# AWS CLI H2 - Computing

**Install**

*curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"*

*sudo yum install unzip*

*unzip awscliv2.zip*

*sudo ./aws/install*

*aws --version*

**First Config (stored in .aws/credentials)**

*aws config*

**Create Security Group for specific VPC id (VPC-id can be omitted)**

*aws ec2 create-security-group --group-name my-sg --description "My security group" --vpc-id vpc-1a2b3c4d*

**Configure Security Group**

*aws ec2 authorize-security-group-ingress --group-name my-sg --protocol tcp --port 22 --cidr 0.0.0.0/0*

for Outbound rules

*aws ec2 authorize-security-group-egress --group-name my-sg --protocol tcp --port 22 --cidr 0.0.0.0/0*

use group-id to be specific instead of group-name

*aws ec2 authorize-security-group-ingress --group-id sg-0123456789abcdef0 --protocol tcp --port 22 --cidr 0.0.0.0/0*

**Describe security group**

*aws ec2 describe-security-groups --group-ids sg-903004f8*

**Delete security Group**

*aws ec2 delete-security-group --group-id sg-903004f8*

**GET AMI ID**

***Ubuntu*** (ami-0557a15b87f6559cf)

*aws ec2 describe-images --owners amazon --filters "Name=name,Values=ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-\*" "Name=virtualization-type,Values=hvm" --query 'Images[\*].[Name,ImageId]' --output text*

***RHEL*** (ami-0c9978668f8d55984)

*aws ec2 describe-images --owners amazon --filters "Name=name,Values=\*" "Name=virtualization-type,Values=hvm" --query 'Images[\*].[Name,ImageId]' --region us-east-1 --output text | grep RHEL-9*

SELF

*aws ec2 describe-images --owners self*

List of versions and names

Ubuntu 14.04 Trusty Tahr

Ubuntu 14.10 Utopic Unicorn

Ubuntu 15.04 Vivid Vervet

Ubuntu 15.10 Wily Werewolf

Ubuntu 16.04 Xenial Xerus (LTS)

Ubuntu 16.10 Yakkety Yak

Ubuntu 17.04 Zesty Zapus

Ubuntu 17.10 Artful Aardvark

Ubuntu 18.04 Bionic Beaver (LTS)

Ubuntu 18.10 Cosmic Cuttlefish

Ubuntu 19.04 Disco Dingo

Ubuntu 19.10 Eoan Ermine

Ubuntu 20.04 Focal Fossa (LTS)

Ubuntu 20.10 Groovy Gorilla

Ubuntu 21.04 Hirsute Hippo

Ubuntu 21.10 Impish Indri

Ubuntu 22.04 Jammy Jellyfish (LTS)

***Amazon Linux*** (ami-00c39f71452c08778)

*aws ec2 describe-images --owners amazon --filters "Name=name,Values=\*al2023-ami-2023\*" --query 'Images[\*].[Name,ImageId]' --region us-east-1 --output text*

**New EC2 Instance**

*aws ec2 run-instances --image-id ami-0c9978668f8d55984 --instance-type t2.micro --key-name vockey --security-group-ids sg-020cfc0d6cdba92da --tag-specifications 'ResourceType=instance, Tags=[{Key=Name, Value=my-first-cli-ec2}]' --count 1*

<securityid> = sg-020cfc0d6cdba92da

<ami-id> = ami-0c9978668f8d55984

**Describe instances**

*aws ec2 describe-instances | grep ‘Value\|InstanceId’*

*aws ec2 describe-instances --filters "Name=tag:Name,Values=back-end"*

**Create AMI based off VM**

*aws ec2 create-image --instance-id i-083f292d00c9cffbc --name "base docker compose image" --description "AMI image created from EC2 base"*

<instance-id> = instance-id from EC2 base

**Delete AMI**

*aws ec2 deregister-image --image-id ami-08754c19d0a63938f*

**SSH into instance from AWS CLI**

- describe instance to gather public Ipv4

- chmod 400 cloud\_vockey.pem

- ssh -i "cloud\_vockey.pem" ec2-user@ec2-54-162-241-237.compute-1.amazonaws.com

**Install docker Ubuntu/Amazon Linux**

sudo yum install -y git

sudo yum install docker

sudo service docker start

sudo usermod -a -G docker ec2-user

sudo mkdir -p /usr/local/lib/docker/cli-plugins/

sudo curl -SL https://github.com/docker/compose/releases/latest/download/docker-compose-linux-x86\_64 -o /usr/local/lib/docker/cli-plugins/docker-compose

sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose

**Install Docker RHEL**

sudo dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo

sudo dnf repolist -v

sudo dnf install --nobest docker-ce

sudo systemctl enable --now docker

sudo docker run hello-world

sudo usermod -aG docker ec2-user

sudo reboot now

***Note: Create AMI after docker-config allows new instances to retain docker configuration***

**Stop all running instances**

aws ec2 stop-instances --instance-ids $(aws ec2 describe-instances --query 'Reservations[\*].Instances[?State.Name==`running`].InstanceId' --output text)

**Troubleshooting Connectivity Issues**

1. Security Groups might be a problem --> Inbound rules?

2. vockey.pem needs permissions: chmod 400 vockey.pem on bastion

3. use Correct username for SSH connect?

4. Check container logs

5. IP addresses change on new lab startup, use aws ec2 describe   
 instances to find correct public ipv4

6. Check health of API in case DB containers are giving errors

7. **Remember new vockey after each Lab reset!!!**