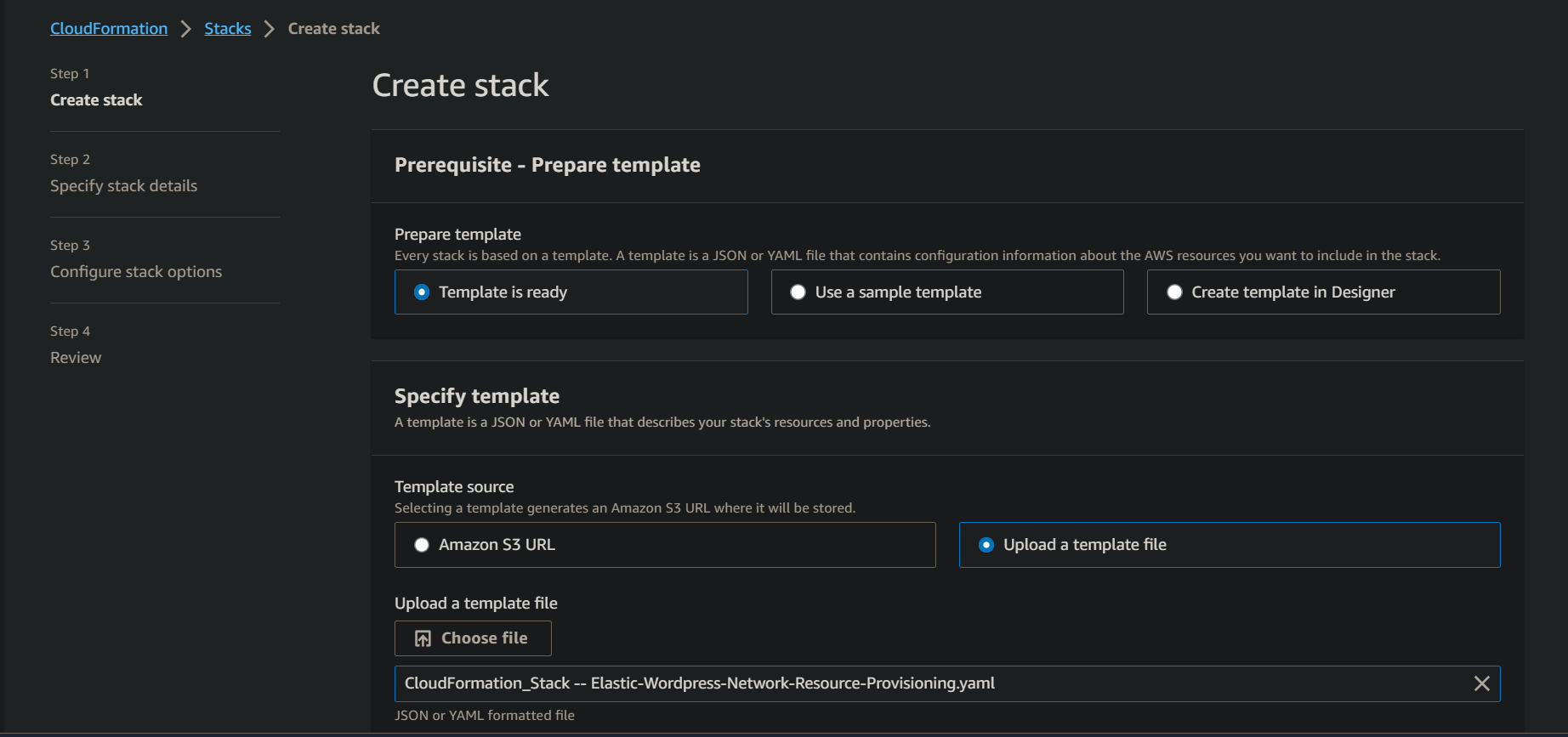
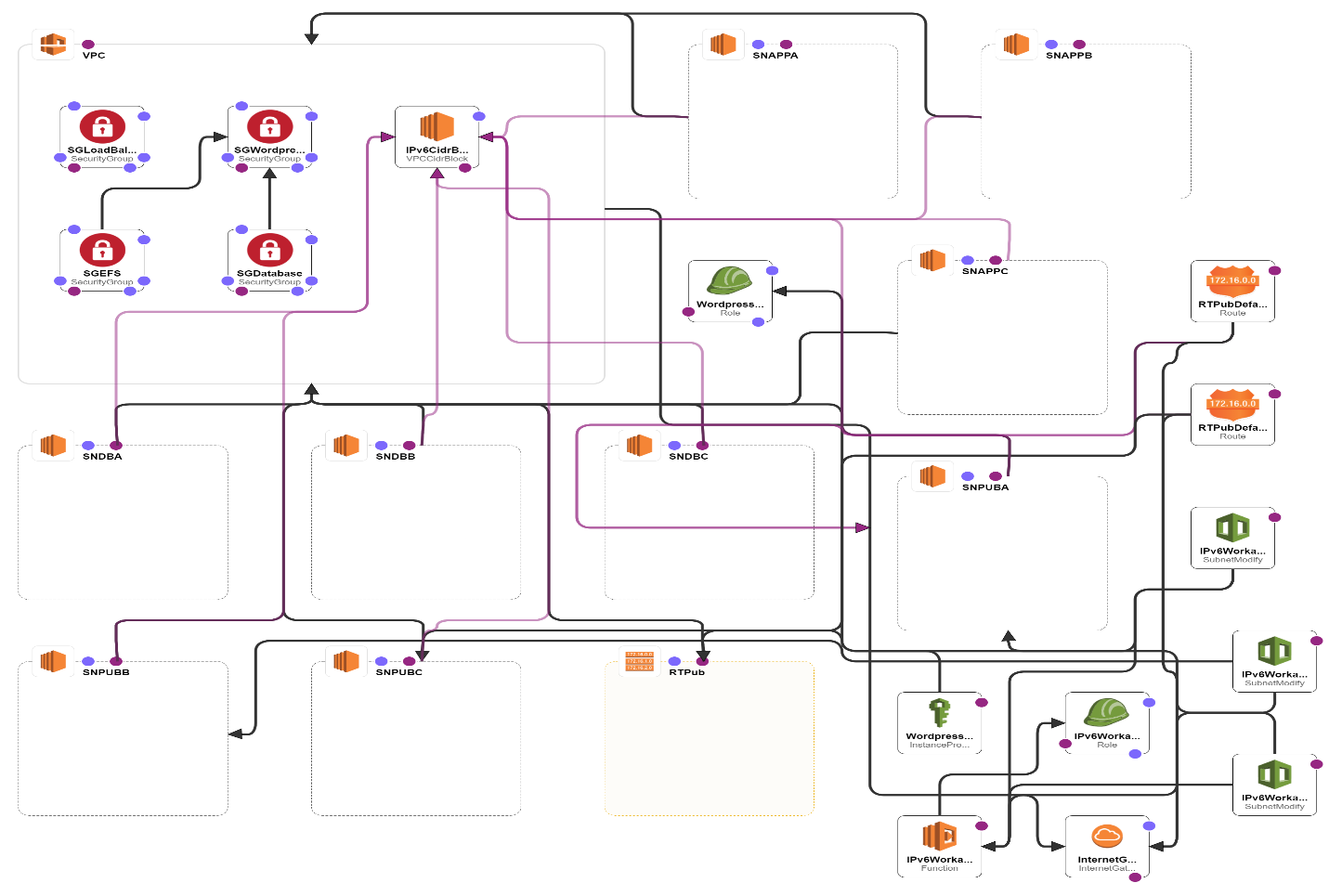
**Stage 1 - Setup the environment and manually build Wordpress**

In Stage 1 we will start with a single EC2 instance running the WordPress blogging engine and this single instance will be running the application itself, the database, and it will be storing the content for all of the blog posts. And for this example we're going to assume it's an animal pictures blog. Now crucially in this first stage you're going to build this server manually to experience all of the different components that need to operate to produce this web application.

First we have to create all the necessary networking resources for the 3-tier Web architecture Application. Instead of manually creating all the resources, we will use CloudFormation stack to automate this process. Click next, keep the defaults options & create the stack. (Refer YAML code – “CloudFormation\_Stack -- Elastic-Wordpress-Network-Resource-Provisioning.yaml” in code & config files folder).



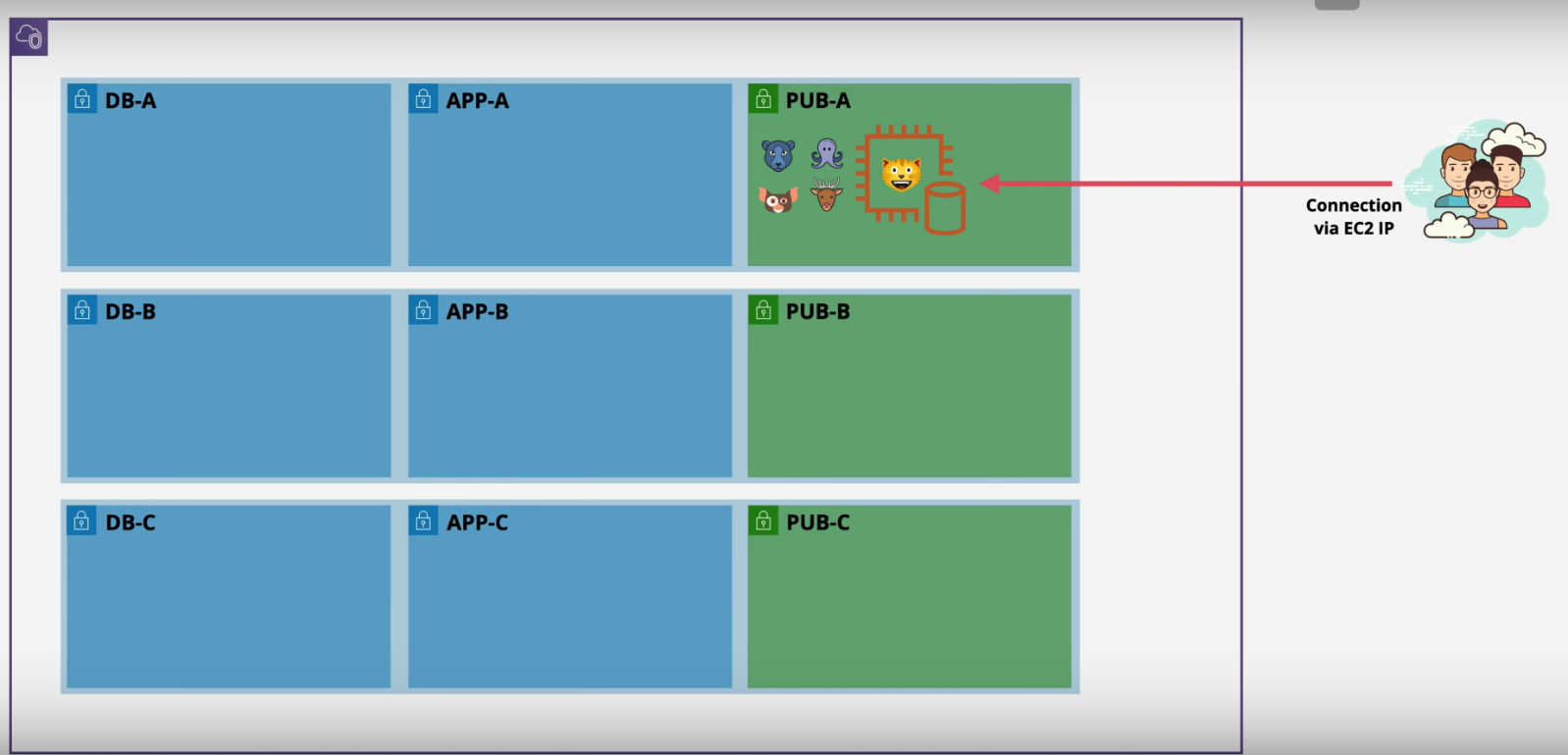
Following is the diagram of all the resources generated by this stack.



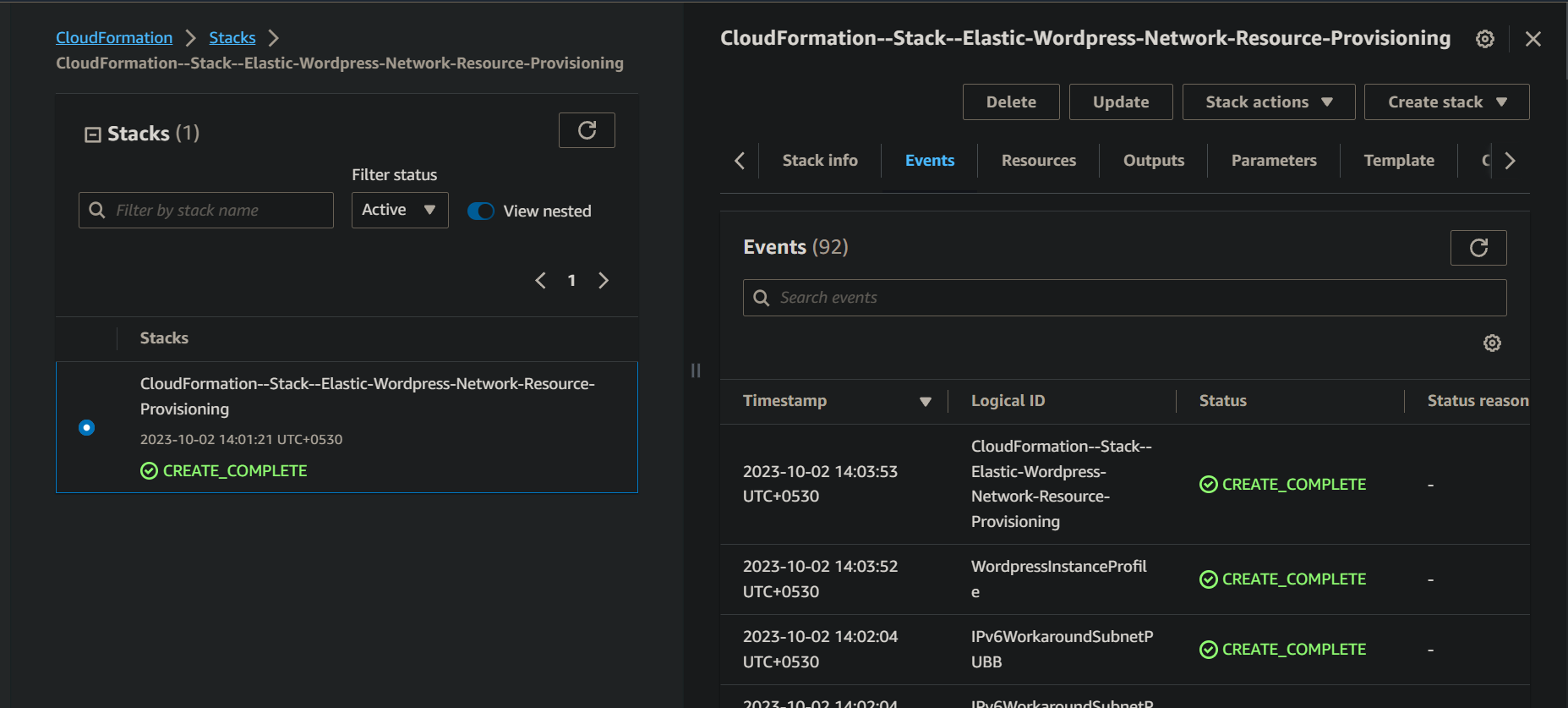
List of Resources created:

1. 1 VPC
2. 6 Private Subnets & 3 Public Subnets
3. 4 Security Groups (For – wordpress, loadbalancer, database, EFS)
4. 2 Routes (for Ipv4 & Ipv6)
5. 1 public RouteTable
6. 2 IAM Roles (For – Wordpress & Ipv6 workaround)
7. 1 Lambda function for Ipv6 workaround connected to Ipv6 workaround IAM Role.
8. 1 IAM instance profile for wordrpress.
9. 1 Internet Gateway.

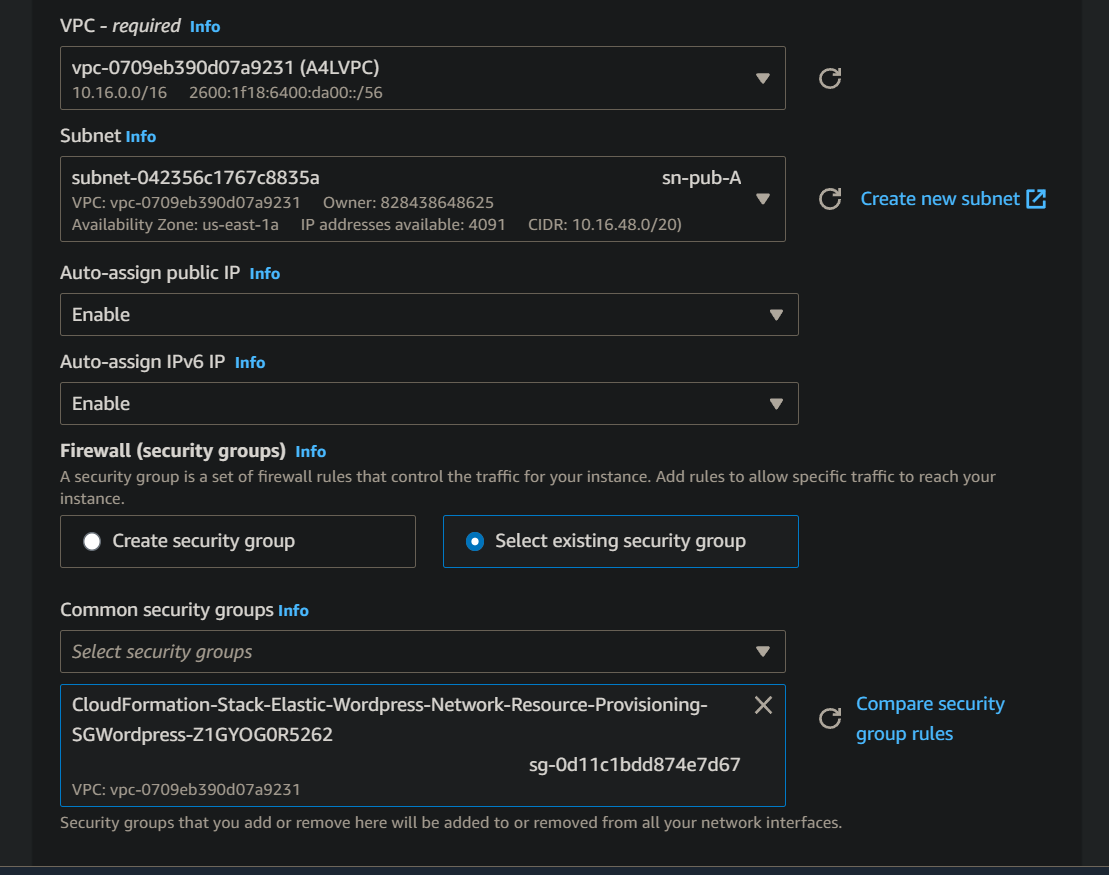
Following is the simplified Architecture diagram of the resources meant to be created:



CloudFormation stack creation successful

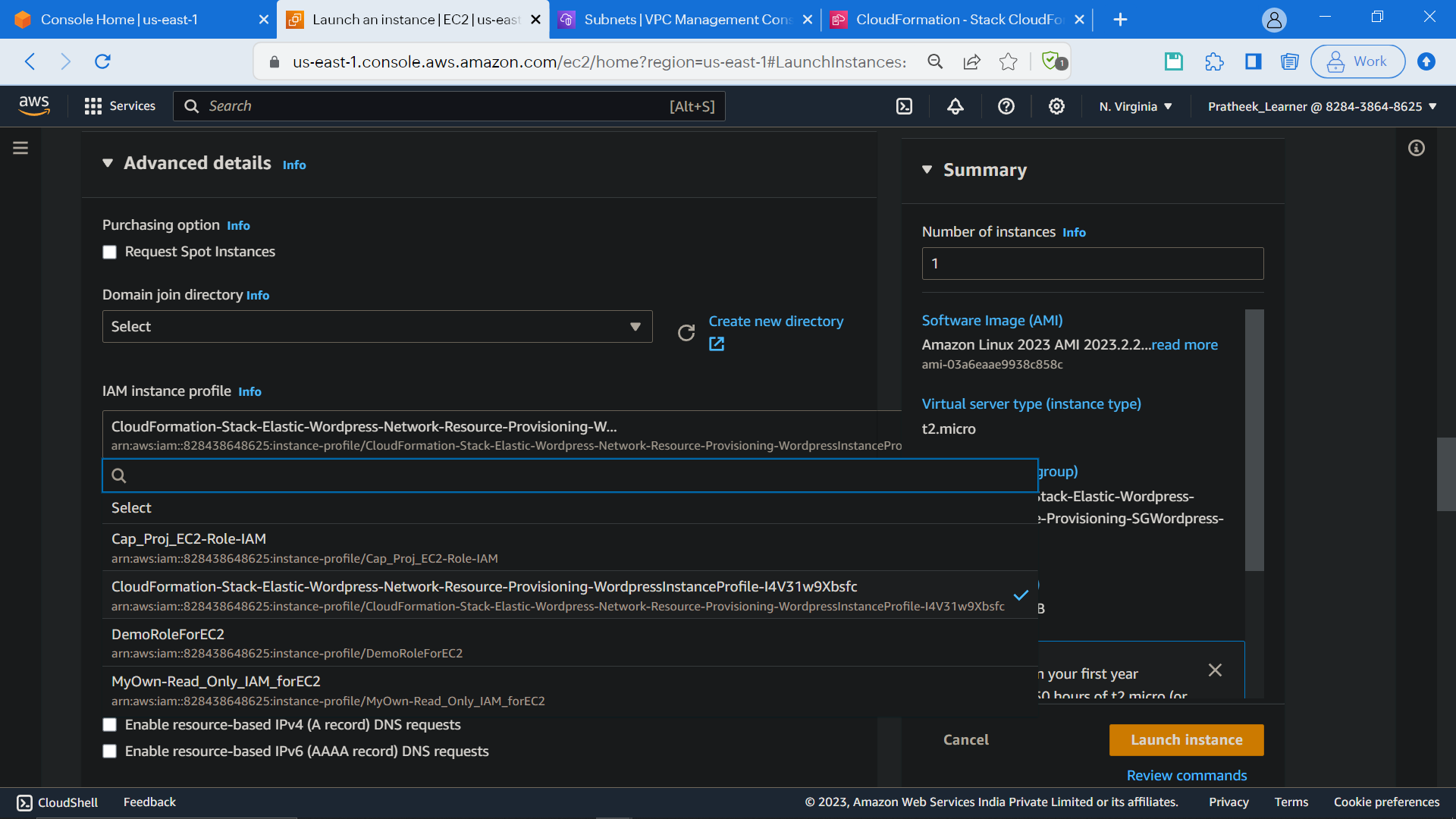


Now let’s create an EC2 WordPress instance. Use the following options for Network settings

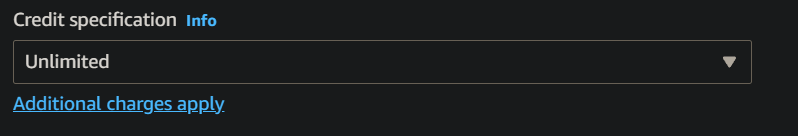


Now in Advanced options tweak following options:

IAM instance profile (Tick option)



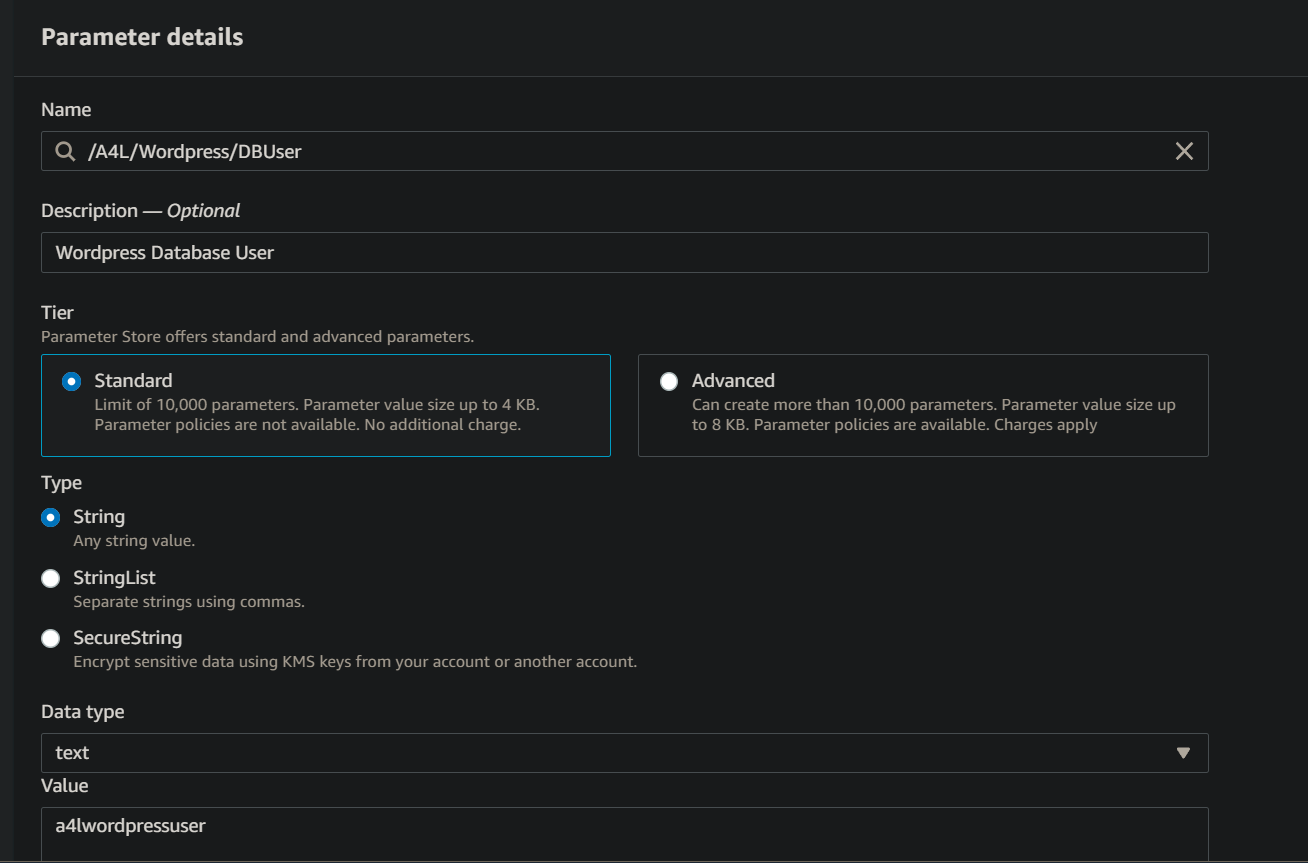
Credit specification:- (standard if no unlimited available)



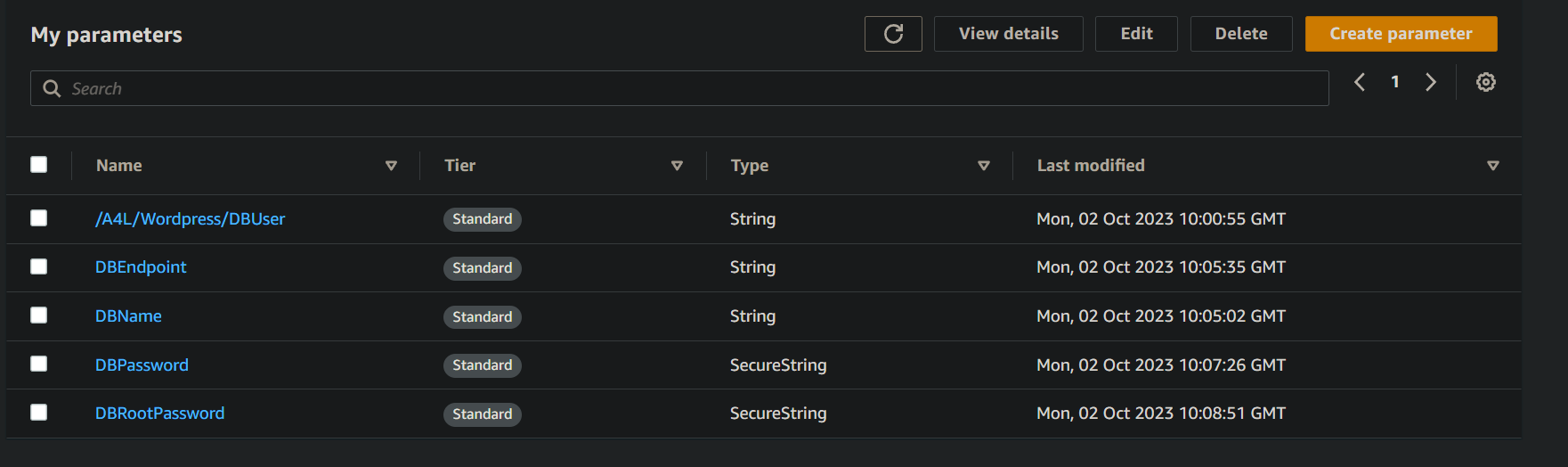
Configuration Done. Create EC2

Now we go to systems manager to automate some build process for future. Storing configuration information within the SSM Parameter store scales much better than attempting to script them in some way Go here & then click create parameter. (Note:- **MAKE SURE WITH THE BELOW ... NO WHITESPACE BEFORE OR AFTER .. MAKE SURE THE UPPER/LOWER CASE IS CORRECT) (Refer txt file “AWS SSM Parameters Config file.txt” in** code & config files folder).

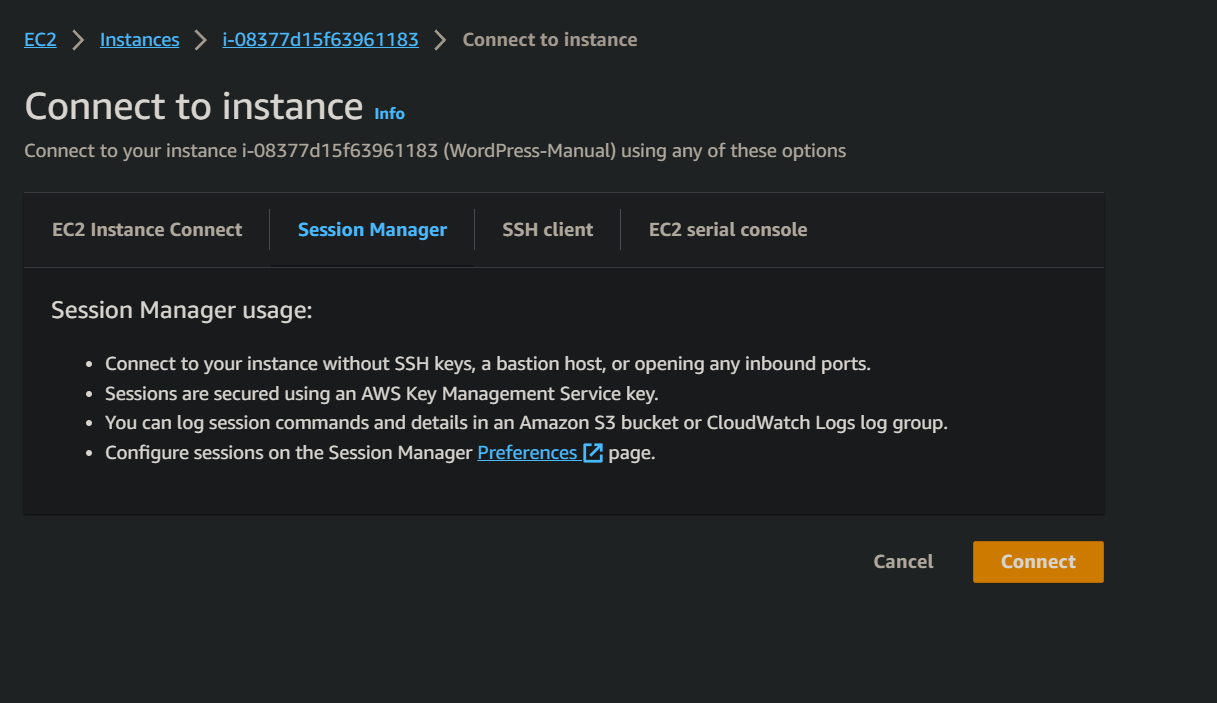




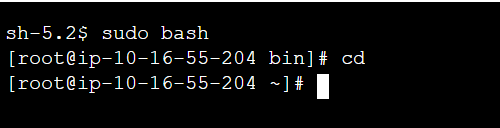
Create parameters for all 4 other parameters like this referring file.



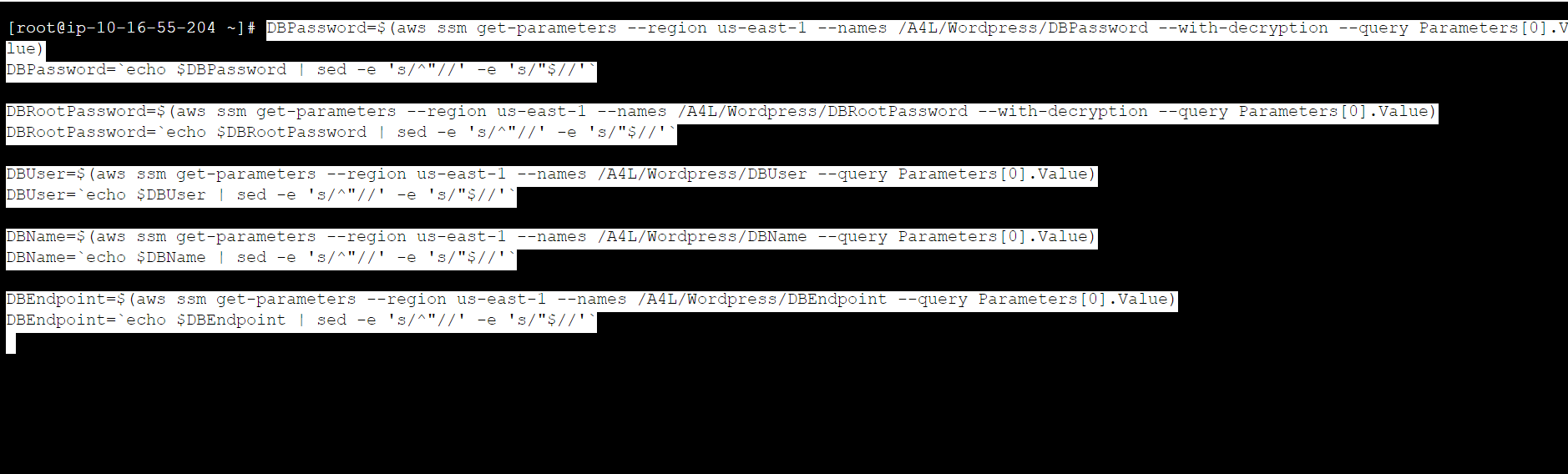
Now connect to EC2 instance using session manager.



Type following 2 commnads



Now run these commands to configure the parameters we have created into this EC2 instance. (Refer txt file “Parameters setup to WordPress EC2.txt” from **in** code & config files folder).



Next command :-

sudo dnf -y update

**Install Pre-Reqs and Web Server:**

Command: - sudo dnf install wget php-mysqlnd httpd php-fpm php-mysqli mariadb105-server php-json php php-devel stress -y

**Set DB and HTTP Server to running and start by default:**

sudo systemctl enable httpd

sudo systemctl enable mariadb

sudo systemctl start httpd

sudo systemctl start mariadb

**Set the MariaDB Root Password:** (Takes root password from the parameter that we have created earlier)

sudo mysqladmin -u root password $DBRootPassword

**Download and extract Wordpress:**

sudo wget http://wordpress.org/latest.tar.gz -P /var/www/html

cd /var/www/html

sudo tar -zxvf latest.tar.gz

sudo cp -rvf wordpress/\* .

sudo rm -R wordpress

sudo rm latest.tar.gz

**Configure the wordpress wp-config.php file:**

sudo cp ./wp-config-sample.php ./wp-config.php

sudo sed -i "s/'database\_name\_here'/'$DBName'/g" wp-config.php

sudo sed -i "s/'username\_here'/'$DBUser'/g" wp-config.php

sudo sed -i "s/'password\_here'/'$DBPassword'/g" wp-config.php

**Configure the wordpress wp-config.php file:** (Replaces some placeholders)

sudo cp ./wp-config-sample.php ./wp-config.php

sudo sed -i "s/'database\_name\_here'/'$DBName'/g" wp-config.php

sudo sed -i "s/'username\_here'/'$DBUser'/g" wp-config.php

sudo sed -i "s/'password\_here'/'$DBPassword'/g" wp-config.php

**Fix Permissions on the filesystem:**

sudo usermod -a -G apache ec2-user

sudo chown -R ec2-user:apache /var/www

sudo chmod 2775 /var/www

sudo find /var/www -type d -exec chmod 2775 {} \;

sudo find /var/www -type f -exec chmod 0664 {} \;

**Create Wordpress User, set its password, create the database and configure permissions:**

sudo echo "CREATE DATABASE $DBName;" >> /tmp/db.setup

sudo echo "CREATE USER '$DBUser'@'localhost' IDENTIFIED BY '$DBPassword';" >> /tmp/db.setup

sudo echo "GRANT ALL ON $DBName.\* TO '$DBUser'@'localhost';" >> /tmp/db.setup

sudo echo "FLUSH PRIVILEGES;" >> /tmp/db.setup

sudo mysql -u root --password=$DBRootPassword < /tmp/db.setup

sudo rm /tmp/db.setup