Component-based Software Development

Amazon Web Services (AWS)
EC2, S3

Dr. Vinod Dubey
SWE 645
George Mason University

Amazon Web Services – useful links

- **Introduction to Amazon Web Services (AWS)**
  - https://aws.amazon.com/ - The link to create your AWS account

Other useful links:

https://www.youtube.com/watch?v=98ya1LiEU00&t=15s

http://aws.amazon.com/s3/

https://aws.amazon.com/training/intro_series/

http://aws.amazon.com/ec2/

Amazon Web Services (Amazon AWS)

Amazon AWS

- Amazon Web Services is a subsidiary of Amazon.com
- Provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis
- Offers compute power, database storage, content delivery and other functionality to help businesses scale and grow

Source: https://en.wikipedia.org/wiki/Amazon_Web_Services

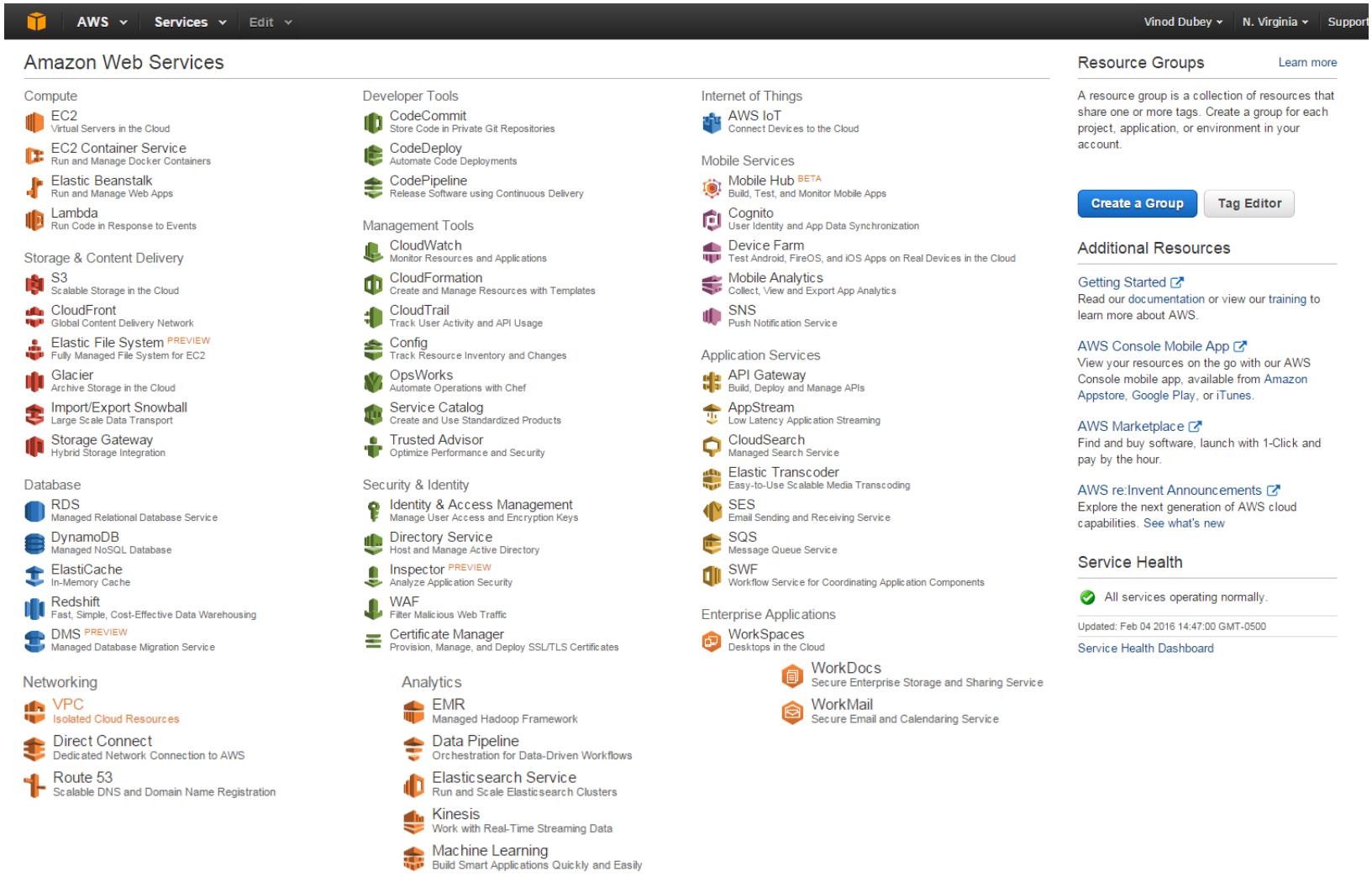
Amazon AWS

- As of 2021, the AWS Cloud spans 84 Availability Zones within 26 geographic regions around the world, with announced plans for 24 more Availability Zones and 8 more AWS Regions in Australia, Canada, India, Israel, New Zealand, Spain, Switzerland, and United Arab Emirates (UAE).

Amazon AWS

- 2006 – Amazon S3 first deployed in the spring, Amazon EC2 in the fall
- 2008 – Elastic Block Storage available.
- 2009 – Relational Database Service
- 2012 – DynamoDB
- ……
- ……
- ……

Amazon AWS – list of services – this is how AWS console looked in the beginning!



Amazon S3

- **Amazon Simple Storage Service (Amazon S3) provides durable, highly-scalable object storage**
  - **Designed for 99.999999999% durability and 99.99% availability**
- **You pay only for the storage you actually use.**
  - There is no minimum fee and no setup cost.
- **Easy to use, with a simple web interface to store and retrieve any amount of data from anywhere on the web.**

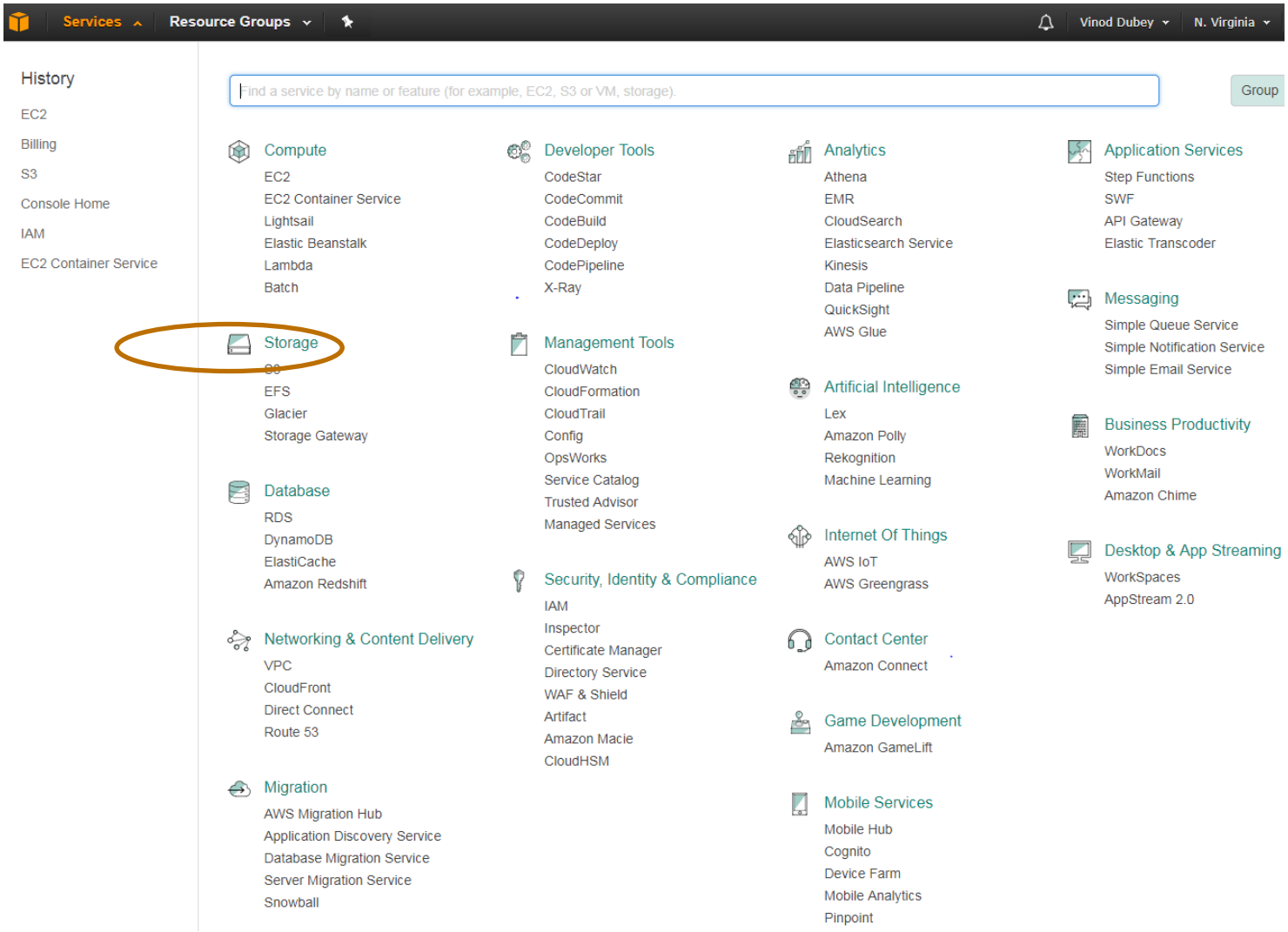
Accessing Amazon S3

- **Amazon S3 can be accessed in a number of ways:**
  - Amazon S3 provides standard-based REST API to programmatically store, retrieve, and manage your data
  - Software Development Kit (SDK), which wraps the underlying REST API
  - Command line interface
  - AWS Management Console, which is an easy to use simple Web interface

Amazon S3 Buckets

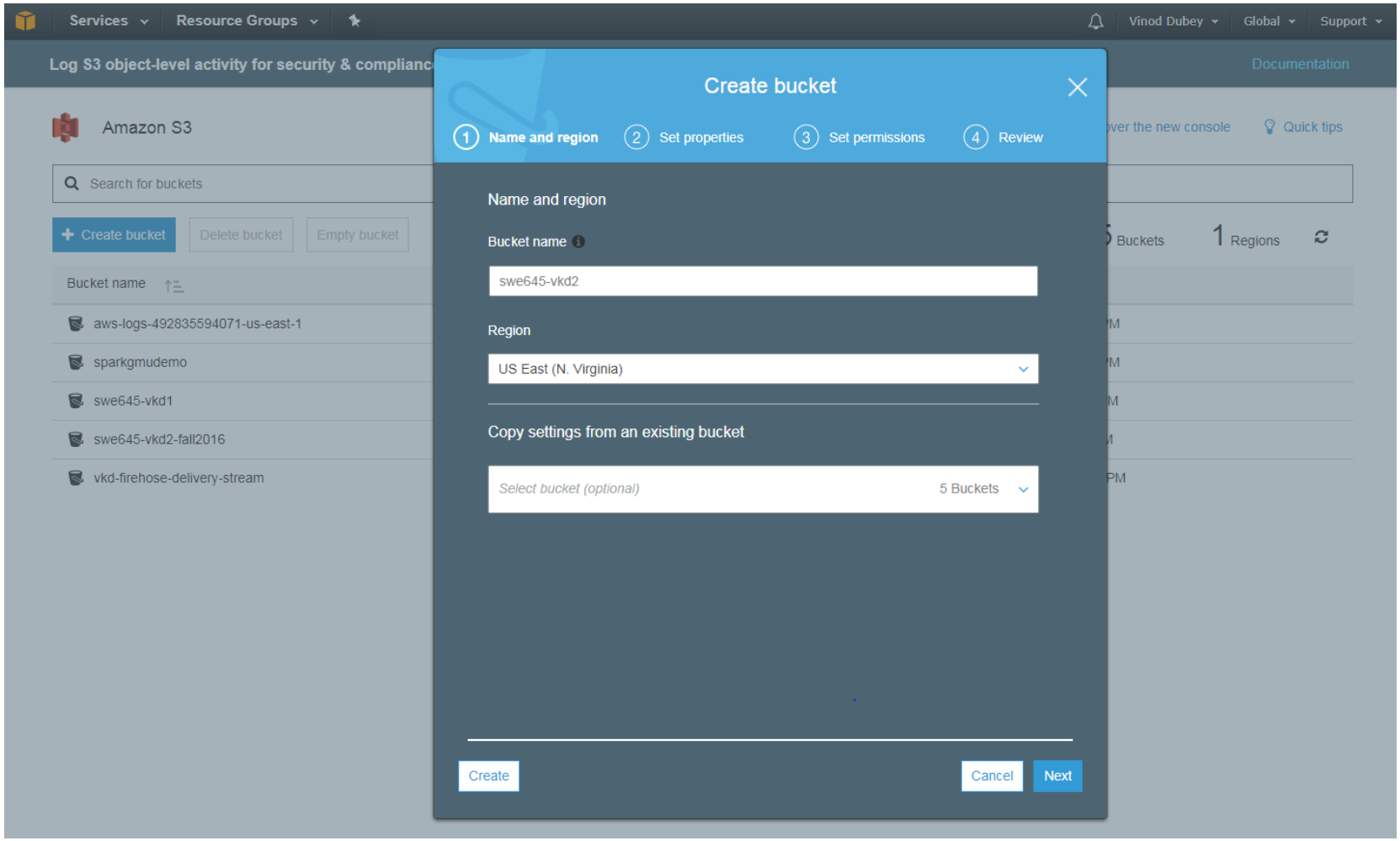
- **Amazon S3 stores data as objects and objects are stored within folders that are called "buckets"**
- **To store an object in Amazon S3, you upload the file, you want to store, to a bucket**
  - You can set permissions on uploaded files so they are private until you want to share them
- **A bucket is a container for objects**
  - You can have one or more buckets
  - For each bucket you can control access to the bucket, such as who can create, delete, and list objects in that bucket
  - A bucket can hold **any number** of **objects**, which are files of up to 5TB.
  - A bucket has a **name** (using lower case) that must be **globally unique**.

Click on the “Amazon S3” icon under “Storage” on AWS Management Console

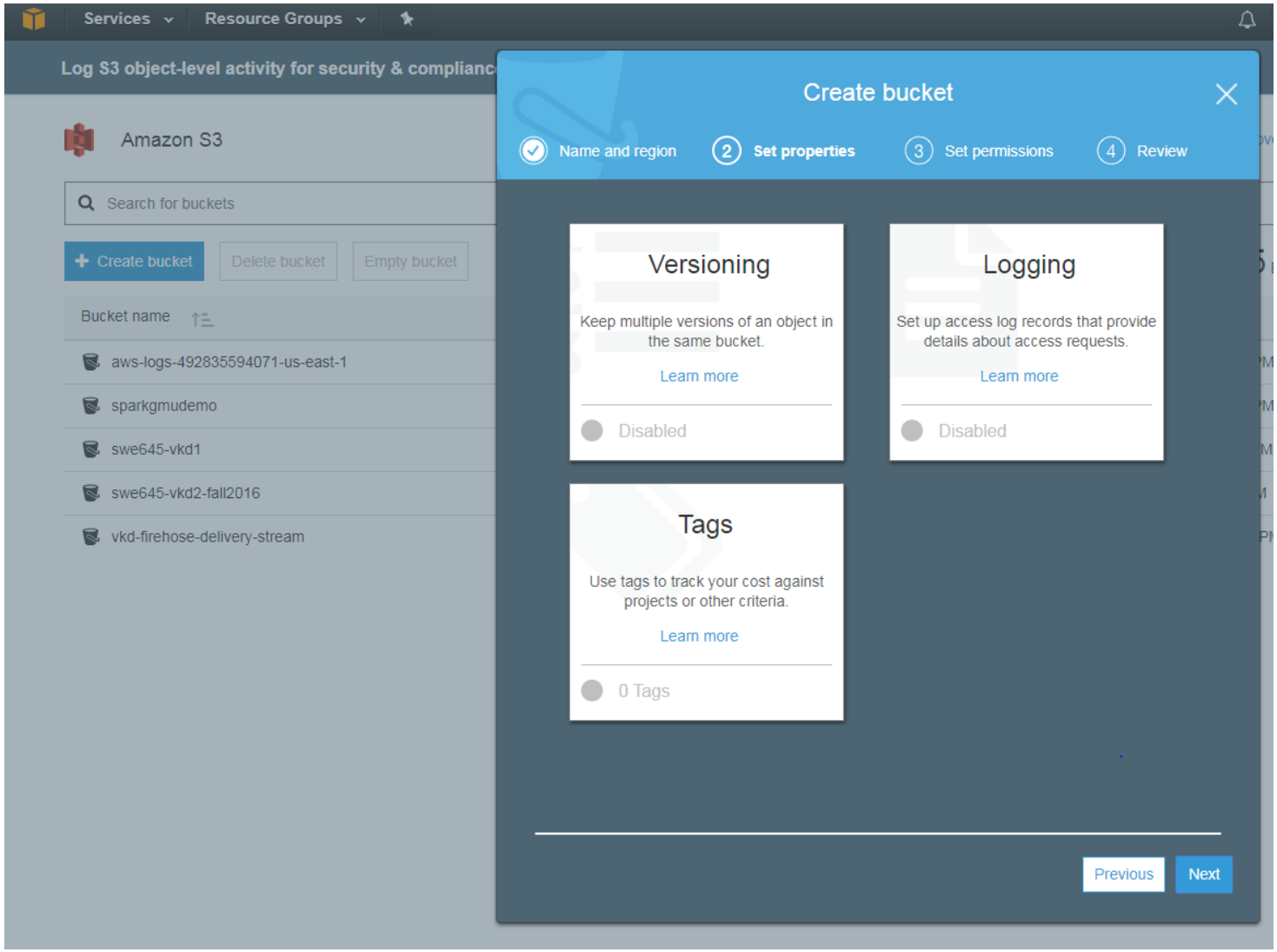


Click Create Bucket to create a place to upload your objects -- Enter a bucket name

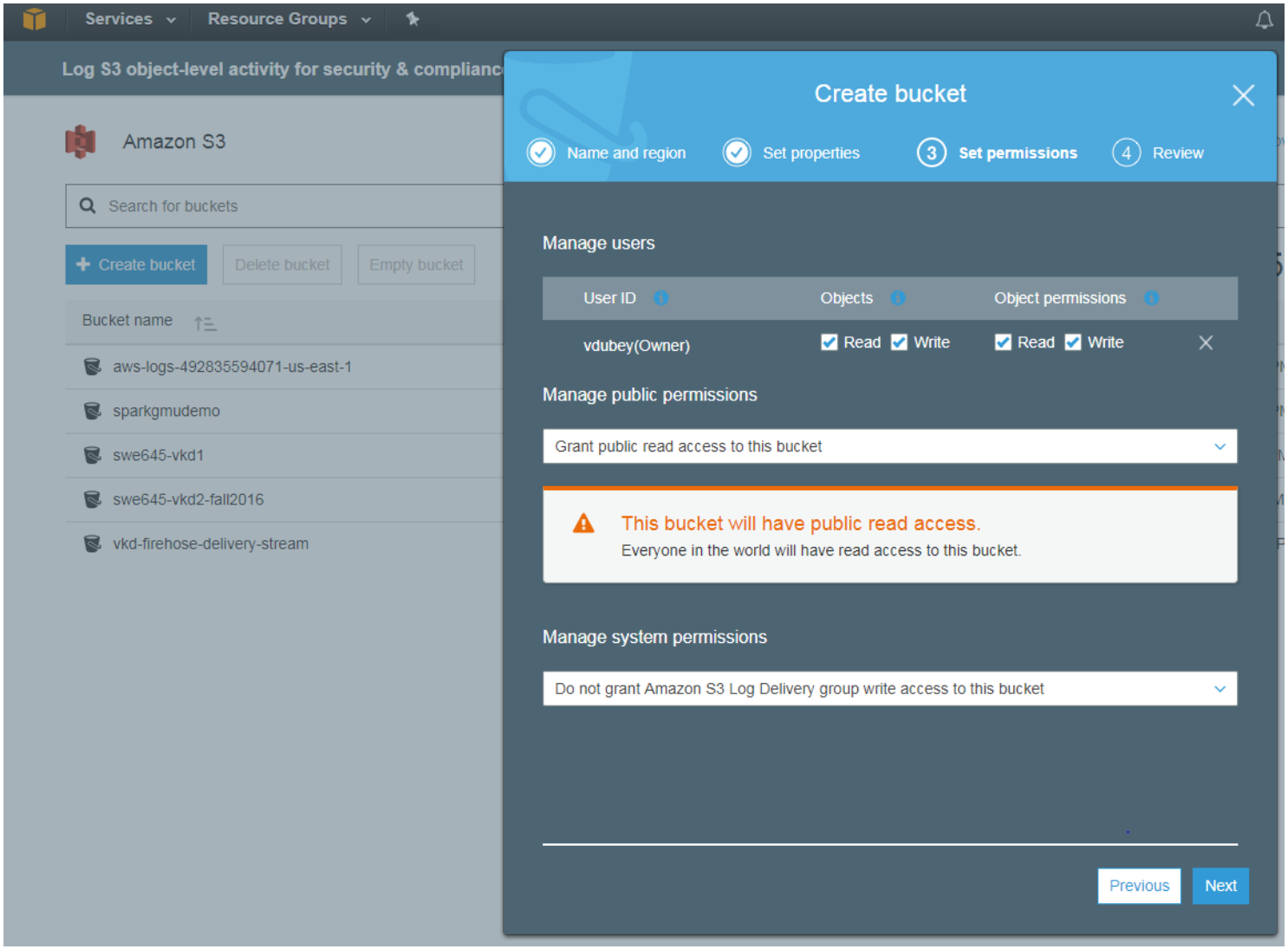
Bucket names are always in lower case and must be globally unique



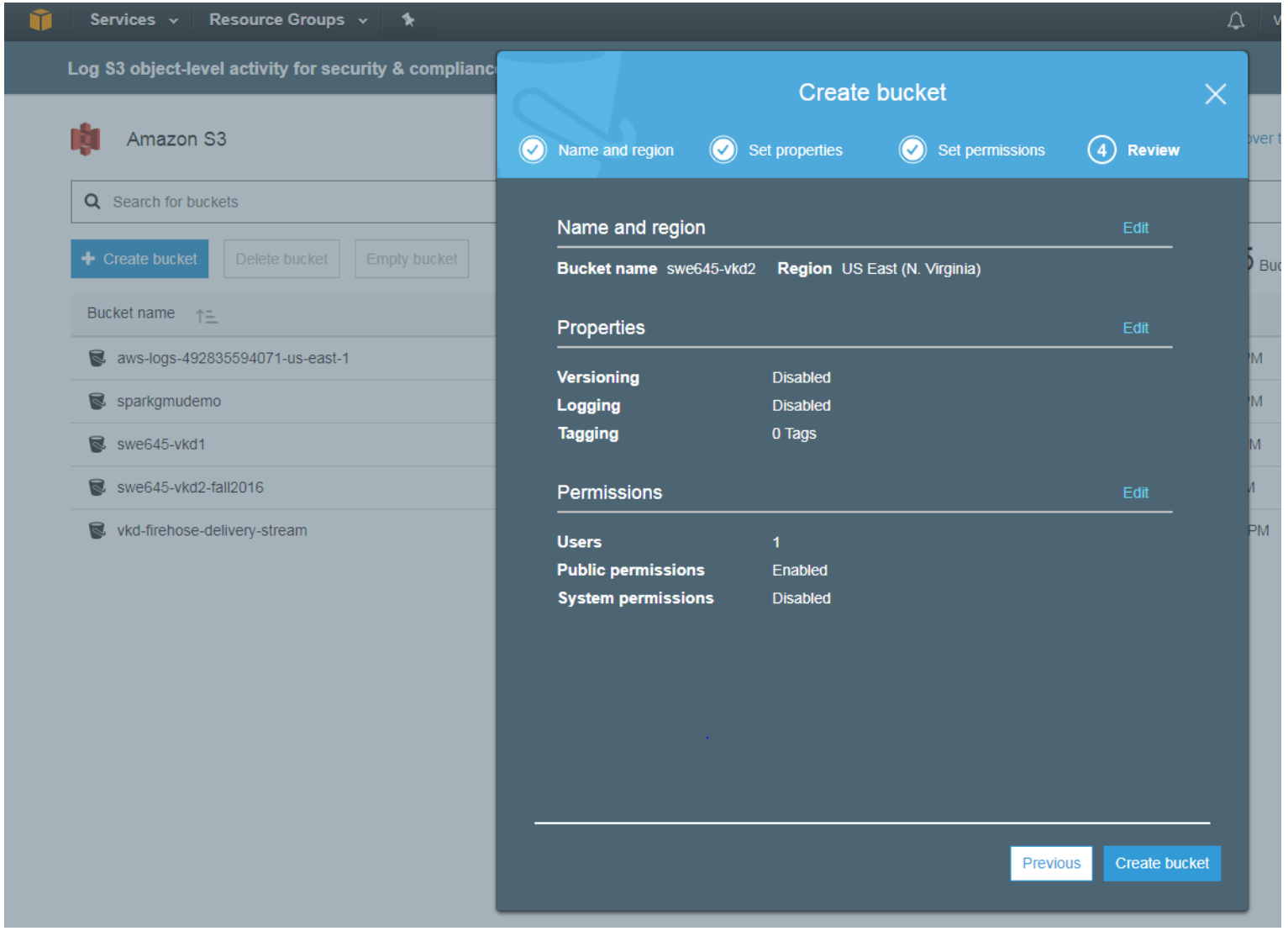
Press Next


Services
Resource Groups
Log S3 object-level activity for security & compliance
Amazon S3
Search for buckets
Create bucket
Delete bucket
Empty bucket
Bucket name
aws-logs-492835594071-us-east-1
sparkgmudemo
swe645-vkd1
swe645-vkd2-fall2016
vkd-firehose-delivery-stream
Create bucket
Name and region
2 Set properties
3 Set permissions
4 Review
Versioning
Keep multiple versions of an object in the same bucket.
Learn more
Disabled
Logging
Set up access log records that provide details about access requests.
Learn more
Disabled
Tags
Use tags to track your cost against projects or other criteria.
Learn more
0 Tags
Previous
Next

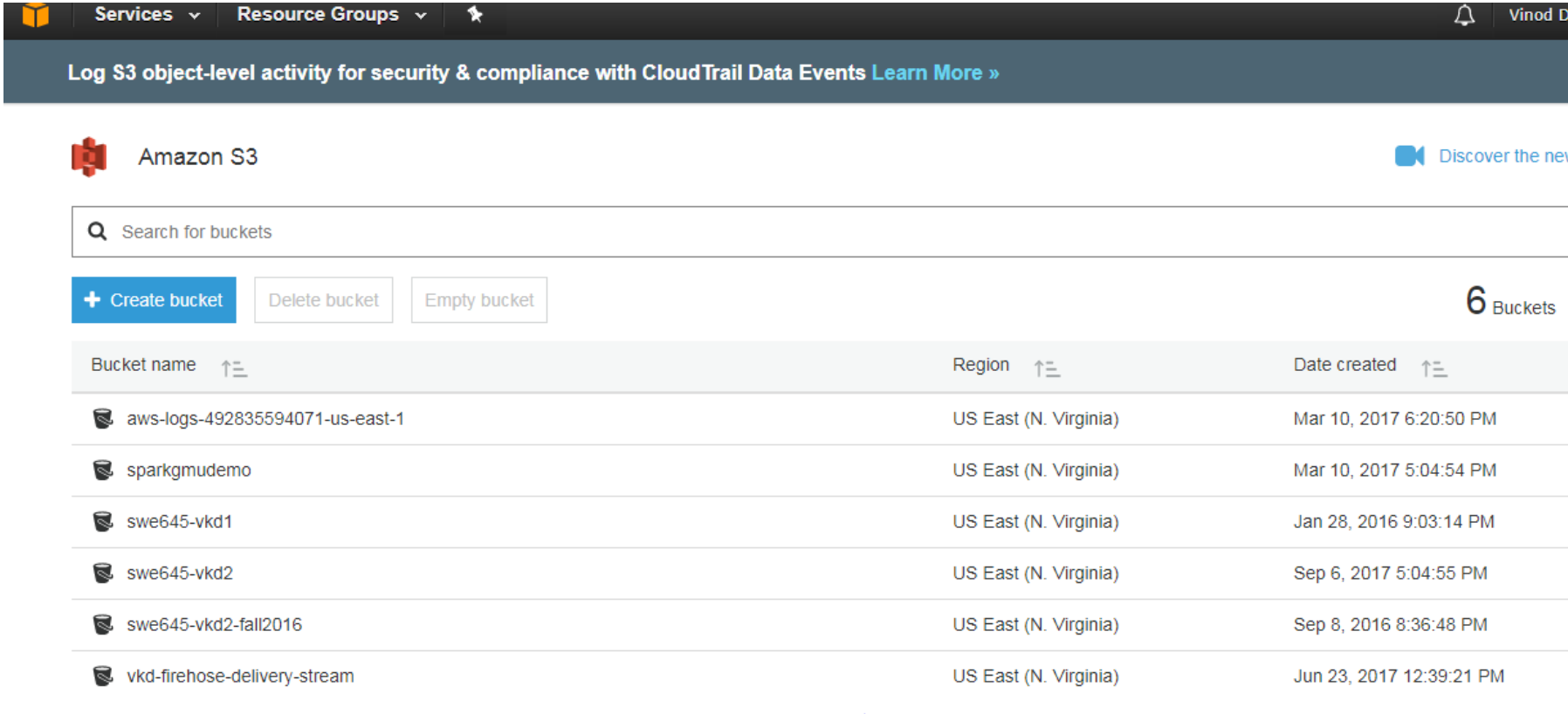
Under Manage Public Permissions, Select "Grant public read access to the bucket"



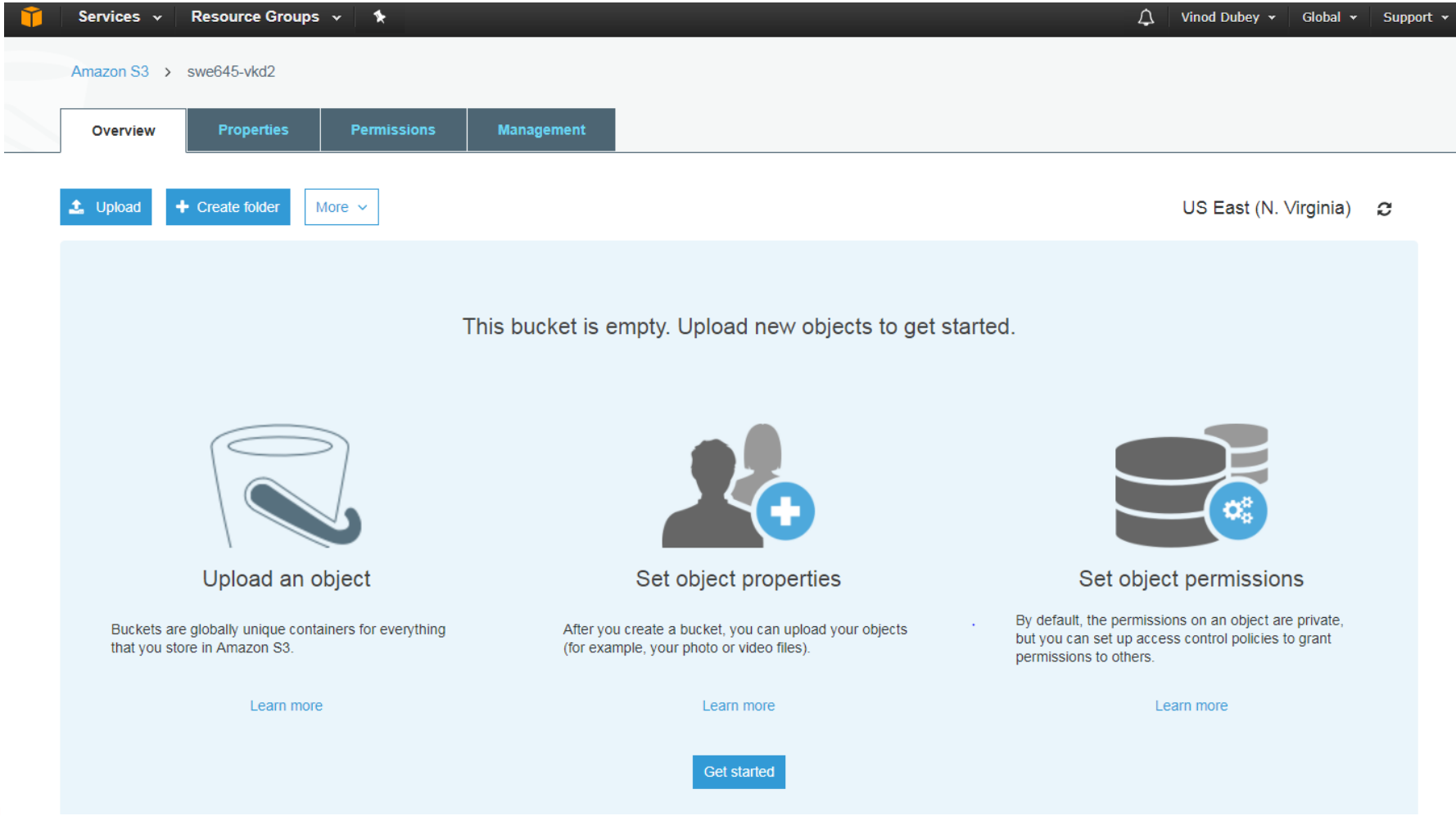
Press “Create Bucket”



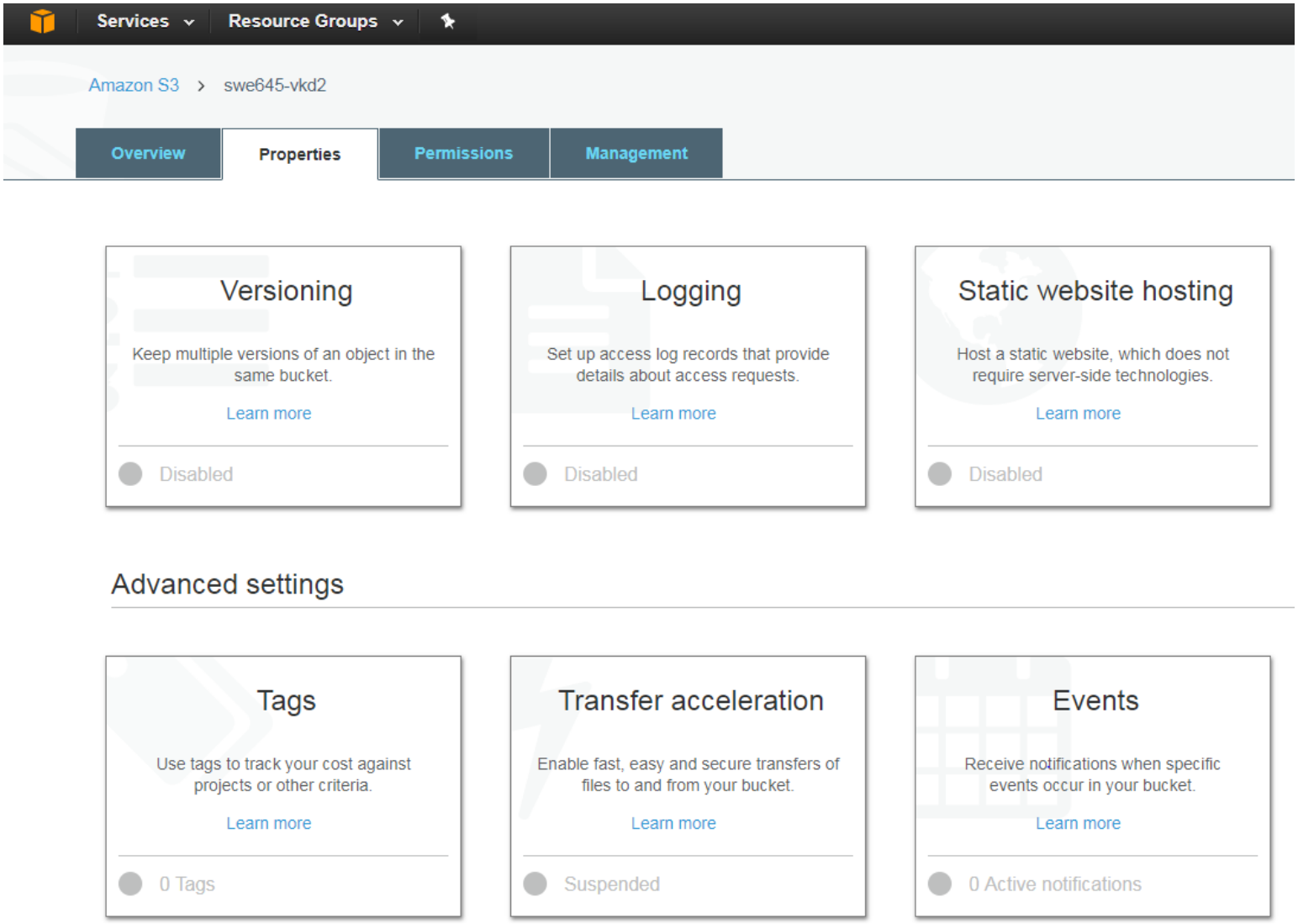
This shows list of S3 buckets in your account. Click on the newly created bucket



There are four tabs under your S3 bucket and link to upload file.

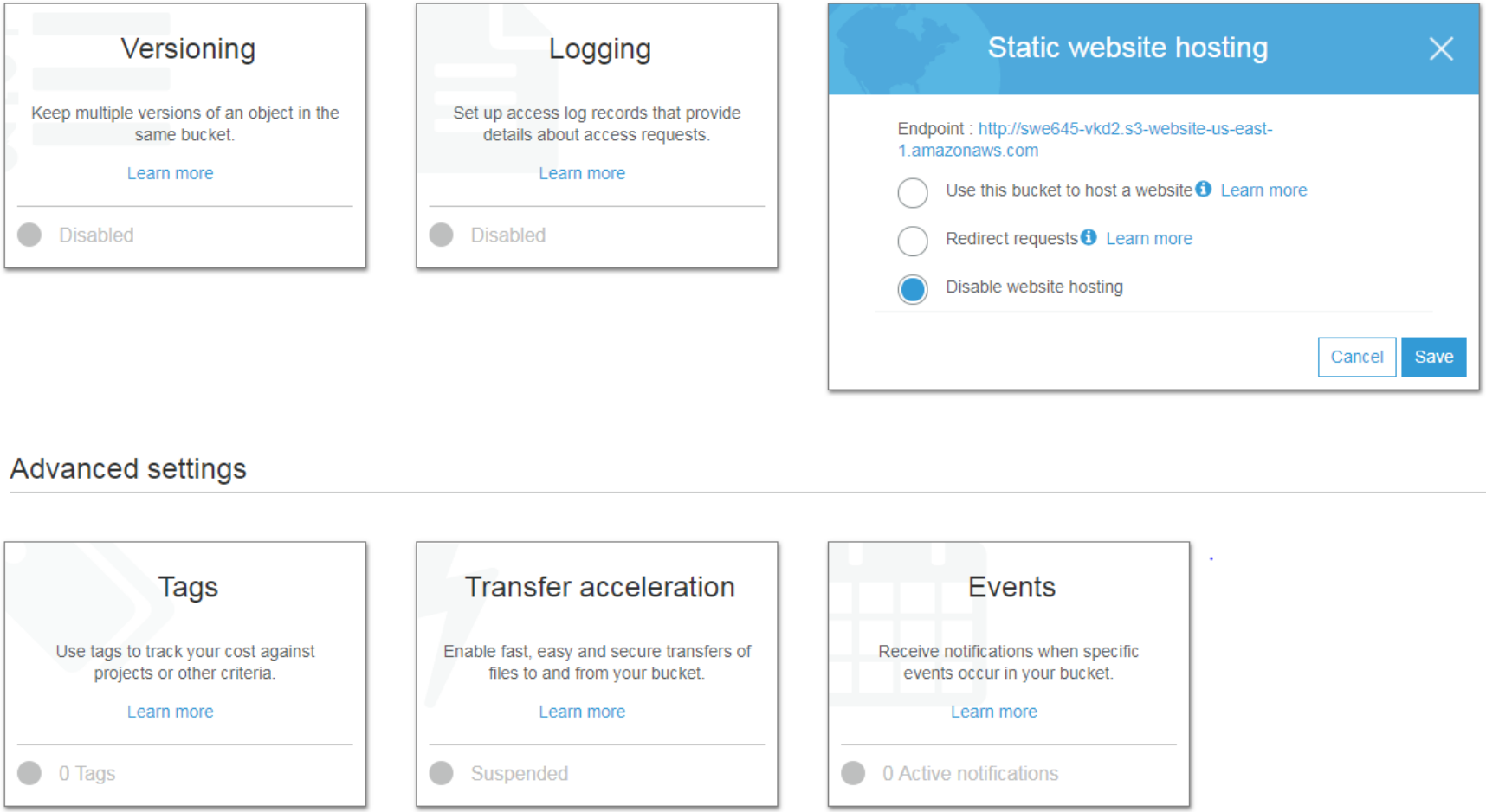


Click on Static website hosting under the Properties tab

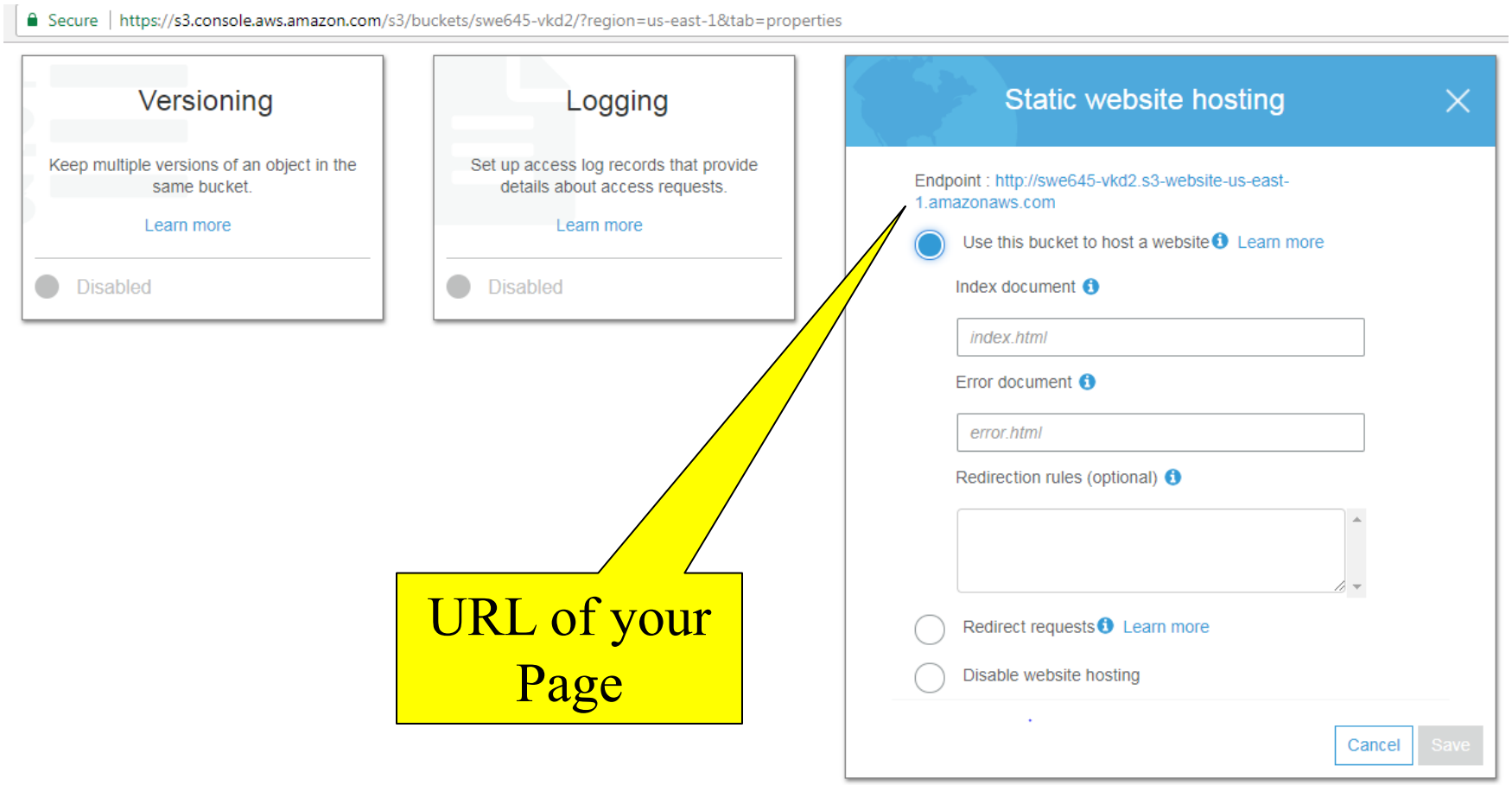


Select radio button for “Use this bucket to host a website”

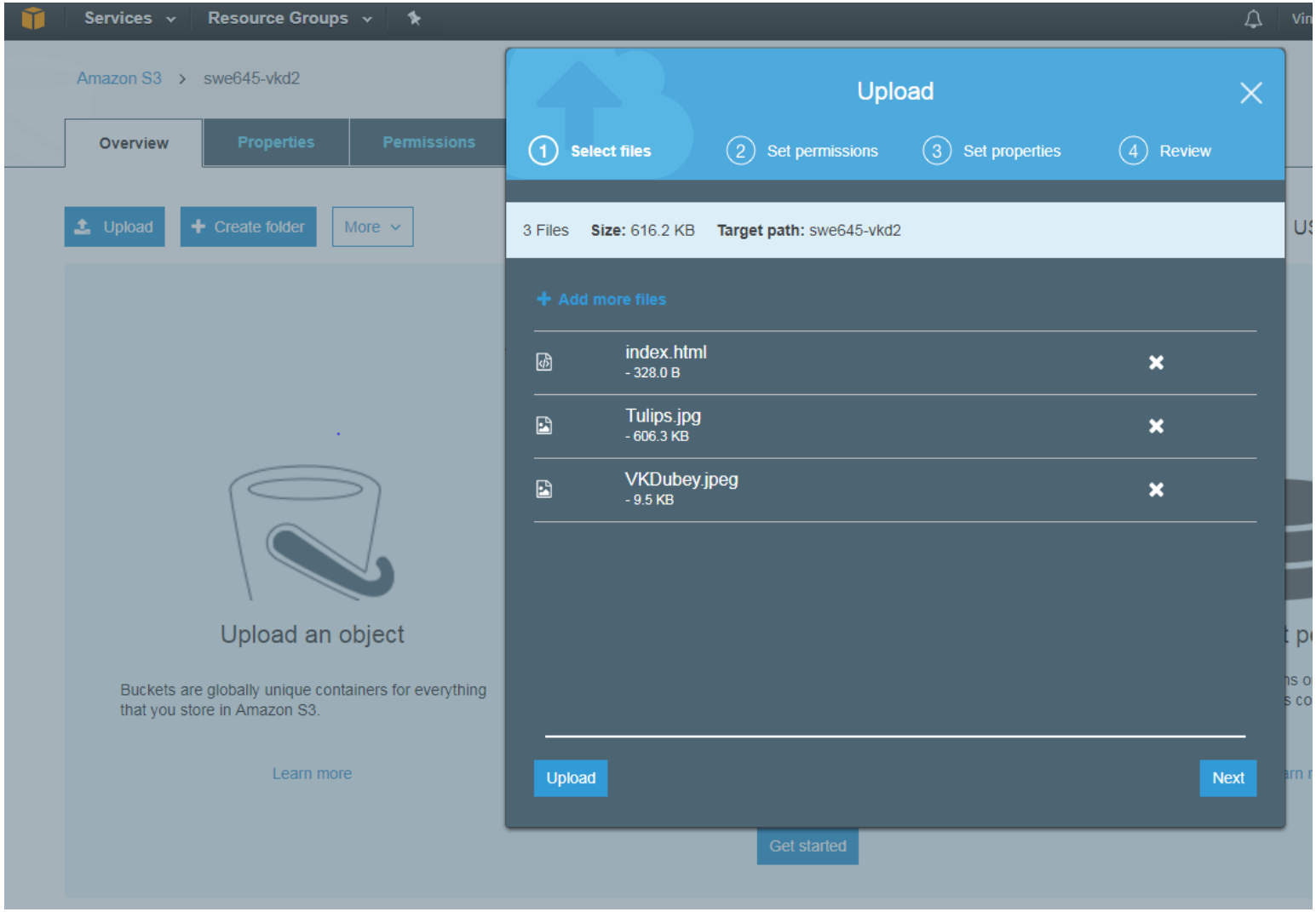
Endpoint is the URL for your website



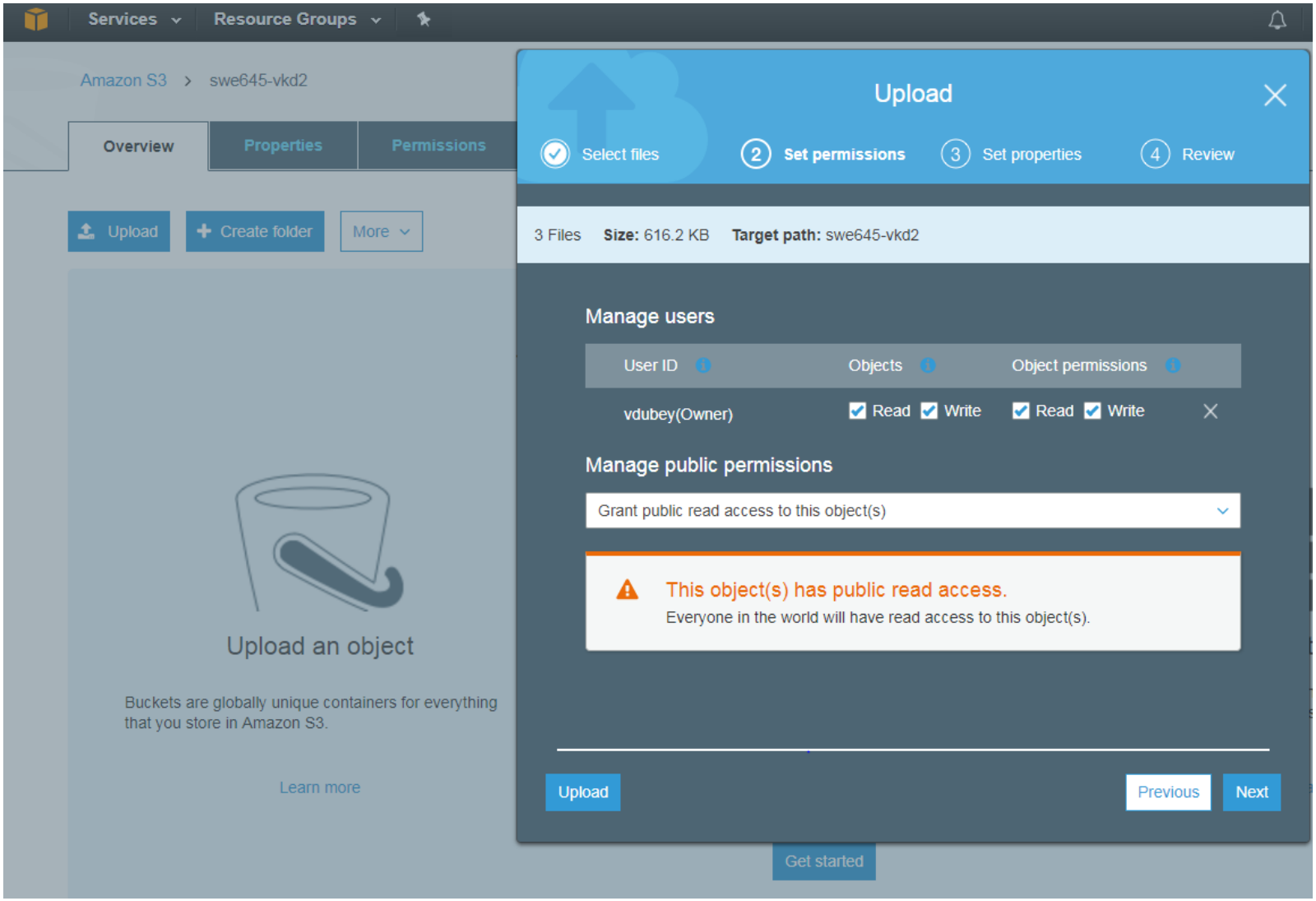
Enter the name of your html file (e.g., index.html) that serves your homepage



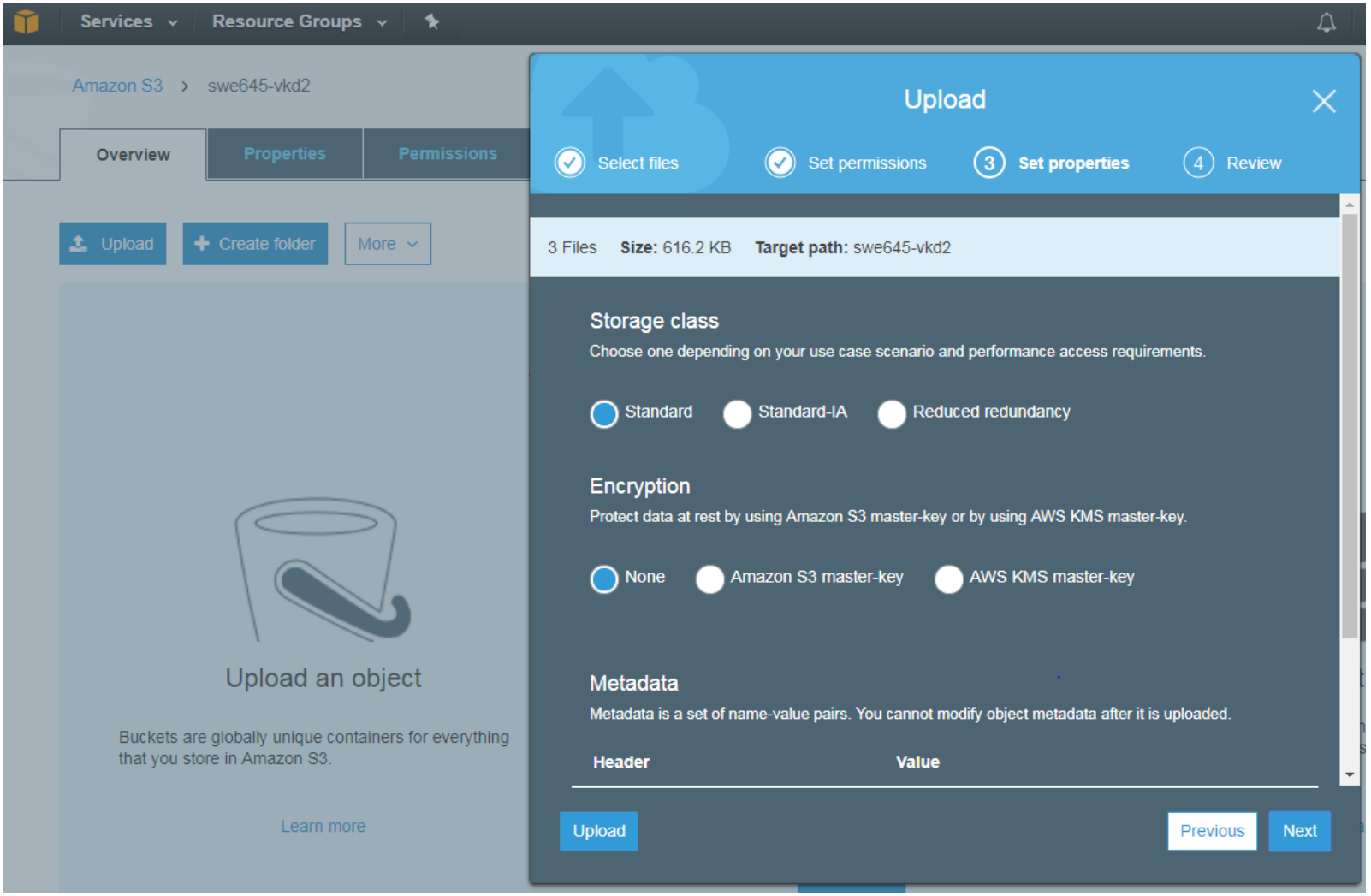
Using Upload under Overview tab, upload index.html and other related files for your page



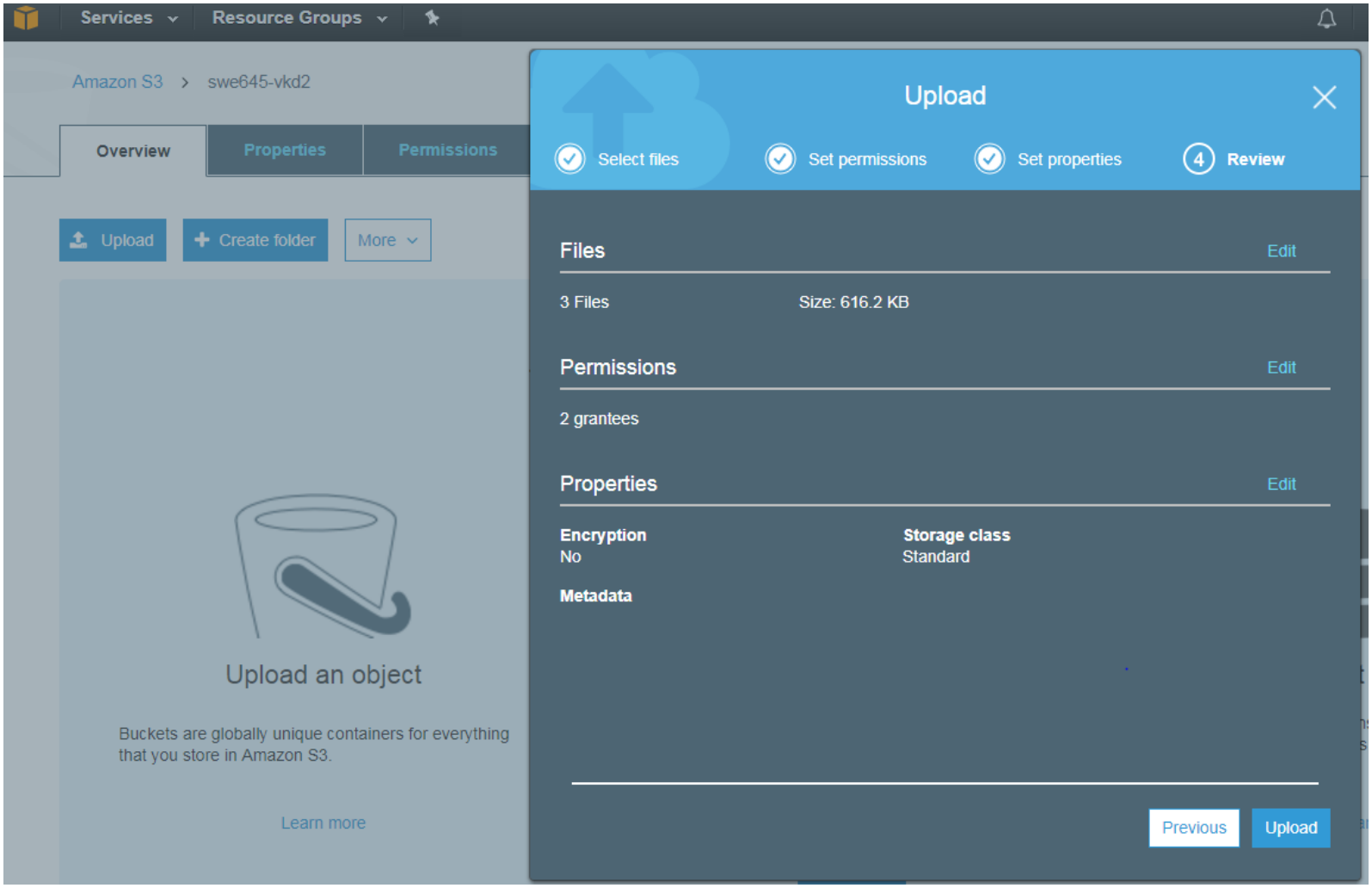
Under “Manage public permissions”, select “Grant public read access to the objects”



Press Next

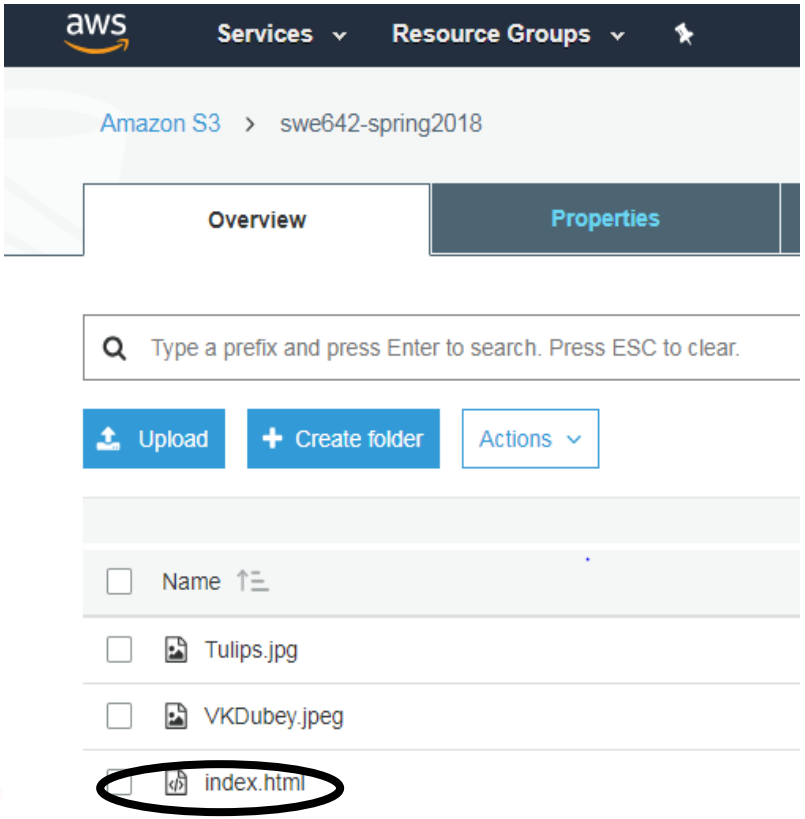

Services
Resource Groups
Amazon S3 > swe645-vkd2
Overview
Properties
Permissions
Upload
Create folder
More
Upload an object
Buckets are globally unique containers for everything that you store in Amazon S3.
Learn more
Upload
Select files
Set permissions
3 Set properties
4 Review
3 Files Size: 616.2 KB Target path: swe645-vkd2
Storage class
Choose one depending on your use case scenario and performance access requirements.
Standard
Standard-IA
Reduced redundancy
Encryption
Protect data at rest by using Amazon S3 master-key or by using AWS KMS master-key.
None
Amazon S3 master-key
AWS KMS master-key
Metadata
Metadata is a set of name-value pairs. You cannot modify object metadata after it is uploaded.
Header
Value
Upload
Previous
Next

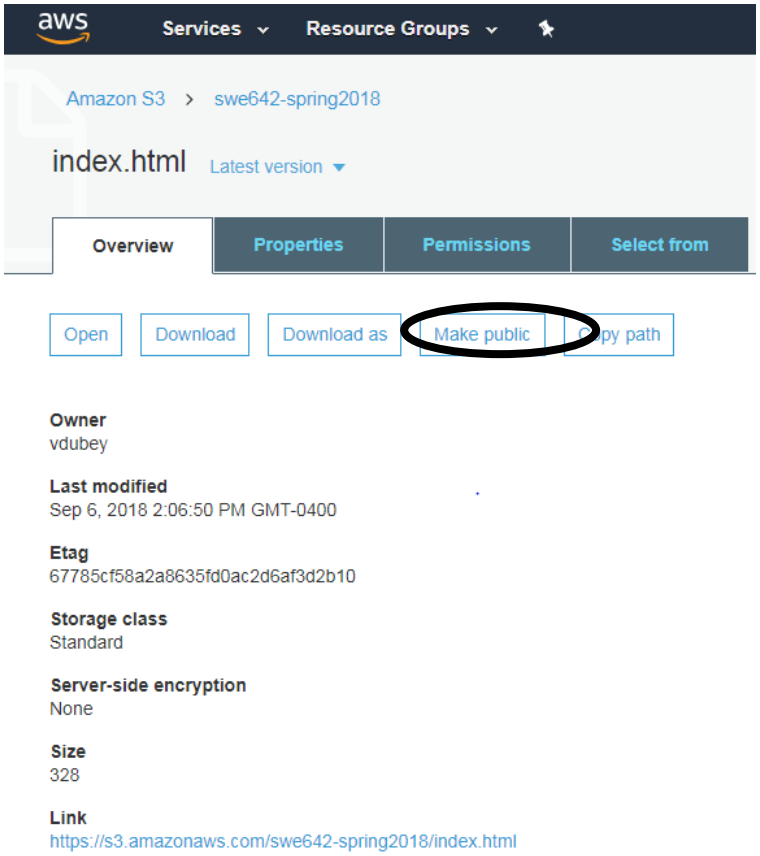
Press Upload



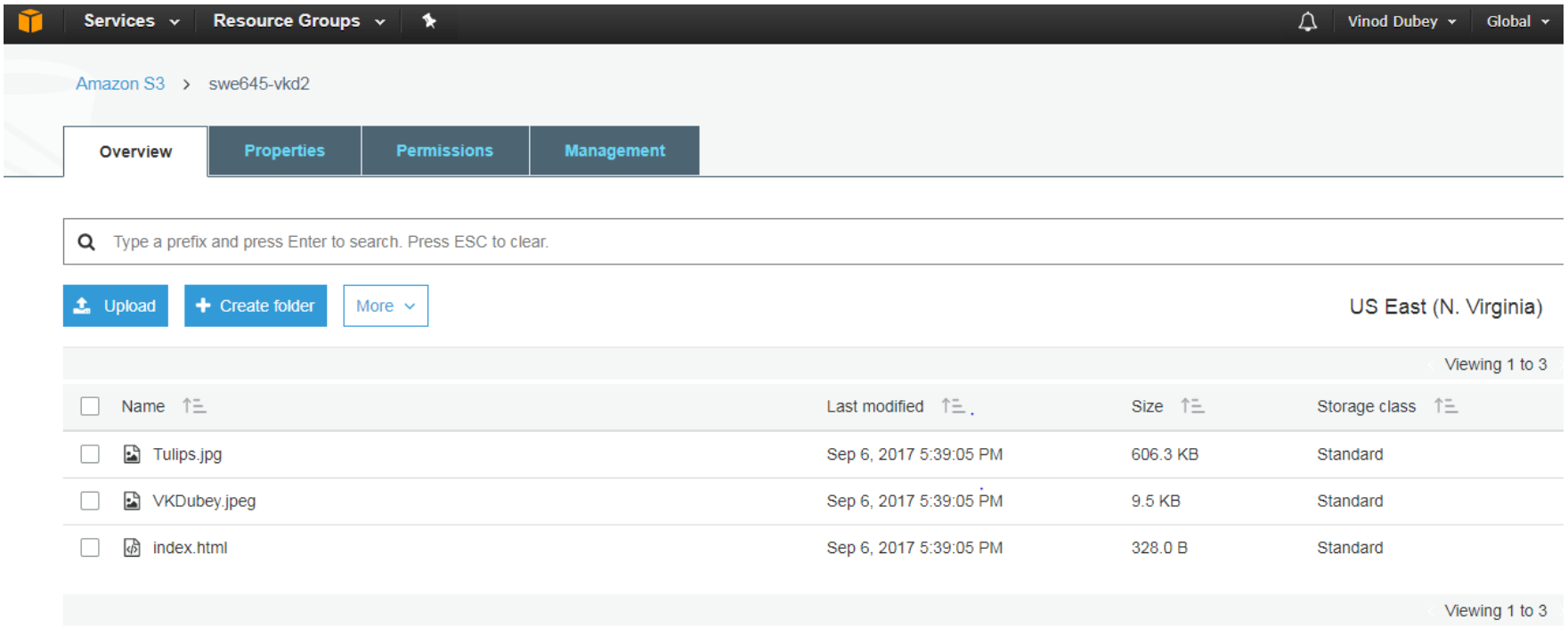
In case of Access errors…

- **If there are any Access Denied type error, then**
  - Click (not select) on each file/object of your Web page at a time
  - On the resulting page, make sure to press 'Make Public'



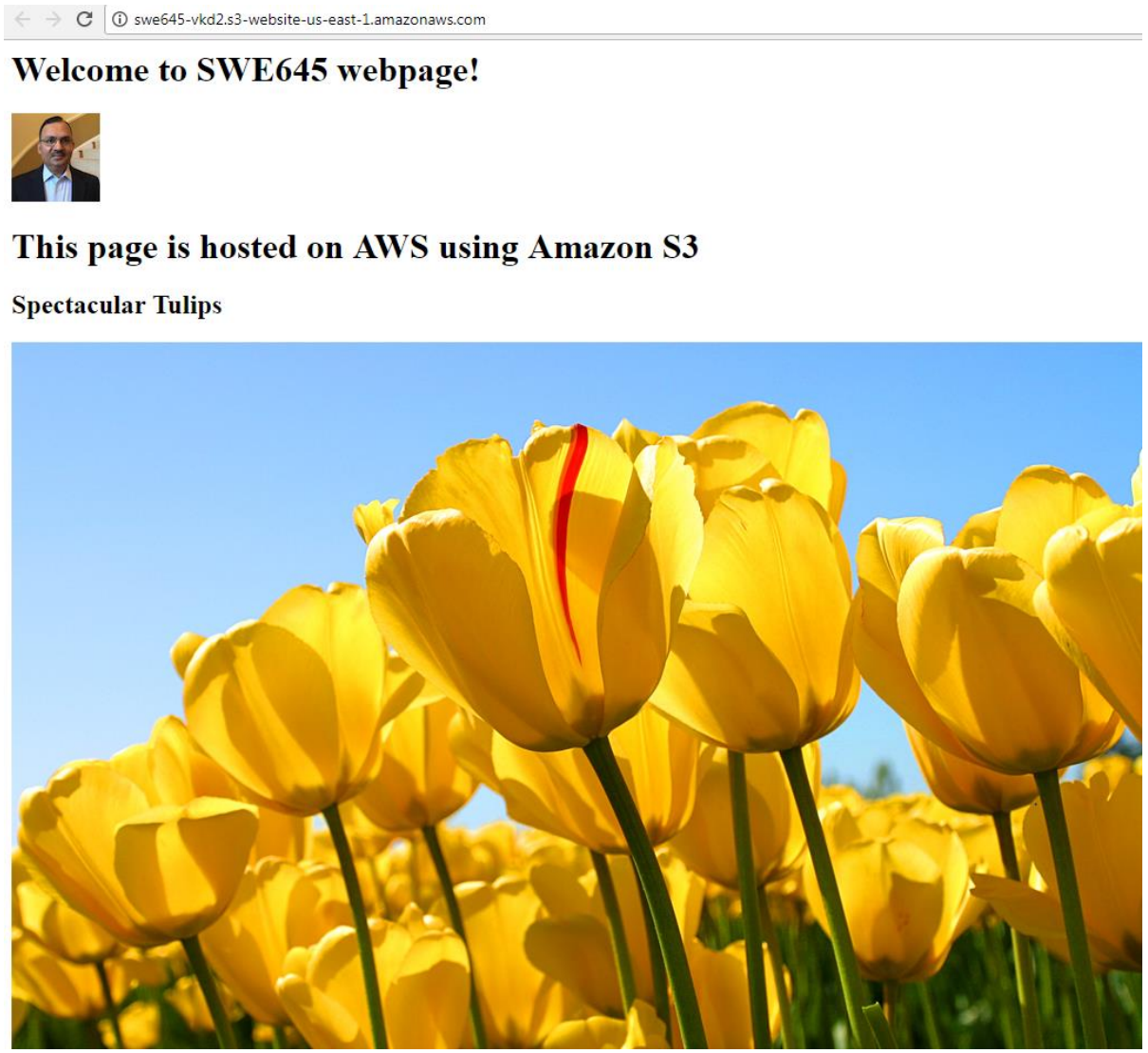


Here are my uploaded files for my website



Here is my website:

http://swe645-vkd2.s3-website-us-east-1.amazonaws.com/



Amazon Elastic Compute Cloud (Amazon EC2)

Virtual Server Hosting

- **Amazon EC2 is a service that provides resizable compute capacity in the AWS cloud.**
- **Using Amazon EC2 eliminates the need to invest in computing hardware up front, which saves money but also allows applications to be developed and deployed faster**
- **You can use Amazon EC2 to launch a virtual machine**
- **Think of it as a virtual server or thousands of virtual servers as your own data center**
- **You can use EC2 to launch a virtual machine and you can configure all of the associated security and networking settings**

- **Provides simple web service interface to obtain and configure capacity for your machine.**

- **Reduces the time required to obtain and boot new server instances to minutes**
  - allowing you to quickly scale capacity, both up and down, as your computing requirements change.
- **Pay-as-you go pricing model – you pay only for capacity that you actually use.**

- **Amazon EC2 provides developers the tools to build failure scalable, resilient, and elastic applications**

Terminology

- **Instance – One running virtual machine**
- **Instance Type – hardware configuration: cores, memory, disk.**
- **Instance Store Volume – Temporary disk associated with instance.**
- **Image (AMI) – Pre-configured templates for provisioning EC2 instances**
  - Stored bits which can be turned into instances
  - Can include O/S like Linux or Windows and/or pre-installed software packages

Terminology

- **Key Pair – Credentials used to access VM from command line.**
- **Region – Geographic location, price, laws, network locality.**
- **Availability Zone – Subdivision of region the is fault-independent.**

Securing Amazon EC2 Instances: Security Groups

- Similar to traditional firewalls
- There are settings that enable you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups

EC2 Instance Types

***There are a wide range of instance types that have varying combinations of CPU power, memory, storage, and networking capacity**

***Range from small "micro" instances for small jobs to high performance "x-large" instances for things like data warehousing**

Model	vCPU	CPU Credits / hour	Mem (GiB)	Storage (GB)
t2.micro	1	6	1	EBS Only
t2.small	1	12	2	EBS Only
t2.medium	2	24	4	EBS Only

Model	vCPU	Mem (GiB)	SSD Storage (GB)
c3.large	2	3.75	2 x 16
c3.xlarge	4	7.5	2 x 40
c3.2xlarge	8	15	2 x 80
c3.4xlarge	16	30	2 x 160
c3.8xlarge	32	60	2 x 320

Use Cases

High performance front-end fleets, web-servers, on-demand batch processing, distributed analytics, high performance science and engineering applications, ad serving, batch processing, MMO gaming, video encoding, and distributed analytics.

Model	vCPU	Mem (GiB)	SSD Storage (GB)
m3.medium	1	3.75	1 x 4
m3.large	2	7.5	1 x 32
m3.xlarge	4	15	2 x 40
m3.2xlarge	8	30	2 x 80

Model	vCPU	Mem (GiB)	SSD Storage (GB)
r3.large	2	15.25	1 x 32
r3.xlarge	4	30.5	1 x 80
r3.2xlarge	8	61	1 x 160
r3.4xlarge	16	122	1 x 320
r3.8xlarge	32	244	2 x 320

Use Cases

We recommend memory-optimized instances for high performance databases, distributed memory caches, in-memory analytics, genome assembly and analysis, larger deployments of SAP, Microsoft SharePoint, and other enterprise applications.

EC2 Pricing Model

http://aws.amazon.com/ec2/pricing/

- **Free Usage Tier**
- **On-Demand Instances**
  - Start and stop instances whenever you like, costs are rounded up to the nearest hour. (Worst price)
- **Reserved Instances**
  - Pay up front for one/three years in advance. (Best price)
  - Unused instances can be sold on a secondary market.
- **Spot Instances**
  - Specify the price you are willing to pay, and instances get started and stopped without any warning as the market changes.

Free Usage Tier

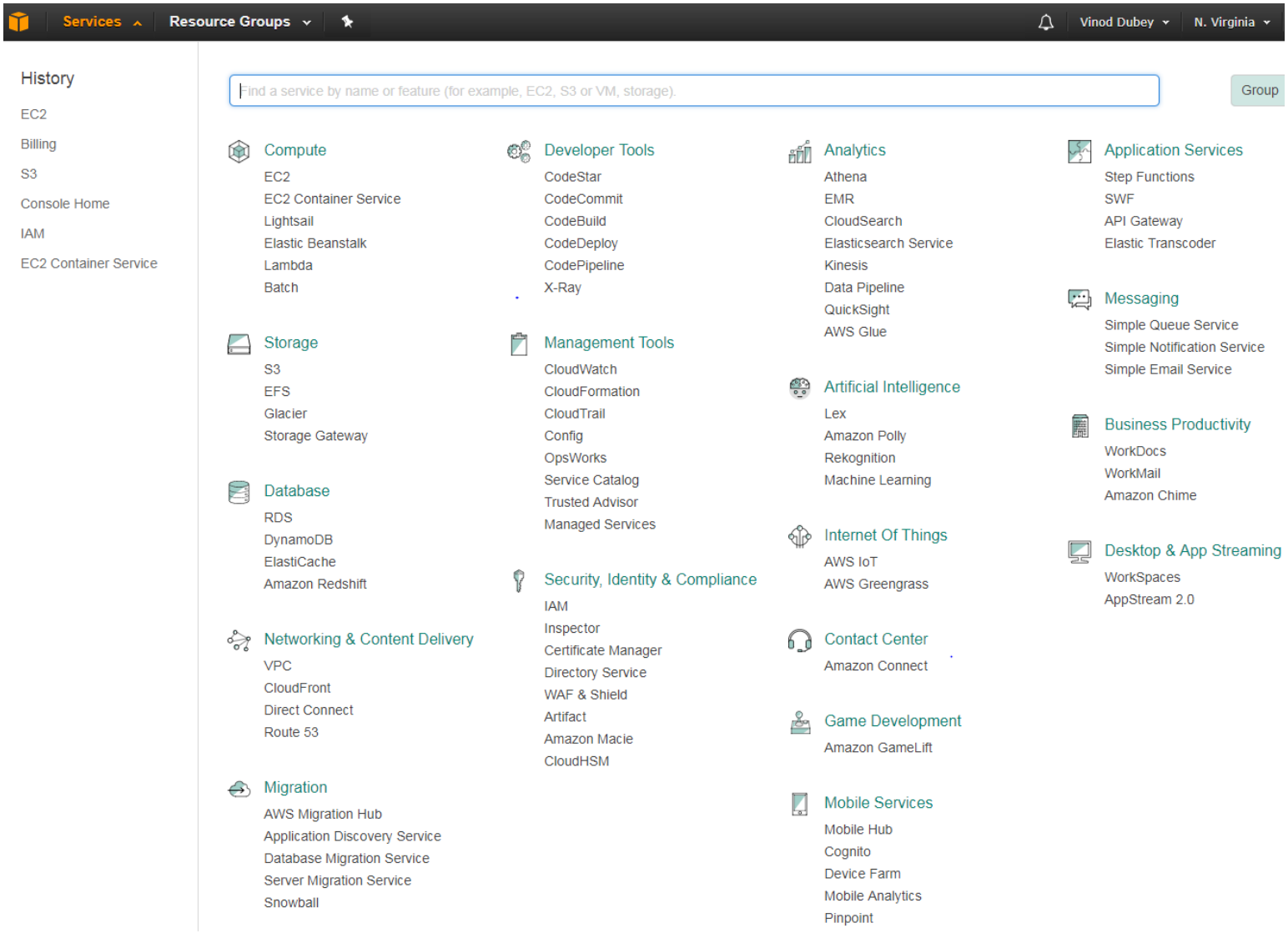
- 750 hours of EC2 running Linux, RHEL, or SLES t2.micro instance usage
- 750 hours of EC2 running Microsoft Windows Server t2.micro instance usage
- 750 hours of Elastic Load Balancing plus 15 GB data processing
- 30 GB of Amazon Elastic Block Storage in any combination of General Purpose (SSD) or Magnetic, plus 2 million I/Os (with Magnetic) and 1 GB of snapshot storage
- 15 GB of bandwidth out aggregated across all AWS services
- 1 GB of Regional Data Transfer

Accessing Amazon EC2

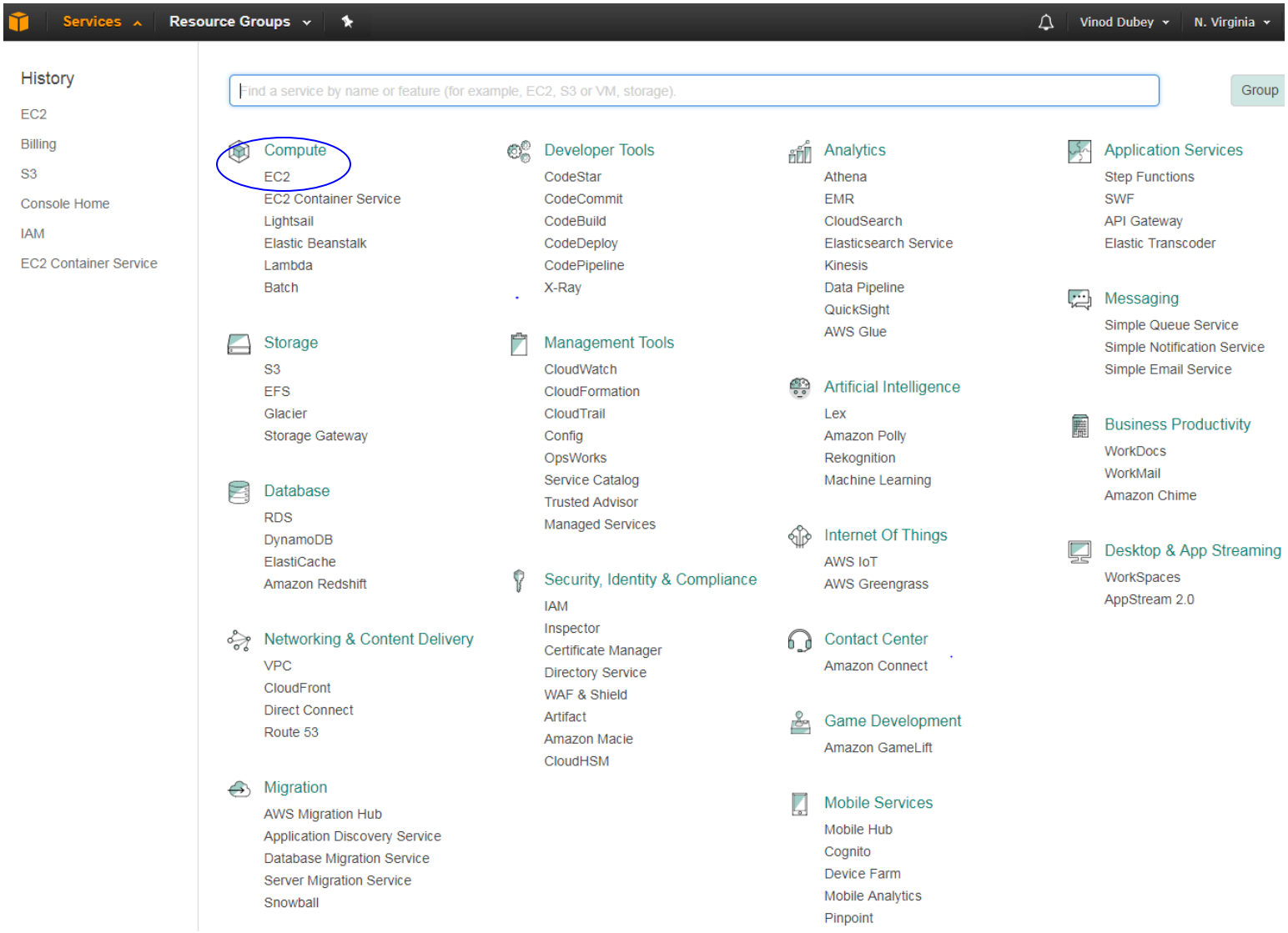
- **A number of ways to use EC2**
  - AWS Management Console – powerful, intuitive and easy to use
  - Command Line Interface
  - API access

An example to create, find, deploy, and terminate an EC2 instance

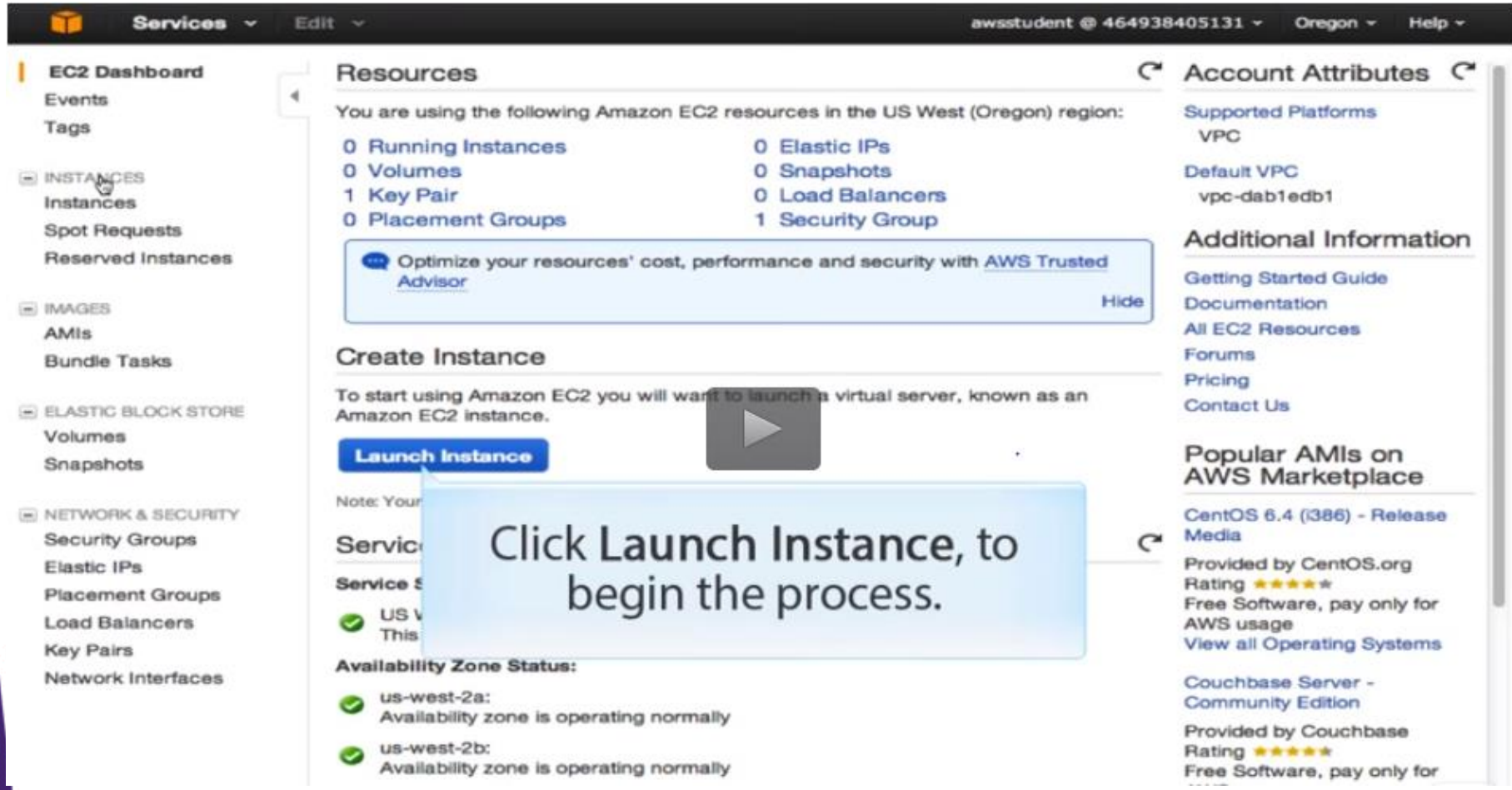
Login to the AWS Management Console to get started



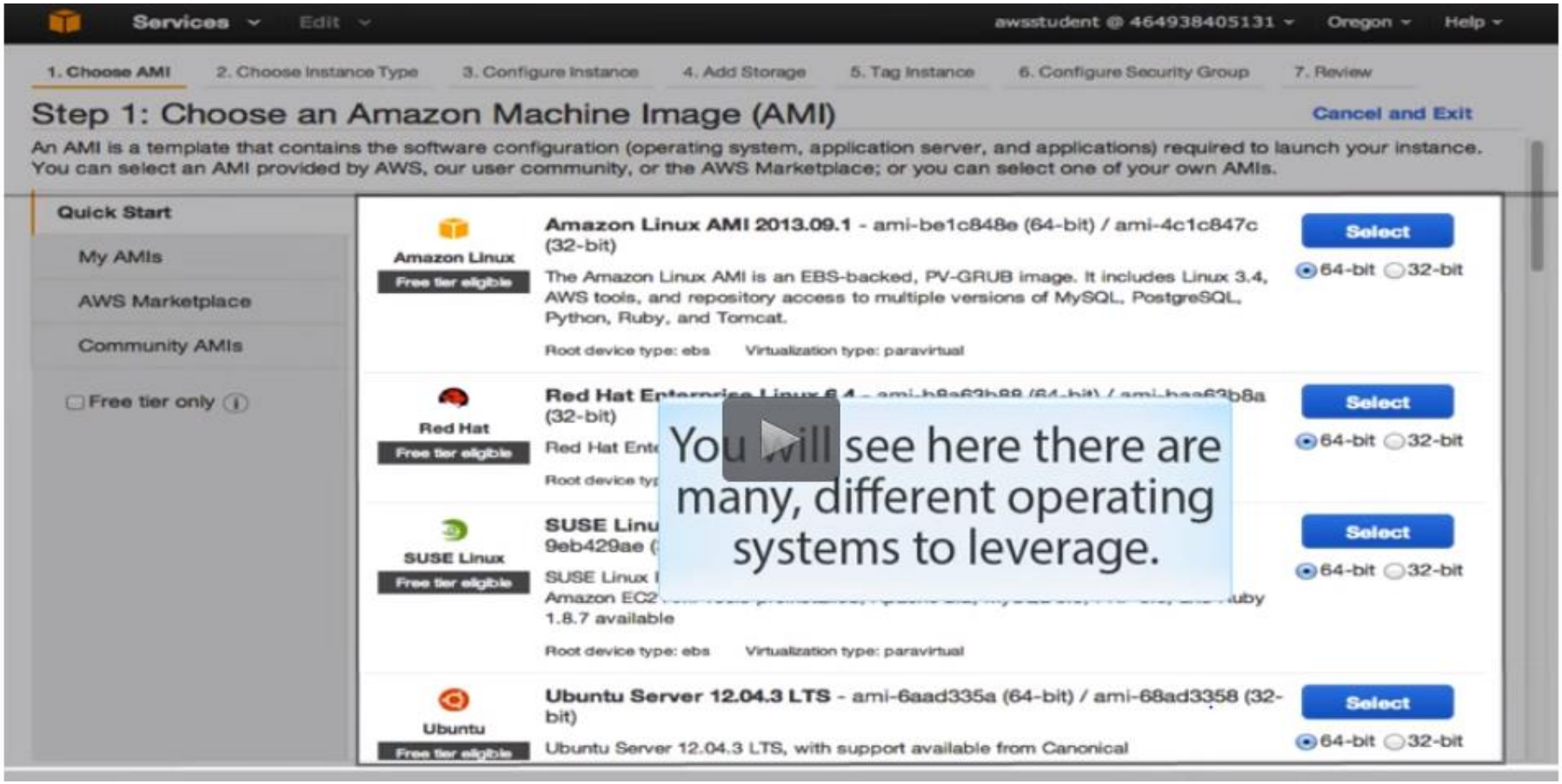
Click Amazon EC2 icon on the AWS Console homepage under Compute



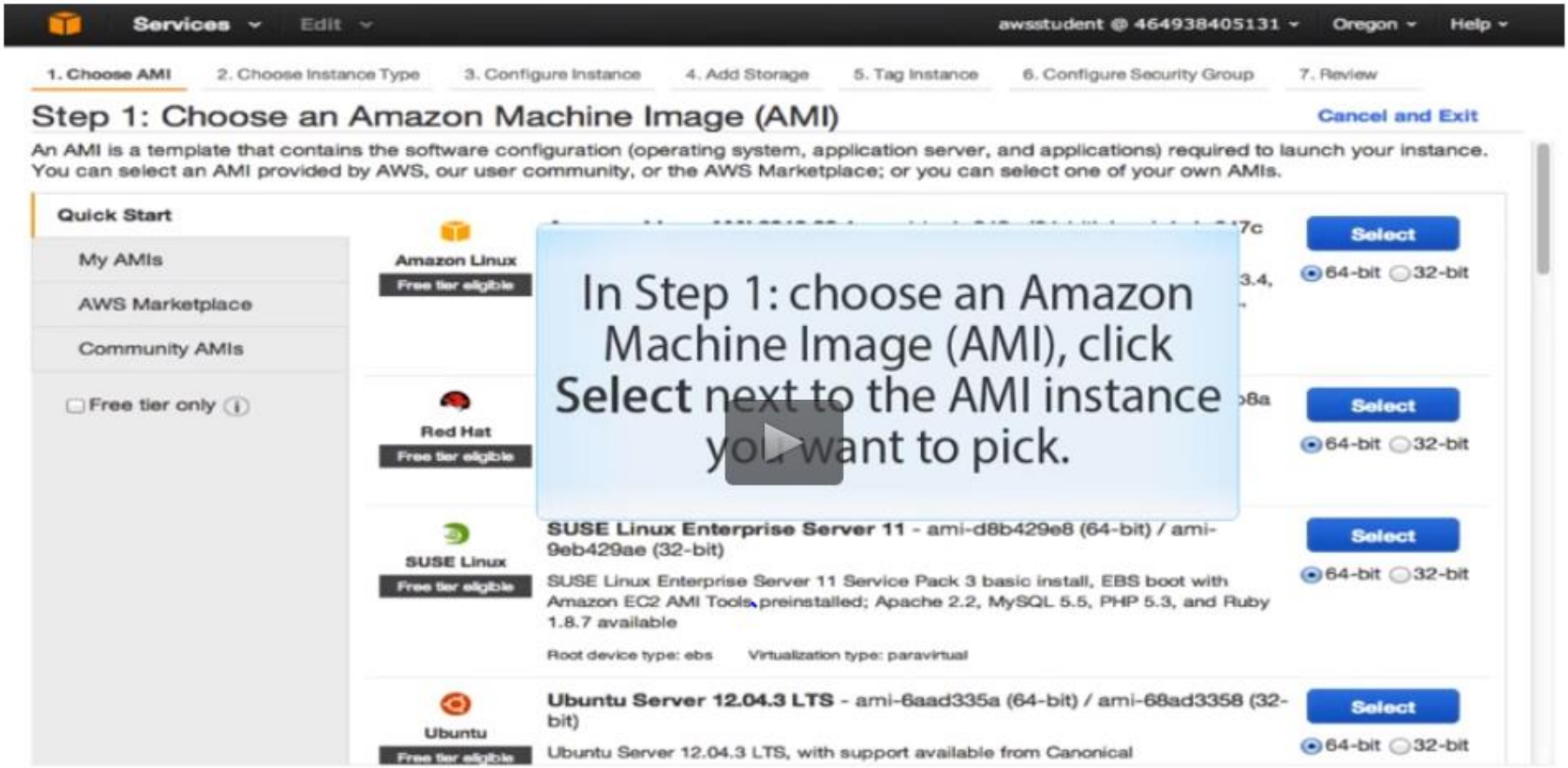
Click “Launch Instance” to begin the process



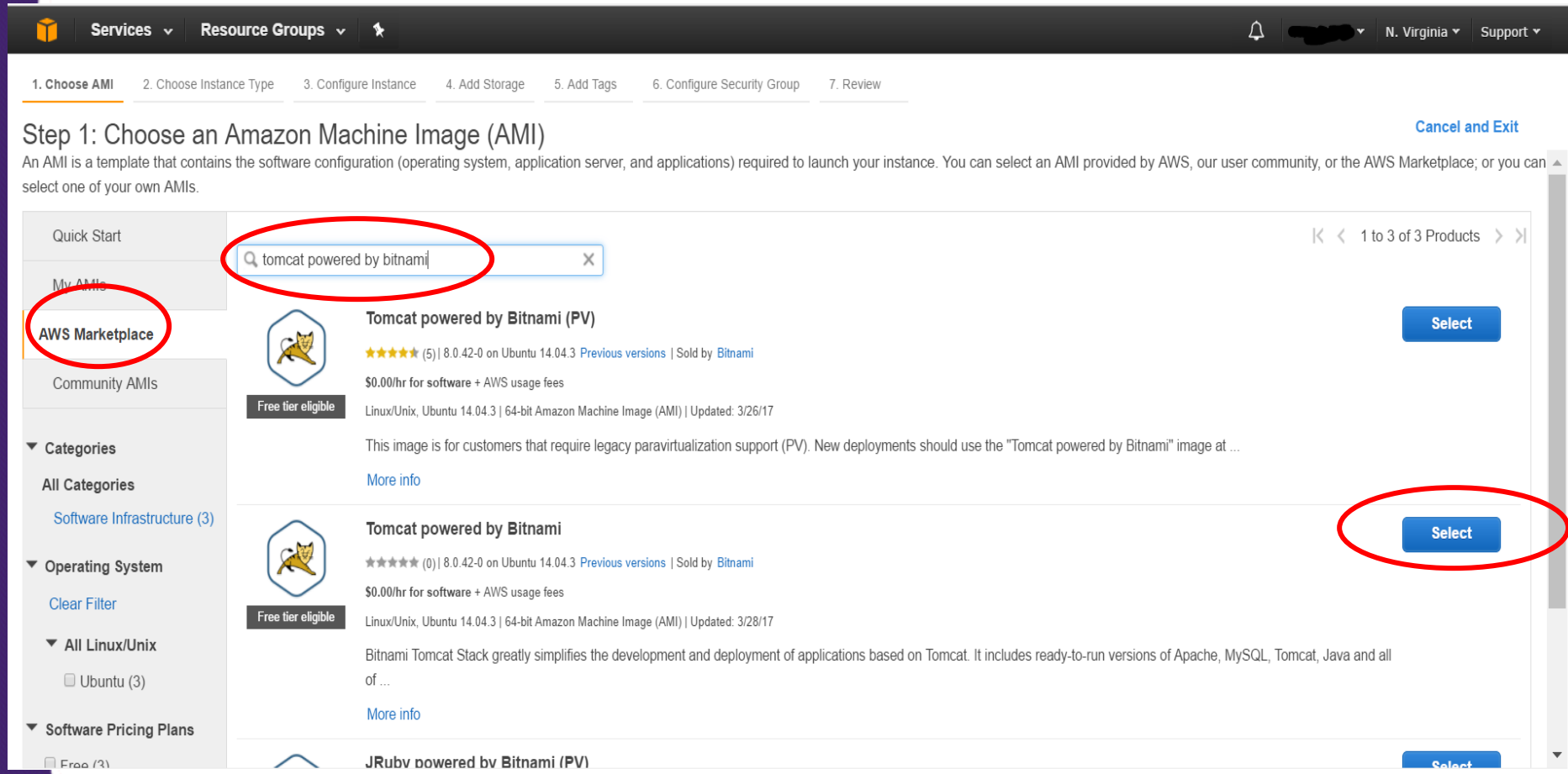
An example to create, find, deploy, and terminate an EC2 instance



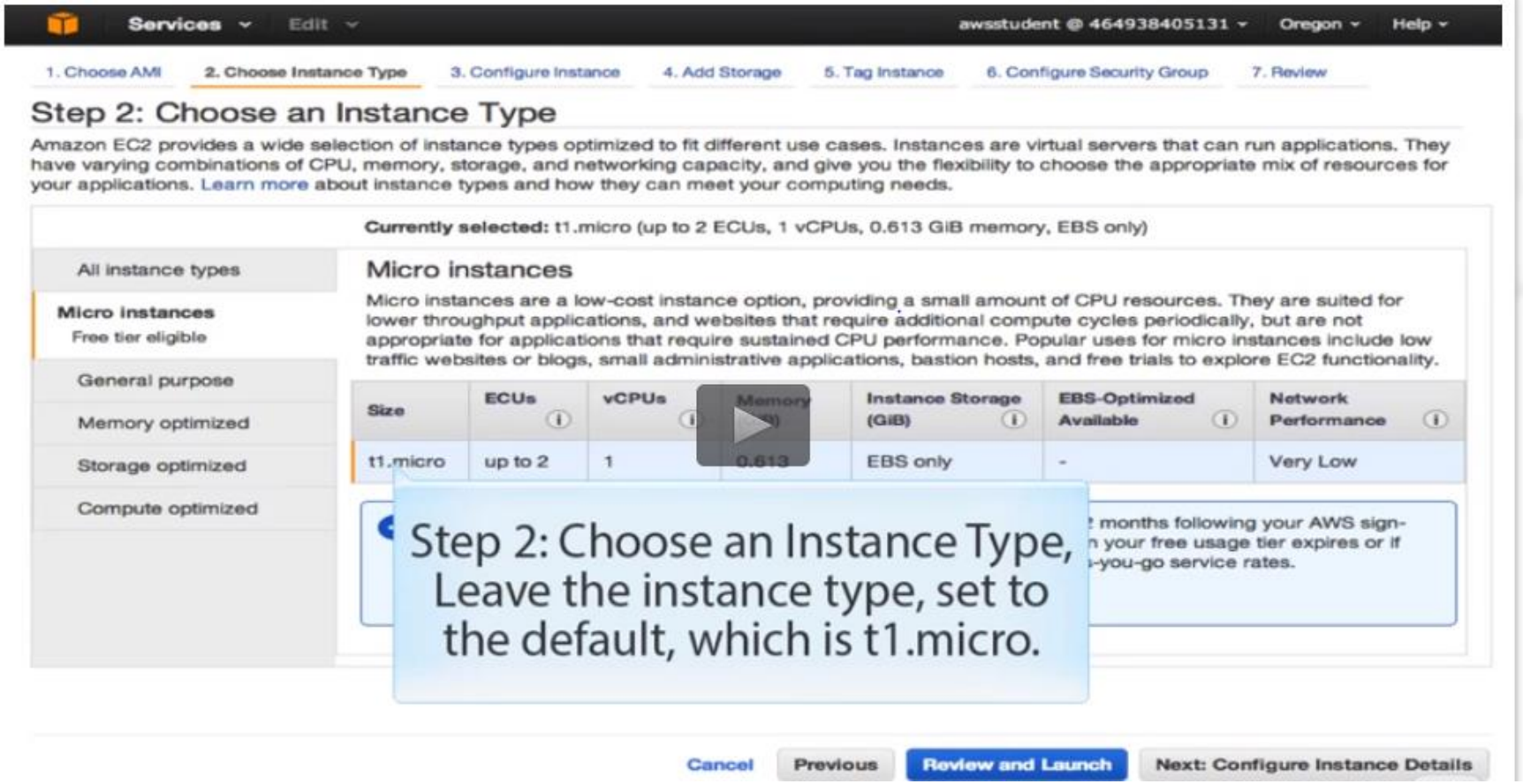
You need to choose an AMI that already has Tomcat installed – see the next page



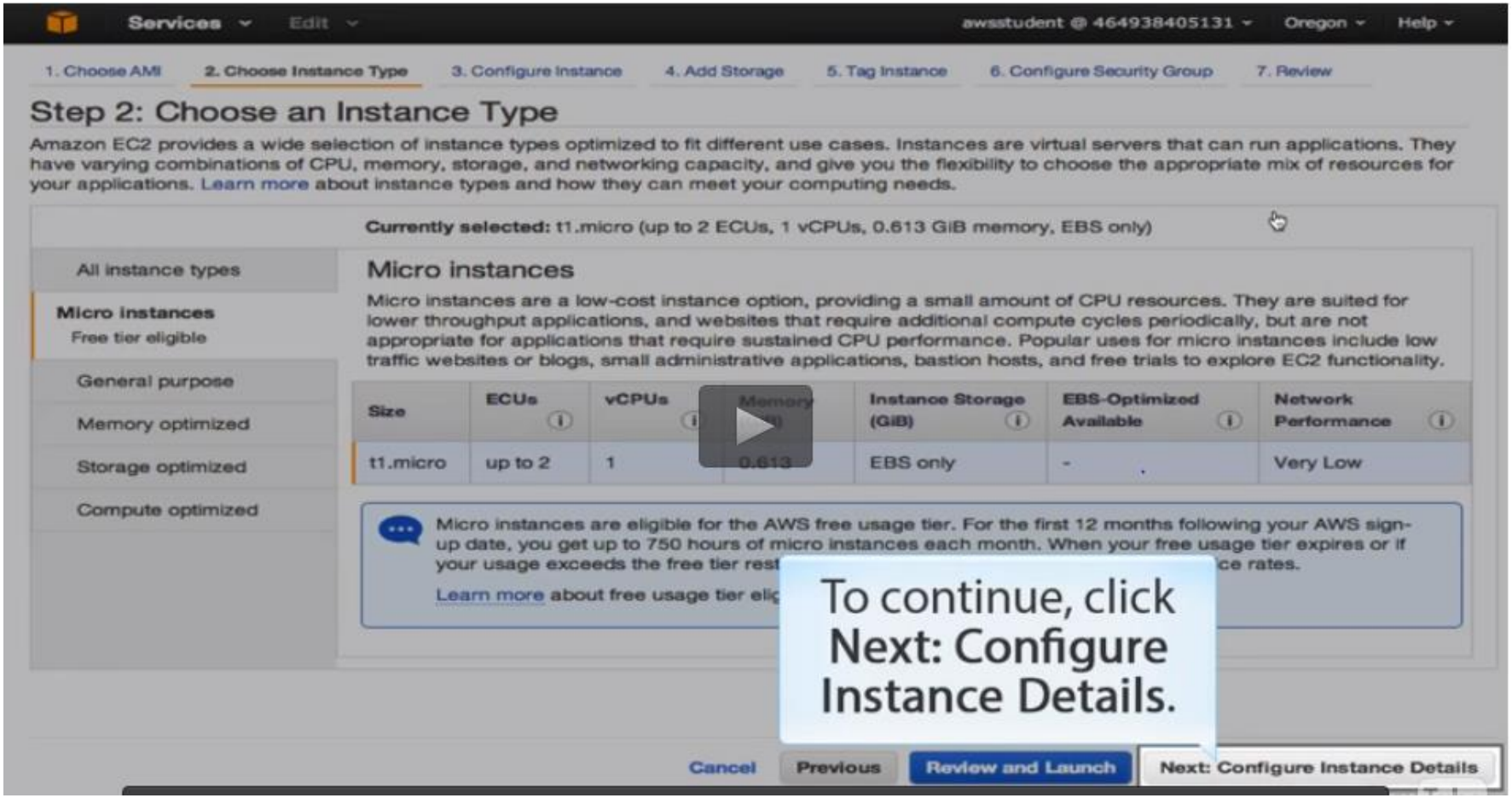
Select AWS Marketplace and search for "Tomcat powered by Bitnami" and press select AMI that has Tomcat and MySQL, as shown below



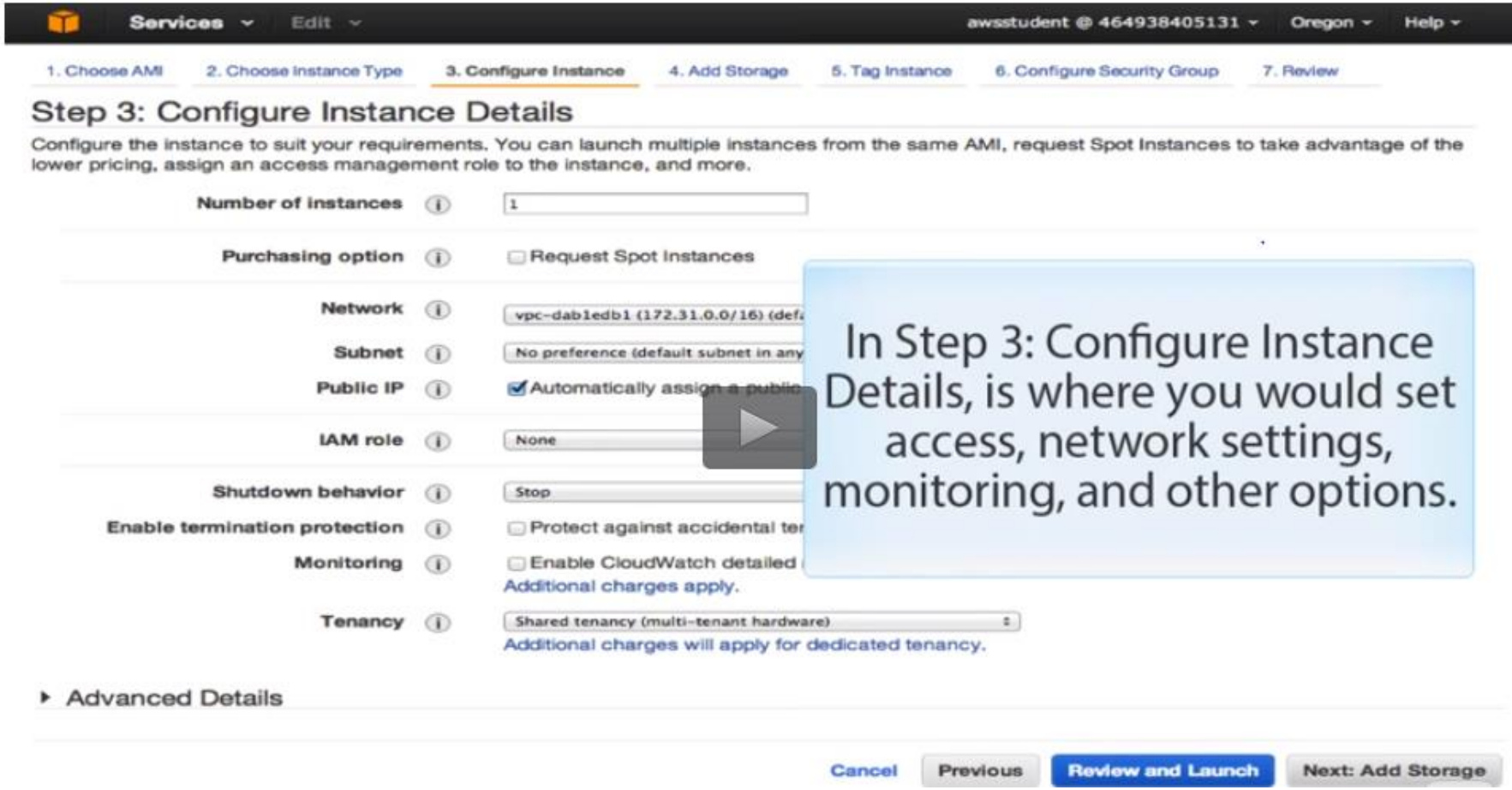
An example to create, find, deploy, and terminate an EC2 instance



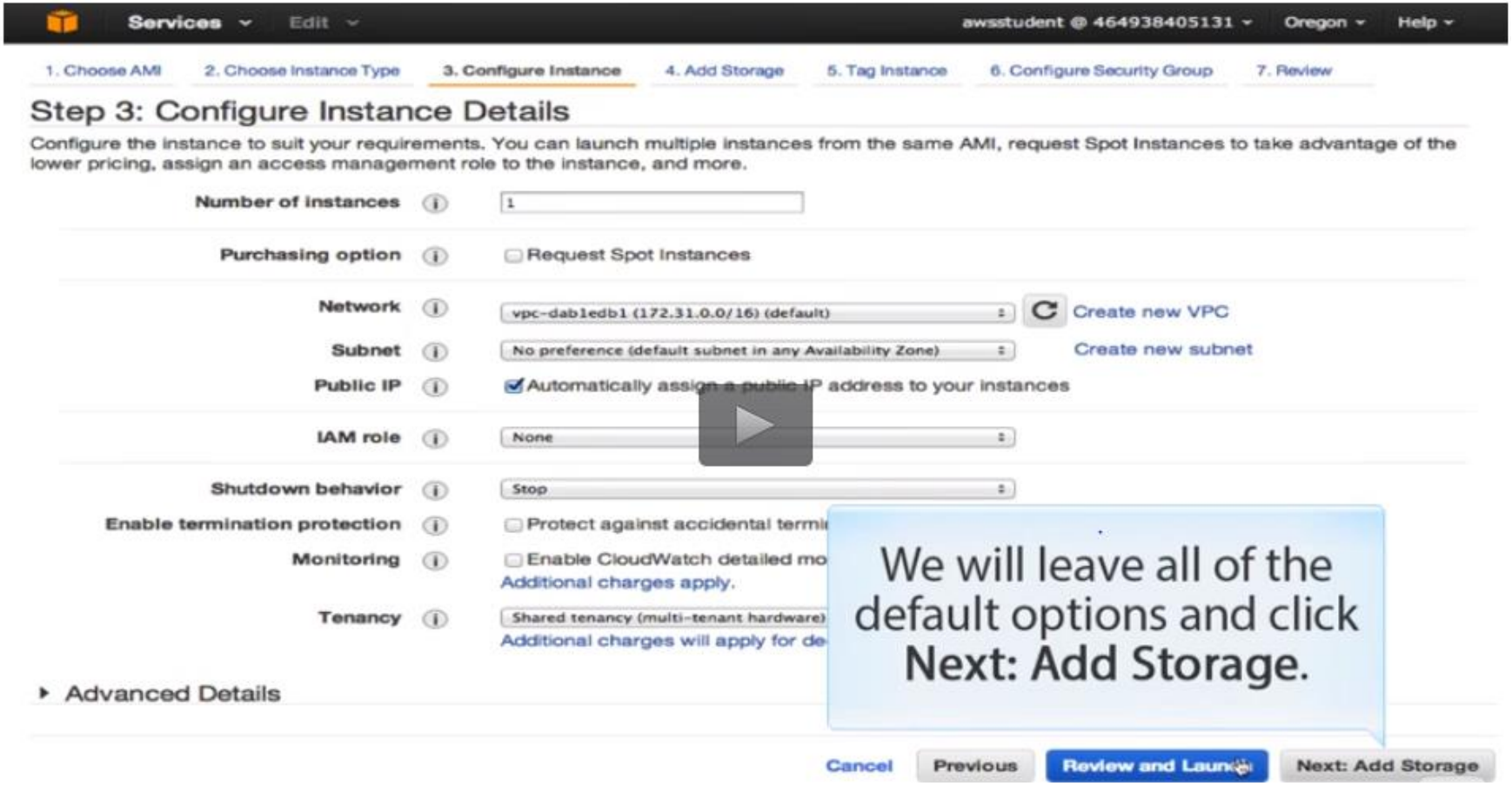
An example to create, find, deploy, and terminate an EC2 instance



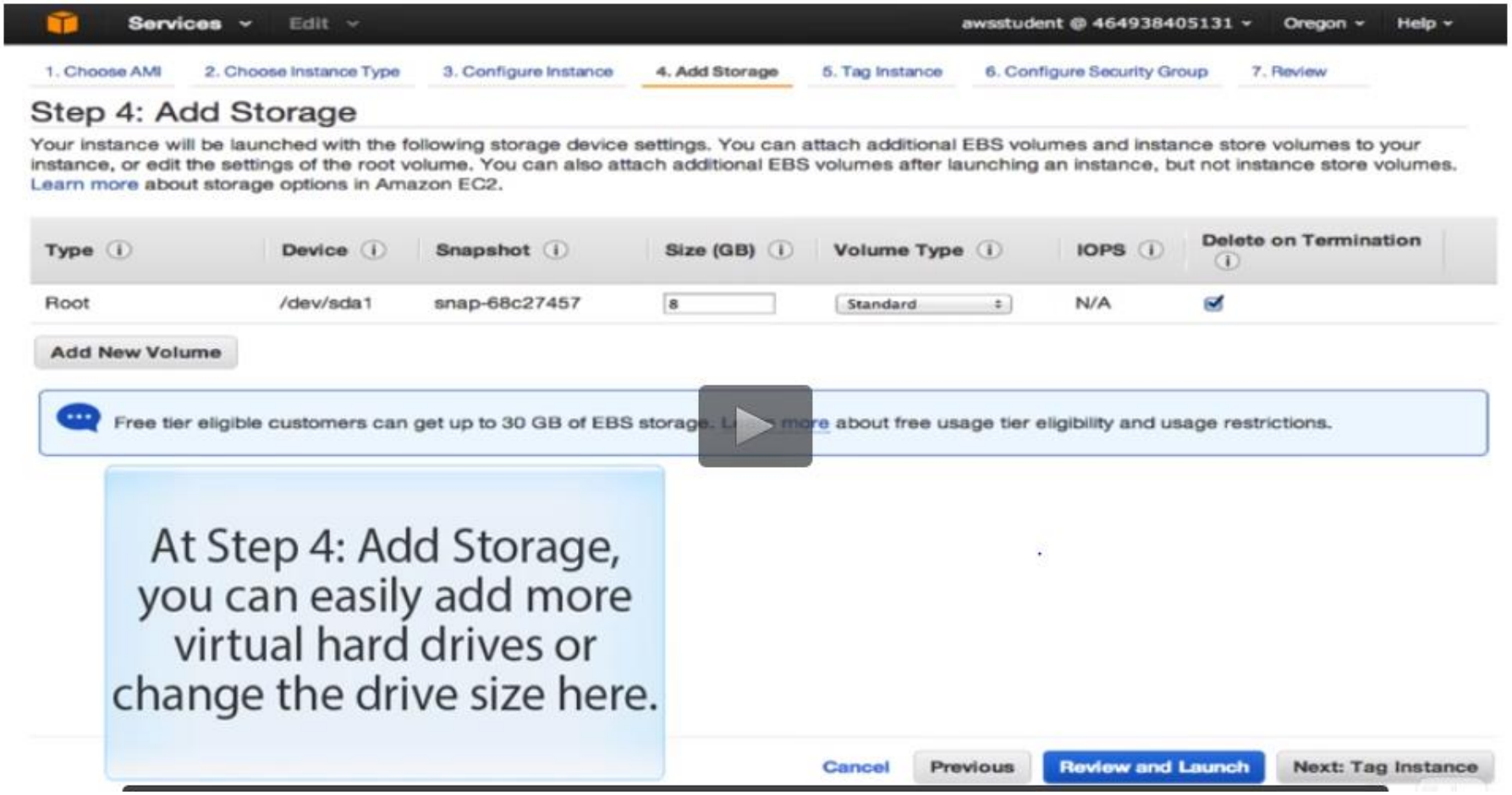
An example to create, find, deploy, and terminate an EC2 instance



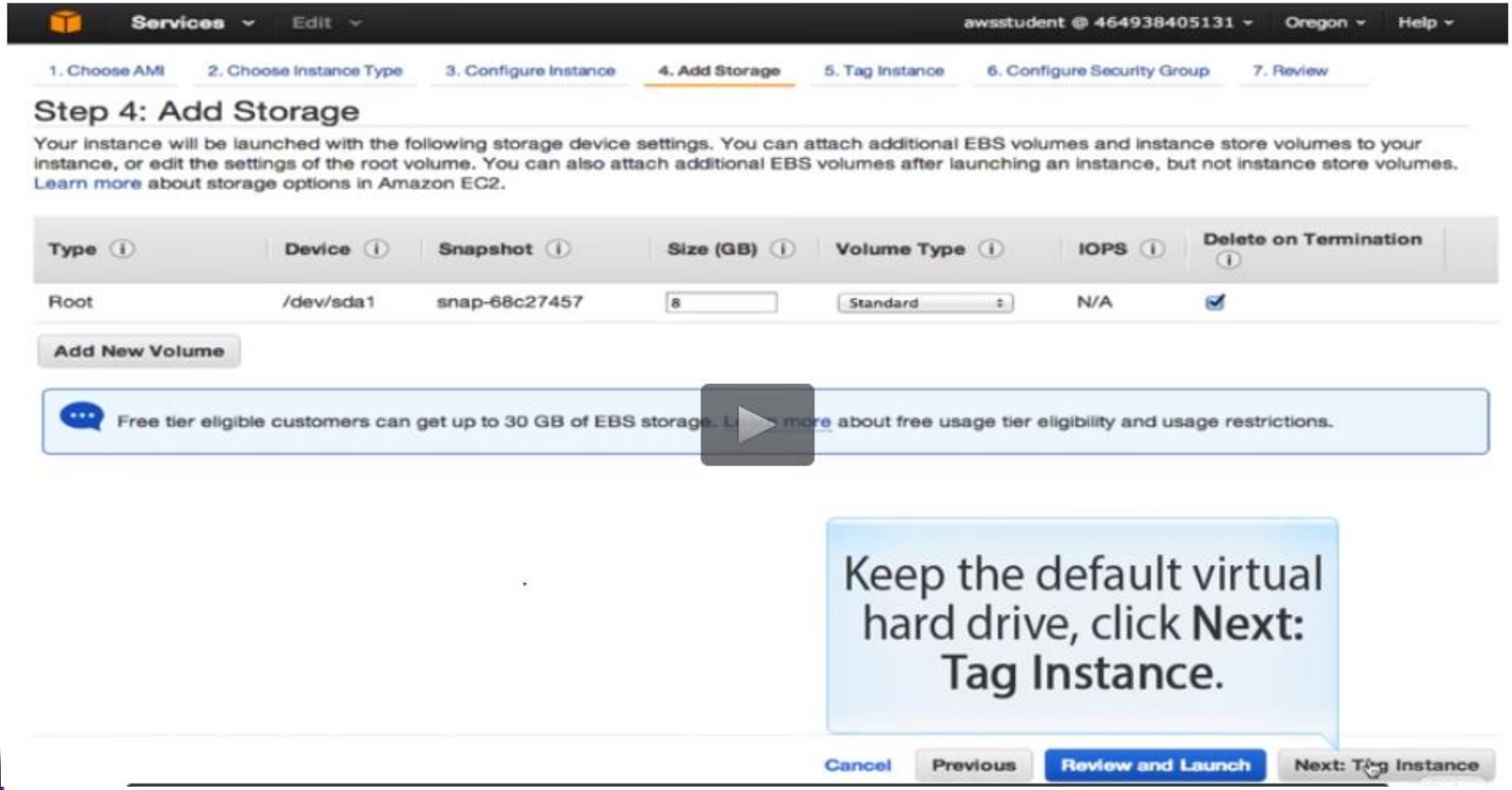
An example to create, find, deploy, and terminate an EC2 instance



An example to create, find, deploy, and terminate an EC2 instance

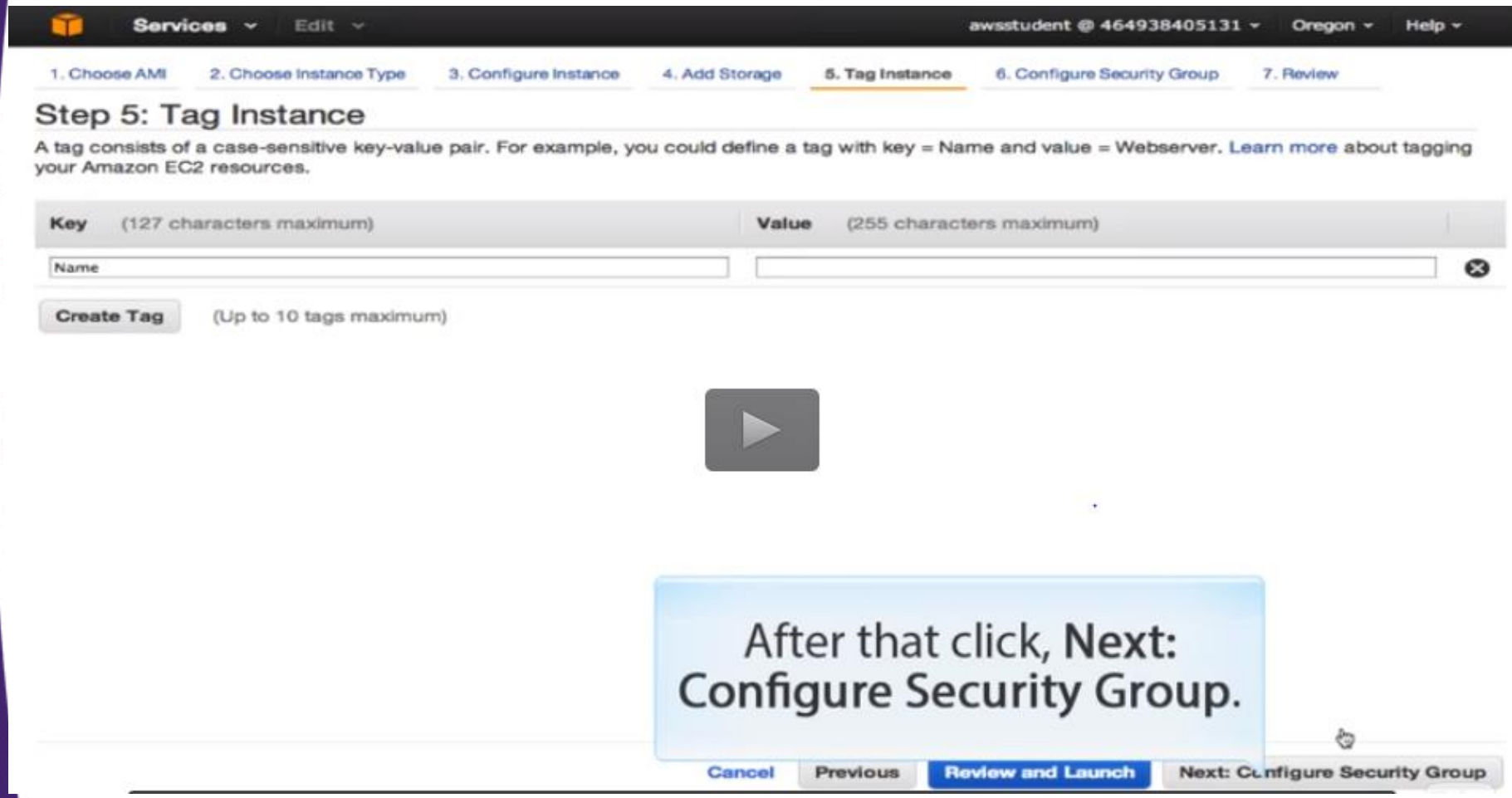


An example to create, find, deploy, and terminate an EC2 instance

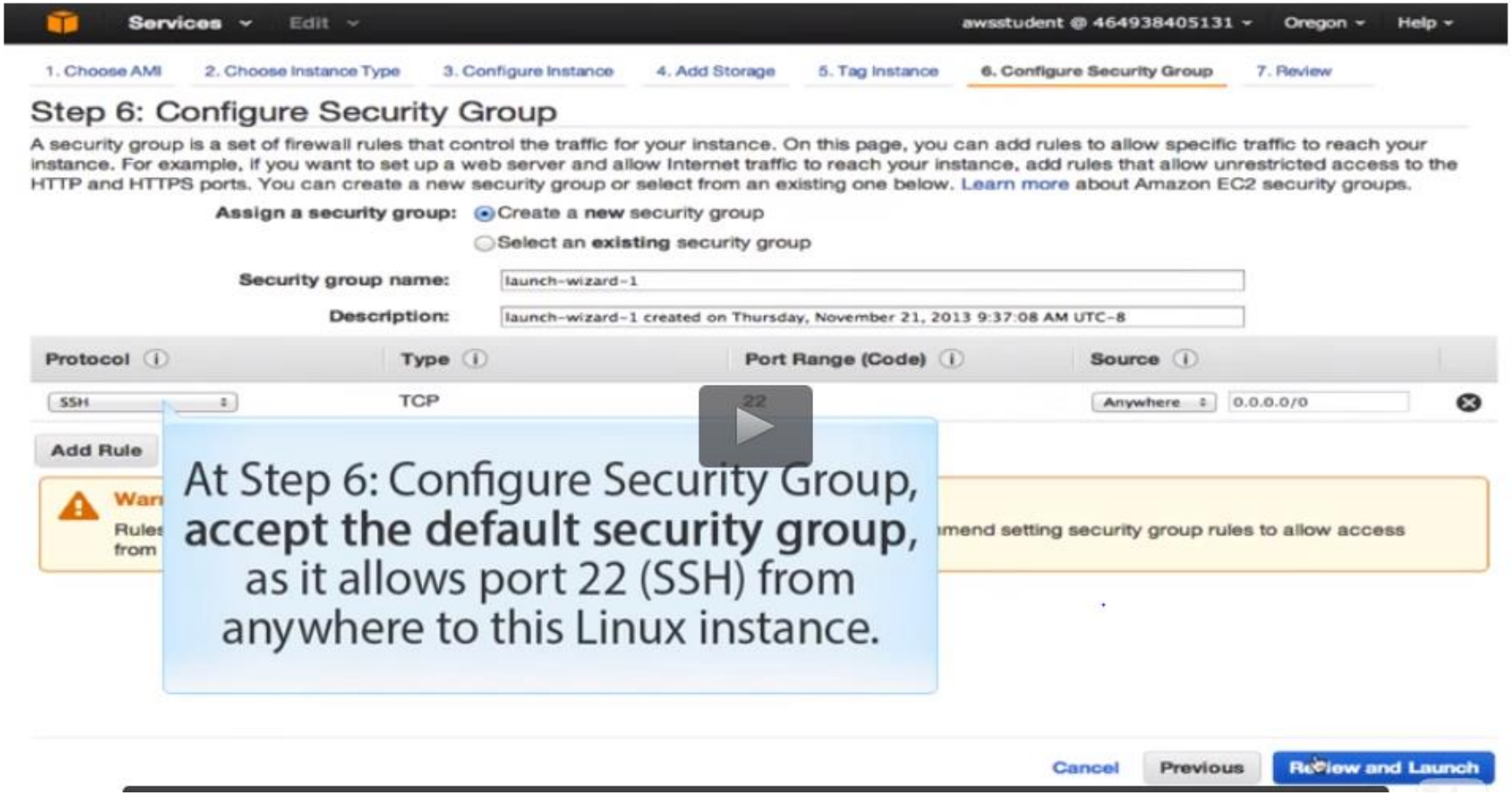


An example to create, find, deploy, and terminate an EC2 instance

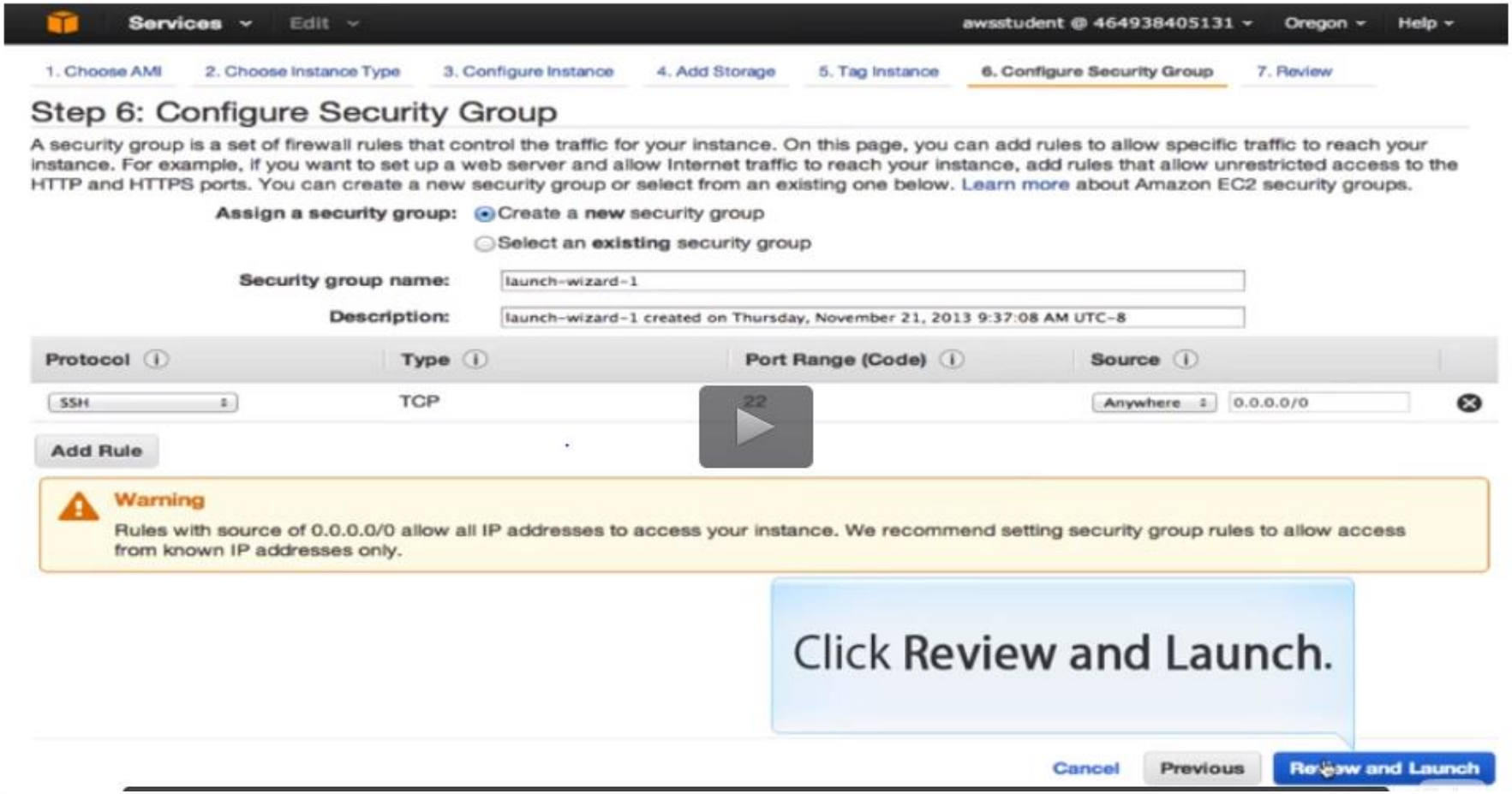
An example to create, find, deploy, and terminate an EC2 instance



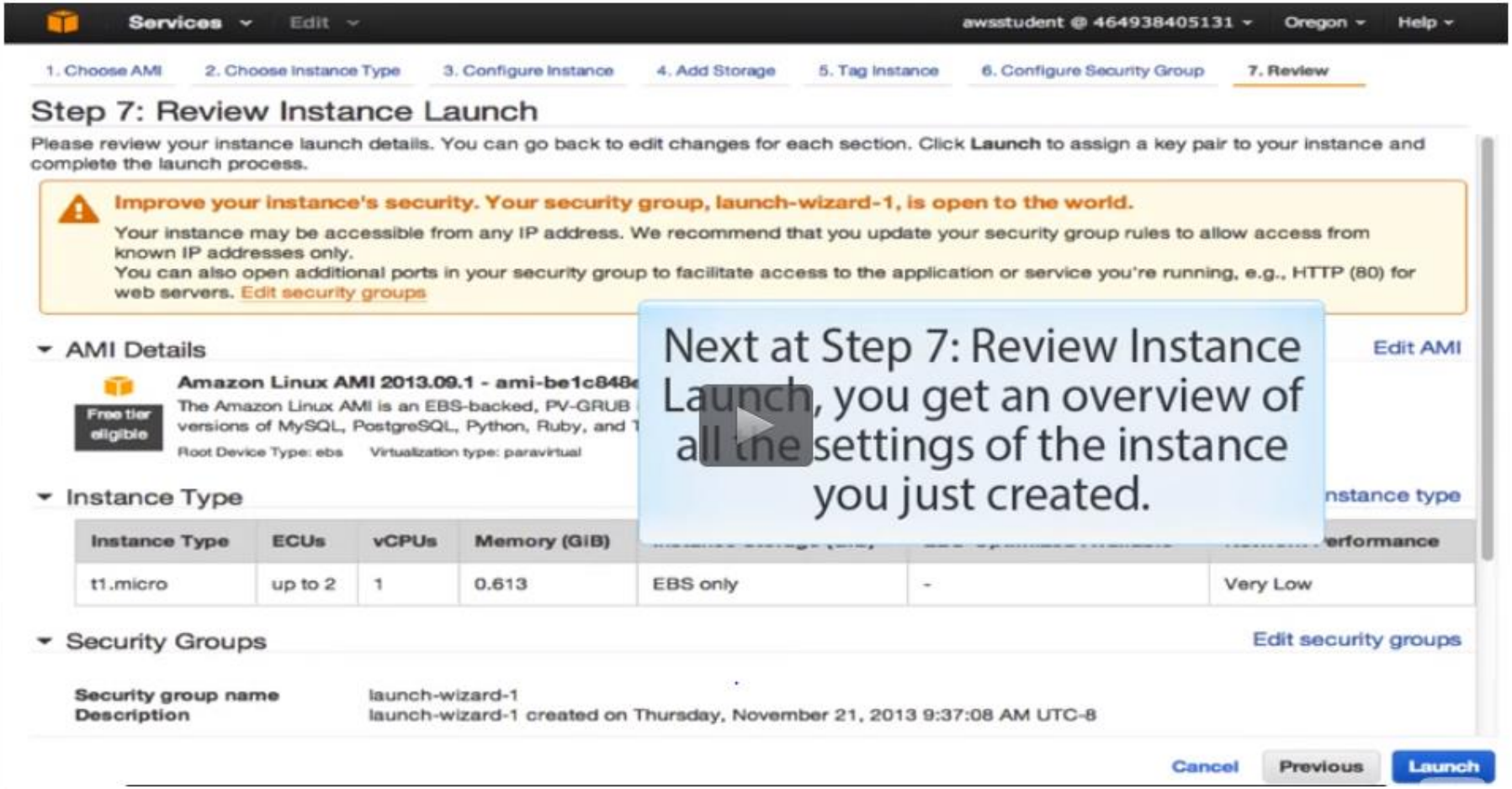
An example to create, find, deploy, and terminate an EC2 instance



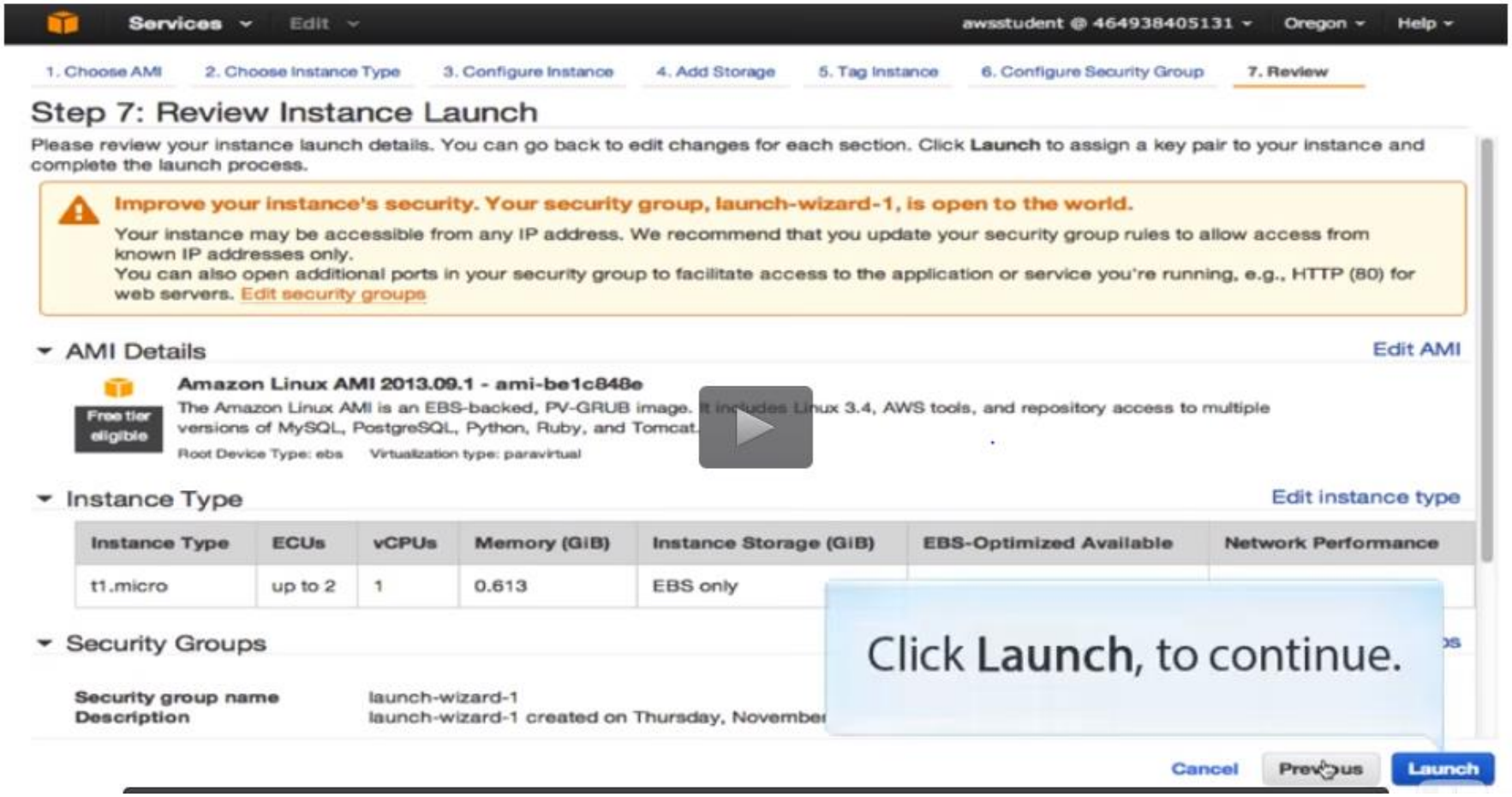
An example to create, find, deploy, and terminate an EC2 instance



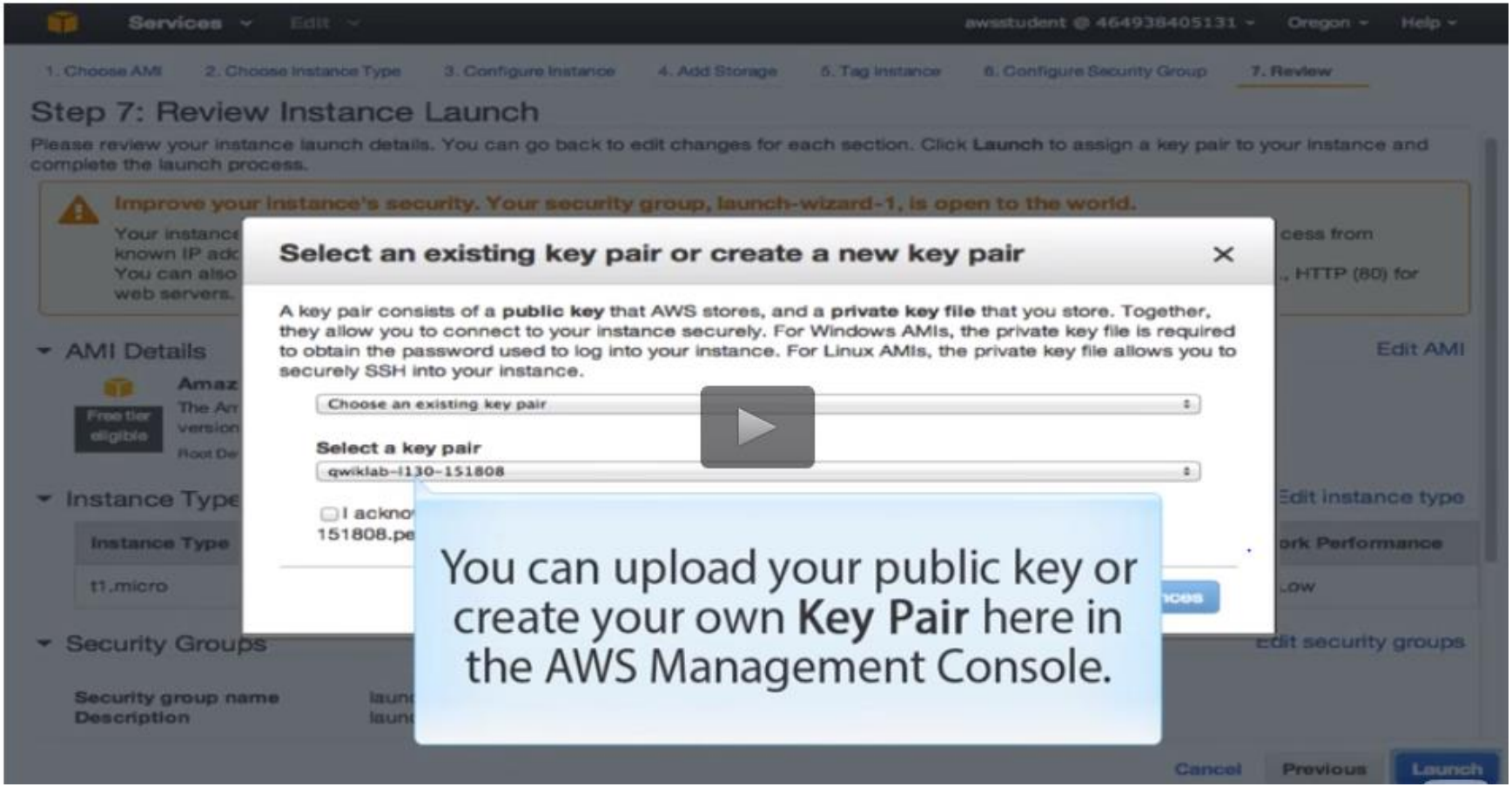
An example to create, find, deploy, and terminate an EC2 instance



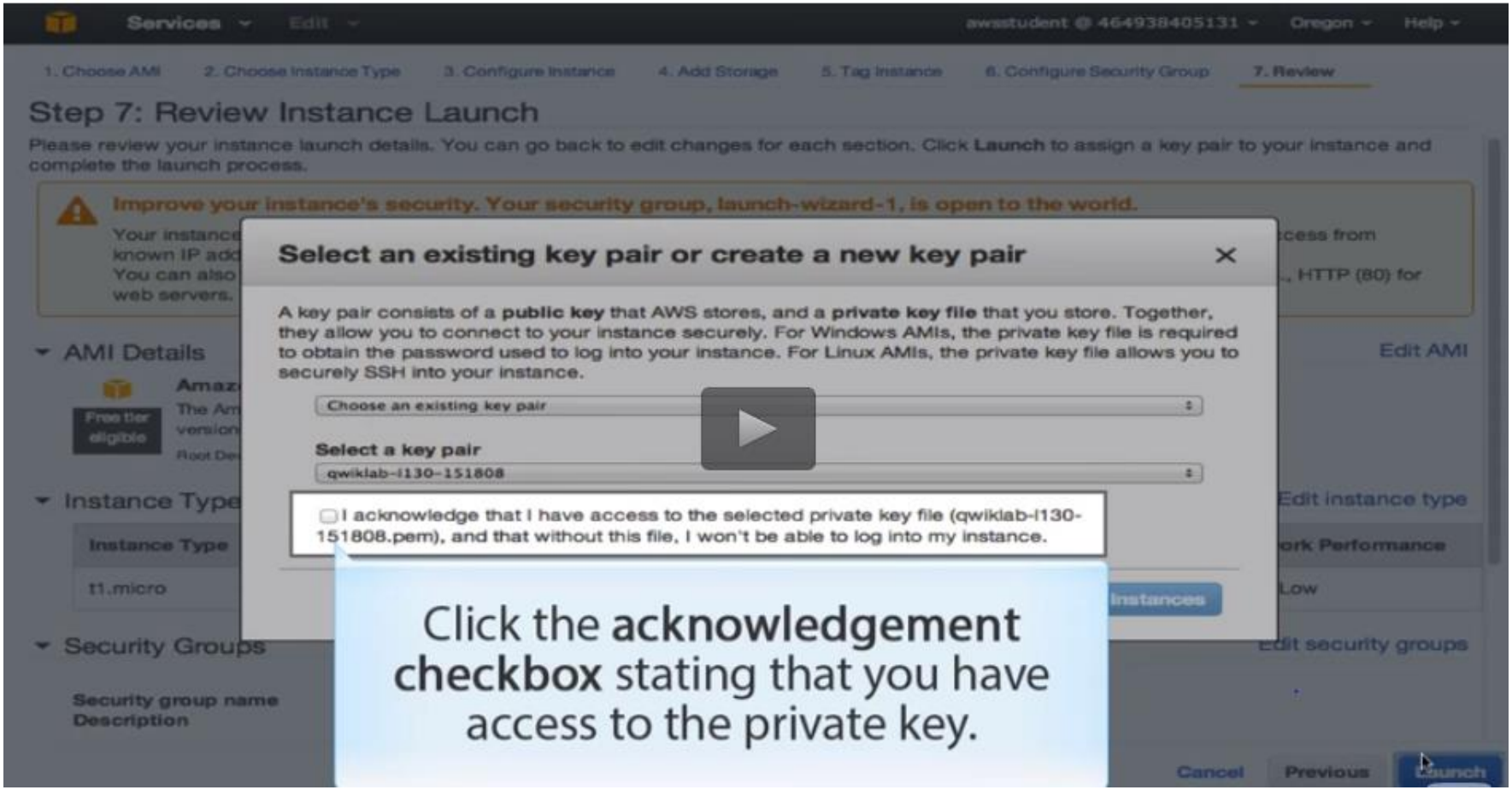
An example to create, find, deploy, and terminate an EC2 instance



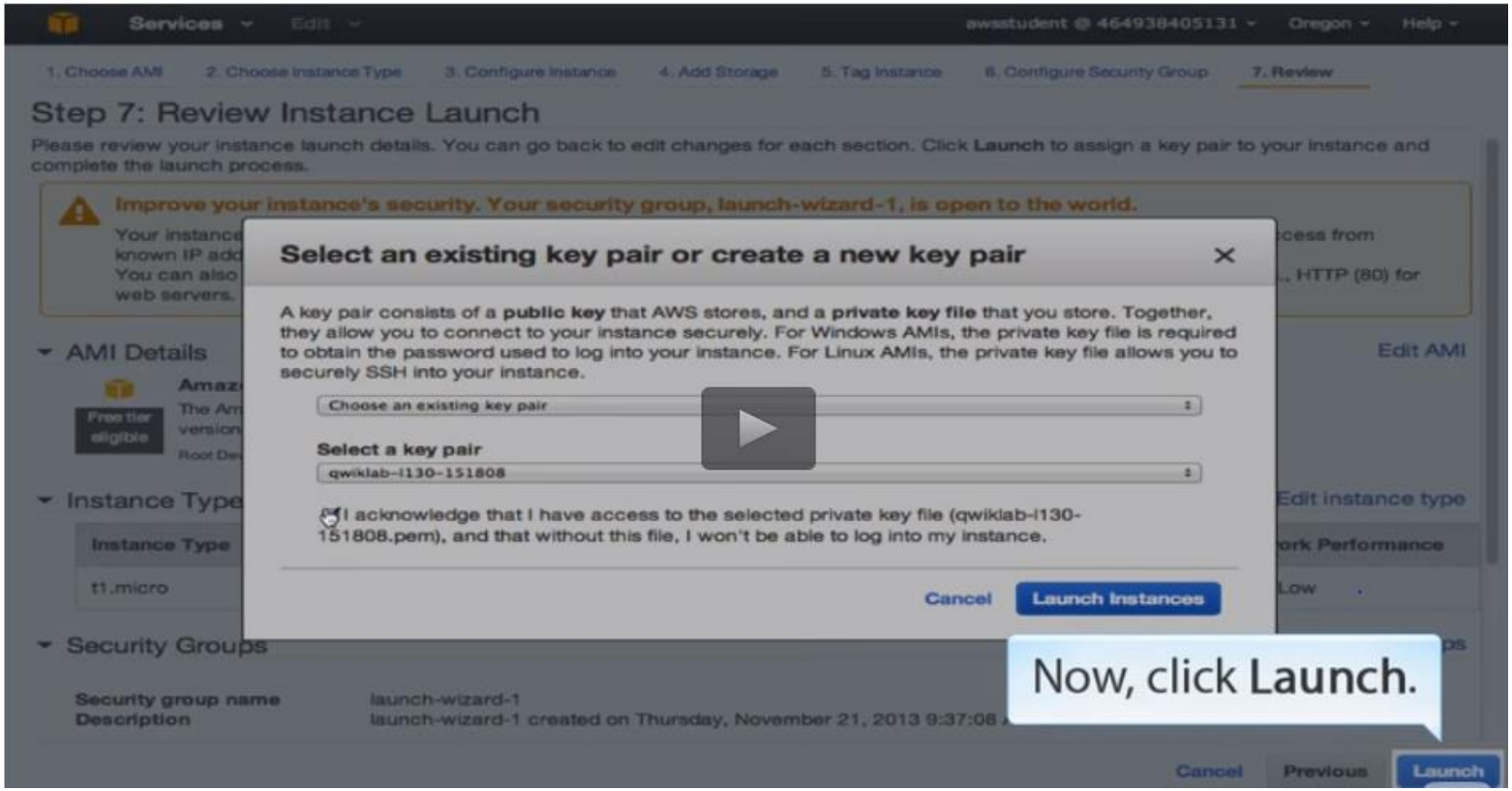
An example to create, find, deploy, and terminate an EC2 instance



An example to create, find, deploy, and terminate an EC2 instance

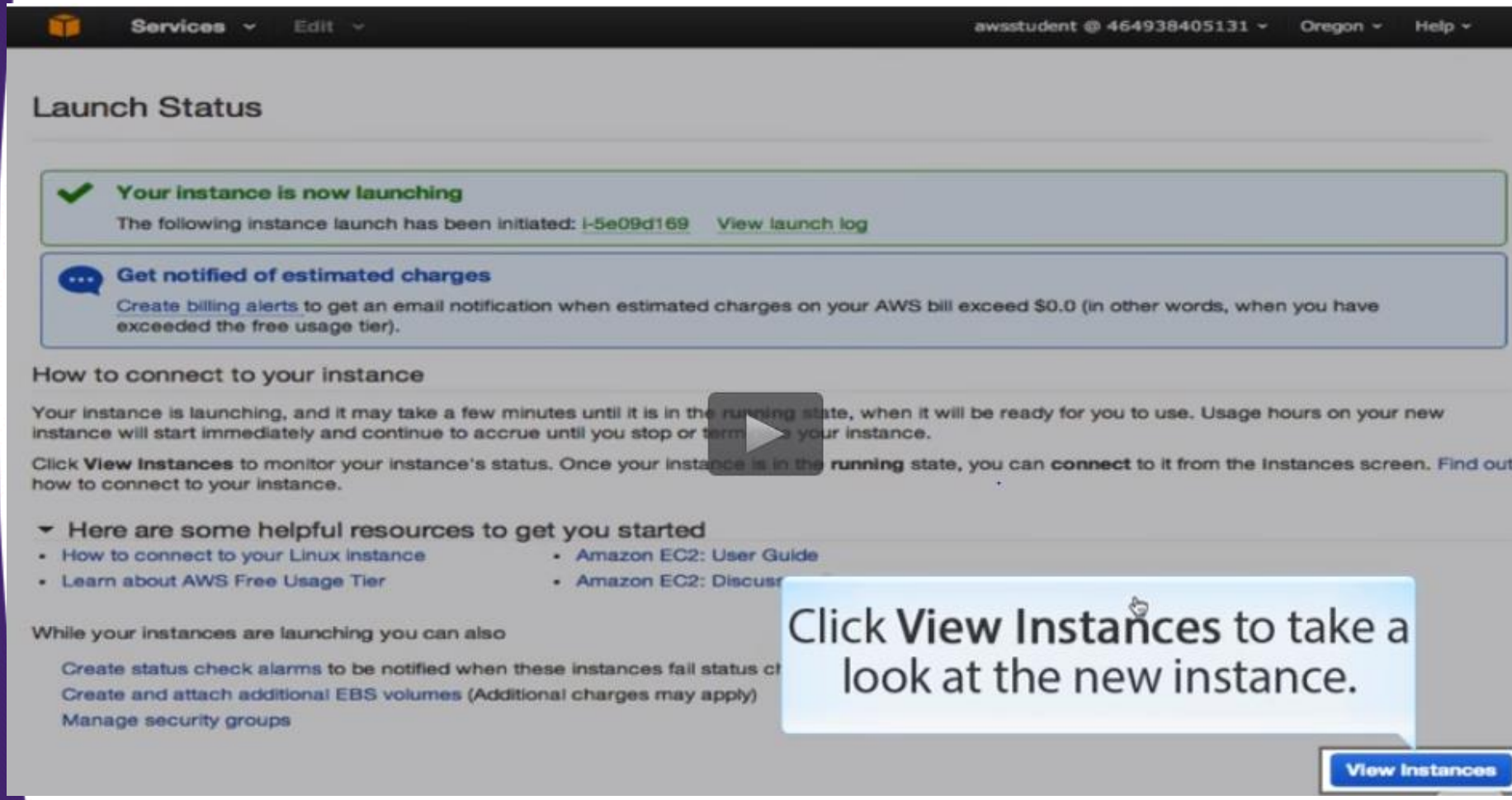


An example to create, find, deploy, and terminate an EC2 instance

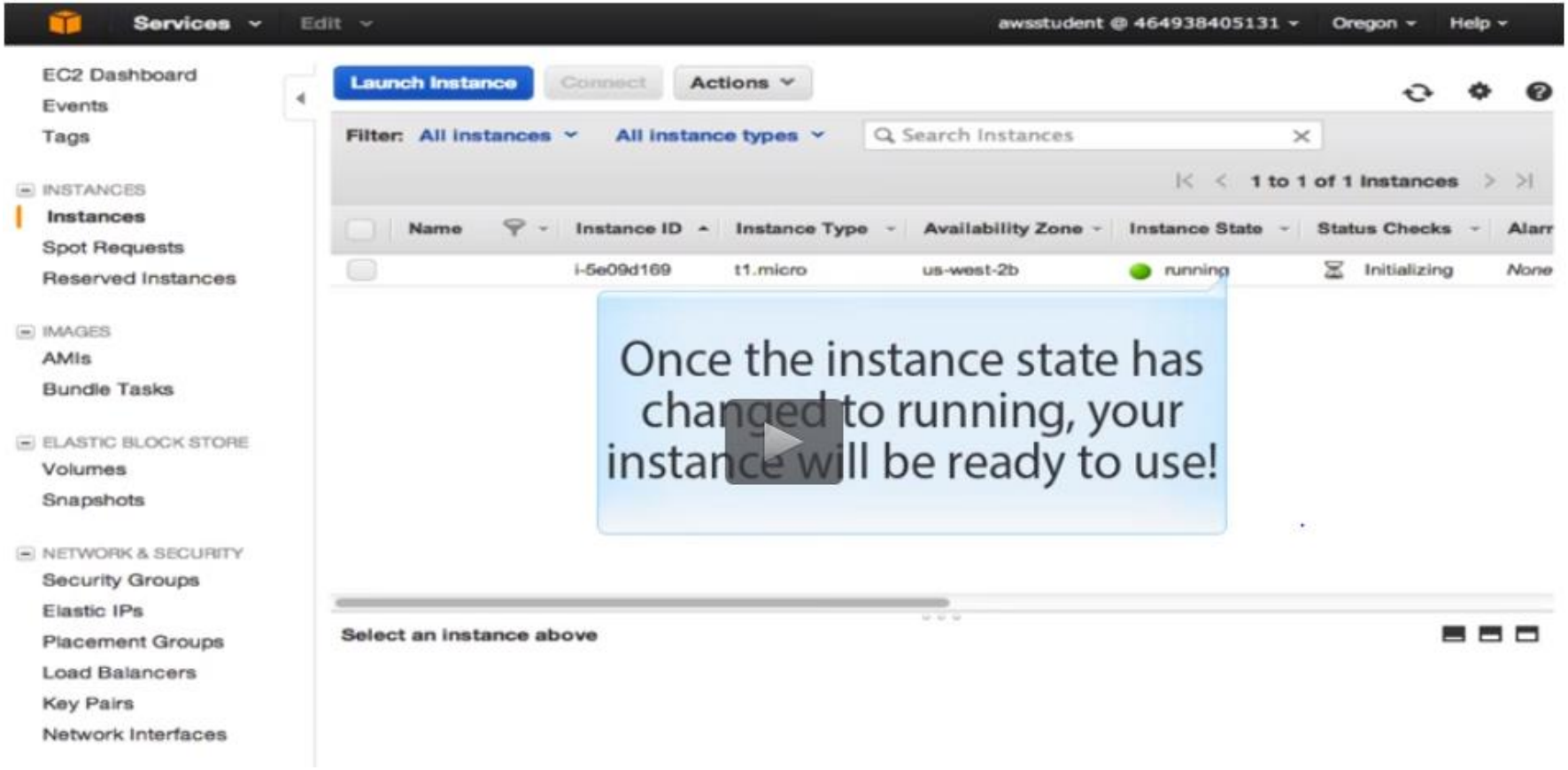


- "Key Pairs are security features that control access to your instance after it is created!"
- If you do not have right Key Pair, you will NOT be able to access the instance

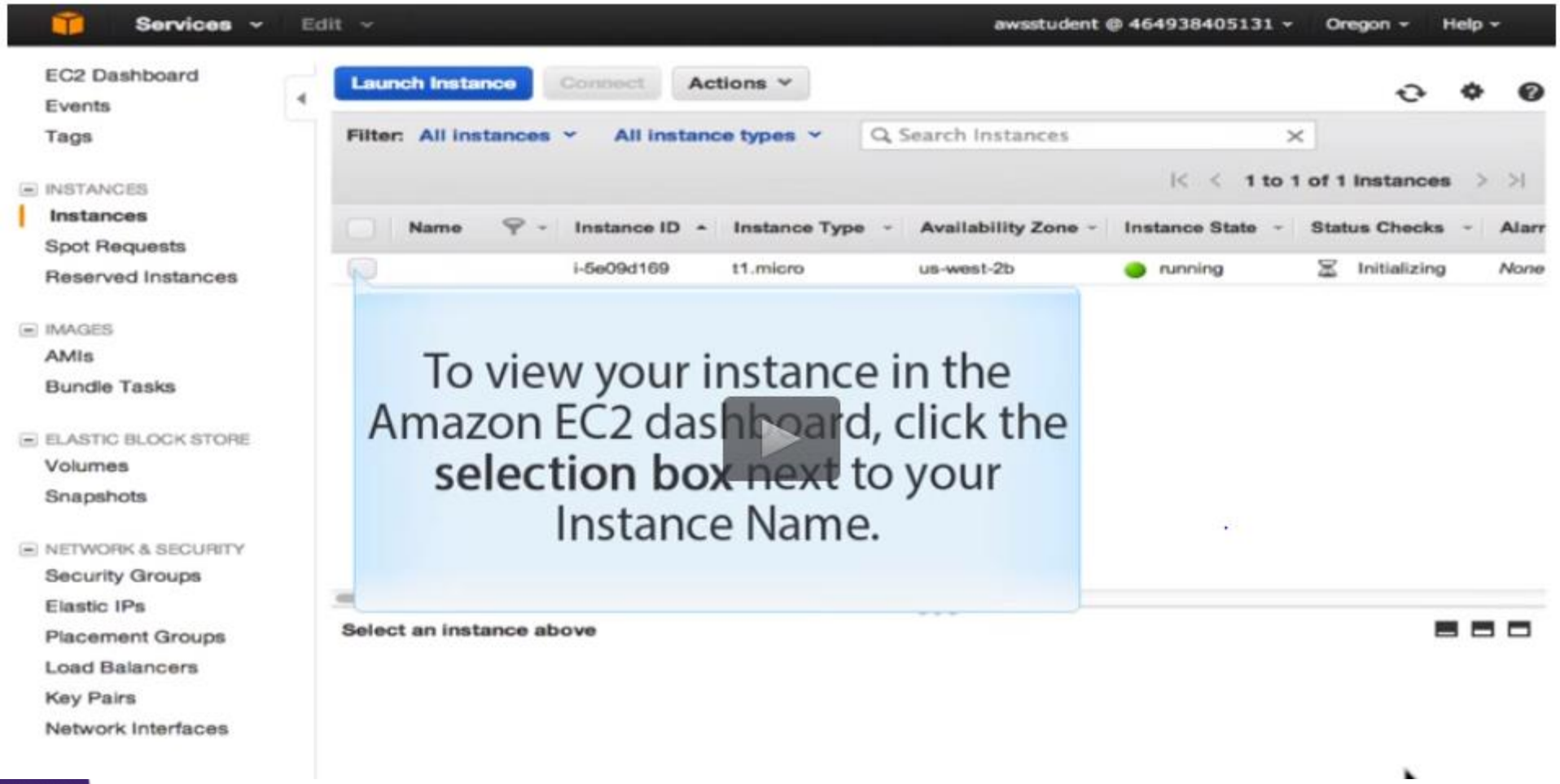
An example to create, find, deploy, and terminate an EC2 instance



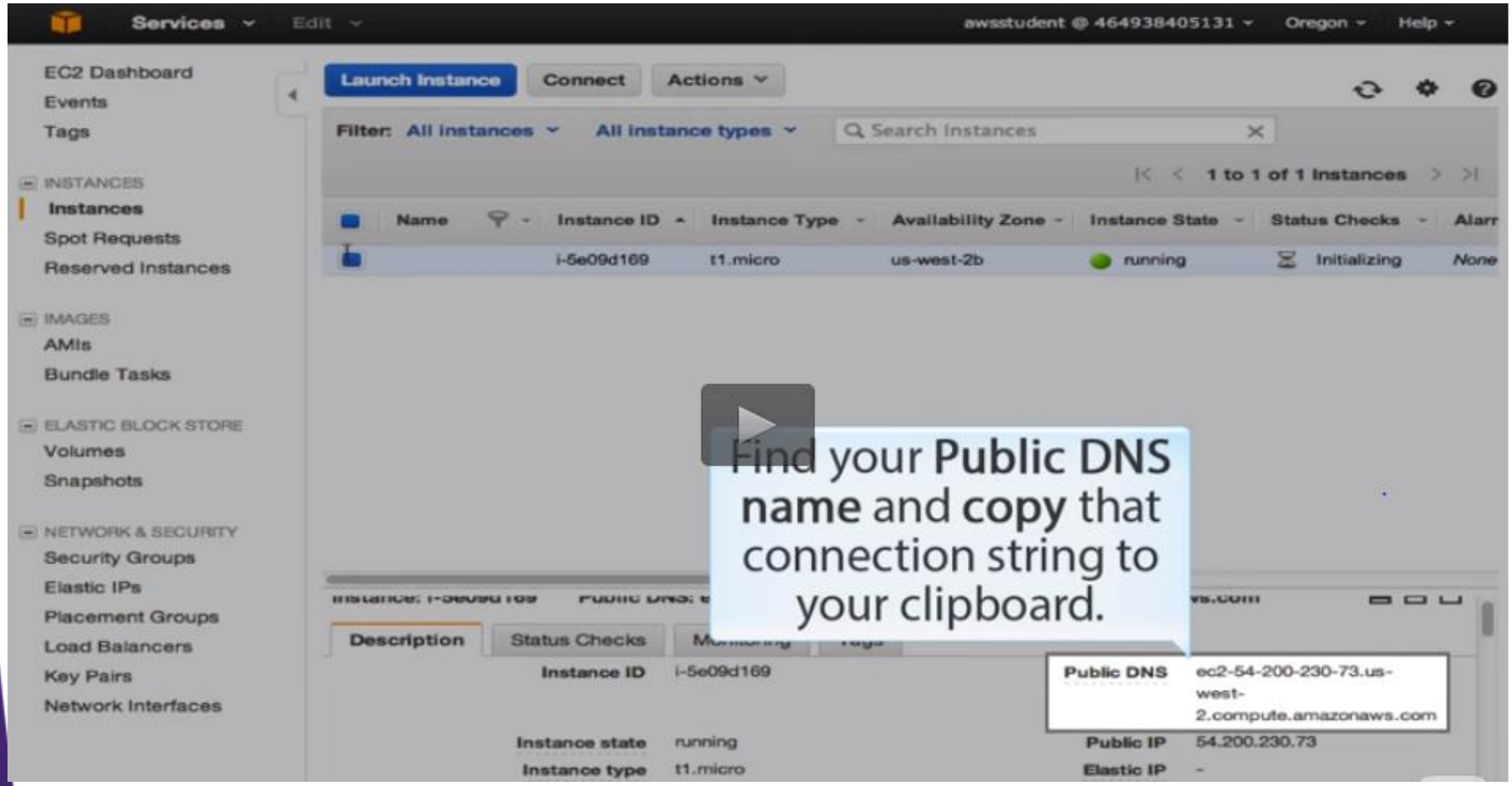
An example to create, find, deploy, and terminate an EC2 instance



An example to create, find, deploy, and terminate an EC2 instance



An example to create, find, deploy, and terminate an EC2 instance



The DNS can be used to access the system exactly as you would if it were in your own data center.

Elastic Block Store

- **An EBS volume is a virtual disk of a fixed size with a block read/write interface. It can be mounted as a file system on a running EC2 instance where it can be updated incrementally.**
- **Unlike an instance store, an EBS volume is persistent.**
- **Fundamental operations:**
  - CREATE a new volume (1GB-1TB)
  - COPY a volume from an existing EBS volume or S3 object.
  - MOUNT on one instance at a time.
  - SNAPSHOT current state to an S3 object.

Durability

- **Amazon claims about EBS:**
  - Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component.

Deploying a WAR file on EC2 instance from a Window-based PC

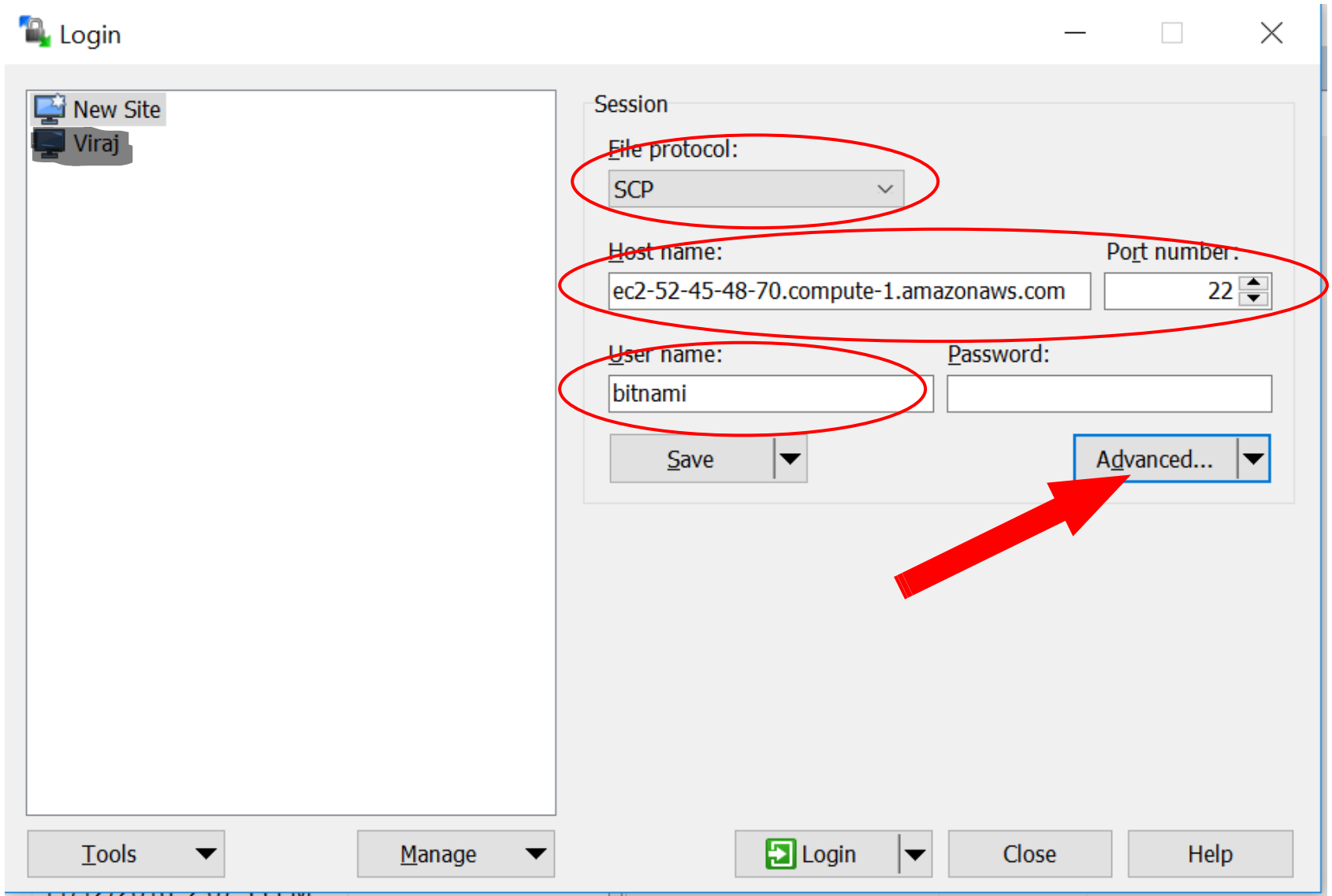
Deploying a WAR file on EC2 instance from a Window-based PC

- **Create an EC2 Instance using an AMI with Tomcat preinstalled**
  - Use an EC2 Amazon Machine Image (AMI) labeled "Tomcat Certified by Bitnami" found under AWS Marketplace to launch your EC2 instance.
  - Provision a free tier eligible Amazon EC2 instance using an Amazon Machine Image (AMI) that may already have Tomcat configured. For example, once you are on EC2 Console, click Launch Instance link. For the AMI, search for key word "Tomcat" in AWS Marketplace and select the Amazon machine image labeled something like "Tomcat Certified by Bitnami" – the one labelled as free tier eligible.
  - Follow the rest of the steps as usual

Deploying a WAR file on EC2 instance from a Window-based PC

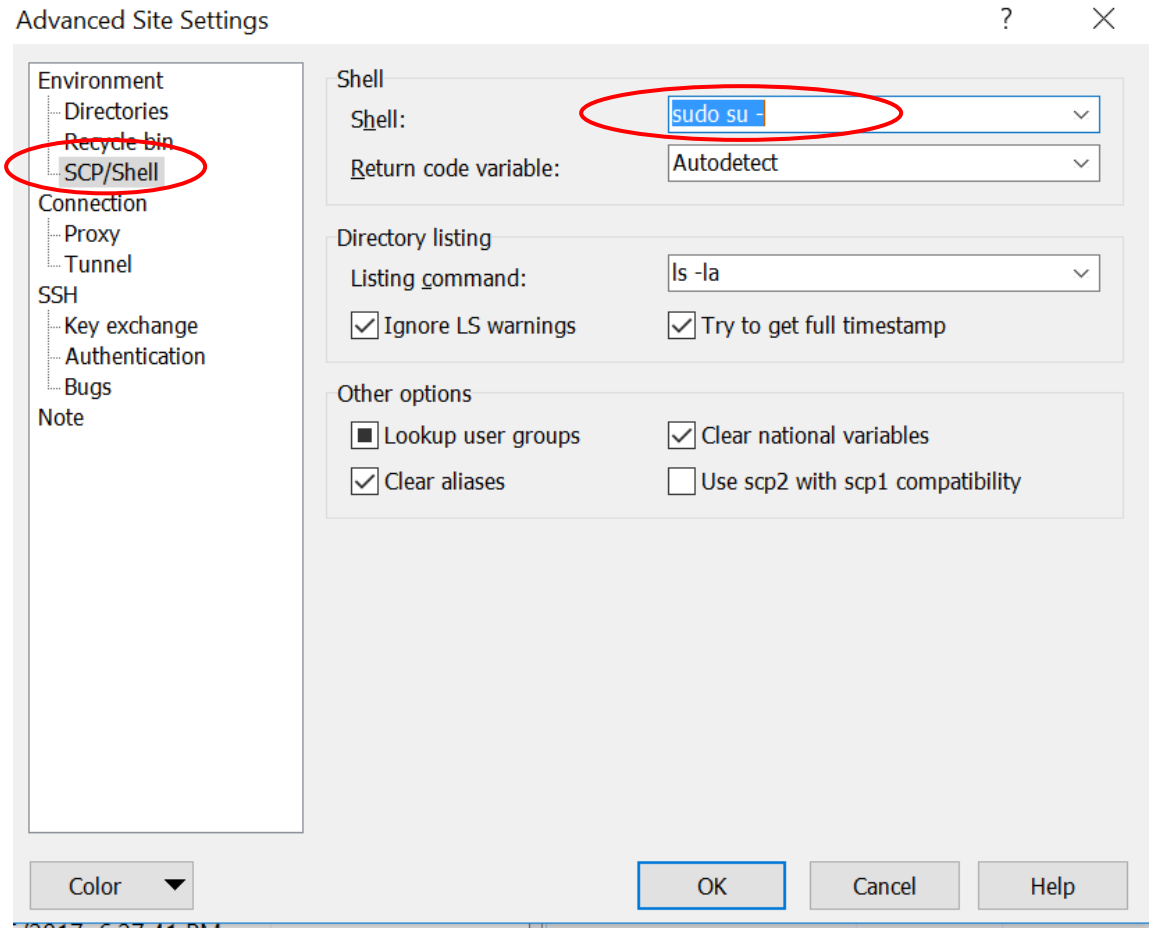
- **Download a FTP client**
  - This demonstration will use WinSCP (download here: https://sourceforge.net/projects/winscp/files/latest/download)
- **Setup a connection to your EC2 instance:**
  - Copy the public DNS of your instance into hostname
  - Select SCP as the file protocol
  - Type in ‘bitnami’ as the username
  - Then select the advanced settings

Setup WinSCP connection



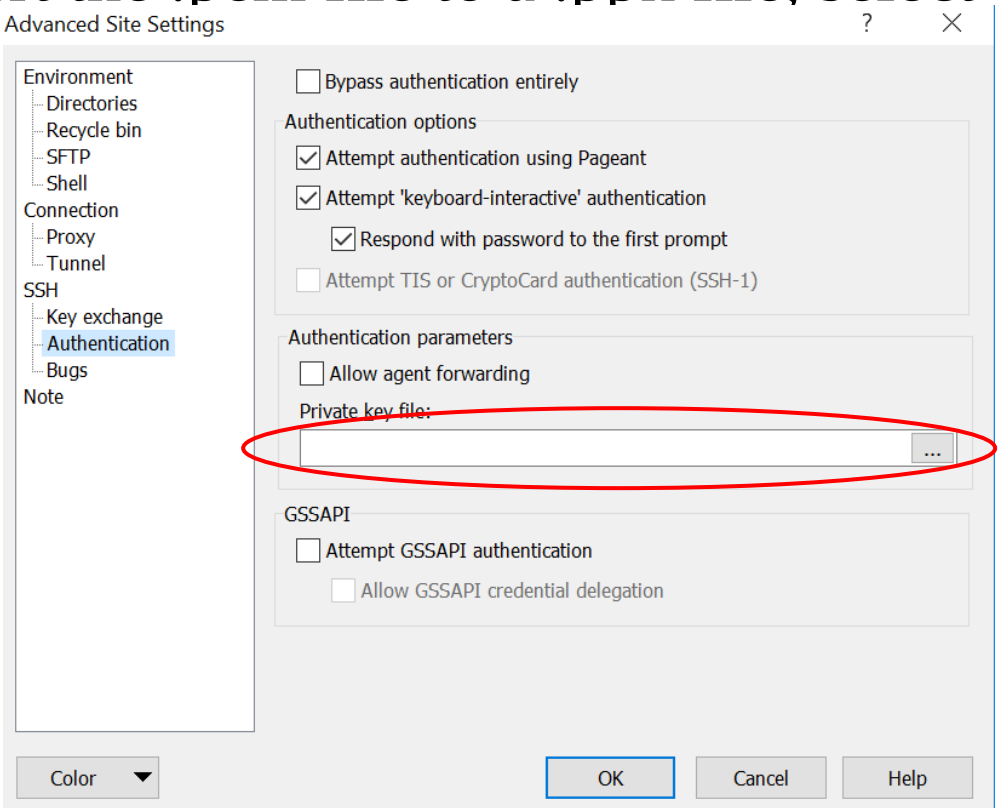
Advanced setting

- **Select sudo su – shell to get super user access on the AMI**



Advanced Settings

- **In authentication, browse to the path of the .pem file of your instance.**
  - On clicking ‘OK’ WinSCP will ask for permission to convert the .pem file to a .ppk file, select OK.



Connect to your instance

- **Save your settings and connect to your instance**
- **Copy your WAR file to the** /opt/bitnami/tomcat/webapps **folder**
- **Access your project through the browser.**

Warmup: Get Started with Amazon

- **Skim through the AWS documentation**
  https://aws.amazon.com/documentation/
- **Sign up for AWS at** http://aws.amazon.com
- **Create a bucket in S3 and upload/download some files.**
- **Create and download a Key-Pair, save it in your home directory –** this can be done while EC2 instance (VM) provisioning.
- **Create an EC2 instance via the AWS Console**
- **Connect to your newly-created EC2 instance like this** **(applicable if accessing from Mac or Unix/Linux.**
  - ssh -i my-aws-keypair.pem ec2-user@ip-address-of-vm
  - When accessing from a Microsoft Window, you will need to use putty and/or WinSCP

Demo Time

http://aws.amazon.com

Instructional Videos and Labs

https://aws.amazon.com/training/intro_series/