

Docker indledning.

Velkommen til dette opgavesæt, hvor du vil lære noget om, og med, Docker ved at udføre nogle opgaver. Nu spørge du nok: "Hvad i h... er Docker?" Svaret på dette spørgsmål er: Docker er en mulighed for at udvikle sine apps, og køre apps, og services, i et lukke kontrolleret miljø. Dette betyder at hvis du f.eks. har specifikke krav til netop DIN webside, database, applikation eller andet software, kan du sende hele opsætning, og software konfiguration, med, når du vil distribuere dit software. Der er mere og mere software der også udkommer som Docker containere. Fordelen er jo at du allerede har konfigureret dit software, eller i hvert fald har lavet "byggevejledninger" til dit software, så uanset om software bliver installeret på en gigantisk server med Terrabyte af RAM og dusinvis af processorer, eller på en Raspberry Pi, vil det køre på samme måde. Dette kan gøres med en Docker container. På den store server vil det måske køre lidt hurtigere, og du kan have flere af samme container kørende.

Vigtig information til opgaverne:

For at kunne få mest mulig ud af dette oplæg er det en meget stor fordel at have kendskab til Linux, alle output eksempler er udført på en Ubuntu Linux 24.04. Man får mest ud af at køre med virtuelle maskiner. Der skal bruges 2 stk. maskiner:

1. En, der er navngivet registry
2. En, der er navngivet worker eller lign. I disse eksempler kalder jeg den for worker. Det er på denne maskiner du udfører hovedparten af eksemplerne.
3. Du skal kunne IP-adresserne på maskinerne, og de skal selvfølgelig være forskellige.
4. Begge maskiner skal kunne kommunikere med hinanden, og med internettet.
5. Du skal have adgang til maskinerne fra den maskine hvorpå du udfører opgaverne. Denne maskine skal du bruge til at kommunikere med SSH ind til de to oprettede (virtuelle) maskiner, Din SSH maskine skal også have en form for webbrowser installeret f.eks. firefox, chrome, edge el.lign. Hvis din webbrowser sidder på en anden PC skal denne selvfølgelig også kunne kontakte dine VM maskiner.
6. Du skal have sudo adgang på begge VM maskiner.

Begge maskiner behøver ikke være super kraftig. Her er en minimumsanbefaling (De kan køre med mindre specs, men så bliver de måske lidt langsom):

1. 4 Gb Ram (4096 Mbyte).
2. 30 Gb Harddisk
3. 2 virtuelle processorer, med 2 kerner hver.
4. evt. CD rom for installation af Ubuntu 24.04

Man skal i opgaverne, også bruge specifikke kommandoer der har en IP adresse i kommandoen, f.eks. til din registry server. Her vil der f.eks. stå `http://ip_på_Registry`. Hvis dette er tilfældet kan det være en god ide at have en notesblok åben. Indsætte kommandoerne her, og så tilpasse de IP adresser man skal bruge. For derefter at indsætte kommandoerne i den aktuelle servers SSH vindue. Man kan også gemme alle kommandoer, tilpasset lige akkurat dit system, da et par af dem vil kunne bruges igen senere.

Hvordan virker Docker så? Jo bare læs videre.

Du har nok hørt om virtuelle maskiner (VM), som er små maskiner der kører i et virtuelt system på nogle større fysiske maskiner. Lad os sammenligne docker og VM'er lidt:



Virtuelle maskiner er som en række parcelhuse der har egen forsyning af vand, varme, kloak, el osv. Fra hosten (Byen).



Docker er mere som et højhus med lejligheder (Containere), der deles om samme vandledning, varme, kloak, el osv. Docker bruger f.eks. din hosts kernel (Hosten er normalt linux), så derfor findes der ikke en docker container, hvori der kører Windows. Nok også fordi der ikke findes docker containere der har en skærmenhed (et virtuelt skærmkort, som der findes i Hypervisorer). Windows bruger jo grafisk bruger interface. Derfor er følgende sætning god at huske på: **En VM kan IKKE køre på Docker, men Docker kan køre på en VM.** Docker er IKKE en Hypervisor.

Jamen jeg har hørt om Kubernetes, Er det ikke bedre end Docker?

Ok.. Jeg vil lige skrive lidt omkring Kubernetes. Du har måske nok hørt om Kubernetes (det er et software system der er opfundet af Google) Kubernetes (bliver også kaldet K8s = K+8 bogstaver+s) Og flere spørger måske: Er det ikke bedre at bruge K8s? Hvertil jeg svarer: K8s er det man kalder en Orchestrator. Den bruges til at STYRE Docker containere (Docker bruger til udvikling, og K8s til at køre containerne). K8s kan bruges til bl.a. dynamisk at overvåge belastningen på f.eks. en webside der er load balanced. Hvis der kommer ekstra meget belastning af en eller anden grund, kan k8s automatisk starte flere containere, som så aflaster de andre. Når så belastningen falder igen, stopper den nogle containere igen, og sparer derved ressourcer. Det skal dog nævnes at K8s er lavet til at køre med cluster opsætning, og frarådes til computercentre med mindre en 3-4 FYSISKE servere. Jeg vil ikke komme nærmere ind på K8s her, da der ikke er de krævede ressourcer til rådighed. Hvis man er mere interesseret kan man jo spørge Google (søgemaskinen, ikke firmaet :)) eller se på denne side: <https://kubernetes.io/>

Lad os komme tilbage til Docker igen. En container er normalt en "kasse" der indeholder én Applikation. Kassen indeholder alt der skal til for at netop denne applikation kan køre. Og derved er den nemt at opdatere. Man opdaterer kassen, og sender hele den opdaterede kasse ud. Ikke noget med at nogle har mere RAM, mindre cpu eller lign. Kassen er den samme du har lavet ALLE steder den kører, og vil køre på samme måde på alle systemer!

Bedst practice er at køre én applikation/service per Container (f.eks. Webserver, og PHP i hver sin container) men man KAN godt lægge flere applikationer/services i samme container (Det prøver vi om lidt i opgave 1..).

Hvis man kører med en VM, vil der normalt kunne køre det, man kan køre på en fysisk maskine. Der kan køre flere applikationer på samme maskine (f.eks. FTP, Webserver, Samba (SMB) osv). I en Docker container er det normalt kun én Applikation pr. container der kører (Separat container til Ftp, Web, Samba mm). Så med Docker kører der multiple containere på en maskine (kan være en VM, Fysisk maskine, eller faktisk også en Windows server! Da man kan få Docker til windows. Det vil jeg ikke komme nærmere ind på. Men der kan findes mere om det her: <https://www.docker.com/products/docker-desktop/>)

Lad os lige se lidt på forskellene mellem en hypervisor, og docker:

	Hypervisor	Docker
Operativ system Support	De fleste OS'er (Selvste Host OS er "Usynlig" for Client OS)	KUN Linux! Bruger HOST maskinens Kernal.
Boot Tid	Afhængig af OS, ressourcer tildelt osv., men op til 1 Minut, eller mere. Lige så lang tid det tager at starte en normal computer, med tilsvarende ressourcer.	Kun Sekunder, efter image er bygget/fremstillet.
Sikkerhed	Adskilt fra Host OS	Afhængig af Linux Kernal, men lukket inde i sin egen container.
Ressourceforbrug	Afhængig af hvad der er tildelt, men der skal f.eks. også afsættes plads på harddisk og i memmory til OS'et (Windows, Linux osv.)	Docker Containere tager ikke mange ressourcer. Kun det nødvendige til at køre lige netop den service den er bygget til.
Hvad kan der køre på systemet?	Flere forskellige OS'er, hver med flere Applikationer.	Flere Applikationer af gangen

På de næste sider vil jeg lige gennemgå de forskellige forudsætninger for at kunne udføre de opgaver der kan udføres lige om lidt.

Forudsætninger for at kunne køre opgaverne:

Det er en god ide at have et program der kan bruges til SSH kommunikation. Dette kan faktisk også bare gøres fra en Windows Powershell prompt, eller Command prompt i Windows 10 eller 11. Der skrives bare **ssh ip-adresse**. Jeg anbefaler dog at bruge et program som putty (<https://www.putty.org/>).

Materialet der bruges (Biblioteker, config filer osv.) kan hentes på hjemmesiden: <https://docker.brianhansen.dk>, og for at kunne overføre disse data til vores docker arbejdscomputer skal man bruge et program, som WinSCP (Står for Windows Secure Copy) Den kan hentes her: <https://winscp.net/eng/download.php>

VIGTIGT!: Filer af typen ".yml" SKAL have alle indryk med, ellers virker det ikke. Dette kan være et problem ved kopiering af tekst fra en PDF fil. Alternativ Ligger alle filer der skal bruges Sammen med det materiale man kan hente til hver af opgaverne, eller i pakken med det hele.

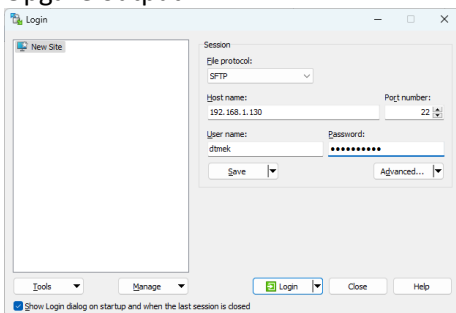
Kort vejledning i WinSCP:

Programmet hentes, og installeres. Jeg regner med at i har prøvet at installere et program på windows før, og jeg bruger interfacet der hedder commander når det skal vælges under installation.

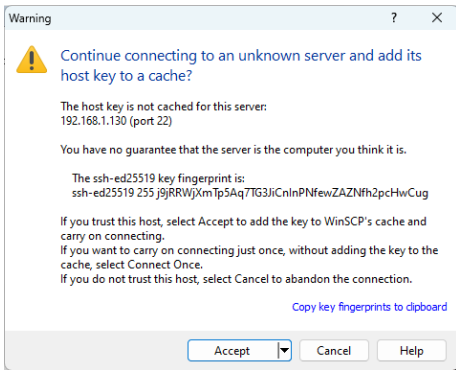
Start programmet.

Først skal vi vælge "File Protocol (SFTP)" så skriver vi IP på den pc du vil overføre noget til, derefter indtastes brugernavn og password, port vil i dette tilfælde altid være 22:

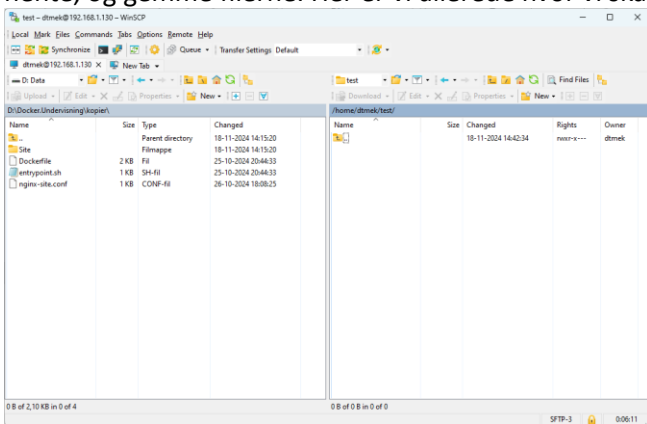
Opgave output:



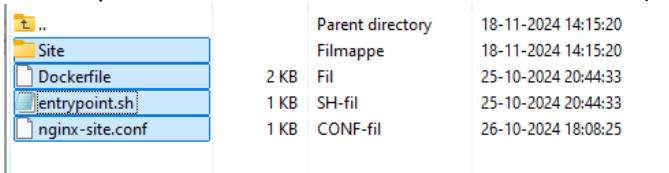
Der vil nok komme denne besked frem (IP og fingerprint, er selvfølgelig ikke det samme), Her svarer man bare "Accept":



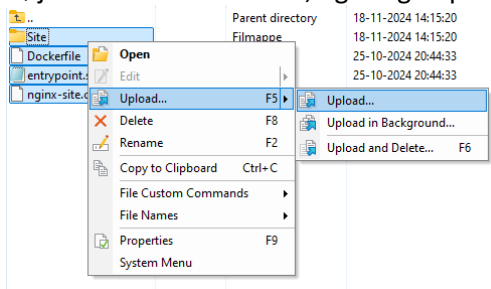
Som man kan se er billedet meget lig en stifinder med 2 vinduer. Venstre vindue er lokal harddisk, højre vindue er remote maskines harddisk. Man kan bladre sig frem til de biblioteker, hvor man henholdsvis vil hente, og gemme filerne. Her er vi allerede hvor vi skal være:



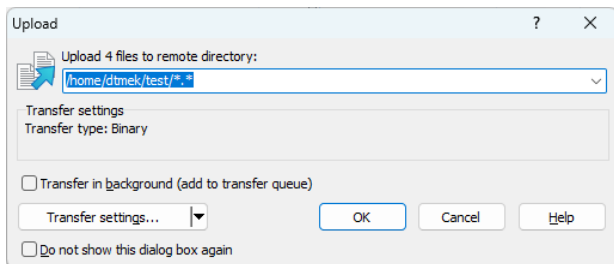
Marker på venstre side alle de filer der skal overføres (Samme måde som i stifinder):



Højreklik i det markerede, og vælg "Upload" og "Upload":



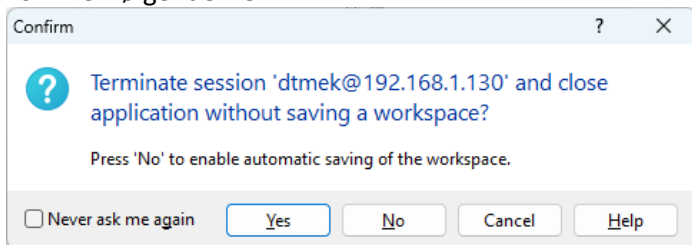
Herefter kommer der et vindue frem hvor man kan vælge hvor det uploadede skal ligge, det er standard samme sti man stod i på højre side, da man højreklikkede. Tryk OK hvis det er rigtig sti:



Måske kan man nå og se en popup, hvor maskinen kopier data, ellers kan man nu se at der er de samme filer på hver side:

D:\Docker.Undervisning\kopier\				/home/dtmek/test/				
Name	Size	Type	Changed	Name	Size	Changed	Rights	Owner
..		Parent directory	18-11-2024 14:15:20	..		18-11-2024 14:42:34	rxr-x---	dtmek
Site		Filmappe	18-11-2024 14:15:20	Site		18-11-2024 14:55:31	rxrxrxr-x	dtmek
Dockerfile	2 KB	Fil	25-10-2024 20:44:33	Dockerfile	2 KB	25-10-2024 20:44:33	rw-rw-r--	dtmek
entrypoint.sh	1 KB	SH-fil	25-10-2024 20:44:33	entrypoint.sh	1 KB	25-10-2024 20:44:33	rw-rw-r--	dtmek
nginx-site.conf	1 KB	CONF-fil	26-10-2024 18:08:25	nginx-site.conf	1 KB	26-10-2024 18:08:25	rw-rw-r--	dtmek

Derefter kan man lukke programmet, som ethvert andet windows program i øverste højre hjørne. Herefter kommer følgende frem:



Her svarer man "Yes" og man er ude af programmet. Hermed har man overført filer til sin worker server.

Vejledning slut.