Introduction - 2

SE234 Advance Software Development

CAMT
College of Arts, Media and Technology
Chiang Mai University

Course Overview

- Continuous Delivery
- Unit Test
- UAT Test
- Integration Test

Grading Criteria

• Attendance	5 %	0%
• Lab	10 %	10%
• Lab Exam	7.5 %	10%
• Project	7.5 %	10%
• Final Examination	20 %	20%

What are we going to learn?

- Deployment
- Container
- Continuous Integration and Delivery
- Automated Test Tools

    - DevOps

Let’s Start

After Finish Implementation

- What is the next step?

Searching for

- **What are the application deployment model?**
-

Deployment Model

- How can we deploy the application to the users
- Set up are required

Deployment Model – Stand alone application

- Fully install in one user computer
- Dependency is required
  - Providing by the setup media
  - Data store
  - Code functions

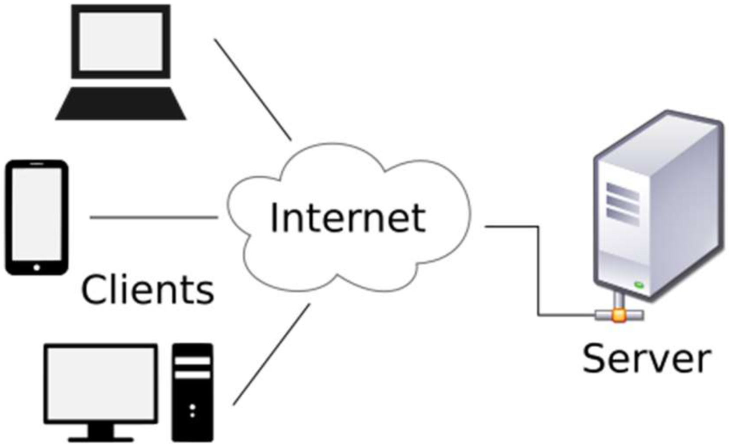
Deployment Model – Stand alone application

- Pros
  - Everything is on the computer
    - No internet required
    - No network required
- Cont.
  - Update problem

- Example?

Deployment Model – Client Server Architecture

- Server
  - Provide the service
  - Mostly data
- Client
  - A thick client
  - Require installation
  - Can work on many functions
    - But required data from the server



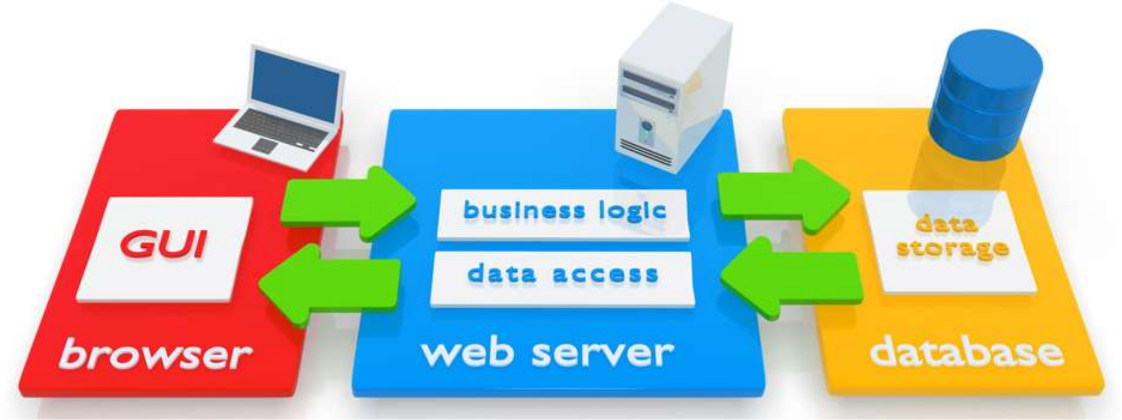
Deployment Model – Client Server Architecture

- Pros
  - Reduce the hardware requirements on the client side
  - Privacy of some data
    - Store in server
- Cons
  - Version consistency between servers and clients

- Examples?

Deployment Model – Web Application

- Thin clients
- The application run on the browser
- The main operations are executed on the web server



Deployment Model – Web Application

- Pros
  - Single point of maintenance
  - Work from everywhere
    - With internet and web browser
- Cons
  - Slow
    - If there are many user at the same time
  - Hard to customize to the regions
- Examples?

Deployment Model – Web Application

- Update version
  - Move some presentation rendering on the web browser
- Previously
  - Only HTML is sent
- Currently
  - With JS
  - Web application run with the Javascript on the client side
  - Request only the required data
  - Similar to the client server architecture.
  - VueJs, Angular, React

Deployment Model – Service Architecture

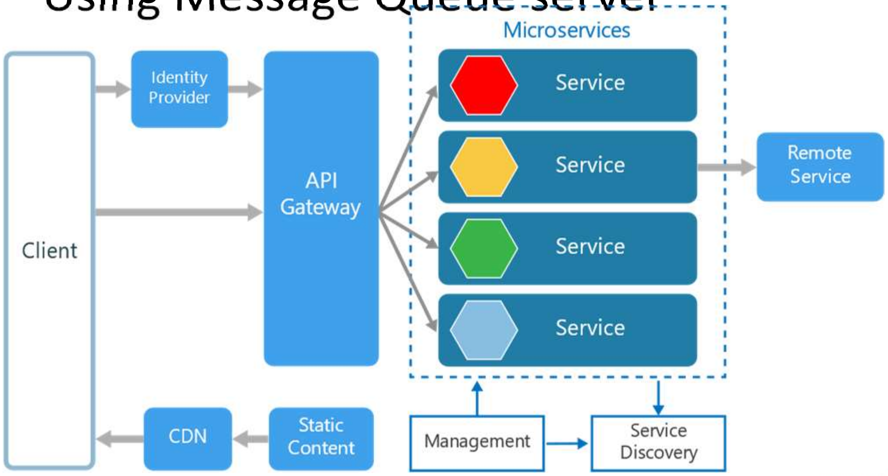
- Provide only a service
- No User interface
- Wait for other system to call for a service

Deployment Model – Service Architecture

- Pros
  - Separate the workload of the servers
  - Scalability
  - Several application can share the implementation
- Cons
  - Need to know the locations and end points
  - Security managements
- Examples
  - Public services

Deployment Model – Micro Service Architecture

- The application provides the service which is used only their own application
- Handle the call by themselves
  - Using Message Queue server



Deployment Model – Micro Service Architecture

- Pros
  - Privacy
    - All services can used in it own application
  - Scalability
    - Easy to add/remove the servers for each microservice
- Cons
  - Use a lot of computers
    - Container technology helps
- Example
  - Netflix, SCB Easy

Required Application

- To deploy your application in each model
  - Other applications are required
- Otherwise you have to write every thing from the scratch

Required Application – Installer

- Packed the implemented execution code
- Gather all dependency
  - Grasp it in the installation files
- Extract files in the target computer
- Setting the computer environments

CAMT
College of Arts, Media and Technology
Chiang Mai University

Required Application - Web server

- Provide the http service
- Allows users to browse on the html files provided by the developer
- Example
  - Nginx, apache

Required Application - Data repository

- Store the persistence information
- Can be used as a service
- Example
  - Database servers
    - MySql, MariaDB, SQL server
  - Repository service
    - Amazon S3, Google Firebase/cloudbase

Required Application - Application Containers

- The framework which provides OS specific operations
- Applications run on the containers
  - Application is not OS specific
  - Can run on the containers
  - In every servers
- Compiler/Interpreter hybrid
- Example
  - Tomcat/ .net Framework

Required Application - Message Queue Server

- Control the queue request for the microservice
- Similar to the load balancer
- Link between the service request and execution
- Example
  - RabbitMQ, ActiveMQ

What about hardware?

- If we use the model which is not the stand-alone model?
  - What are the hardware that we need?

On-premise server

- The company buy one computer for the company
- Pro
  - Fully control
  - Can manage the upgrade with out own cost
  - Privacy
- Con
  - Is it suitable for our application?
    - Cost/Performance
  - Maintenance cost

Cloud server

- Subscription
  - Rent from the provider
  - Mostly as the IAAS
    - Infrastructure as a service
  - Subscription fee depending on the size

Cloud Server

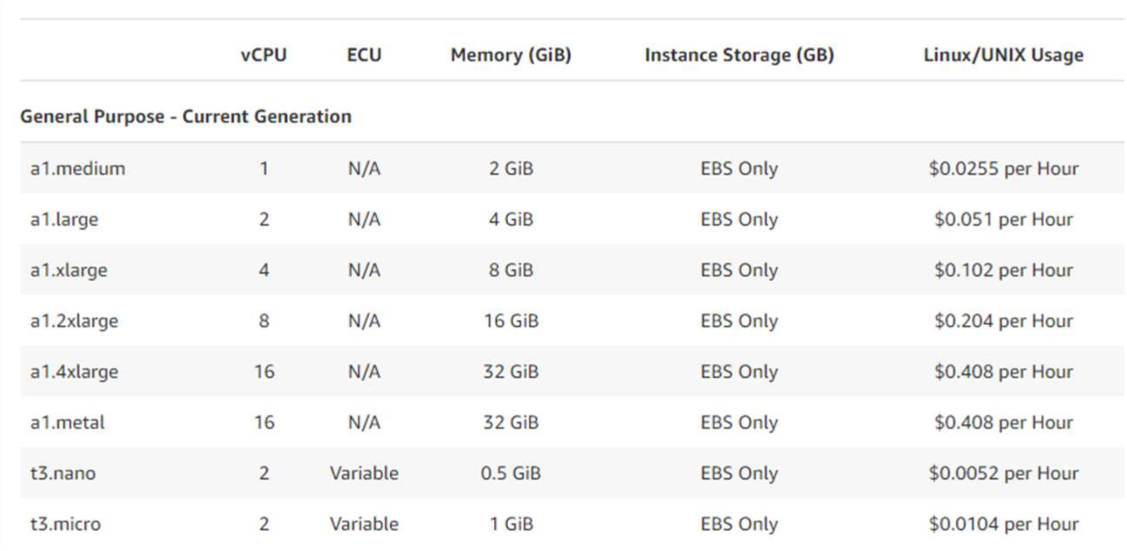
- Pro
  - Better Cost/performance
    - We can change the size to suit our work any time
  - Maintenance
    - Do not have to do it ourselves
  - Availability
- Con
  - Data security
  - Ownership

Type of IAAS

- Virtual Machine
  - Different name regarding to the operators
- Vendors
  - AWS – Elastic Cloud
  - Google – Compute Engine
  - Microsoft – Azure virtual machine
  - Digital Ocean – Droplet
  - etc

What we need to setup?

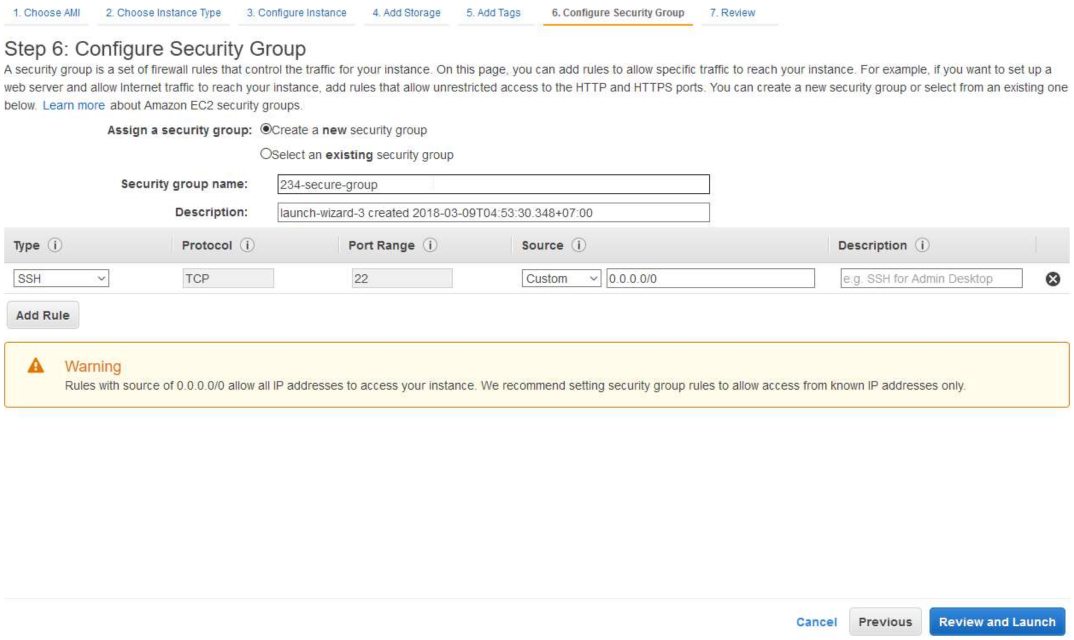
- The size of virtual machine
- Size and price is related
- With the OS



	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
a1.medium	1	N/A	2 GiB	EBS Only	$0.0255 per Hour
a1.large	2	N/A	4 GiB	EBS Only	$0.051 per Hour
a1.xlarge	4	N/A	8 GiB	EBS Only	$0.102 per Hour
a1.2xlarge	8	N/A	16 GiB	EBS Only	$0.204 per Hour
a1.4xlarge	16	N/A	32 GiB	EBS Only	$0.408 per Hour
a1.metal	16	N/A	32 GiB	EBS Only	$0.408 per Hour
t3.nano	2	Variable	0.5 GiB	EBS Only	$0.0052 per Hour
t3.micro	2	Variable	1 GiB	EBS Only	$0.0104 per Hour

Add the required resources

- Volume
  - Size of the hard disk
- Security group
  - Who can access your virtual machine



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: 234-secure-group

Description: launch-wizard-3 created 2018-03-09T04:53:30.348+07:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

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.

Security

- Protocol to be sent
  - TCP/UDP
- Allowed Ip address
  - 0.0.0.0 – allow all for IPV4
  - ::0 – allow all for IPV6
- Port
  - Port which allow to be accessed

Default Port Number

- Http - 80
- Https – 443
- Secure Shell – 22
- Ftp – 21
- And other ports we can use

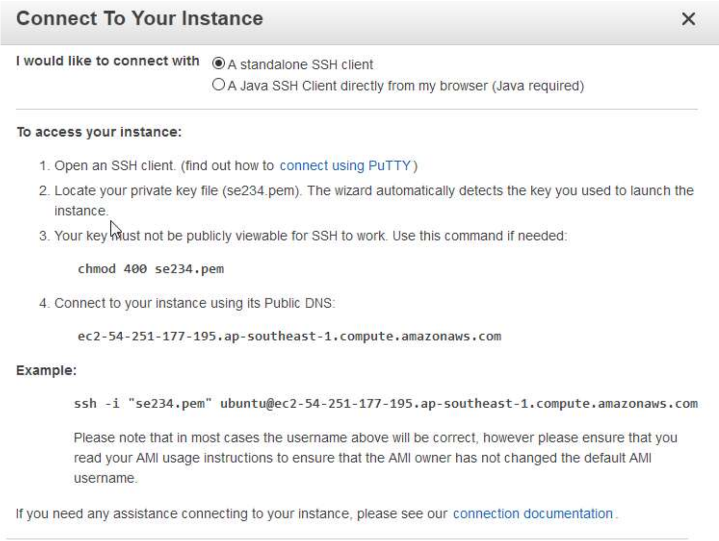
Securties

- What we know?
- What we have?
- What we are?

In aws,

- The public-private key is used
- The .pem (public key) is given
- In the vm, the private key is store
- Each connection must used with the public key

To connect we need Internet address



Connect To Your Instance ✕

I would like to connect with ● A standalone SSH client
○ 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 (se234.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 se234.pem
```

4. Connect to your instance using its Public DNS:

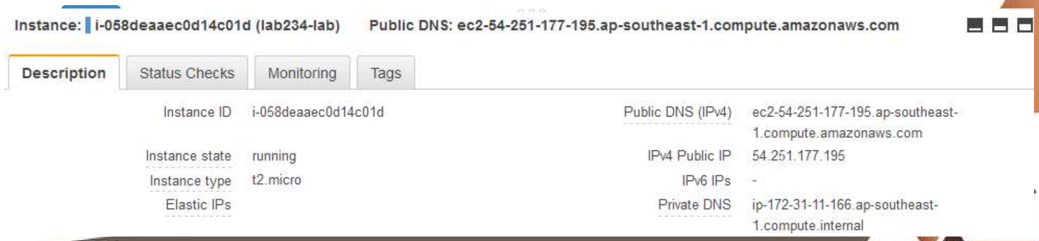
```
ec2-54-251-177-195.ap-southeast-1.compute.amazonaws.com
```

Example:

```
ssh -i "se234.pem" ubuntu@ec2-54-251-177-195.ap-southeast-1.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.



Instance: **i-058deaaec0d14c01d (lab234-lab)** **Public DNS: ec2-54-251-177-195.ap-southeast-1.compute.amazonaws.com**

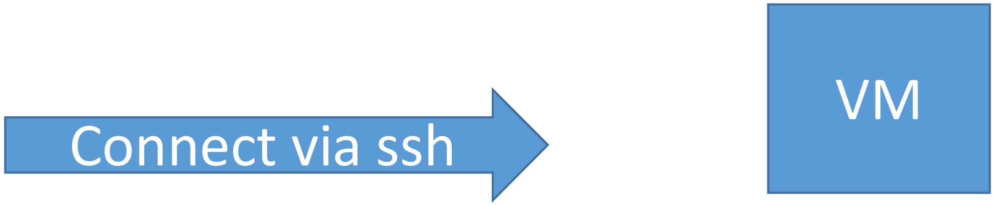
Instance ID	i-058deaaec0d14c01d	Public DNS (IPv4)	ec2-54-251-177-195.ap-southeast-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	54.251.177.195
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-11-166.ap-southeast-1.compute.internal

Internet address

- IP – internet Protocols
- URL – Uniform Resource Locator

Now we can connect and setup the VM

This Photo by Unknown Author is licensed under CC BY-SA



Connect via ssh

VM

Host Computer

Your computer

Ssh is

- Secure shell
  - Encrypt all connection with public-private key
- Connection are
  - Command line and response
  - Content sent
    - File, list of data (via the Winscp, Cyberduck)

The software Installation Guideline

- Regarding to the OS

Q & A