

2110415 Software-Defined Systems Term Project

“Infrastructure-as-Code”

You are going to deploy a personal cloud storage software called NextCloud on Amazon web services using an infrastructure-as-code software called Terraform.

1. Find out what NextCloud is to get some ideas on what you will be doing.
2. Download and setup Terraform on your computer (<https://www.terraform.io/>)
3. Create a terraform script that will create and deploy
 - a. A new EC2 for installing and running the NextCloud app instance from source code (<https://nextcloud.com/install/#instructions-server>)
 - b. A new instance on EC2, install and run MariaDB database
 - c. A new S3 bucket and configure NextCloud to use it as a **primary** storage for NextCloud
(https://docs.nextcloud.com/server/latest/admin_manual/configuration_files/primary_storage.html#simple-storage-service-s3)
 - d. Various other services/configurations (IAM User, VPC, Internet Gateway, security groups, etc.) necessary to make this a fully functioning NextCloud service on AWS

I suggest you read these 3 documentations on how to set up NextCloud.

1. https://docs.nextcloud.com/server/latest/admin_manual/installation/example_ubuntu.html
2. https://docs.nextcloud.com/server/latest/admin_manual/installation/source_installation.html#apache-configuration-label
3. https://docs.nextcloud.com/server/latest/admin_manual/configuration_server/occ_command.html#config-commands

Note : The goal of the project is to be able to automate deployment from scratch (source code), i.e., continuous deployment (CD). **You are not allowed to deploy NextCloud using any other means.** So, no docker image, NextCloud appliance, or snap package manager. Doing so would result in getting a score of 0 for this project. However, feel free to study those provisioning scripts for ideas. Additionally, you may want to set up NextCloud on an EC2 instance or in your virtualbox manually to see what you need to configure after installation.

Additional requirements:

1. Your script shall accept the terraform.tfvars file and configure the region, availability_zone, bucket_name, etc. accordingly. (See : <https://gist.github.com/theminer3746/150c52809e0235e99265ee1e252912cf>)
2. For this project, we will use the **Ubuntu Server 20.04 LTS (HVM)** AMI. Because different regions use different AMI ID's for the same image, a single AMI ID can not be used across regions. You will need to specify the AMI ID in your configuration file (see above).
3. You should take security into consideration when creating security groups. Think of all the scenarios thoroughly. For example, the database instance should not be accessible by anyone on the Internet, and should only be accessible by the app instance's ip/subnet only. This translates to only allow incoming tcp connections to port 3306 (MariaDB's port) on the database instance from the app instance's ip/subnet only.

- Note that your AWS keys should never be included in the Terraform script. Anyone who has set up the proper ~/.aws/credentials should be able to run your script.

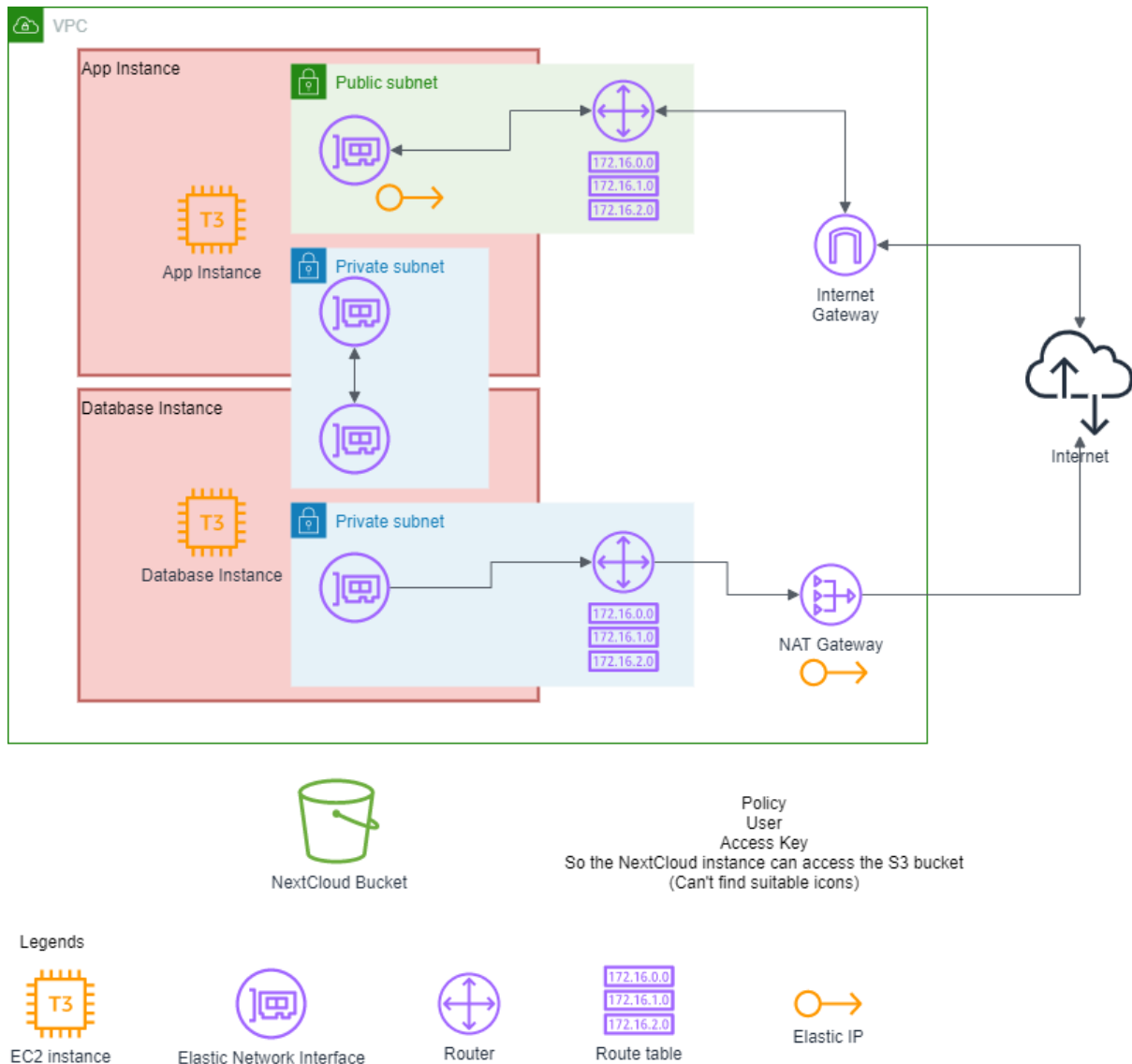


Figure 1: Your infrastructure should look like this after running terraform apply. Any unfamiliar services (i.e., Elastic Network Interface, Elastic IP, VPC) will require additional self-study.

Diagram explanation:

- An EC2 instance (App) running NextCloud with 2 elastic network interfaces.
 - Interface 1 has an elastic IP attached and is in a publicly accessible subnet associated with a routing table and an Internet gateway. Users will connect to NextCloud using the elastic IP on this interface.
 - Interface 2 is in a private subnet used only to communicate with the database instance.
- An EC2 instance (Database) running MariaDB, also with 2 elastic network interfaces.
 - Interface 1 is in a private subnet with a route table associated with a NAT gateway which is associated with an elastic IP, so that your database instance can access the Internet to download MariaDB.

Note : As a good security practice, people must not be able to connect to the database instance from the Internet directly. Therefore, we use a NAT gateway.

- b. Interface 2 is in a private subnet used only to communicate with the app instance.

Terraform resources that you may need to use (in no particular order) :

1. aws_vpc
2. aws_subnet
3. aws_network_interface
4. aws_security_group

Note : Declaring ingress and egress rules in this resource is buggy (see <https://github.com/hashicorp/terraform-provider-aws/issues/20484>). So only use this resource to declare the security group and use the aws_security_group_rule resource to add the rules.

5. aws_security_group_rule
6. aws_instance

Note : You may use the user_data property to pass provision script to the instance (see

https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance#user_data)

7. aws_internet_gateway
8. aws_nat_gateway
9. aws_route
10. aws_route_table

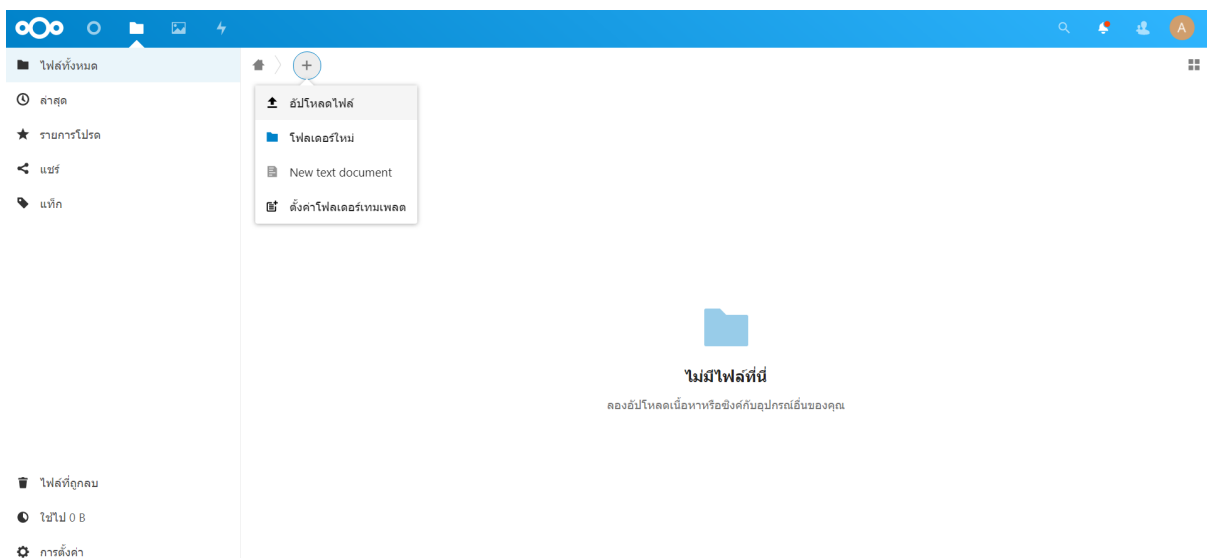
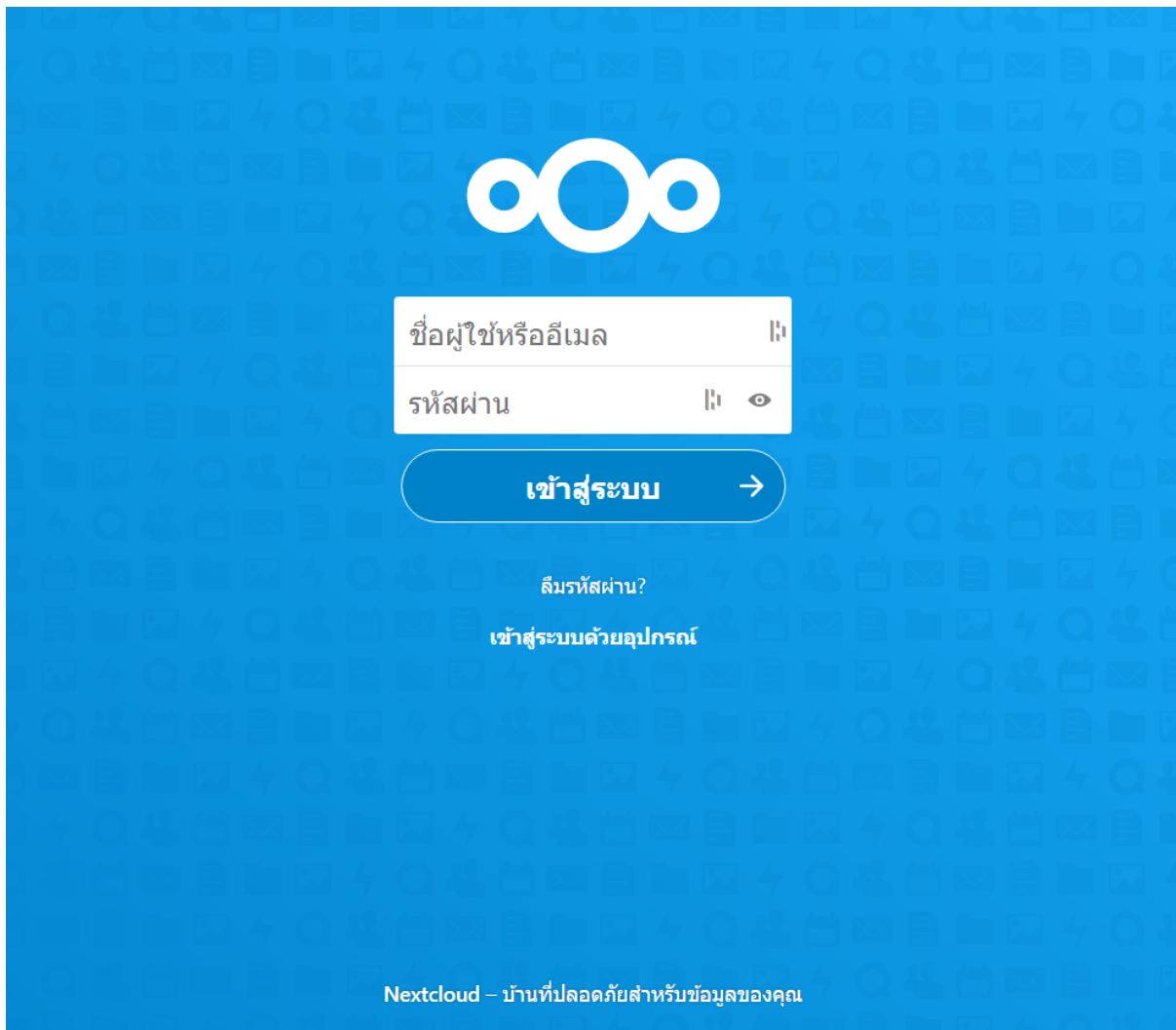
Note : Resource aws_route_table is buggy, just like aws_security_group (see <https://github.com/hashicorp/terraform-provider-aws/issues/20756>)

11. aws_route_table_association
12. aws_eip
13. aws_s3_bucket
14. aws_iam_access_key
15. aws_iam_user
16. aws_iam_policy

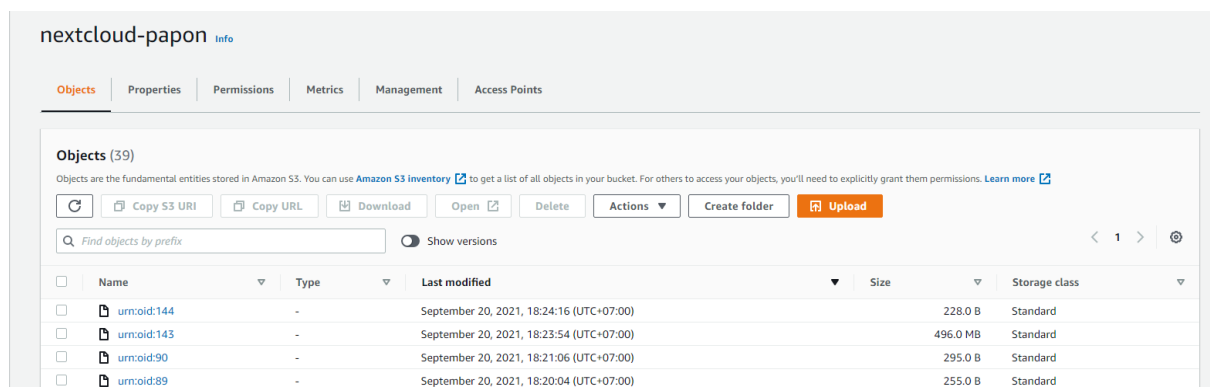
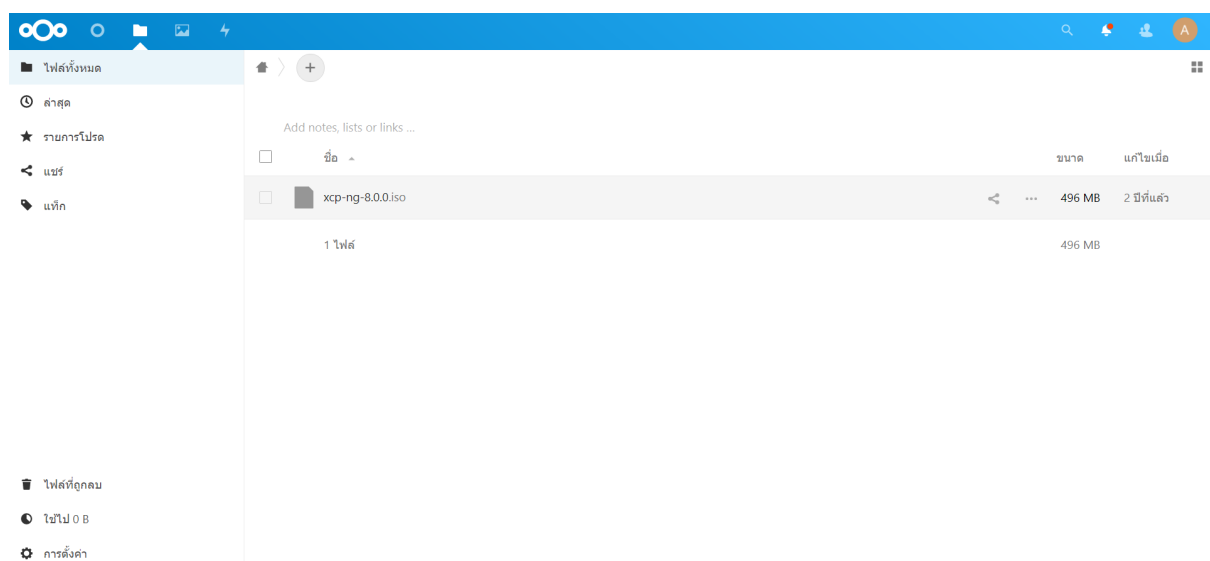
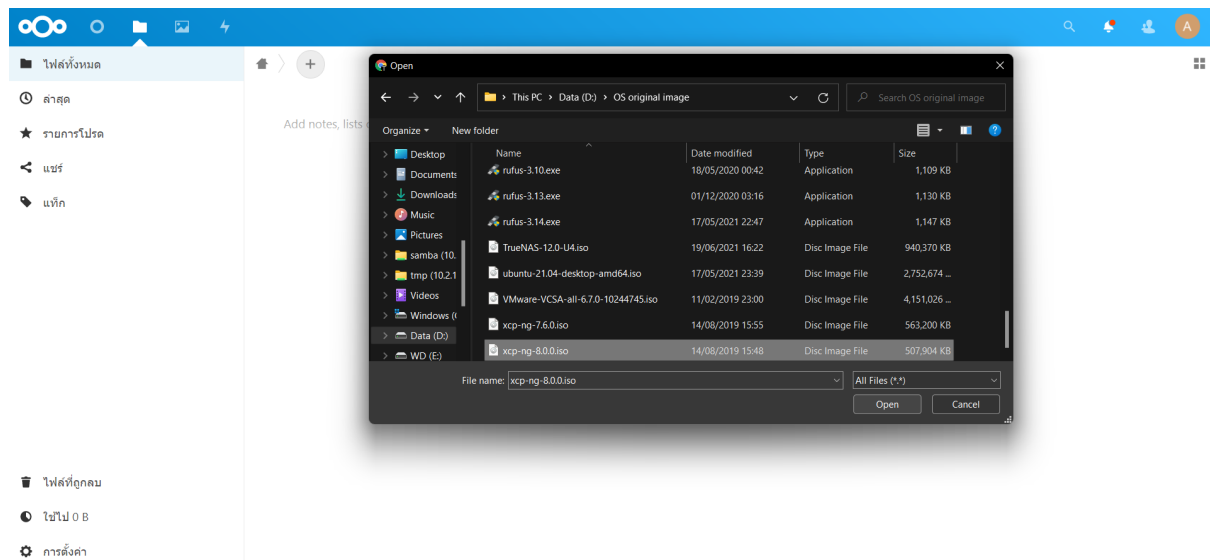
This is not a definitive list. You can use resources not in this list if you want.

Version Oct. 25, 2021

After running **terraform apply**, you should see the Nextcloud login page.



Version Oct. 25, 2021



After uploading your file to NextCloud, you should see your file(s) in the S3 bucket. If your file is larger than 5GB, NextCloud may break it up into multiple smaller files.

What to submit by the deadline November 21, 2021 @23:00:

1. Your terraform script.
2. A link to your youtube video (< 5 mins long) that
 - a. Demo's your terraform apply
 - b. Show the instances and S3 bucket it provisions
 - c. Demo successful install of NextCloud and upload a file through your NextCloud
 - d. Show the objects in your S3 bucket after file upload

What to do on demo day November 22, 2021 @9:00 (in class):

Instructors will assign you a region and you will change your terraform.tfvars file to configure your script to deploy resources in that region in your AWS account. You will run your script with the config file. After the script is done, you will demo the same content as your recorded video, and answer some questions about what you have done. Note that you should test deploying your script to other regions to confirm you understand the process and changes required prior to demo day.