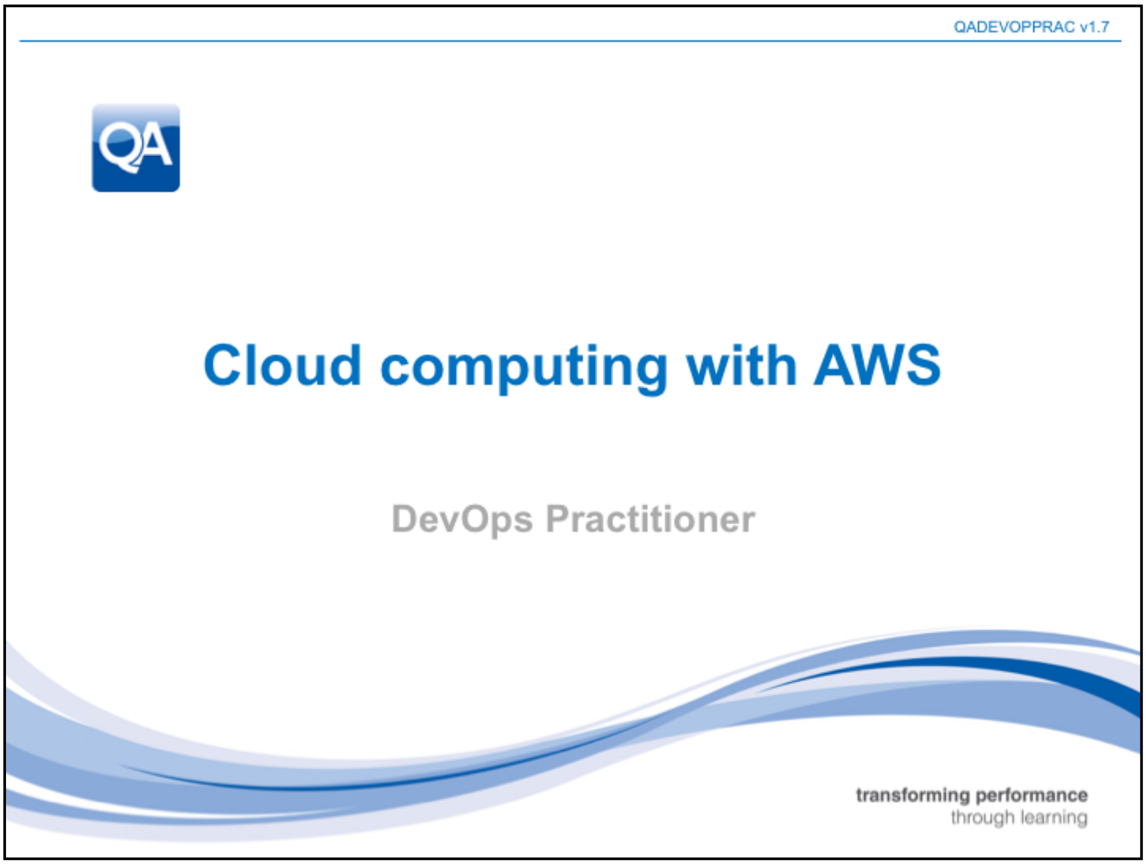
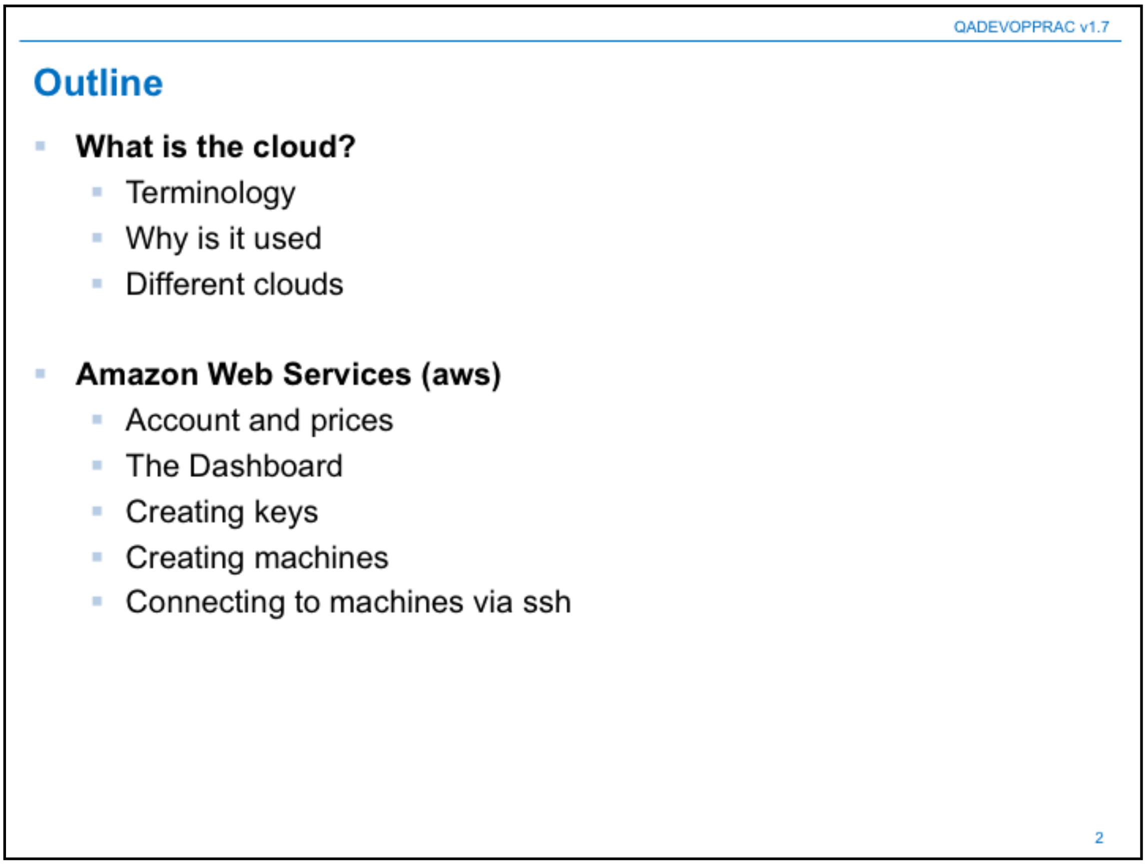


QADEVOPPRAC v1.7

Cloud computing with AWS

DevOps Practitioner

transforming performance
through learning

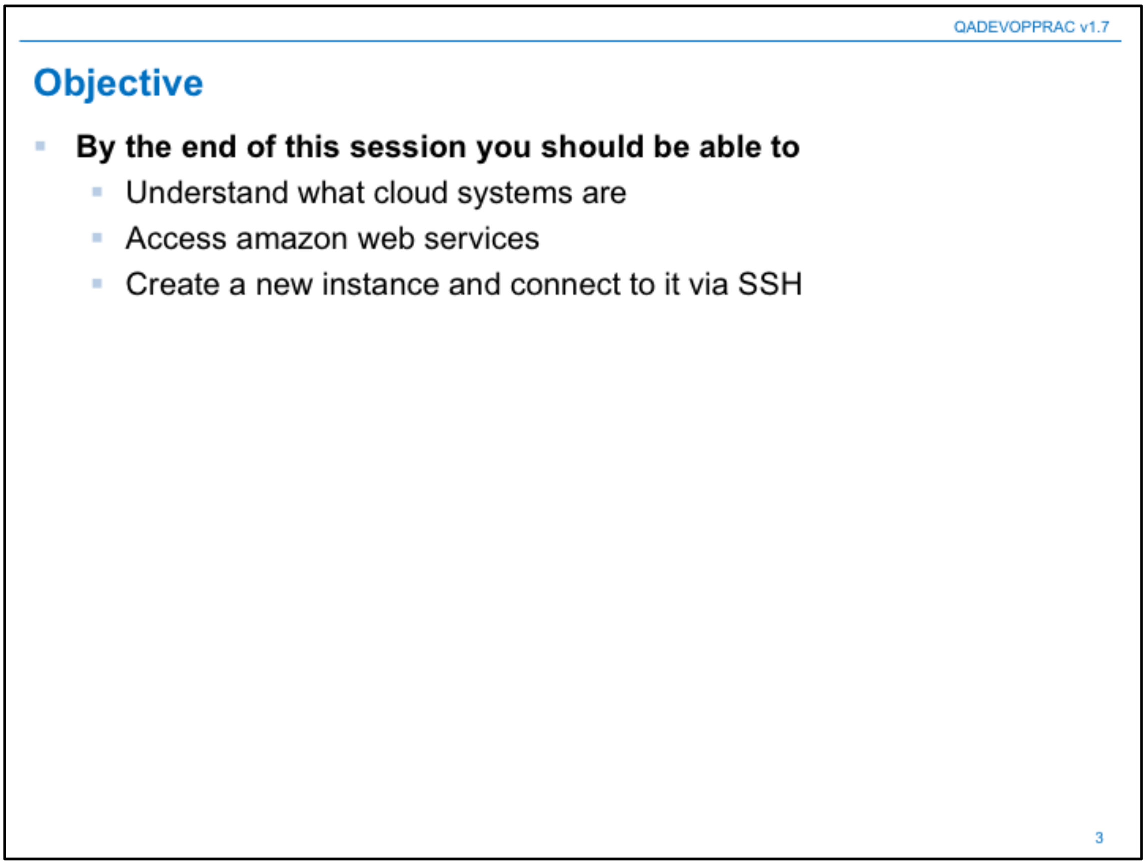


QADEVOPPRAC v1.7

Outline

- **What is the cloud?**
  - Terminology
  - Why is it used
  - Different clouds
- **Amazon Web Services (aws)**
  - Account and prices
  - The Dashboard
  - Creating keys
  - Creating machines
  - Connecting to machines via ssh

2

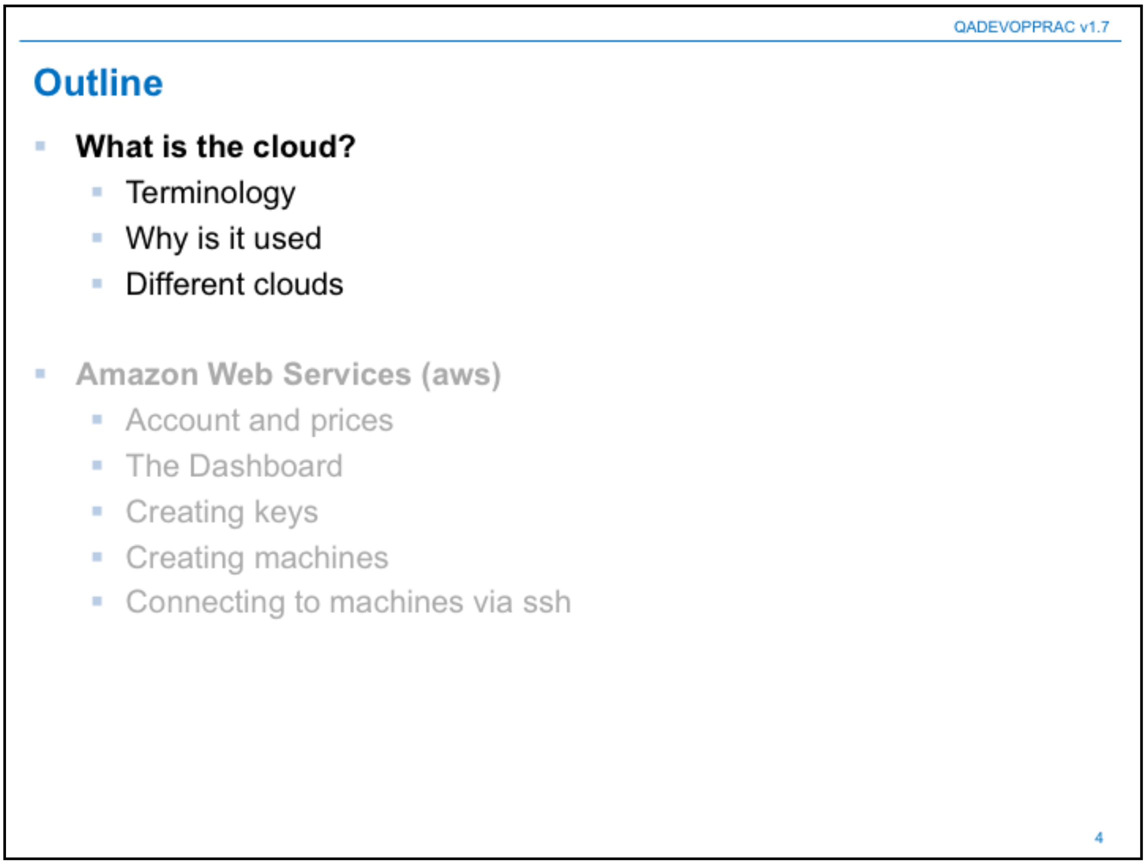


QADEVOPPRAC v1.7

Objective

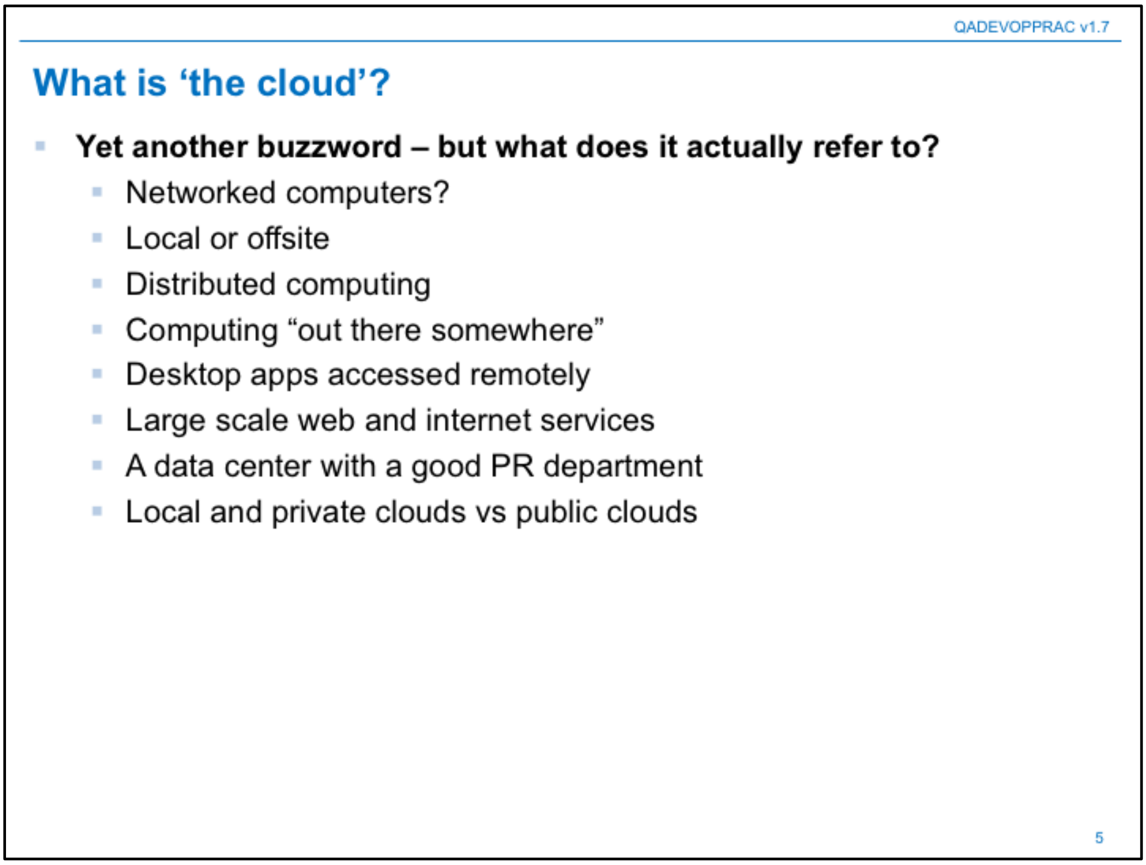
- **By the end of this session you should be able to**
  - Understand what cloud systems are
  - Access amazon web services
  - Create a new instance and connect to it via SSH

3



Outline

- **What is the cloud?**
  - Terminology
  - Why is it used
  - Different clouds
- **Amazon Web Services (aws)**
  - Account and prices
  - The Dashboard
  - Creating keys
  - Creating machines
  - Connecting to machines via ssh



QADEVOPPRAC v1.7

What is ‘the cloud’?

- **Yet another buzzword – but what does it actually refer to?**
  - Networked computers?
  - Local or offsite
  - Distributed computing
  - Computing “out there somewhere”
  - Desktop apps accessed remotely
  - Large scale web and internet services
  - A data center with a good PR department
  - Local and private clouds vs public clouds

5

The **cloud** is a network of servers, and each server has a different function.

Some servers use computing power to run applications or "deliver a service."

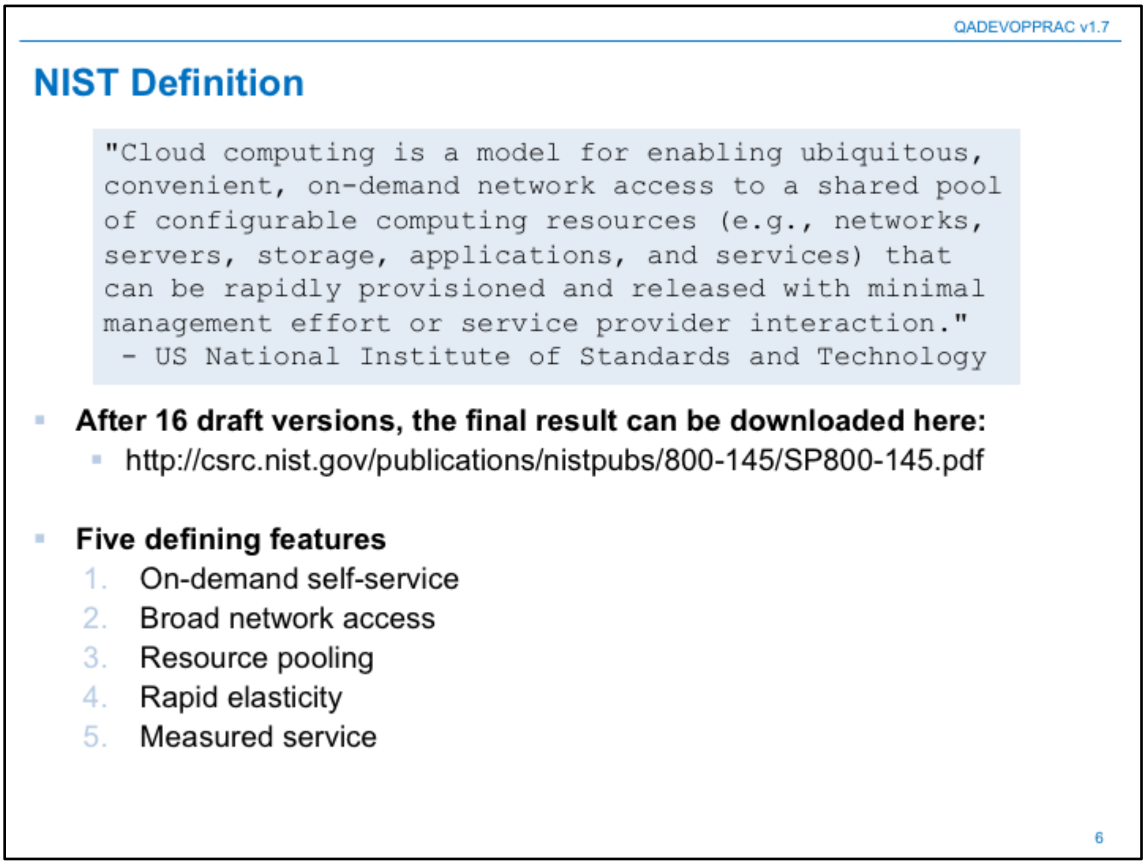
Other servers in the network are responsible for storing data.

“Cloud” is a buzzword that vaguely suggests the promise and convenience of being able to access files from anywhere.

And yet as long as it’s easy to read email on our phones and watch movies on our laptops, we generally don’t take the time to wonder

where our data actually goes, how it gets there, and what happens to it on its way.

PR – public related



QADEVOPPRAC v1.7

NIST Definition

```
"Cloud computing is a model for enabling ubiquitous,
convenient, on-demand network access to a shared pool
of configurable computing resources (e.g., networks,
servers, storage, applications, and services) that
can be rapidly provisioned and released with minimal
management effort or service provider interaction."
 - US National Institute of Standards and Technology
```

- **After 16 draft versions, the final result can be downloaded here:**
  - http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
- **Five defining features**
  1. On-demand self-service
  2. Broad network access
  3. Resource pooling
  4. Rapid elasticity
  5. Measured service

6

https://networkingexchangeblog.att.com/enterprise-business/5-essential-characteristics-of-cloud-computing/

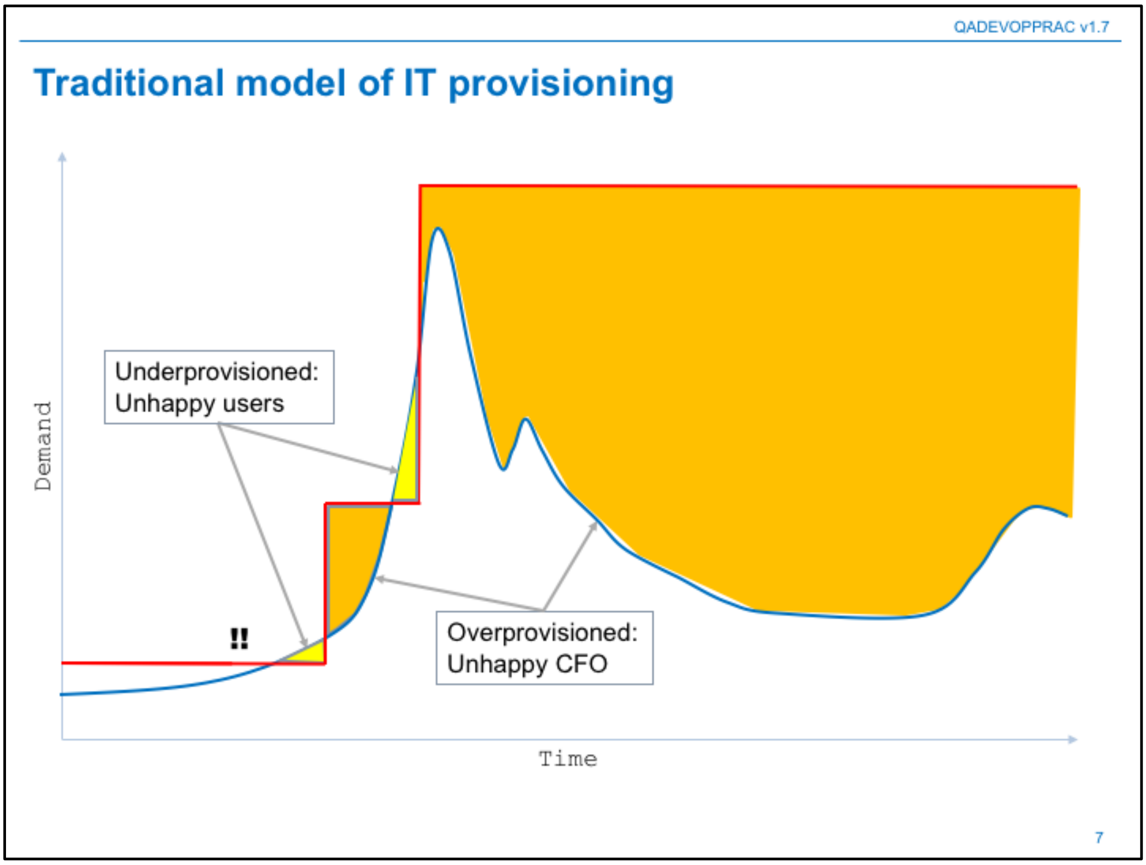
The National Institute of Standards and Technology's definition of cloud computing identifies "five essential characteristics":

1. ***On-demand self-service.*** A consumer can unilaterally provision computing capabilities, such as *server time and network storage, as needed automatically without requiring human interaction with each service provider*.
2. ***Broad network access.*** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., *mobile phones, tablets, laptops, and workstations*).
3. ***Resource pooling***. The provider's computing resources are pooled *to serve multiple consumers using a multi-tenant model*, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.
4. ***Rapid elasticity.*** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.
   To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.
5. ***Measured service.*** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service

(e.g., *storage, processing, bandwidth, and active user accounts*). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

What do these five characteristics mean to you?

- **From a developer standpoint**, cloud computing allows you to flexibly deliver integrated content, applications, and services to any device, anywhere, anytime, in a seamlessly scalable model, using and paying for only the resources you need, when you need them.
- **From an IT standpoint**, cloud computing allows organizations to free themselves from having to procure and allocate expensive hardware, software, and networking resources, or employ large teams to manage and support infrastructure.
- **From a business standpoint**, cloud computing enables providers to serve businesses of all sizes, especially SMBs.



QADEVOPPRAC v1.7

Traditional model of IT provisioning

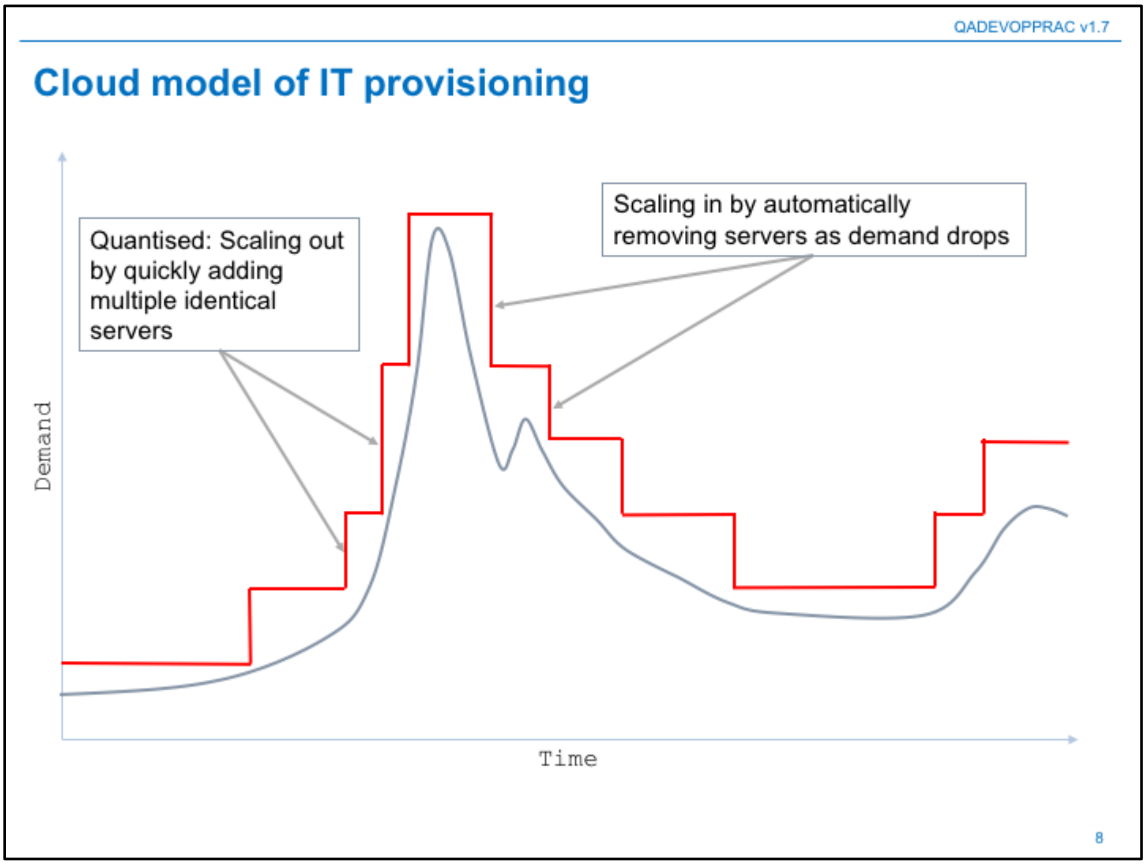
Underprovisioned: Unhappy users

Overprovisioned: Unhappy CFO

7

All introductions to cloud computing are obliged to contain two graphs like this slide and the next; together, they are the "hello, world" of cloud computing.

The **chief financial officer** (**CFO**) or chief financial and operating officer (CFOO) is a corporate officer primarily responsible for managing the financial risks of the corporation



QADEVOPPRAC v1.7

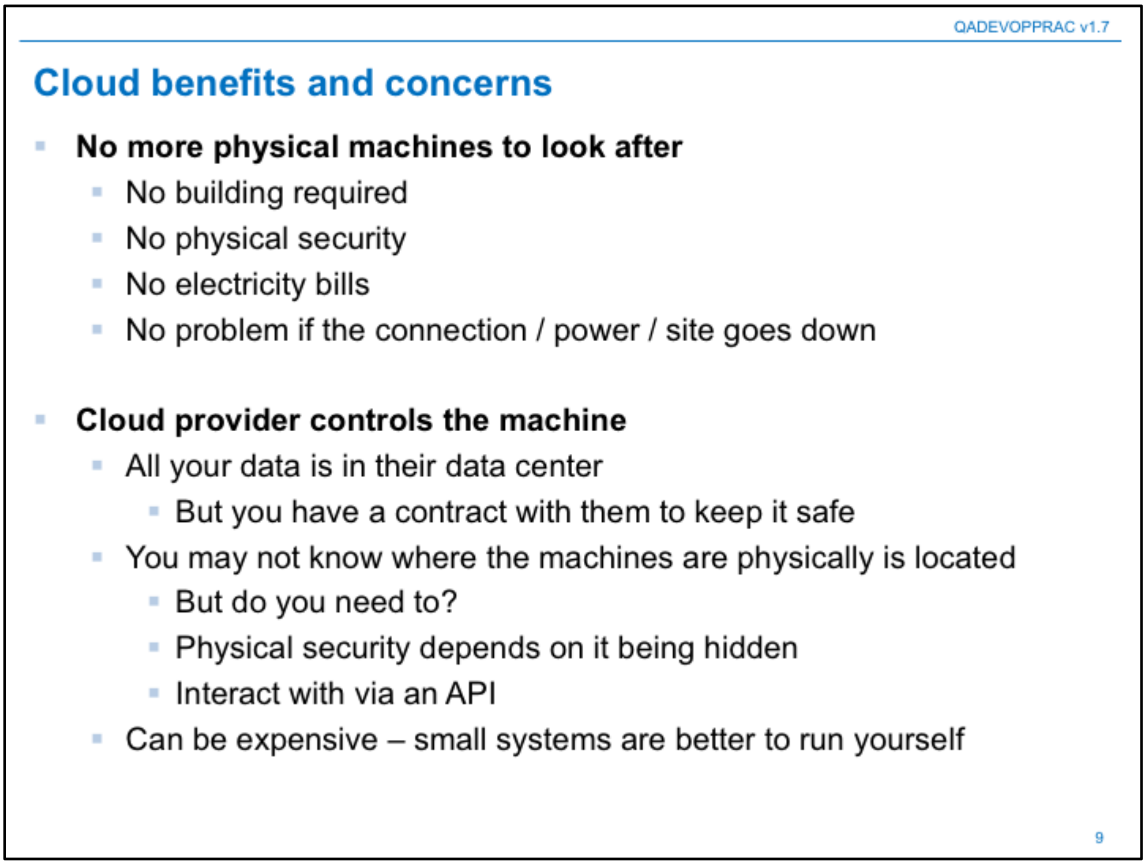
Cloud model of IT provisioning

Quantised: Scaling out by quickly adding multiple identical servers

Scaling in by automatically removing servers as demand drops

Notice that the graph implies scaling out rather than up: adding multiple identical servers, rather than increasing the power of existing servers. This is typical of cloud architectures.

Most importantly, notice that being able to scale out more quickly has allowed us to never be under-provisioned, and scaling in has prevented us from ending up massively over-provisioned.



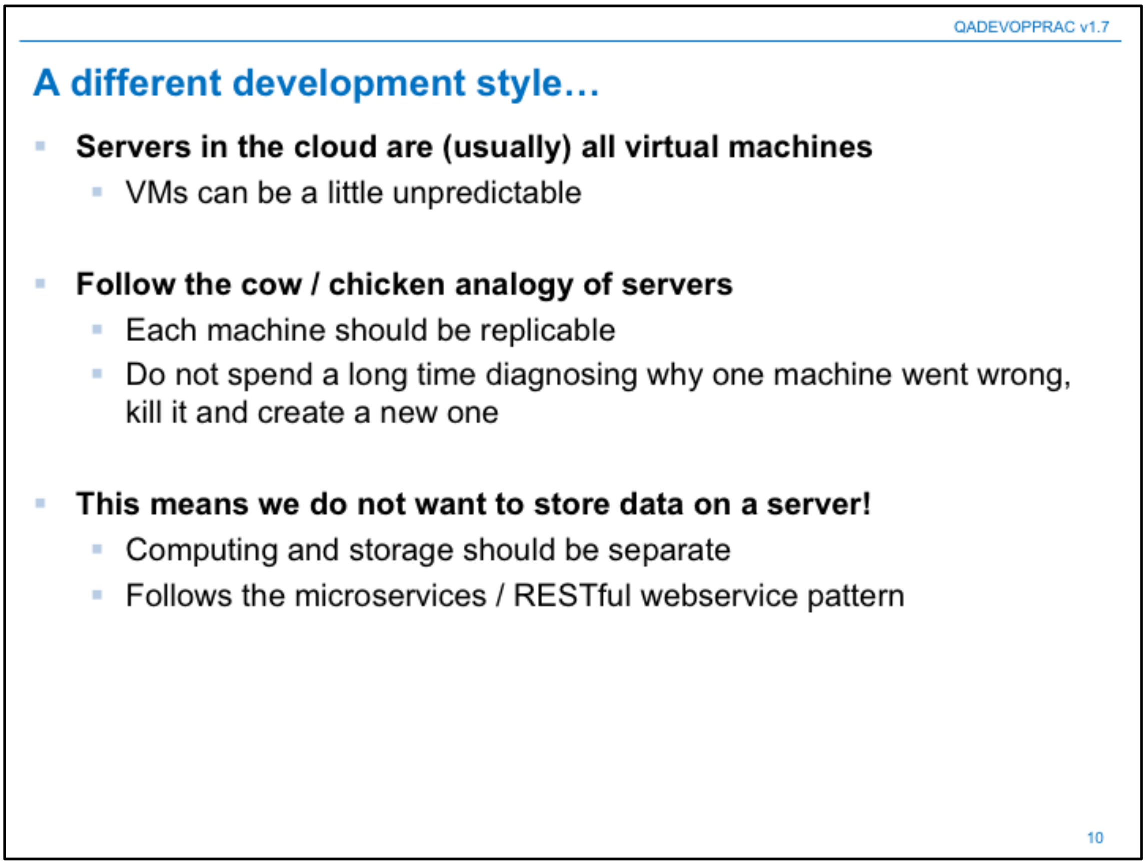
QADEVOPPRAC v1.7

Cloud benefits and concerns

- **No more physical machines to look after**
  - No building required
  - No physical security
  - No electricity bills
  - No problem if the connection / power / site goes down

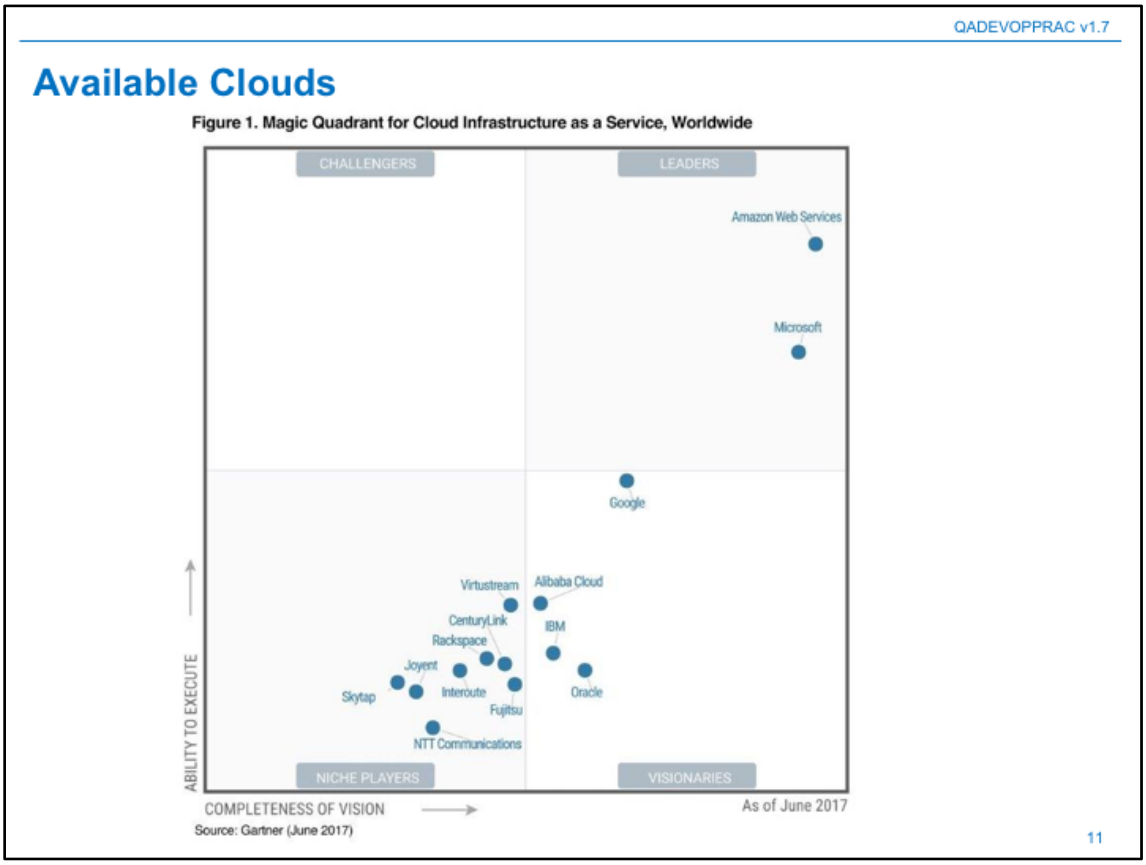
- **Cloud provider controls the machine**
  - All your data is in their data center
    - But you have a contract with them to keep it safe
  - You may not know where the machines are physically is located
    - But do you need to?
    - Physical security depends on it being hidden
    - Interact with via an API
  - Can be expensive – small systems are better to run yourself

9



A different development style…

- **Servers in the cloud are (usually) all virtual machines**
  - VMs can be a little unpredictable
- **Follow the cow / chicken analogy of servers**
  - Each machine should be replicable
  - Do not spend a long time diagnosing why one machine went wrong, kill it and create a new one
- **This means we do not want to store data on a server!**
  - Computing and storage should be separate
  - Follows the microservices / RESTful webservice pattern



QADEVOPPRAC v1.7

Available Clouds

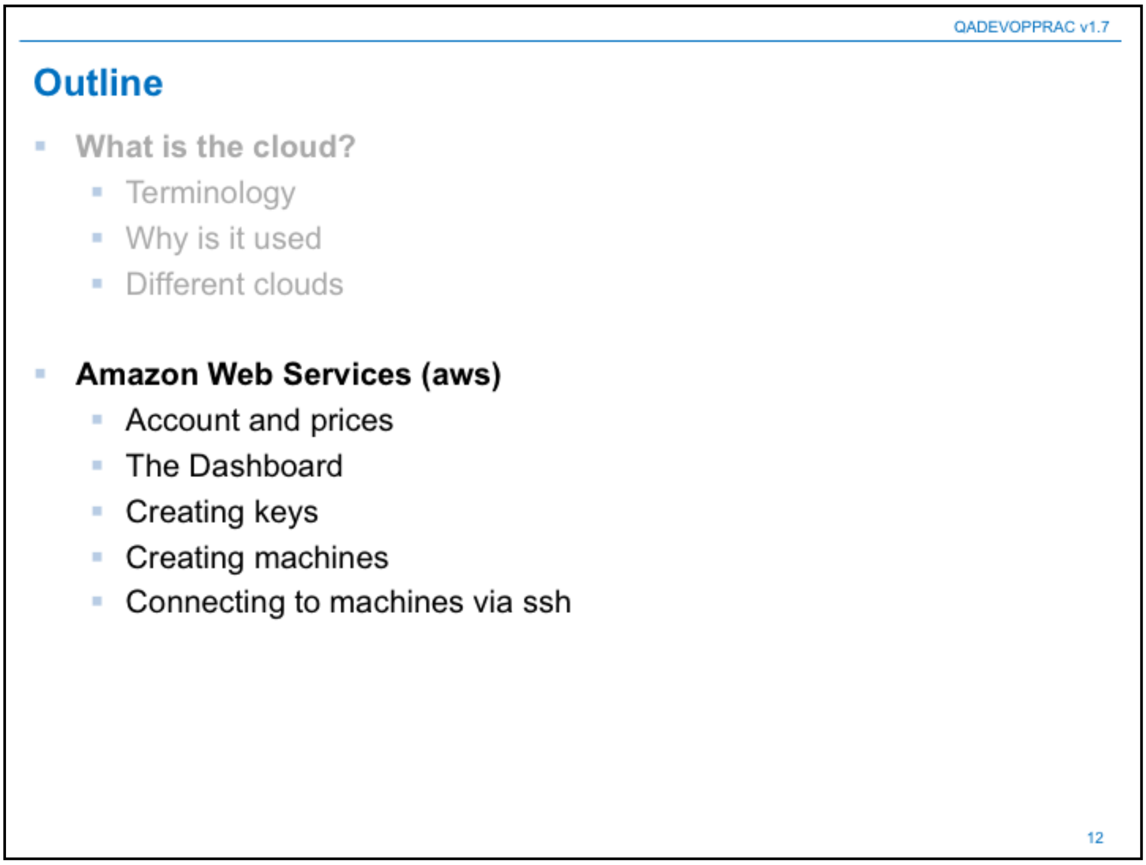
Figure 1. Magic Quadrant for Cloud Infrastructure as a Service, Worldwide

Source: Gartner (June 2017)

11

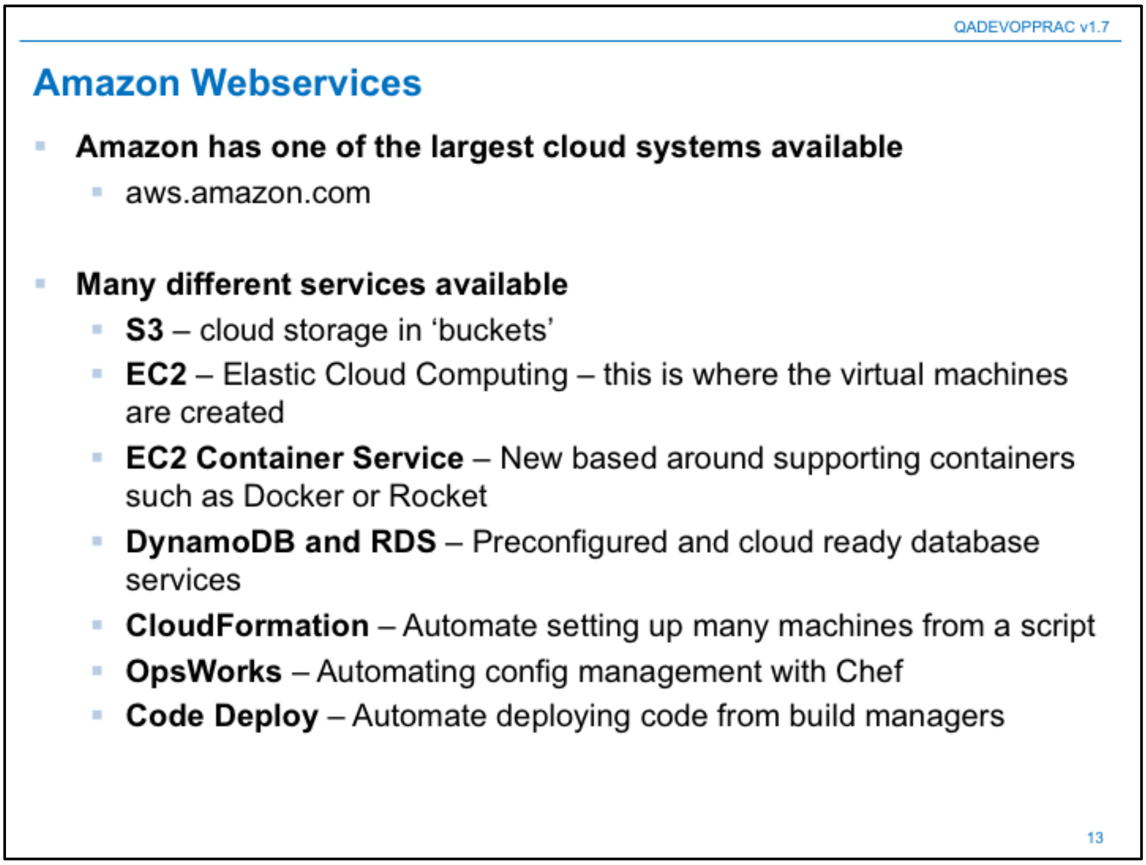
Image: https://www.gartner.com/doc/reprints?id=1-2G2O5FC&ct=150519&st=sb - updated 15/11/2017

The market for cloud IaaS is dominated by two leading service providers. Other service providers have responded by launching new offerings, but customers must carefully manage the risks of adopting less-mature offerings.



Outline

- What is the cloud?
  - Terminology
  - Why is it used
  - Different clouds
- **Amazon Web Services (aws)**
  - Account and prices
  - The Dashboard
  - Creating keys
  - Creating machines
  - Connecting to machines via ssh



QADEVOPPRAC v1.7

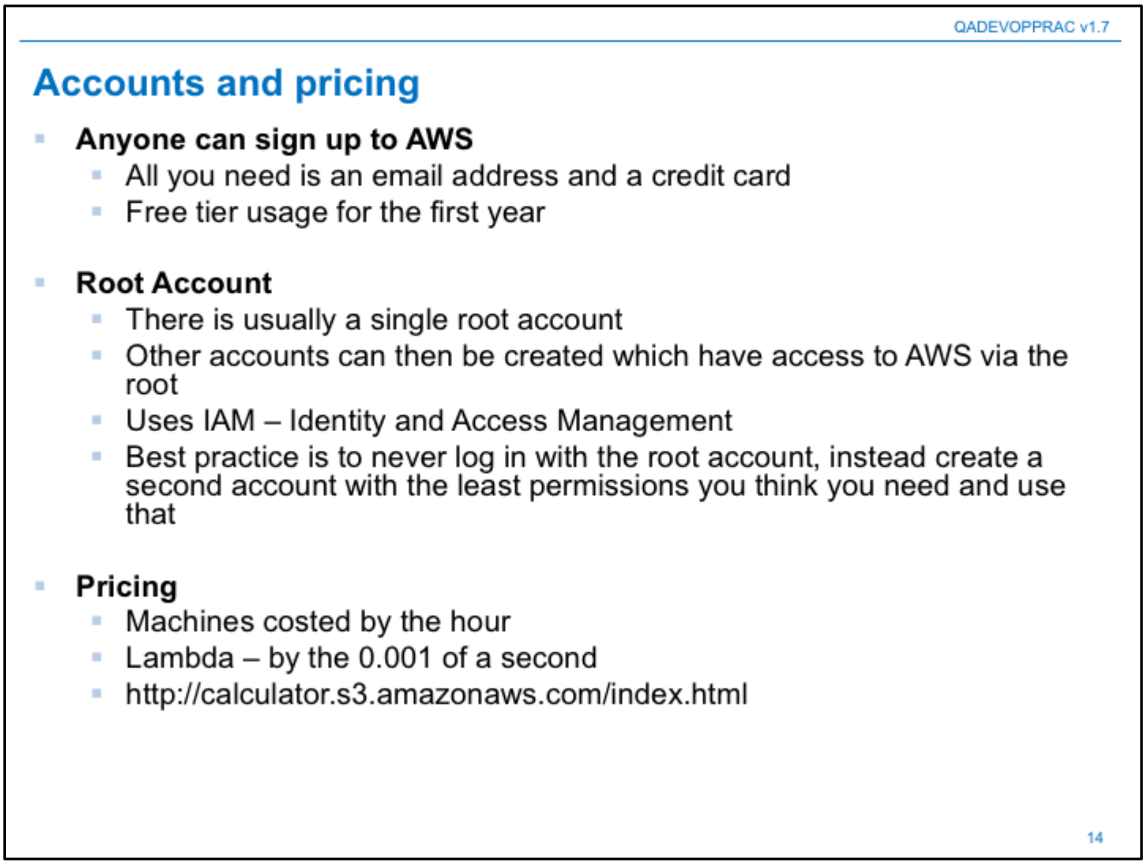
Amazon Webservices

- **Amazon has one of the largest cloud systems available**
  - aws.amazon.com
- **Many different services available**
  - **S3** – cloud storage in ‘buckets’
  - **EC2** – Elastic Cloud Computing – this is where the virtual machines are created
  - **EC2 Container Service** – New based around supporting containers such as Docker or Rocket
  - **DynamoDB and RDS** – Preconfigured and cloud ready database services
  - **CloudFormation** – Automate setting up many machines from a script
  - **OpsWorks** – Automating config management with Chef
  - **Code Deploy** – Automate deploying code from build managers

13

There are many different services available in AWS. This course isn’t aimed at showing you everything in AWS’s cloud services, but rather how we can build some of these systems ourselves. We will primarily be using EC2 (and possibly a little of S3) to bring up and kill servers as and when we require them.

QA offer a wide range of aws specific courses if you want to know more about any of these things!

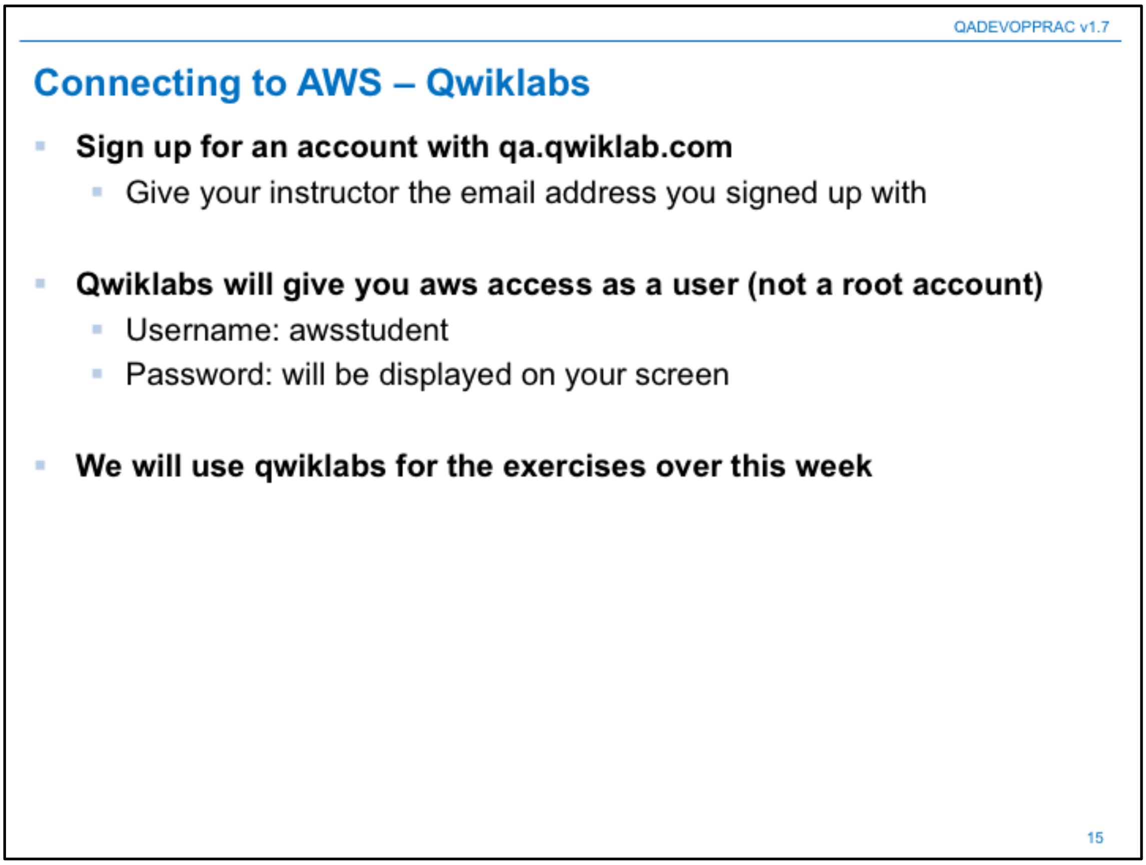


QADEVOPPRAC v1.7

Accounts and pricing

- **Anyone can sign up to AWS**
  - All you need is an email address and a credit card
  - Free tier usage for the first year
- **Root Account**
  - There is usually a single root account
  - Other accounts can then be created which have access to AWS via the root
  - Uses IAM – Identity and Access Management
  - Best practice is to never log in with the root account, instead create a second account with the least permissions you think you need and use that
- **Pricing**
  - Machines costed by the hour
  - Lambda – by the 0.001 of a second
  - http://calculator.s3.amazonaws.com/index.html

14

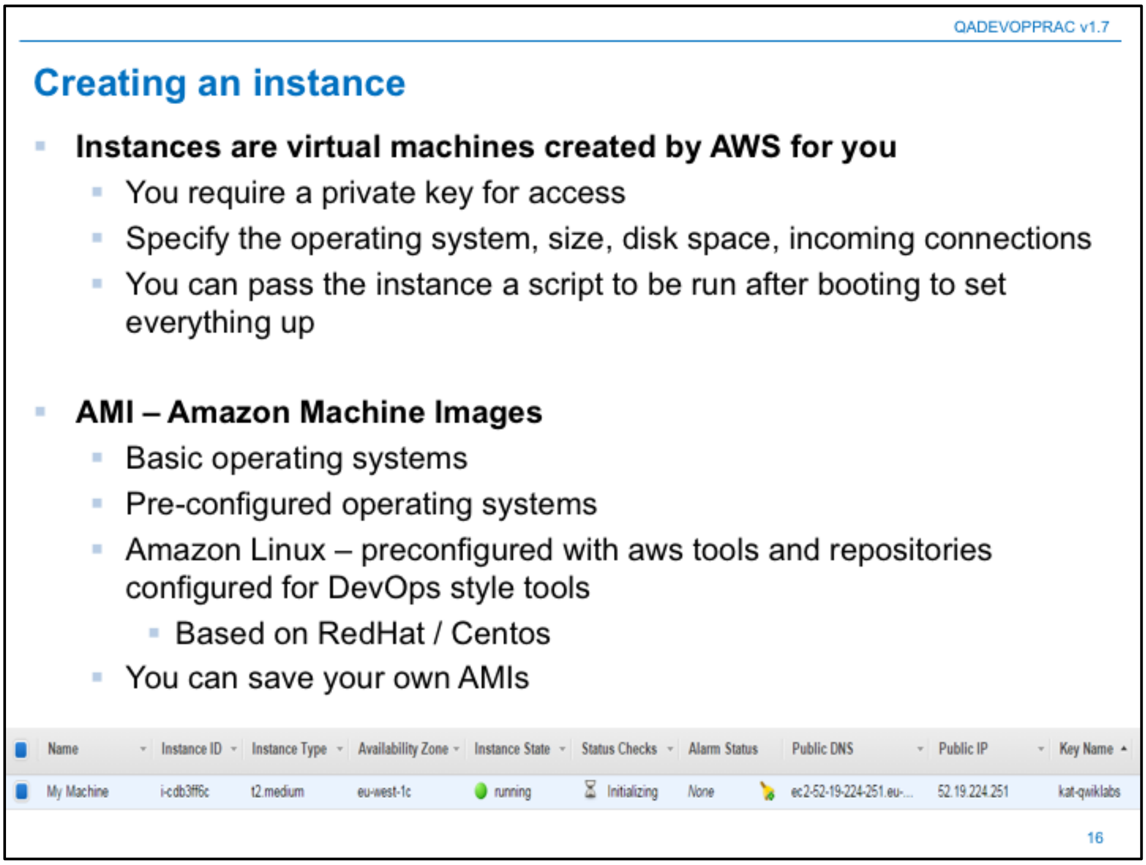


QADEVOPPRAC v1.7

Connecting to AWS – Qwiklabs

- **Sign up for an account with qa.qwiklab.com**
  - Give your instructor the email address you signed up with
- **Qwiklabs will give you aws access as a user (not a root account)**
  - Username: awsstudent
  - Password: will be displayed on your screen
- **We will use qwiklabs for the exercises over this week**

15



QADEVOPPRAC v1.7

Creating an instance

- **Instances are virtual machines created by AWS for you**
  - You require a private key for access
  - Specify the operating system, size, disk space, incoming connections
  - You can pass the instance a script to be run after booting to set everything up
- **AMI – Amazon Machine Images**
  - Basic operating systems
  - Pre-configured operating systems
  - Amazon Linux – preconfigured with aws tools and repositories configured for DevOps style tools
    - Based on RedHat / Centos
  - You can save your own AMIs

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name
My Machine	i-cdb3ff6c	t2.medium	eu-west-1c	● running	Initializing	None	ec2-52-19-224-251.eu-...	52.19.224.251	kat-qwiklabs

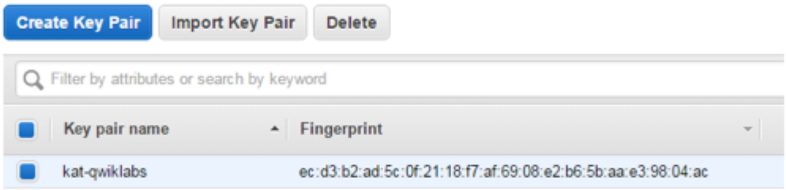
16

Your instance will have two IP addresses, public and private. Your private IP address will not change and can be accessed by any other machine on that subnet.

Your public IP address will be recreated and change every time you stop the machine. This will become relevant if you are running something on the machine that requires the public hostname and IP address to not change (Gitlab and Puppet being two main culprits)

Creating a key

- **When you create an instance you will be asked which key you want to use for access**
  - All AWS instances use a public / private key pair for access
  - Used directly for SSH access
  - Used to decrypt the password for RDP
- **Either create a new key when you are creating the instance**
- **Or use the link on the left hand side**
  - Keep your private key in a safe place
  - You will only get the chance to download it once!

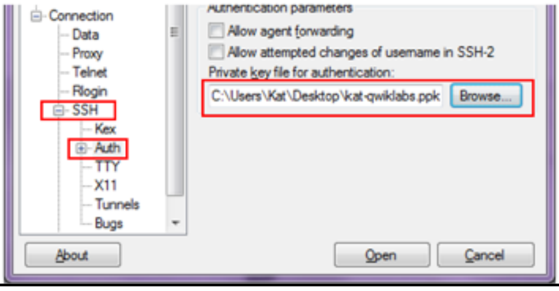


☑	Key pair name	Fingerprint
☑	kat-qwiklabs	ec:d3:b2:ad:5c:0f:21:18:f7:af:69:08:e2:b6:5b:aa:e3:98:04:ac

Connecting to your machine – Windows

- **Windows does not have a build in SSH terminal we can use**
  - Download PuTTY and PuTTyGen
  - We need to change our key from .pem to .ppk format
    - PuTTyGen will do this, open the .pem key, then choose “Save private key”

- **To connect to your instance**
  - Add the key
  - Username is defined by the ami (ubuntu / ec2-user etc.)
  - Use the public IP address of the server

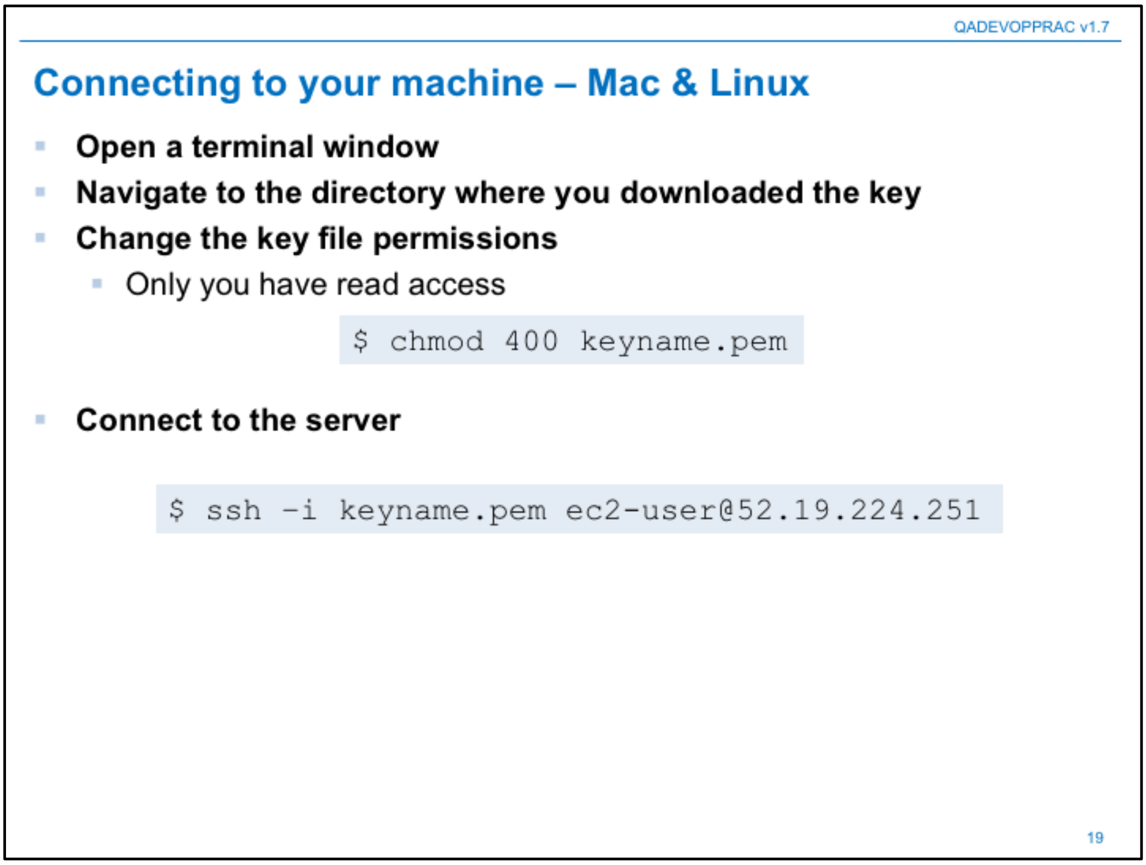


Authentication parameters

□ Allow agent forwarding

□ Allow attempted changes of username in SSH-2

Private key file for authentication: C:\Users\Kat\Desktop\kat-qwiklabs.ppk



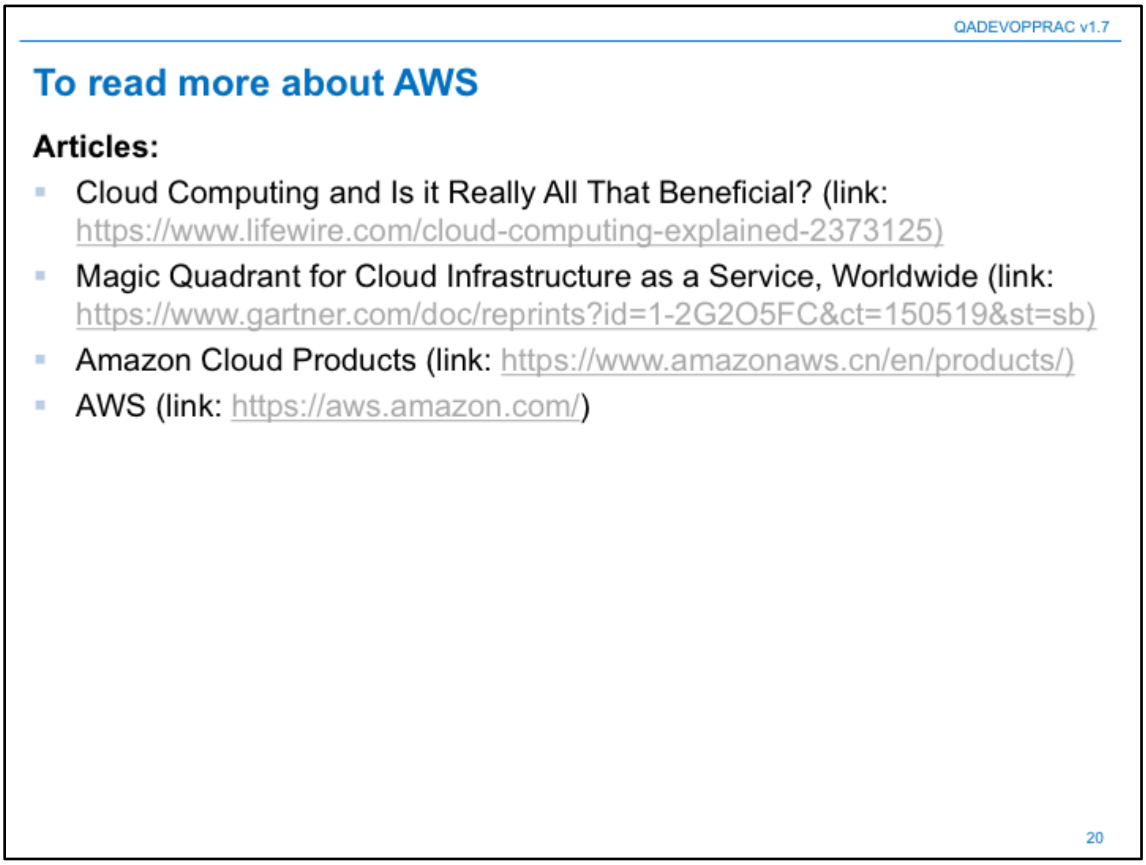
Connecting to your machine – Mac & Linux

- **Open a terminal window**
- **Navigate to the directory where you downloaded the key**
- **Change the key file permissions**
  - Only you have read access

```
$ chmod 400 keyname.pem
```

- **Connect to the server**

```
$ ssh -i keyname.pem ec2-user@52.19.224.251
```

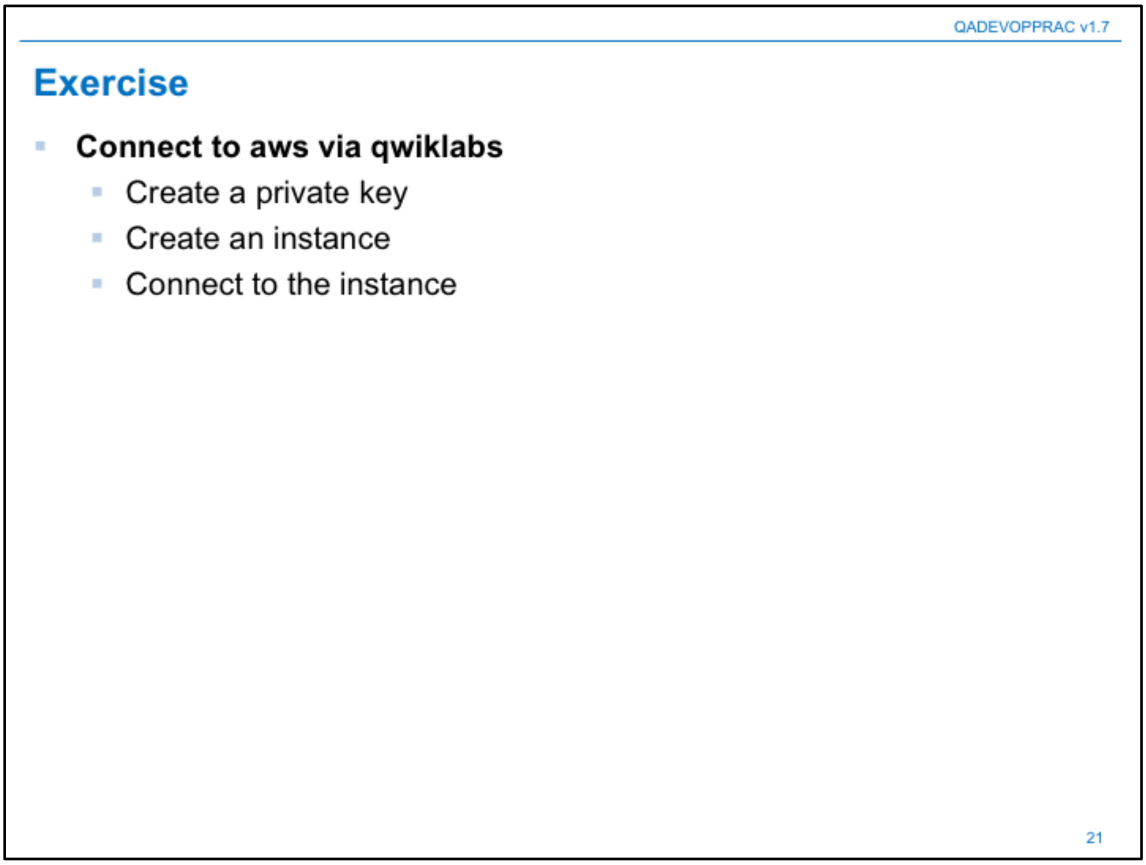


QADEVOPPRAC v1.7

To read more about AWS

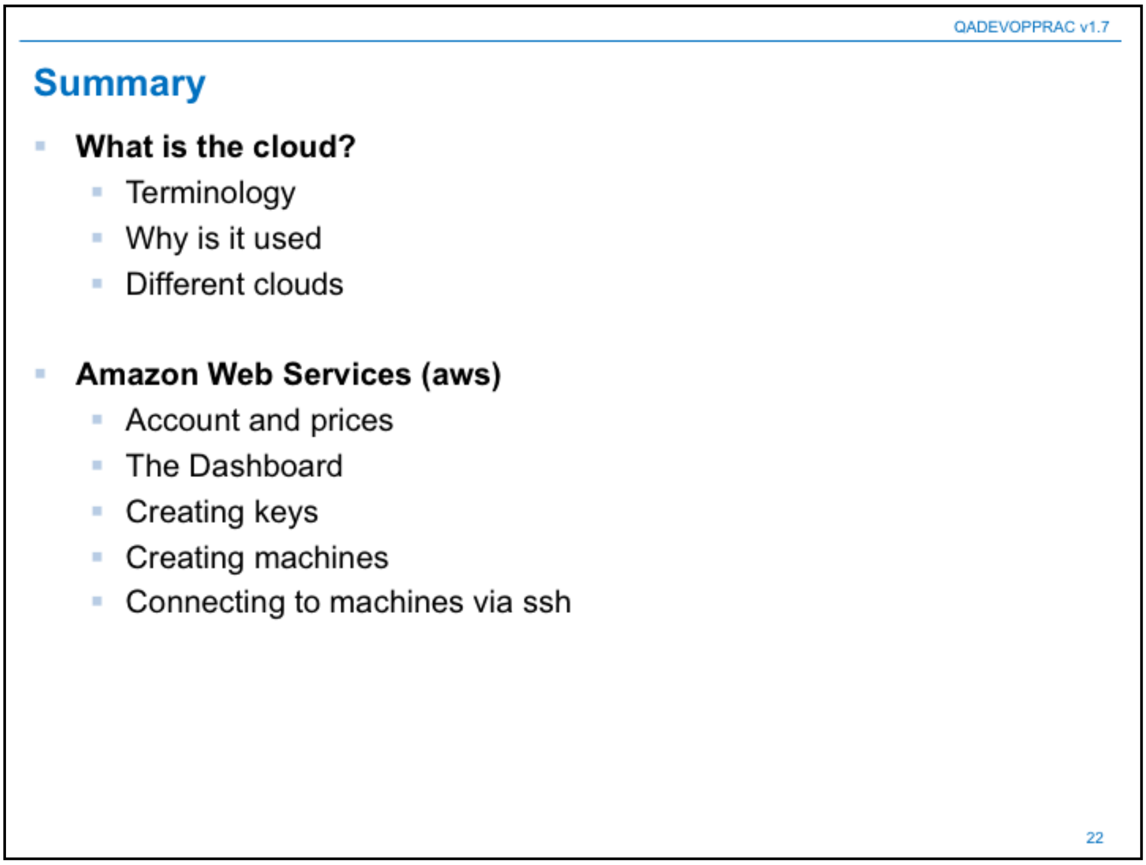
Articles:

- Cloud Computing and Is it Really All That Beneficial? (link: https://www.lifewire.com/cloud-computing-explained-2373125)
- Magic Quadrant for Cloud Infrastructure as a Service, Worldwide (link: https://www.gartner.com/doc/reprints?id=1-2G2O5FC&ct=150519&st=sb)
- Amazon Cloud Products (link: https://www.amazonaws.cn/en/products/)
- AWS (link: https://aws.amazon.com/)



Exercise

- **Connect to aws via qwiklabs**
  - Create a private key
  - Create an instance
  - Connect to the instance



Summary

- **What is the cloud?**
  - Terminology
  - Why is it used
  - Different clouds
- **Amazon Web Services (aws)**
  - Account and prices
  - The Dashboard
  - Creating keys
  - Creating machines
  - Connecting to machines via ssh