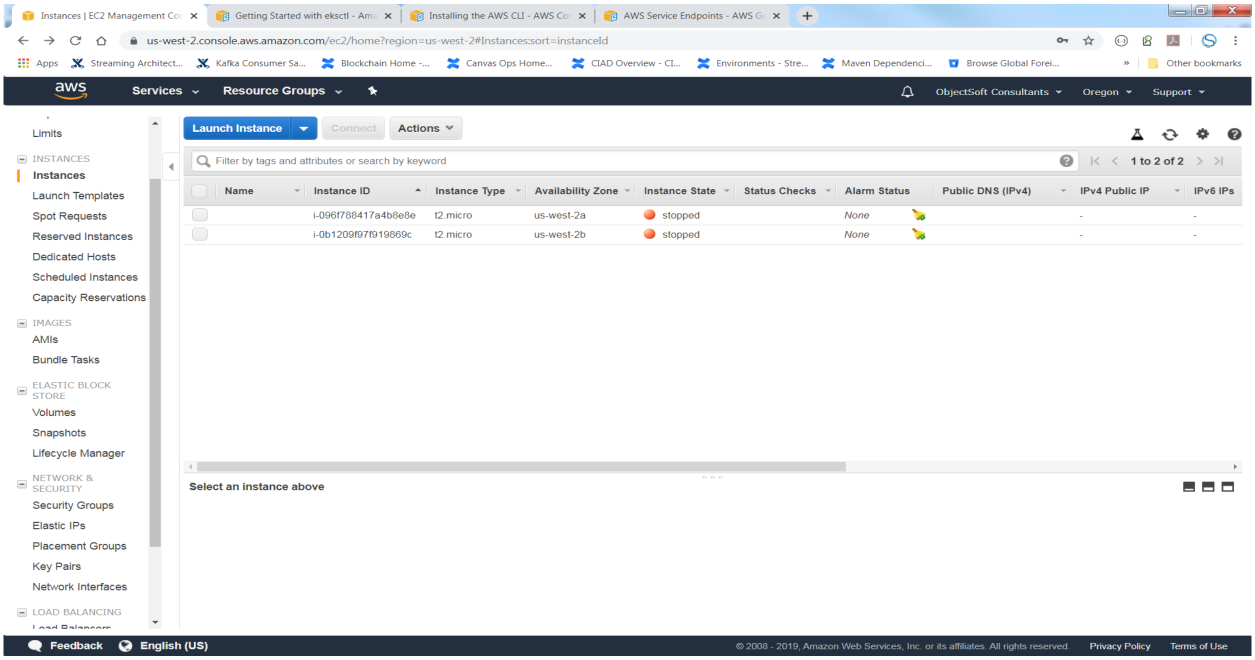
Log into AWS account, select Oregon region (us-west-2) and click on Launch Instance button.



us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#Instances:sort=instanceId

Filter by tags and attributes or search by keyword

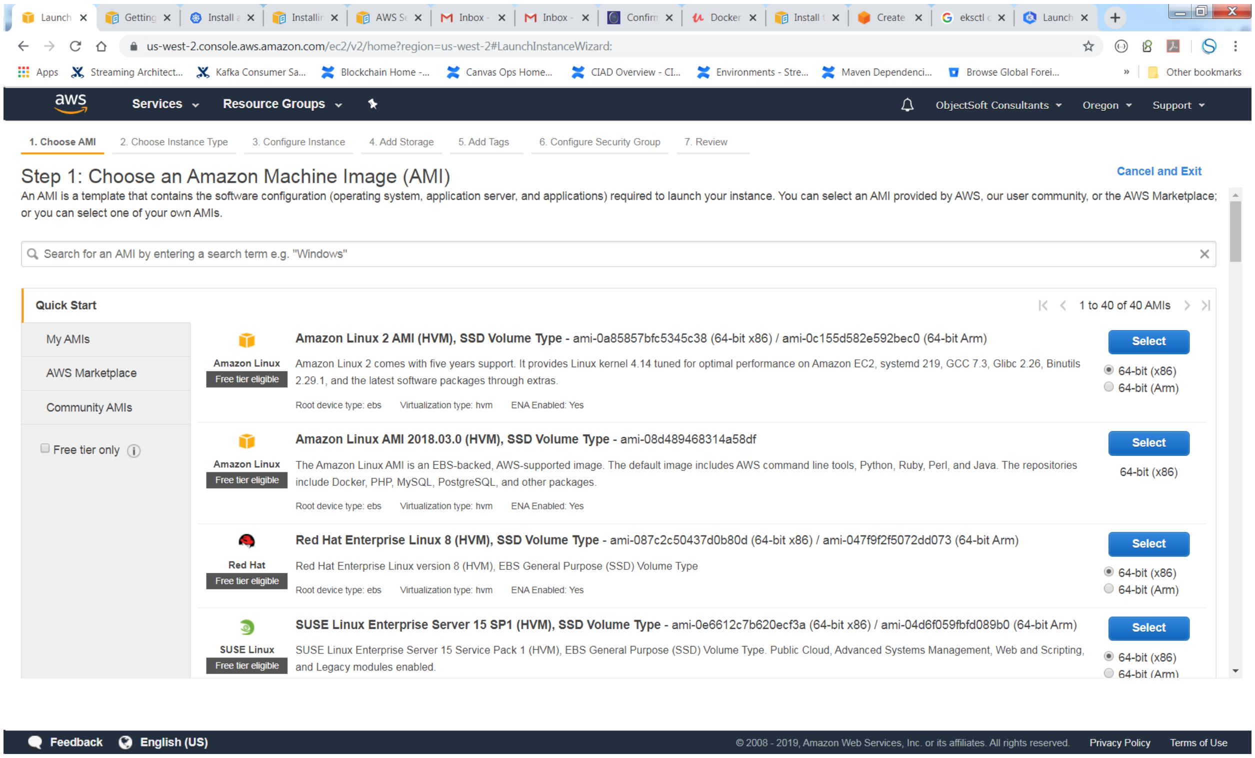
1 to 2 of 2

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
	i-096f788417a4b8e8e	t2.micro	us-west-2a	stopped		None		-	-
	i-0b1209f97f919869c	t2.micro	us-west-2b	stopped		None		-	-

Select an instance above

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Select Amazon Linux 2 AMI:



us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

1 to 40 of 40 AMIs

Amazon Linux — Free tier eligible
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0a85857bfc5345c38 (64-bit x86) / ami-0c155d582e592bec0 (64-bit Arm)
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select ● 64-bit (x86) ○ 64-bit (Arm)

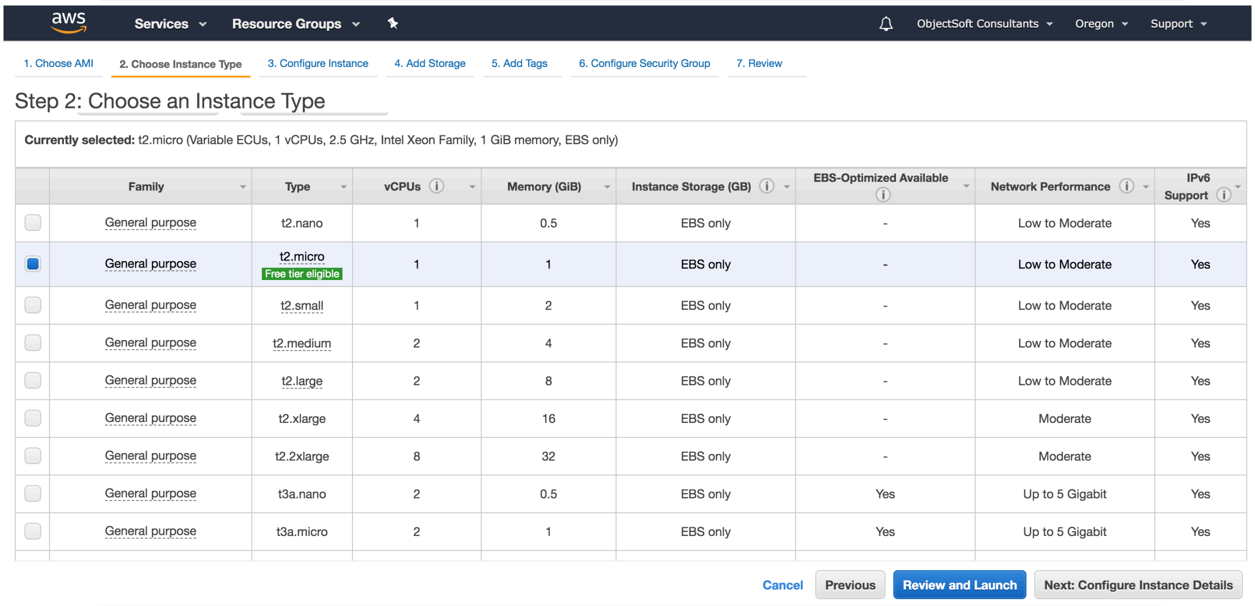
Amazon Linux — Free tier eligible
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-08d489468314a58df
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select 64-bit (x86)

Red Hat — Free tier eligible
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-087c2c50437d0b80d (64-bit x86) / ami-047f9f2f5072dd073 (64-bit Arm)
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select ● 64-bit (x86) ○ 64-bit (Arm)

SUSE Linux — Free tier eligible
SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-0e6612c7b620ecf3a (64-bit x86) / ami-04d6f059fbfd089b0 (64-bit Arm)
SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Select ● 64-bit (x86) ○ 64-bit (Arm)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Select t2.micro as instance type.

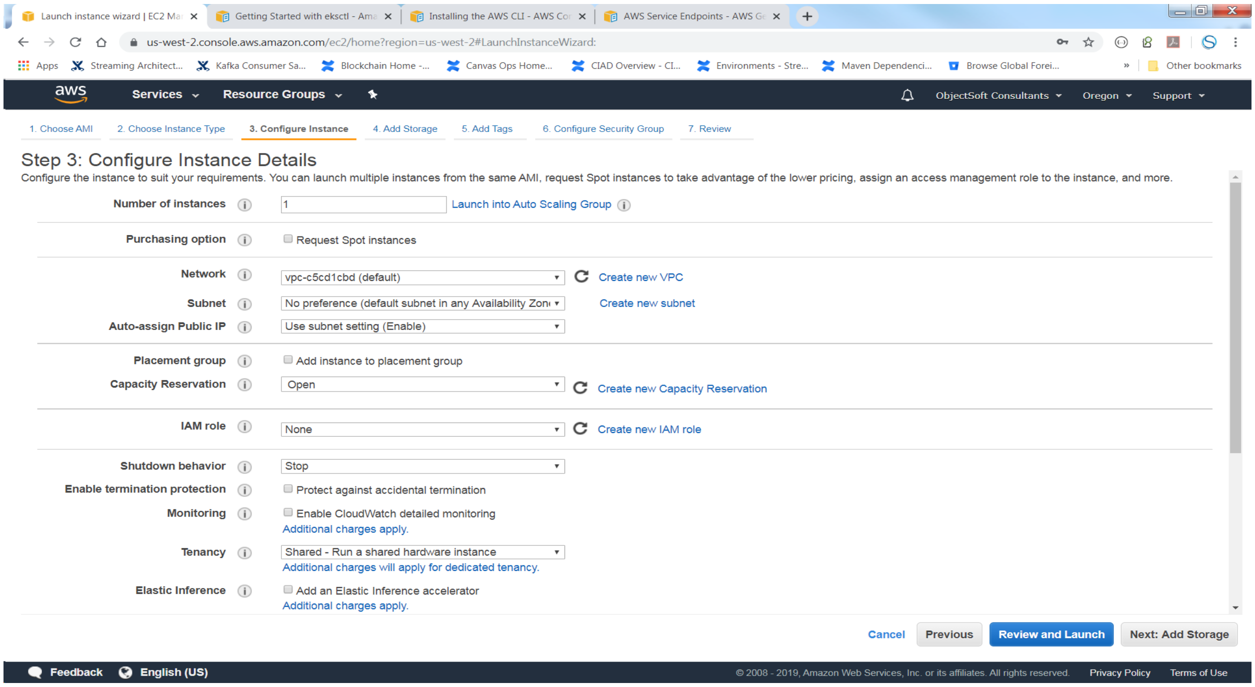


Step 2: Choose an Instance Type

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
□	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
□	General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes

Leave everything as default on Configure Instance Details Screen

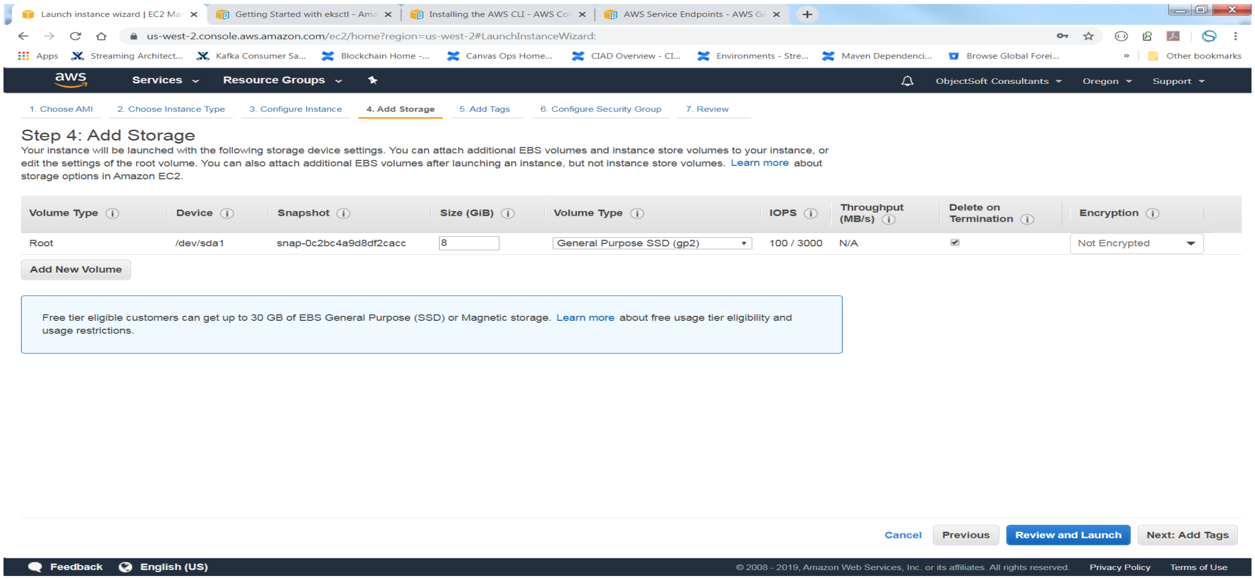


Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances	
Network	vpc-c5cd1cbd (default)	Create new VPC
Subnet	No preference (default subnet in any Availability Zon	Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	□ Add instance to placement group	
Capacity Reservation	Open	Create new Capacity Reservation
IAM role	None	Create new IAM role
Shutdown behavior	Stop	
Enable termination protection	□ Protect against accidental termination	
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.	
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.	
Elastic Inference	□ Add an Elastic Inference accelerator Additional charges apply.	

Leave everything default on Add Storage screen:



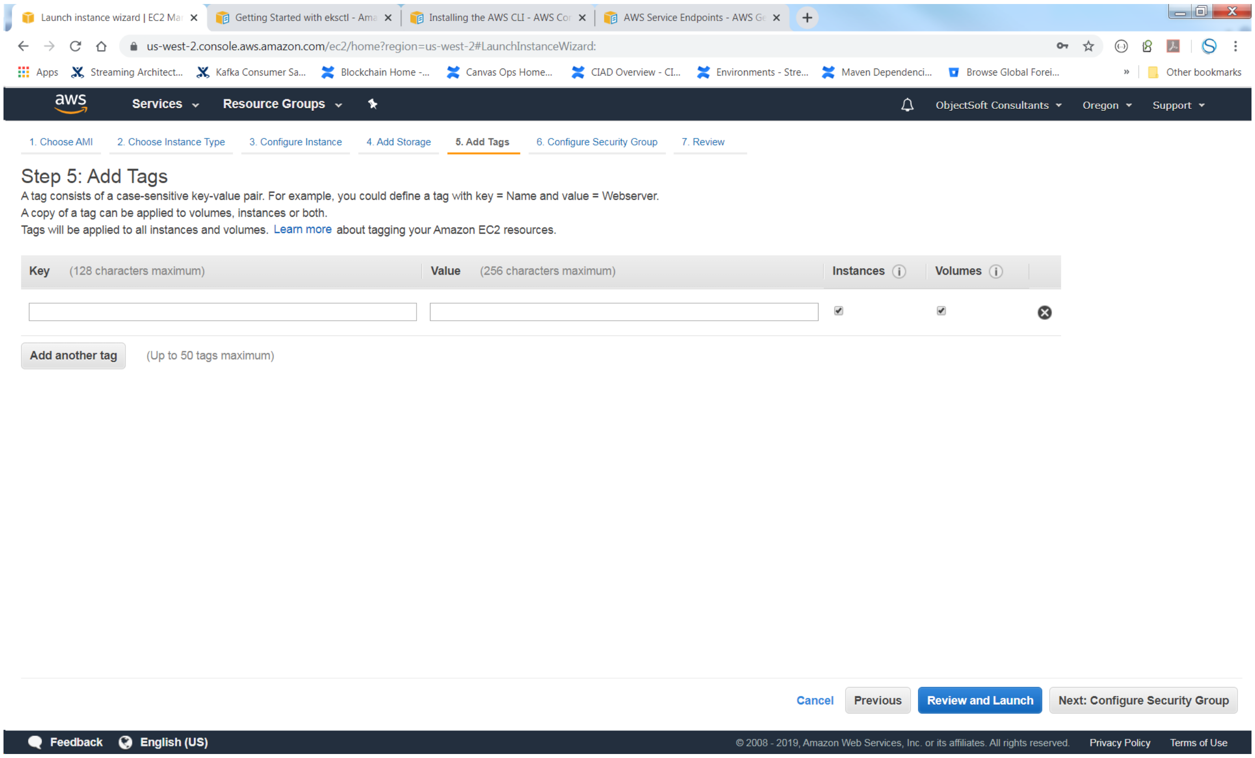
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0c2bc4a9d8df2cacc	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

No need to Add Tags



Step 5: Add Tags

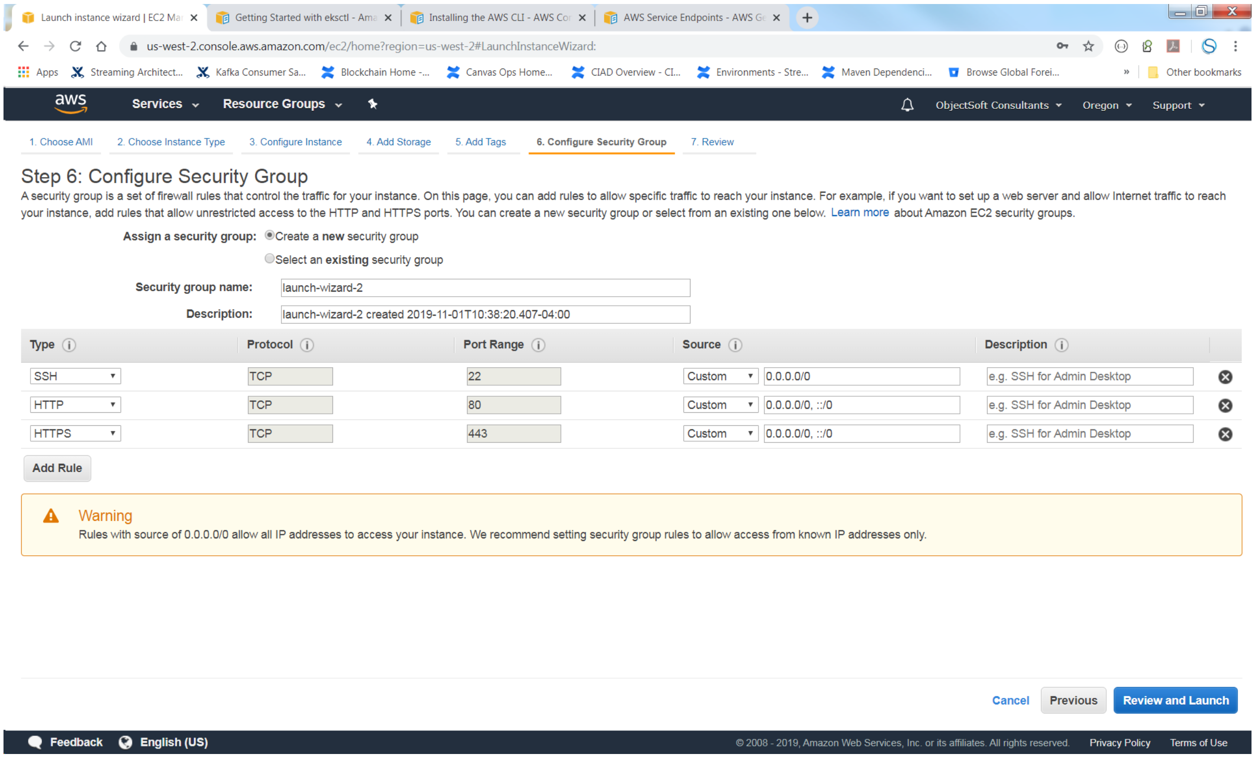
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
		☑	☑

Add another tag (Up to 50 tags maximum)

Cancel
Previous
Review and Launch
Next: Configure Security Group
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

On Configure Security Group screen, add new Rules as below:



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2019-11-01T10:38:20.407-04:00

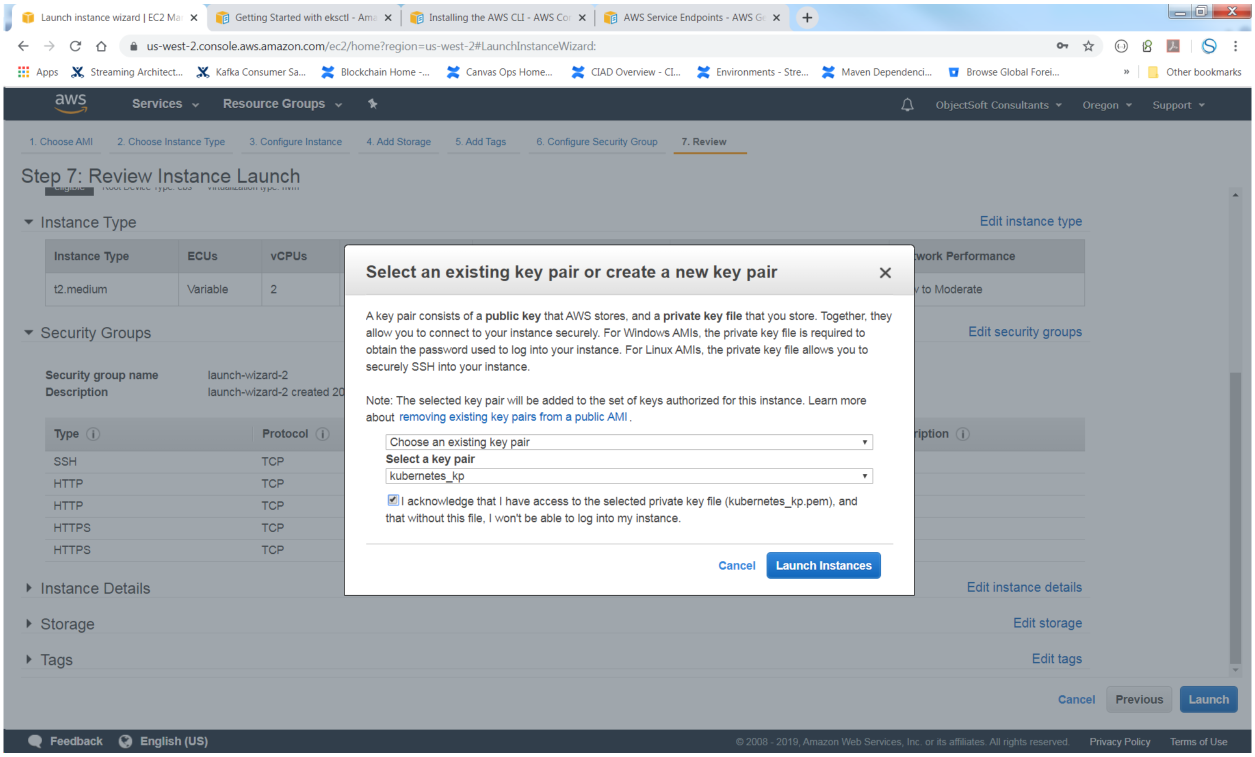
Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel
Previous
Review and Launch
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Either create a new keypair or use an existing one to Launch an instance:



Step 7: Review Instance Launch

Instance Type

Instance Type	ECUs	vCPUs	…work Performance
t2.medium	Variable	2	…y to Moderate

Security Groups

Security group name launch-wizard-2
Description launch-wizard-2 created 20…

Type	Protocol	…ription
SSH	TCP	
HTTP	TCP	
HTTP	TCP	
HTTPS	TCP	
HTTPS	TCP	

Instance Details

Storage

Tags

Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Choose an existing key pair ▾

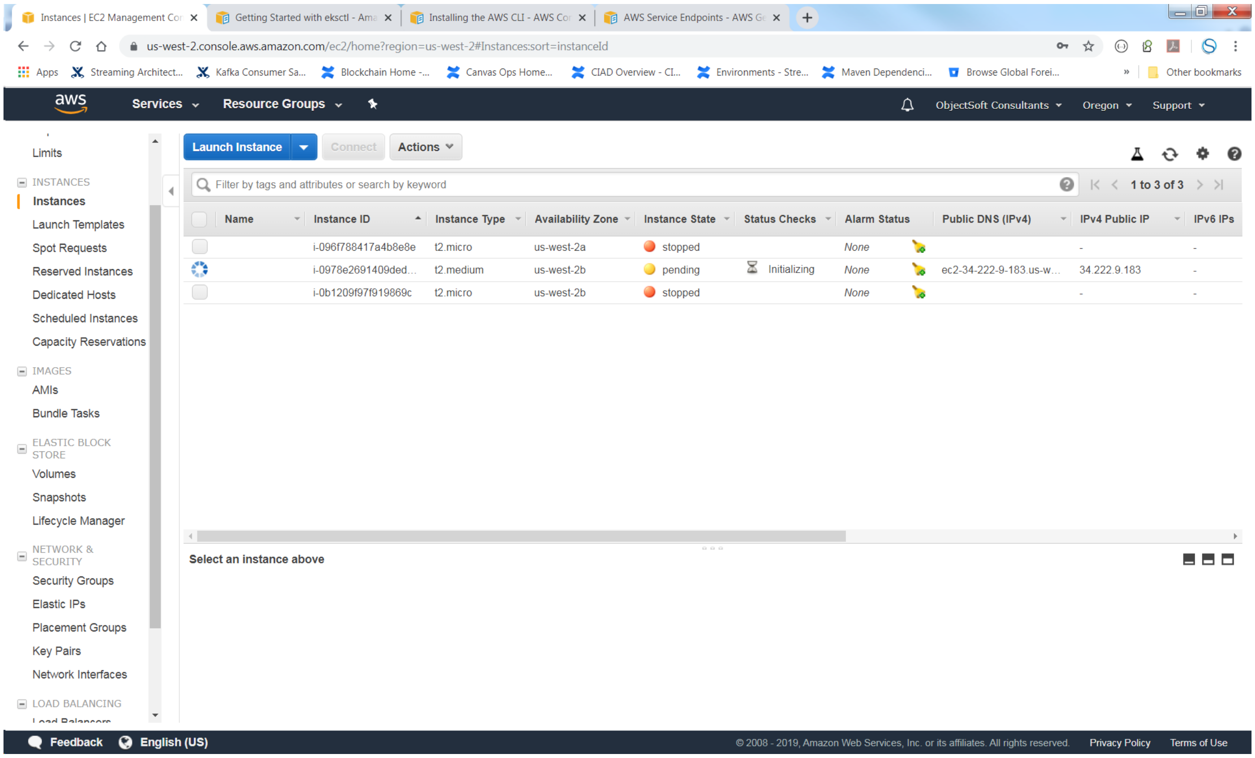
Select a key pair

kubernetes_kp ▾

☑ I acknowledge that I have access to the selected private key file (kubernetes_kp.pem), and that without this file, I won't be able to log into my instance.

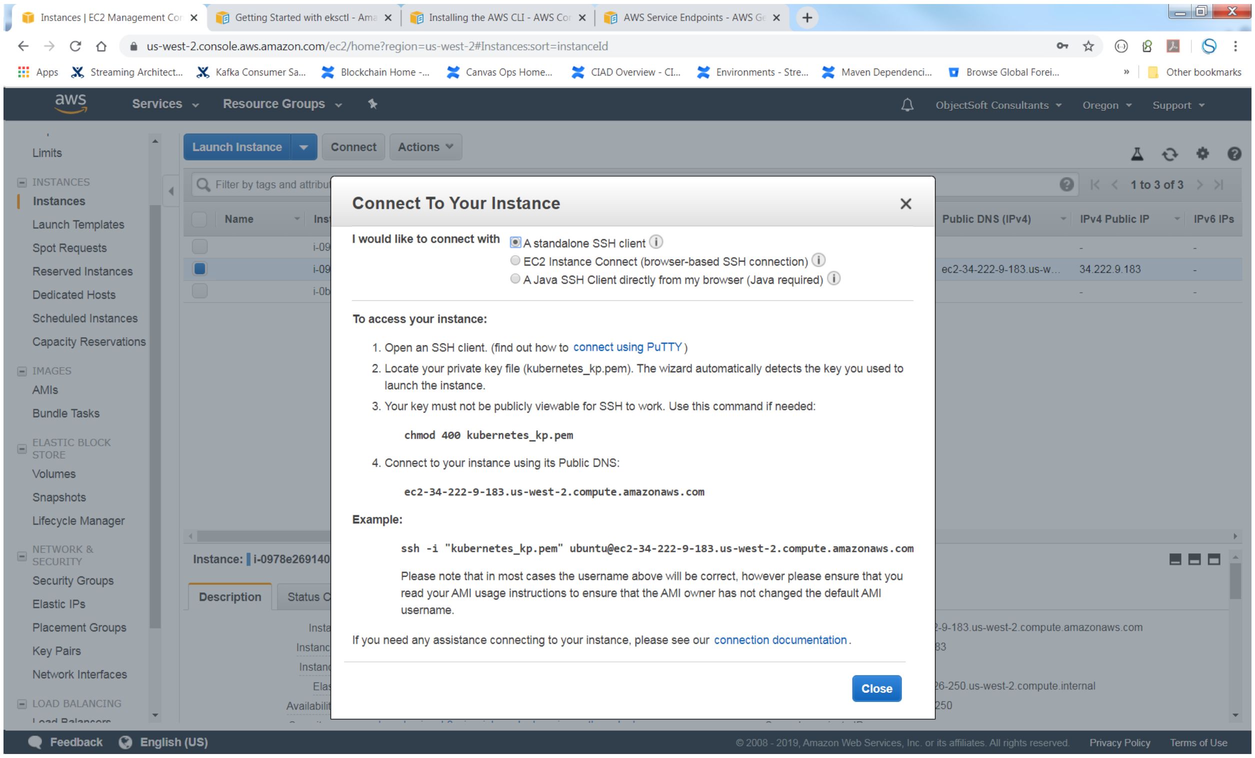
Cancel | Launch Instances

View Instance on EC2 Console:



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
	i-096f788417a4b8e8e	t2.micro	us-west-2a	🟥 stopped		None		-	-
	i-0978e2691409ded…	t2.medium	us-west-2b	🟡 pending	Initializing	None	ec2-34-222-9-183.us-w…	34.222.9.183	-
	i-0b1209f97f919869c	t2.micro	us-west-2b	🟥 stopped		None		-	-

Connect with EC2 Instance using Standalone SSH client



Connect To Your Instance

I would like to connect with ● A standalone SSH client ○ EC2 Instance Connect (browser-based SSH connection) ○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (kubernetes_kp.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 kubernetes_kp.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-34-222-9-183.us-west-2.compute.amazonaws.com
```

Example:

```
ssh -i "kubernetes_kp.pem" ubuntu@ec2-34-222-9-183.us-west-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

After connecting with your instance, update binaries:

```
[ec2-user ~]$ sudo yum update
```

Install Python3

```
[ec2-user ~]$ sudo yum install python3 -y
```

Verify Installation

```
[ec2-user ~]$ which python3
```

Install pip3

```
curl -O https://bootstrap.pypa.io/get-pip.py
```

```
python3 get-pip.py --user
```

Verify pip3 is installed correctly

```
sudo pip3 --version
```

Install latest AWS CLI

```
sudo pip3 install awscli  --upgrade -- user
```

Configure your AWS CLI Credentials

```
$ aws configure
AWS Access Key ID [None]: <<entries from Security Credential file in AWS Console>>
AWS Secret Access Key [None]: <<entries from Security Credential file in AWS Console>>
Default region name [None]: us-west-2
Default output format [None]: json
```

Install eksctl Command Line Utility

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/download/
latest_release/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

Move the extracted binary to /usr/local/bin.

```
sudo mv /tmp/eksctl /usr/local/bin
```

Test that your installation was successful with the following command.

```
eksctl version
```

Install Kubectl

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl
```

Make the kubectl binary executable.

```
chmod +x kubectl
```

Move the binary in to your PATH.

```
sudo mv kubectl /usr/local/bin/kubectl
```

Test to ensure the version you installed is up-to-date

```
kubectl version
```

Installing aws-im-authenticator

```
curl -o aws-iam-authenticator https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019-08-22/bin/linux/amd64/aws-iam-authenticator
```

Apply execute permissions to the binary

```
chmod +x aws-iam-authenticator
```

Copy the binary to a folder in your $PATH

```
sudo mv aws-iam-authenticator /usr/local/bin
```

Test that the aws-iam-authenticator binary works

```
aws-iam-authenticator help
```

Create your Amazon EKS cluster and Linux worker nodes with the following command

```
eksctl create cluster --name development --version 1.14 --region eu-west-1 --nodegroup-name standard-workers --node-type t2.micro --nodes 3 --nodes-min 2 --nodes-max 4 --node-ami auto
```

Create a kubeconfig for Amazon EKS

```
aws eks --region eu-west-1 update-kubeconfig --name development
```

Test your configuration

```
kubectl get svc
```

Output:

```
NAME             TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
svc/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   1m
```

Verify running nodes:

```
kubectl get nodes
```