

Øving 6

For Fredrik Johnsen

Oppgave 1

Først opprettet jeg en ressursgruppe i *West Europe*.

```
fredrik [ ~ ]$ az group create --name rg-demo-container --location westeurope
{
  "id": "/subscriptions/b9350ae8-22fe-4d2b-aff9-f76480a78621/resourceGroups/rg-demo-container",
  "location": "westeurope",
  "managedBy": null,
  "name": "rg-demo-container",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
fredrik [ ~ ]$
```

Deretter la jeg til en "hello-world"-container i denne ressursgruppen.

```
fredrik [ ~ ]$ az container create --resource-group rg-demo-container --name ci-demo-we-001 --image mcr.microsoft.com/azuredocs/aci-helloworld --dns-name-label tindemoaci001 --ports 80
Resource provider 'Microsoft.ContainerInstance' used by this operation is not registered. We are registering for you.
Registration succeeded.
{
  "confidentialComputeProperties": null,
  "containers": [
    {
      "command": null,
      "environmentVariables": [],
      "image": "mcr.microsoft.com/azuredocs/aci-helloworld",
      "instanceView": {
        "currentState": {
          "detailStatus": "",
          "exitCode": null,
          "finishTime": null,
          "startTime": "2024-11-03T14:36:18.037000+00:00",
          "state": "Running"
        }
      }
    }
  ]
}
```

Jeg verifiserte at containeren ble opprettet ved å sjekke statusen i portalen, og jeg fikk tilgang til den via både FQDN og IP-adresse.

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

fredrik.johnsen@tispfa...
TISP IKA STUDENT

Home > Resource groups >

Resource groups

TISP IKA Student

+ Create ... Group by none

You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.

rg-demo-container
Resource group

Search

+ Create Manage view Delete resource group Refresh Export to CSV Group by none

Overview

- Activity log
- Access control (IAM)
- Tags
- Resource visualizer
- Events
- Settings
 - Deployments
 - Security
 - Deployment stacks
 - Policies
 - Properties

Essentials

Subscription (move)
Azure for Students

Subscription ID
b9350ae8-22fe-4d2b-aff9-f76480a78621

Tags (edit)
Add tags

Deployments
No deployments

Location
West Europe

Resources Recommendations

Filter for any field...

Type equals all Location equals all Add filter

Name	Type	Location
ci-demo-we-001	Container instances	West Europe

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

fredrik.johnsen@tispfa...
TISP IKA STUDENT

Home > Resource groups > rg-demo-container >

ci-demo-we-001

Container instances

Search

Start Restart Stop Delete Refresh Give feedback

Overview

- Activity log
- Access control (IAM)
- Tags
- Settings
- Monitoring
- Automation
- Help

Essentials

Resource group (move) : rg-demo-container

Status : Running

Location : West Europe

Subscription (move) : Azure for Students

Subscription ID : b9350ae8-22fe-4d2b-aff9-f76480a78621

Tags (edit) : Add tags

SKU : Standard

OS type : Linux

IP address (Public) : 4.175.246.130

FQDN : timdemoaci001.westeurope.azurecontainer.io

Container count : 1

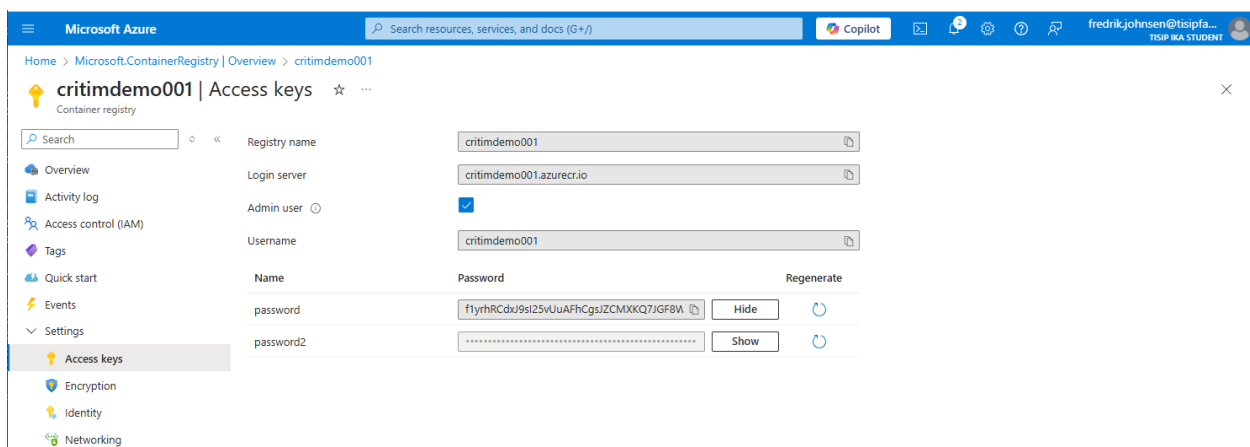
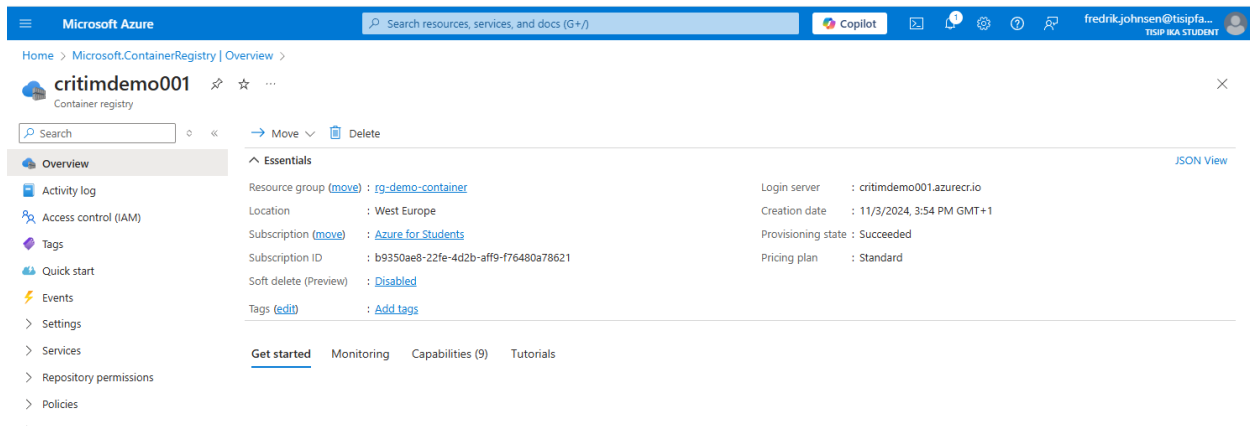
CPU Memory

timdemoaci001.westeurope.azurecontainer.io

Welcome to Azure Container Instances!



Videre opprettet jeg et container registry og hentet ut nødvendig påloggingsinformasjon (credentials) for å kjøre script i Cloud Shell.



Jeg lastet ned koden fra GitHub og åpnet den i Visual Studio Code i Cloud Shell.

Da jeg forsøkte å bygge prosjektet, fikk jeg feilmeldingen "TaskOperationsNotAllowed". Etter litt søk på Google fant jeg ut at dette skyldtes at jeg bruker en trial-versjon og ikke en pay-as-you-go-konto. Jeg vurderte det som utenfor oppgavens scope å finne en workaround for denne begrensningen.

Oppgave 2

Denne oppgaven fulgte veiledningen beskrevet her:

[Microsoft Learn: Deploy a Kubernetes Application to AKS](#)

Jeg startet med å opprette et AKS-kluster og koblet deretter til klusteret fra Cloud Shell.

Microsoft Azure

Search resources, services, and docs (G+/I)

Copilot

fredrik.johnsen@tisipfa...
TISIP IBA STUDENT

Home >

microsoft.aks-1730649143419 | Overview

Deployment

Search

Delete Cancel Redeploy Download Refresh

Overview

Inputs

Outputs

Template

Deployment is in progress

Deployment name: microsoft.aks-1730649143419
Subscription: Azure for Students
Resource group: rg-demo-aks

Start time: 11/3/2024, 4:53:49 PM
Correlation ID: 6cae5fb4-306a-422e-aca6-7cc770459917

Deployment details

Resource	Type	Status	Operation details
aks-demo001	Microsoft.ContainerService/man...	Created	Operation details

Microsoft Defender for Cloud
Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials
[Start learning today >](#)

Microsoft Azure

Search resources, services, and docs (G+/I)

Copilot

fredrik.johnsen@tisipfa...
TISIP IBA STUDENT

Home > microsoft.aks-1730649143419 | Overview

aks-demo001
Kubernetes service

Search

Create Connect Start Stop Delete Refresh Open in mobile Give feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Cost analysis

Kubernetes resources

Settings

Monitoring

Automation

Help

Essentials

Resource group: rg-demo-aks

Power state: Running

Cluster operation status: Succeeded

Subscription: Azure for Students

Location: UK South

Subscription ID: b9350ae8-22fe-4d2b-aff9-f76480a78621

Tags: Add tags

Kubernetes version: 1.29.10

API server address: aks-demo001-dns-dwdlkob0.hcp.uksouth.azmk8s.io

Network configuration: Azure CNI Overlay

Node pools: 1 node pool

Container registries: Attach a registry

Get started Properties Monitoring Capabilities (5) Recommendations (0) Tutorials

Kubernetes services

Encryption type: Encryption at-rest with a platform-managed key

Virtual node pools: Not enabled

Node pools

Node pools: 1 node pool

Kubernetes versions: 1.29.10

Node sizes: Standard_DS2_v2

Configuration

Kubernetes version: 1.29.10

Auto Upgrade Type: Patch

Automatic upgrade scheduler: Every week on Sunday (recommended)

Node security channel type: Node image

Networking

API server address: aks-demo001-dns-dwdlkob0.hcp.uksouth.azmk8s.io

Network configuration: Azure CNI Overlay

Pod CIDR: 10.244.0.0/16

Service CIDR: 10.0.0.0/16

DNS service IP: 10.0.0.10

Cilium dataplane: Not enabled

Network Policy: None

Load balancer: Standard

Private cluster: Not enabled

Authorized IP ranges: Not enabled

Application Gateway ingress controller: Not enabled

Integrations

```
fredrik [ ~ ]$ az aks get-credentials --resource-group rg-demo-aks --name aks-demo001
Merged "aks-demo001" as current context in /home/fredrik/.kube/config
fredrik [ ~ ]$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
aks-agentpool-40533995-vmss000000	Ready	<none>	5m31s	v1.29.10

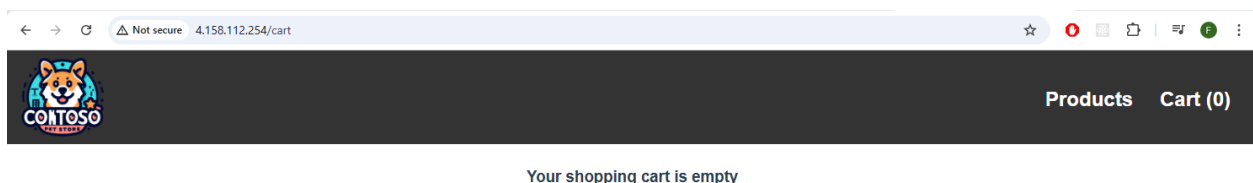
```
fredrik [ ~ ]$
```

Ved hjelp av Visual Studio Code la jeg inn konfigurasjonsfilen fra veiledningen som definerer klusteroppsettet, og startet klusteret ved å kjøre `kubectl apply`. Jeg kunne da bekrefte at opprettelsen ble fullført.

```
Bash
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
fredrik [ ~ ]$ code .
fredrik [ ~ ]$ kubectl apply -f aks-store-quickstart.yaml
deployment.apps/rabbitmq created
configmap/rabbitmq-enabled-plugins created
service/rabbitmq created
deployment.apps/order-service created
service/order-service created
deployment.apps/product-service created
service/product-service created
deployment.apps/store-front created
service/store-front created
fredrik [ ~ ]$ kubectl get pods~
error: the server doesn't have a resource type "pods~"
fredrik [ ~ ]$ kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
order-service-76d7f5b8f5-gkv5g      0/1     PodInitializing    0           56s
product-service-7566c548bd-vbfc2    0/1     ContainerCreating  0           56s
rabbitmq-6ddd848578-dbvzs           1/1     Running            0           56s
store-front-7cc6c7bb67-n6s7l        1/1     Running            0           56s
fredrik [ ~ ]$
```

Under oppstart gikk imidlertid *Product*-tjenesten inn i en "crash-loop", så jeg fikk dessverre ikke sett denne delen i aksjon.



aks-agentpool-40533995-vmss000000 | Pods ...

Node

Search

Refresh

Show labels

Give feedback

Overview

Pods

YAML

Events

Name	Namespace	Ready	Status	Restart count	Age	Pod IP
coredns-597bb9d4db-2dzbs	kube-system	1/1	Running	0	18 minutes	10.244.0.167
coredns-autoscaler-689db4649c-qx2gn	kube-system	1/1	Running	0	18 minutes	10.244.0.112
konnektivity-agent-9f97cc5f-csh5h	kube-system	1/1	Running	0	18 minutes	10.244.0.161
konnektivity-agent-9f97cc5f-jm747	kube-system	1/1	Running	0	18 minutes	10.244.0.55
azure-cns-c7k5z	kube-system	1/1	Running	0	18 minutes	10.224.0.4
azure-ip-masq-agent-d8v27	kube-system	1/1	Running	0	18 minutes	10.224.0.4
cloud-node-manager-nmmkr	kube-system	1/1	Running	0	18 minutes	10.224.0.4
csi-azuredisk-node-c6k9j	kube-system	3/3	Running	0	18 minutes	10.224.0.4
csi-azurefile-node-d7knm	kube-system	3/3	Running	0	18 minutes	10.224.0.4
kube-proxy-qsm4p	kube-system	1/1	Running	0	18 minutes	10.224.0.4
coredns-597bb9d4db-ptczp	kube-system	1/1	Running	0	17 minutes	10.244.0.92
metrics-server-7b685846d6-7ffmc	kube-system	2/2	Running	0	17 minutes	10.244.0.9
metrics-server-7b685846d6-nwqj6	kube-system	2/2	Running	0	17 minutes	10.244.0.60
eraser-controller-manager-6d5c649c8-75zpw	kube-system	1/1	Running	0	17 minutes	10.244.0.242
eraser-aks-agentpool-40533995-vmss000000-hr7je	kube-system	0/3	Completed	0	17 minutes	10.244.0.249
azure-wi-webhook-controller-manager-bfc8c986c-g6k...	kube-system	1/1	Running	0	17 minutes	10.244.0.76
azure-wi-webhook-controller-manager-bfc8c986c-btg...	kube-system	1/1	Running	0	17 minutes	10.244.0.126
order-service-76d7f5b8f5-gkv5g	default	1/1	Running	0	8 minutes	10.244.0.213
product-service-7566c548bd-vbfc2	default	0/1	CrashLoopBackOff	4	8 minutes	10.244.0.243
rabbitmq-6ddd848578-dbvzs	default	1/1	Running	0	8 minutes	10.244.0.25
store-front-7cc6c7bb67-n657l	default	1/1	Running	0	8 minutes	10.244.0.110

Besvarelse av oppgaver:

Jeg bruker ChatGPT til å lage gode svar, da det her i stor grad er notater til eksamensforberedelser, sånn jeg ser det.

Hva er fordelene med å benytte AKS fremfor en enkel Container Instance? Hva er fordelene med å benytte en enkel Container Instance fremfor et AKS cluster?

Å bruke Azure Kubernetes Service (AKS) fremfor en enkel Azure Container Instance (ACI) har flere fordeler, spesielt for mer komplekse applikasjoner som krever skalering, høy tilgjengelighet, og avansert orkestrering. Her er noen av de viktigste fordelene:

1. Skalering og orkestrering

- **AKS** gir innebygd støtte for skalering, både horisontalt (flere instanser av en applikasjon) og vertikalt (opp/ned-gradering av node-størrelse). Kubernetes orkestrerer og håndterer skaleringen dynamisk basert på behov.
- **ACI** tilbyr enkle ressursbehov for individuelle containere uten kompleks skalering. Den er bra for enkle, ad hoc-oppgaver, men gir ikke fleksibiliteten AKS tilbyr med klusterbasert skalering.

2. Høy tilgjengelighet

- **AKS** støtter distribusjon av containere over flere noder i et kluster, noe som sikrer høy tilgjengelighet selv om en node går ned. Kubernetes vil sørge for at pods flyttes til tilgjengelige noder.
- **ACI** er mindre egnet for høy tilgjengelighet, da det kun kjører enkle instanser som ikke har den samme redundansen eller feiloverlappingen som AKS gir.

3. Automatisk oppdatering og selvhelbredelse

- I **AKS** vil Kubernetes automatisk sørge for at ønsket antall containere alltid kjører, og den vil også rulle oppdateringer uten nedetid ved å gradvis oppdatere pods.

- **ACI** har ingen innebygd funksjon for selvhelbredelse eller automatiske oppdateringer. Hvis en container stopper, må den restartes manuelt eller via en ekstern løsning.

4. Støtte for komplekse applikasjoner og tjenester

- **AKS** er ideell for mikrotjenestearkitekturer hvor man kan orkestrere flere containere som må kommunisere med hverandre. AKS tilbyr tjenester som nettverkspolitikk, lastbalansering og DNS-oppslag internt i klusteret.
- **ACI** er best egnet for enklere applikasjoner eller oppgaver som ikke trenger kompleks samhandling med andre containere.

5. Integrasjon med CI/CD og DevOps-prosesser

- **AKS** støtter DevOps-arbeidsflyter veldig godt, og integreres med CI/CD-verktøy for automatisert deploy og testing. AKS kan også integreres med Helm for enkel applikasjonsutplassering.
- **ACI** har begrenset støtte for DevOps, da den mangler verktøy og strukturer som brukes til komplekse pipeline-konfigurasjoner.

Når bør man velge ACI?

- ACI kan være et godt valg for korte, engangs-kjøringer, batch-behandling, eller når man vil kjøre en isolert container uten behov for vedvarende infrastruktur eller skalerbarhet.

Når bør man velge AKS?

- AKS er ideell for store, komplekse applikasjoner med behov for skalerbarhet, pålitelighet, og muligheten til å håndtere mange containere som kjører sammen.

Hva er forskjellen mellom Azure Virtual Machines og Azure Containers, og gi fordeler og ulemper med hver tilnærming?

Det er en betydelig forskjell mellom Azure Virtual Machines (VMs) og Azure Containers, og hver tilnærming har unike fordeler og ulemper avhengig av bruksområdet og behovene til applikasjonen.

1. Azure Virtual Machines (VMs)

Azure VMs er fullverdige virtuelle maskiner som kjører sitt eget operativsystem (Windows eller Linux) og lar deg kjøre programmer som om du hadde tilgang til en fysisk server. Du har full kontroll over systemkonfigurasjonen, inkludert operativsystem, nettverk, og lagring.

Fordeler med Azure VMs:

- **Full kontroll over OS og maskinvare:** Du kan tilpasse alt fra operativsystem til programvare, nettverksinnstillinger og lagringskonfigurasjoner.
- **Kjøring av tradisjonelle applikasjoner:** Egner seg for applikasjoner som krever spesifikke OS-konfigurasjoner, eller for eldre applikasjoner som ikke er tilpasset containere.
- **Permanent lagring:** VMs har vedvarende diskere som kan opprettholde data mellom omstarter.
- **Støtte for kompleks infrastruktur:** Egner seg godt for applikasjoner med behov for dypere integrasjon med andre tjenester eller behov for spesialiserte nettverkstopologier.

Ulemper med Azure VMs:

- **Ressursintensivt og tregere oppstartstid:** Hver VM inneholder hele operativsystemet, så det tar lengre tid å starte opp og bruker flere ressurser sammenlignet med containere.
- **Krevende vedlikehold:** Selv om VMs kan automatiseres, kreves det fortsatt vedlikehold for OS-oppdateringer, sikkerhetskonnfigurasjoner, og optimalisering.

- **Høyere kostnader:** VMs har ofte høyere driftskostnader sammenlignet med containere, spesielt for applikasjoner som krever hyppig skalering.

2. Azure Containers (Azure Container Instances og Azure Kubernetes Service)

Azure Containers gir en lettvektsplattform hvor applikasjoner kan kjøres i isolerte containere, som deler operativsystemet til den underliggende maskinvaren. Containere har alle nødvendige filer for å kjøre applikasjonen, men de er mindre og raskere enn virtuelle maskiner.

Fordeler med Azure Containers:

- **Rask oppstart og lettvekts:** Containere starter på sekunder siden de ikke trenger å laste et helt operativsystem. Dette gjør dem ideelle for applikasjoner med variable belastningsmønstre.
- **Skalerbarhet og fleksibilitet:** Containere kan enkelt skaleres horisontalt (flere containere) og integreres med orkestreringssystemer som Kubernetes for dynamisk ressursfordeling.
- **Konsistens og portabilitet:** Med containere kan utviklere lage applikasjoner som oppfører seg konsistent, uansett hvor de kjøres, noe som gir fleksibilitet mellom ulike utviklings-, test- og produksjonsmiljøer.
- **Lavere kostnader:** Containere deler operativsystemet på vertsnivå, noe som gir lavere ressursbruk og dermed ofte lavere kostnader sammenlignet med VMs.

Ulemper med Azure Containers:

- **Mindre kontroll over OS:** Siden containere deler vertens operativsystem, har man mindre fleksibilitet til å tilpasse OS-nivåkonfigurasjoner og spesialinnstillinger.
- **Ikke egnet for alle typer applikasjoner:** Applikasjoner som krever et spesifikt OS-miljø eller lavnivå tilgang til maskinvare fungerer ikke alltid godt i containere.

- **Datavarelagring og vedvarende tilstand:** Containere er typisk kortlivede og ephemeral, noe som betyr at vedvarende lagring kan være mer komplekst å sette opp sammenlignet med VMs.

Når bør man velge Azure Virtual Machines?

- For applikasjoner som krever et tilpasset operativsystem, dyp systemkonfigurasjon, eller hvis man trenger en server som skal kjøre kontinuerlig med vedvarende datalagring.
- Eldre applikasjoner som ikke er tilpasset containere, eller applikasjoner med spesifikke krav til OS og nettverksinnstillinger.

Når bør man velge Azure Containers?

- For moderne mikrotjenestearkitekturer som trenger rask skalering og enkel oppstart, samt for applikasjoner som har høye variasjoner i belastning eller kortsiktige behandlingsbehov.
- Applikasjoner som kan dra nytte av portabilitet og konsistens mellom miljøer, eller som trenger et lettvektskjøremiljø uten omfattende OS-konfigurasjoner.

Sammendrag:

- **Azure VMs** gir full kontroll og egner seg for komplekse, permanente tjenester med spesifikke krav til operativsystem.
- **Azure Containers** tilbyr en lettvekts og skalerbar løsning for moderne applikasjoner og mikrotjenester, men med mindre kontroll på OS-nivå.

Valget avhenger av applikasjonskrav, kostnader, og hvor stor fleksibilitet og kontroll som trengs i miljøet ditt.

Etterord: Hadde litt trøbbel med å få gjort oppgavene, da den ene oppgaven jeg prøvde å utføre ikke var støttet med trial, og i tillegg ville ikke den ene av ressursene i clusteret starte opp. Ellers gikk det greit.

