**AWS Certification**

Inhaltsverzeichnis

[SSH 2](#_Toc33683281)

[Windows 2](#_Toc33683282)

[Windows 10 2](#_Toc33683283)

[Linux/Mac 2](#_Toc33683284)

[EC2 2](#_Toc33683285)

[EC2 Instance Types 2](#_Toc33683286)

[EC2 User Data 2](#_Toc33683287)

[EC2 AMI 3](#_Toc33683288)

[EC2 Launch Types 3](#_Toc33683289)

[Creating Linux (Bootstrap) Instance 4](#_Toc33683290)

[Security Groups 4](#_Toc33683291)

[Good to Know 4](#_Toc33683292)

[EC2 Mid Point Quiz 5](#_Toc33683293)

# SSH

## Windows

1. Load PrivateKey in PuttyGen – Select All Files to view .pem
2. Save Private Key (filename.ppk)
3. Open Putty
4. Enter Connection Data
   1. Host name (ec2-user@PublicIP address)
   2. Saved Sessions – Session name
   3. Connection/SSH/Auth/Browse – file.ppk
   4. Click on Session – Save

## Windows 10

1. Right Click .pem file
2. Eigenshaften
3. Select Sicherheit / Erweitert
4. Besitzer: Love Richard ([p171100@mesroot.net](mailto:p171100@mesroot.net))
5. Erweitert
6. Vererbung deaktivieren
   1. Select System – Entfernen
   2. Select Administrator – Entfernen
   3. OK
7. Open PowerShell tool
8. Enter: SSH –i (filePath\fileName) ec2-user@PUBLICIP

## Linux/Mac

1. Place .pem file in Root Folder
2. From Terminal – chmod 0400 filename.pem
3. From Terminal – ssh –I filename.pem ec2-user@PUBLICIP

# EC2

<https://ec2instances.info>

EC2 – Hibernate

* In memory (RAM) state is preserved
* Instance Boot is much faster
* RAM state is written to file in the root EBS volume
* Root volume must be encrypted
  + ONLY supported by: C3, C4, C5, M3, M4, M5, R3, R4 and R5
  + Must be less than 150GB
  + Not supported for bare metal instances
  + AMI: Linux 2 + Linux AMI, Ubuntu and Windows
  + Only available for reserved and On-demand instances
  + 60 day hibernation limit

## EC2 Instance Types

* R: Application that need a lot of RAM – in Memory caches
* C: applications that needs good CPU – compute / database
* M: applications that are balanced (think “medium”) – general /web application
* I: applications that need good local I/O (instance storage) – databases
* G: applications that need a GPU – video rendering / machine learning
* T2 / T3: burst instances (up to capacity)
* T2 / T3: unlimited: unlimited burst

## EC2 User Data

1. It is possible to create a EC2 User Data script
   1. Script will only run once when machine starts
   2. Only when Instance is first started
2. Script is used to automate boot tasks
   1. Install updates
   2. Install software
   3. Download common files from Internet
   4. And….
3. EC2 user Data Script runs as the root user

## EC2 AMI

AMI’s are built for a specific region (!) They are not global (!)

* AMI is an Image used to create instances
* Can be built for Linux or Windows Machines

1. Pre-installed packages that are needed
2. Faster Boot time (no need for ec2 user data at boot time)
3. Machine comes configured with monitoring / enterprise software
4. Security concerns – control over machines in network
5. Control of maintenance and updates of AMIs over time
6. Active Directory Integration Out of the Box
7. Installing your app ahead of time (for fast deploys when auto-scaling)
8. Use someone else’s AMI that is pre-configured

Cross Account AMI Copy

* You can share AMI with another AWS Account
  + This does not accept ownership
  + If you copy an AMI shared with you, you are the owner of the target AMI in your account
* Limits
  + Cannot copy encrypted AMI shared from another account
  + Cannot copy AMI with associated billing Point code shared with you
    - To copy, launch EC2 instance with shared AMI, then create a AMI from this instance.

Public AMIs:

* You can rent other people’s AMI by the hour
* Don’t use AMI you don’t trust
* Some AMIs might come with malware and not be secure

AMI Storage

* AMI lives in S3
* S3 is where most backups will live
* AMIs are private by default and lock for your account by region
* Can make AMIs public and share with other AWS accounts or sell them on AMI market place
* 0.02 per Gig for first 50TB
* Delete old AMIs

Creating an Image

* Right Click Instance
* Create Image
* Name / Description – Create Image
* Attach Security and Key

# EC2 Launch Types

1. OnDemand:
2. Reserved: Instances:
   1. Long workloads (>= 1 year)
3. Convertible Reserved Instances:
   1. Long workloads with flexible instances
4. Scheduled Reserved Instances:
   1. Launch within a time window you reserve
5. Spot Instances:
   1. Short workloads, for cheap, can lose instances
6. Dedicated Instances:
   1. No other Customers will share the instance
7. Dedicated hosts:
   1. Book entire physical server, control instance placement

## Creating Linux (Bootstrap) Instance

1. Create Amazon Linux 2 AMI
2. Select t2.micro
   1. Configure Instance Details
3. Advanced Details
   1. User Data:

#!/bin/bash

# install httpd (Linux2 version)

yum update -y

yum install -y httpd.x86\_64

systemctl start httpd.service

systemctl enable httpd.service

echo "Hallo zusammen $(hostname -f)" > /var/www/html/index.html

* Ssh –i (filePath of local .pem file) ec2-user@ip to connect
  + ssh -i C:\Users\p171100\Desktop\2020\Training\AWS\KeyName.pem ec2-user@PublicIP
* Sudo su
* Yum update –y
* Curl localhost:80 – to see if it all worked

# Security Groups

## Good to Know

1. Can be attached to multiple instances
2. Act as a “firewall” to EC2 instance
3. Regulate
   1. Access to ports
   2. Authorize IP Ranges
   3. Control Inbound and Outbound Network
4. Instance can have multiple security groups
5. Security Groups are locked down to a REGION / VPC combination
6. Live OUTSIDE the EC2 instance
7. Good to have one Security group for SSH access
8. If the application is not accessible (time out), then it’s a security group issue
9. If the application gives a “Connection refused” error, then it’s an application error or it’s not launched
10. Default – Inbound traffic is blocked
11. Default – Outbound traffic is authorized

# EC2 Mid Point Quiz

1. Any EC2 that ends in a “letter” is an “availability Zone” - **ap-northeast-1a**
2. Availability Zone are in isolated centers –
   1. this helps guarantee that multi AZ won't all fail at once (due to a meteorological disaster for example). Read more here: <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.RegionsAndAvailabilityZones.html>
3. Organizations DON’T exist in IAM
4. IAM Users are NOT defined by region
   1. IAM is global and encompasses all regions
5. IAM User can belong to multiple groups
6. You are getting started with AWS and your manager wants things to remain simple yet secure. He wants management of engineers to be easy, and not re-invent the wheel every time someone joins your company. What will you do?
   1. Create multiple IAM Users and Groups, and assign policies to the Groups. New Users will be added to the Groups
7. You only pay for EC2 instance state when it is “running”.
8. You are getting a permission error exception when trying to SSH into your Linux Instance
   1. The key is missing permissions chmod 0400
9. You are getting a network timeout when trying to SSH into your EC2 instance
   1. Any timeout errors (not just in SSH but also HTTP for example) means a misconfiguration of your security groups
10. When a security group is created, what is the default behavior?
    1. Deny ALL inbound traffic and allow ALL outbound traffic
11. Security Groups can reference:
    1. IP address
    2. CIDR Block
    3. Other Security Groups
12. Security Groups CAN NOT reference:
    1. DNS Names
13. You want to provide startup instructions to your EC2 instances, you should be using:
    1. EC2 User Data

# Placement Groups

## 3 Placement Group strategies

### Cluster

* Places instances into low latency group in SINGLE availability zone
* Pro – Great network (10gbps between instances)
* Con – Rack fails, all instances fail at the same time
  + Use Case: Big Data job
  + Application needs extremely low latency and high network throughput

### Spread

* Spreads instances across underlying hardware
* Max of 7 instances per availability zone – critical applications
* Pro – Can span multiple Availability Zones
* Pro – Reduced risk for simultaneous failure
* Pro- EC2 on different hardware
* Con – Limited to 7 instances per availability zone
* Use Case –
  + Application needs to maximize high availability
  + Critical Applications where each EC2 must be isolated from failure

### Partition

* Spreads instances across many different partitions (which rely on different sets of racks) within an Availability Zone
* Hadoop, Cassandra, Kafka
* Can have 7 partitions per Avaailability Zone
* Can hold 100s of EC2 instances
* Instances do not share racks with the instances in other partitions
* Partition failure will not affect other Partitions
* Instances can gain partition metadata

# Elastic Network Interfaces

* Logical component that represents a virtual network card
* Can have the following attributes:
  + Primary private IPv4, one or more secondary IPv4
  + One elastic IP per private IPv4
  + One Public IPv4
  + One or more security groups
  + A MAC address
* You can create ENI independently and attach them on the fly to EC2 for failover
* Bound to a specific AZ.

# EC2 Exam Review

* EC2 is billed by the second
* Linux / Mac we use SSH Windows use Putty
* SSH in on port 22
* Timeout issues => Security Group Issues
* Permission issues on the SSH key => run “chmod 0400”
* Security groups can reference other Security Groups instead of IP Ranges
* Difference of Public, Private and Elastic IP address
* Create AMI to preinstall software on EC2 => faster boot
* AMI can be copied across regions and accounts
* An AMI created for a region, can only be seen in this region
* AMI is region locked and the same ID cannot be used across regions
* Creating an AMI after installing the applications allows you to start more EC2 instances directly from that AMI, hence bypassing the need to install the application (as it's already installed)
* EC2 Instances can be started in placement group
  + Cluster
  + Spread
  + Partition

### 4 EC2 Launch Modes

* Ondemand -
* Reserved
* Spot Instances
* Dedicated Hosts

### Know the Instance Types

* R
* C
* M
* I
* G
* T2/T3

# ELB (Internal + External) & ASG

## Availability:

### Vertical Scaling:

Increase instance size

### Horizontal Scaling:

Increase Number of Instances

### High Availability:

Run instances for the same application across multiple AZ

## Overview:

LB’s are servers that forward internet traffic to multiple EC2 Instances

Handles failures of downstream instances

Provides SSL (Https) for website

Enforce stickness with Cookies

High Availability

Separate Public and Private Traffic

## 

## Health Checks

Enable LB to know if Instance is available to reply to requests

If response is not 200 (OK), instance is unhealthy

## Troubleshooting

* 4xx errors are client induced errors
* 5xx errors are application induced errors
* LB Errors 503 means at capacity or no registered target
* If LB cannot connect to application, check security groups!

### Classic Load Balancer (Http, Https (Layer 7), TCP (Layer4))

### Application Load Balancer (Http, Https, WebSocket)

### Network Load Balancer (TCP, TLS (secure TCP), UDP)