

UMBRACO CMS

Prestandajämförelse av Umbraco CMS i molnet
Umbraco Cloud vs Microsoft Azure

Kurs | Examensarbete

Skola | Teknikhögskolan

Ort / Datum | Göteborg, 2025-05-28

Namn | Sanna Wiklund

Utbildning | .NET-Utvecklare

Handledare | William Enander

Innehållsförteckning

Ordlista.....	2
Förord.....	3
Abstract.....	4
Sammanfattning.....	4
1. Inledning	5
1.1 Bakgrund.....	5
1.1.1 Umbraco	5
1.1.2 Umbraco Cloud	5
1.1.3 Azure	5
1.2 Syfte	6
1.3 Problemformulering.....	6
1.3.1 – Subfråga	6
1.3.2 - Subfråga.....	6
1.3.3 - Subfråga.....	6
1.4 Avgränsningar och fokus.....	6
1.5 Metod/Arbetsätt	6
1.5.1 Tekniska förutsättningar	7
1.5.2 Testmetodik.....	7
2. Teori.....	8
3. Resultat.....	9
3.1 Mätvärden	9
3.1.1 Inbyggd funktionalitet.....	10
3.1.2 Avgränsade mätvärden.....	11
3.2 Testresultat	12
4. Diskussion.....	16
4.1 Prestandaskillnader mellan Cloud och Azure.....	17
4.2 Resursförbrukning vid trafiktappar	17
4.2.1 Svarstider och sidladdningstider under belastning	17
4.2.2 Reflektioner och förbättringar	18
5. Slutsatser	18
5.1 Rekommendationer	19
Bilagor	19
Testdata.....	19
Referenslista	20

Ordlista

- **CMS (Content Management System)** – Ett system för att skapa, hantera och publicera digitalt innehåll, ofta använt för webbplatser. Exempel är Umbraco, WordPress och Drupal.
- **Hosting** – Processen att lagra och tillhandahålla en webbplats på en server.
- **SaaS (Software as a Service)** – Mjukvara som nås via internet, t.ex. Google Docs eller Dropbox, där leverantören hanterar allt.
- **PaaS (Platform as a Service)** – En plattform för att bygga, köra och hantera applikationer utan att hantera själva infrastrukturen, t.ex. Azure App Service.
- **Azure** – Microsofts molnplattform som erbjuder tjänster för hosting, databaser, AI, lagring etc
- **Azure App Service** – En PaaS-tjänst inom Azure för att hosta webbappar utan att hantera servrar.
- **Open Source** – Programvara med öppen källkod som kan användas, ändras och distribueras fritt beroende på licens.
- **MIT licens** – En öppen källkodslicens som tillåter fri användning, modifiering och distribution av mjukvara, med minimala restriktioner – så länge upphovsrättsmeddelandet behålls.
- **Prestanda** – Webbplatsens hastighet och responsivitet under olika belastningsnivåer.
- **Skalbarhet** – Förmågan att hantera ökad trafik och resursanvändning genom att utöka kapacitet, exempelvis genom lastbalansering eller autoskalning.
- **Belastningstest** – Ett test där en webbplats utsätts för simulerad trafik för att analysera hur den presterar vid olika belastningsnivåer.
- **Svarstid** – Tiden det tar för servern att svara på en förfrågan från en användare.
- **CPU (Central Processing Unit)** – Den centrala processorenheten, dvs den primära komponenten i en dator som agerar som ett ”kontrollcenter”.
- **RAM (Random Access Memory)** – Temporärt minne som används för att lagra data för snabb åtkomst vid körning av applikationer.
- **IDE (Integrated Development Environment)** - Ett integrerat utvecklingsverktyg som kombinerar flera funktioner som kodredigering, felsökning, versionshantering och byggverktyg i ett och samma gränssnitt. Exempel på IDE:er är Visual Studio
- **Visual Studio** - En utvecklingsmiljö (IDE) från Microsoft som används för att skriva, testa och debugga kod, främst för .NET-applikationer.
- **DDoS (Distributed Denial of Service)** - En typ av cyberattack där många datorer överbelastar en server eller webbplats med trafik, vilket gör tjänsten otillgänglig för användare.
- **Igloo** - Ett färdigt design- och funktionsbibliotek utvecklat för Umbraco CMS som innehåller en uppsättning fördefinierade komponenter, mallar och byggblock.
- **Igloo Widget** - En enskild komponent inom Igloo-biblioteket som representerar ett specifikt innehållsblock eller funktion, exempelvis en bildmodul, textsektion eller knapp.
- **Data Aggregation** - Processen att samla in data från flera källor och sammanställa det till en enhetlig vy eller rapport.
- **TTFB (Time to First Byte)** – Tiden det tar innan användaren får det första svaret från servern.
- **http_req_duration (avg, p95) - Avg (average)** visar den genomsnittliga svarstiden för samtliga förfrågningar.
 - **p95 (95:e percentilen)** visar den svarstid inom vilken 95 % av alla förfrågningar besvarades.
- **URL (Uniform Resource Locator)** - En standardiserad adress som används för att ange sökvägen till en resurs på internet, till exempel en webbsida eller ett API-endpoint. T.ex. <https://www.exempel.se/api/data>.
- **TXT**: Filformat som innehåller textfiler med enbart enkel text utan formatering.
- **CSV**: Comma Separated Value-filer lagrar information avgränsad av komma istället för att lagra den i kolumner.

Förord

Jag har haft förmånen att genomföra både min första och andra LIA-period hos Webmind, en webbyrå baserad i Göteborg med specialisering inom utveckling av webbplatser med hjälp av Umbraco CMS.

Praktiken ledde till min första kontakt med ett CMS, och det blev således även min introduktion till att arbeta med Umbraco som plattform. Detta är även anledningen till att jag valt att analysera just Umbraco i detta arbete.

Jag vill rikta ett **varmt tack** till Webmind för det förtroende och stöd som jag har fått under min tid hos dem. Jag vill särskilt tacka för tillgång till sådant som installation av testmiljöer, licenser och andra resurser som möjliggjort genomförandet av detta arbete – något som inte hade varit genomförbart med endast de gratistillgängliga versionerna av Umbraco.

Abstract

This thesis aims to investigate and analyze how the choice of hosting affects the performance of Umbraco platforms — specifically comparing Umbraco Cloud with Umbraco hosted on Microsoft Azure.

To collect relevant data quantitative tests were conducted in both environments. The study followed a hands-on testing methodology using the performance testing tool K6, where five different types of performance tests were carried out. These tests were performed both before and after content was added to each respective website.

The results show that the choice of hosting environment plays a critical role in performance. Azure provides better stability, capacity, and more advanced monitoring capabilities, but it also requires more system resources and technical expertise.

Umbraco Cloud on the other hand is easier to manage and use, making it a more attractive option for organizations with limited resources.

This study offers insights that can hopefully support both developers and users of Umbraco in choosing the most suitable platform based on their specific needs and requirements.

Sammanfattning

Detta examensarbete går ut på att undersöka och analysera hur val av hosting påverkar prestandan på Umbraco-plattformar – Umbraco Cloud och Umbraco hostat på Microsoft Azure.

För att samla in relevant data har kvantitativa tester genomförts i dessa två miljöer. Arbetet utgick från en praktisk testmetodik där fem olika typer av belastningstester genomfördes med hjälp av testverktyget K6. Dessa tester utfördes både innan och efter att innehåll byggdes på respektive webbplats.

Sammanfattningsvis visar arbetet att valet av hostingmiljö spelar en avgörande roll för prestandan. Azure erbjuder bättre stabilitet och kapacitet, samt fler tekniska möjligheter vid övervakning av webbplatsen. Men det kräver även fler resurser och teknisk kunskap. Umbraco Cloud är enklare att använda och hantera, vilket gör det till ett attraktivt projekt för organisationer med begränsade resurser.

Studien bidrar till insikter som förhoppningsvis kan vägleda både utvecklare och användare av Umbraco vid val av plattform, utifrån dess specifika behov och krav.

1. Inledning

1.1 Bakgrund

Molnbaserade plattformar har blivit alltmer populära tack vare dess flexibilitet när det kommer till faktorer som lagring, hantering och bearbetning av data. Detta möjliggör även för organisationer att snabbt anpassa sig till förändrade behov utan att behöva investera i en kostsam och underhållskrävande infrastruktur. Denna utveckling har lett till en ökad efterfrågan på molnbaserade CMSer, där Umbraco är en av de mer välkända och etablerade alternativen.

I detta examensarbete undersöks två olika metoder för att hosta en Umbraco-webbplats: **Umbraco Cloud** och **Umbraco på Microsoft Azure**. Syftet är att jämföra prestanda mellan dessa två lösningar samt analysera deras respektive för- och nackdelar utifrån ett tekniskt och praktiskt perspektiv.

1.1.1 Umbraco

Umbraco [1] är ett open source CMS skrivet i C# (.NET) med MIT-licens, som är känt för sin flexibilitet och användarvänlighet. Det används av både små och stora företag och erbjuder en kraftfull plattform för att hantera webbplatser, e-handelslösningar och andra digitala tjänster.

Umbraco erbjuds som en fristående plattform som kan köras på exempelvis Microsoft Azure, vilket är en **Platform as a Service (PaaS)**. Detta innebär att utvecklare själva ansvarar för hostingmiljön samt hantering av uppdateringar, säkerhet och skalbarhet. Denna lösning ger stor frihet och kontroll, men kräver samtidigt ett större tekniskt ansvar.

1.1.2 Umbraco Cloud

För organisationer som vill minska det tekniska ansvaret utan att kompromissa med funktionalitet, erbjuds **Umbraco Cloud** [2] – en **Software as a Service (SaaS)**-lösning. Här sköts all infrastruktur, hosting, uppdateringar och säkerhet av Umbraco själva, vilket frigör resurser för användaren att fokusera på innehåll och utveckling.

Umbraco Cloud är en helhetslösning som kombinerar plattformens kraftfulla funktioner med fördelarna av en underhållsfri molntjänst, och är därför ett attraktivt alternativ för företag som prioriterar enkelhet och driftsäkerhet.

1.1.3 Azure

Azure [3] är en molnplattform utvecklad av Microsoft, som erbjuder en rad olika tjänster för bland annat hosting, datalagring och applikationsutveckling. Anledningen till att just Azure har valts som alternativ hostingplattform är för att Umbraco också tillhör Microsoft, då det är

utvecklat i .NET. Denna samhörighet gör Azure till ett relevant och logiskt val som jämförelseplattform i detta sammanhang.

1.2 Syfte

Syftet med arbetet är att med hjälp av diverse belastningstester kunna utgöra vilka faktorer som påverkar prestandan på en Umbraco-plattform, och hur dessa skiljer sig baserat på val av hosting för Umbraco-plattformen i fråga.

1.3 Problemformulering

Min primära frågeställning är:

Hur påverkar val av hosting prestanda för Umbraco-Webplatser?

För att besvara den primära frågeställningen har följande del-frågor tagits fram för att gå djupare in på de olika perspektiven som är relevanta för undersökningen:

1.3.1 – Subfråga

Vilka specifika prestandaskillnader uppstår mellan en Umbraco-webbplats hostad på en PaaS (Azure), jämfört med en Umbraco Cloud lösning under olika belastningsnivåer?

1.3.2 - Subfråga

Hur ser resursförbrukningen ut mellan de två hostingalternativen vid trafiktoppar?

1.3.3 - Subfråga

Hur skiljer sig svarstider mellan Umbraco Cloud och Azure-hosting vid olika belastningsnivåer?

1.4 Avgränsningar och fokus

Som tidigare nämnt kommer detta arbete att ha fokus på prestanda, men varför just prestanda? Enligt rapporten **Svenskarna och internet** [4] från Internetstiftelsen, använder 96 procent av befolkningen i åldrarna 15 år och uppåt internet, varav 91 procent av dessa använder det dagligen. Detta innebär att vi i mycket stor utsträckning interagerar med digitala system, webbplatser och applikationer. Hur väl dessa tjänster presterar är inte bara av betydelse för användarupplevelsen, utan även för de aktörer som tillhandahåller tjänsterna.

På grund av detta kommer detta arbete gå ut på att undersöka vilka faktorer som påverkar prestandan, snarare än att fokusera på vilka faktorer som prestandan i sig kan påverka. Aspekter som kostnader för leverantören och säkerhet är exempel på områden där prestanda spelar en stor roll, men dessa kommer inte diskuteras i denna rapport.

1.5 Metod/Arbetssätt

Målet med arbetet är att få en utökad förståelse för prestanda genom analysering av data. Denna data kommer att tas fram med hjälp av kvantitativa tester.

1.5.1 Tekniska förutsättningar

För att möjliggöra en rättvis jämförelse mellan testresultaten har testmiljöerna konfigurerats med så likvärdiga förutsättningar och tekniska specifikationer som möjligt.

Båda miljöerna är konfigurerade i .NET 8 med Umbraco version 13.6.0 och Igloo version 13.1.5.

1.5.1.1 Testmiljöer

Min Umbraco Cloud-lösning är baserad på plattformspanen **Starter** [5], Igloo har blivit installerat. Eftersom de tester som ska genomföras liknar en överbelastningsattack (DDoS), så finns det ett behov av att köra lösningen på en dedikerad serverresurs. Då man inte kan styra fördelning av resurser i Umbraco Cloud-miljön manuellt, har Starter-planen utökats med hjälp av tillägget **Dedicated 1** [6] som erbjuds i Umbraco Cloud-lösningar. Detta innebär att plattformen automatiskt tilldelas en egen resurs i Umbracos infrastruktur. Denna plan erbjuder två CPU-kärnor och 3,5 GB ram.

För att efterlikna min cloud-lösning blev den alternativa hostingmiljön konfigurerad som en **Azure App Service** [7]. Denna instans är konfigurerad med motsvarande resursspecifikationer (*2 CPU-kärnor och 3,5 GB ram - Tjänsteprenumeration Basic B2*), som även denna körs på en enskild resurs. På denna plattform har en instans av Umbraco CMS installerats tillsammans med en Igloo-installation, för att säkerställa ytterligare ett lager av jämförbarhet mellan miljöerna.

Användningen av Igloo underlättar när det kommer till innehållshantering, och det möjliggör en snabb uppbyggnad av innehåll för båda miljöernas webbplatser. Inom Umbraco kallas innehållet för content och det kan bestå av exempelvis bild/video-filer, texter och widgets [8].

1.5.1.1.1 Umbraco och content

I Umbraco lagras allt content i en relationsdatabas, vanligtvis i Microsoft SQL Server, vilket är det som används via en lokal instans i SQL Server Management Studio (SSMS).

Varje innehållselement som exempelvis media och metadata lagras som en post i databasen, där strukturella relationer och innehållsvärden hanteras via tabeller.

Detta innebär att varje förfrågan till webbplatsen kan medföra ett flertal databasanrop, särskilt vid dynamiska renderingar. På grund av detta kan mängden innehåll på en webbsida ha en direkt påverkan på prestandan, och för att undersöka detta kommer prestandatester att utföras både med och utan content. Syftet med detta är kort och gott att se hur mängden content påverkar sådant som svarstider och resursutnyttjande.

1.5.2 Testmetodik

För att samla den data som behövs för att kunna jämföra och analysera prestandan på mina plattformar kommer kvantitativa prestandatester att genomföras.

1.5.2.1 K6

När det kommer till utförandet av prestandatesterna var **K6** [9], som är ett open source-testverktyg skapat av Grafana Labs, ett givet val. K6 installeras som en applikation på datorn, men kan även köras i en i Docker-container.

Testerna skrivs i JavaScript och körs lokalt i en projektmiljö i valfri IDE, i detta arbete används **Visual Studio**. Respektive webbplats har sedan en separat projektuppsättning. Under testkörning skickas förfrågningar mot angivna URL:er, och resultaten presenteras direkt i terminalen. Där kan bland annat svarstider, CPU-användning och minneshantering avläsas.

Genom möjligheten att utforma och köra testerna på egen hand tillkommer full kontroll över både testdata och genomförande. Detta säkerställer att den insamlade informationen är relevant för just denna studie. Till skillnad från liknande studier som riskerar att inte vara fullt lika passande och pålitliga relevanta dessa miljöer.

1.5.2.1.1 Testcases

För att få en så bred bild som möjligt av prestandan i de olika miljöerna används fem typer av tester. Som tidigare nämnt i 1.5.1.1.1 **Umbraco och content** så kommer dessa tester initialt att utföras utan någon form av content byggt på respektive test-webbsidor. När den första omgången av tester är klar kommer webbsidorna att fyllas med simpelt content, för att sedan testas på nytt.

- **Baseline Test** (grundläggande prestanda)
- **Load Test** (simulerad trafik)
- **Stress Test** (belastning utöver kapacitet)
- **Spike Test** (plötslig hög belastning)
- **Soak Test** (långvarig belastning)

För att säkerställa rättvis jämförelse kommer samma typer av tester med identiska parametrar, att genomföras mot båda miljöerna.

2. Teori

Flera studier har undersökt hur användare uppfattar svarstider på webbplatser, och var gränsen går mellan acceptabla och oacceptabla fördröjningar. **Jakob Nielsen**, expert inom dator- och webbanvändbarhet [10] föreslår i artikeln *Website Response Times* [11], att svarstider på 0.1 sekunder ger en känsla av omedelbar respons, och svarstider under 1 sekund håller användares tankeprocess obruten. När fördröjningen tar upp till 10 sekunder riskerar däremot användaren att tappa fokus, och allt efter detta leder ofta till att slutanvändaren stänger ner hemsidan helt.

I en branschrapport från **SpeedCurve**, *The Psychology of Site Speed and User Happiness* [12], påpekas det att användare ofta upplever svarstider som upp till 15% längre än vad de faktiskt är. Rapporten betonar vikten av visuell återkoppling under laddning, som exempelvis load-bars, för att minska känslan av väntan. Liknande resultat framkommer i artikeln *The*

Science of Waiting ... and Waiting ... for Your Page to Load [13], där de nämner hur webbutvecklare använder sig av designmönster som placeholders för att skapa illusionen av snabbhet. Dessa visuella element signalerar att innehåll är på väg att laddas, vilket förbättrar den upplevda prestandan trots faktiska fördröjningar.

Sammanfattningsvis tyder dessa studier och tidigare undersökningar på att svarstiden bör hållas under 5 sekunder, men helst under en sekund, för att bevara en positiv användarupplevelse och uppfattningen av ett system som svarar direkt. Samt att både teknisk och visuell optimering spelar en avgörande roll i hur svarstider uppfattas av användaren.

3. Resultat

Som nämnt i 1.5.2.1.1 *Testcases* så delades undersökningen in i fem testscenarier, som genomfördes med hjälp av verktyget **K6** (version 0.57.0). Detta verktyg möjliggör anpassning av testets upplägg i så kallade **stages**, där varje steg definieras utifrån **duration** (tidslängd) och **target** (antal virtuella användare). Denna struktur avgör testets karaktär och belastningsnivå.

Figur 3.1: Exempelkod från K6-script

```
import http from 'k6/http';
import { sleep } from 'k6';

export const options = {
  stages: [
    { duration: '10m', target: 200 },
    { duration: '5h', target: 200 },
    { duration: '10m', target: 0 },
  ],
};

export default function () {
  http.get('https://sannasandbox.euwest01.umbraco.io/');
  sleep(1);
}
```

Samtliga tester utfördes mot mina två plattformar, genom att skicka GET-Requests från testerna mot önskad URL. Det är dock viktigt att notera att det förekom vissa begränsningar i testmiljön i Azure-applikationen. På grund av användandet av ett studentkonto med ett kredittak på 100 USD som nådde sin maxgräns, kunde de tre avslutande testerna med content (*Load*, *Stress* och *Spike*) inte slutföras i Azure-miljön. Därför är Load, Stress och Spike-resultaten endast jämförbara mellan plattformarna utan content.

3.1 Mätvärden

Vid genomförda tester generar K6 ett flertal mätvärden som kan användas för vidare analys av systemets prestanda.

Figur 3.2: Exempel på K6-restresultat i terminalfönstret.

```
data_received.....: 2.5 MB 21 kB/s
data_sent.....: 33 kB 270 B/s
http_req_blocked.....: avg=1.78ms min=0s med=0s max=210.93ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=419.74µs min=0s med=0s max=21.06ms p(90)=0s p(95)=0s
http_req_duration.....: avg=98.96ms min=62.04ms med=89.15ms max=1.17s p(90)=117.64ms p(95)=146.93ms
  { expected_response:true }...: avg=98.96ms min=62.04ms med=89.15ms max=1.17s p(90)=117.64ms p(95)=146.93ms
http_req_failed.....: 0.00% 0 out of 331
http_req_receiving.....: avg=1.79ms min=0s med=773.9µs max=14.63ms p(90)=4.6ms p(95)=7.69ms
http_req_sending.....: avg=347.03µs min=0s med=330.4µs max=3.05ms p(90)=635.4µs p(95)=695.05µs
http_req_tls_handshaking.....: avg=1.26ms min=0s med=0s max=168.5ms p(90)=0s p(95)=0s
http_req_waiting.....: avg=96.82ms min=57.44ms med=86.93ms max=1.17s p(90)=116.36ms p(95)=146.04ms
http_reqs.....: 331 2.73639/s
iteration_duration.....: avg=1.1s min=1.06s med=1.09s max=2.39s p(90)=1.12s p(95)=1.15s
iterations.....: 331 2.73639/s
vus.....: 9 min=1 max=9
vus_max.....: 10 min=10 max=10
```

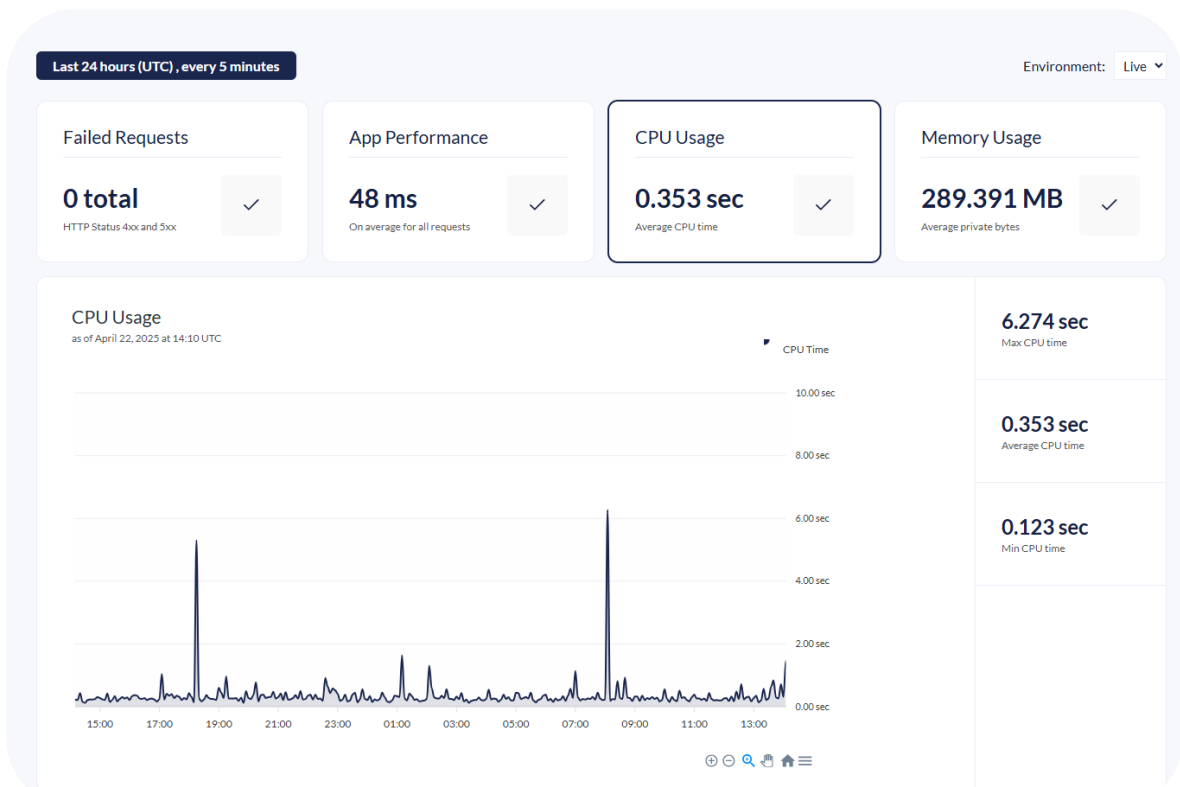
3.1.1 Inbyggd funktionalitet

För att komplettera den data som genererades via K6 användes även plattformarnas inbyggda övervakningsverktyg.

3.1.1.1 Umbraco Insights

Umbraco Cloud tillhandahåller **Application Insights** [14], där det är möjligt att ta del av mätvärden för CPU-användning, minnesförbrukning samt applikationsprestanda.

Figur 3.3: Bild på Umbraco Insights.

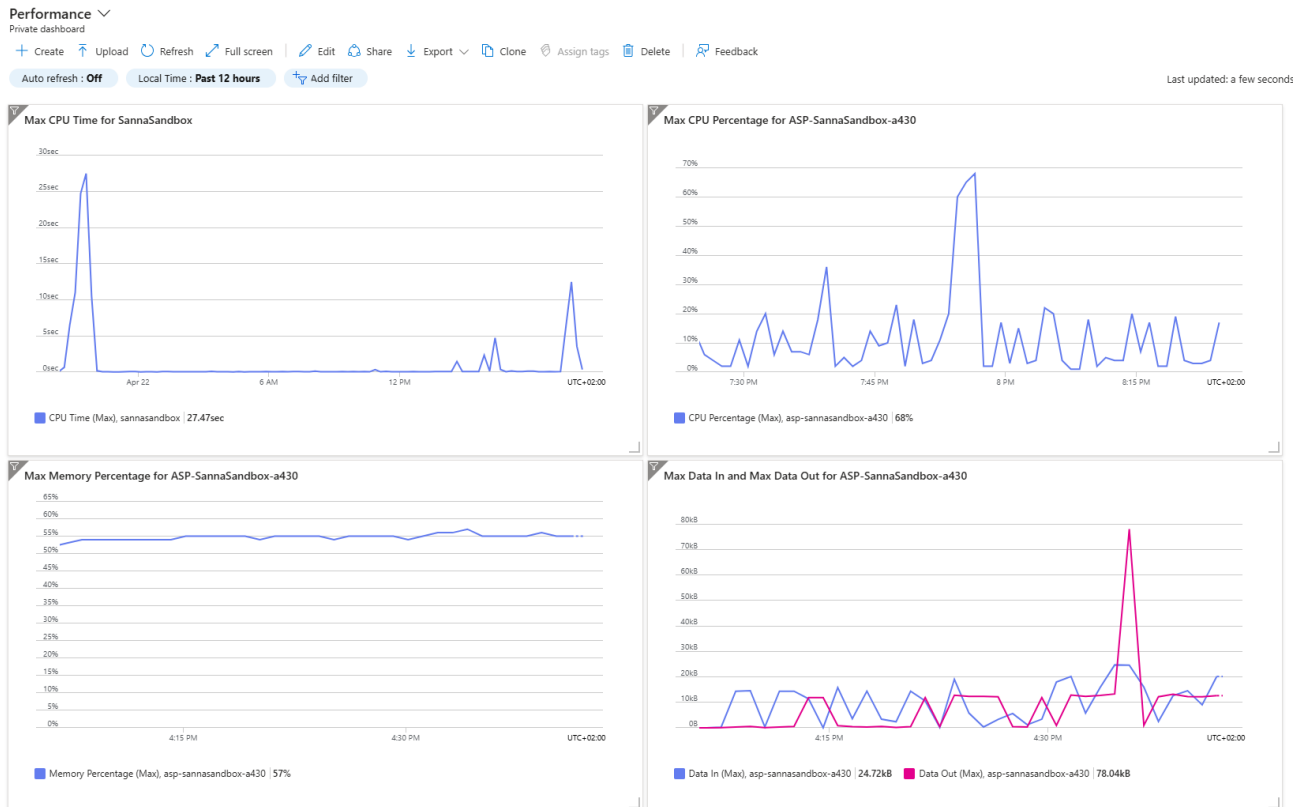


Dessa mätvärden innehåller Max, Min och Avg-värden under tidsintervallet. Granuläriteten för dessa mätpunkter är som lägst 5 minuter.

3.1.1.2 Azure Dashboard

Azure App Service erbjuder användaren möjligheten att skapa anpassade **Application Insight Dashboards** [15] med valfria mätpunkter. I denna dashboard valdes 4 mätpunkter för övervakning: Maximal CPU-användning (%), CPU-tid (millisekunder), minnesförbrukning (%) samt bandbredd in och ut (Megabit/s). Azure har en mätpunkts-granulärhet på en minut.

Figur 3.4: Bild på Azure Dashboard



3.1.2 Avgränsade mätvärden

I denna studie har en avgränsning gjorts till de mätvärden som är mest relevanta för att kunna besvara rapportens syfte och forskningsfrågor.

Ur de resultatvärden som K6 bistår med kommer fokusen att ligga på följande parametrar:

1. **http_req_duration (avg och p95)** - Detta är den totala tiden för en HTTP-förfrågan från start till slut, vilket ger en samlad bild av användarupplevelsen.
2. **http_req_waiting (TTFB – Time To First Byte)** – Mäter hur snabbt servern kan börja svara efter att den fått en förfrågan. Höga tider här tyder ofta på långsam serverbearbetning eller flaskhalsar i backend.
3. **http_req_failed** - Anger hur tillförlitlig tjänsten är. Ett högt värde här betyder att något gått fel, oavsett svarstider.

Utöver dessa värden från K6 har även prestandadata insamlats från plattformarnas egna övervakningsverktyg (**Umbraco Insights & Azure Dashboard**):

1. **CPU-användning** - Visar hur stor andel av serverns processorkraft som förbrukats under testet. Höga värden kan tyda på att servern arbetar nära som kapacitetsgräns.
2. **RAM-användning** – Visar hur mycket arbetsminne som applikationen förbrukat. Om användningen är konstant hög kan det tyda på minnesläckor eller bristande resurshantering.

Eftersom Azure och Umbraco presenterar resursförbrukning i olika format krävdes en konvertering för att kunna sammanställa resultaten i ett gemensamt och jämförbart format. Azure redovisar CPU och RAM-användning i procent, medan Umbraco redovisar CPU-användning i sekunder och minnesförbrukning i megabyte (MB).

För att kunna jämföra RAM-förbrukningen omvandlades Azures procentuella värden till MB genom formeln: $(\text{värdet i procent} / 100) \times 3500$, där 3500 motsvarar det totala tillgängliga RAM-minnet i testmiljön (3,5 GB).

Figur 3.5: Formel för omvandling från procent till MB

$$\left(\frac{Value_{pct}}{100}\right) * 3500$$

Umbracos CPU-användning, som redovisas i sekunder över femminuters-intervaller, omvandlades till procentuell CPU-användning genom formeln: där 5×60 representerar totala antalet sekunder under mätintervallet.

Figur 3.5: Formel för omvandling från sekunder till procent

$$\left(\frac{Value_s}{5 * 60}\right) * 100$$

3.2 Testresultat

Baseline

Syftet med Baseline-testet var att simulera en låg trafiknivå under kort tid, i syfte att skapa en referenspunkt för efterföljande tester. Testet inleddes med en virtuell användare under en minut, följt av tio virtuella användare under ytterligare en minut.

Utan content

Azure miljön uppvisade lägre svarstider och väntetider än Umbraco, med skillnader på över 70% för majoriteten av mätvärdena.

Tabell 3.1: Tabell med testresultat från Baseline-test utan content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	23,88 ms	35,20 ms	22,87 ms	0,00 % (0 av 335)	4 %	1925 MB

UMBRACO	91,34 ms	124,69 ms	89,66 ms	0,00 % (0 av 694)	3,2 %	350 MB
---------	----------	-----------	----------	----------------------	-------	--------

Med content

Vid test med innehåll uppmättes svarstider och väntetider i Azure-miljön som var cirka 75-85% lägre än för Umbraco, trots att RAM-användningen skiljer sig med 1 328 MB mellan lösningarna där Azure har betydligt högre användning.

Tabell 3.2: Tabell med testresultat från Baseline-test med content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	23,88 ms	35,20 ms	22,87 ms	0,00 % (0 av 692)	15 %	1995 MB
UMBRACO	106,61 ms	147,81 ms	99,61 ms	0,00 % (0 av 331)	2,25 %	667 MB

Load

Load-testet utformades för att utvärdera webbplatsernas kapacitet att hantera förväntad normalbelastning över 35 minuter. Under testet gavs ökade belastningen successivt från 0 till 50, 100 och slutligen 200 virtuella användare, med intervaller på 5–10 minuter mellan varje ökning.

Utan content

Vid Load-test uppvisade Azure svarstider som var cirka 70-78% lägre än Umbraco, med en snarlik CPU-användning och en väsentligt högre RAM-belastning.

Tabell 3.3: Tabell med testresultat från Load-test utan content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	26,76 ms	50,56 ms	25,87 ms	0,00 % (0 av 31 2610)	72 %	1995 MB
UMBRACO	90,44 ms	194,78 ms	88,43 ms	0,00 % (2 av 15 1034)	69 %	304 MB

Med content

I Load-testet med content uppvisade Umbraco betydligt högre svarstider jämfört med testet utan content. CPU-belastningen nådde även sin maxgräns vilket tyder på begränsad kapacitet under ökad last för Umbraco.

Tabell 3.4: Tabell med testresultat från Load-test med content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>
UMBRACO	218,44 ms	730,64 ms	212,21 ms	0.00 % (2 av 151 034)	100 %	637 MB

Soak

Soak-testet hade som syfte att mäta systemens stabilitet under en längre tidsperiod. Testet inleddes med en gradvis ökning från 0 till 200 användare under tio minuter. Därefter bibehölls denna belastningsnivå i fem timmar, följt av en nedtrappning tillbaka till 0 användare under tio minuter.

Utan content

Under Soak-testet visade Azure cirka 78% kortare genomsnittliga svarstider och 70% lägre 95-svarstider jämfört med Umbraco, trots att båda kördes med maximal CPU-belastning.

Tabell 3.5: Tabell med testresultat från Soak-test utan content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	39,44 ms	144,17 ms	38,30 ms	0,00 % (0 av 68 90274)	100 %	2065 MB
UMBRACO	180,09 ms	476,66 ms	178,10 ms	0,00 % (21 av 315 0590)	100 %	355 MB

Med content

I Soak-testet med content ökade svarstiderna markant för båda plattformarna jämfört med testet utan content, men Azure behöll ett tydligt prestandaövertag över Umbraco, som dessutom visar en viss instabilitet genom ett antal misslyckade förfrågningar.

Tabell 3.6: Tabell med testresultat från Soak-test med content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	256.23 ms	910.02 ms	253.95 ms	0.00 % (0 av 49 16200)	100 %	2625 MB
UMBRACO	734.28 ms	1.58 s	722.71 ms	0.00% (96 av 2144196)	100 %	490 MB

Spike

Målet med Spike-testet var att undersöka hur plattformarna hanterar plötsliga trafikökningar. Testet inleddes med tio användare under tio sekunder, därefter ökade belastningen snabbt till 500 användare under tio sekunder. 500 användare bevarades under tre minuters tid för att sedan trappas ner till 0 under två minuter.

Utan content

Trots att både Azure och Umbraco nådde maxgränsen för CPU-användningen presterade Azure nästan fyra gånger så snabbt som Umbraco vad gäller genomsnittlig svarstid, och närmre tre gånger snabbare svarstid för 95:e percentilen, vilket tydligt visar på hur de presterar under en plötslig hög belastning.

Tabell 3.7: Tabell med testresultat från Spike-test utan content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	331,83 ms	1270,00 ms	330,94 ms	0,00 % (0 av 147686)	100 %	1960 MB
UMBRACO	1520,00 ms	3260,00 ms	1520,00 ms	0,00 % (4 av 48 638)	100 %	355 MB

Med content

Svarstiderna för Umbraco dubblerades nästan mellan testet utan content och testet med content, vilket indikerar på att förmågan att hantera trafiktappar försämras med content. Dessutom tillkom även en liten andel misslyckade förfrågningar, som inte förekom utan content.

Tabell 3.8: Tabell med testresultat från Spike-test med content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>
UMBRACO	2.78 s	4.89 s	2.77 s	0.04 % (16 av 32 589)	100 %	720 MB

Stress

Stress-testet syftade till att identifiera systemens brytpunkter och flaskhalsar vid hög belastning. Antalet användare ökades gradvis i tio-minuters intervaller från 50 till 100, vidare till 200 → 300 → 400 och slutningen 500 användare. Efter maximal belastning följde en nedtrappning till 0 användare. Testet pågick i totalt 65 minuter.

Utan content

Även under Stress-test kan vi se att Azure resulterade i svarstider som är i snitt tre gånger snabbare än Umbracos, trots full CPU-kapacitet. Detta pekar på att Azure hanterar den gradvis ökade belastningen på ett effektivt sätt.

Tabell 3.8: Tabell med testresultat från Stress-test utan content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	183,55 ms	965,71 ms	182,16 ms	0,00 % (0 av 1248200)	100 %	2240 MB
UMBRACO	626,38 ms	1930,00 ms	623,90 ms	0,00 % (9 av 524838)	100 %	430 MB

Med content

Jämfört med Stress-testet utan content ökade Umbracos genomsnittliga svarstider med mer än det dubbla, och antalet misslyckade requests ökade från 9 till 24. Detta visar en tydlig negativ påverkan på prestandan vid hög belastning när content finns på sidan.

Tabell 3.9: Tabell med testresultat från Stress-test med content.

PLATTFORM	DURATION (AVG)	DURATION (P95)	WAITING	FAILED	CPU (MAX)	RAM (MAX)
AZURE	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>	<i>Data saknas</i>
UMBRACO	1390,00 ms	3550,00 ms	1390,00 ms	0,00 % (24 av 356227)	100 %	828 MB

4. Diskussion

Syftet med arbetet var att undersöka hur valet av hosting påverkar prestandan för Umbraco-webbplatser utifrån olika infallsvinklar. Inför arbetet fanns det inga tydliga förväntningar på vad utfallet skulle bli, mer än att Azure är ett mer etablerat alternativ än Umbraco Cloud. Denna förutsättningslösa utgångspunkt har lett till ett kontinuerligt lärande under arbetets gång.

4.1 Prestandaskillnader mellan Cloud och Azure

Testerna visade tydliga skillnader i prestanda mellan de två hostingalternativen, där Azure visade konsekvent kortare svarstider än Umbraco Cloud i alla tester, där svarstiden i många fall var 3–5 gånger snabbare. Baserat på den forskning som finns kring hur avgörande svarstider är för användarupplevelsen, tyder det på att detta är en faktor som man inte bör bortse från när det kommer till prestanda. Det bekräftar även att val av hostingmiljö påverkar prestandan kraftigt, både med och utan content.

4.2 Resursförbrukning vid trafiktoppar

Vid tester under hög belastning noterades det att båda plattformarna ofta låg på 100% CPU-användning, men RAM-användningen skilde sig däremot markant. Azure förbrukade mellan 3 – 6 gånger mer RAM än Umbraco Cloud. Även om det är svårt att med säkerhet fastställa orsaken till detta, så är en trolig förklaring skillnader i plattformarnas underliggande infrastruktur.

Azure App Service använder exempelvis inbyggda loggnings och övervakningsfunktioner, samt komponenter som IIS (för webbserverhantering), och Kudu (för deployment och diagnostikverktyg). Dessa komponenter bidrar sannolikt till både ett ökat resursuttag och en bättre prestanda.

4.2.1 Svarstider och sidladdningstider under belastning

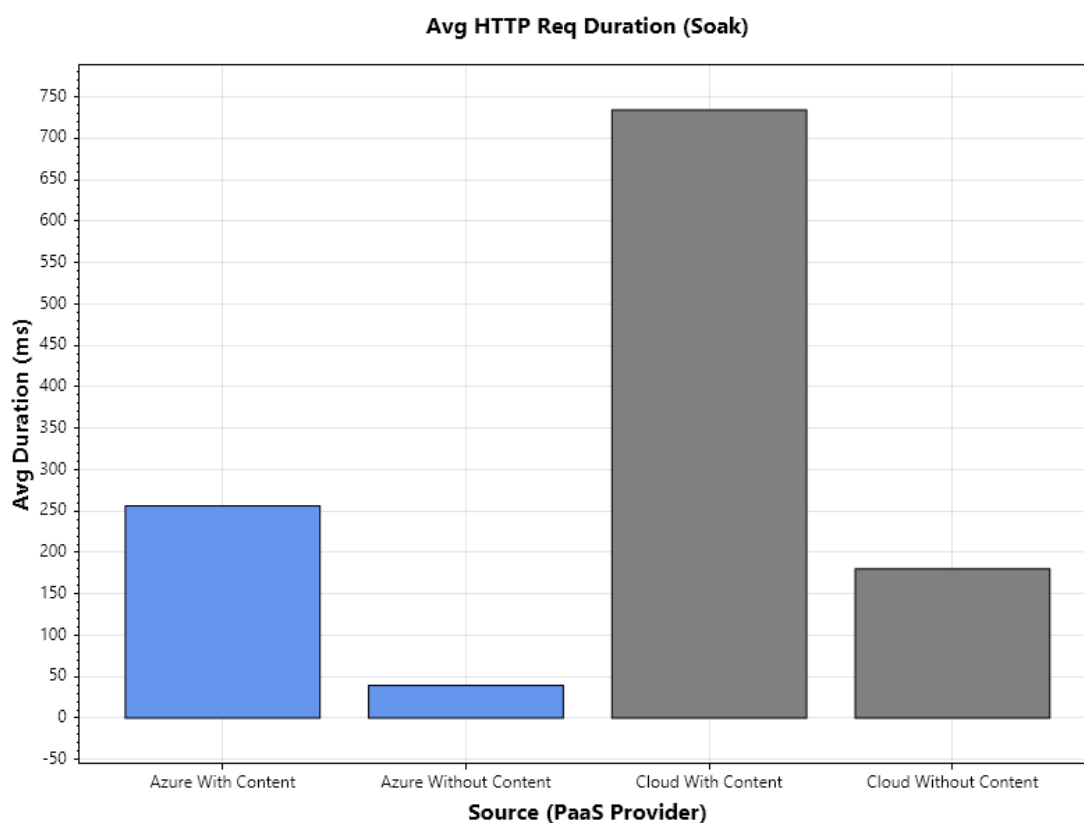
Umbraco Cloud presterade relativt stabilt vid en lägre belastning, men svarstiderna ökade snabbt vid trafiktoppar där resultatet pendlade från några hundra millisekunder till över 3 sekunder i vissa tester.

Azure höll svarstiderna under kontroll betydligt längre och visade generellt en högre tolerans för belastning. Detta tyder på att Azure har en bättre optimering som underlättar för simultana förfrågningar och resursfördelningar.

Tabell 4.1: Tabell med förväntade resultat för mätvärden.

MÄTVÄRDE	BRA	ACCEPTABELT	DÅLIGT
Genomsnittlig svarstid (duration)	< 100 ms	100–300 ms	> 300 ms
95:e percentilen (p95 duration)	< 500 ms	500–1000 ms	> 1000 ms
TTFB (serverrespons)	< 200 ms	200–500 ms	> 500 ms

Figur 4.1: Diagram med resultat från samtliga Soak-tester.



När man tar svarstiderna utifrån de förväntade mätvärdena och tittar på samtliga testresultat för Soak-testet kan vi se att Azure ligger på ungefär 250 ms som högst, vilket klassas som en acceptabel svarstid. Medan Cloud är uppe vid 750 ms som högst, vilket är långt förbi gränsen på vad som är ”underkänt” i genomsnittlig svarstid.

4.2.2 Reflektioner och förbättringar

En viktig insikt från arbetet är betydelsen av att utforma prestandatester utifrån realistiska användningsscenarier. Under testperioden uppstod en begränsning relaterad till otillräcklig övervakning av kreditförbrukningen i Azure. Detta resulterade i att Load, Stress och Spike testerna för Azure-miljön med innehåll inte kunde genomföras enligt plan. Med en större fokus på kreditförbrukningen hade testerna möjligtvis kunnat justerats något för att göra alla tester möjliga utan att ta slut på krediterna.

Trots detta bedöms den insamlade datamängden vara tillräcklig för att möjliggöra en jämförande och rättvis analys av plattformarnas styrkor och svagheter.

5. Slutsatser

Arbetet och rapporten har uppfyllt sitt syfte och besvarat de forskningsfrågor som formulerades i [1.3 Problemformulering](#). Undersökningen visar tydligt att valet av hosting har en betydande påverkan på prestandan för Umbraco-webbplatser. I denna specifika jämförelse

visade sig Azure prestera markant bättre när det gäller skalbarhet och stabilitet. Plattformen erbjöd även mer avancerade möjligheter till övervakning och insikter via sin flexibla dashboard. Detta kom dock till priset av en högre RAM-användning.

Det är viktigt att betona att detta inte innebär att Umbraco Cloud är ett dåligt alternativ, utan det är en fullt tillräcklig och effektiv lösning för många typer av webbplatser. Särskilt med tanke på hur simpelt det är att konfigurera och administrera till skillnad från Azure App Service, som kräver mer vid uppstart och underhåll. Sedan finns det även andra faktorer att ta hänsyn till vid val av plattform, som kostnad, säkerhet och framtida utvecklingsbehov.

Givet de tekniska förutsättningarna i denna undersökning där båda miljöerna var konfigurerade med 3,5 GB RAM och 2 CPU-kärnor, är det inte förvånande att vissa prestandamätningar visade begränsningar. Exempelvis som i de fall där testerna nådde 100% RAM-förbrukning. Det är viktigt att ha i åtanke att denna konfiguration är låg jämfört med vad som vanligtvis används i skarpa, fullskaliga Umbraco-lösningar. I verkliga scenarion skulle en standardwebbplats sällan driftsättas med så pass snäva resurser, om inte förväntad trafik och belastning är väldigt låg.

Därför bör resultatet inte ses som en exakt spegling av hur dessa plattformar skulle prestera i en verklig miljö, utan mer som en indikator på hur skillnaderna i arkitektur och kapabilitet påverkar prestandan. Oavsett skala eller resursnivå kommer det med stor sannolikhet att finnas en viss procentuell skillnad i prestanda mellan lösningarna, eftersom de är olika i hur de är byggda och hanterar sina resurser i grunden.

5.1 Rekommendationer

För utvecklare som står inför valet mellan Umbraco Cloud och Umbraco hostat via Azure bör valet grunda sig i webbplatsens krav på prestanda, belastning och kontroll.

Umbraco Cloud lämpar sig bäst för mindre till medelstora webbplatser där användarvänlighet och enkelhet är prioriterade, medan Azure är att föredra vid höga krav på skalbarhet, stabilitet och teknisk kontroll.

Bilagor

Testdata

I samband med detta examensarbete har ett C#-program tagits fram, som hanterar inläsning och bearbetning av rådata från de K6-prestandatester som utförts. Programmet läser in testresultat i .txt-format, går igenom innehållet och omvandlar det till strukturerad data i .csv-format.

Efter konverteringen analyseras .csv-filerna, där resultat från de olika testmiljöerna jämförs och visualiseras. Jämförelsen presenteras direkt i terminalen, medan grafer genereras automatiskt och sparas sedan i en mapp i projektet.

All insamlad rådata är organiserad i mapparna Azure With/Without Content och Umbraco With/Without Content.

För att ta del av programmet och rå-datan – Besök [K6Comparer](#) [16] på min Github.

Referenslista

- [1] **Umbraco, "Say hello to full flexibility," Umbraco, 07 05 2025. [Online].**
Available: <https://umbraco.com/try-umbraco-cms/>
- [2] **Umbraco, "Your All-in-One Managed Umbraco Solution," Umbraco, 07 05 2025. [Online].**
Available: <https://umbraco.com/products/umbraco-cloud/>
- [3] **Microsoft, "Gör det möjligt med kodning," Microsoft, 07 05 2025. [Online].**
Available: <https://azure.microsoft.com/sv-se>
- [4] **Internetstiftelsen, "Svenskarna och internet 2016 - Sammanfattning," Internetstiftelsen, 2016. [Online].**
Available: <https://svenskarnaochinternet.se/rapporter/svenskarna-och-internet-2016/sammanfattning/>
[Accessed 07 05 2025].
- [5] **Umbraco, "Scalable Solutions, Transparent Pricing," 07 05 2025. [Online].**
Available: <https://umbraco.com/pricing/>
- [6] **Umbraco, "Dedicated Resources," 07 05 2025. [Online].**
Available: <https://umbraco.com/products/umbraco-cloud/dedicated-resources/>
- [7] **Microsoft, "Azure App Service på Windows – prissättning," Microsoft, 07 05 2025. [Online].**
Available: <https://azure.microsoft.com/sv-se/pricing/details/app-service/windows/>
- [8] **Umbraco, "Explore Igloo Widgets," Igloo, 07 05 2025. [Online].**
Available: <https://demo.iglootheme.com/widgets/>
- [9] **Grafana, "What is Grafana k6?," Grafana, 07 05 2025. [Online].**
Available: <https://grafana.com/oss/k6/>
- [10] **Wikipedia, "Jakob Nielsen (datalog)," Wikipedia, 11 12 2019. [Online].**
Available: [https://sv.wikipedia.org/wiki/Jakob_Nielsen_\(datalog\)](https://sv.wikipedia.org/wiki/Jakob_Nielsen_(datalog))
[Accessed 07 05 2025].
- [11] **J. Nielsen, "Website Response Times," NNGroup, 10 06 2010. [Online].**
Available: <https://www.nngroup.com/articles/website-response-times/>.
[Accessed 07 05 2025].
- [12] **SpeedCurve, "The Psychology of Site Speed and User Happiness," SpeedCurve, 31 05 2024. [Online].**
Available: <https://www.speedcurve.com/web-performance-guide/the-psychology-of-web-performance/>
[Accessed 07 05 2025].
- [13] **Bryan Gardiner, "The Science of Waiting ... and Waiting ... for Your Page to Load," Wired, 25 08 2016. [Online].**
Available: <https://www.wired.com/2016/08/science-waiting-waiting-page-load/>
[Accessed 07 05 2025].

- [14] **Umbraco, "Application Insights," Umbraco, 07 05 2025. [Online].**
Available: <https://docs.umbraco.com/umbraco-cloud/set-up/application-insights>
- [15] **Microsoft, "Application Insights Overview dashboard," Microsoft, 12 07 2024. [Online].**
Available: <https://learn.microsoft.com/en-us/azure/azure-monitor/app/overview-dashboard>
[Accessed 07 05 2025].
- [16] **Sanna Wiklund, "K6 Result Comparer, 08 05 2025". [Online].**
Available: <https://github.com/sannawiklund/K6ResultComparer>