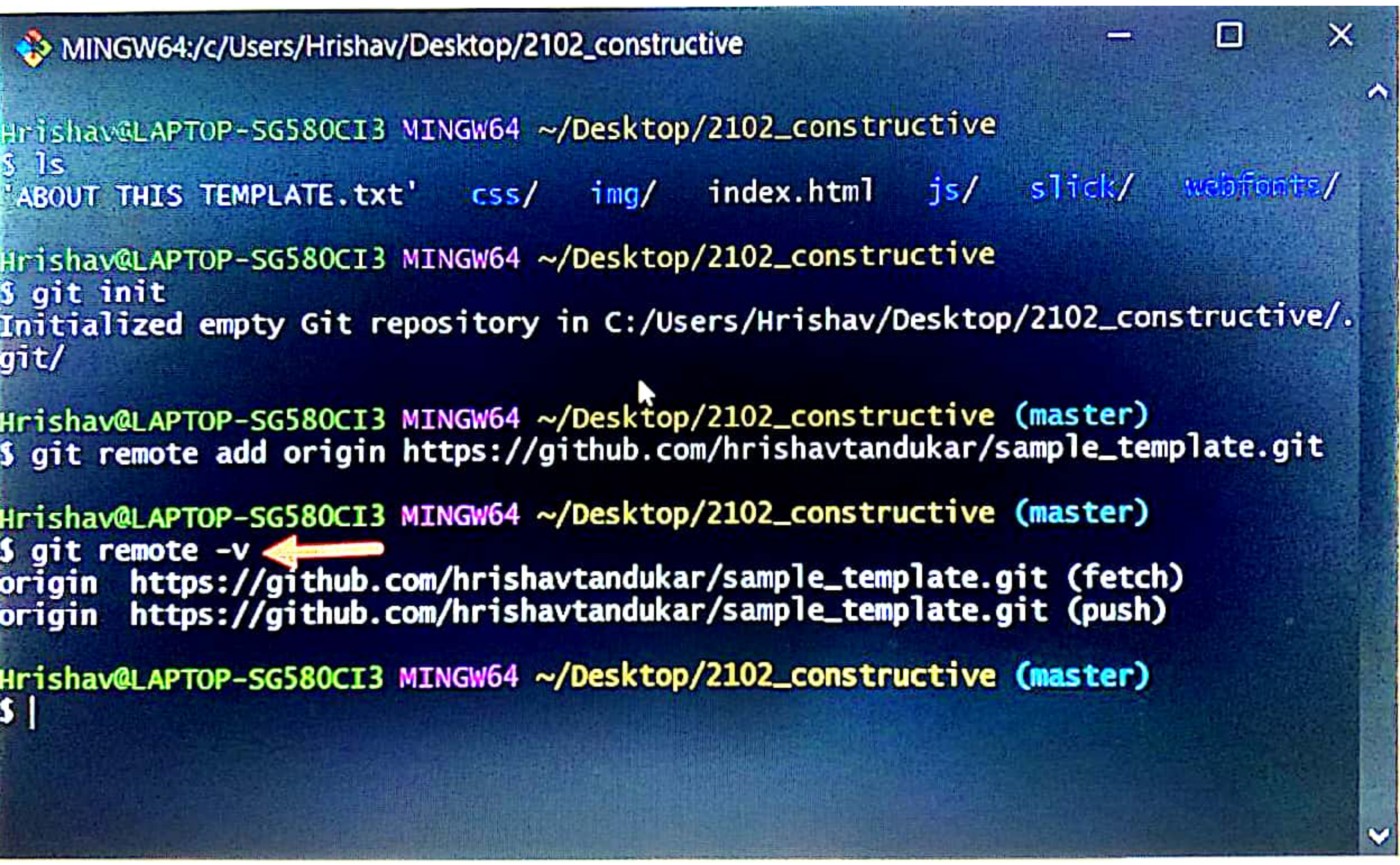
④ Add local repository
Command → git add, This command stage all to files in the directory.
⑤ Commit repository -
Command → git commit -m "first commit msg"

⑥ Now copy the remote repository URL provides by github to you when you published your repository on github.

Command → git remote add origin https://github.com/yourname/yourrepoName.git

⑦ push local repository to github

Command → git push origin master



MINGW64:/c/Users/Hrishav/Desktop/2102_constructive

```
Hrishav@LAPTOP-SG580CI3 MINGW64 ~/Desktop/2102_constructive
$ ls
'ABOUT THIS TEMPLATE.txt'   css/   img/   index.html   js/   slick/   webfonts/

Hrishav@LAPTOP-SG580CI3 MINGW64 ~/Desktop/2102_constructive
$ git init
Initialized empty Git repository in C:/Users/Hrishav/Desktop/2102_constructive/.
git/

Hrishav@LAPTOP-SG580CI3 MINGW64 ~/Desktop/2102_constructive (master)
$ git remote add origin https://github.com/hrishavtandukar/sample_template.git

Hrishav@LAPTOP-SG580CI3 MINGW64 ~/Desktop/2102_constructive (master)
$ git remote -v
origin  https://github.com/hrishavtandukar/sample_template.git (fetch)
origin  https://github.com/hrishavtandukar/sample_template.git (push)

Hrishav@LAPTOP-SG580CI3 MINGW64 ~/Desktop/2102_constructive (master)
$ |
```