

Linux и облачные вычисления

Создание сервера в AWS

Создание и запуск сервера Linux в AWS. Использование Putty и WinSCP.

Оглавление

[Linux и облачные вычисления](#)

[Облачные вычисления](#)

[Amazon Web Services](#)

[Создание и запуск сервера Linux в AWS](#)

[Использование Putty и WinSCP](#)

[Конвертация приватного ключа из pem в ppk](#)

[Подключение к экземпляру AWS](#)

[Практическое задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

Linux и облачные вычисления

Linux — это название семейства Unix-подобных операционных систем (ОС) на базе ядра Linux. Большинство операционных систем семейства Linux распространяются бесплатно, и их код является открытым, то есть доступным для всех. Linux широко применяется в профессиональных системах, в частности в проектах Data Science.

Серверы с установленными дистрибутивами Linux используются для хранения и обработки данных, на них также работают модели машинного обучения. Очень часто в производственных условиях на рабочих станциях программистов используется ОС Windows, а на удаленных серверах, доступ к которым программист получает с рабочей станции, — Linux.

Облачные вычисления

Облачные вычисления (англ. Cloud Computing) — это общее название работы с вычислительными ресурсами — серверами, дисками, процессорами, программным обеспечением, — которые оперативно предоставляются провайдером по требованию клиента.

Облачные вычисления дают возможность клиенту не закупать собственное оборудование и ПО, а пользоваться теми средствами вычислений, которые есть у провайдера (как бы взять их в аренду). Это позволяет значительно экономить, особенно если данные средства вычисления требуются не на постоянной основе, а время от времени.

Особенно выгодна такая схема работы в тех случаях, когда требуется часто менять конфигурацию оборудования. Для Data Science это очень актуально, так как для данных большого объема могут понадобиться изменения такого параметра сервера, как оперативная память. Это потребовало бы времени и средств. В случае применения облачных вычислений такое изменение вносится за считанные минуты и небольшое повышение арендной платы.

Другой пример выгодного применения облачных вычислений — использование видеокарт (то есть графических процессоров — GPU, от англ. graphics processing unit) для ускорения работы искусственных нейронных сетей. Если нейросети не используются ежедневно, то выгоднее не покупать GPU, а использовать сервер с GPU в облаке.

Amazon Web Services

В данном уроке мы научимся создавать сервер с ОС Linux в одном из облачных сервисов — Amazon Web Services (сокращенно — AWS).

Amazon Web Services — один из примеров общедоступного облачного сервиса. Это коммерческое публичное облако, которое поддерживается и развивается компанией Amazon с 2006 года. Предоставляет частным и корпоративным клиентам ресурсы для облачных вычислений: виртуальные серверы, ресурсы хранения, облачные базы данных, облачное связующее ПО, облачные бессерверные вычисления и средства разработки.

Для регистрации аккаунта в AWS нужно пройти по ссылке: [Amazon Web Services](https://aws.amazon.com/ru/).

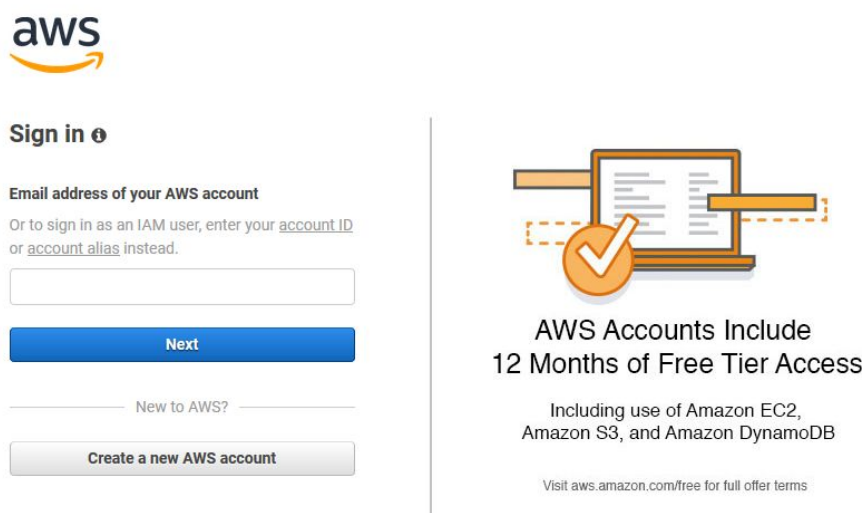
Далее нажимаем на кнопку «Создать аккаунт AWS». Откроется страница с формой, в которой нужно заполнить поля: email, пароль и имя аккаунта AWS.

Затем нужно выбрать тип аккаунта — профессиональный или личный. Выбираем личный, так как мы используем его для своего обучения. Затем заполняем все необходимые сведения. Потребуется ввести данные кредитной карты, так как сервисы AWS платные.

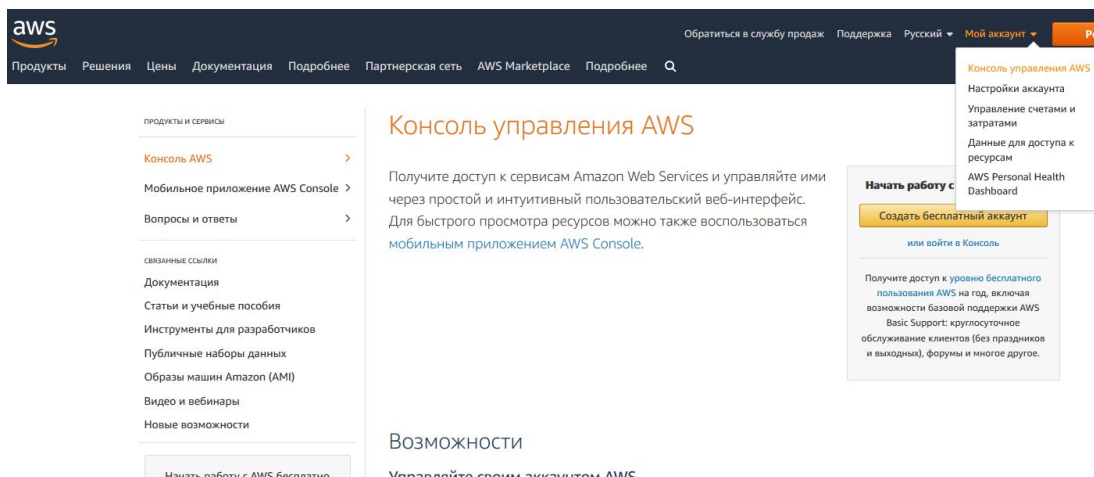
Нужно обращать внимание на то, что если вы не пользуетесь сервисами, но они не отключены, то счет за их использование все равно будет формироваться, и в начале следующего месяца средства будут списаны с карты. Забывчивость может стоить денег. При спорных вопросах следует незамедлительно обращаться в техническую поддержку Amazon Web Services.

После того как аккаунт зарегистрирован, нужно зайти в консоль AWS по ссылке [Amazon.com Sign In](https://aws.amazon.com/signin/).

Появится форма входа в AWS:

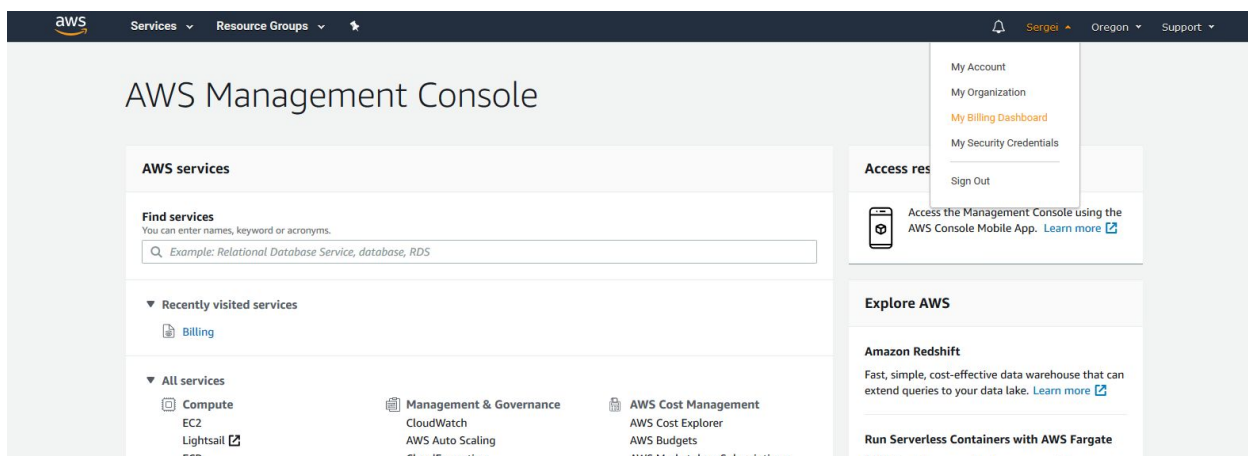


Либо можно зайти со страницы AWS, перейдя из раздела «Мой аккаунт» на страницу «Управление консолью AWS» следующим образом:

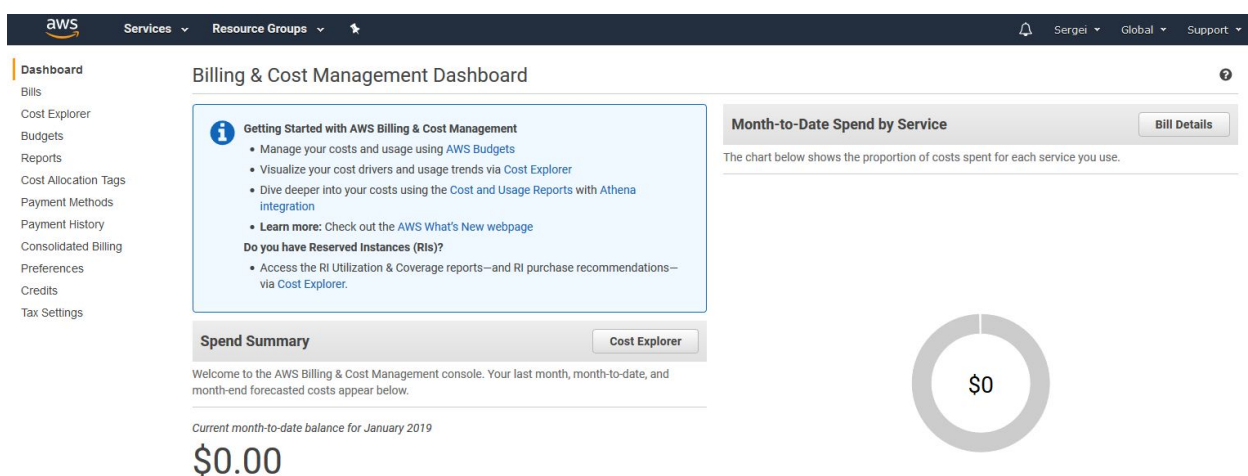


Так вы перейдете на страницу входа в AWS.

После того как вы впервые зайдете в аккаунт, рекомендуем ознакомиться со страницей **My Billing Dashboard**. Один из способов перейти на нее:



Вот как выглядит сама страница My Billing Dashboard:



В 2019 году AWS предоставляет для новых клиентов 12 месяцев бесплатного доступа к ряду сервисов на определенное количество часов. Этих сервисов нам хватит для данного курса, но рекомендуем периодически проверять раздел My Billing Dashboard и при спорных ситуациях писать в Support Center (службу поддержки) AWS.

В случае платного использования рекомендуется ознакомиться с ценами на экземпляры: [цены на Amazon EC2](#).

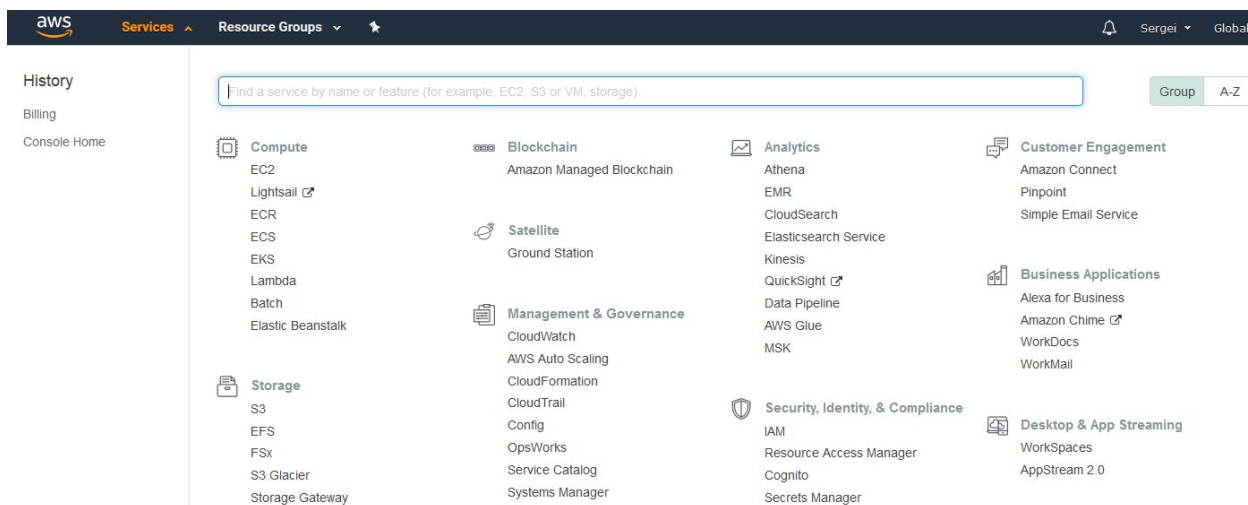
В данном курсе мы будем работать с экземплярами по требованию, а конкретно — с экземпляром **t2.micro**. При платном использовании по требованию на данный момент (2019 год) его аренда стоит около 1 американского цента в час. Если по данному экземпляру нет бесплатной акции для новичков, следует быть готовым, что средства за его использование спишутся.

После того как вы освоитесь с сервисами AWS, можно будет рассматривать другие планы их использования, но для данного курса экземпляры по требованию, или On Demand в английском варианте, — наиболее простой вариант. Он подразумевает почасовую оплату, поэтому после завершения использования экземпляра следует его остановить, чтобы не тратить лишнего.

Подробная информация об экземплярах по требованию: [цены на Amazon EC2 — оплата ресурсов по требованию](#).

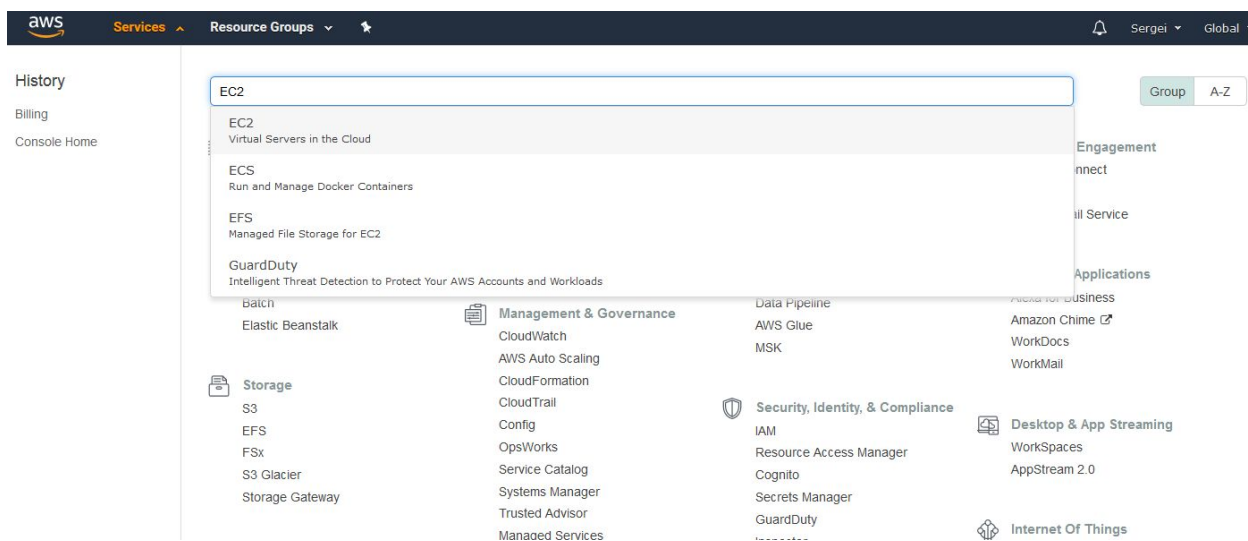
Создание и запуск сервера Linux в AWS

Для начала нужно выбрать раздел Services (находится в левом верхнем углу в шапке сайта). Появится список сервисов, которые предоставляет AWS:

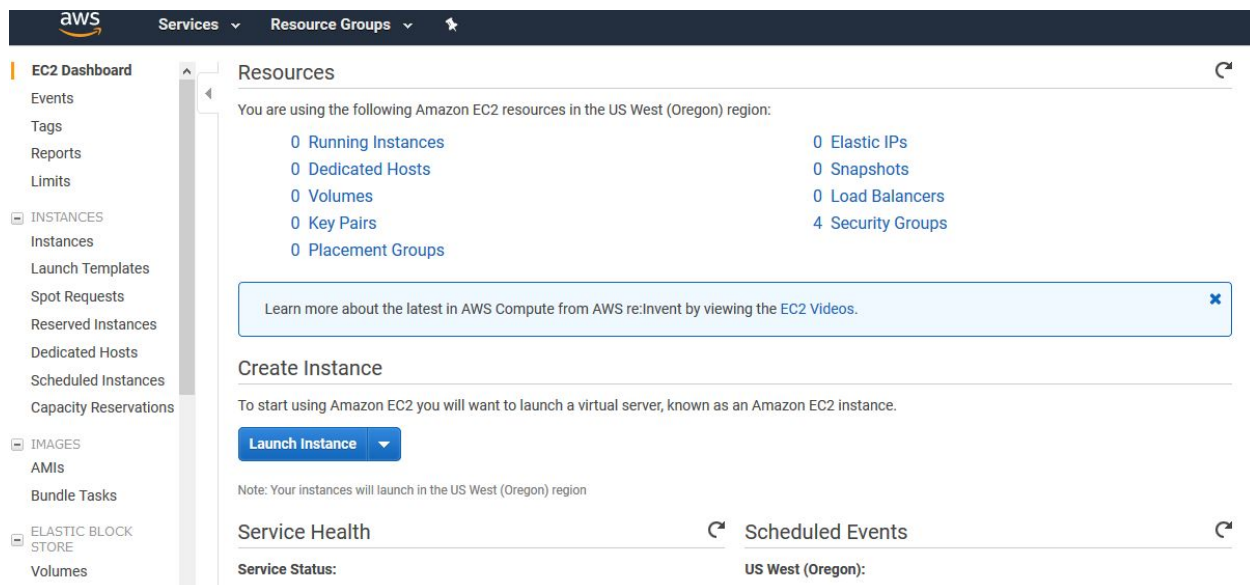


Нам потребуется только один из них — EC2 (сокращение от Elastic Computing). С помощью EC2 мы можем создавать виртуальные серверы в облаке.

Если этот сервис не виден, то можно написать его название в строку поиска сверху:

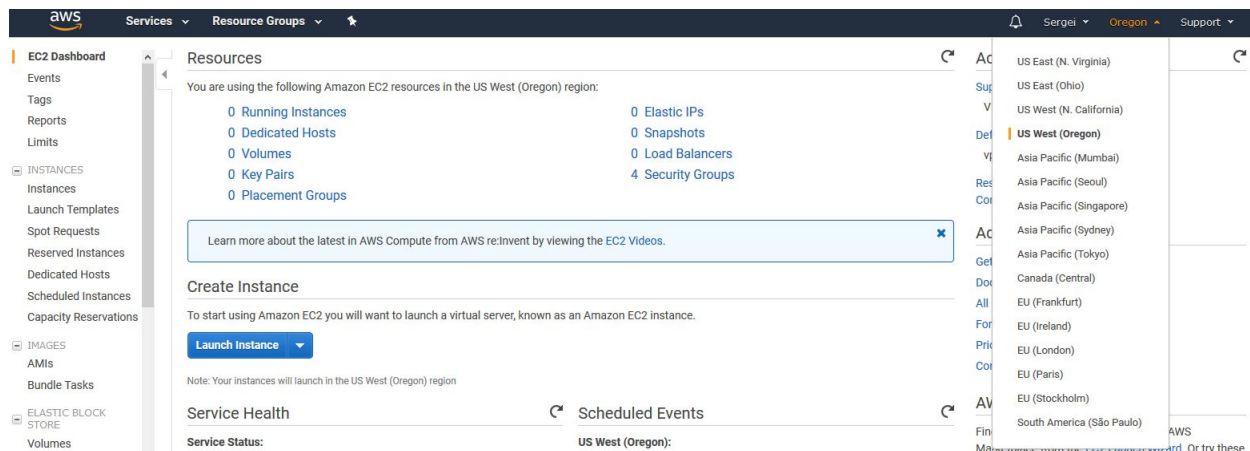


Переходим в EC2 и видим страницу EC2 Management Console:

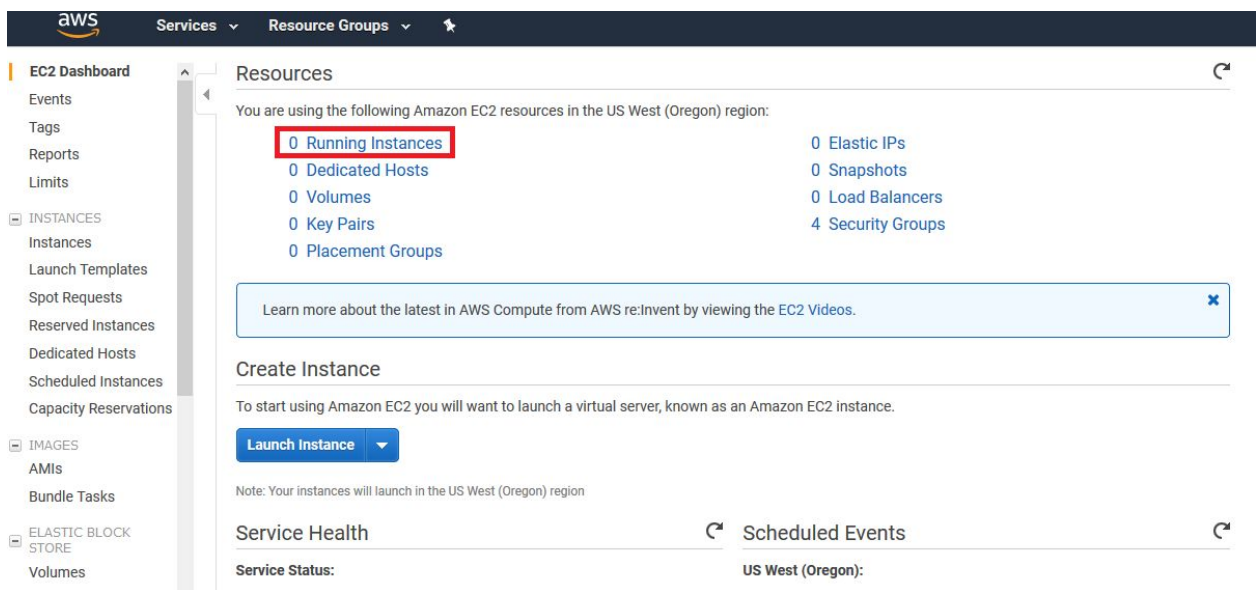


В AWS можно выбирать регионы, в которых будут находиться виртуальные серверы. Чем ближе сервер к месту вашего расположения, тем быстрее будет доступ к нему. Для нашего курса регион значения не имеет. Также для начинающих пользователей могут быть доступны не все регионы. На этой странице видно, что это консоль для виртуальных серверов, располагающихся на западе США, в штате Орегон — то есть здесь можно видеть не все серверы, которые запущены из-под данного аккаунта, а только те, которые находятся в этом штате.

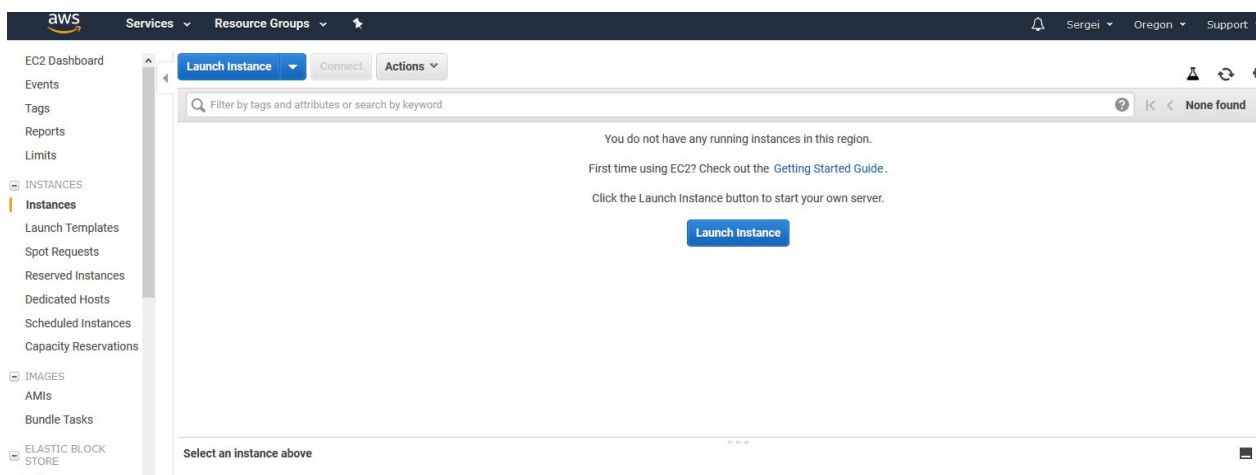
Перейти в консоль управления серверами, находящимися в другой местности, можно, выбрав регион в выпадающем списке:



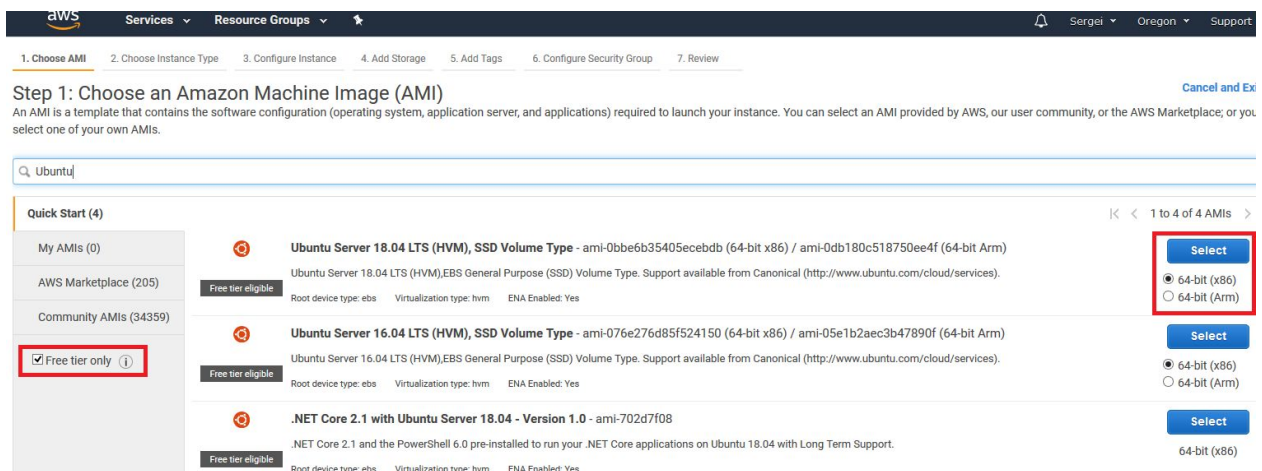
На странице управления EC2 можно перейти к имеющимся серверам, нажав на ссылку Running Instances:



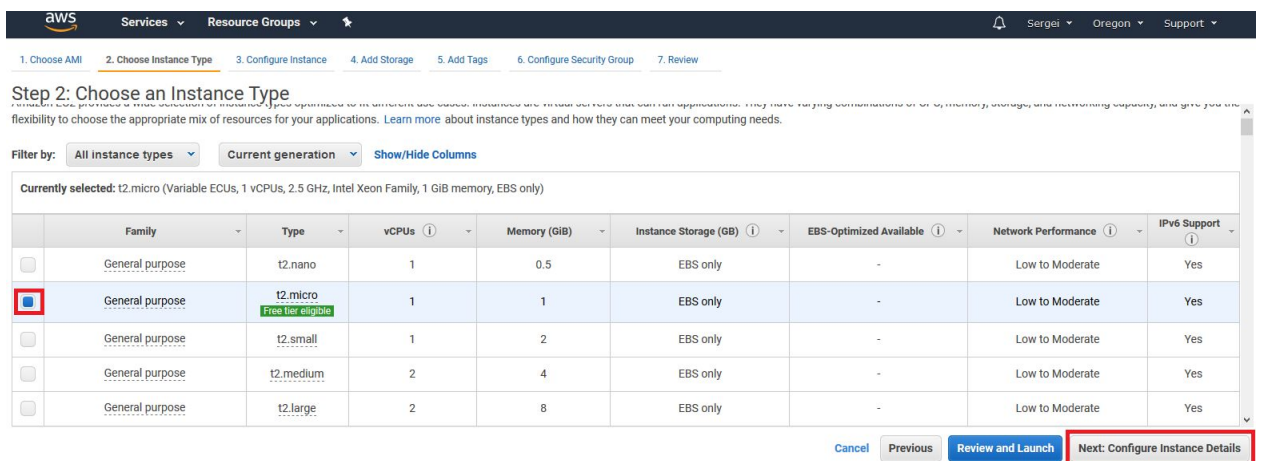
Если вы только начали пользоваться аккаунтом, то открывшийся список серверов будет пуст:



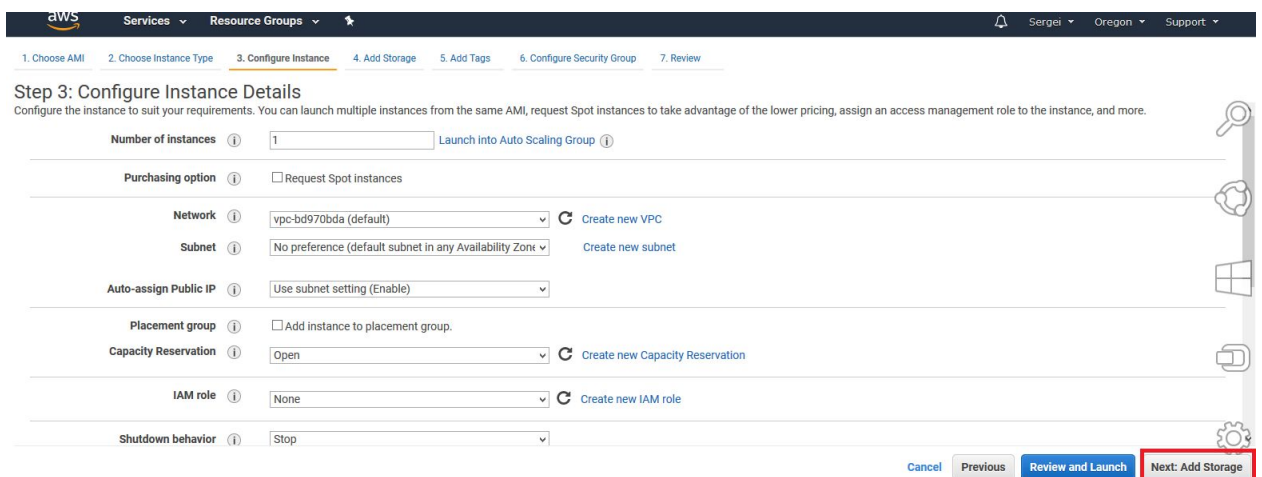
Чтобы создать виртуальный сервер, нужно нажать на кнопку **Launch Instance**. Появится первый шаг — нужно выбрать **AMI (Amazon Machine Image)**. Это образ дистрибутива с операционной системой, в нем могут содержаться установленные программы или даже данные. Отметим чекбокс **Free Tiers Only**, чтобы выбирать только из бесплатных образов, и в строке поиска введем Ubuntu (это операционная система, основанная на Debian GNU/Linux). Выберем 18 версию этой ОС, нажмем на кнопку **Select**:



После нажатия **Select** мы переходим ко второму шагу — выбираем тип экземпляра. Это будет тот тип, который бесплатен для новых клиентов AWS (об этом говорит подпись **Free tier eligible**). В данном случае это **t2.micro**. Затем нажимаем на кнопку **Next: Configure Instance Detail**:



Далее мы оказываемся на шаге 3 — здесь можно оставить все по умолчанию и нажать на кнопку **Next: Add Storage**:



На шаге 4 выбираем размер диска — оставим 8 Гб по умолчанию. Нужно проверить, что отмечен чекбокс **Delete on termination**: чтобы диск также удалялся, если захотим удалить экземпляр. Дисковое пространство может быть платным (с почасовым или иным тарифом), поэтому необходимо

следить за его использованием. Новым клиентам дается возможность использовать определенное количество гигабайт бесплатно (для более подробной информации нужно пройти по ссылкам, указанным на данной странице). Для перехода на следующий шаг нажимаем кнопку **Next: Add Tags**:

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	Encrypted
Root	/dev/sda1	snap-0576704bcf883d5ee	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

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.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

На следующем шаге оставляем все по умолчанию и нажимаем на кнопку **Next: Configure Security Group**:

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 (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
This resource currently has no tags			

Choose the [Add tag](#) button or [click to add a Name tag](#).
Make sure your [IAM policy](#) includes permissions to create tags.

[Add Tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

На шаге 6 требуется создать **Security Group**. Из выпадающего списка выберем **My IP**, чтобы экземпляр был доступен только с того IP, с которого он создается. Такое ограничение делается в целях безопасности, чтобы сервером не смогли воспользоваться с другого IP-адреса. При этом, если вы в дальнейшем будете работать с другого IP (например, решите заходить не из дома, а с рабочего компьютера), всегда можно зайти в консоль управления на сайте AWS и отредактировать группу безопасности, поменяв допустимый IP. Чтобы еще усилить безопасность экземпляра, можно изменить порт 22 (в поле **Port Range**) на другое значение. Далее нажимаем на кнопку **Review and Launch**:

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:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom	gIDR, IP or Security Group

Add Rule

Cancel Previous **Review and Launch**

На заключительном шаге 7 нужно еще раз убедиться в правильности конфигурации экземпляра. И далее нажать кнопку **Launch**:

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0bbe6b35405ecebdb

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2019-01-07T22:30:06.898+03:00

Cancel Previous **Launch**

Появится всплывающее окно с требованием выбрать приватный ключ. Если такого ключа нет, то нужно его создать.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0bbe6b35405ecebdb

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2019-01-07T22:30:06.898+03:00

Cancel Previous **Launch**

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

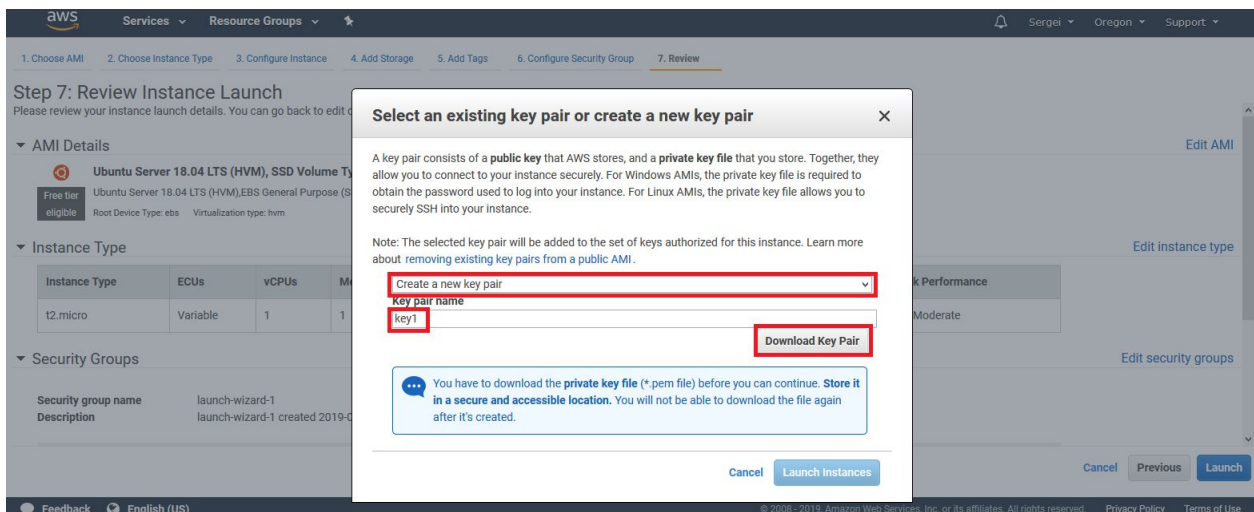
Select a key pair

No key pairs found

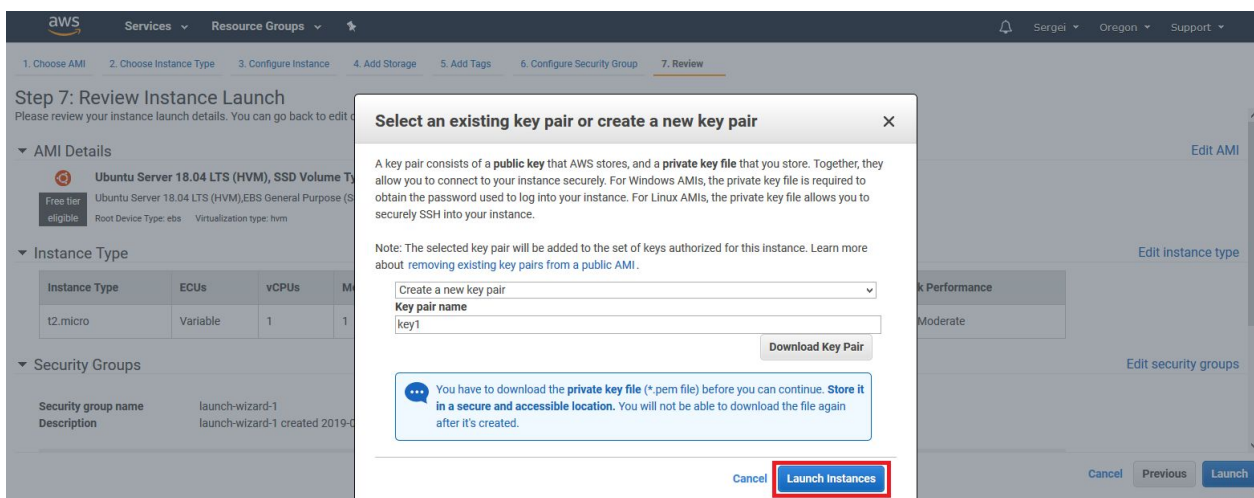
No key pairs found
You don't have any key pairs. Please create a new key pair by selecting the **Create a new key pair** option above to continue.

Cancel **Launch Instances**

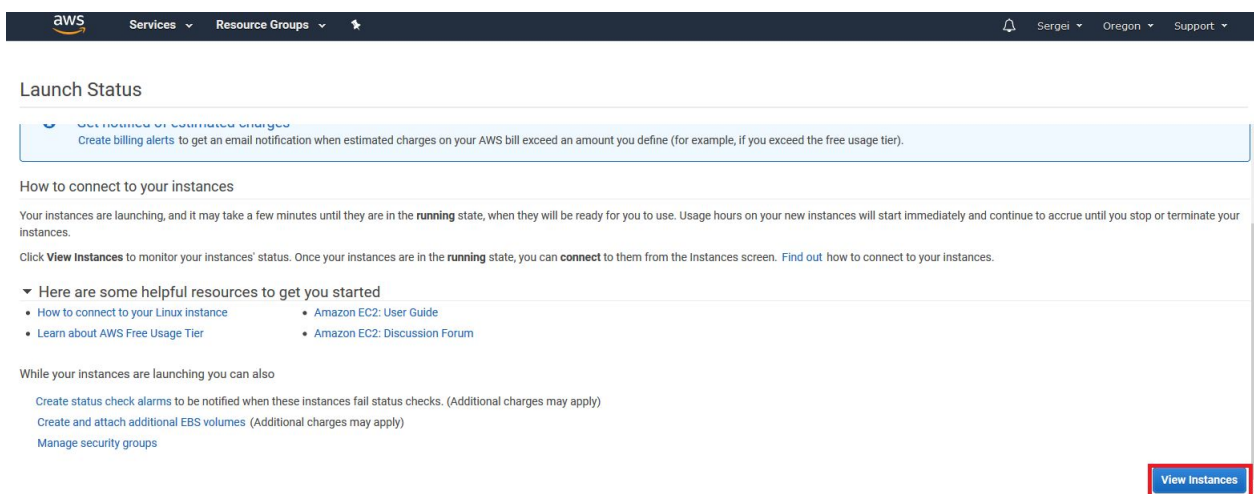
В выпадающем списке выбираем **Create a new key pair**, в поле ниже вводим название приватного ключа (в данном случае это **key1**, но можно придумать любое другое) и нажимаем на кнопку **Download Key Pair**:



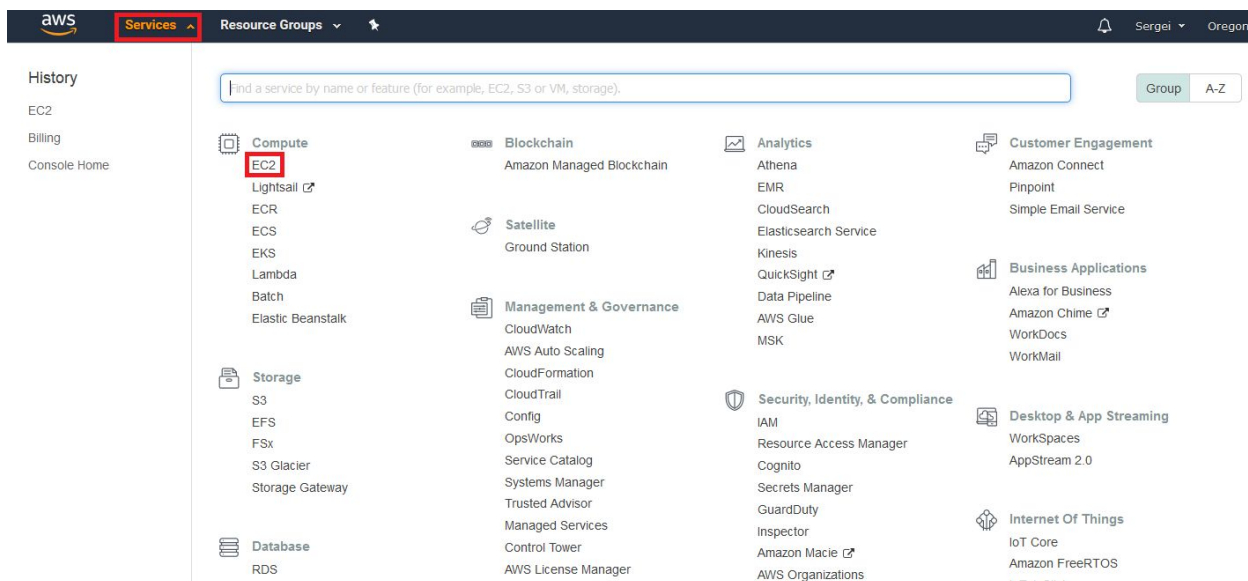
Нужно сохранить файл приватного ключа в надежное место и после этого нажать кнопку **Launch Instances**:



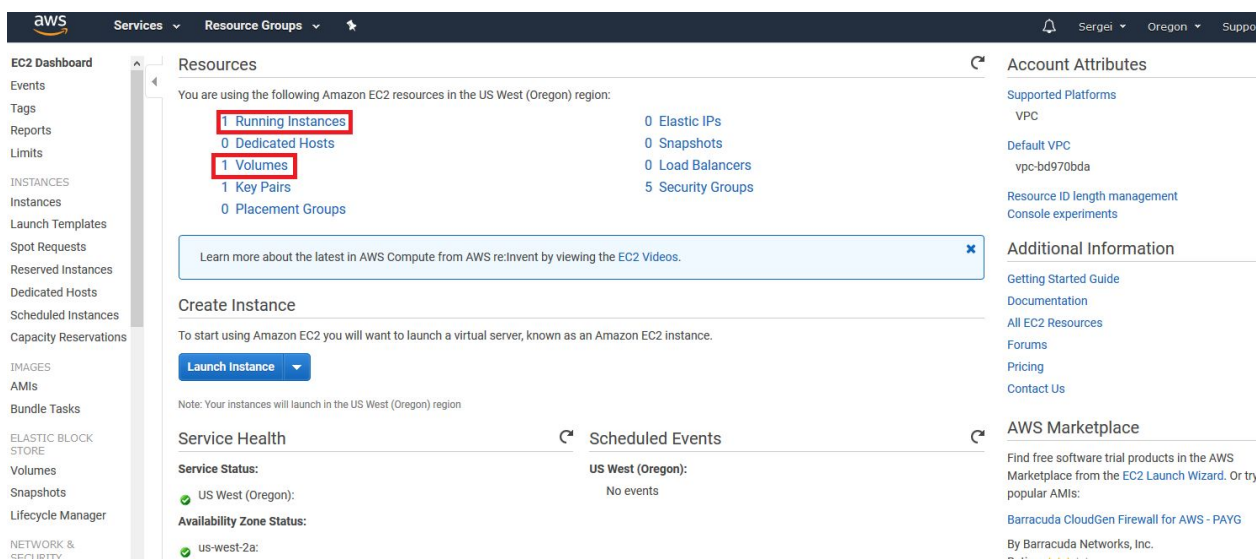
На следующей странице можно нажать на кнопку **Launch Instances**, чтобы увидеть запущенный экземпляр:



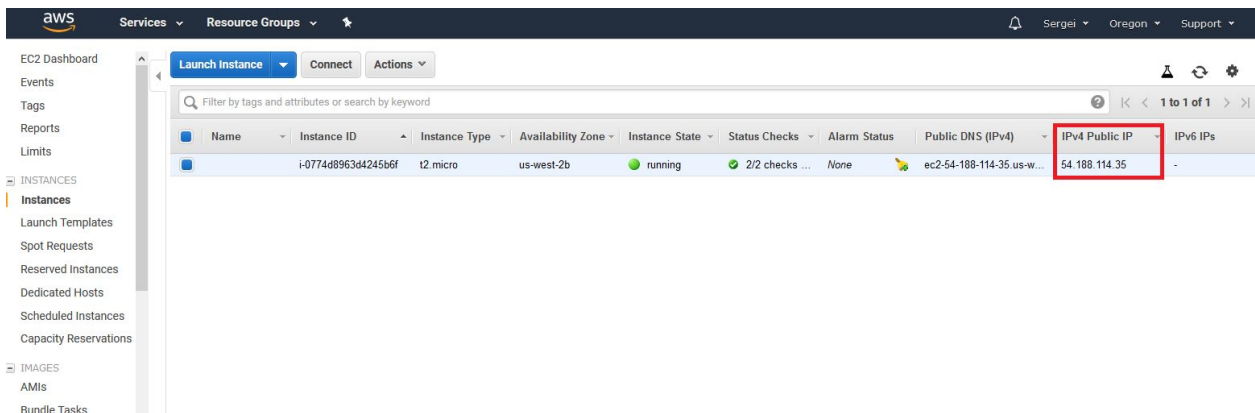
Или можно перейти в раздел **Services** и выбрать **EC2**:



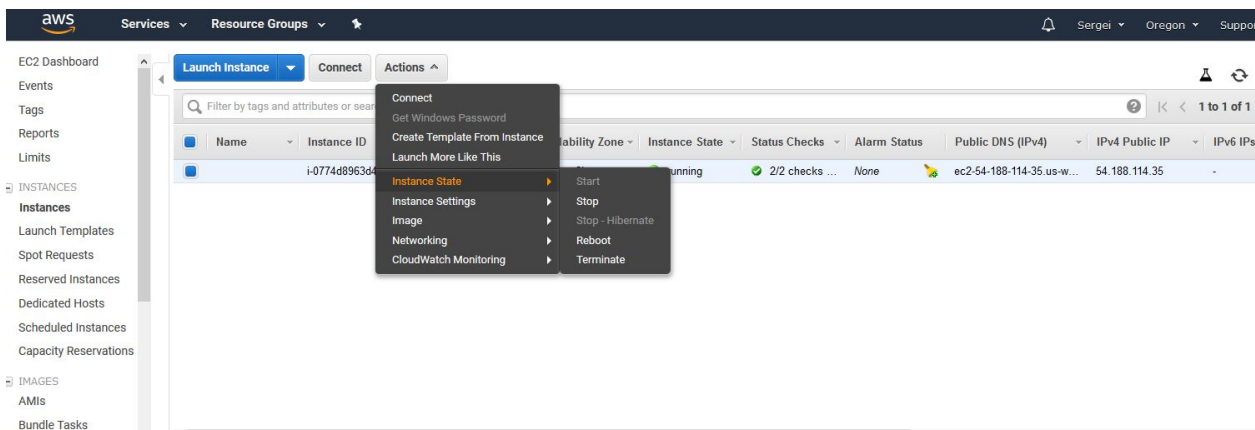
Можно увидеть, что запущен один экземпляр (**1 Running Instances**) и используется один диск (**1 Volumes**).



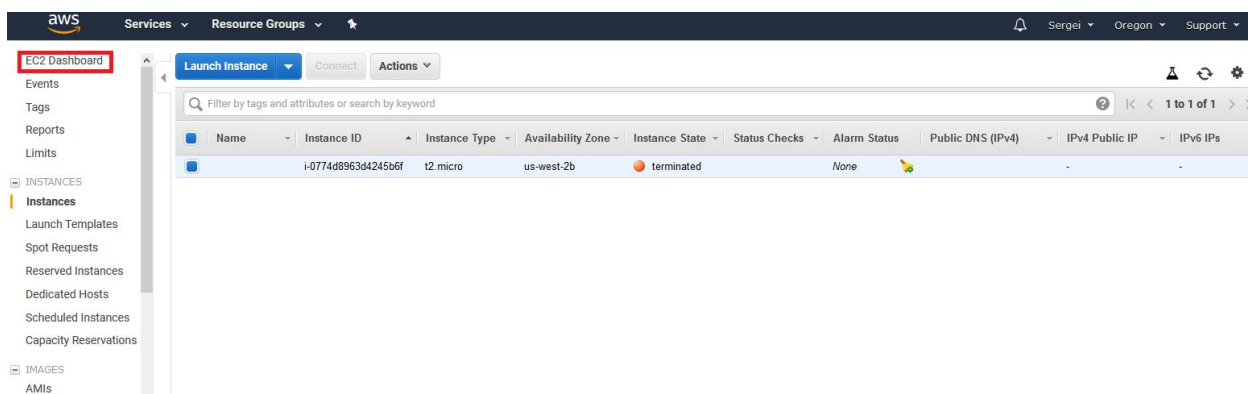
Пройдя по ссылке **Running Instances**, можно увидеть все созданные экземпляры. Для подключения к серверу нужно знать его IP-адрес:



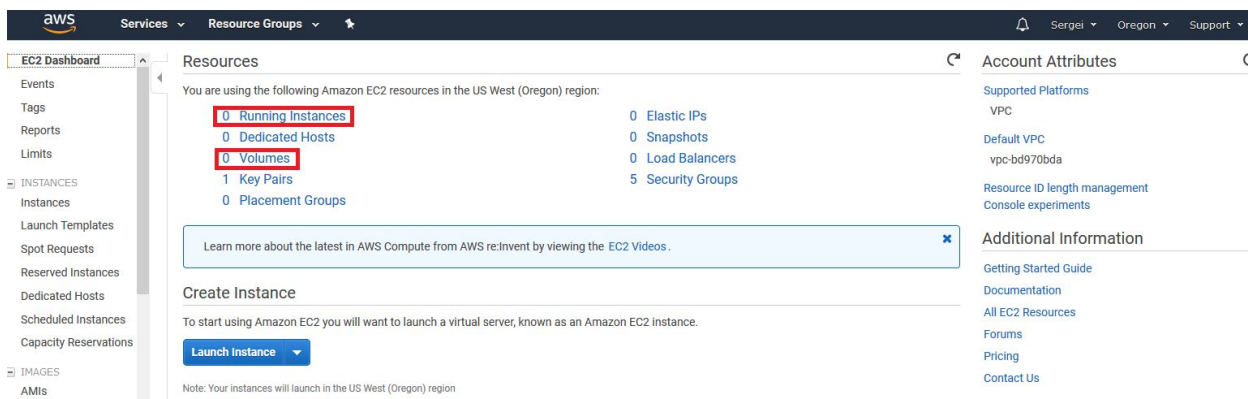
Изменить состояние экземпляра можно, открыв меню **Actions** и выбрав в нем **Instance State**:



Перезагрузить экземпляр можно с помощью **Reboot** (после перезагрузки сменится IP-адрес). Для остановки сервера можно использовать **Stop**, при этом сохранится конфигурация экземпляра, кроме IP-адреса. При остановленном экземпляре почасовая плата за него (если он не бесплатный) перестанет взиматься, но может продолжить взиматься плата за диск, если он не удален. Управлять диском можно, если из раздела EC2 перейти по ссылке **Volumes**. При выборе **Terminate** экземпляр полностью удалится. Если диск не нужен, следует убедиться, что он тоже удален. Когда удален экземпляр, нужно перейти в **EC2 Dashboard** по ссылке, отмеченной на скриншоте:



В **EC2 Dashboard**, если мы завершили работу с экземпляром и удалили его с помощью опции **Terminate**, мы должны увидеть, что ни одного экземпляра не запущено (**0 Running Instances**) и не используется ни одного диска (**0 Volumes**):



Использование Putty и WinSCP

Когда создан сервер в AWS, потребуется обращаться к нему — администрировать сервер, запускать программы и команды из терминала. Эти действия мы будем производить с помощью **PuTTY** и **WinSCP**.

PuTTY — свободно распространяемый клиент для протоколов удаленного доступа, включая SSH, Telnet, rlogin.

PuTTY позволяет подключиться к удаленному узлу и управлять им — например, сервером Linux. Программа является свободно распространяемым приложением с открытым исходным кодом и выпущена под open-source лицензией MIT.

В PuTTY можно сохранять список и параметры подключений для повторного использования. Заходя на сервер, не нужно повторно вводить IP-адрес или имя сервера, логин, пароль и другую информацию.

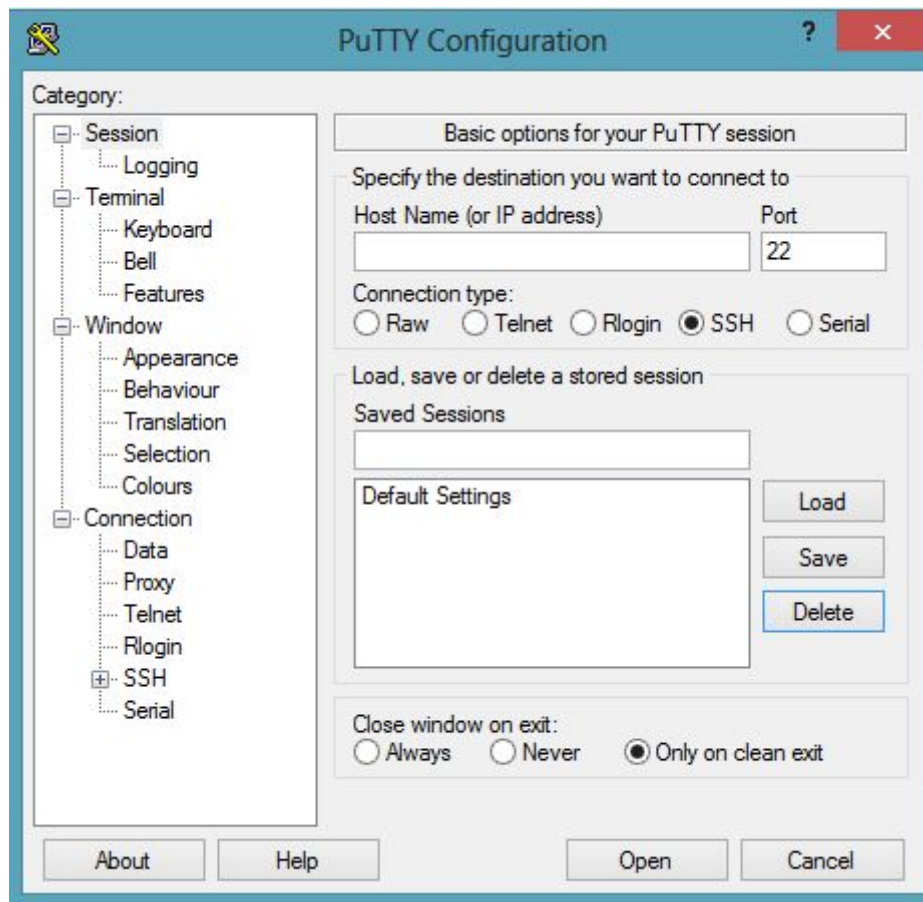


Рис. 1. PuTTY — конфигурация подключения

Для работы с удаленными серверами нам понадобятся входящие в состав PuTTY утилиты:

- **PuTTY** — сам клиент для Telnet и SSH (putty.exe);
- **PuTTYgen** — утилита для генерации RSA- и DSA-ключей (puttygen.exe).

WinSCP — это графический клиент SFTP (SSH File Transfer Protocol) для Windows с открытым исходным кодом. Предназначен для защищенного копирования файлов между компьютером и серверами, поддерживающими эти протоколы.

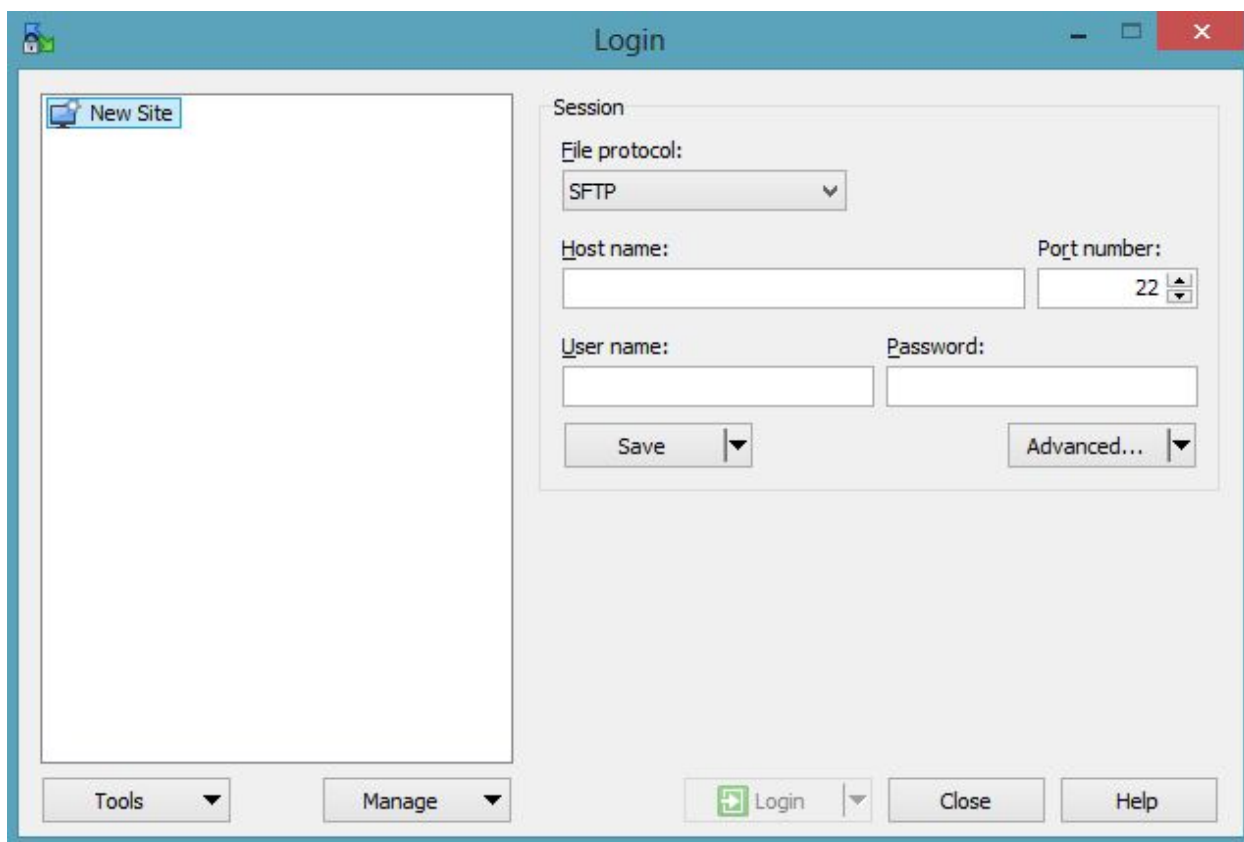


Рис. 2. WinSCP — конфигурация подключения

Ссылка для скачивания PuTTY: [Download PuTTY: latest release](#) (на этой же странице можно скачать puttygen.exe).

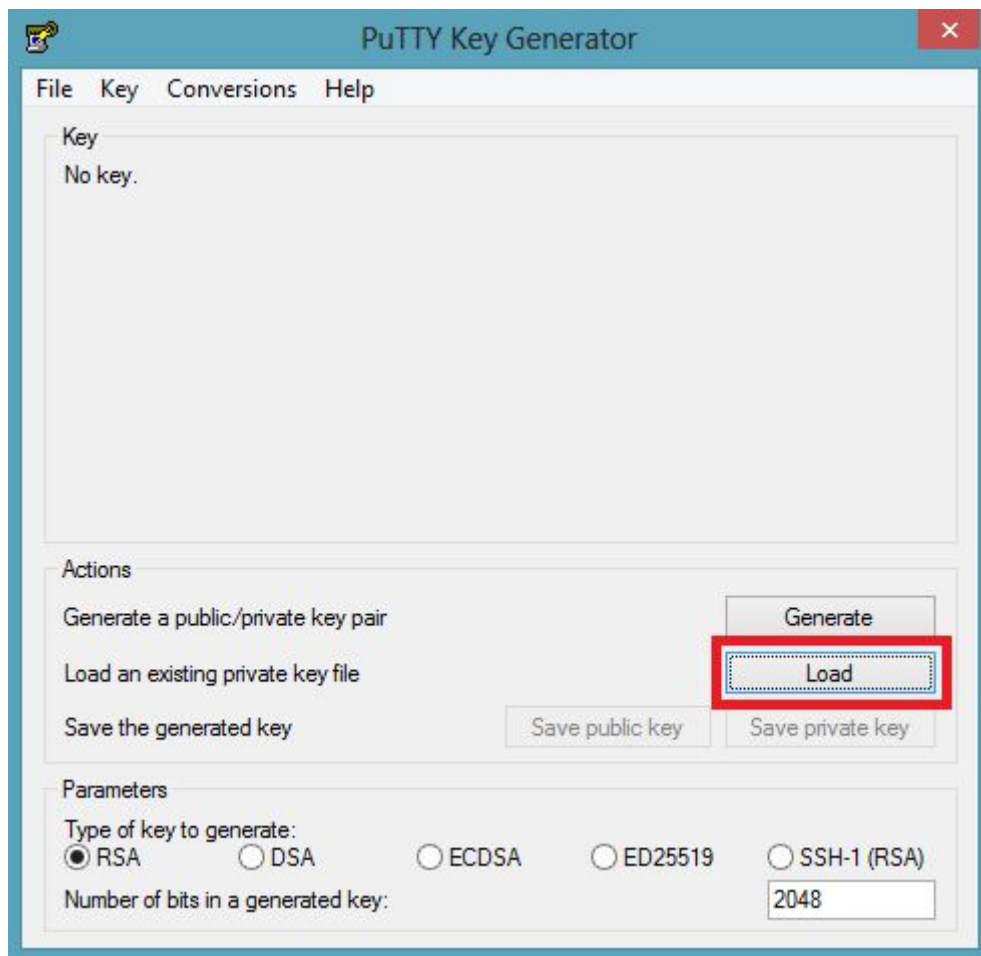
Ссылка для скачивания WinSCP: [WinSCP Download](#).

Конвертация приватного ключа из pem в ppk

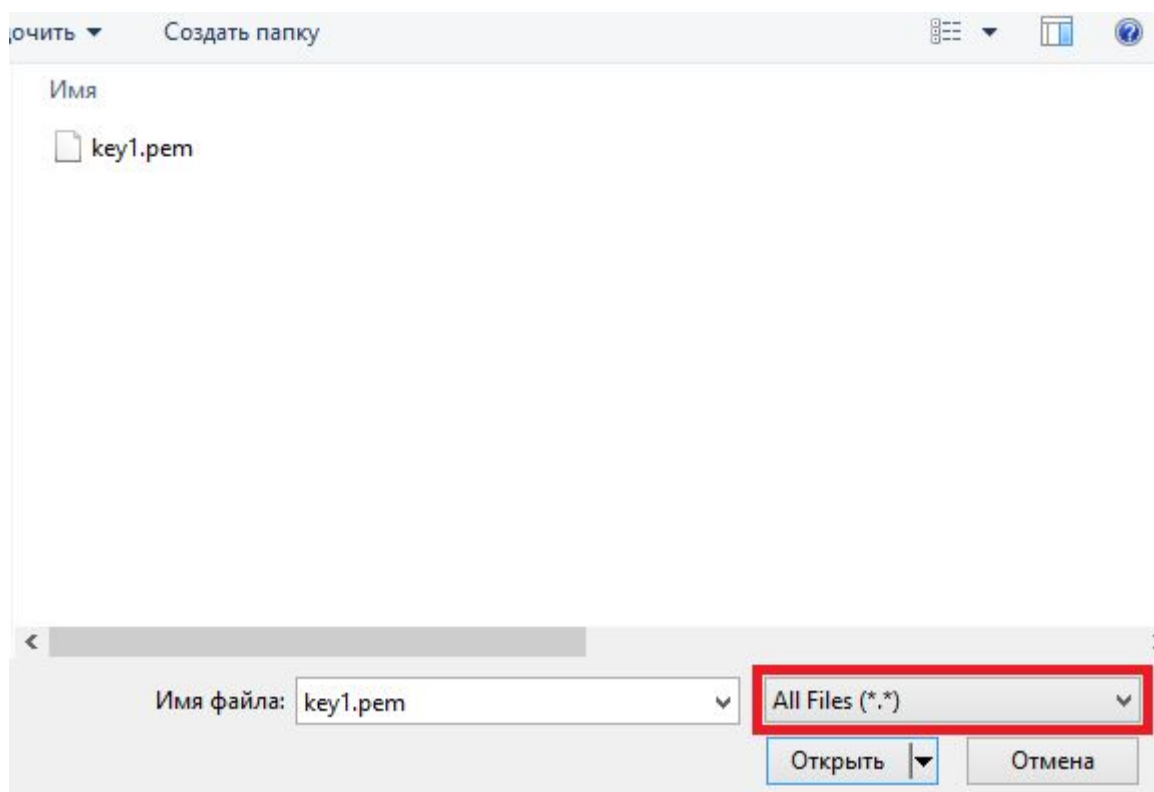
Созданный нами приватный ключ имеет расширение **.pem**, но, чтобы подключаться к серверу, нам потребуется конвертировать его в формат **.ppk**.

Запустим **puttygen.exe**. Найти и скачать его можно здесь: [Download PuTTY](#).

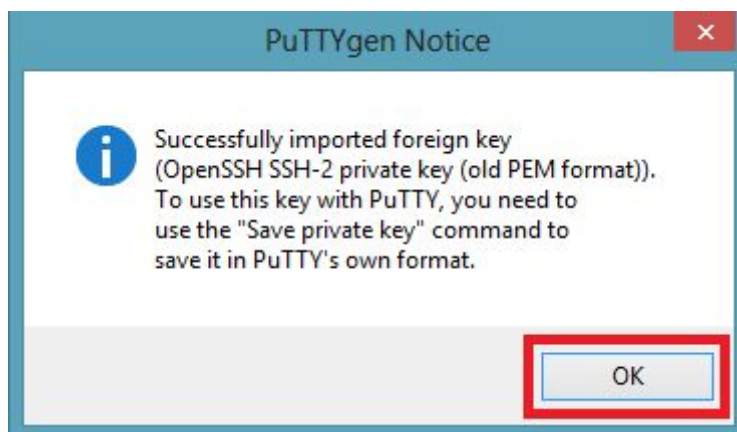
В поле **Type of key to generate** нужно выбрать **RSA** (если такого нет, то **SSH-2 RSA**). Нажимаем на кнопку **Load**:



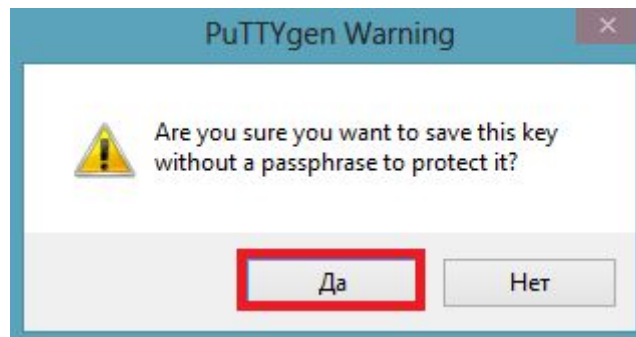
Выбираем файл с приватным ключом (при этом нужно выбрать, чтобы были показаны все файлы — **All files(*.*)**) и загружаем файл:



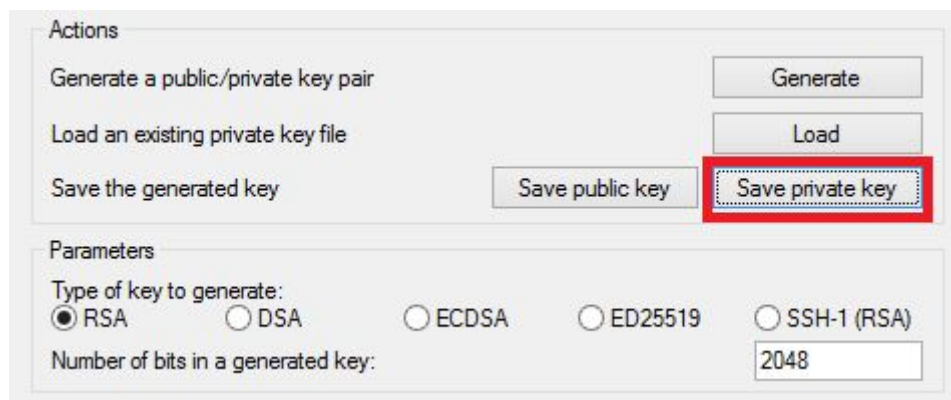
Появится всплывающее окно с текстом **Successfully imported foreign key....** Нужно нажать **OK**:



Далее появится всплывающее окно с предупреждением о том, что не заполнено **Key passphrase**. Нужно нажать «Да»:



Далее нажимаем **Save private key** и сохраняем файл с расширением **.ppk**. Нужно также ввести имя для приватного ключа — оно должно быть такое же, как у ключа с расширением **.pem**, то есть отличаться только расширением:

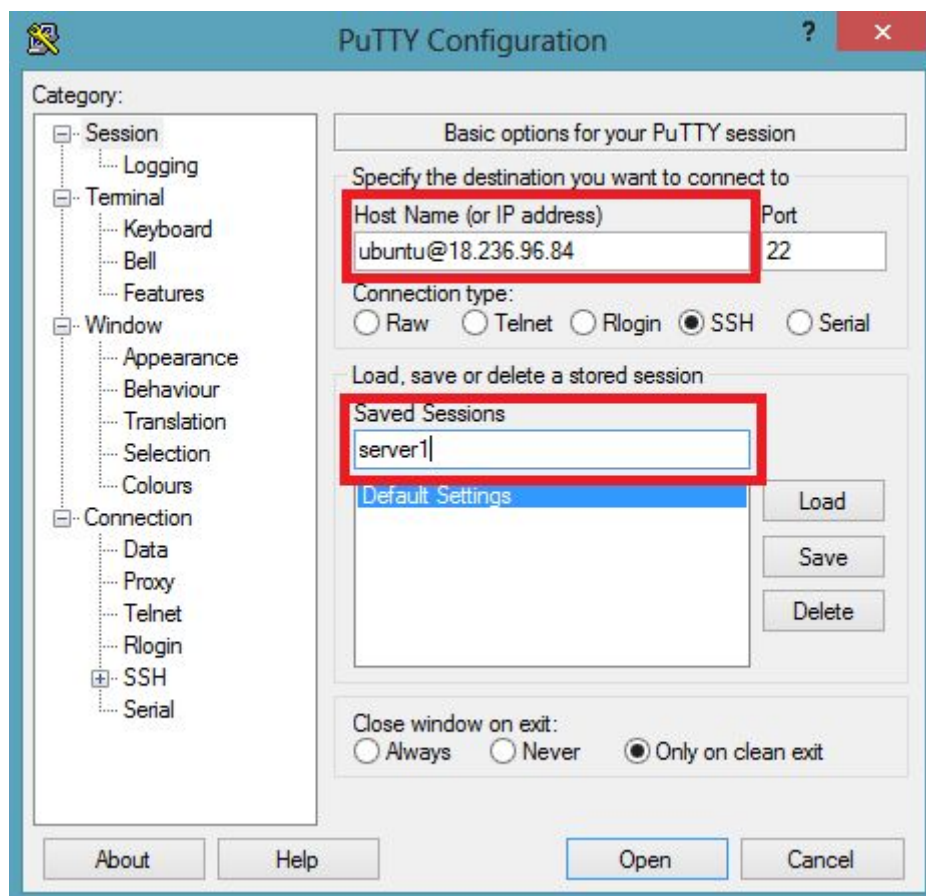


Когда приватный ключ сохранился в выбранную папку, программу **Puttygen** можно закрыть.

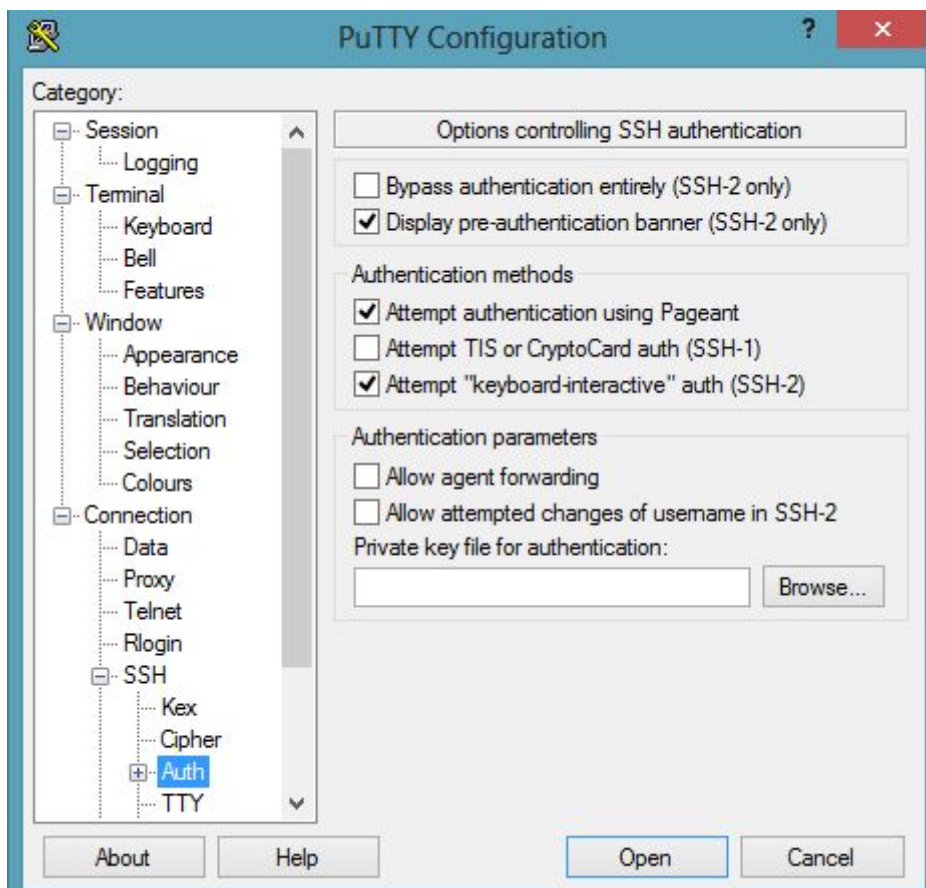
Подключение к экземпляру AWS

Запускаем **Putty** (перед этим требуется установить программу).

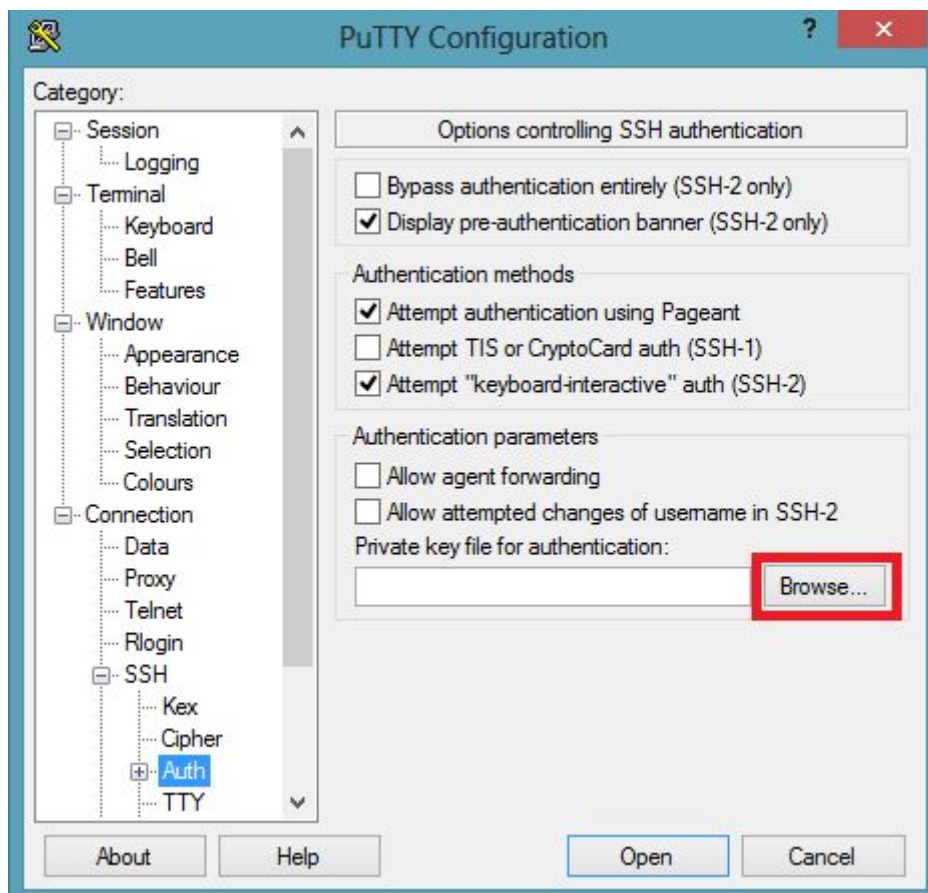
Мы окажемся в разделе **Session**. В нем нужно заполнить поле с IP-адресом сервера (**Host name (or IP address)**), причем сначала нужно указать имя пользователя (по умолчанию первый пользователь на серверах Ubuntu в AWS имеет имя **ubuntu**), далее идет символ **@**, а затем — IP-адрес сервера. Также нужно придумать название для подключения (**Saved Session**) — на скриншоте это **server1**:



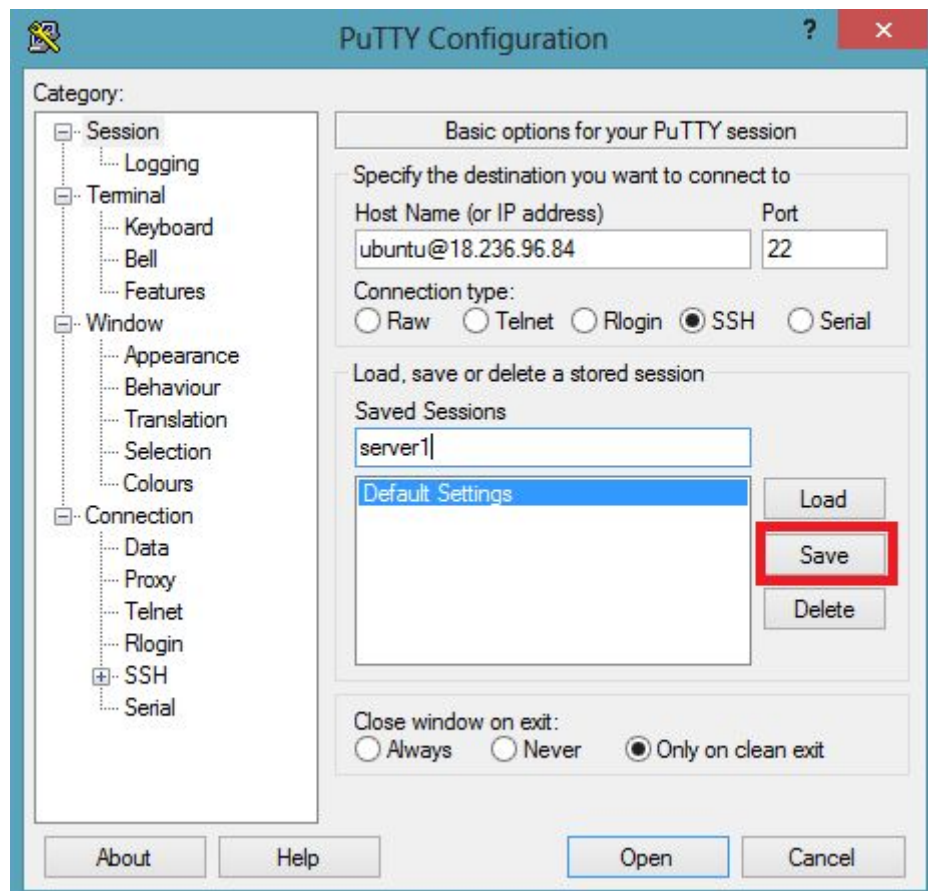
Далее заходим в раздел **SSH**, выбираем **Auth**:



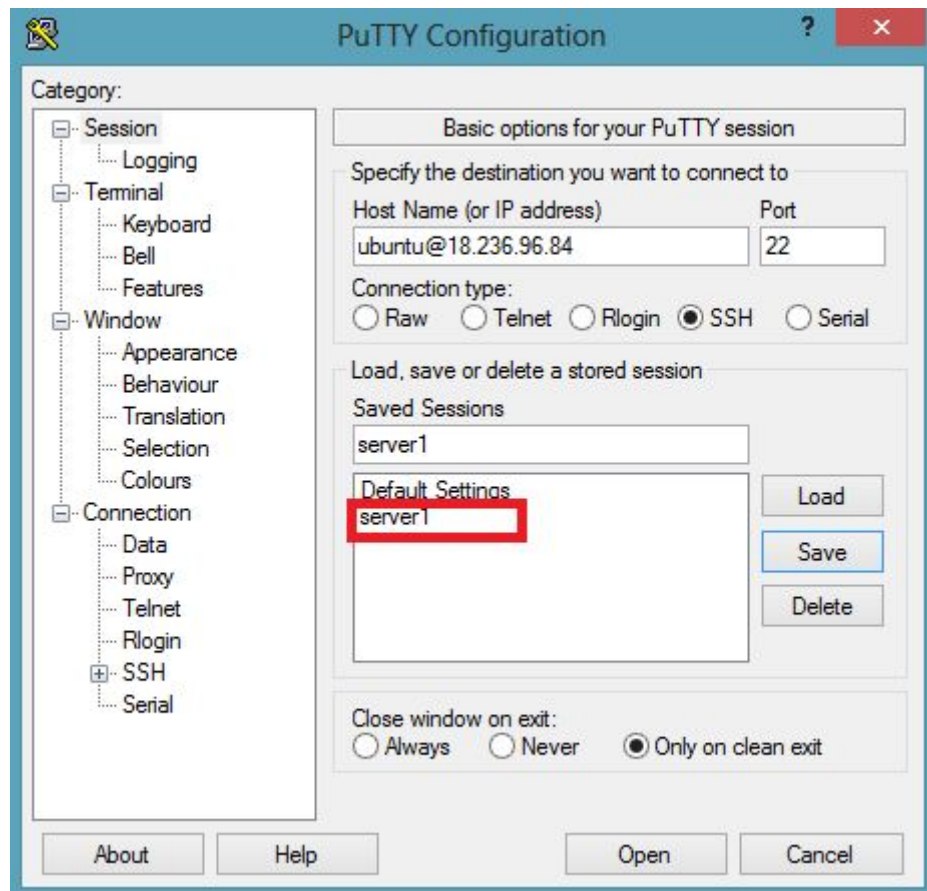
Нажимаем на **Browse** и выбираем файл **.ppk**, который мы получили после конвертации:



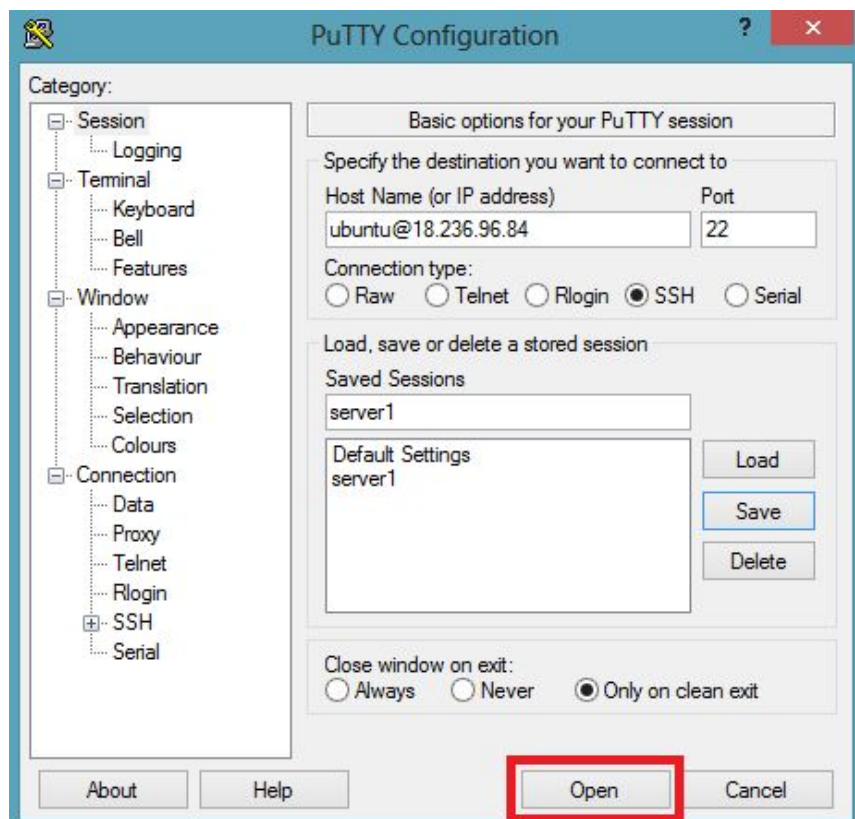
Далее нужно вернуться в раздел Session и нажать на кнопку Save:



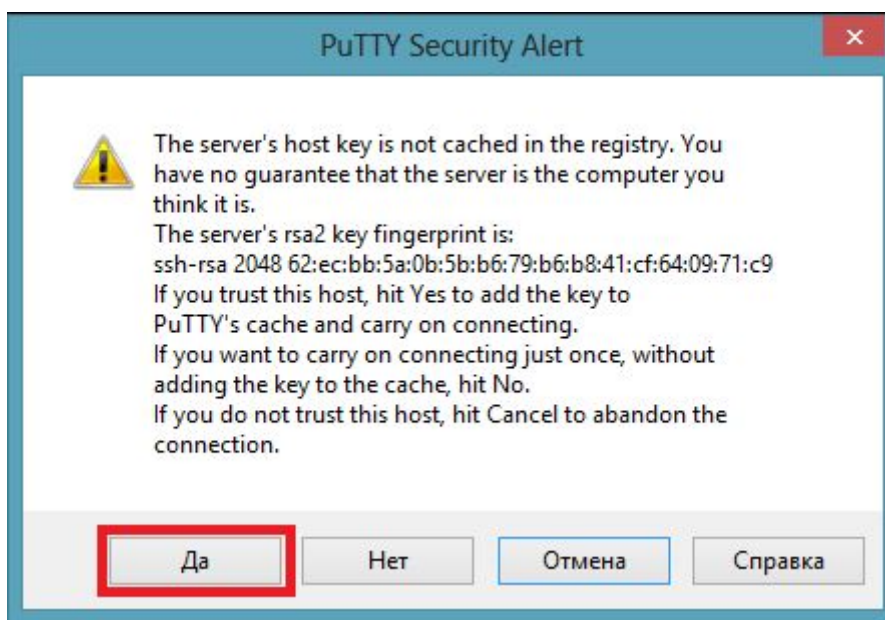
Название подключения **server1** появится в списке подключений:



Чтобы зайти на сервер, нужно либо кликнуть на название подключения, либо выделить это название и нажать на кнопку **Open**:



Так как мы впервые заходим на этот сервер, появится всплывающее окно с предупреждением. Нужно нажать «Да»:



Загрузится терминал сервера, в котором мы будем работать на следующих уроках:

```
Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-29-5:~$
```

В конце каждого сеанса рекомендуем завершать работу с сервером в **EC2 Management Console** командой **Terminate** (она доступна через меню **Actions-Instance State**), а после — проверять в **EC2 Dashboard**, что нет запущенных экземпляров и используемых дисков (**0 Running Instances** и **0 Volumes**). А для новых занятий создавать новый экземпляр, чтобы закреплять навыки. При этом можно использовать имеющийся приватный ключ.

Работу с утилитой WinSCP подробно рассмотрим в последующих уроках.

Практическое задание

1. Создать аккаунт в Amazon Web Services.

2. Создать экземпляр на сайте AWS.
3. Подключиться к экземпляру с помощью Putty.

Дополнительные материалы

1. [Уровень бесплатного пользования AWS.](#)
2. [Сказ о том, как я лоханулся с Amazon Web Services.](#)
3. [Что такое облачные вычисления?](#)

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. [Amazon Web Services — что это такое?](#)
2. [Центр ресурсов для начала работы.](#)
3. [Как пользоваться Putty.](#)