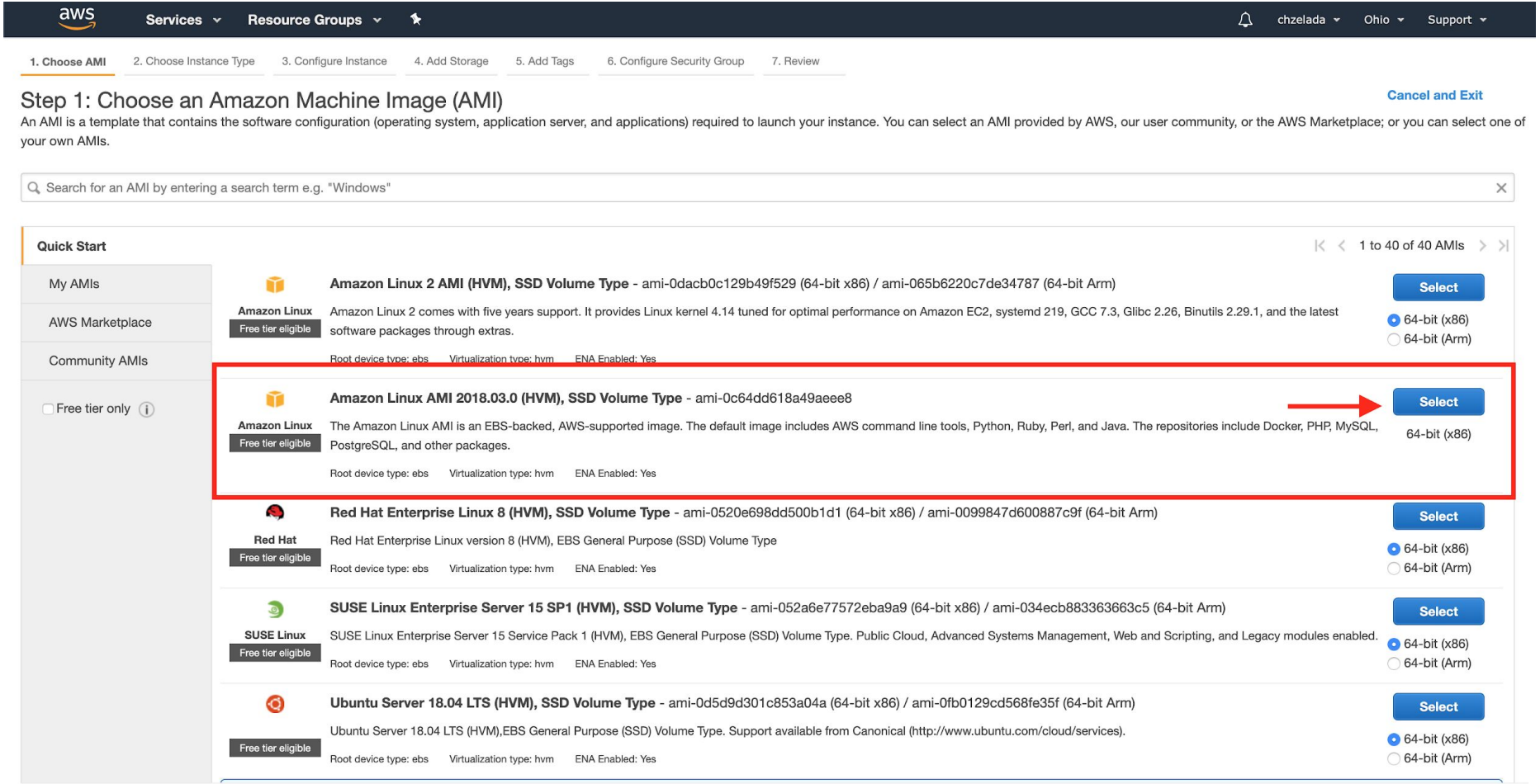
Docker en EC2

Carlos Zelada


aws
Services
Resource Groups
chzelada
Ohio
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
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"
Quick Start
1 to 40 of 40 AMIs
My AMIs
AWS Marketplace
Community AMIs
Free tier only
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0dacb0c129b49f529 (64-bit x86) / ami-065b6220c7de34787 (64-bit Arm)
Amazon Linux
Free tier eligible
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 AMI 2018.03.0 (HVM), SSD Volume Type - ami-0c64dd618a49aeee8
Amazon Linux
Free tier eligible
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 Enterprise Linux 8 (HVM), SSD Volume Type - ami-0520e698dd500b1d1 (64-bit x86) / ami-0099847d600887c9f (64-bit Arm)
Red Hat
Free tier eligible
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 Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-052a6e77572eba9a9 (64-bit x86) / ami-034ecb883363663c5 (64-bit Arm)
SUSE Linux
Free tier eligible
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.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select
64-bit (x86)
64-bit (Arm)
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0d5d9d301c853a04a (64-bit x86) / ami-0fb0129cd568fe35f (64-bit Arm)
Free tier eligible
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select
64-bit (x86)
64-bit (Arm)

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

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
☐	General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

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

Edit AMI

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0c64dd618a49aeee8

Free tier eligible

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

Instance Type

Edit 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

Edit security groups

Security group name launch-wizard-7
Description launch-wizard-7 created 2019-11-24T09:09:11.094-06:00

Type	Protocol	Port Range	Source	Description

This security group has no rules

Instance Details

Edit instance details

Storage

Edit storage

Tags

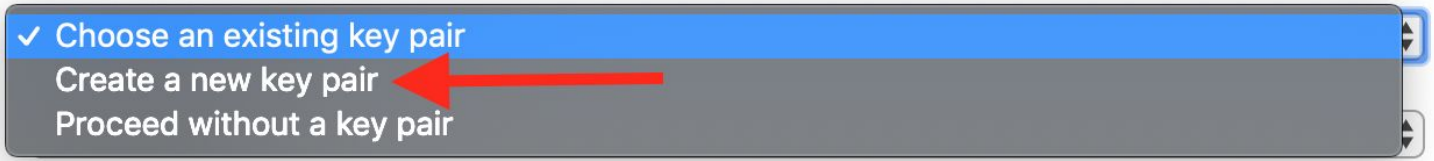
Edit tags

Cancel Previous **Launch**

Select an existing key pair or create a new key pair

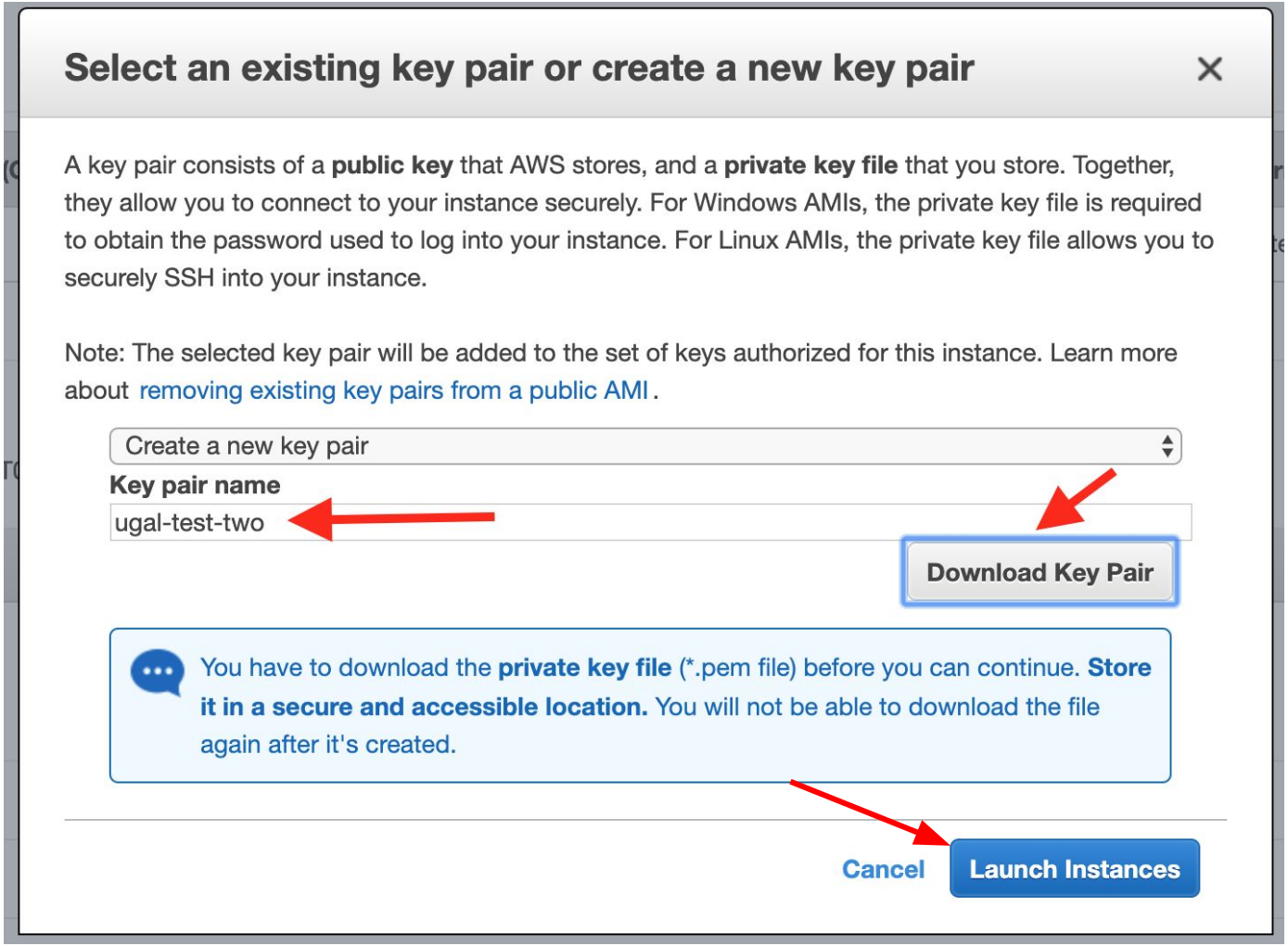
A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key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

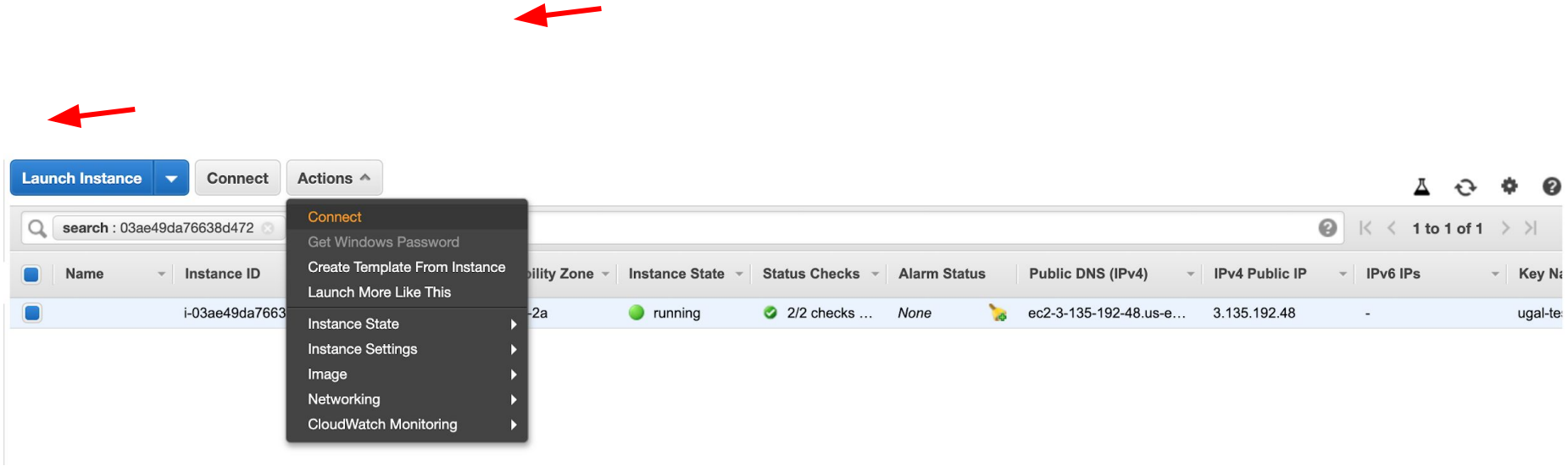
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

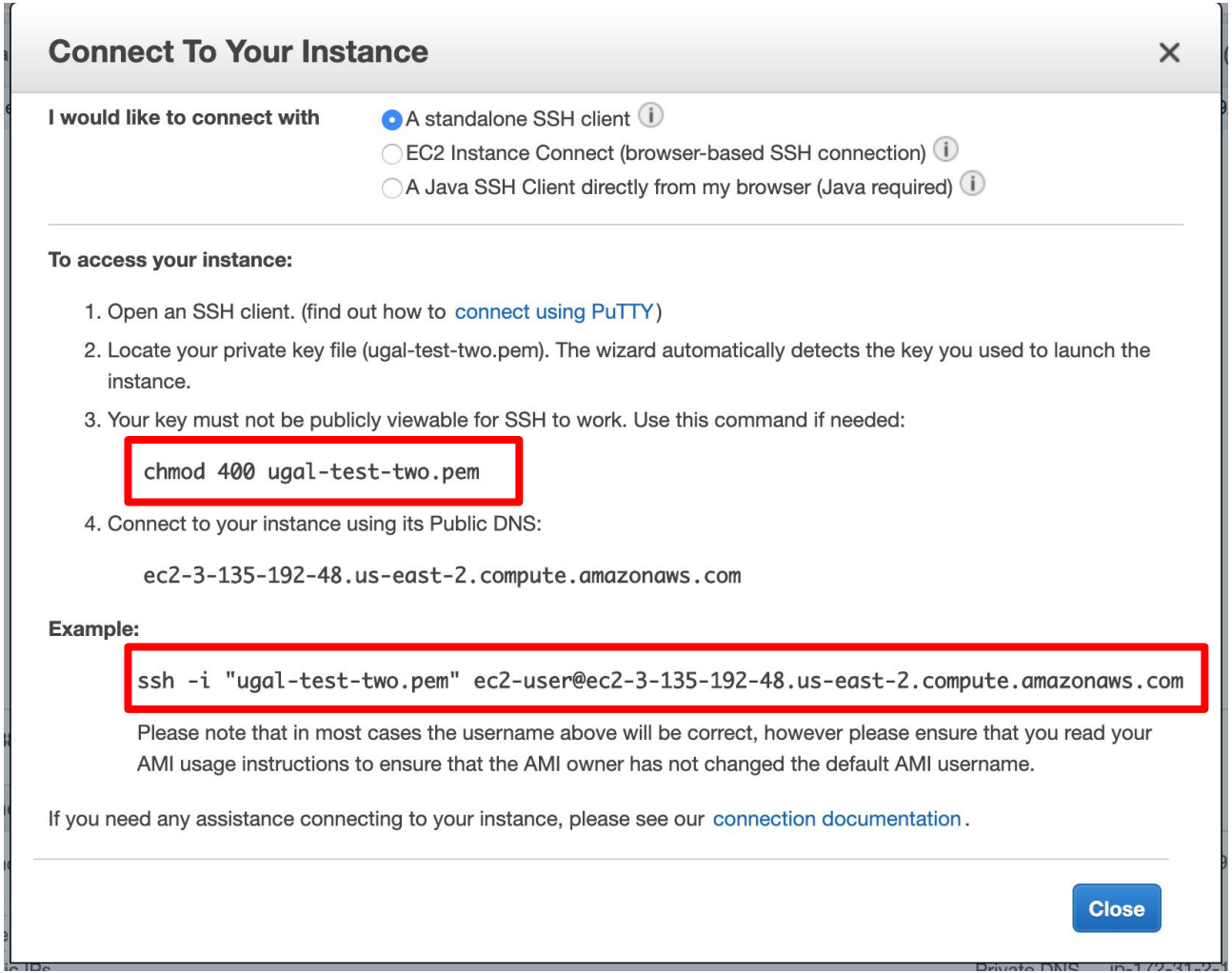


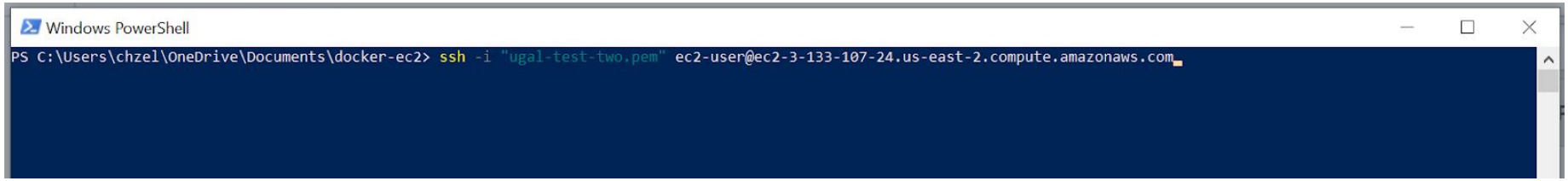
I acknowledge that I have access to the selected private key file (docker-test.pem), and that without this file, I won't be able to log into my instance.

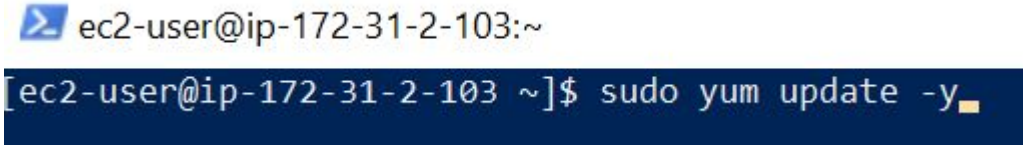
Cancel Launch Instances

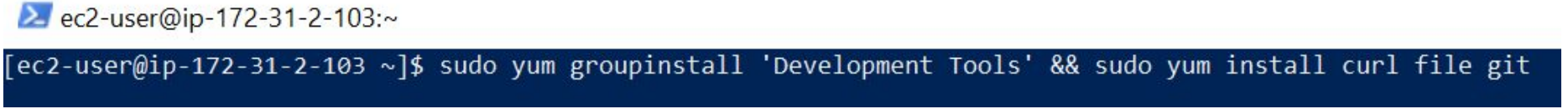

Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your 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.
Create a new key pair
Key pair name
ugal-test-two
Download Key Pair
You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.
Cancel
Launch Instances

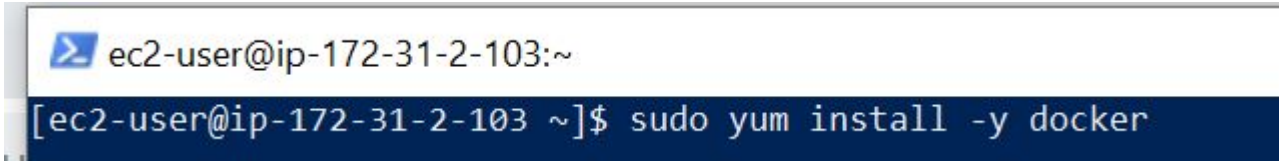

Launch Instance
Connect
Actions
Connect
Get Windows Password
Create Template From Instance
Launch More Like This
Instance State
Instance Settings
Image
Networking
CloudWatch Monitoring
search : 03ae49da76638d472
1 to 1 of 1
Name
Instance ID
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
IPv4 Public IP
IPv6 IPs
running
2/2 checks ...
None
ec2-3-135-192-48.us-e...
3.135.192.48

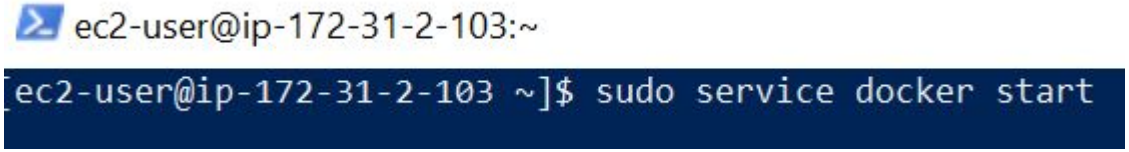

Connect To Your Instance
I would like to connect with
A standalone SSH client
EC2 Instance Connect (browser-based SSH connection)
A Java SSH Client directly from my browser (Java required)
To access your instance:
1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (ugal-test-two.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 ugal-test-two.pem
4. Connect to your instance using its Public DNS:
ec2-3-135-192-48.us-east-2.compute.amazonaws.com
Example:
ssh -i "ugal-test-two.pem" ec2-user@ec2-3-135-192-48.us-east-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.
Close

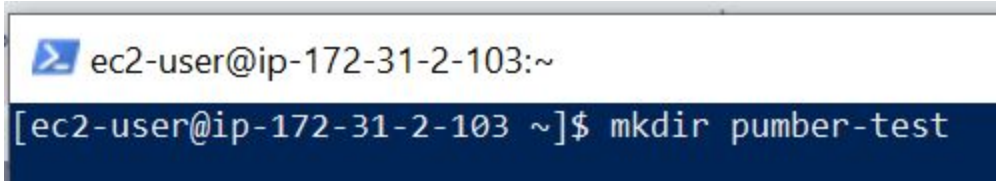

Windows PowerShell
PS C:\Users\chzel\OneDrive\Documents\docker-ec2> ssh -i "ugal-test-two.pem" ec2-user@ec2-3-133-107-24.us-east-2.compute.amazonaws.com

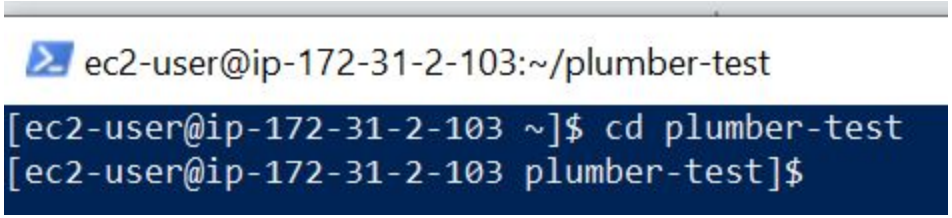

ec2-user@ip-172-31-2-103:~
[ec2-user@ip-172-31-2-103 ~]$ sudo yum update -y

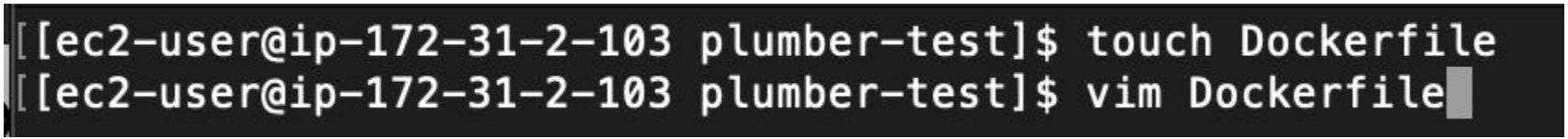

ec2-user@ip-172-31-2-103:~
[ec2-user@ip-172-31-2-103 ~]$ sudo yum groupinstall 'Development Tools' && sudo yum install curl file git


ec2-user@ip-172-31-2-103:~
[ec2-user@ip-172-31-2-103 ~]$ sudo yum install -y docker


ec2-user@ip-172-31-2-103:~
[ec2-user@ip-172-31-2-103 ~]$ sudo service docker start


ec2-user@ip-172-31-2-103:~
[ec2-user@ip-172-31-2-103 ~]$ mkdir pumber-test


ec2-user@ip-172-31-2-103:~/plumber-test
[ec2-user@ip-172-31-2-103 ~]$ cd plumber-test
[ec2-user@ip-172-31-2-103 plumber-test]$


[[ec2-user@ip-172-31-2-103 plumber-test]$ touch Dockerfile
[[ec2-user@ip-172-31-2-103 plumber-test]$ vim Dockerfile

```
# start from the rocker/r-ver:3.5.0 image
FROM rocker/r-ver:3.5.0

# install the linux libraries needed for plumber
RUN apt-get update -qq && apt-get install -y \
  libssl-dev \
  libcurl4-gnutls-dev

# install plumber
RUN R -e "install.packages('plumber')"

# copy everything from the current directory into the container
COPY / /

# open port 8888 to traffic
EXPOSE 8888

# when the container starts, start the main.R script
ENTRYPOINT ["Rscript", "main.R”]
```

- **:wq** - write and quit
- **:q!** - quit no save
- **[ESC] dd** - delete line
- **i** - editar
- **[ESC]** - Salir edicion
- mas

```
[ec2-user@ip-172-31-2-103 plumber-test]$ touch plumber.R
[ec2-user@ip-172-31-2-103 plumber-test]$ touch main.R
```

```
[ec2-user@ip-172-31-2-103 plumber-test]$ ls
Dockerfile  main.R  plumber.R
```

```
sudo docker build -t plumber_test .
```

```
Successfully tagged plumber_test:latest
[ec2-user@ip-172-31-2-103 plumber-test]$ sudo docker run --rm -p 8888:8888 plumber_test
```

Plumber

```
[[ec2-user@ip-172-31-2-103 plumber-test]$ sudo docker run -d --rm -p 8888:8888 plumber_test
236c1f71cf07ec0654c884a8ea8a74044762e04ac9540a3eea86e76fdf83c391
[[ec2-user@ip-172-31-2-103 plumber-test]$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                    NAMES
236c1f71cf07        plumber_test        "Rscript main.R"    7 seconds ago       Up 6 seconds        0.0.0.0:8888->8888/tcp   gallant_bohr
```