

40 yhp

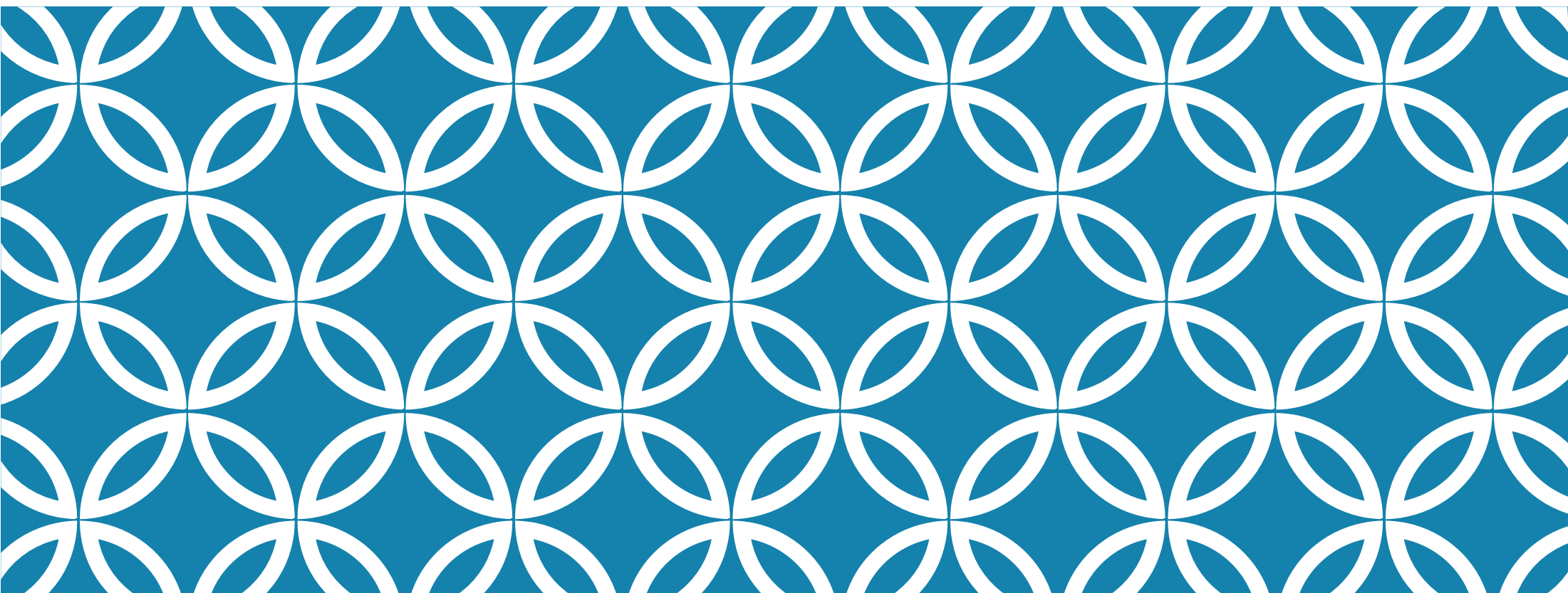
Campus Mölndal

---

# GRUNDLÄGGANDE MOLNAPPLIKATIONER



VECKA 5



# CI/CD PIPELINES

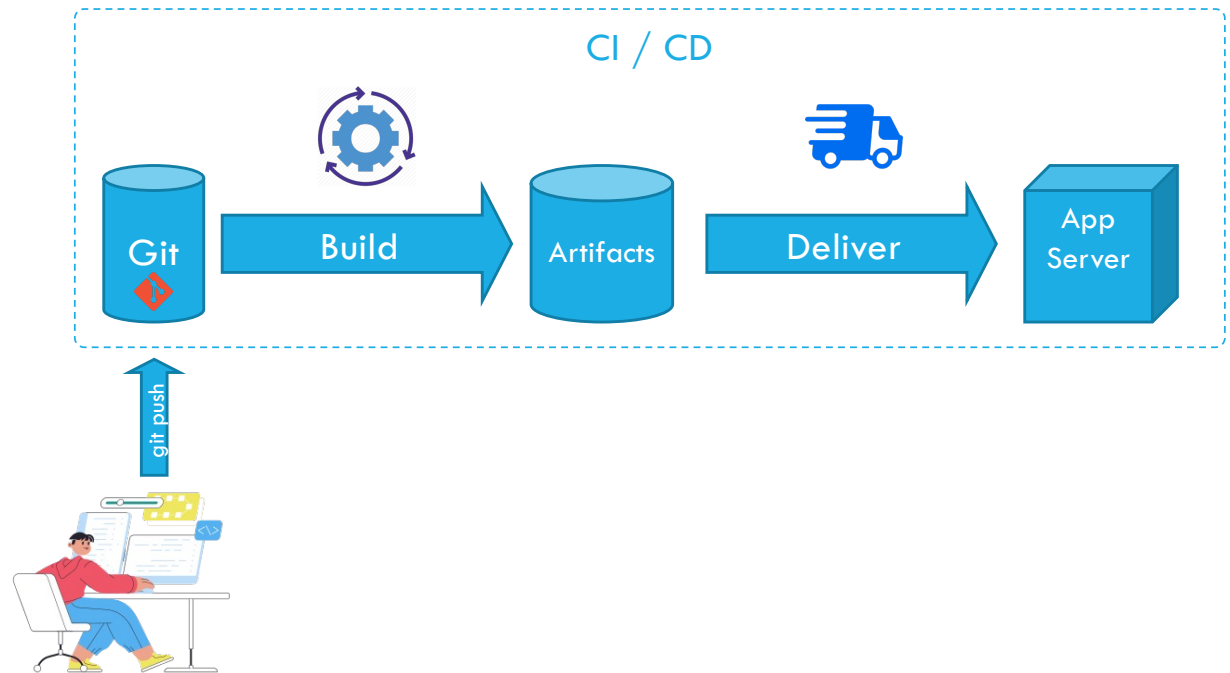
Vecka 5

# VECKANS MÅL

## CI/CD PIPELINE

Förstå syftet med en CI/CD pipeline och hur den passar in i utvecklingsarbetet

Skapa en CI/CD pipeline med Github Actions som driftsätter en webbadapplikation på en virtuell server



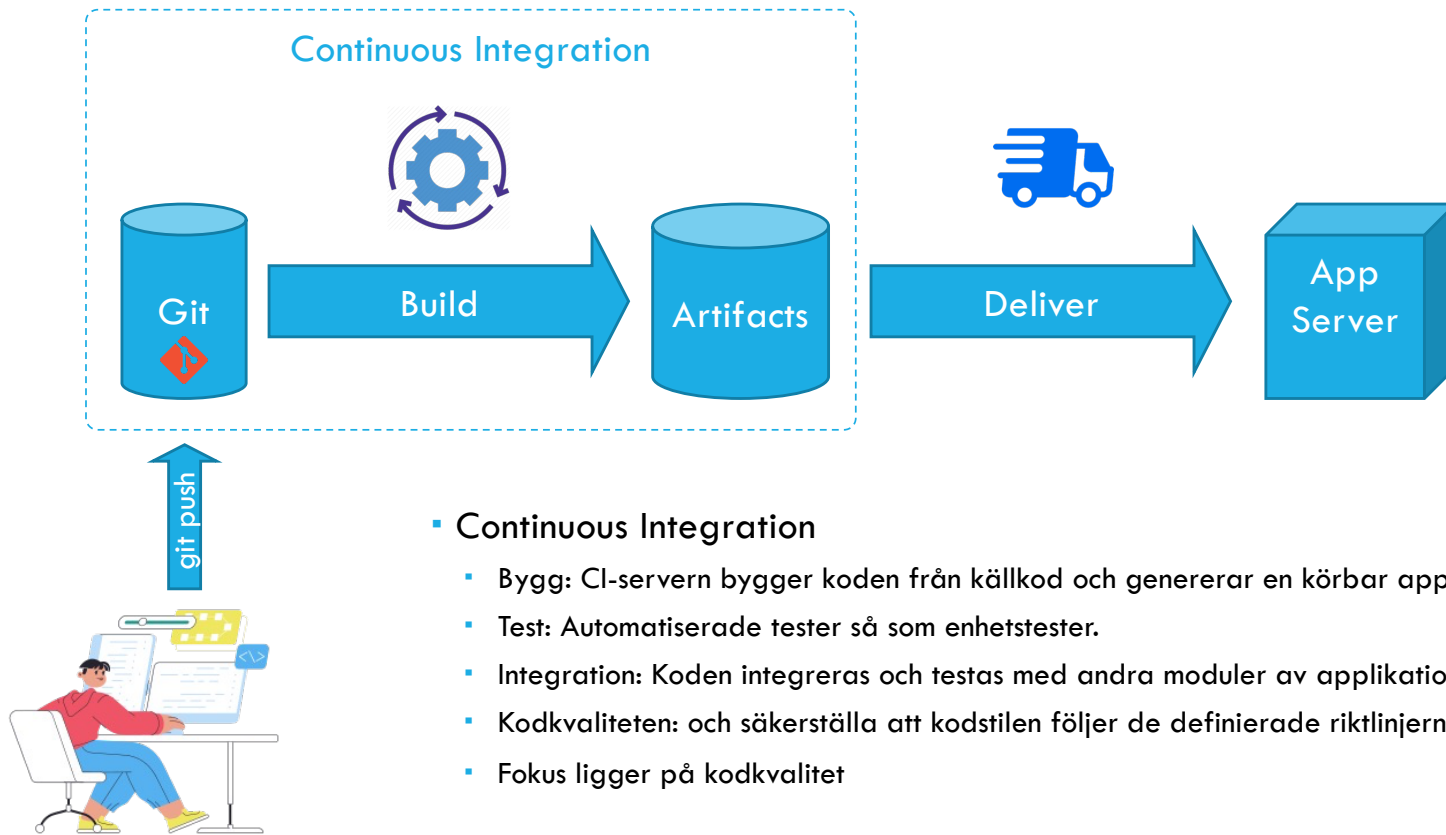
# VAD ÄR EN CI/CD PIPELINE?

- Continuous Integration/Continuous Delivery (Deployment)
  - automatiserar releaseprocessen för mjukvaruapplikationer
  - bygg, testa och driftsätt
  - möjliggör en snabbare releaseprocess med tätare releaser till en lägre kostnad
  - viktigt instrument i den agila arbetsprocessen
  - färre förändringar i varje release gör den lättare att hantera och minskar risken för fel

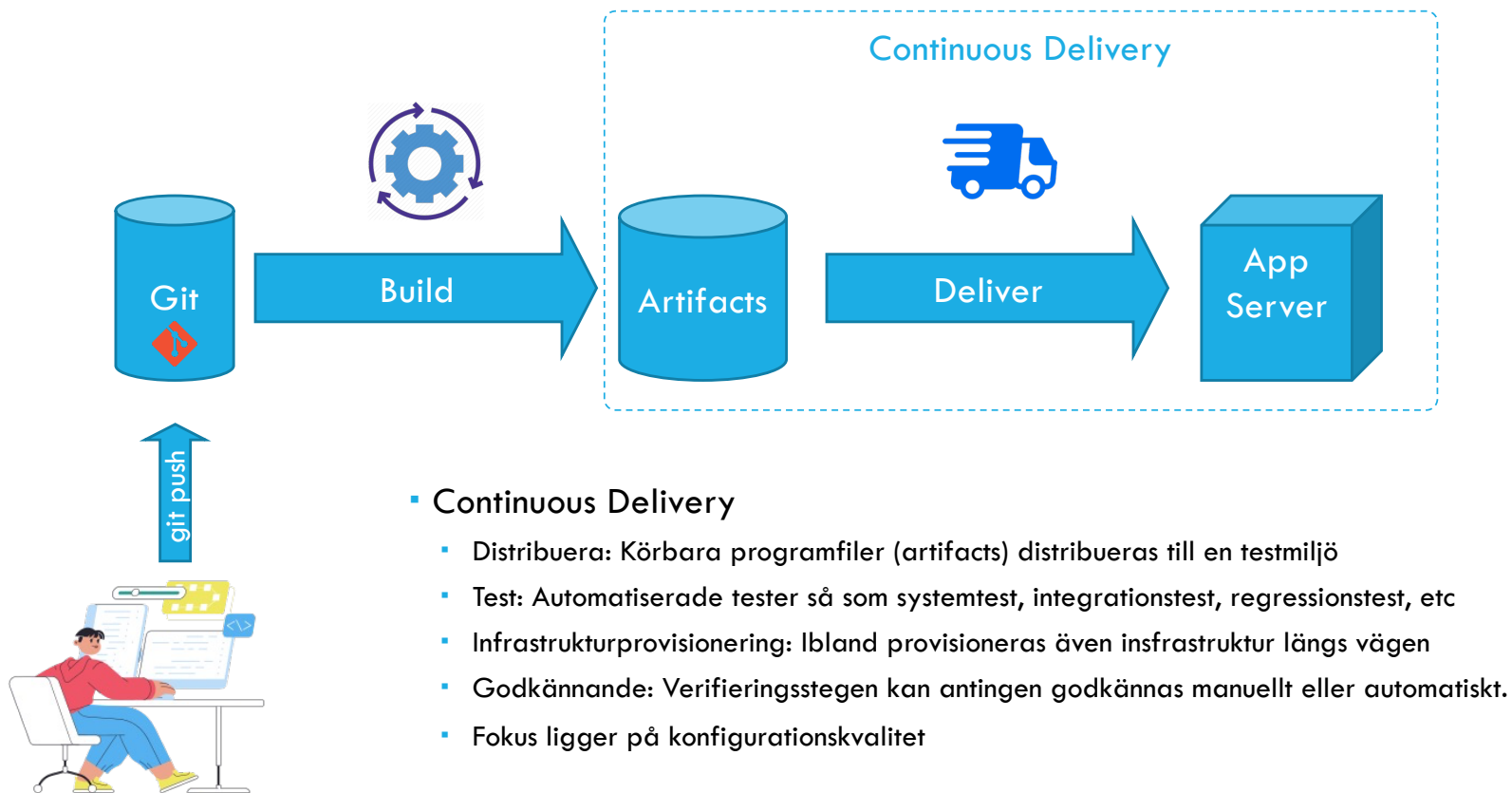
Continuous **Delivery** – Kedjan stannar upp för manuella godkännanden innan PROD

Continuous **Deployment** – Driftsättningen är helt automatisk

# CONTINUOUS INTEGRATION (CI)



# CONTINUOUS DELIVERY (CD)



# CI / CD

## Continuous Integration / Continuous Delivery

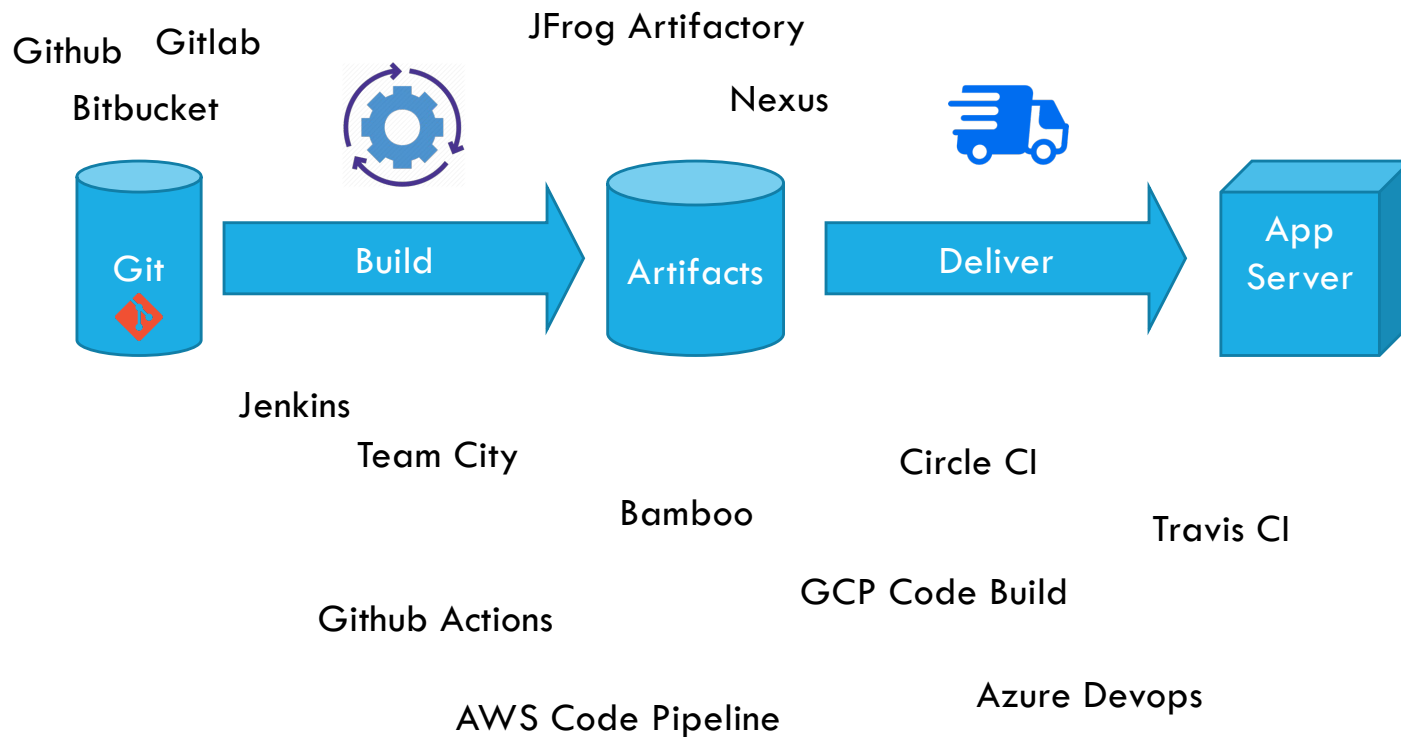


### CI / CD Pipeline

- En traditionell releasecykel är ofta en gång i månaden
- Med en CI/CD pipeline kan det göras varje dag om det behövs
- Infrastrukturprovisionering gör att infrastrukturen har ett väldefinierat tillstånd innan testerna börjar
- Automatiska tester underlättar och ökar hastigheten på pipeline



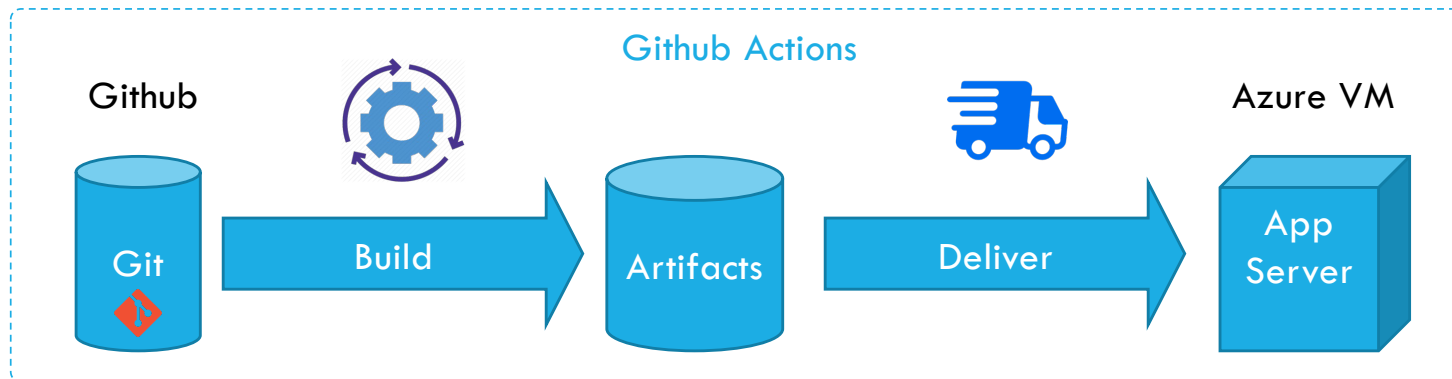
# NÅGRA OLIKA PRODUKTER



# DEMO: GITHUB ACTIONS

Mål:

Bygga och deploya en webapp på en Azure VM med hjälp av Github Actions



# VAD ÄR GITHUB ACTIONS?

- Integrerad molnbaserad CI/CD-tjänst från GitHub.
- Använder **YAML-baserade** konfigurationsfiler för att definiera olika workflow, jobb och steg som ska utföras.
- **jobs** sker parallellt medan **steps** sker i ordning
- Ett jobb utförs av en **Runner**, som kan vara antingen
  - github-hosted
  - self-hosted



# VARFÖR GITHUB ACTIONS?

Förutsättningar: repo på Github och VM på Azure

Här är några argument för att välja Github Actions:

1. Enkel integration med Github-repo
2. Molnbaserad CI/CD-lösning
3. Stort ekosystem av integrationer
4. Enkel konfiguration via en YAML som versionshanteras tillsammans med koden
5. Förvärvades av Microsoft 2018
6. Gratis att använda för öppen källkod
7. Stor community

# VAD ÄR YAML?

- YAML Ain't Markup Language
  - representerar data i en strukturerad form, liknande JSON
  - YAML ska göra det lättare att läsa för människor
  - kan hantera kommentarer

indenteringen beskriver datastrukturen!

- ett YAML-dokument *bör* inledas med ---

## YAML

```
---
# Strängar
name: "John Doe"
description: 'This is a YAML example'

# Numeriska värden
age: 30
price: 9.99

# Booleska värden
is_enabled: true
is_active: false

# Listor
fruits:
  - apple
  - orange
  - banana

# Dictionaries (eller objekt)
person:
  name: John
  age: 30
  address:
    street: 123 Main St
    city: Anytown

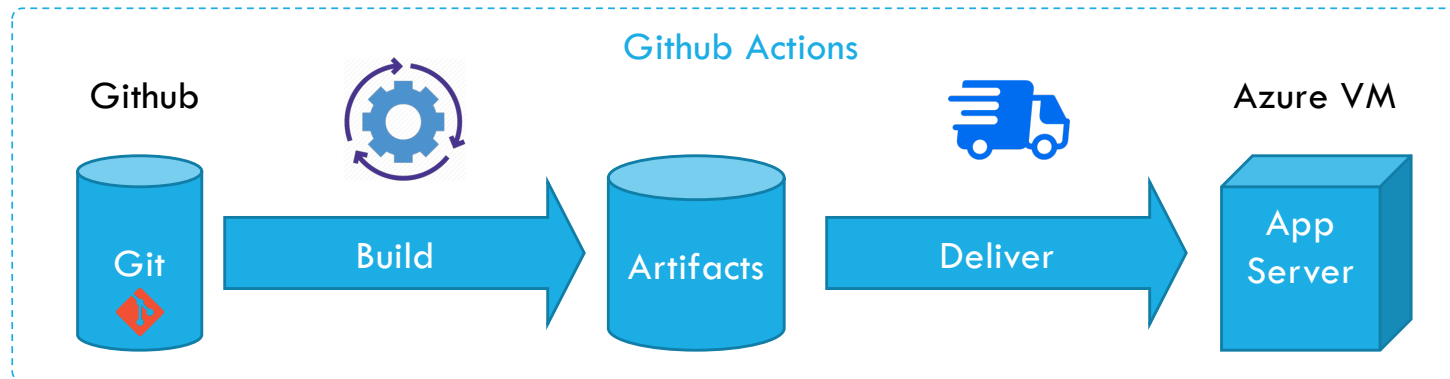
# Null-värden
no_value: null
```

## JSON

```
{
  "name": "John Doe",
  "description": "This is a JSON example",
  "age": 30,
  "price": 9.99,
  "is_enabled": true,
  "is_active": false,
  "fruits": [
    "apple",
    "orange",
    "banana"
  ],
  "person": {
    "name": "John",
    "age": 30,
    "address": {
      "street": "123 Main St",
      "city": "Anytown"
    }
  },
  "no_value": null
}
```

<https://www.json2yaml.com>

# GITHUB ACTIONS



# GITHUB ACTIONS

## Github Actions

```
name: <workflow-namn>

on:
  <trigger-configuration>

jobs:
  <job-namn>:
    runs-on: <runner-typ>
    steps:
      - <steg-konfiguration>
      - <steg-konfiguration>
      ...
```

**name** namnet på ett **workflow**

**on** definierar **triggern** när ett workflow ska köras

- exempelvis varje gång en ny commit görs på en specifik branch

**jobs** definierar en eller flera **jobb** som ska köras ett workflow

- Varje jobb har
  - ett namn
  - en runner
  - en lista med steg

**steps** definierar en åtgärd som ska utföras

- exempelvis att checka ut koden eller köra ett kommando



# GITHUB ACTIONS

## Github Actions

steps:

- name: Install .NET  
uses: actions/setup-dotnet@v3  
with:  
dotnet-version: 7.0.x
- name: Check out repo  
uses: actions/checkout@v3
- name: Build and publish  
run: |  
dotnet build  
dotnet publish -o \${env.DOTNET\_ROOT}/publish
- name: Upload artifacts  
uses: actions/upload-artifact@v3  
with:  
name: my-artifacts  
path: \${env.DOTNET\_ROOT}/publish

**name** namnet på ett step

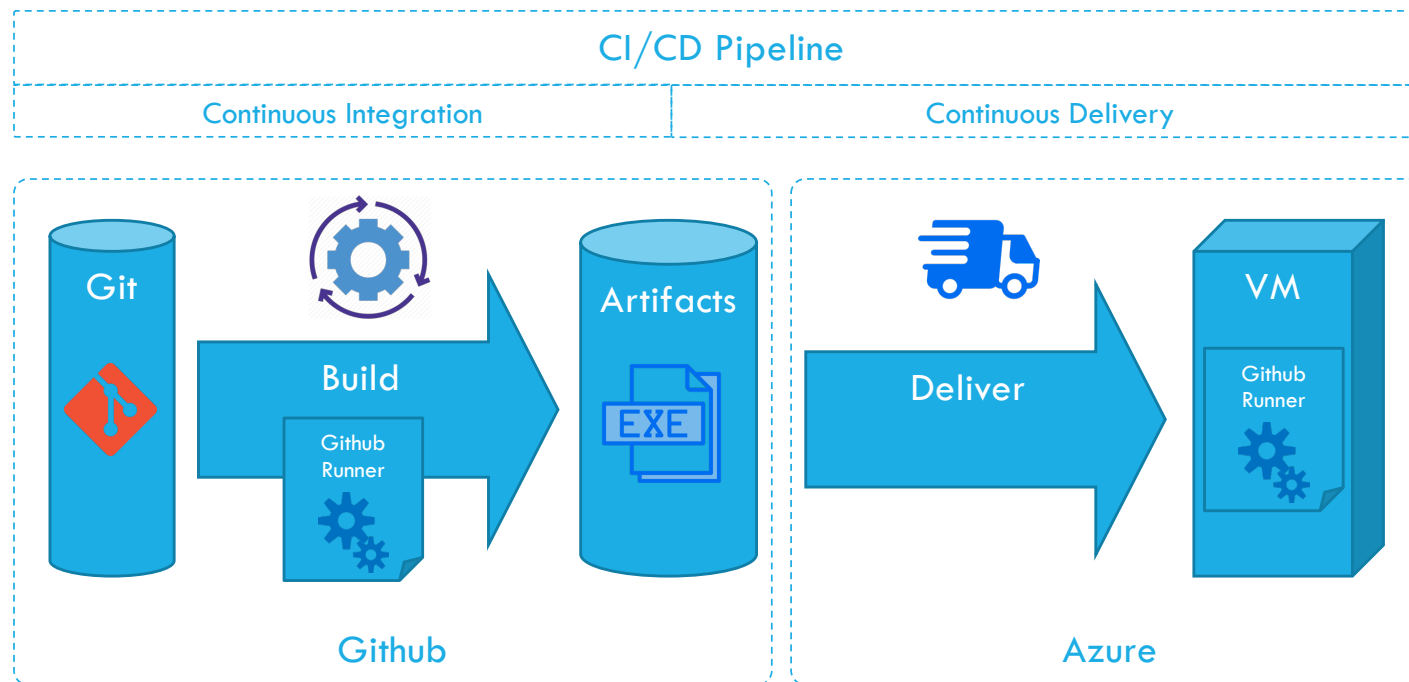
**uses** refererar till en action som finns definierat i ett repo

**with** tillhandahåller parametrar till action-scriptet

**run** kör kommandon (runnern avgör vilken typ. ex. Ubuntu = bash)

Github Actions har även stöd för **miljövariabler** och **secrets**

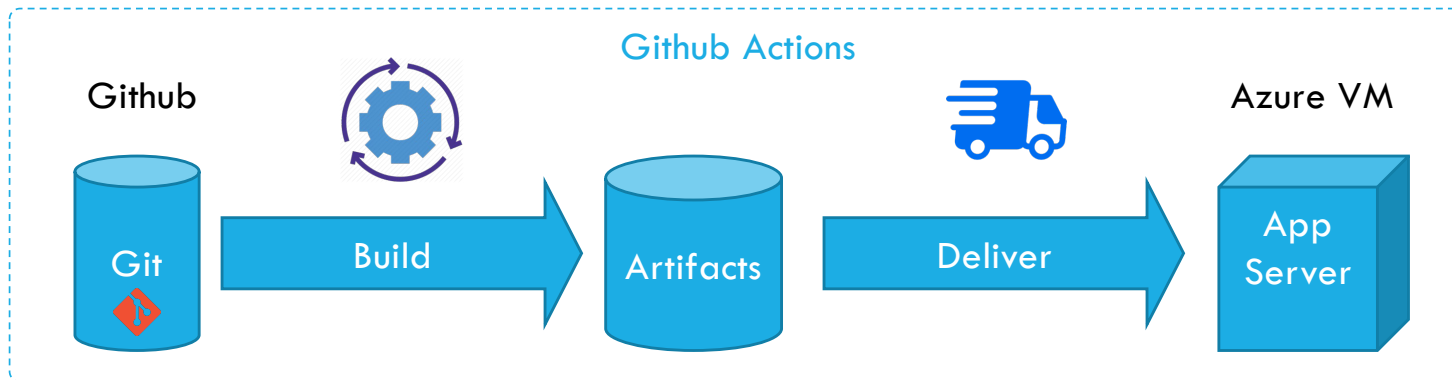
# DEMO: GITHUB ACTIONS



## Mål

Bygga och deploya  
en webapp på en VM  
med hjälp av Github  
Actions

# ÖVNING: CI/CD PIPELINE



- 1) Skapa ett nytt repo på Github
- 2) "Utveckla" din .Net webapp och checka in den
- 3) Skapa en "action" som bygger din applikation
- 4) Provisionera en Azure VM och installera en "Runner"
- 5) Skapa en "action" som laddar ner programartefakterna till din VM
- 6) Fortsätt automatisera till dess applikationen kan driftsättas utan manuella steg