# Network Security: Enabling HTTPS port in Public Subnets

**We'll cover the following**



## Objective #

Make our instances inaccessible from the internet.

## Steps #

- Only allow the HTTPS port in the public subnets.

---

## Allow only the HTTPS port in public subnets #

Once the hosts are running inside private subnets and with the private security group, we can remove ports 8443 and 22 from the public security group. If we had done this in the previous step, it would have prevented users from reaching our application until the new hosts were created.

```
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId: !Ref VPC
    GroupDescription:
      !Sub 'Security group for ${AWS::StackName}'
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 443
        ToPort: 443
        CidrIp: 0.0.0.0/0
    Tags:
      - Key: Name
        Value: !Ref AWS::StackName
```

stage.yml

**Line #7:** Only port 443 is allowed in the public subnet.

Now let's deploy and test.

```
./deploy-infra.sh


========== Deploying setup.yml ==========

Waiting for changeset to be created..

No changes to deploy. Stack awsbootstrap-setup is up to date


========== Packaging main.yml ==========


========== Deploying main.yml ==========

Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - awsbootstrap
[
    "https://prod.the-good-parts.com",
    "https://staging.the-good-parts.com"
]
```

terminal

```
for run in {1..20}; do curl -s https://staging.the-good-parts.com; done | sort | uniq -c
11 Hello HTTPS World from ip-10-0-187-72.ec2.internal in awsbootstrap-Staging-10LP6MF0TQC9Y
9 Hello HTTPS World from ip-10-0-222-16.ec2.internal in awsbootstrap-Staging-10LP6MF0TQC9Y
```

terminal

```
for run in {1..20}; do curl -s https://prod.the-good-parts.com; done | sort | uniq -c
10 Hello HTTPS World from ip-10-0-128-220.ec2.internal in awsbootstrap-Prod-1PT61TNHUQWTE
10 Hello HTTPS World from ip-10-0-248-112.ec2.internal in awsbootstrap-Prod-1PT61TNHUQWTE
```

terminal

Our instances are now isolated from the internet, and the only way to reach them is through the load balancer.

```
git add stage.yml
git commit -m "Only allow port 443 in public subnet"
git push
```

terminal

> **Note:** All the code has been already added and we are pushing it on our

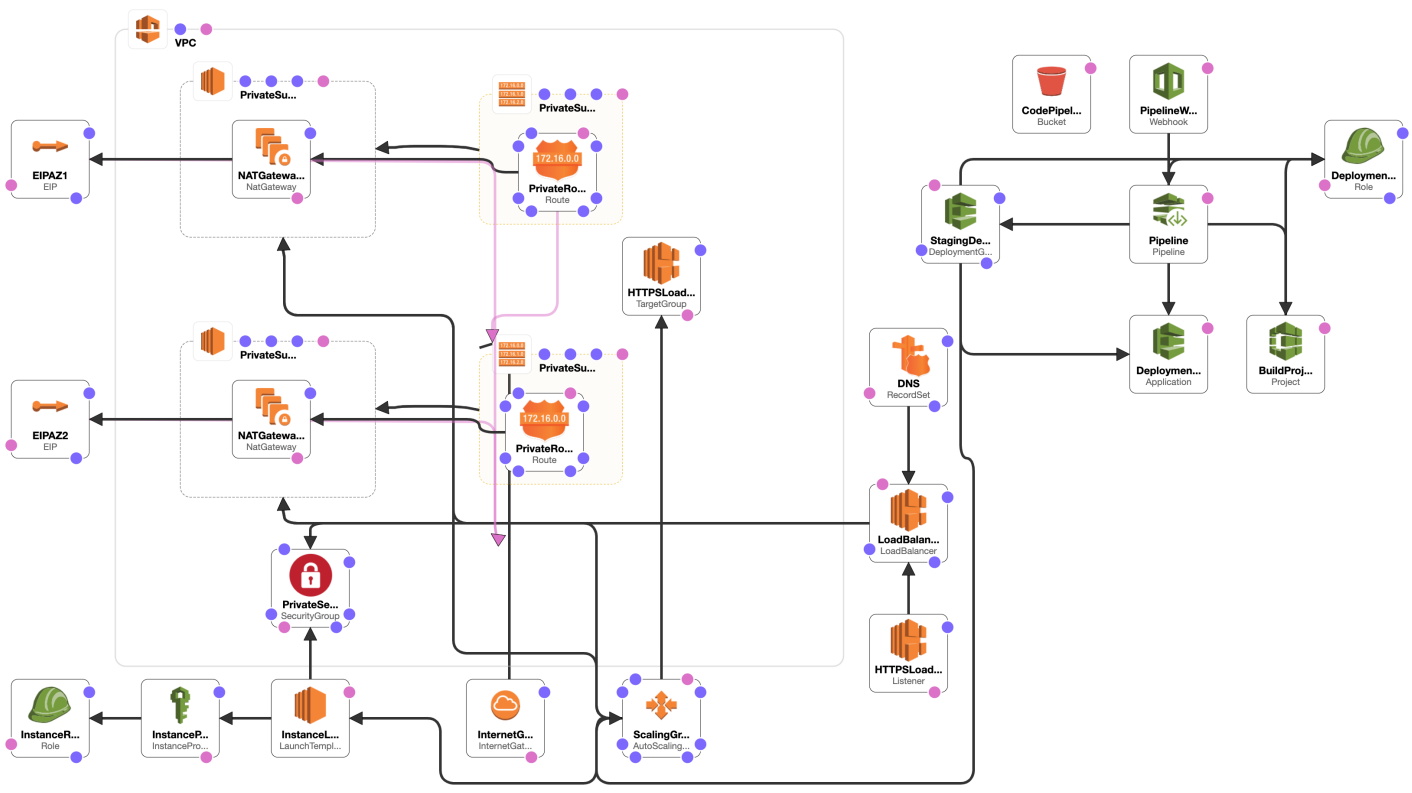
repository as well.

This code requires the following API keys to execute:

| | |
|---|---|
| username | Not Specified... |
| AWS_ACCESS_KE... | Not Specified... |
| AWS_SECRET_AC... | Not Specified... |
| AWS_REGION | us-east-1 |
| Github_Token | Not Specified... |

```
{
  "name": "aws-bootstrap",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "start": "node ./node_modules/pm2/bin/pm2 start ./server.js --name hello_aws --log ../logs/app
    "stop": "node ./node_modules/pm2/bin/pm2 stop hello_aws",
    "build": "echo 'Building...'"
  },
  "dependencies": {
    "pm2": "^4.2.0"
  }
}
```

In order to get a pictorial view of our developed cloudformation stack so far, below is the design view which shows the resources we created and their relationships.




Network Security - Enabling HTTPS port

In the next lesson, we will wrap up our discussion on this course.