

Mer molntjänster

Dessutom: deployment!

Dagens agenda

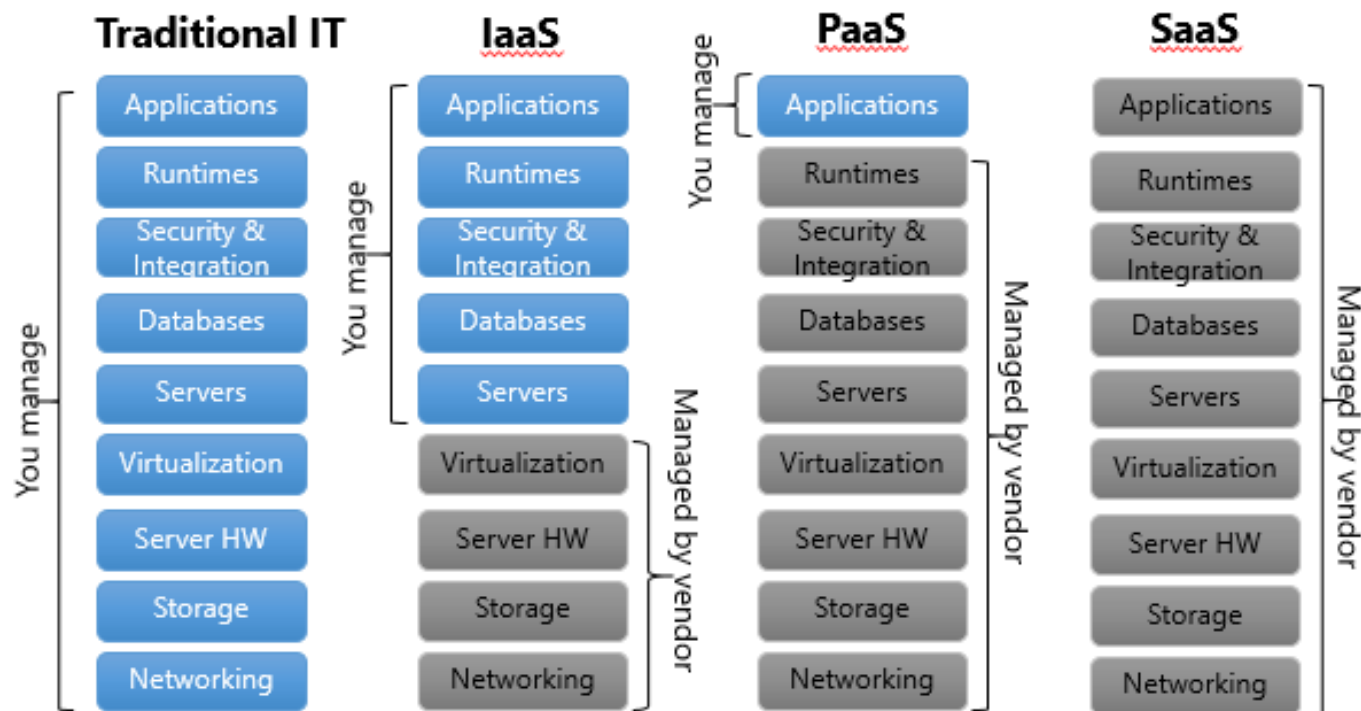
- Repetition från måndagens föreläsning
- Deployment – vad är det?
- Driftsättning i molnmiljö
 - Docker
 - Azure

Vad är Cloud Computing?

NIST (National Institute of Standards and Technologies):

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.

Driftmöjligheter

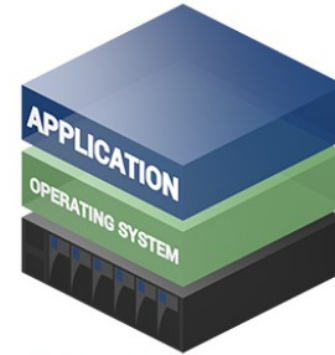


Olika typer av moln

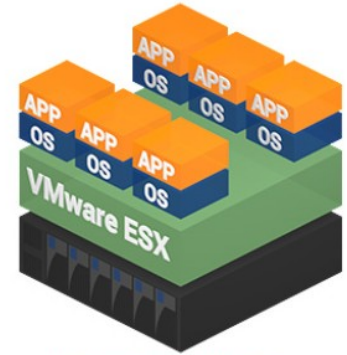
- Publika moln
- Privata moln
- Hybridmoln

Virtualisering

- Abstraktion av en dators fysiska resurser.
 - Flera virtuella maskiner kan dela på samma hårdvara
 - En **hypervisor** eller *Virtual Machine Manager*, **VMM**, ser till att de virtuella maskinerna får tillgång till hårdvaran
 - De virtuella datorernas operativsystem och applikationer ser inte någon skillnad på deras virtuella miljö jämfört med en dedikerad fysisk dator.



Classic Server Installation



Virtualized server Installation

Containers

- En lite lättviktigare approach till virtualisering
 - Flera containers delar på en gemensam OS-kärna
 - Enklare att sätta upp och ta ner – snabbare reaktionstid
- Deklarativ virtualisering – beskriv VAD du behöver, inte HUR det ska realiseras

Tekniska vinster

- Specialisering
- Skalfördelar
- Elasticitet
- Redundans
- DevOps

Ekonomiska vinster

- Förutsägbart
- Skalbart
- Låga uppstartskostnader

Docker

Vad är Docker?

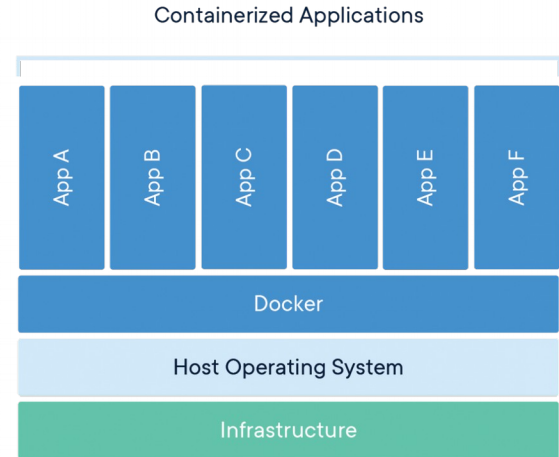


Den vanligaste varianten av container-miljö

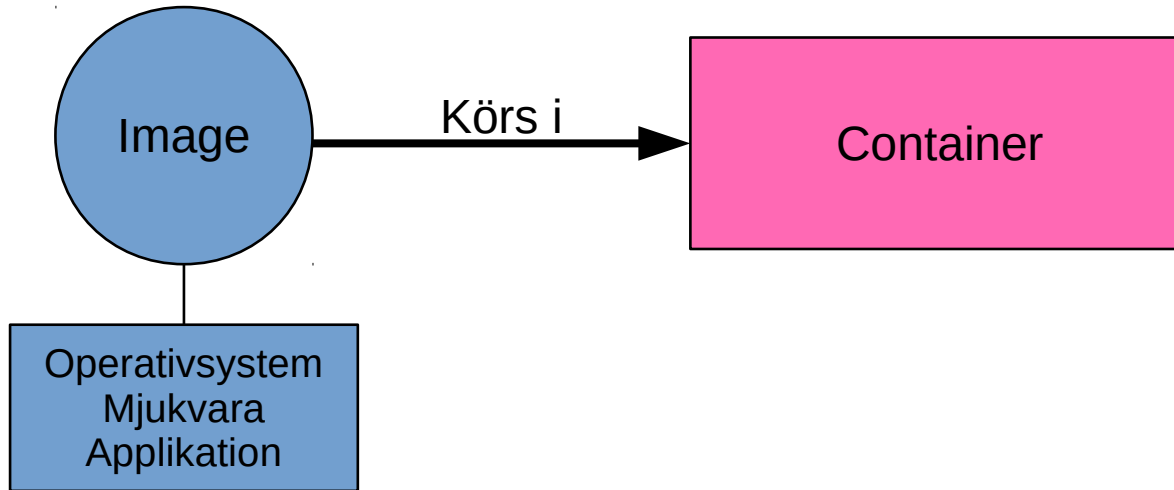
- Baseras på Linux
- Körbar både på server och utvecklingsmaskin
- Plattformsoberoende och välstödd – enkelt att flytta mellan molnleverantörer
- Kan provisioneras från exempelvis ett git-repository – utmärkt för DevOps

Varför containers?

- Deklarativ infrastruktur – beskriv VAD du vill ha, inte HUR det ska realiseras
- Immutabel infrastruktur – inget ändras i drift, bara i releaser
- Enkelt att flytta applikationer
- Enkelt att låta applikationer växa vid behov. ← Bra skit!



Image? Container?



Vad är en image?

En “komplett” avbildning:

- Operativsystem (minus kärnan)
- All stödjande mjukvara
- Vår applikation

Hur körs en container?

Containern är aktiv så länge imagen har något att köra

- Om imagen kör exempelvis en server, är containern aktiv tills den stängs av
- Om imagen kör exempelvis en installation, avslutas containern när installationen är färdig

Dockerfile – recept för images

- Images skapas med hjälp av Dockerfiles
- En Dockerfile är ett “recept” på en körmiljö – ungefär som composer.json i PHP
- Images i Docker kan bygga på varandra, så vi kan “ärva” egenskaper från andra images

Dockerfile – recept för images

FROM php:7-alpine

Maintainer Johan Holmberg <johan.holmberg@mau.se>

MongoDB

```
RUN apk update && apk add autoconf openssl-dev g++ make && \
    pecl install mongodb && \
    docker-php-ext-enable mongodb && \
    apk del --purge autoconf openssl-dev g++ make
```

LDAP

```
RUN apk add --no-cache --virtual .build-deps $PHPIZE_DEPS icu-dev openssl-dev \
    apk --update add php7-ldap && \
    docker-php-ext-install ldap && \
    docker-php-ext-enable ldap && \
    apk del .build-deps
```

Composer

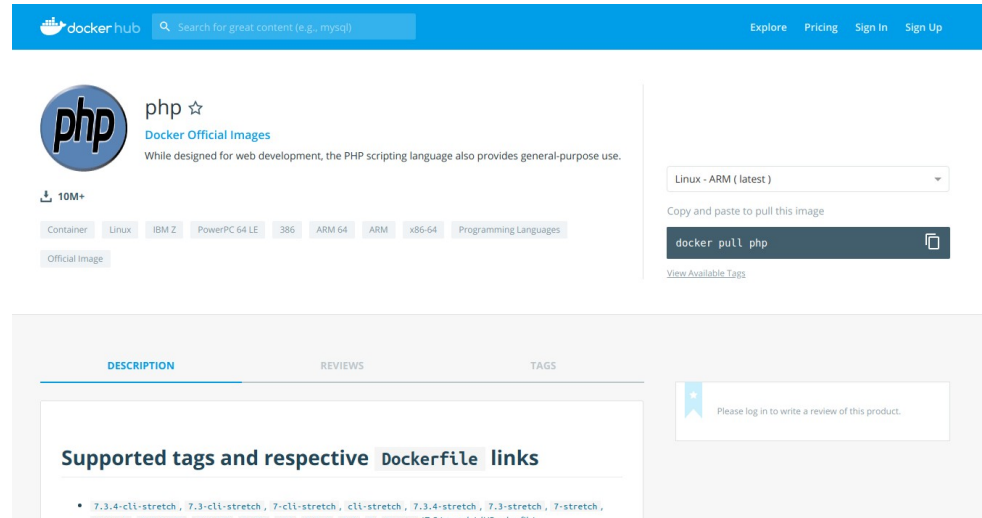
```
RUN apk --no-cache add --virtual .composer-rundeps git subversion openssh mercurial tini bash patch make zip unzip
```

```
RUN echo "memory_limit=-1" > "$PHP_INI_DIR/conf.d/memory-limit.ini" \
&& echo "date.timezone=${PHP TIMEZONE:-UTC}" > "$PHP_INI_DIR/conf.d/date_timezone.ini"
```

```
RUN apk add --no-cache --virtual .build-deps zlib-dev libzip-dev \  
&& docker-php-ext-configure zip --with-libzip \  
&& docker-php-ext-install -i$(getconf NPROCESSORS_ONLN) zip \  
#>
```

Docker Hub

Här hittar vi färdiga images att köra eller bygga vidare på



The screenshot shows the Docker Hub interface for the 'php' image. At the top is a blue navigation bar with the Docker Hub logo, a search bar, and links for 'Explore', 'Pricing', 'Sign In', and 'Sign Up'. Below the navigation bar, the 'php' image is featured with its logo, a star, and the text 'Docker Official Images'. A description states: 'While designed for web development, the PHP scripting language also provides general-purpose use.' To the left of the description, it shows '10M+' downloads and a list of supported architectures: Container, Linux, IBM Z, PowerPC 64 LE, 386, ARM 64, ARM, x86-64, and Programming Languages. Below this is a button labeled 'Official Image'. To the right of the description, there is a dropdown menu showing 'Linux - ARM (latest)', a text box for copying the image name, and a button labeled 'docker pull php'. Below the button is a link to 'View Available Tags'. At the bottom of the page, there is a section titled 'Supported tags and respective Dockerfile links' with a list of tags: '7.3.4-cli-stretch', '7.3-cli-stretch', '7-cli-stretch', 'cli-stretch', '7.3.4-stretch', '7.3-stretch', '7-stretch', 'stretch', '7.3.4-stretch', '7.3-stretch', '7-stretch', 'stretch', '7.3.4-stretch', '7.3-stretch', '7-stretch', 'stretch'. To the right of this section is a box with a plus icon and the text 'Please log in to write a review of this product.'

Volumes

- Gemensamma lagringsytor för våra containers
- Är i sin enklaste form endast en katalog som är mappad mot en plats i containern

Demo!

Johan visar hur man bygger en image och kör den

Docker-compose.yml

- Anvisningar för HUR Docker ska köras
- Kan hantera flera olika containers:
 - En applikation kan bestå av många containers
 - Bra för exempelvis microservices

Docker-compose.yml

```
version: '3'
```

```
services:
```

```
  web:
```

```
    build: .
```

```
    restart: on-failure
```

```
    command: php -S 0.0.0.0:8080 -t app/public
```

```
    ports:
```

```
      - 8080:8080
```

```
      - 443:443
```

```
    volumes:
```

```
      - ./app
```

```
  composer:
```

```
    image: composer/composer:php7
```

```
    ports:
```

```
      - "8080:80"
```

```
    command: install
```

```
    restart: "no"
```

```
    volumes:
```

```
      - ./app
```

Deployment – vad är det?

Deployment



Vad är deployment?

Deployment = driftsättning

- Vi testar mjukvaran
- Vi säkerställer att allt fungerar
- Vi sätter koden i produktion
- Vi gör det aldrig på en fredag

Delivery



Vad är delivery?

Delivery = leverans

- Driftsättning av kod
- Publicering av dokumentation
- Publicering av testresultat

Integration



Integration

- Få alla delar av mjukvaran att fungera ihop
 - I ett mindre PHP-projekt är detta sällan ett problem
 - I större projekt med många utvecklingsteam kommer problem att uppstå
 - Använd integrationstester!
 - Testa ordentligt innan driftsättning!

What's the fuzz about?

Driftsättning är läskigt!

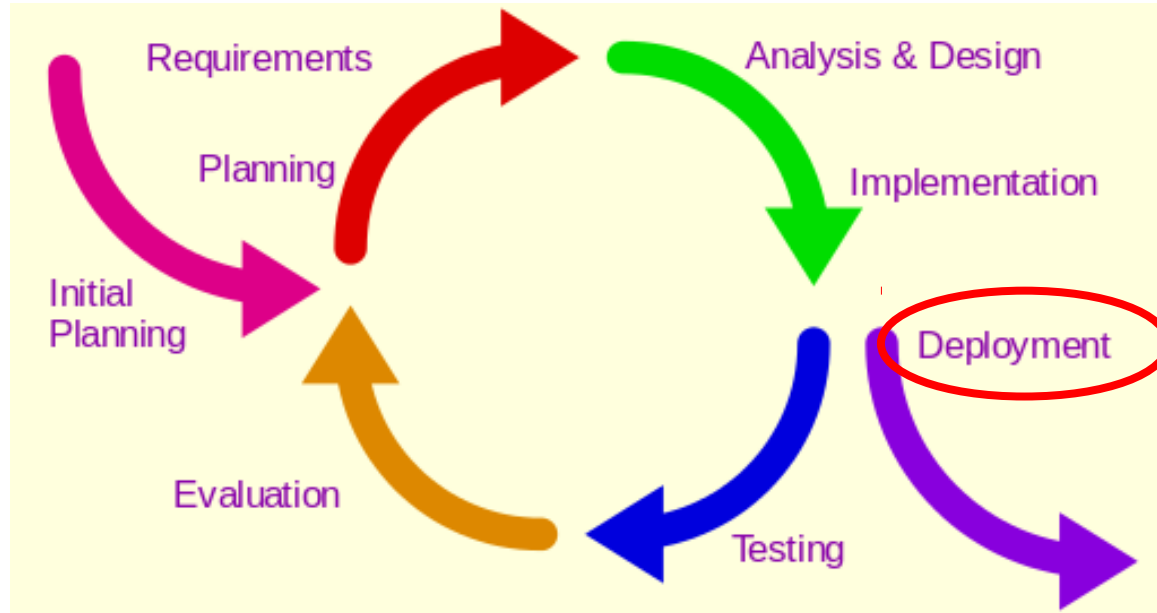
- Den naiva lösningen är manuell (som med FTP)
- Många steg → sårbart
- Lösningen är beroende av applikationen

Kan vi förenkla det här?

What's the fuzz about?

Yes! Med automatiserad deployment!

Under utvecklingen



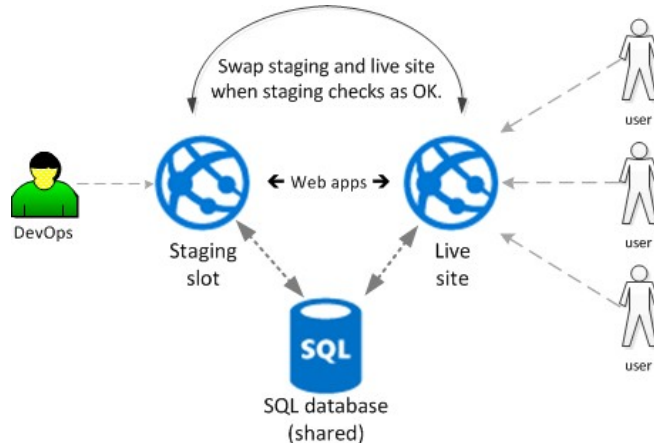
Wikipedia, Iterative and incremental development -
https://en.wikipedia.org/wiki/Iterative_and_incremental_development

Deployment i två steg

- Staging
 - Här körs utvecklingsversionen av en mjukvara (typ develop)
- Produktion
 - Här körs mjukvaran på riktigt (typ master)

Driftsättning med slots

Använder två “applikationsplatser”, som växlar mellan att vara staging och produktion:



Redgate Software, *Deploying an Entity Framework Database into Production.*

<https://www.red-gate.com/simple-talk/dotnet/net-framework/deploying-an-entity-framework-database-into-production/>

CI, CD och CI/CD

Continuous Integration (CI)

- Automatiserade tester vid varje push till develop
- Ser till att koden alltid är vältestad och fungerande

Continuous Delivery (CD)

- Automatiserade tester och byggen vid varje push till master
- Ser till att applikationen alltid är levererbar

Continuous Deployment (CI/CD)

- Automatiserad leverans efter CI och CD
- Ser till att applikationen alltid kör den senaste versionen av koden

Wikipedia, *Continuous integration*. https://en.wikipedia.org/wiki/Continuous_integration

Wikipedia, *Continuous delivery*. https://en.wikipedia.org/wiki/Continuous_delivery



ar revererbar

D)

Cl o

kör o

Con
ntine



Wikipedia, C
Wikipedia, C

Driftsättning i molnmiljö

Vad är deployment?

Deployment = driftsättning

- Vi testar mjukvaran
- Vi säkerställer att allt fungerar
- Vi sätter koden i produktion
- Vi gör det aldrig på en fredag

Azure

Översikt

- Genomgång av Azure
- Deployment till Azure