Practice 1 Guidelines

Getting Started with AWS: EC2, S3, Monitoring with CloudWatch Lambda

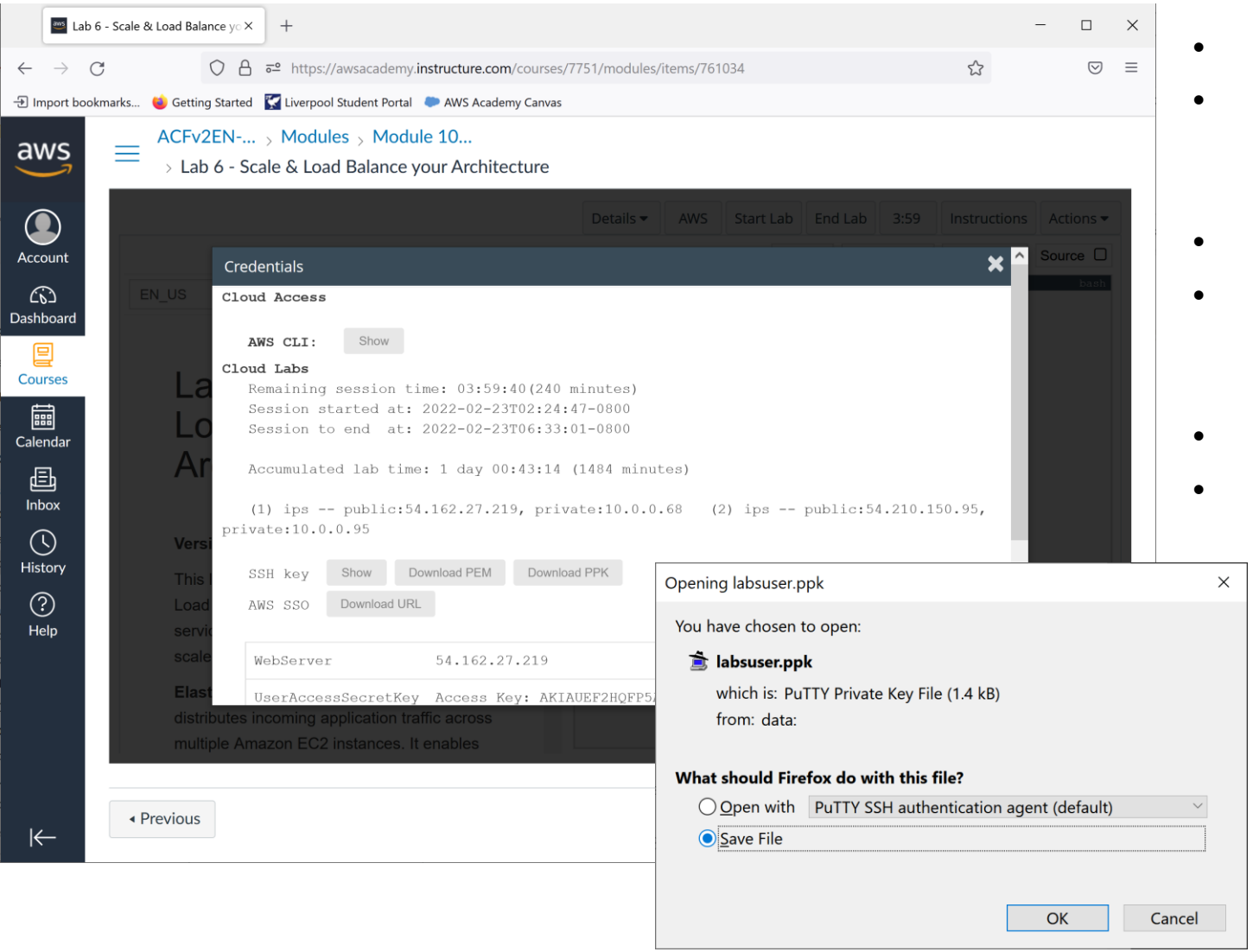
Dr. Yuri Demchenko
University of Amsterdam

Outline

- AWS Academy Sandbox

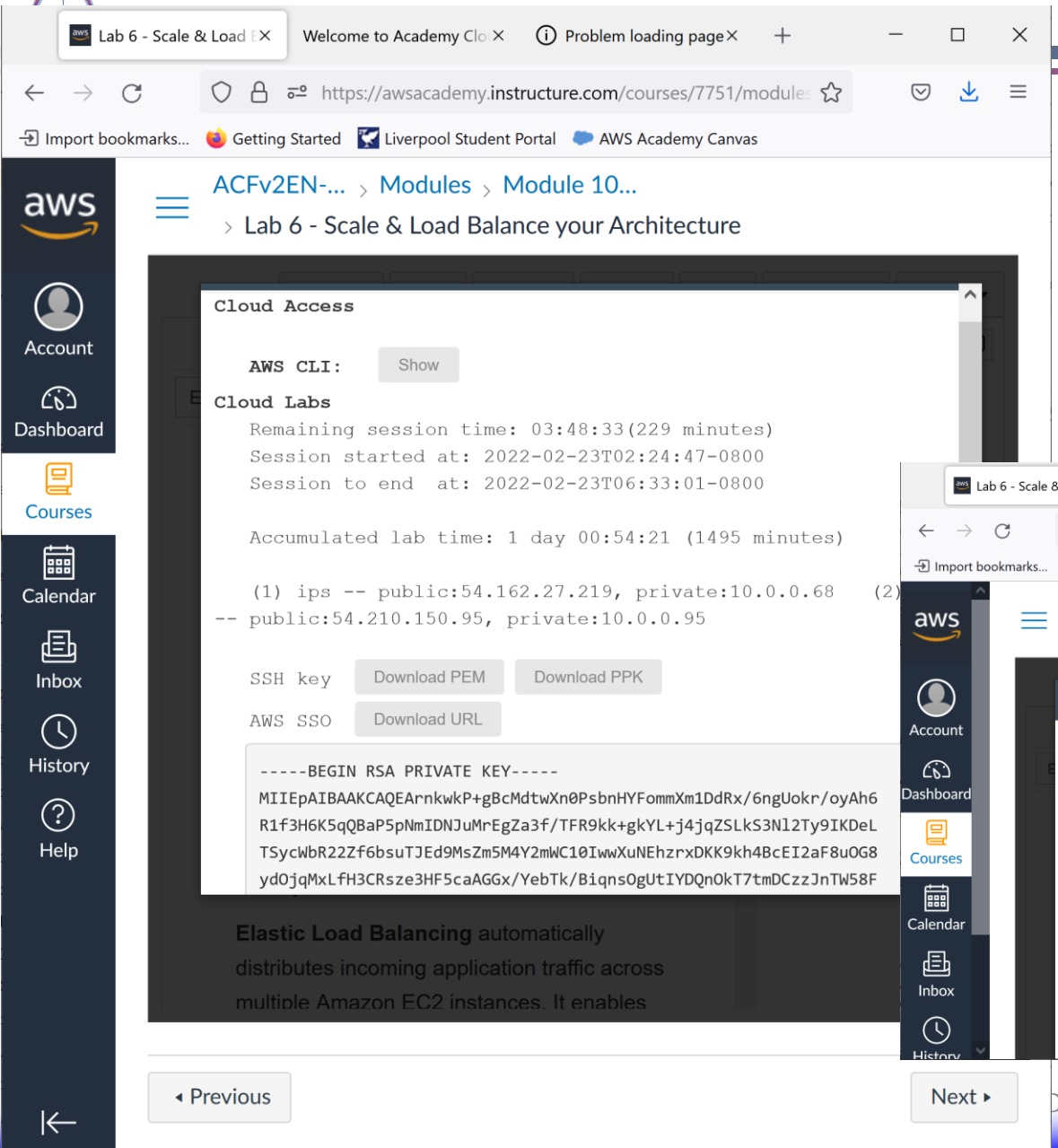
- EC2
  - Creating webserver on VM instance (deploying LAMP server)
- Configuring SSH client
  - Setting up tunnel to access cloud based services
- S3
- Adding monitoring to EC2 and S3
- Lambda
- Setting up CLI environment

AWS Academy Sandbox – Download Access Credentials

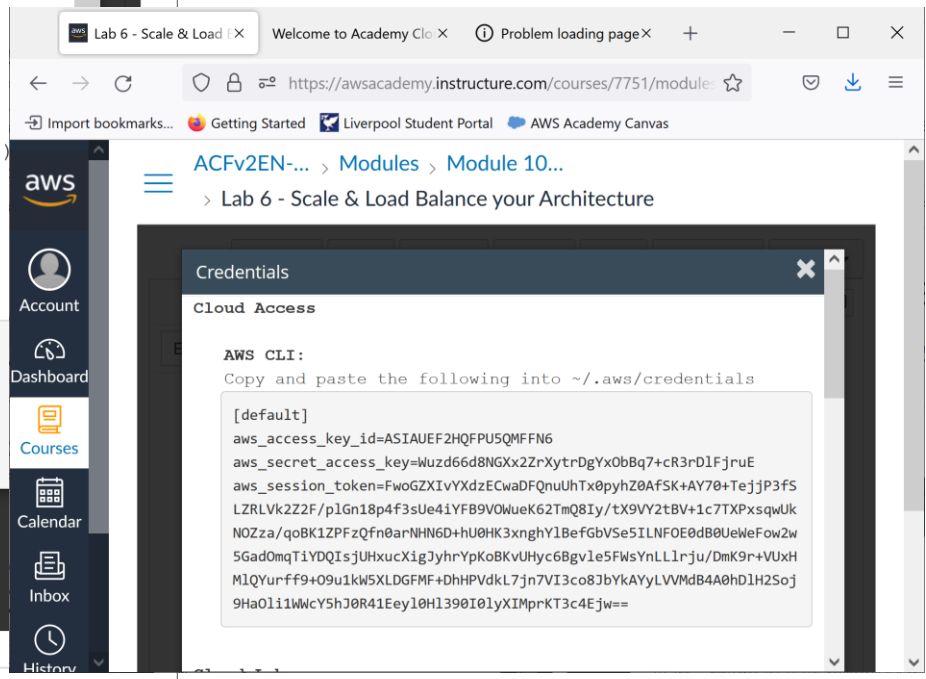


- Start Lab or Sandbox
- Dropdown Details > AWS > Show
- Download SSH key
- Optionally, you can use SSO service by AWS
- Test WebServer URL
- Use AWS CLI

Sandbox Access: SSH client and AWS CLI

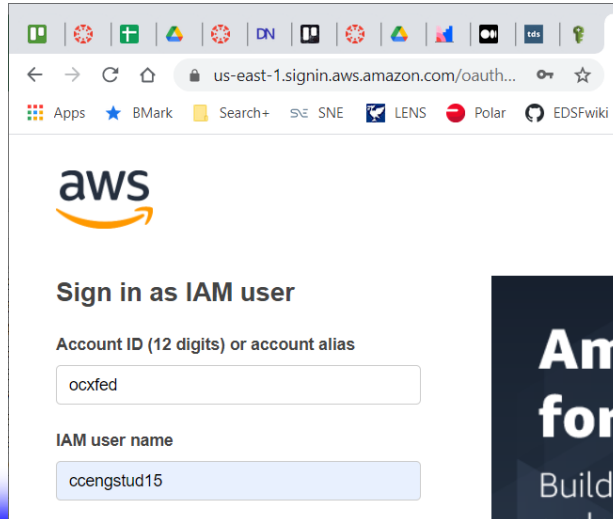


- For SSH client
  - Download SSH key
- For AWS CLI
  - Install AWS CLI
  - Modify/paste SSH key into ~/.aws/credentials

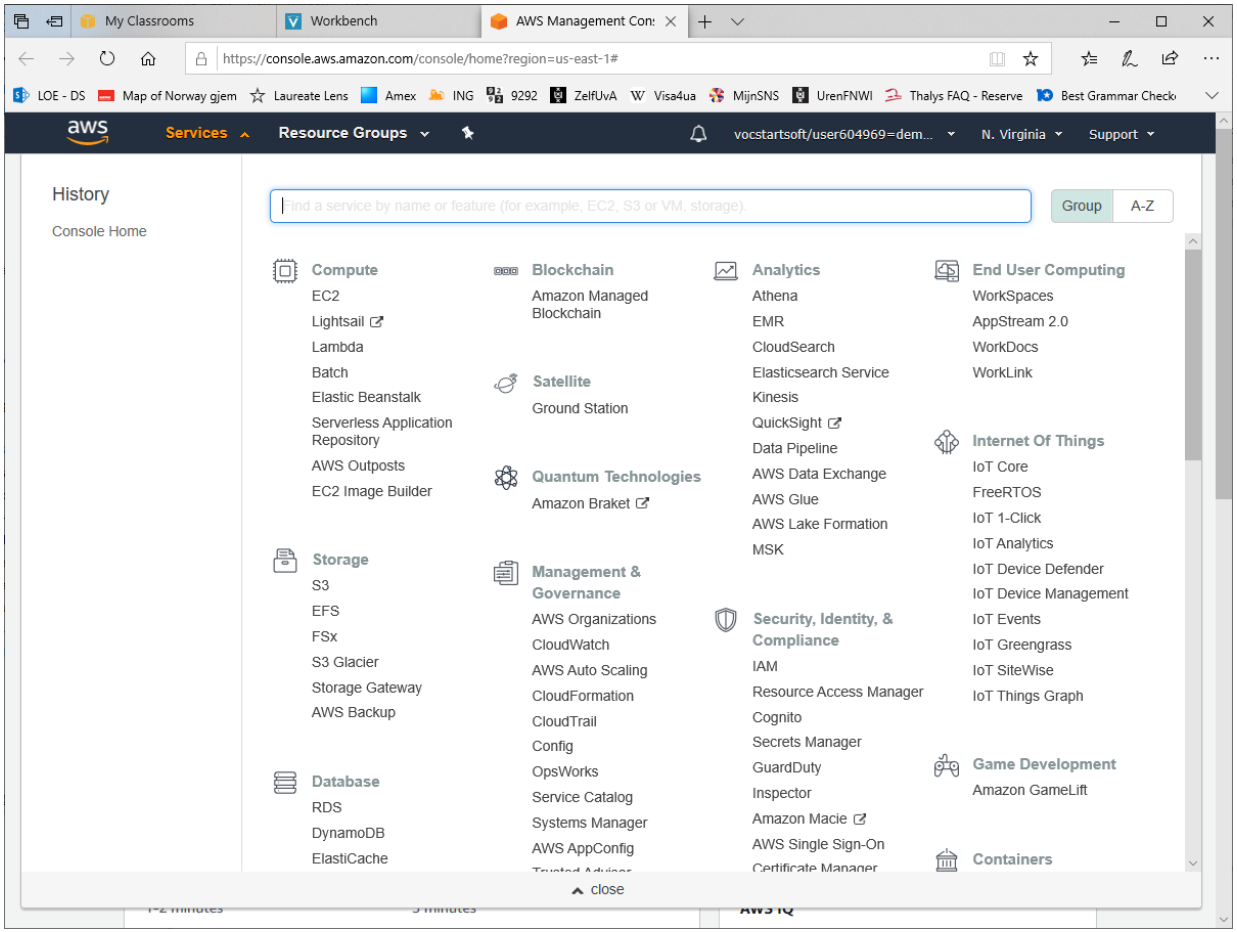


Accessing AWS portal

- From AWS Academy Lab sandbox
  - Use AWS class Sandbox
  - Start lab and access directly from the lab sandbox
- For separately configured class or project
  - Create AIM group and obtain group signin link
  - Example: Class or project Portal access: link, username, password
    - https://ocxfed.signin.aws.amazon.com/console
    - IAM user: cceng#
    - Password: ******



AWS Portal - Services

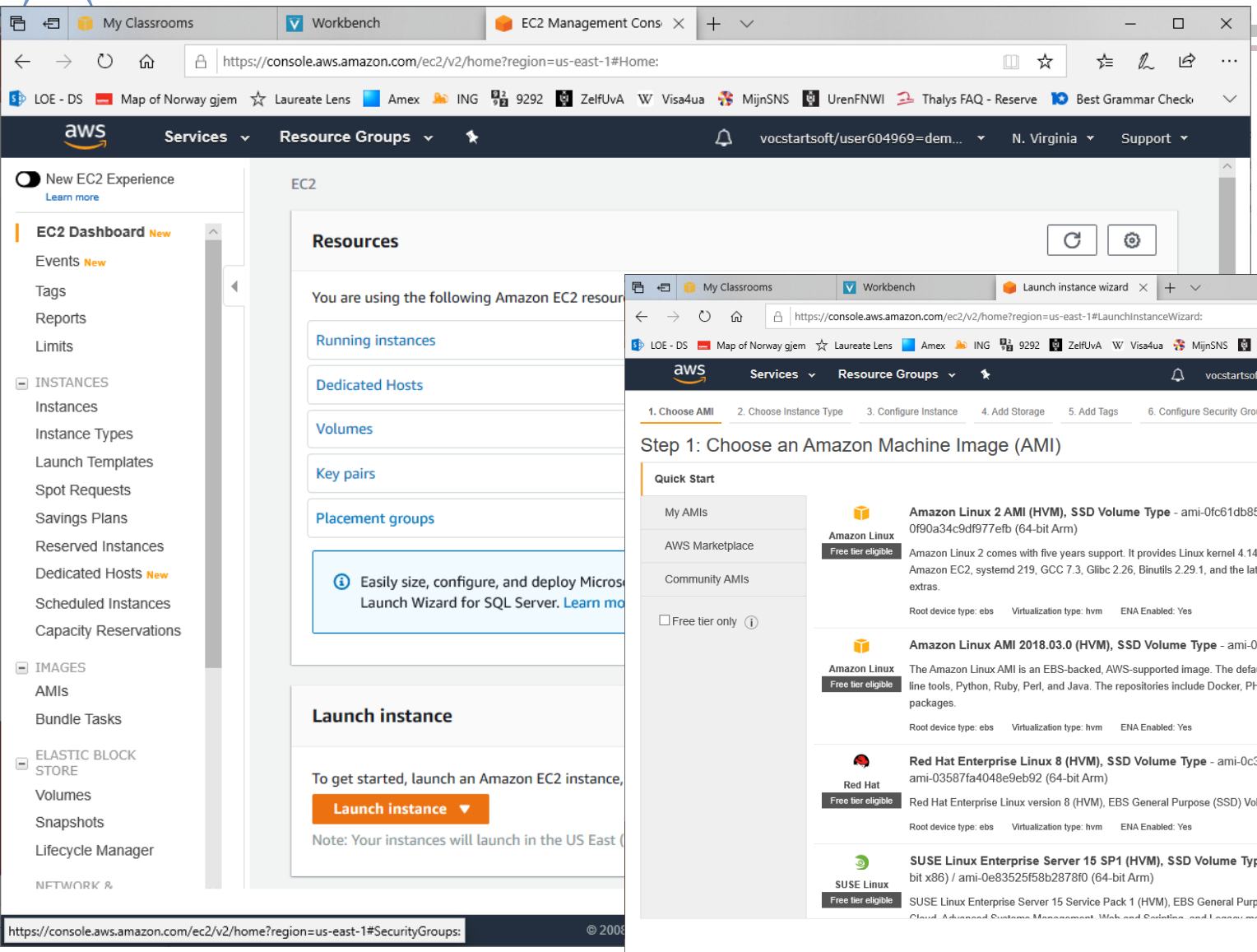


- Navigate through offered AWS cloud services
- Investigate services in each group Compute, Storage, Databases, Management, Security, Machine Learning, others

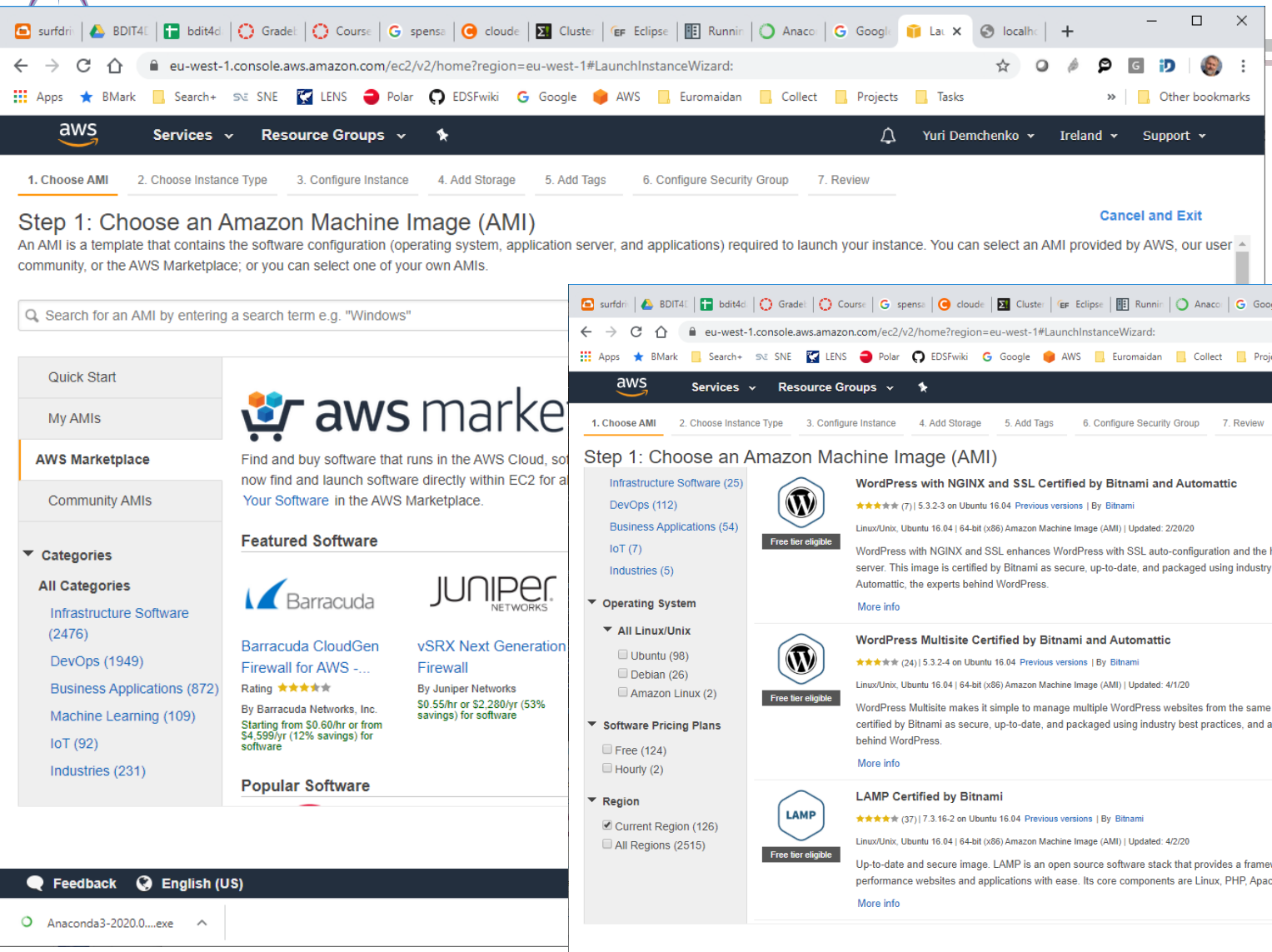
EC2 VM Instance: Naming and Tagging

- VM instance configuration and deployment best practices
- Use recognizable names for VM names, metadata, tags, etc
- E.g., in the shared class environment use identifiable VM names, e.g.
  - ccengstud10fist, ccengstud02test*, cceng08fist
  - where “ccengstud**” is you account ID
- Use similar naming for tags

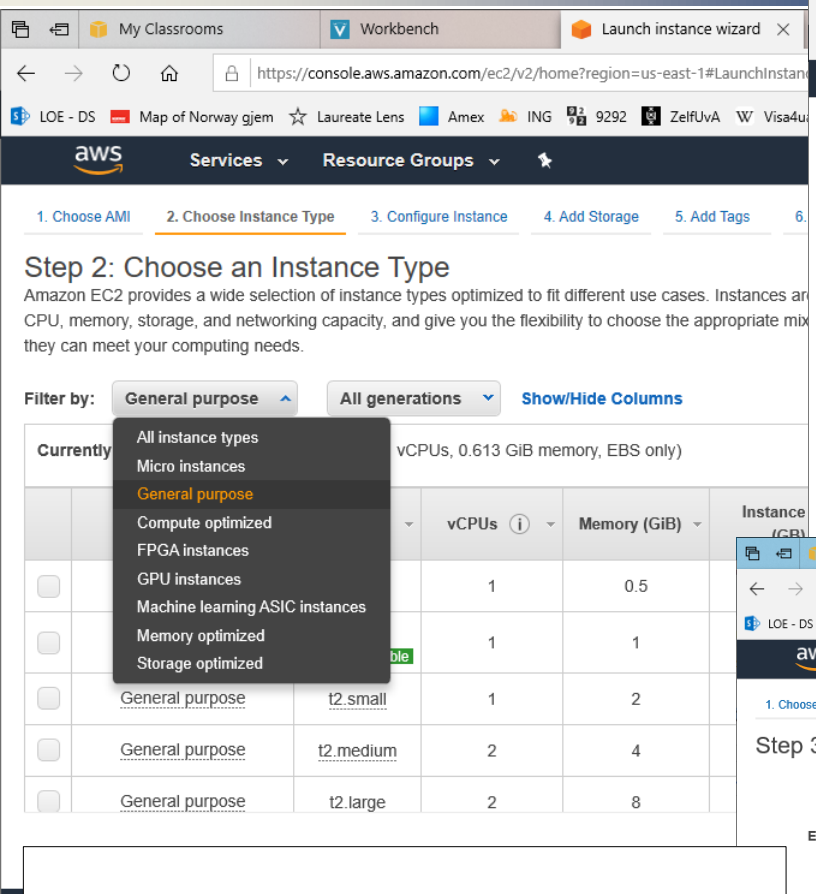
Select your AMI configuration When launching new VM instance

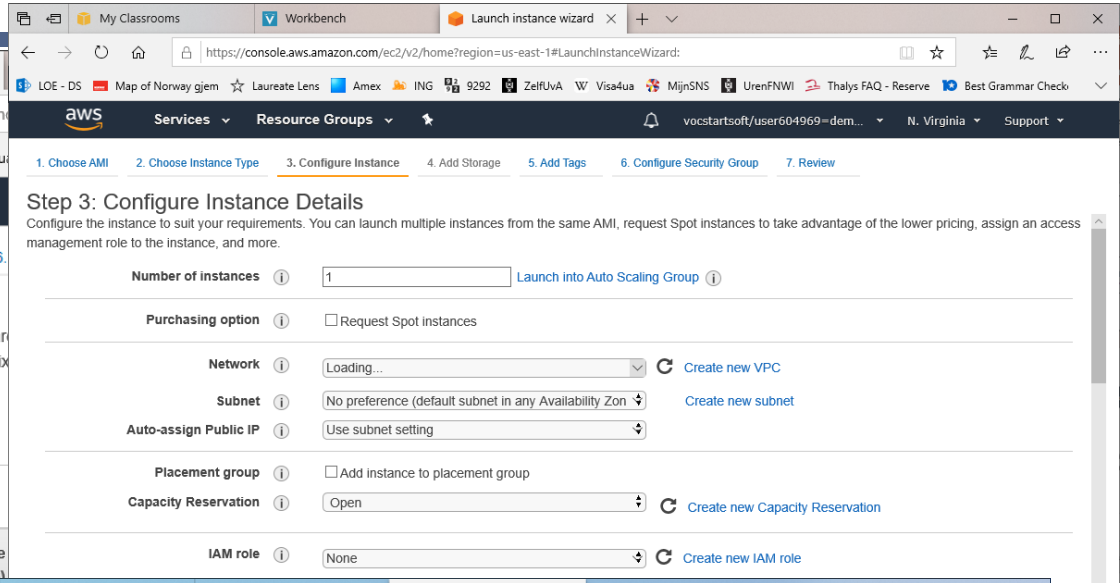


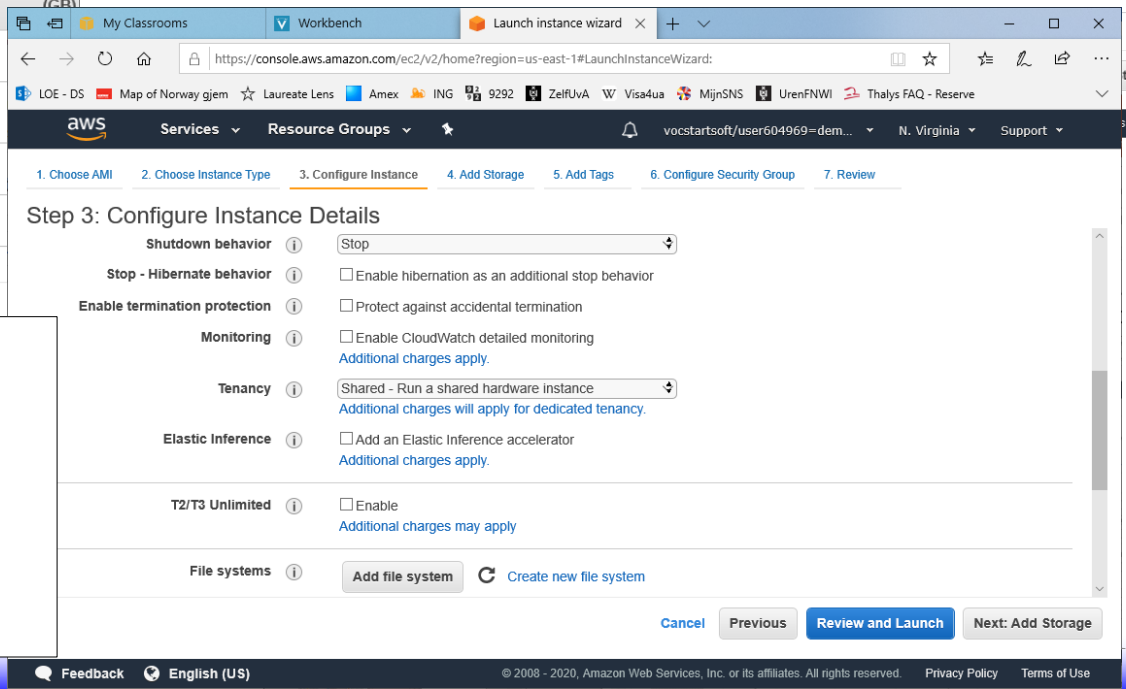
AWS Marketplace offers 3rd party VMs



Go through instance configuration details: Start from selecting instance type (Micro types for education purposes)

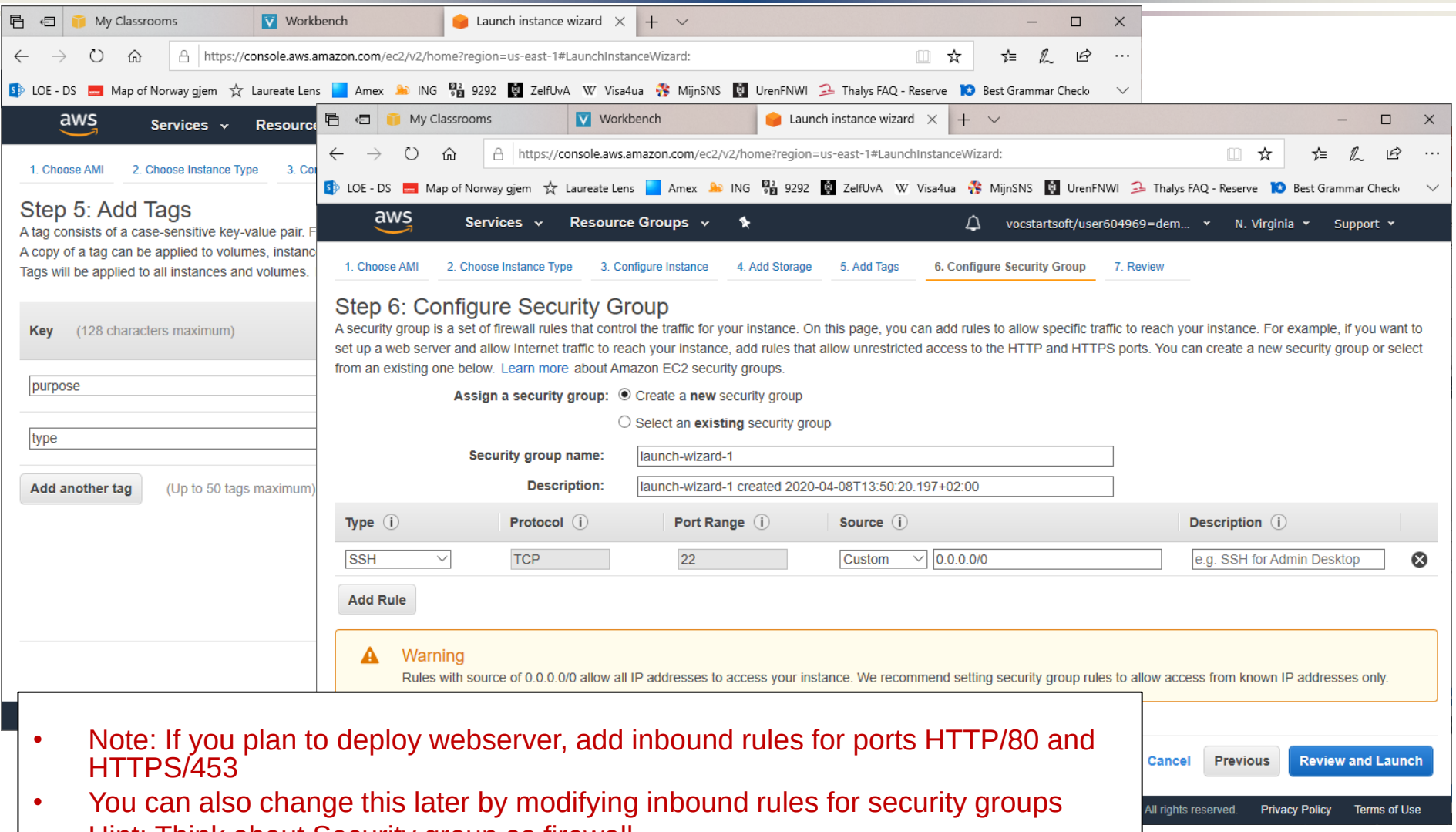






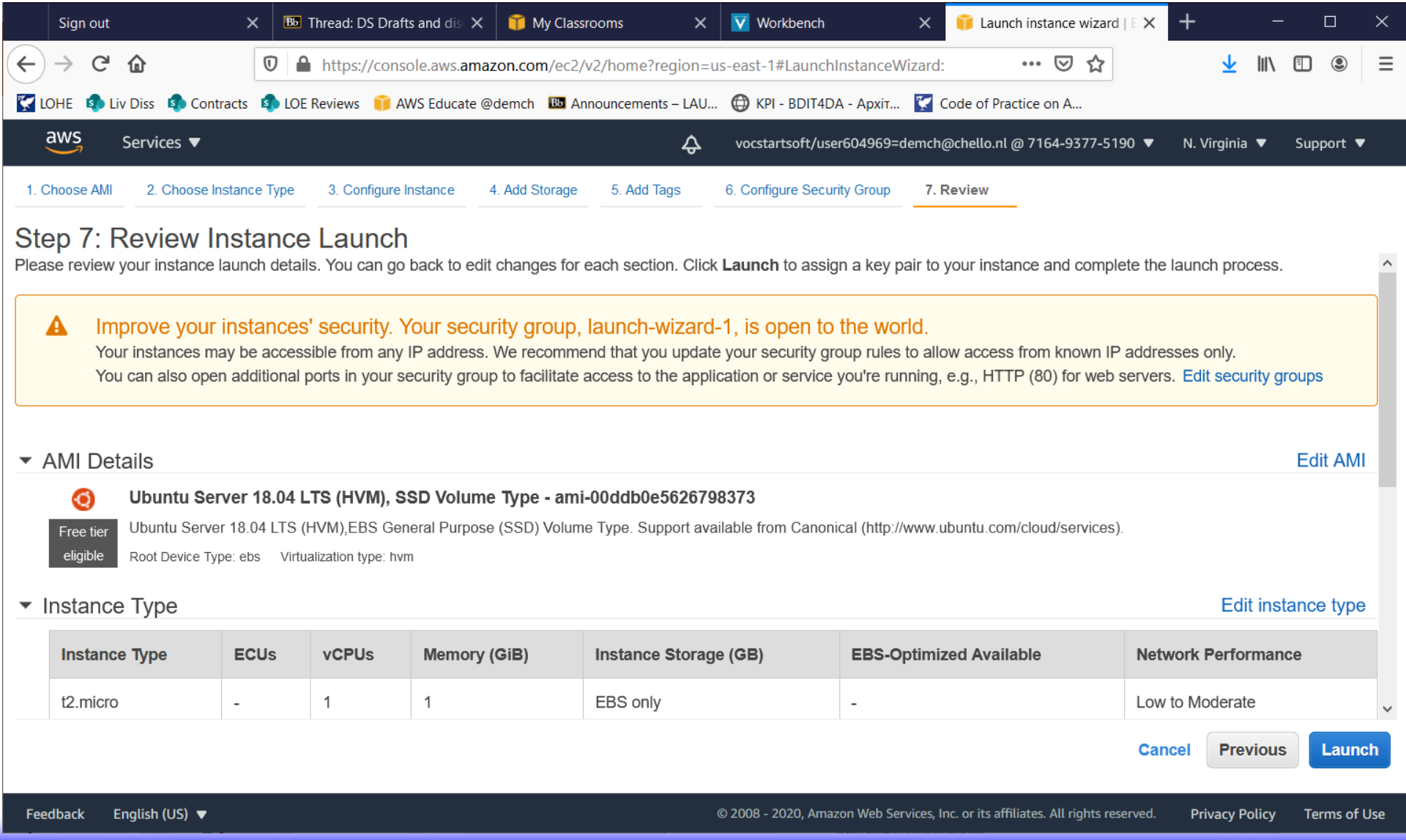
- Pay attention to configuration details
- Select new or already created VPC and subnet
- Enable detailed CloudWatch monitoring
- Decide on tenancy option

Add tags and Select Security Group: Use already created Security Group for your application

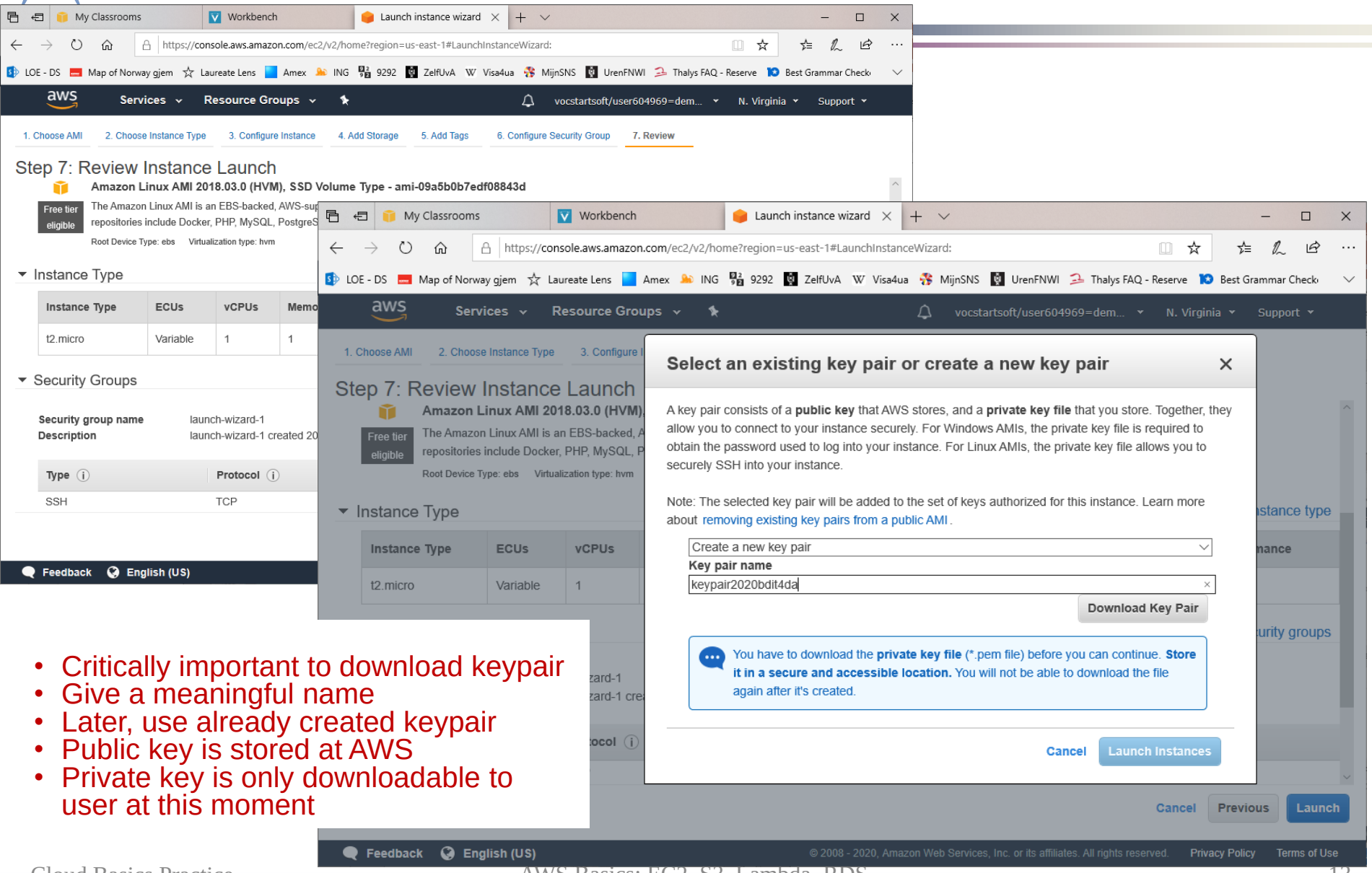


- Note: If you plan to deploy webserver, add inbound rules for ports HTTP/80 and HTTPS/453
- You can also change this later by modifying inbound rules for security groups
- Hint: Think about Security group as firewall

Final Step 7 before launch

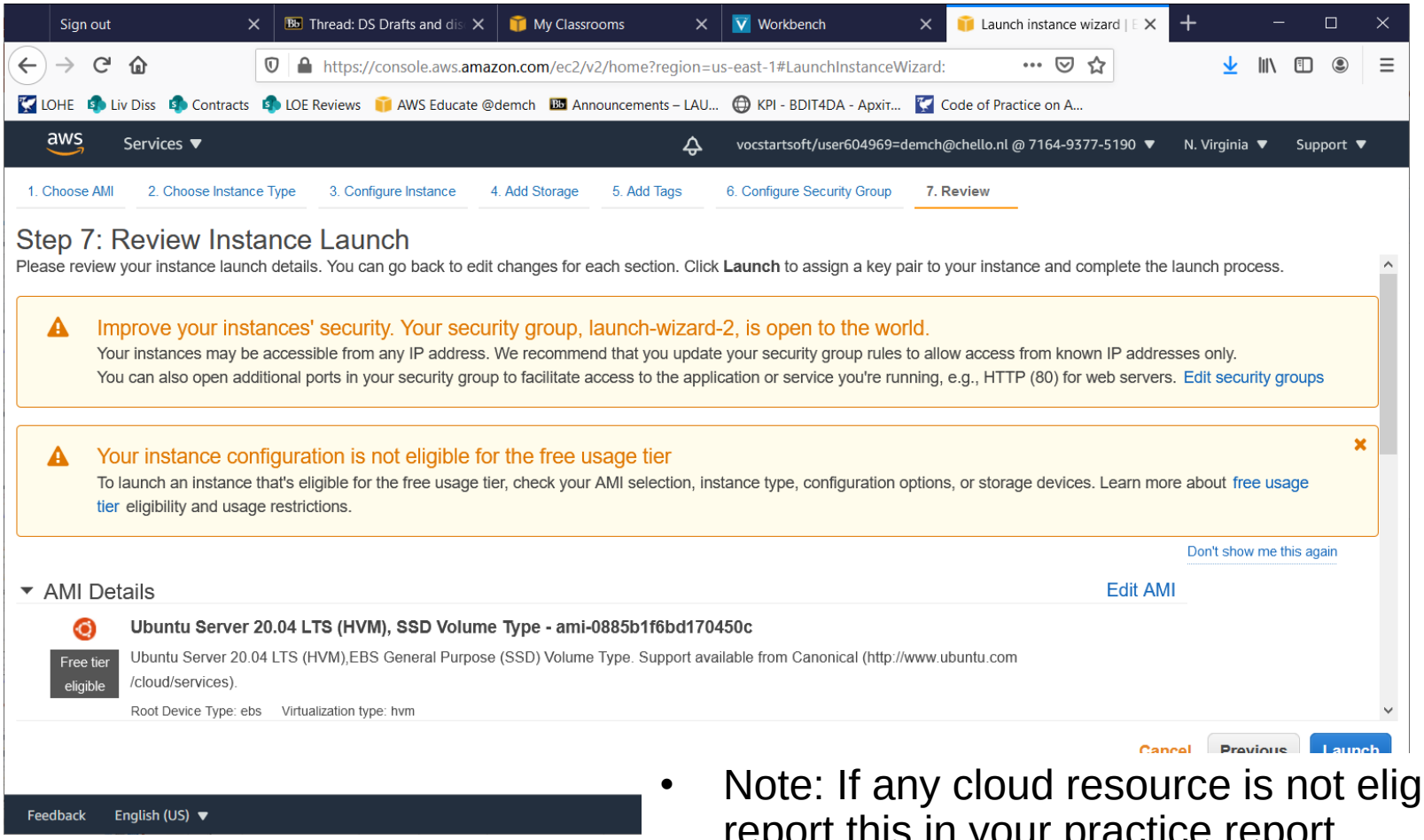


Launching AMI and downloading keypair



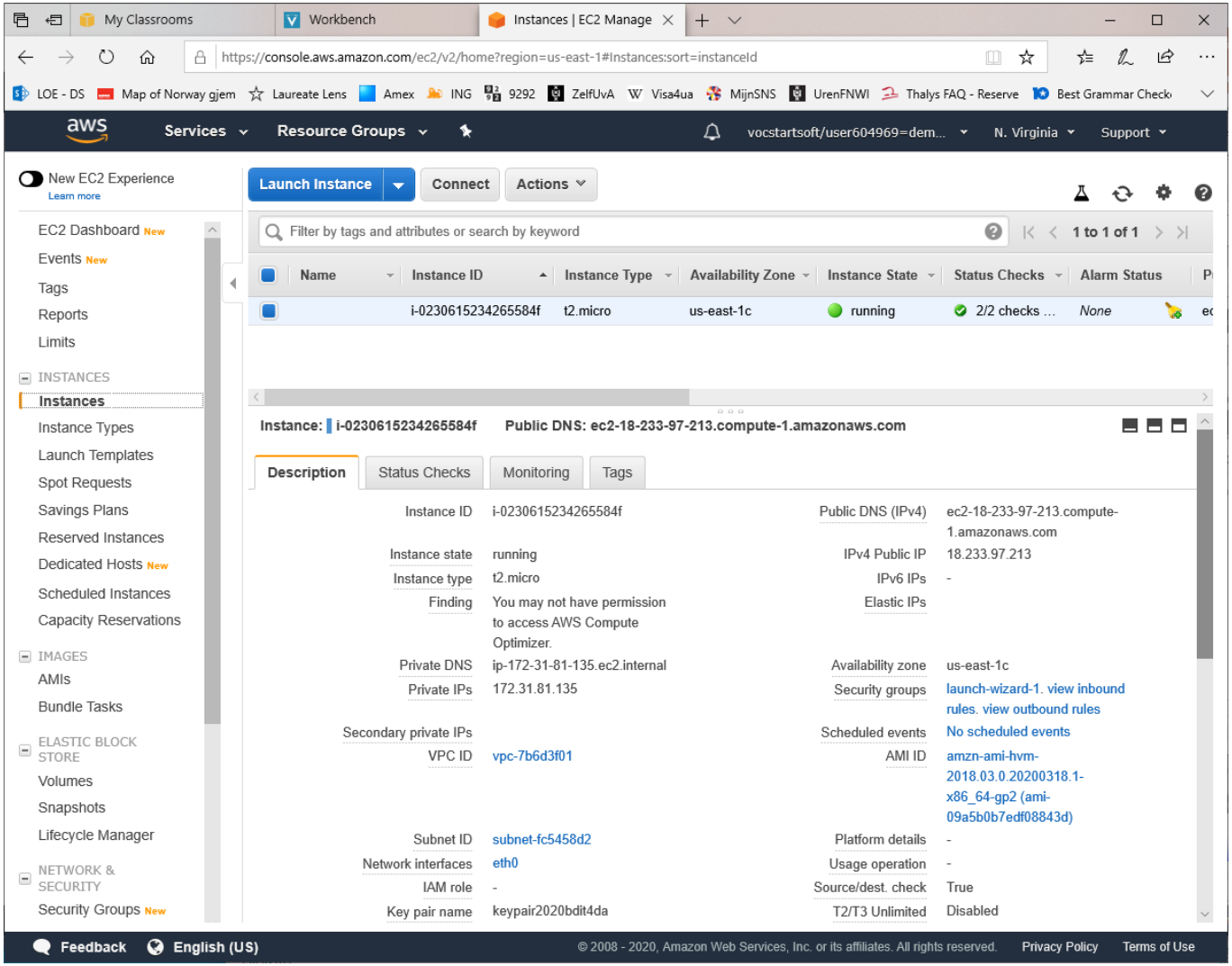
- Critically important to download keypair
- Give a meaningful name
- Later, use already created keypair
- Public key is stored at AWS
- Private key is only downloadable to user at this moment

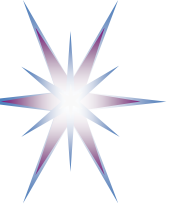
Launching Spot Instances – Is not eligible to AWS Academy Educate account



- Note: If any cloud resource is not eligible, report this in your practice report
- Try to find a solution, e.g. change region, use another instance type

VM Instance Overview





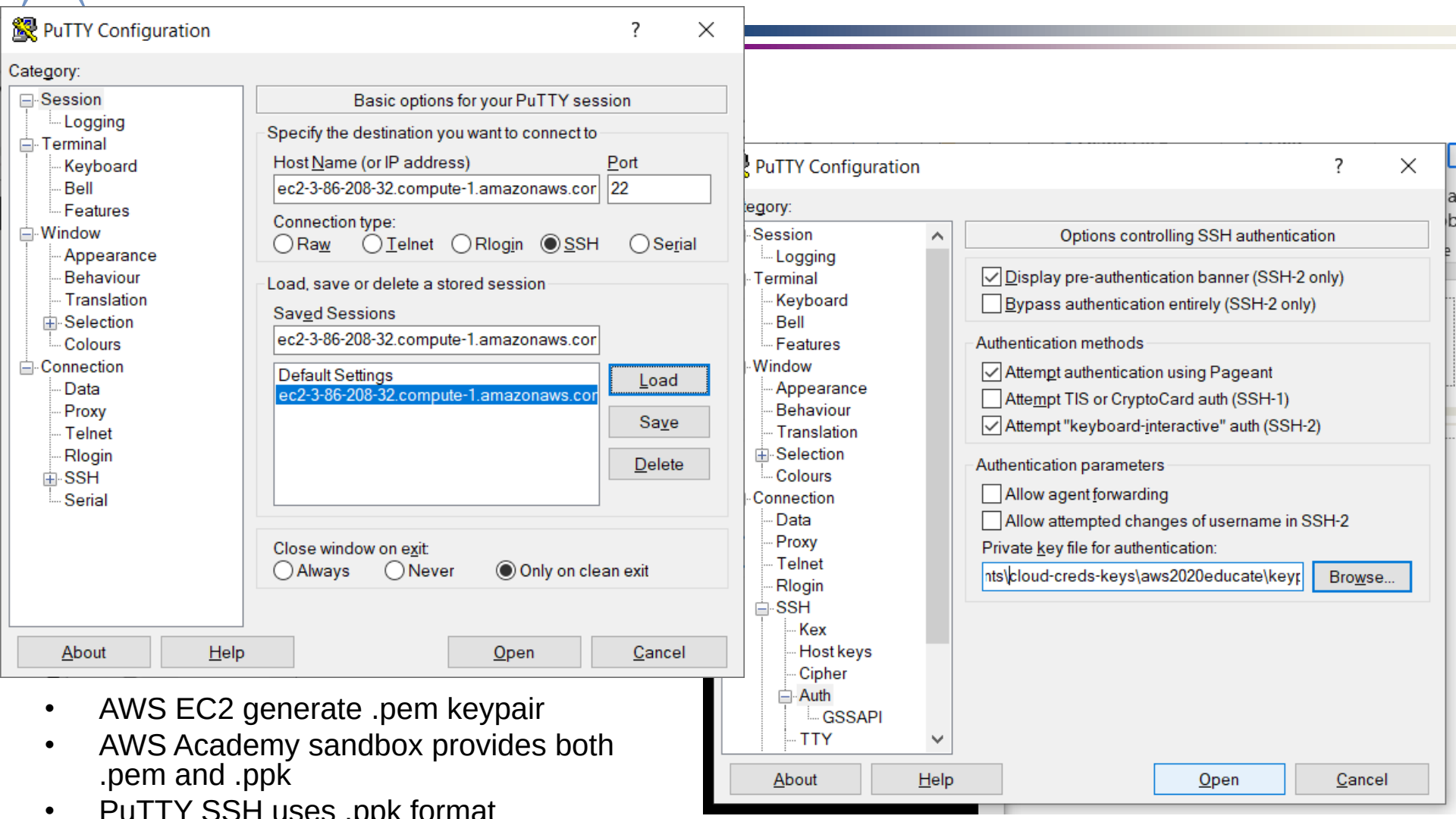
Example EC2 Instance Network Configuration

- Public DNS (IPv4): ec2-3-86-208-32.compute-1.amazonaws.com
- Public IPv4: 3.86.208.32
- Private DNS: ip-172-31-87-72.ec2.internal
- Private IP: ip-172-31-87-72.ec2.internal
- VPC ID: vpc-e3247699
- Subnet ID: subnet-449d906a

Accessing EC2 VM Instance with SSH client

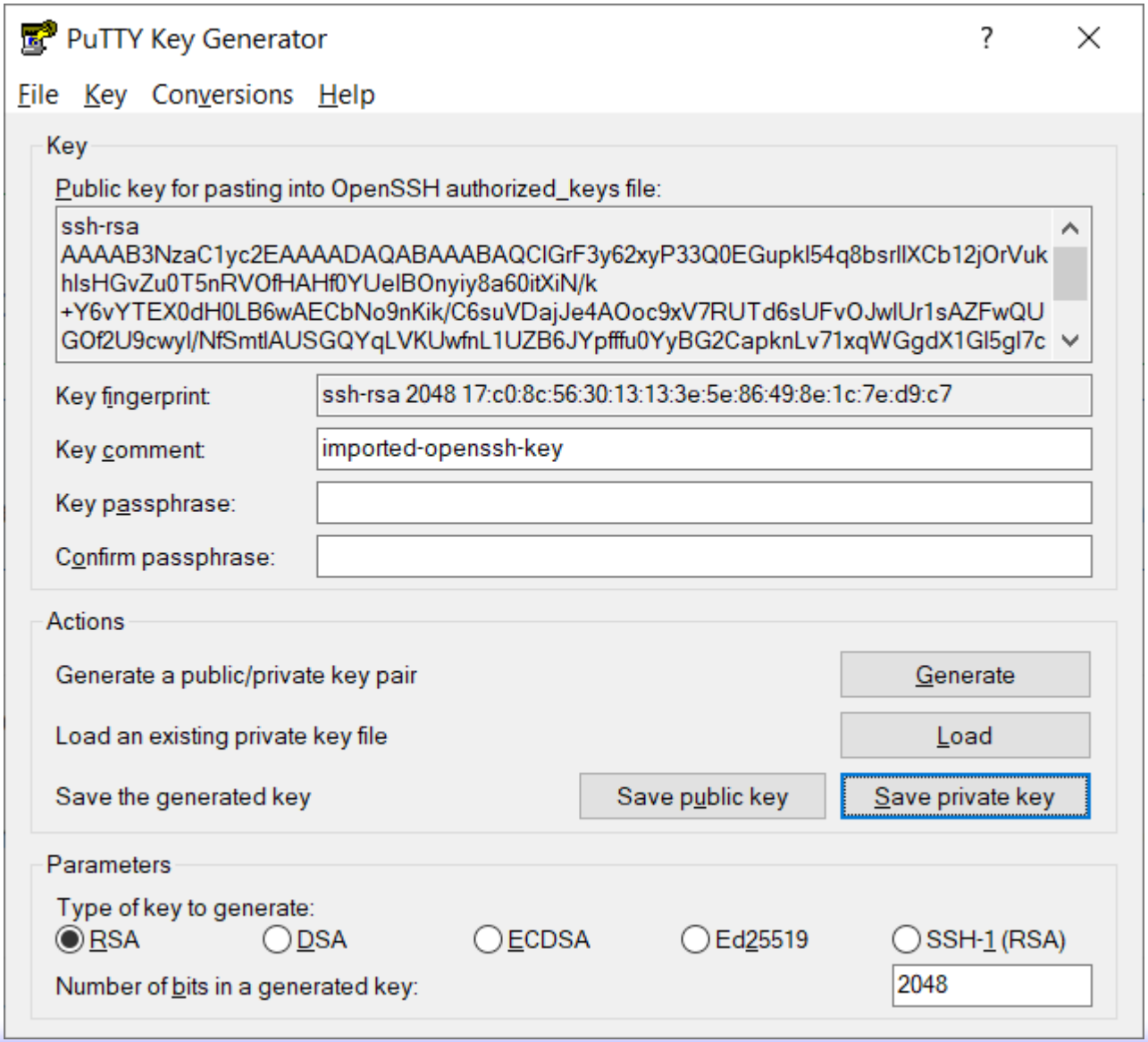
- SSH is a Secure Shell protocol
- Uses PKI based authentication
- You need to have private and public key
- Private key is only known to the user and used for authentication
- Public key is stored at the client/remote side

Configuring PuTTY SSH Client for Windows



- AWS EC2 generate .pem keypair
- AWS Academy sandbox provides both .pem and .ppk
- PuTTY SSH uses .ppk format
- Use PUTTYGEN.EXE to convert key

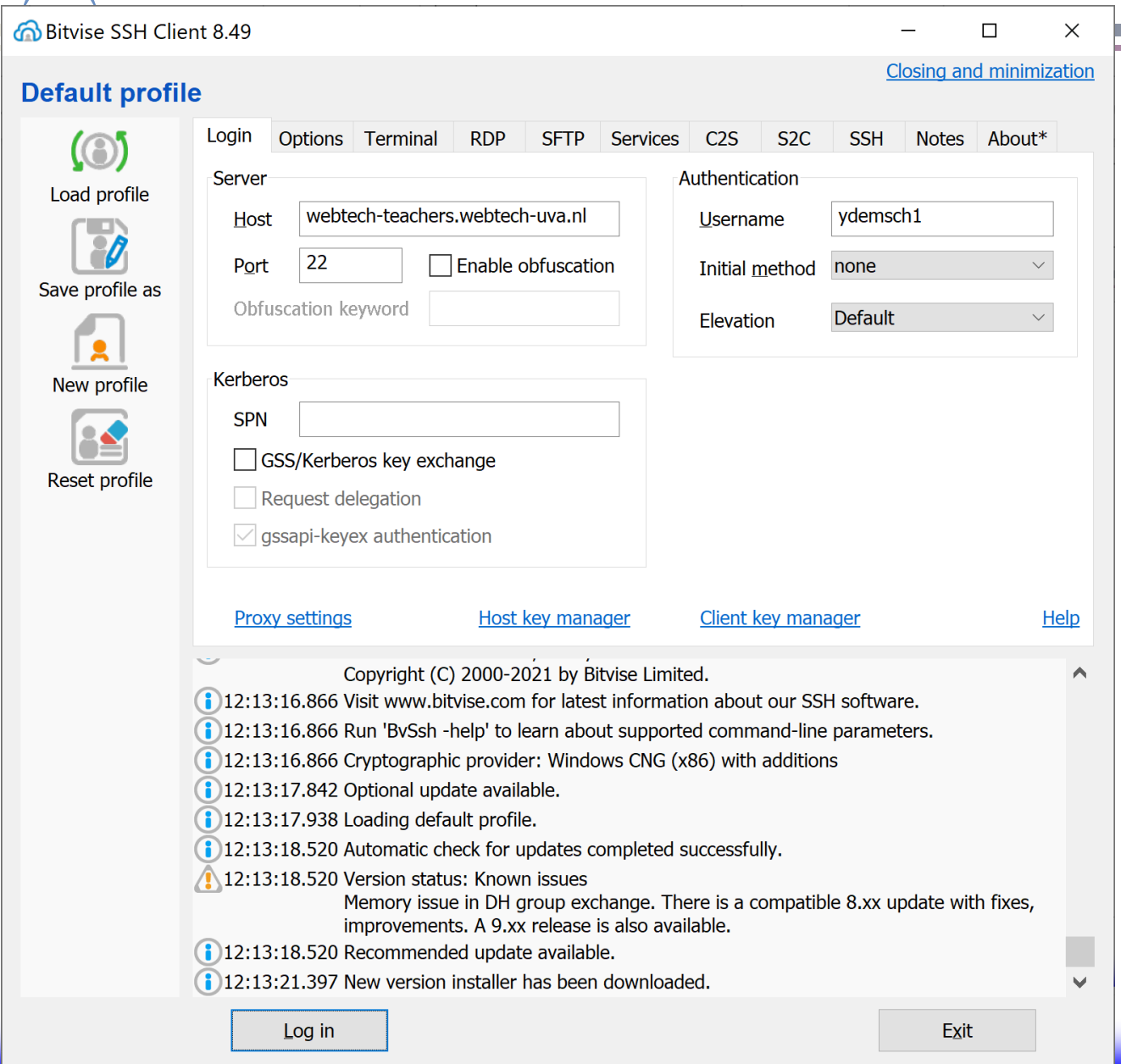
Using PuTTY Key Generator



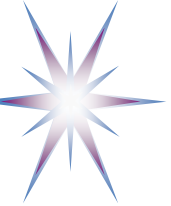
- Transform .pem key to .ppk
- Use recognizable key names
- Set password, e.g. pswd (optional)

Bitvise SSH Client

https://www.bitvise.com/ssh-client-download



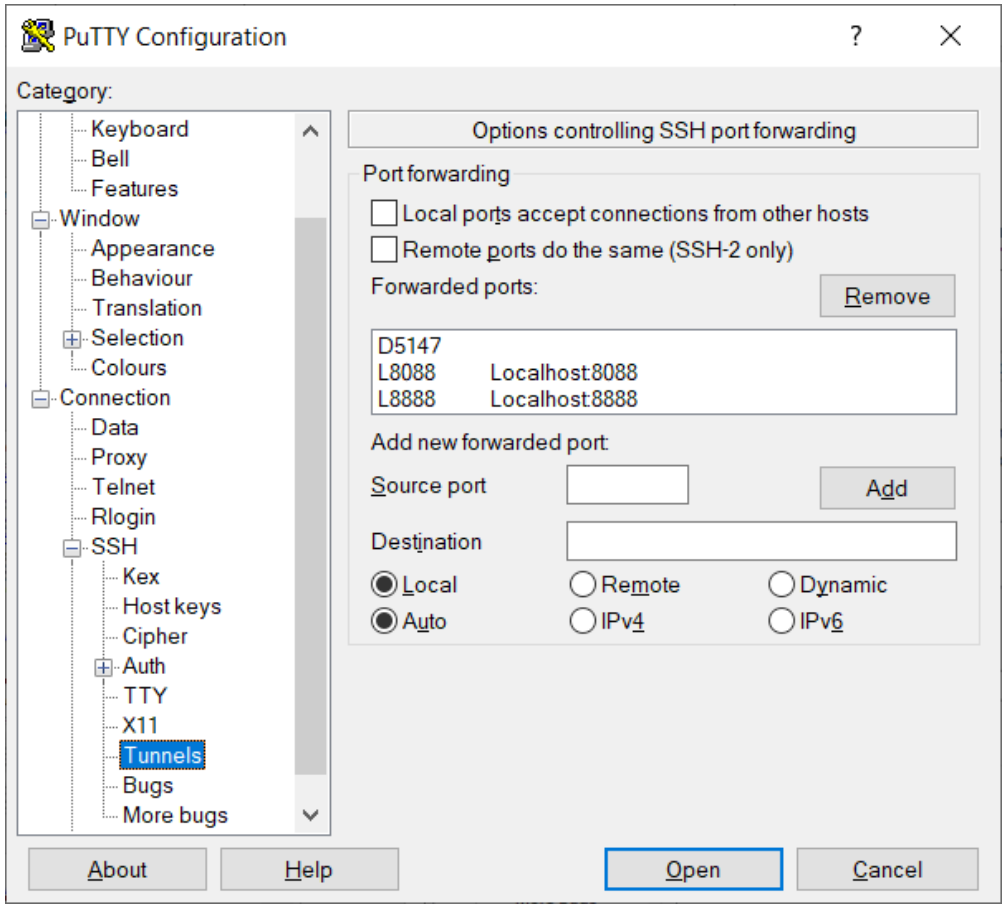
- Supports SSH, SFTP
- Key formats .pem and .ppk
- Multiple keys management



Setting up SSH Tunnel

- SSH Tunnel is required to access many cloud based services from the local/laptop client
  - Hadoop cluster, RDS database, SQL Workbench
- Configure SSH client for tunnel

Configuring Tunnels for known Ports



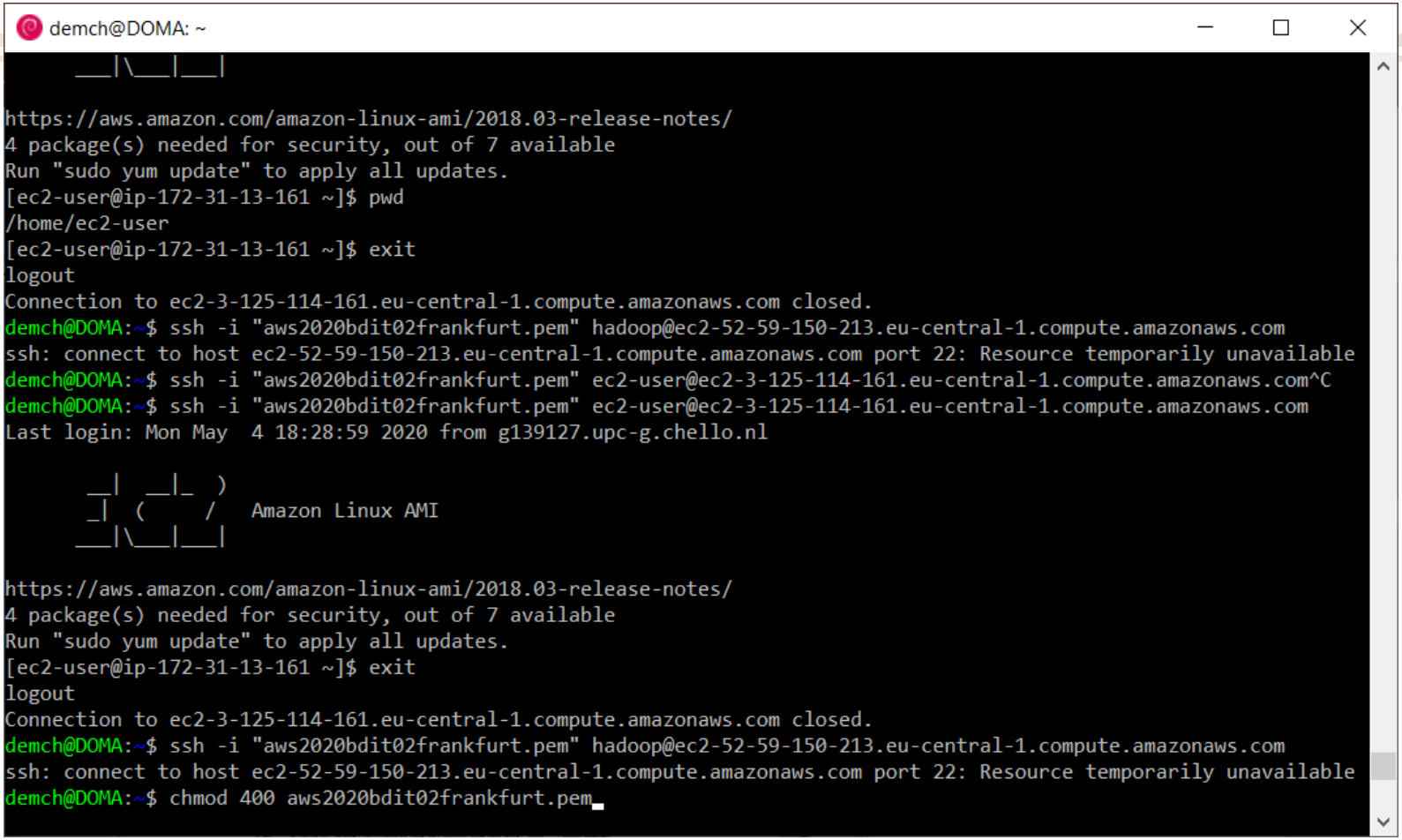
- Using PuTTY SSH client
- First, configure dynamic tunnel to remote cluster (RDS or EMR)
  - Port 8157
  - Using Source port and Dynamic radio button
  - Example:
    - D5147
    - Source L8088
      Destination Localhost:8088
    - Source L8888
      Destination Localhost:8888
- Configure other local ports
  - See EMR ports table
- This will create a secure tunnel by forwarding a port (the “destination port”) on the remote server to a port (the “source port”) on the local host (127.0.0.1 or localhost).

Configuring SSH tunnel on Windows for phpMyAdmin

https://docs.bitnami.com/virtual-machine/faq/get-started/access-phpmyadmin/

- In order to access phpMyAdmin via SSH tunnel, you need an SSH client. In the instructions below we have selected PuTTY, a free SSH client for Windows and UNIX platforms.
  - The first step is to configure PuTTY.
- Once you have your SSH client correctly configured and you have confirmed that you can successfully access your instance using SSH, you need to create an SSH tunnel in order to access phpMyAdmin. Follow these steps:
  - In the “Connection -> SSH -> Tunnels” section, add a new forwarded port by introducing the following values:
    - Source port: 8888
    - Destination: localhost:8888
    - Remember that if you are redirecting HTTP requests to the HTTPS port, you must use destination port 443 instead of 80.
  - This will create a secure tunnel by forwarding a port (the “destination port”) on the remote server to a port (the “source port”) on the local host (127.0.0.1 or localhost).
  - Click the “Add” button to add the secure tunnel configuration to the session. You’ll see the added port in the list of “Forwarded ports”.
- PuTTY configuration
  - In the “Session” section, save your changes by clicking the “Save” button.
  - Click the “Open” button to open an SSH session to the server. The SSH session will now include a secure SSH tunnel between the two specified ports.
- Access the phpMyAdmin console through the secure SSH tunnel you created, by browsing to http://127.0.0.1:8888/phpmyadmin.
- Log in to phpMyAdmin by using the following credentials:
  - Username: root
  - Password: application password. (Refer to our FAQ to learn how to find your application credentials).

Access VM Instance from Linux (Mac systems)



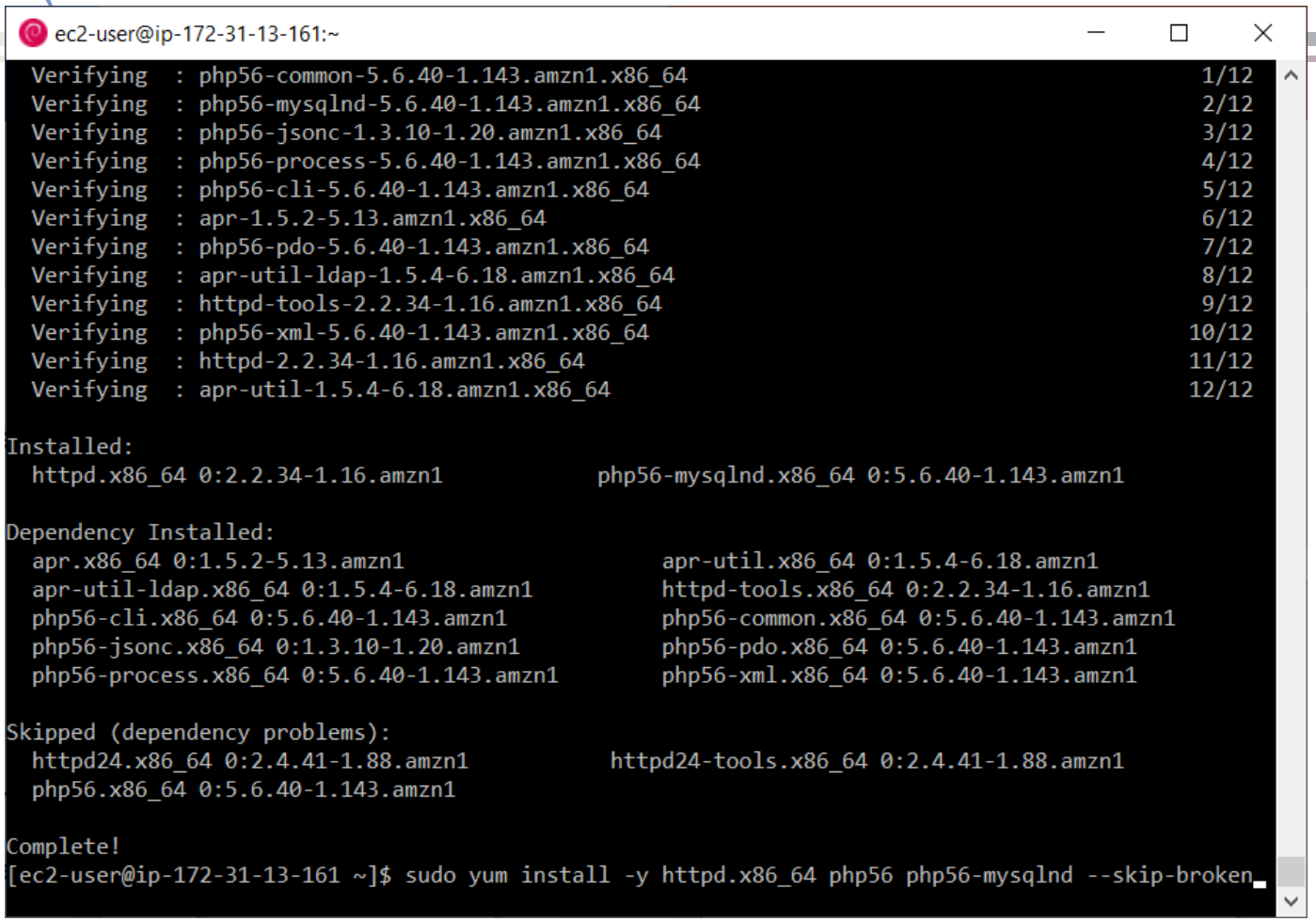
- Your key must not be publicly viewable for SSH to work. Use this command if needed: chmod 400 aws2020bdit02frankfurt.pem
- To establish a connection to the master node, type the following command. ssh -i ~/aws2020bdit02frankfurt.pem hadoop@ec2-52-59-150-213.eu-central-1.compute.amazonaws.com

Configuring VM instance to run Webserver

Configuring VM instance to run Webserver

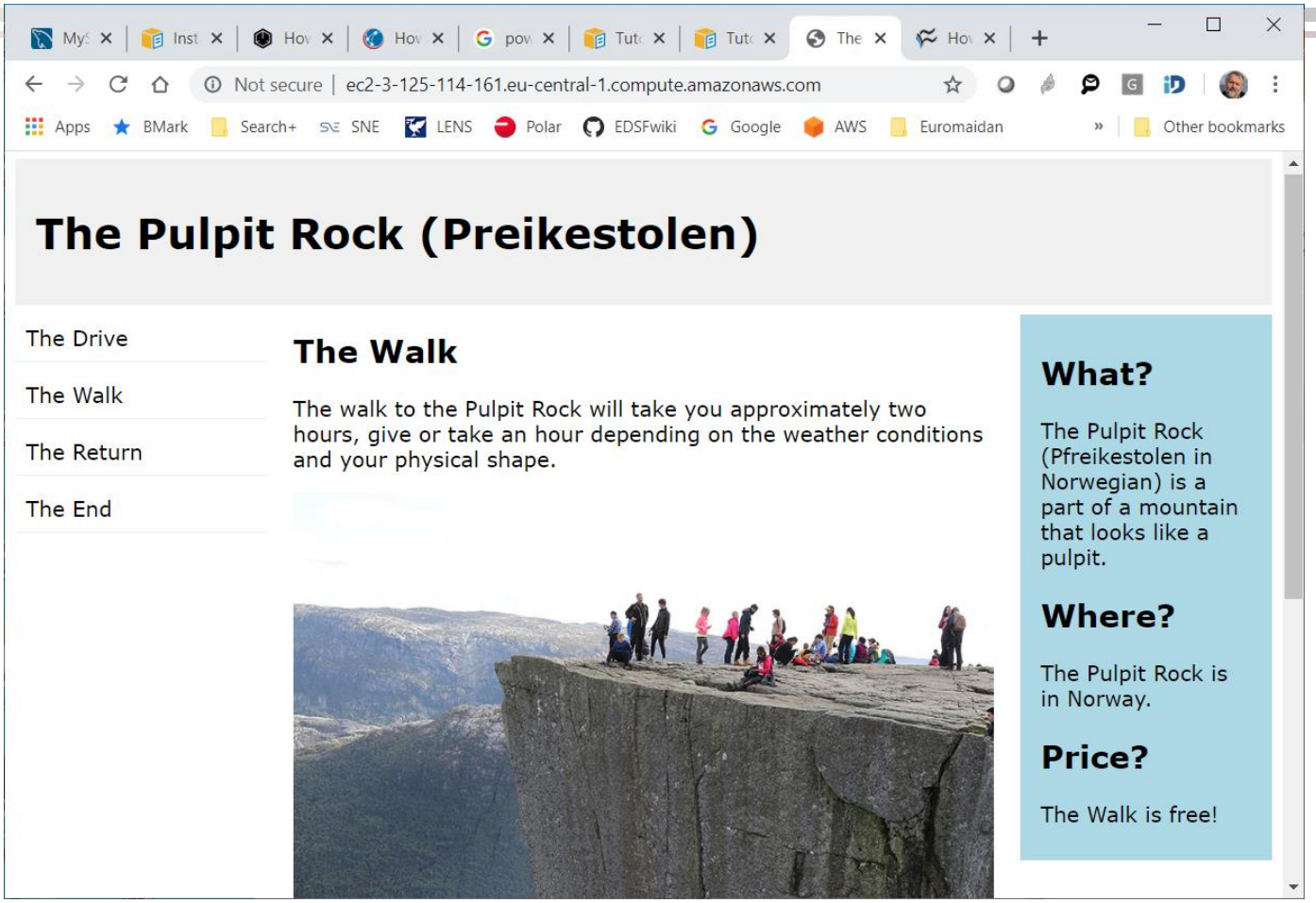
- Connect to ec2 instance with ssh: all commands below are for the VM instance
- Install Apache webserver with PHP (for testing)
  - **$ sudo yum update -y**
  - **$ sudo yum install -y httpd.x86_64 php56 php56-mysqlnd**
- Start the HTTP server
  - **$ sudo service httpd start**
- Open web browser to check if server is running
  - http:/xxxxxxxxx.compute-1.amazonaws.com – Use public DNS name for your ec2 instance
  - Note: Check that your security group allows inbound traffic for ports HTTP/80 and HTTPS/453. If no connection available, modify inbound rules: Select security group (default – launch-wizzard-1) > Actions > Edit inbound rules
- Auto start the webserver server with each restart of instance
  - **$ sudo chkconfig httpd on**
  - **$ chkconfig --list httpd** -- verify
- Setting permissions for the Apache web server
  - **$ sudo groupadd www**
  - **$ sudo usermod -a -G www ec2-user**
  - **$ sudo chown -R root:www /var/www**
  - **$ sudo chmod 2775 /var/www**
  - **$ find /var/www -type d -exec sudo chmod 2775 {} +**
  - **$ find /var/www -type f -exec sudo chmod 0664 {} +**

Installing HTTP and PHP packages



- sudo yum install -y httpd24 php72 mysql57-server php72-mysqlnd

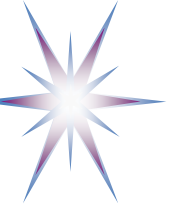
Uploading Webserver content



- Placing your test content in /var/www/html/
- Use SCP or SFTP configured with access key similar to PuTTY
-

Commands for controlling services

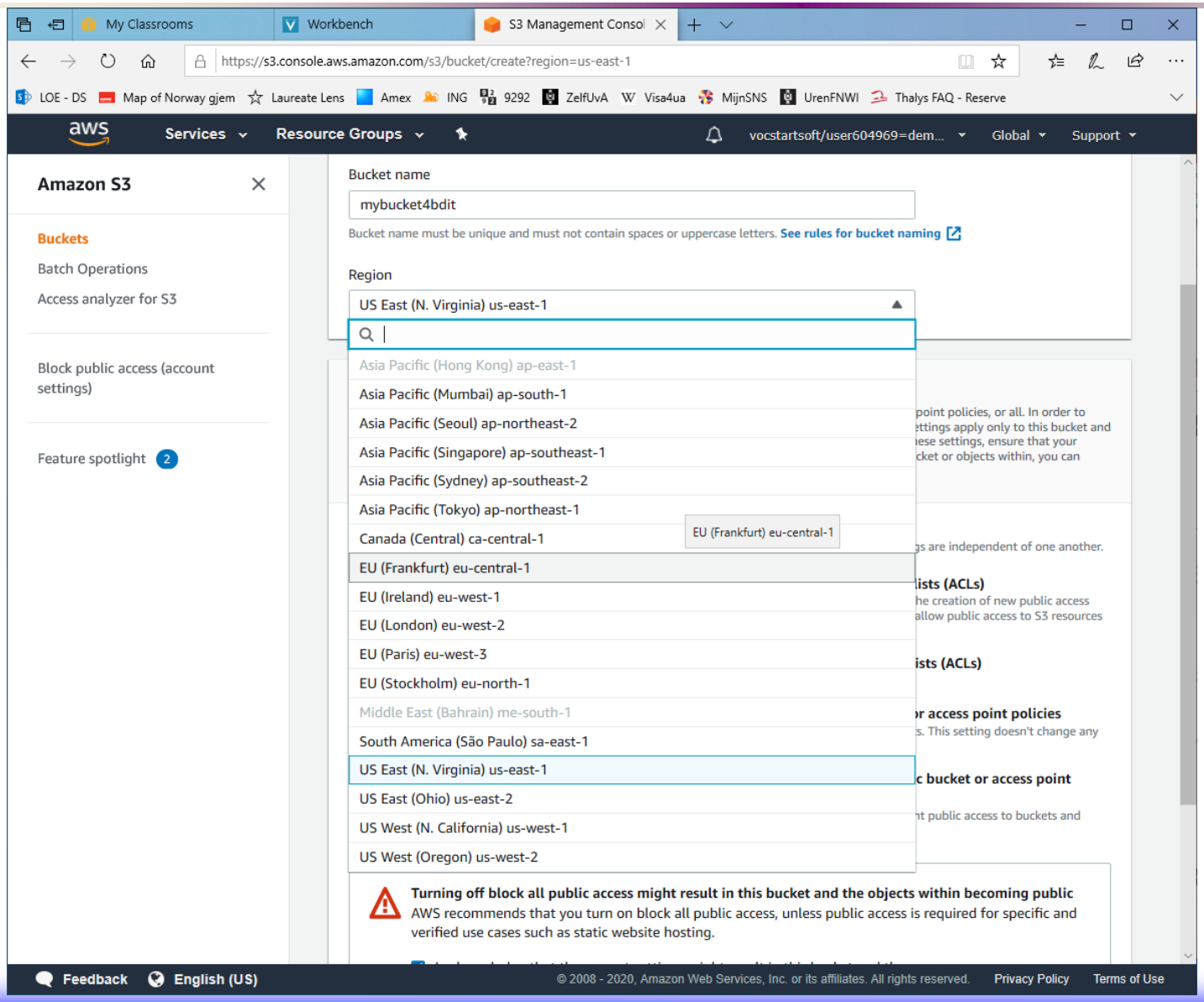
- Two officially adopted methods for controlling services:
- systemctl: CentOS, Ubuntu, Redhat, Fedora
  - sudo systemctl stop httpd
  - sudo systemctl start httpd
  - sudo systemctl restart httpd
- service: Debian
  - sudo service httpd start
    sudo service httpd stop
    sudo service httpd restart



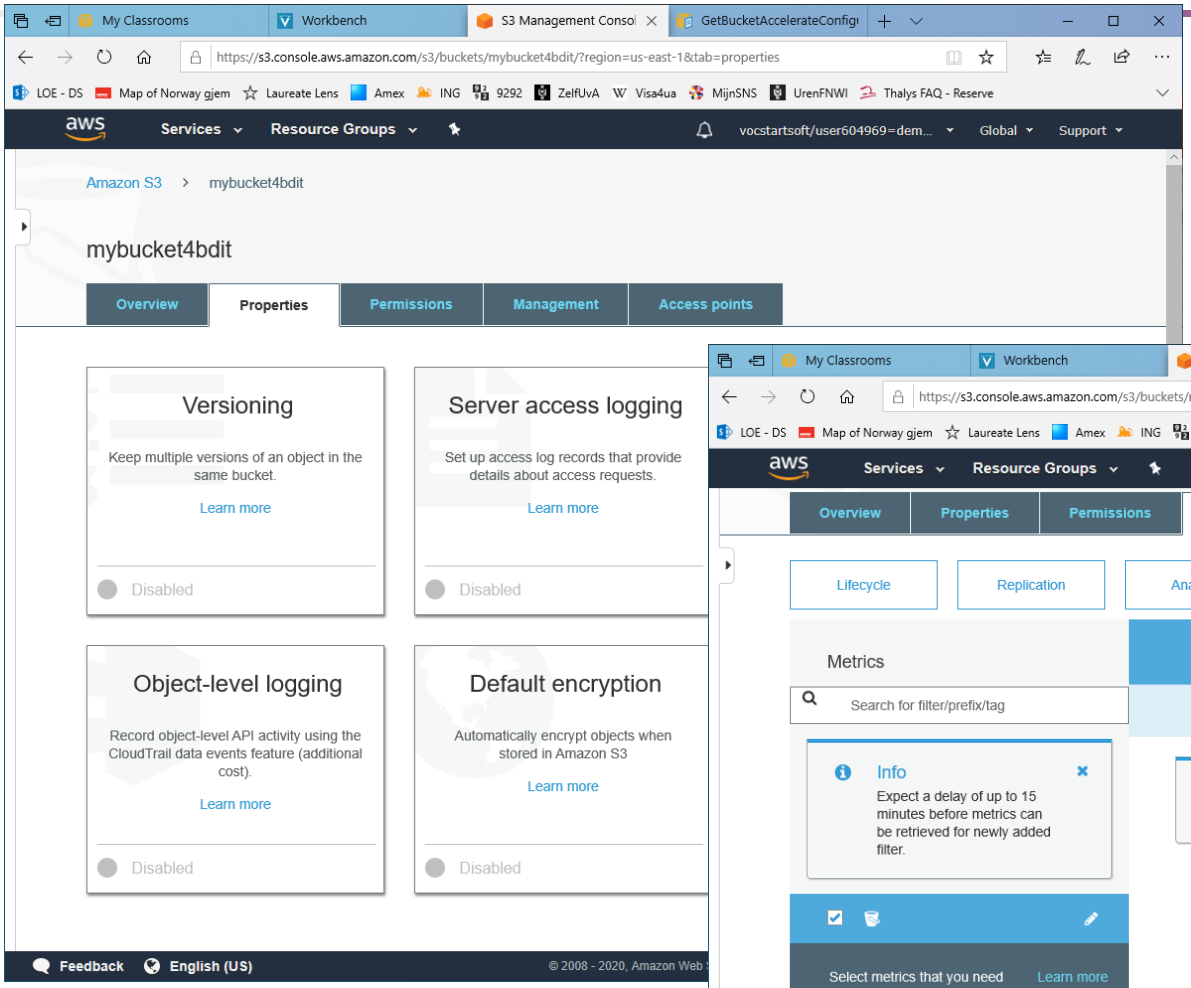
Working with S3 Storage

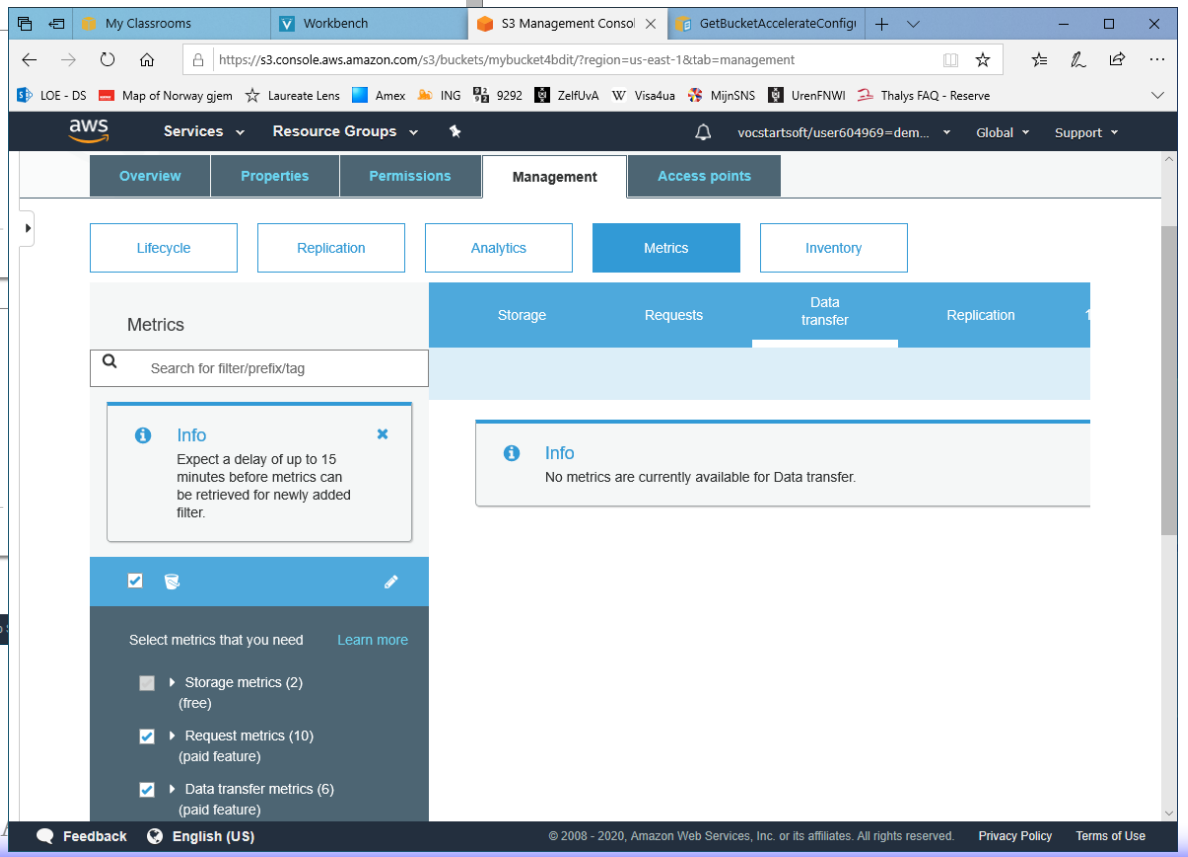
- Create Bucket
- Configure security
- Add monitoring

S3 Storage: Create bucket & Select Region

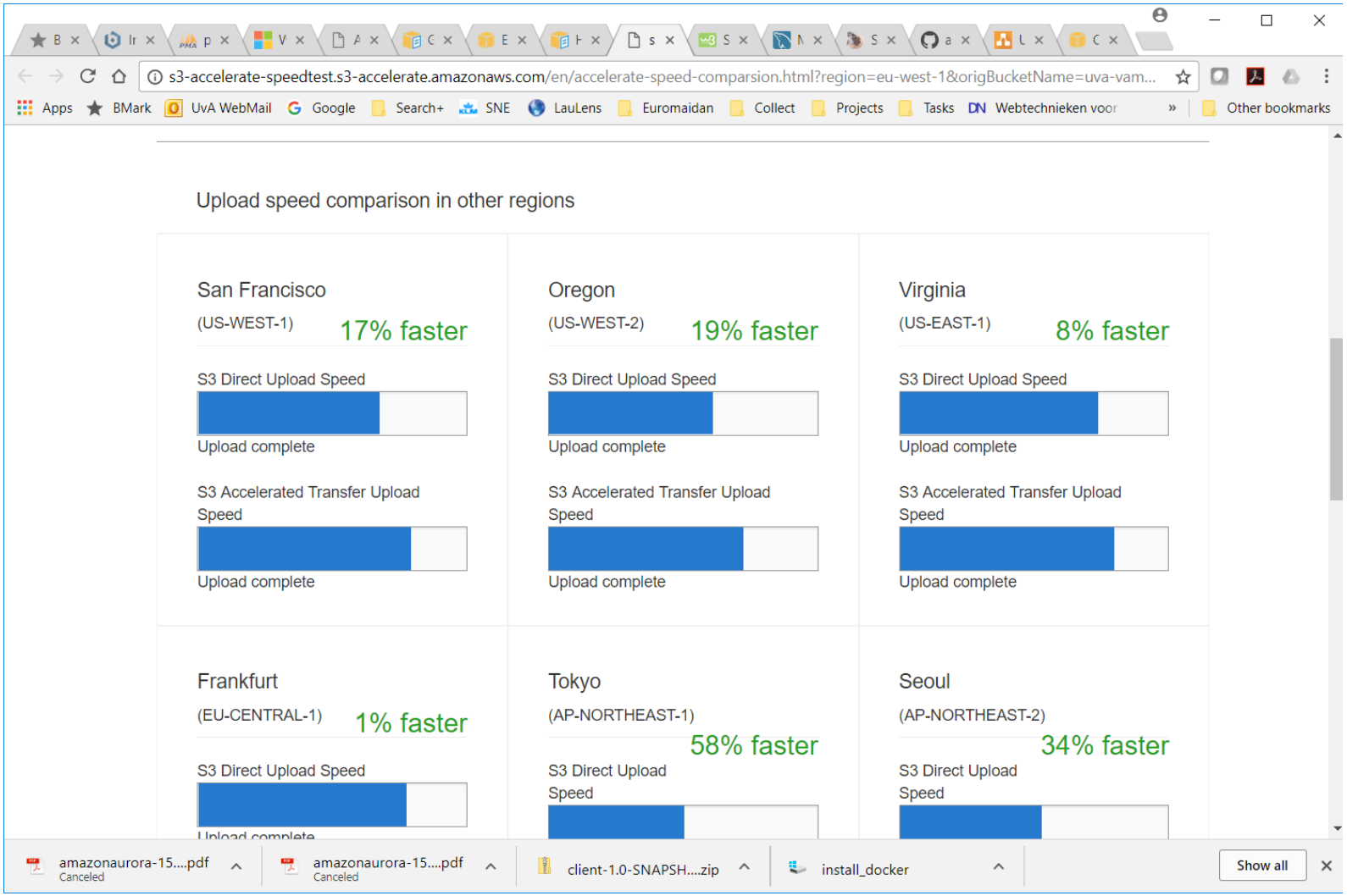


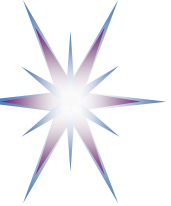
Configure S3 Bucket Properties and Management aspects (monitoring and metrics)





S3: Upload acceleration (example)



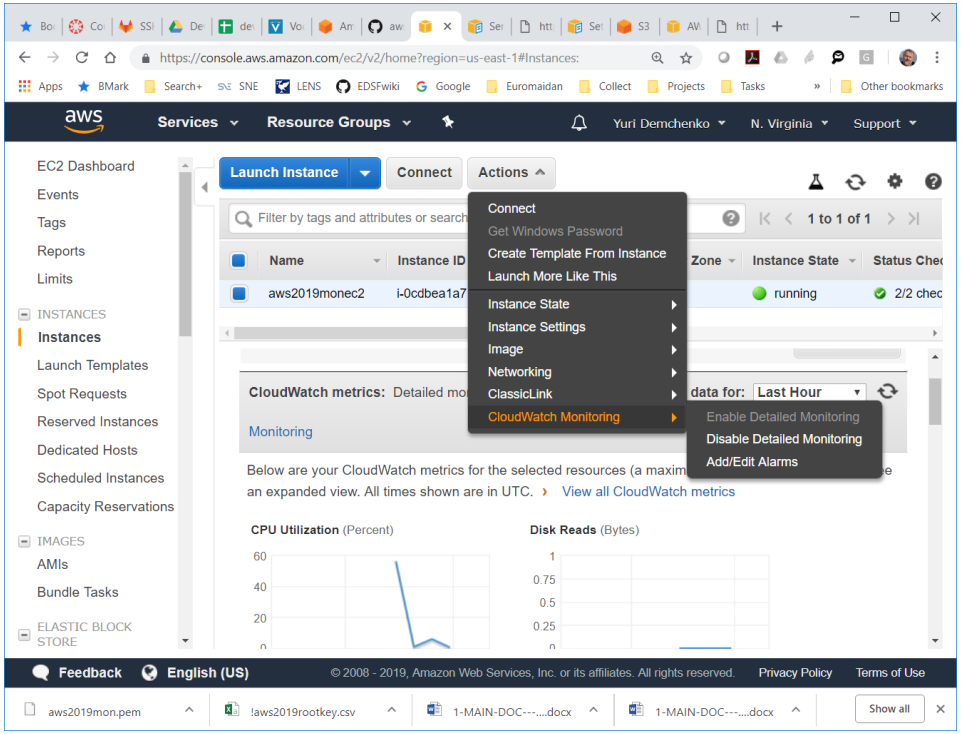


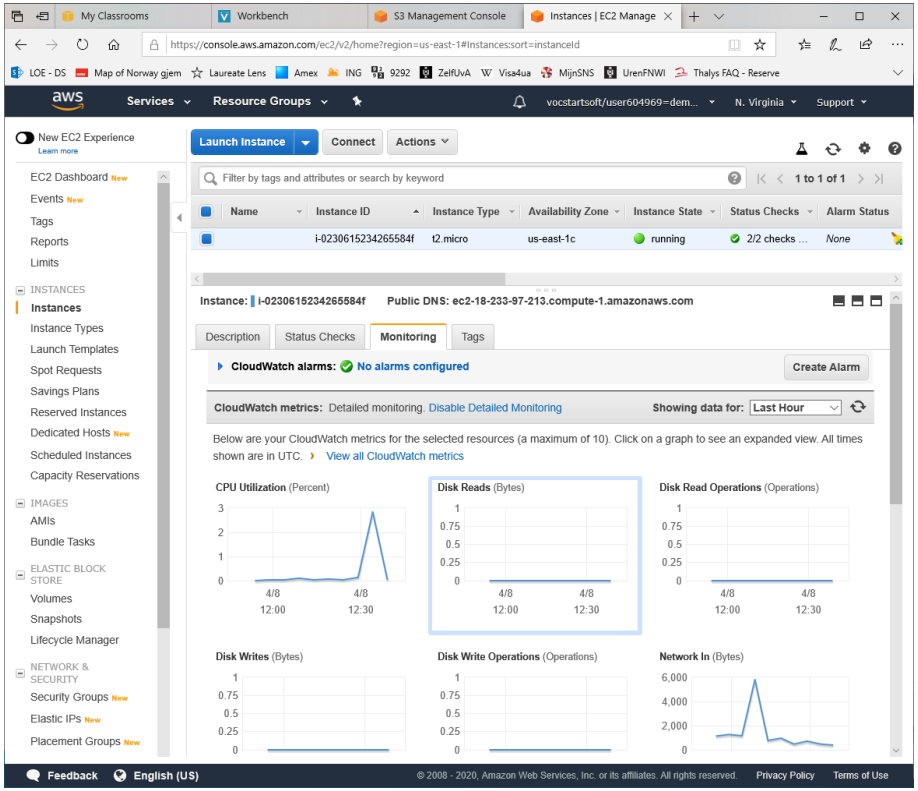
Adding Monitoring to EC2 and S3

- For EC2: Enable detailed monitoring for EC2 instance during deployment
- For S3: configure monitoring after S3 bucket is created

EC2 instances and Monitoring - Prepare

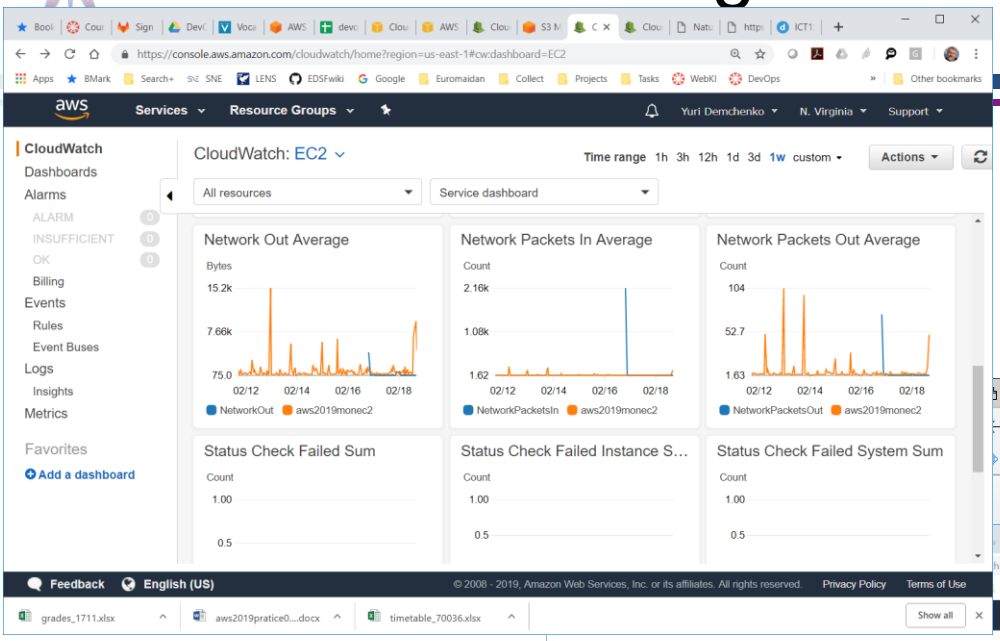
- Create/download Key Pair
  - aws2020mon: fingerprint c9:98:3b:a9:a4:ed:3e:da:94:51:8a:c0:80:0b:4b:10:5e:e4:7e:bf
- Create Instance: Wordpress by Bitnami
  - vpc-4fcc132a – subnet-85003ead | us-east-1a

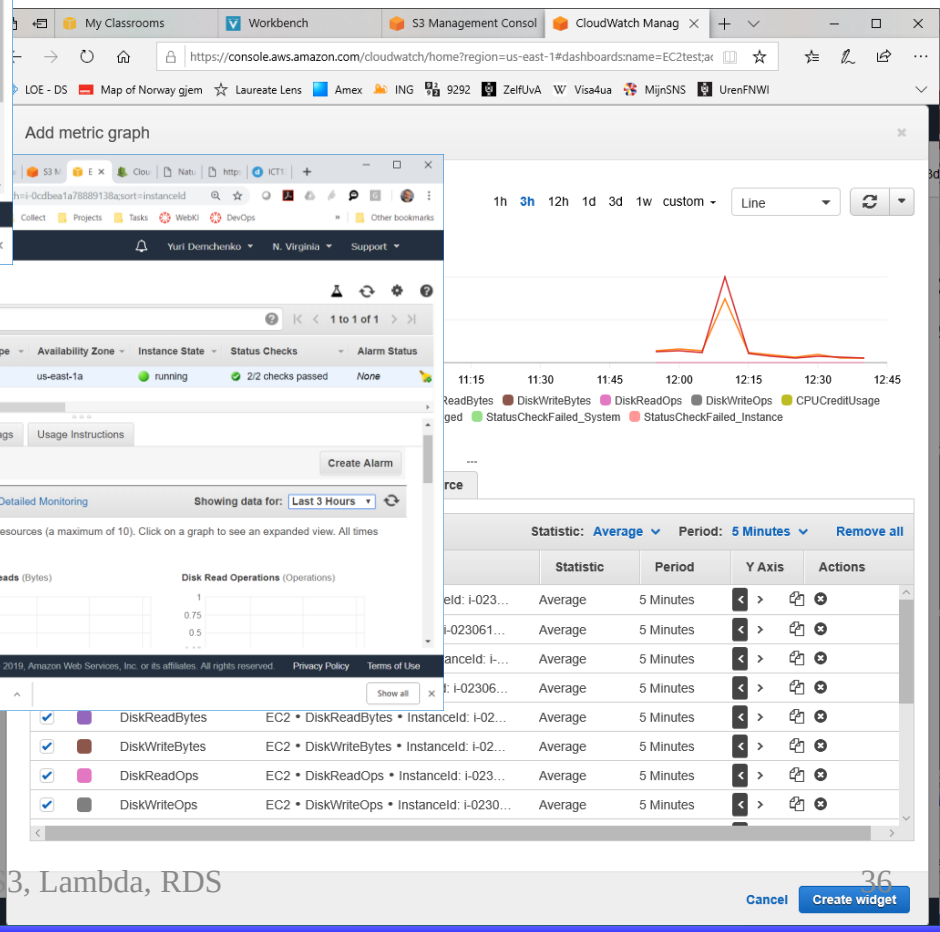


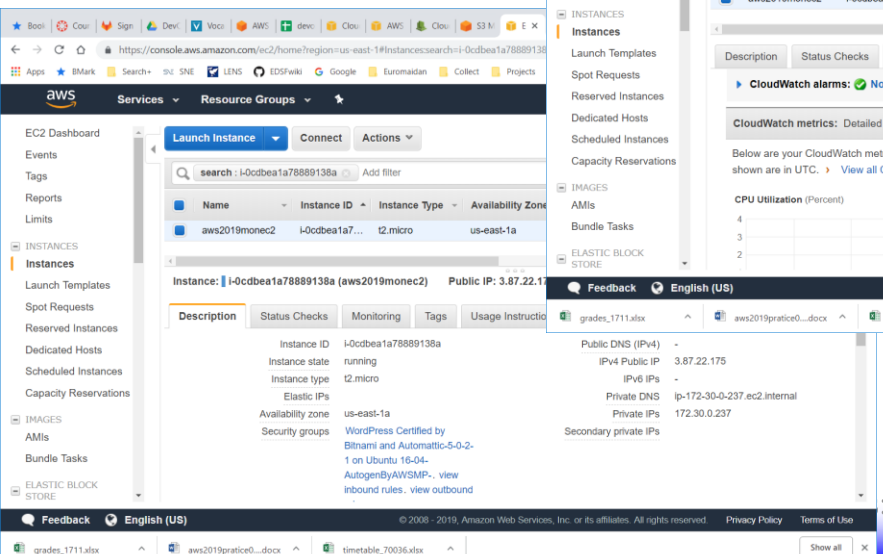


EC2 Monitoring Dashboard

- Configure EC2 metrics – Many
- Create dashboard





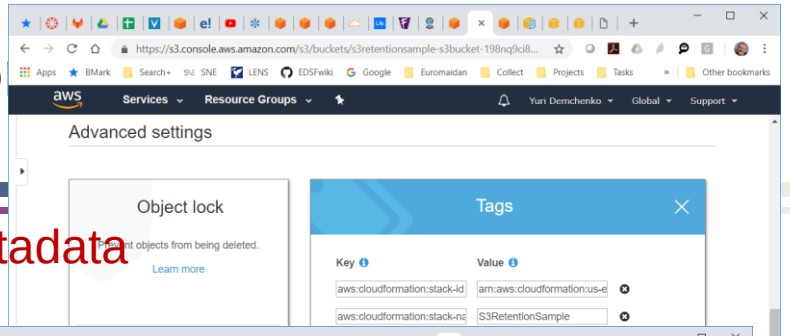


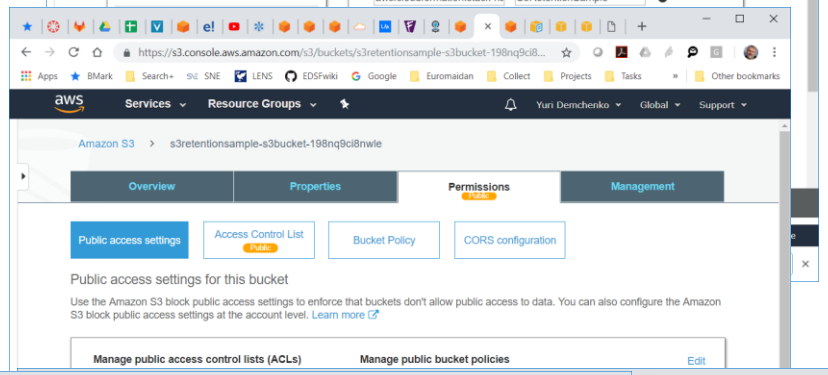
S3 Bucket properties and monito

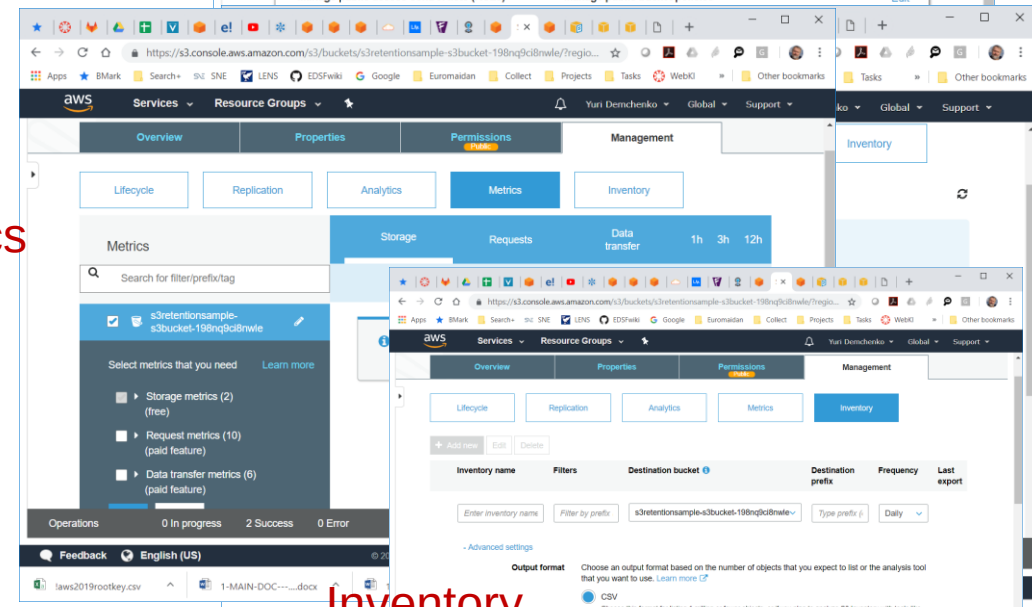
Advance setting

- Name arn:aws:s3:::s3retentionsample-s3bucket-198nq9ci8nwle
- s3retentionsample-s3bucket-198nq9ci8nwle

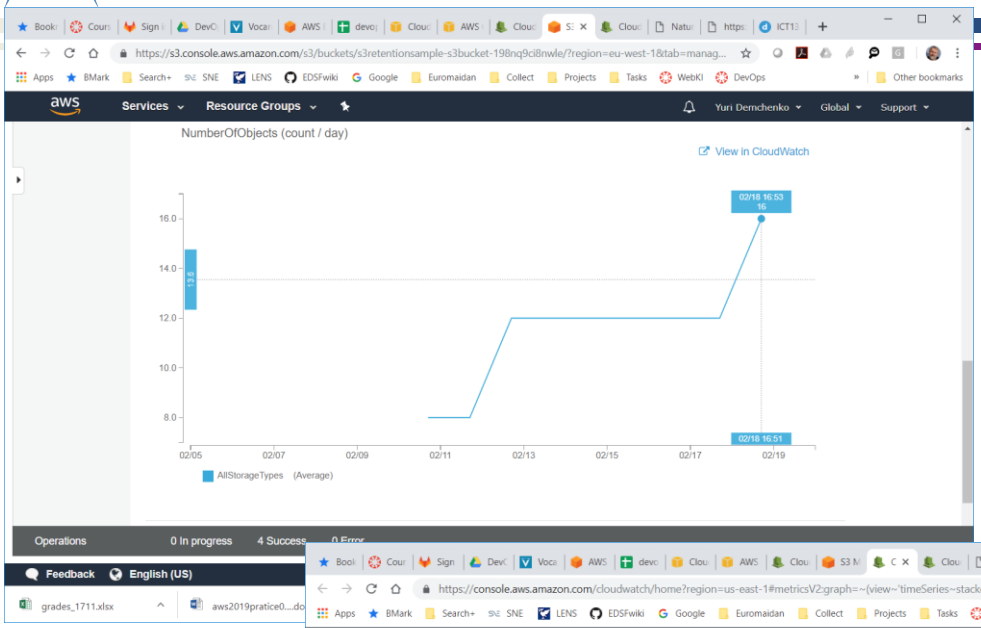
- Tagging: instances to track and clean
- Permissions/access control
- Management
  - Lifecycle rules e.g. move to Gleischer
  - Transition – expiration
- Analytics
- Metrics
  - Storage metrics (Free)
  - Request metrics (paid)
  - Data transfer (paid)
- Inventory

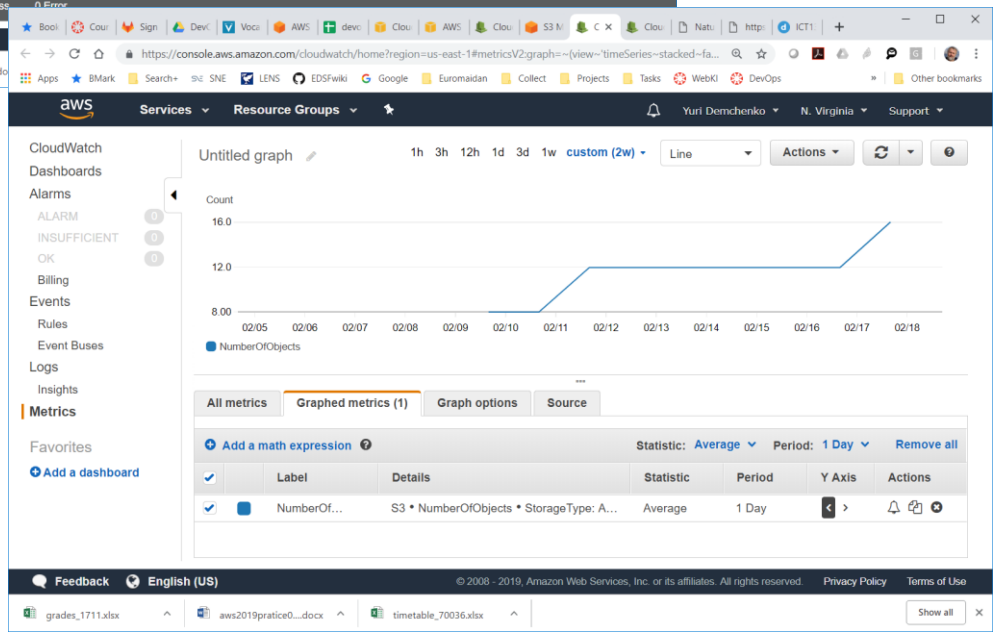




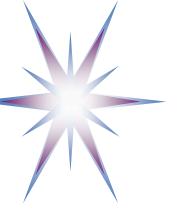


S3 metrics and Dashboard metrics



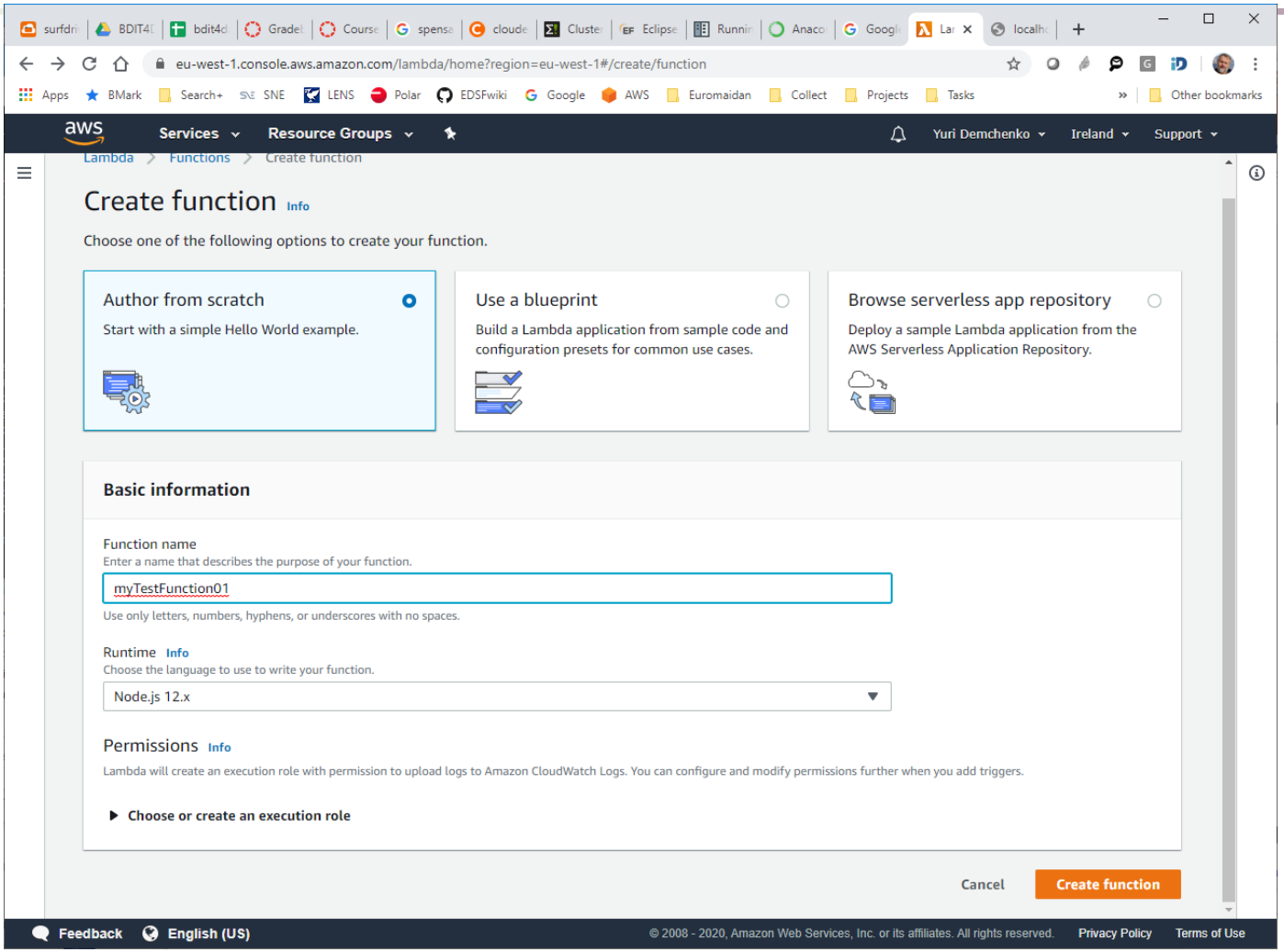


- Be aware of delay in collecting and displaying metrics

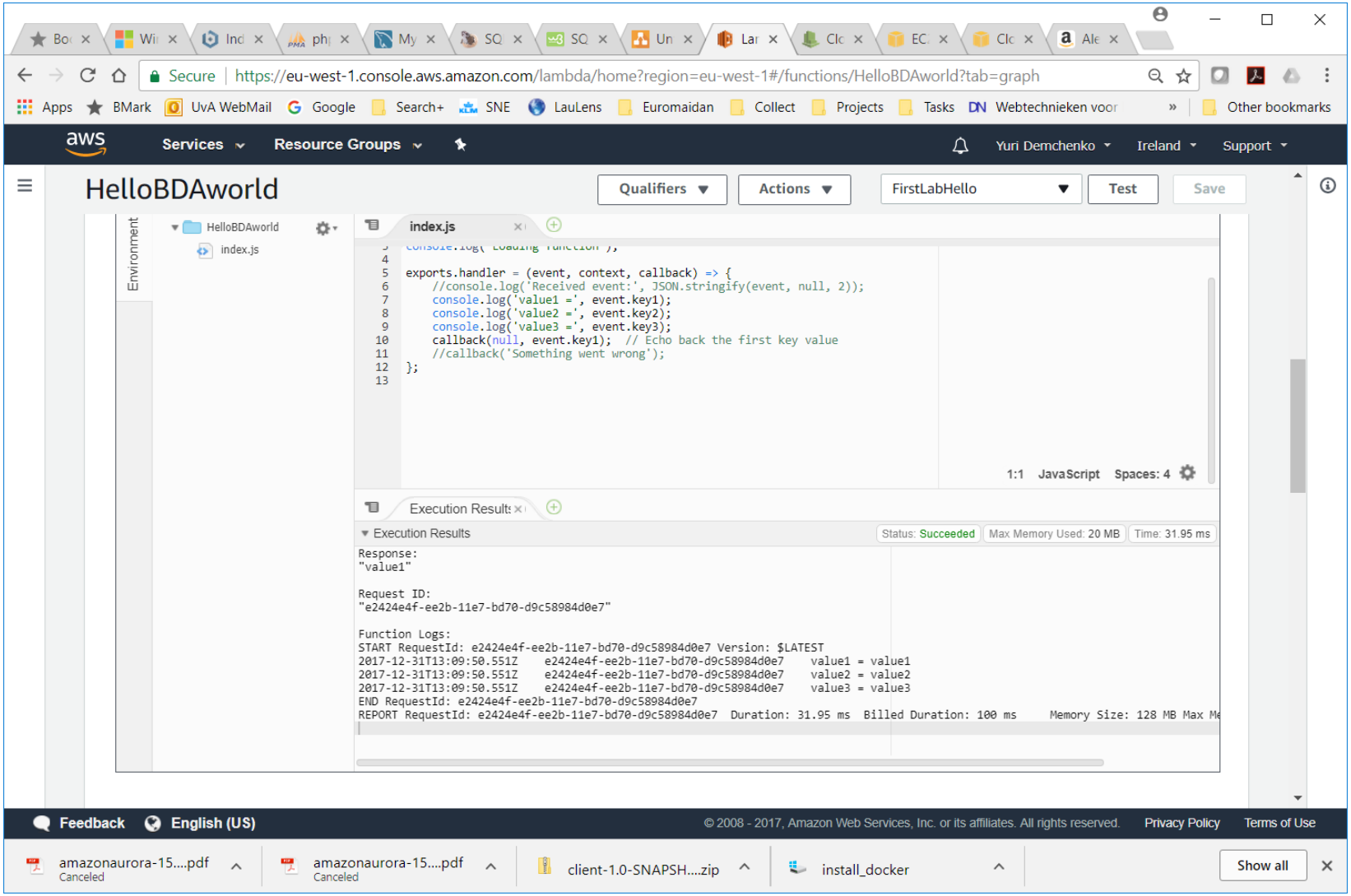


Amazon Lambda Serverless Computing

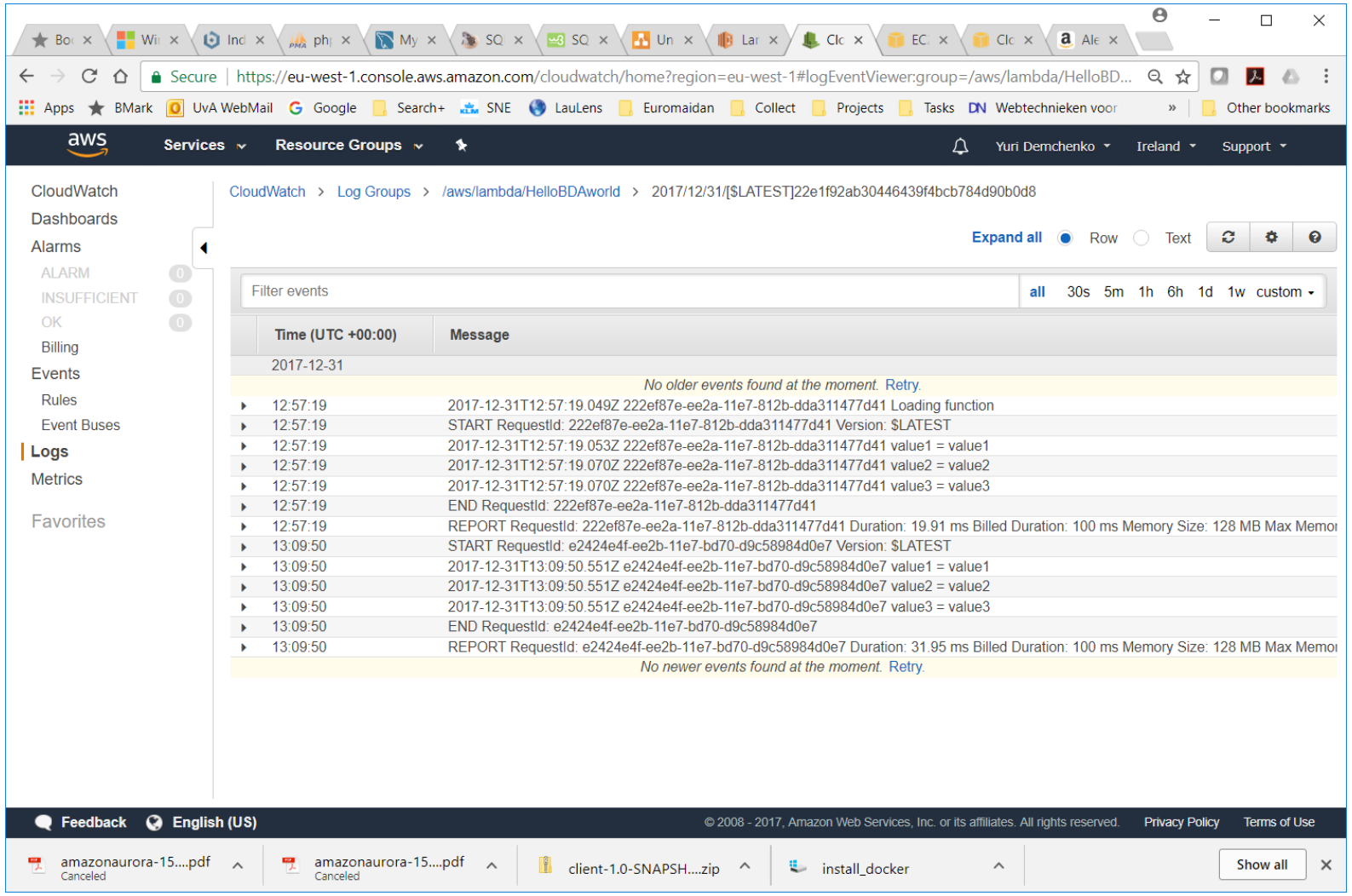
Starting with AWS Lambda (for Serverless Computing)



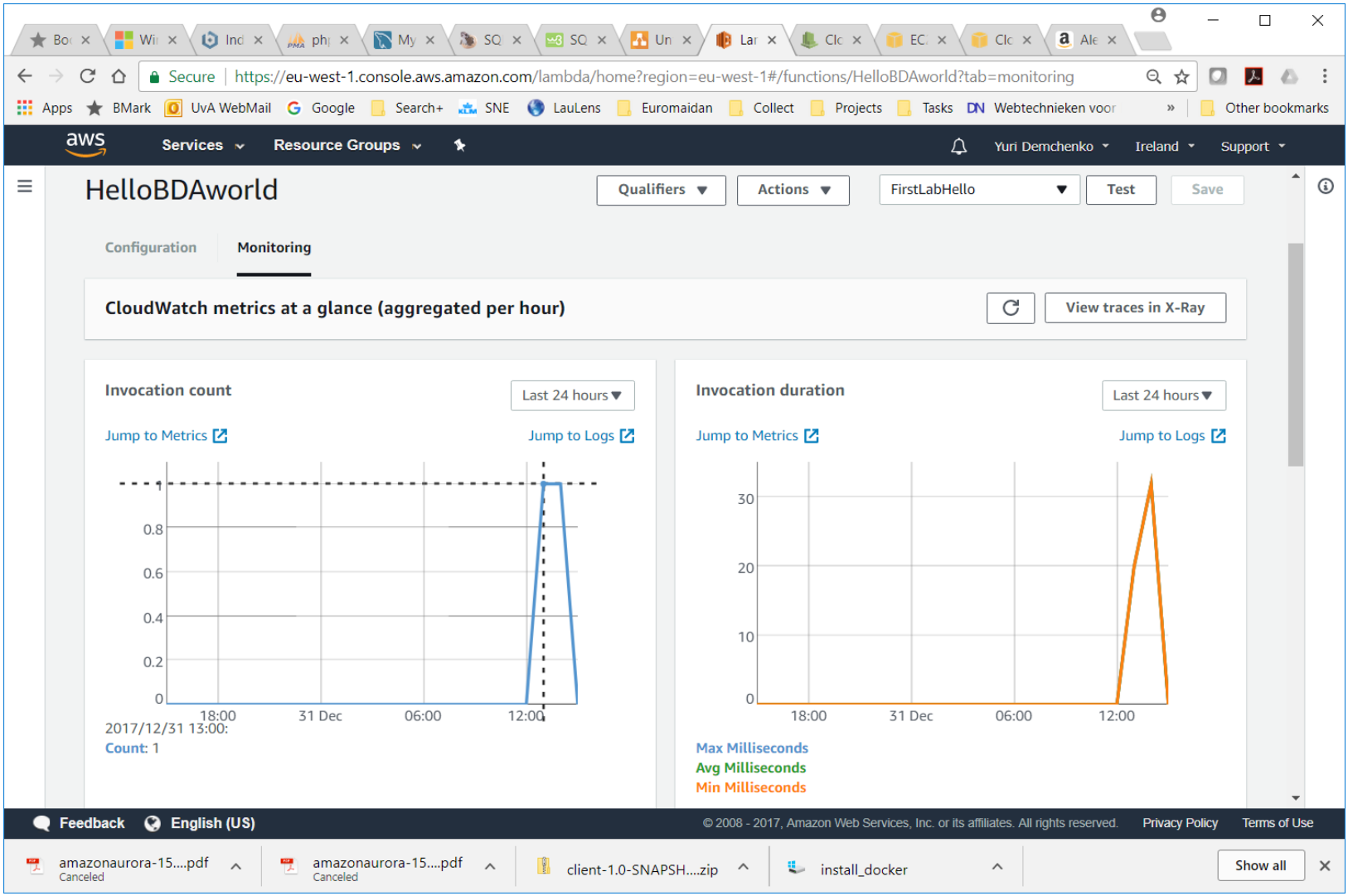
Lambda FirstLabHello (1)

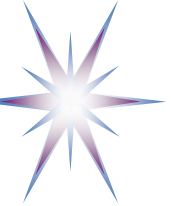


Lambda FirstLabHello (2)



Lambda FirstLabHello (3) – CloudWatch Monitoring



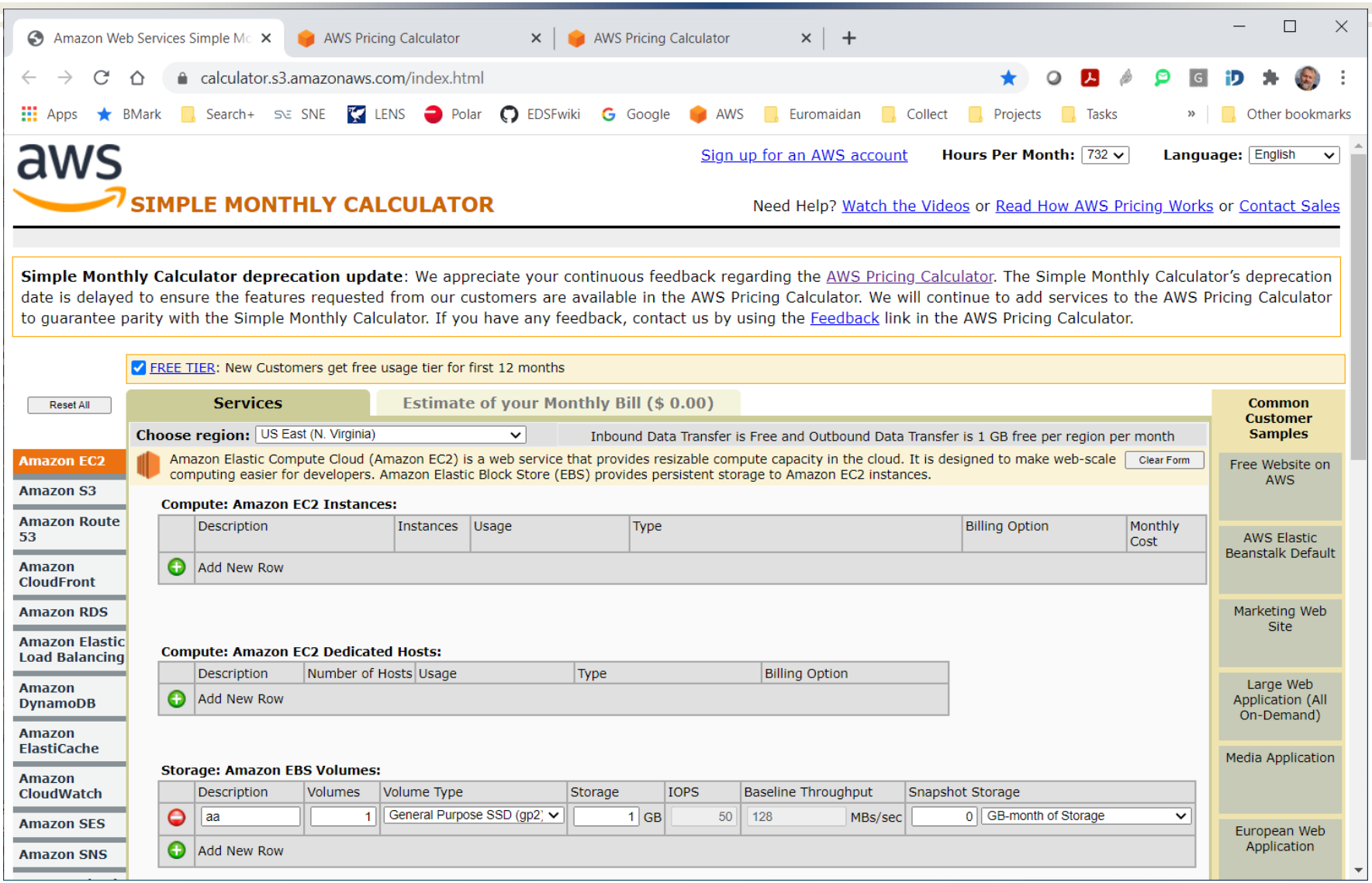


AWS Pricing Calculator

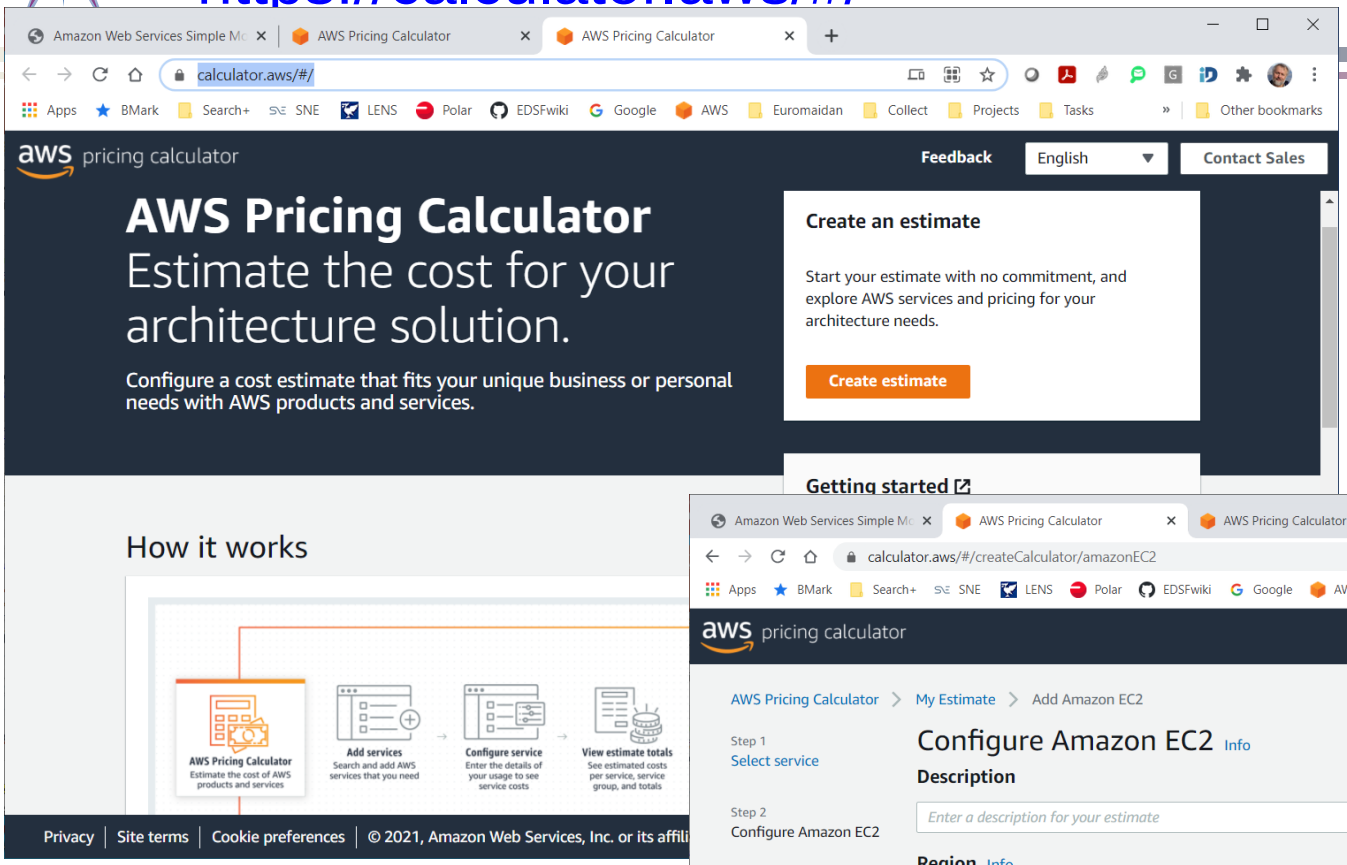
- Be aware of cloud cost
- Use AWS Pricing Calculator https://calculator.aws/#/

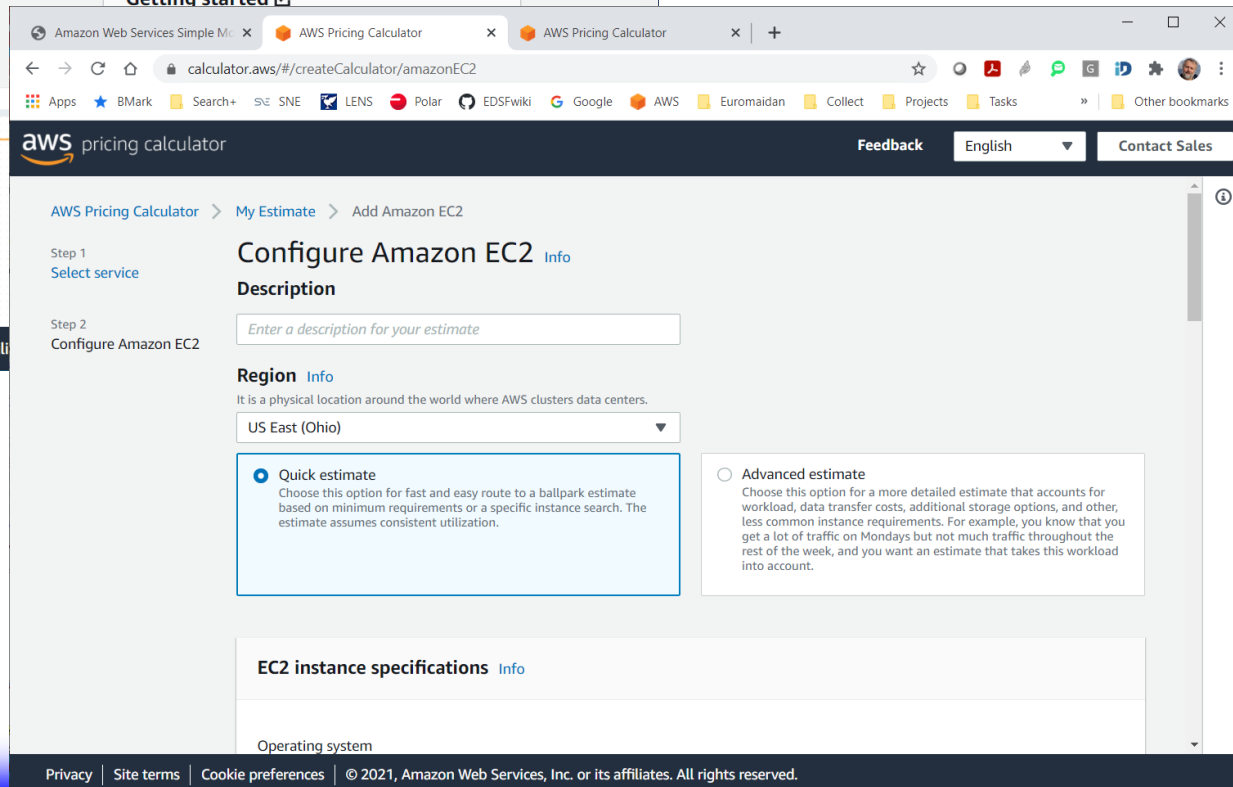
Simple Monthly Calculator – To be deprecated soon

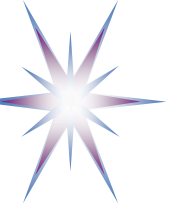
https://calculator.s3.amazonaws.com/index.html



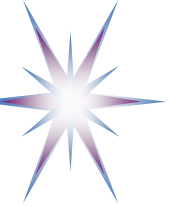
New AWS Pricing Calculator - https://calculator.aws/#/







Automating AWS resources deployment with CloudFormation

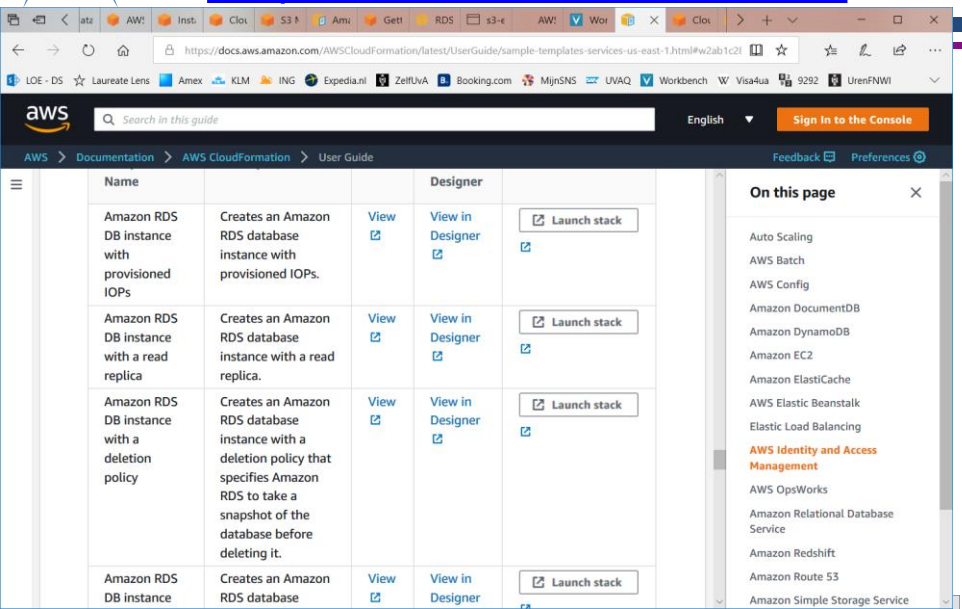


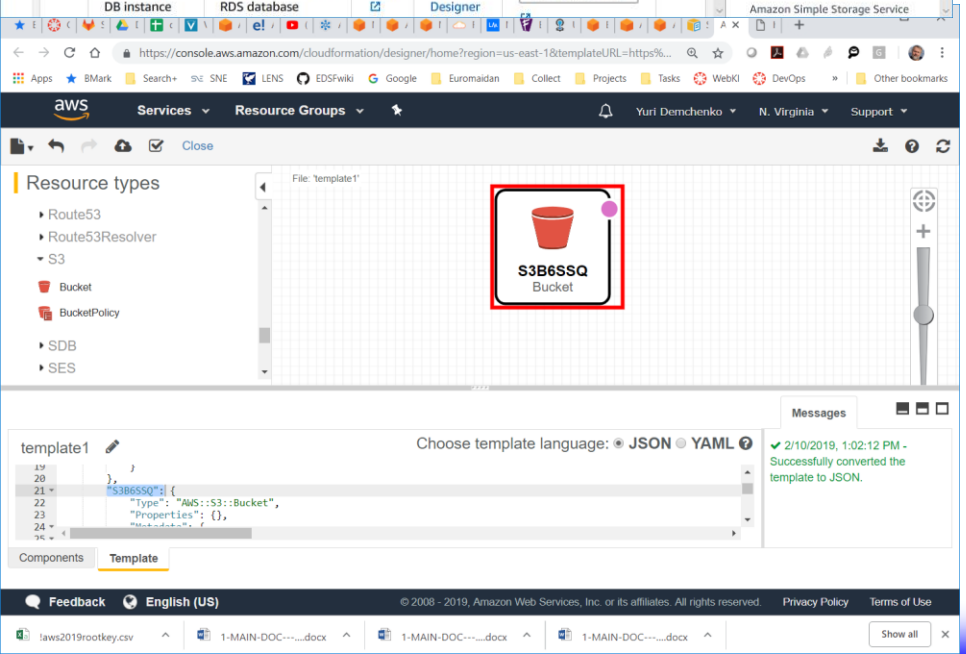
Using CloudFormation for Resource Deployment

- CloudFormation is the internal cloud resources configuration and automated provisioning tool
- CloudFormation template describes configuration of a single or multiple resources
- CloudFormation template is re-usable but not composable

Creating S3 bucket via CF template

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/sample-templates-services-us-east-1.html






```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template
S3_Website_Bucket_With_Retain_On_Delete: Sample template
showing how to create a publicly accessible S3 bucket
configured for website access with a deletion policy of
retain on delete. **WARNING** This template creates an S3
bucket that will NOT be deleted when the stack is deleted.
You will be billed for the AWS resources used if you create
a stack from this template.",

  "Resources" : {
    "S3Bucket" : {
      "Type" : "AWS::S3::Bucket",
      "Properties" : {
        "AccessControl" : "PublicRead",
        "WebsiteConfiguration" : {
          "IndexDocument" : "index.html",
          "ErrorDocument" : "error.html"
        }
      },
      "DeletionPolicy" : "Retain"
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Value" : { "Fn::GetAtt" : [ "S3Bucket", "WebsiteURL"
] },
      "Description" : "URL for website hosted on S3"
    },
    "S3BucketSecureURL" : {
      "Value" : { "Fn::Join" : [ "", [ "https://", {
"Fn::GetAtt" : [ "S3Bucket", "DomainName" ] } ] ] },
      "Description" : "Name of S3 bucket to hold website
content"
    }
  }
}
```

Create -> Upload Files -> Default properties for files

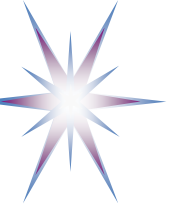
Closer look at S3 bucket template

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template S3_Website_Bucket_With_Retain_On_Delete: Sample template
showing how to create a publicly accessible S3 bucket configured for website access with a deletion policy of
retain on delete. **WARNING** This template creates an S3 bucket that will NOT be deleted when the stack is
deleted. You will be billed for the AWS resources used if you create a stack from this template.",

  "Resources" : {
    "S3Bucket" : {
      "Type" : "AWS::S3::Bucket",
      "Properties" : {
        "AccessControl" : "PublicRead",
        "WebsiteConfiguration" : {
          "IndexDocument" : "index.html",
          "ErrorDocument" : "error.html"
         }
      },
      "DeletionPolicy" : "Retain"
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Value" : { "Fn::GetAtt" : [ "S3Bucket", "WebsiteURL" ] },
      "Description" : "URL for website hosted on S3"
    },
    "S3BucketSecureURL" : {
      "Value" : { "Fn::Join" : [ "", [ "https://", { "Fn::GetAtt" : [ "S3Bucket", "DomainName" ] } ] ] },
      "Description" : "Name of S3 bucket to hold website content"
    }
  }
}
```



Setting up your Cloud Development Environment

- Python and PIP
- Windows PowerShell
- Windows Extension for Linux
- AWS CLI
  - On Windows and PowerShell
  - On Linux

Python and PIP

- Ubuntu 18.04 ships with Python 3, as the default Python installation.
- Complete the following steps to install pip (pip3) for Python 3:
- Start by updating the package list using the following command:
  sudo apt update
- Use the following command to install pip for Python 3:
  sudo apt install python3-pip
  The command above will also install all the dependencies required for building Python modules.
- Once the installation is complete, verify the installation by checking the pip version:
  pip3 --version

Installing Python 3.8 on Linux/Debian/Ubuntu

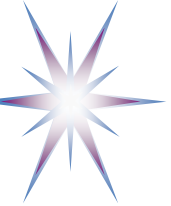
- **How to Install Python 3.8 on Ubuntu 18.04**
- https://linuxize.com/post/how-to-install-python-3-8-on-ubuntu-18-04/
- Check
  - $ python3.8 –version
- **Installing pip for Python 3**
  - sudo apt install python3-pip
- Check
  - Pip3 –version
  - pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)

Windows PowerShell and WSL

- https://docs.microsoft.com/en-us/windows/wsl/faq
- Available in Microsoft Store
- Install on Windows 10: From PowerShell
  - Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
- Run WSL
  - **Set up a new Linux user account**
- Update and upgrade packages
  - sudo apt update && sudo apt upgrade
-
- Access local machine's filesystem from within the Linux Bash shell
  - Local drives mounted under the /mnt folder, e.g. C: drive is mounted under /mnt/c:

AWS CLI

- Download the AWS CLI MSI installer for Windows (64-bit) at https://awscli.amazonaws.com/AWSCLIV2.msi
- To confirm version
  - > aws –version
  - > aws-cli/2.0.6 Python/3.7.4 Windows/10 botocore/2.0.0
- Usage:
  - aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
- To see help text, you can run:
  - aws help
  - aws <command> help
  - aws <command> <subcommand> help



AWS CLI on Linux

- Prerequisite: Python 3+ and python-pip
- From Windows Extension for Linux Debian
  - sudo apt-get install python-pip python-dev build-essential
- $ sudo apt install awscli
  - Option: sudo pip install awscli
  - Other: sudo pip install –-upgrade –-user awscli
- Check
  - aws –version
  - aws-cli/1.16.113 Python/3.7.3 Linux/4.4.0-18362-MIcrosoft botocore/1.12.103

Copy key to working directory

- To copy a file from your current directory into another directory called /tmp/, enter:
  $ cp filename /tmp
  $ ls /tmp/filename
  $ cd /tmp
  $ ls
  $ rm filename