



รายละเอียดโครงการ
1101193 โครงการการบริหารระบบ

ชื่อโครงการ
The Spoiler's Edge

รายชื่อสมาชิกในกลุ่ม
B6601621 นางสาวณัฐนิชา พวงอินทร์
B6601829 นางสาวปรียานุช จิตรโคตร
B6602451 นายอพิเชษฐ์ อุตสำห์
B6626587 นายธนดิษฐ์ เกิดมะเร็ง
B6646082 นางสาวชลลปิล เตระมะ

เสนอ
ผู้ช่วยศาสตราจารย์ ดร.ทรงยุทธ เพิ่มผล

สำนักวิทยาศาสตร์และศิลปดิจิทัล
มหาวิทยาลัยเทคโนโลยีสุรนารี

รายละเอียดโครงการ

1101193 โครงการการบริหารระบบ

ชื่อโครงการ

The Spoiler's Edge

รายชื่อสมาชิกในกลุ่ม

B6601621 นางสาวณัฐนิชา พวงอินทร์

B6601829 สาวปริยานุช จิตรโคตร

B6602451 นายอพิเชษฐ์ อุตส่าห์

B6626587 นายธนดิษฐ์ เกิดมะเร็ง

B6646082 นางสาวชลลสปีล เดระมะ

ที่มาและความสำคัญของโครงการ

ปัจจุบันอินเทอร์เน็ตมีบทบาทสำคัญในชีวิตประจำวันของเราอย่างมาก หนึ่งในรูปแบบการใช้งานที่แพร่หลายคือการดูหนังออนไลน์ ซึ่งเป็นการใช้งานอินเทอร์เน็ตเพื่อความบันเทิง การดูหนังผ่านเว็บไซต์ต่างๆ ได้กลายเป็นกิจกรรมที่นิยม เนื่องจากความสะดวกสบายและความหลากหลายของเนื้อหาที่มีให้เลือกชม เทคโนโลยีการพัฒนาเว็บได้พัฒนาไปไกลตามยุคสมัย ทำให้การแนะนำหนังออนไลน์ต้องมีความน่าสนใจดึงดูดผู้คนและมีประสิทธิภาพในการใช้งาน

คณะผู้จัดจึงได้พัฒนาเว็บไซต์สำหรับการแนะนำหนังชื่อว่า The Spoiler's Edge ซึ่งรวบรวมหนังตามประเภทต่างๆ เช่น โรแมนติก ดราม่า ตลก แฟนตาซี ประวัติศาสตร์ เพื่อแนะนำให้ตรงกับความชอบของผู้ใช้งาน นอกจากนั้น คณะผู้จัดทำยังได้พัฒนาระบบการให้คะแนนและรีวิวจากผู้ชมเพื่อช่วยให้ผู้ใช้คนอื่นสามารถตัดสินใจเลือกชมหนังได้ง่ายขึ้น คณะผู้จัดทำยังพัฒนาฟีเจอร์การค้นหาที่สามารถกรองหนังตามปีที่ออกฉาย ผู้กำกับ นักแสดง และยังมีการแนะนำช่องทางการรับชมหนังที่ผู้ใช้งานชอบให้อีกด้วย นอกจากนี้คณะผู้จัดทำยังเพิ่มฟีเจอร์การจัดอันดับหนังยอดนิยมประจำสัปดาห์และเดือน เพื่อให้ผู้ใช้ได้ติดตามแนวโน้มและกระแสนิยมของหนังที่น่าสนใจในช่วงเวลานั้นๆ โดยหวังผลเพื่อให้ผู้ใช้สามารถค้นหาหนังที่ตรงกับความสนใจได้อย่างรวดเร็วและง่ายดาย

ดังนั้น การใช้อินเทอร์เน็ตและเทคโนโลยีในการพัฒนาเว็บแนะนำหนัง จึงเป็นการสร้างประสบการณ์การชมหนังที่มีประสิทธิภาพและความหลากหลาย ช่วยให้ผู้ใช้สามารถเข้าถึงเนื้อหาที่ตรงใจและได้รับความพึงพอใจสูงสุด

วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาเว็บไซต์ ให้แนะนำหนังตามความชอบผู้ใช้งานอย่างมีประสิทธิภาพ
2. เพื่อสร้างระบบให้คะแนนและรีวิวหนังจากผู้ชมเพื่อช่วยในการตัดสินใจเลือกชมหนัง
3. เพื่อเป็นการพัฒนาทักษะในการทำงานเป็นกลุ่ม
4. เพื่อเป็นการพัฒนาทักษะในการออกแบบและพัฒนาเว็บไซต์

ประโยชน์ที่คาดว่าจะได้รับ

1. สร้างทักษะการพัฒนาทักษะในการทำงานเป็นกลุ่ม
2. สร้างทักษะพัฒนาเว็บไซต์ที่สามารถแนะนำหนังอย่างมีประสิทธิภาพ
3. สร้างทักษะในการพัฒนาทักษะในการออกแบบและพัฒนาเว็บไซต์
4. สร้างทักษะในการศึกษาและประยุกต์ใช้เทคโนโลยีใหม่ๆ ในการพัฒนาเว็บไซต์

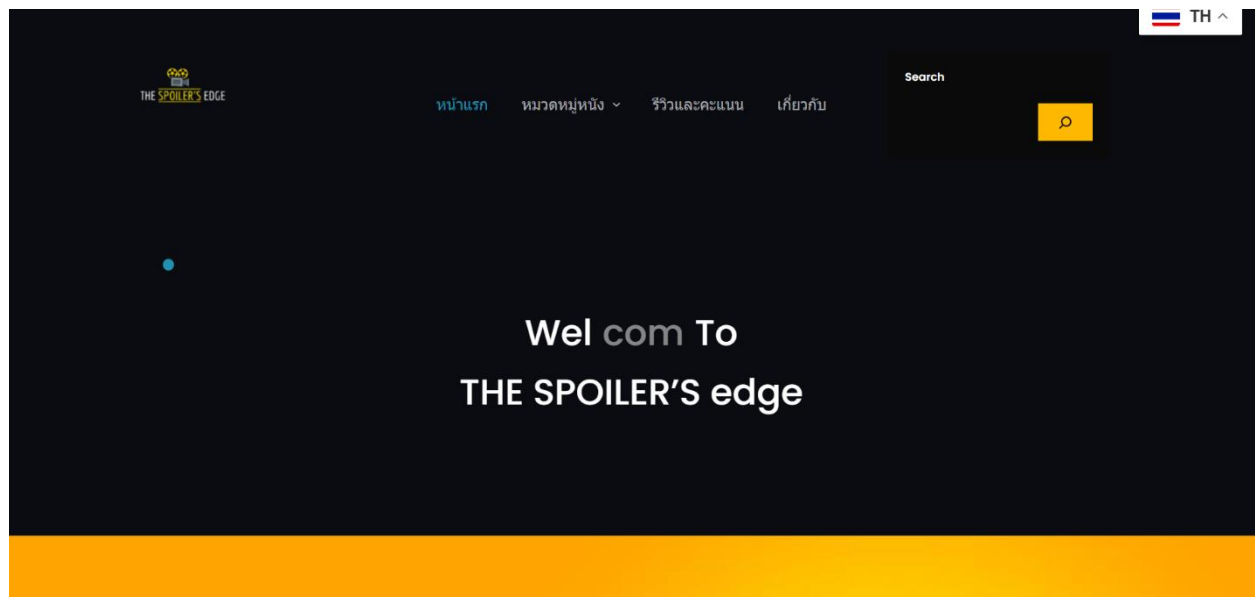
ขอบเขตของโครงการ

1. การรวบรวมข้อมูลหนังตามประเภทต่างๆ : รวบรวมและจัดหมวดหมู่ข้อมูลหนังตามประเภท เช่น โรแมนติก, ดราม่า, ตลก, แฟนตาซี, ประวัติศาสตร์ จากแหล่งข้อมูลที่น่าเชื่อถือ เพื่อใช้ในการแนะนำหนังให้ตรงกับความชอบของผู้ใช้งาน
2. การพัฒนาระบบให้คะแนนและรีวิวหนังจากผู้ชม : สร้างฟีเจอร์ที่ให้ผู้ใช้งานสามารถให้คะแนนและเขียนรีวิวเกี่ยวกับหนังที่เคยชม โดยพัฒนาเว็บไซต์จากการใช้ Wordpress และจัดการฐานข้อมูลด้วย MySQL โดย phpMyAdmin นอกจากนี้ยังมี Docker Desktop ช่วยให้นักพัฒนาสามารถสร้าง ทดสอบ และรันแอปพลิเคชันได้อย่างง่ายดาย
3. การพัฒนาฟีเจอร์การค้นหาที่สามารถกรองหนังตามเกณฑ์ที่กำหนด : พัฒนาระบบการค้นหาที่สามารถกรองหนังตามปีที่ออกฉาย, ผู้กำกับ, นักแสดง และประเภท ด้วยการใช้ Plug in ต่างใน Wordpress เข้ามาช่วยในการพัฒนาเว็บไซต์
4. การแนะนำช่องทางการรับชมหนัง : เพิ่มฟีเจอร์ที่แนะนำช่องทางการรับชมหนัง เช่น เว็บไซต์สตรีมมิ่ง, โรงภาพยนตร์, และแพลตฟอร์มอื่นๆ ที่เหมาะสมในการดึงข้อมูลและแสดงผลในเว็บไซต์
5. การพัฒนาฟีเจอร์การจัดอันดับหนังยอดนิยม : สร้างระบบการจัดอันดับหนังยอดนิยมประจำสัปดาห์และเดือน โดยใช้เทคนิคการวิเคราะห์ข้อมูลจากการให้คะแนนและรีวิวของผู้ใช้ เพื่อให้ผู้ใช้ติดตามแนวโน้มและกระแสนิยมของหนังในช่วงเวลานั้น

เครื่องมือที่นำมาใช้

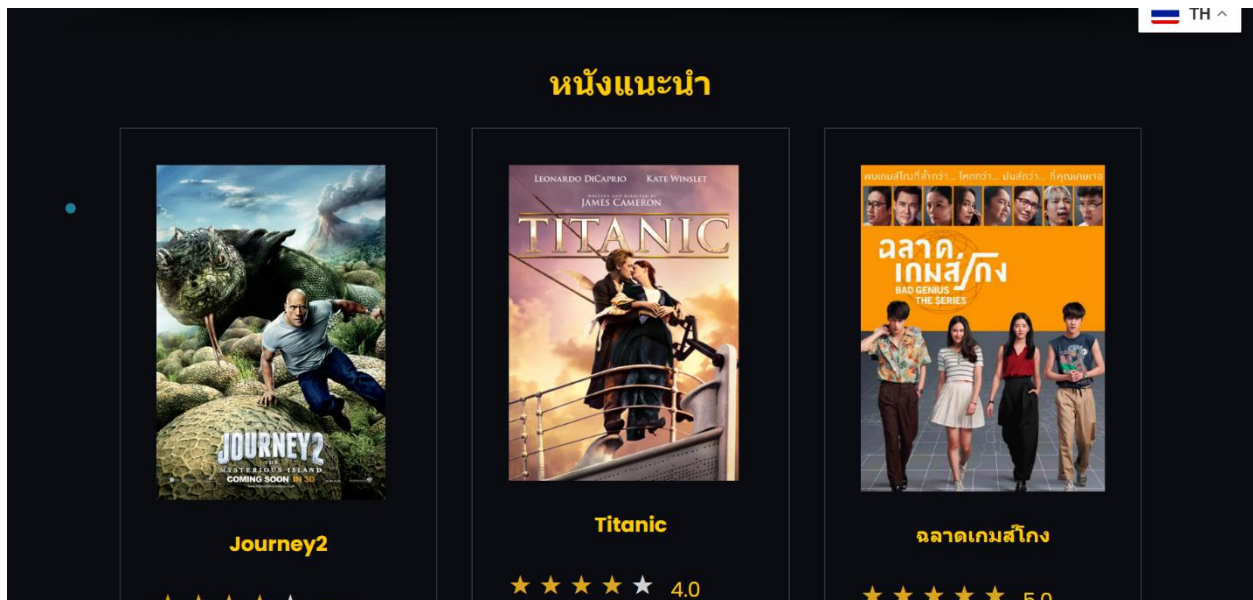
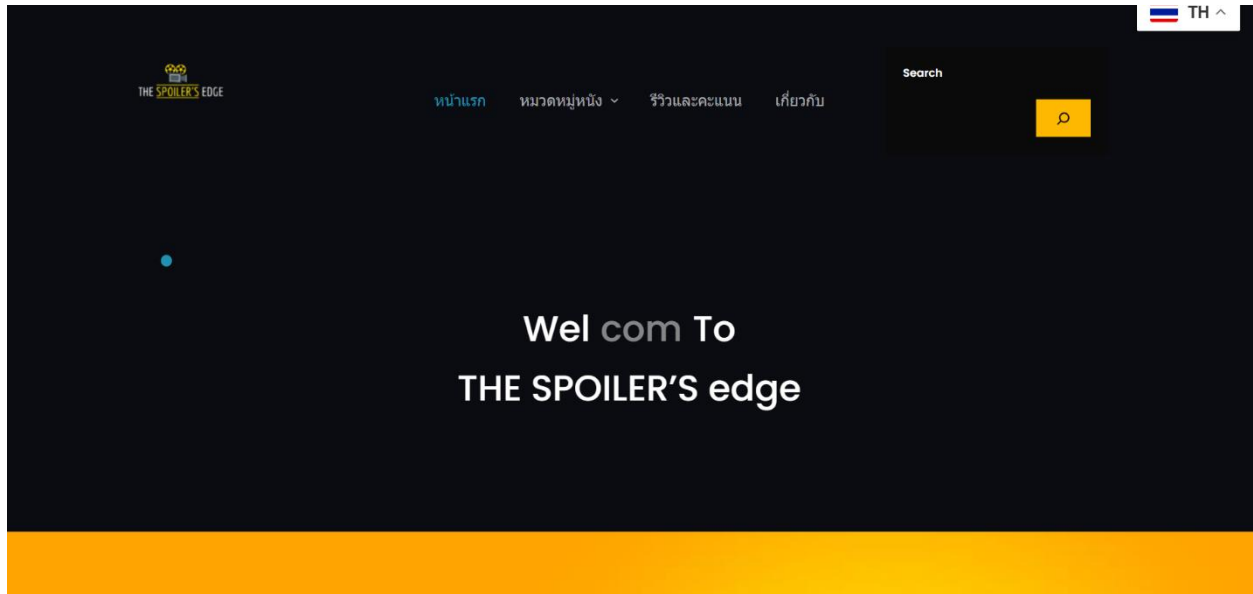
1. Docker Desktop เป็นแพลตฟอร์มสำหรับการคอนเทนเนอร์ไรเซชัน ที่ช่วยให้นักพัฒนาสามารถสร้าง ทดสอบ และรันแอปพลิเคชันได้อย่างง่ายดาย โดยใช้คอนเทนเนอร์เพื่อเพิ่มความยืดหยุ่นและ ประสิทธิภาพในการพัฒนา
2. WordPress เป็นระบบจัดการเนื้อหา (CMS) ที่ได้รับความนิยมอย่างแพร่หลาย ช่วยในการออกแบบ และพัฒนาเว็บไซต์อย่างมีประสิทธิภาพ พร้อมทั้งมีชุมชนนักพัฒนาและผู้ใช้งานที่คอยสร้างสรรค์และ พัฒนาเพิ่มเติมอย่างต่อเนื่อง
3. phpMyAdmin เป็นเครื่องมือที่ช่วยในการจัดการฐานข้อมูล MySQL ผ่านทางเว็บเบราว์เซอร์ โดยให้ ความสะดวกในการใช้งานแก่ผู้ดูแลและผู้พัฒนาระบบ สามารถทำการจัดการฐานข้อมูลได้หลากหลาย เช่น การสร้าง การแก้ไข และการลบตารางฐานข้อมูล รวมถึงการสำรองและกู้คืนข้อมูล เป็นต้น
4. Mysql ใช้ในการจัดระบบฐานข้อมูลที่สำคัญที่สำคัญในการเก็บรักษาและจัดการข้อมูลของแอปพลิเคชัน และมีหน้าที่สำคัญในการรองรับและบริหารจัดการข้อมูลในแอปพลิเคชันและระบบต่างๆ
5. Visual Studio Code ใช้เป็นเครื่องมือที่สำคัญในการพัฒนาซอฟต์แวร์ที่ช่วยให้สามารถเขียนแก้ไข เว็บไซต์ได้อย่างมีประสิทธิภาพและมีความสะดวกสบาย
6. Kubernetes ใช้เป็นระบบคอนเทนเนอร์ที่มีหน้าที่สำคัญในการจัดการและควบคุมคอนเทนเนอร์และ แอปพลิเคชันในสถานการณ์การรันแบบมาโหนดและจัดการแหล่งทรัพยากรต่างๆ ในองค์กรและควบคุม กระบวนการการเชื่อมต่อและการทำงานของแอปพลิเคชันอย่างมีประสิทธิภาพ

รายละเอียดตัวเว็บ Wordpress

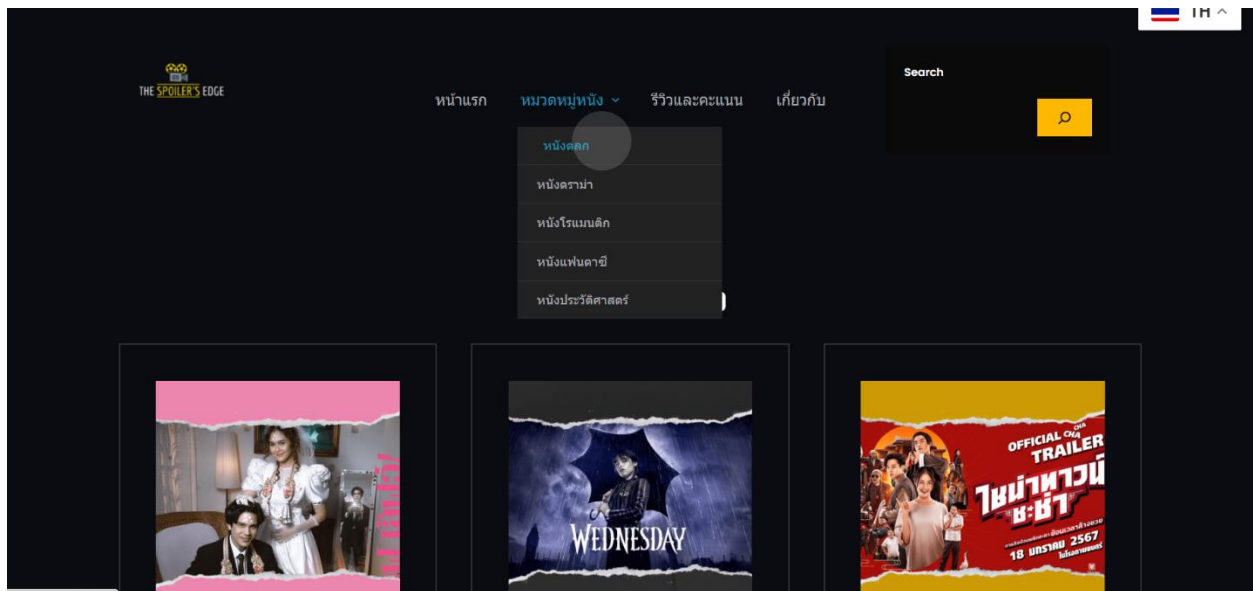


หลักการทำงานของตัวเว็บ จะมีส่วนประกอบดังนี้

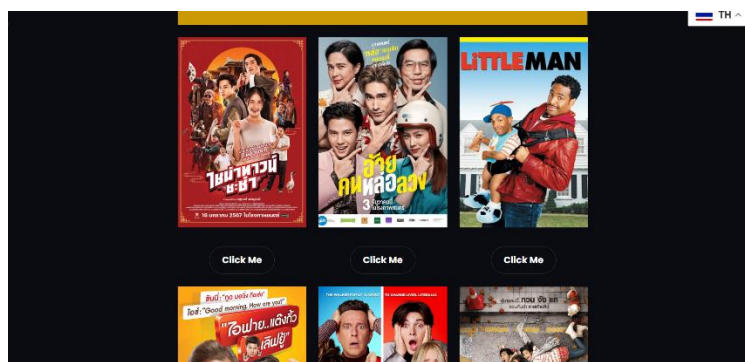
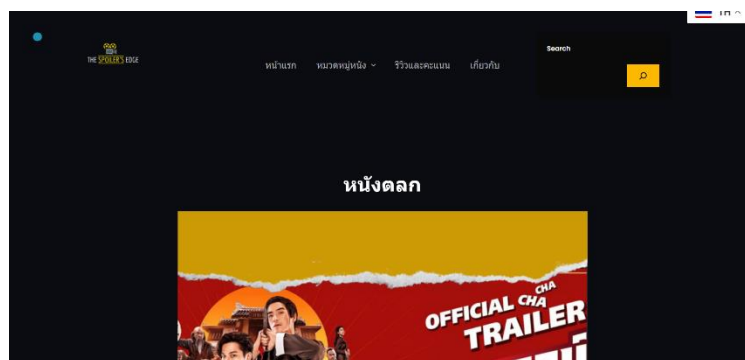
- หน้า home คือหน้าแรกที่แสดงหนังแนะนำของตัวเว็บ และสามารถค้นหาหนัง หรือประเภทหนังที่สนใจได้



- หน้าหมวดหมู่หนัง คือหน้าที่สามารถเลือกหมวดหมู่หนังตามที่เราสนใจได้

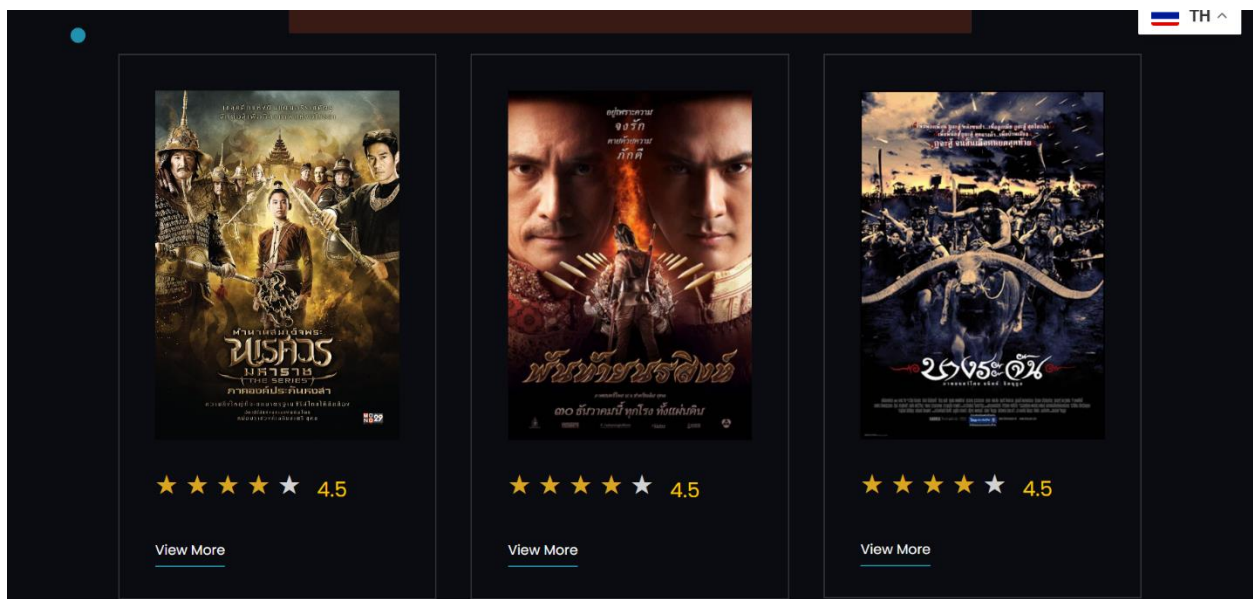
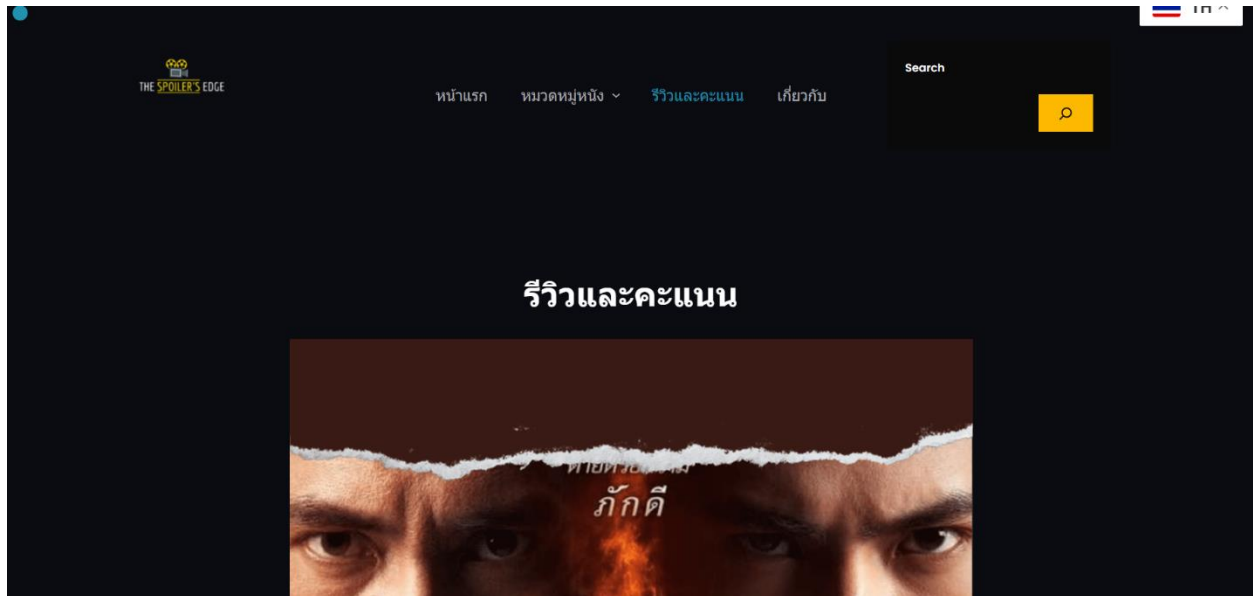


ตัวอย่าง หมวดหมู่หนังตลก

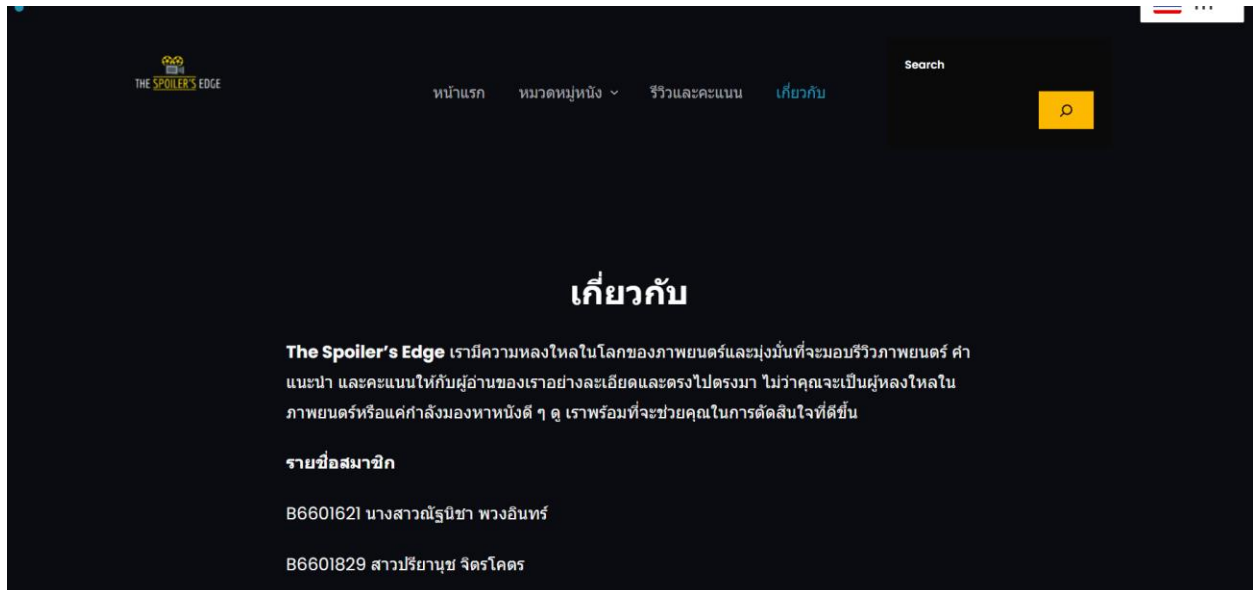


- หน้ารีวิวและคะแนน คือหน้าที่ใช้สำหรับรีวิว ใส่ดาวว่าหนังแต่ละเรื่องเป็นที่นิยมมากน้อยแค่ไหน และสามารถดูรายละเอียดต่างๆของหนังแต่ละเรื่องได้ โดยกดคำว่า view more

ตัวอย่างเช่น รีวิวและคะแนนของหมวดหมู่หนังประวัติศาสตร์



- หน้าเกี่ยวกับ คือหน้าที่แสดงชื่อของคณะผู้จัดทำ รวมถึงเนื้อหาเล็กน้อยเกี่ยวกับตัวเว็บ The Spoiler's Edge



Plug in ที่ใช้

ทั้งหมด (5) กำลังใช้งาน (3) ไม่ได้อใช้งาน (2) ปลดใช้งานอัตโนมัติ (0)			Search installed plugins
<div>คำสั่งค้นหา</div> <div>นำไปใช้</div>			5 รายการ
<input type="checkbox"/>	ปลั๊กอิน	คำอธิบาย	อัปเดตอัตโนมัติ
<input type="checkbox"/>	Akismet Anti-spam: Spam Protection Activate ลบ	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam . Akismet Anti-spam keeps your site protected even while you sleep. To get started: activate the Akismet plugin and then go to your Akismet Settings page to set up your API key. รุ่น 5.3.3 โดย Automattic - Anti-spam Team ดูรายละเอียด	เปิดใช้งานอัปเดตอัตโนมัติ
<input type="checkbox"/>	GTranslate เปิดใช้งาน Settings	Translate your website and make it multilingual. For support visit GTranslate Support Forum . รุ่น 3.0.6 โดย Translate AI Multilingual Solutions ดูรายละเอียด	เปิดใช้งานอัปเดตอัตโนมัติ
<input type="checkbox"/>	Hello Dolly Activate ลบ	นี่ไม่ใช่เป็นเพียงปลั๊กอิน แต่มันแสดงถึงความหวังและความศรัทธาของคนทั้งรุ่น ซึ่งถูกร่องในคำเพียงสองคำที่มีชื่อเสียงมากที่สุดโดยหลุยส์ อารัม สตรอง: Hello, Dolly. เมื่อเปิดการใช้งานคุณจะได้เห็นเนื้อเพลงที่สุ่มมาจากเพลง Hello, Dolly ที่มขหวานหน้าจอกับผู้ควบคุมของคุณในทุก ๆ หน้า รุ่น 1.7.2 โดย Matt Mullenweg ดูรายละเอียด	เปิดใช้งานอัปเดตอัตโนมัติ
<input type="checkbox"/>	Site Reviews เปิดใช้งาน Settings Help About	Receive and display reviews on your website รุ่น 7.0.18 โดย Paul Ryley ดูรายละเอียด	เปิดใช้งานอัปเดตอัตโนมัติ
<input type="checkbox"/>	Smart Slider 3 เปิดใช้งาน Analyze & Repair Go Pro	The perfect all-in-one responsive slider solution for WordPress. รุ่น 3.5.1.23 โดย Nextend ดูรายละเอียด	เปิดใช้งานอัปเดตอัตโนมัติ
<input type="checkbox"/>	ปลั๊กอิน	คำอธิบาย	อัปเดตอัตโนมัติ
<div>คำสั่งค้นหา</div> <div>นำไปใช้</div>			5 รายการ

หน้าของ phpMyAdmin

phpMyAdmin interface showing the Structure tab for the `wordpress_db` database. The table list includes:

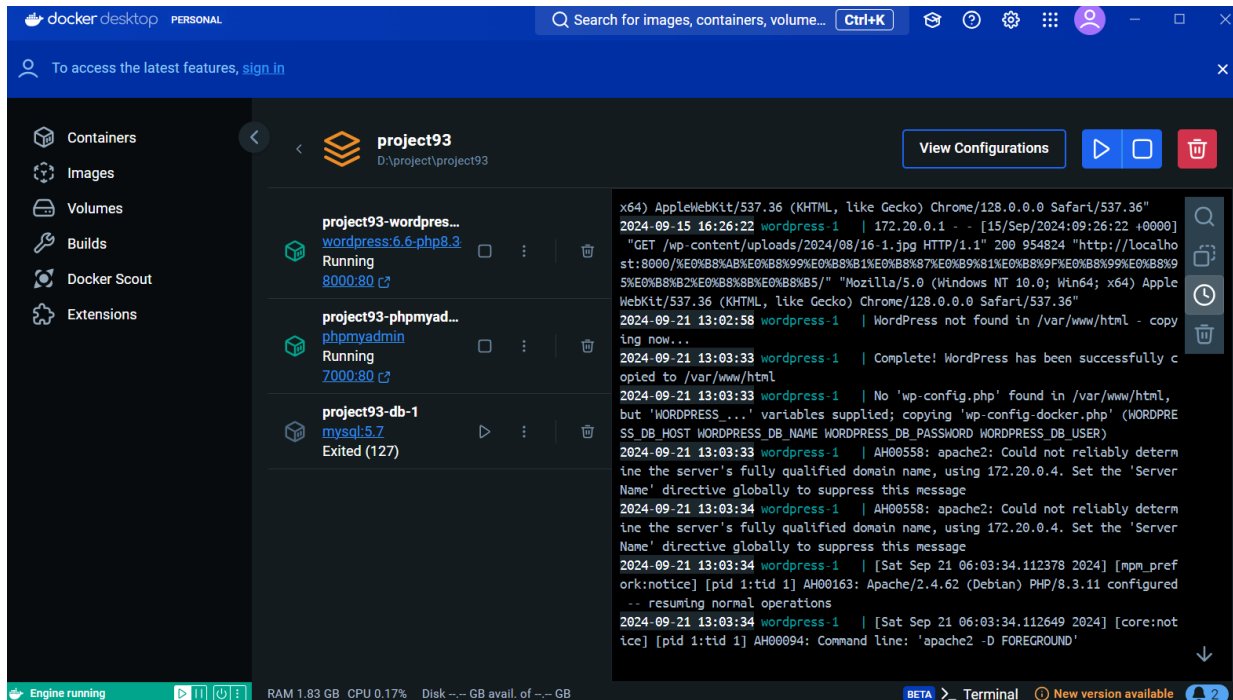
Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> wp_actionscheduler_actions	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_520_ci	128.0 KiB	-
<input type="checkbox"/> wp_actionscheduler_claims	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_actionscheduler_groups	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_actionscheduler_logs	★ Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_commentmeta	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_comments	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_520_ci	96.0 KiB	-
<input type="checkbox"/> wp_e_events	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_glsr_assigned_posts	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_glsr_assigned_terms	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_glsr_assigned_users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_glsr_ratings	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_links	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_nextend2_image_storage	★ Browse Structure Search Insert Empty Drop	63	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_nextend2_section_storage	★ Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_unicode_520_ci	144.0 KiB	-
<input type="checkbox"/> wp_nextend2_smartslider3_generators	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	16.0 KiB	-
<input type="checkbox"/> wp_nextend2_smartslider3_sliders	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_520_ci	112.0 KiB	-
<input type="checkbox"/> wp_nextend2_smartslider3_sliders_xref	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_nextend2_smartslider3_slides	★ Browse Structure Search Insert Empty Drop	25	InnoDB	utf8mb4_unicode_520_ci	160.0 KiB	-
<input type="checkbox"/> wp_options	★ Browse Structure Search Insert Empty Drop	310	InnoDB	utf8mb4_unicode_520_ci	2.5 MiB	-
<input type="checkbox"/> wp_postmeta	★ Browse Structure Search Insert Empty Drop	3,363	InnoDB	utf8mb4_unicode_520_ci	1.8 MiB	-

Continuation of the phpMyAdmin interface showing the Structure tab for the `wordpress_db` database. The table list continues with:

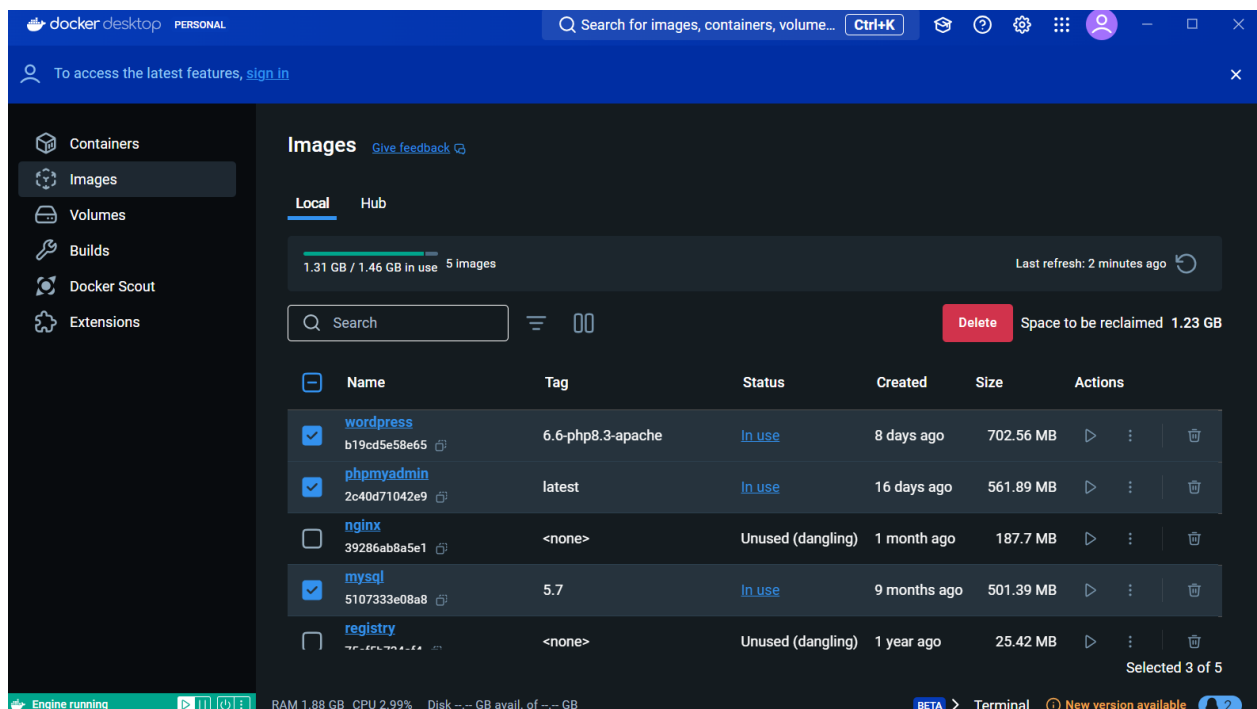
<input type="checkbox"/> wp_nextend2_smartslider3_generators	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	16.0 KiB	-
<input type="checkbox"/> wp_nextend2_smartslider3_sliders	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_520_ci	112.0 KiB	-
<input type="checkbox"/> wp_nextend2_smartslider3_sliders_xref	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_nextend2_smartslider3_slides	★ Browse Structure Search Insert Empty Drop	25	InnoDB	utf8mb4_unicode_520_ci	160.0 KiB	-
<input type="checkbox"/> wp_options	★ Browse Structure Search Insert Empty Drop	310	InnoDB	utf8mb4_unicode_520_ci	2.5 MiB	-
<input type="checkbox"/> wp_postmeta	★ Browse Structure Search Insert Empty Drop	3,363	InnoDB	utf8mb4_unicode_520_ci	1.8 MiB	-
<input type="checkbox"/> wp_posts	★ Browse Structure Search Insert Empty Drop	1,420	InnoDB	utf8mb4_unicode_520_ci	6.8 MiB	-
<input type="checkbox"/> wp_termmeta	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_terms	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_term_relationships	★ Browse Structure Search Insert Empty Drop	35	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_term_taxonomy	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_trp_dictionary_th_en_us	★ Browse Structure Search Insert Empty Drop	152	InnoDB	utf8mb4_unicode_520_ci	112.0 KiB	-
<input type="checkbox"/> wp_trp_gettext_en_us	★ Browse Structure Search Insert Empty Drop	27	InnoDB	utf8mb4_unicode_520_ci	64.0 KiB	-
<input type="checkbox"/> wp_trp_gettext_original_meta	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci	64.0 KiB	-
<input type="checkbox"/> wp_trp_gettext_original_strings	★ Browse Structure Search Insert Empty Drop	27	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_trp_gettext_th	★ Browse Structure Search Insert Empty Drop	27	InnoDB	utf8mb4_unicode_520_ci	64.0 KiB	-
<input type="checkbox"/> wp_trp_original_meta	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_520_ci	64.0 KiB	-
<input type="checkbox"/> wp_trp_original_strings	★ Browse Structure Search Insert Empty Drop	152	InnoDB	utf8mb4_unicode_520_ci	64.0 KiB	-
<input type="checkbox"/> wp_usermeta	★ Browse Structure Search Insert Empty Drop	27	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_users	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_520_ci	64.0 KiB	-
34 tables		Sum			5,689 InnoDB latin1_swedish_ci	12.9 MiB 0 B

การทำงานของ Docker Desktop

- ส่วนของ Container



- ส่วนของ image



การทำงานของ Docker Compose

```
1  services:
2    db:
3      image: mysql:5.7
4      restart: always
5      networks:
6        - web_network
7      volumes:
8        - ./mysql:/var/lib/mysql
9        - ./config/uploads.ini:/usr/local/etc/php/conf.d/uploads.ini
10     environment:
11       - MYSQL_ROOT_PASSWORD=1234
12       - MYSQL_DATABASE=wordpress_db
13       - MYSQL_USER=wordpress
14       - MYSQL_PASSWORD=wordpress
15     wordpress:
16       depends_on:
17         - db
18       image: wordpress:6.6-php8.3-apache
19       restart: always
20       networks:
21         - web_network
22       volumes:
23         - ./wordpress:/var/www/html
24       ports:
25         - 8000:80
26       environment:
27         - WORDPRESS_DB_HOST=db:3306
28         - WORDPRESS_DB_USER=wordpress
29         - WORDPRESS_DB_PASSWORD=wordpress
30         - WORDPRESS_DB_NAME=wordpress_db
31     phpmyadmin:
32       depends_on:
33         - db
34       image: phpmyadmin
35       restart: always
36       networks:
37         - web_network
38       environment:
39         - PMA_ARBITRARY=1
40         - PMA_HOST=db
41         - PMA_USER=root
42         - PMA_PASSWORD=1234
43         - MYSQL_ROOT_PASSWORD=1234
44         - UPLOAD_LIMIT=500M
45         - PMA_UPLOAD_SIZE=500M
46         - PMA_MAX_FILE_SIZE=500M
47       ports:
48         - 7000:80
49
50     networks:
51       web_network:
52         external: true
```

MySQL Service:

- Image: mysql:5.7
- เริ่มต้น container อัตโนมัติเสมอ (restart: always)
- ใช้ volume เพื่อเก็บข้อมูลฐานข้อมูล MySQL ในโฟลเดอร์ ./mysql:/var/lib/mysql
- กำหนดค่าตัวแปรสภาพแวดล้อม:
 - รหัสผ่าน root: MYSQL_ROOT_PASSWORD=1234
 - ชื่อฐานข้อมูล: MYSQL_DATABASE=wordpress_db
 - ชื่อผู้ใช้: MYSQL_USER=wordpress
 - รหัสผ่านของผู้ใช้: MYSQL_PASSWORD=wordpress

WordPress Service:

- ขึ้นอยู่กับ MySQL service (depends_on)
- Image: wordpress:6.6-php8.3-apache
- เริ่มต้น container อัตโนมัติเสมอ (restart: always)
- ใช้ volume เพื่อเก็บข้อมูล WordPress ในโฟลเดอร์ ./wordpress:/var/www/html
- เปิดพอร์ต 8800:80 เพื่อเข้าถึง WordPress
- กำหนดค่าตัวแปรสภาพแวดล้อม:
 - โฮสต์ฐานข้อมูล: WORDPRESS_DB_HOST=db:3306
 - ผู้ใช้ฐานข้อมูล: WORDPRESS_DB_USER=wordpress
 - รหัสผ่านฐานข้อมูล: WORDPRESS_DB_PASSWORD=wordpress
 - ชื่อฐานข้อมูล: WORDPRESS_DB_NAME=wordpress_db

phpMyAdmin Service:

- ขึ้นอยู่กับ MySQL service (depends_on)
- Image: phpmyadmin
- เริ่มต้น container อัตโนมัติเสมอ (restart: always)
- ใช้เครือข่ายเดียวกับ WordPress และ MySQL (networks: web_network)
- กำหนดค่าตัวแปรสภาพแวดล้อม:
 - เปิดใช้งาน phpMyAdmin: PMA_ARBITRARY=1
 - ชื่อโฮสต์ของฐานข้อมูล: PMA_HOST=db

- ชื่อผู้ใช้: PMA_USER=root
 - รหัสผ่านของ root: PMA_PASSWORD=1234
 - กำหนดขนาดไฟล์อัปโหลดสูงสุด: UPLOAD_LIMIT=500M, PMA_UPLOAD_SIZE=500M, PMA_MAX_FILE_SIZE=500M
- เปิดพอร์ต 7770:80 เพื่อเข้าถึง phpMyAdmin

Network:

- networks: ระบุเครือข่ายที่บริการใน Docker Compose จะใช้
- web_network: ชื่อของเครือข่ายที่กำหนดไว้
- external: true: บอกว่าเครือข่ายนี้เป็นเครือข่ายภายนอกที่ถูกสร้างขึ้นนอก Docker Compose และจะไม่ถูกสร้างหรือจัดการโดย Docker Compose เอง

การใช้ Docker Swarm

- จะใช้ Code อันเดียวกันกับ Docker Compose
- ใช้คำสั่งเพื่อทำการกำหนดให้เครื่องเป็น Manager Node ของ Swarm Cluster

```
PS D:\project\project93> docker swarm init
Swarm initialized: current node (mrpoahkcc17ucz11ur1fbj41v) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3go2iewmuqgg2h12tj3ebelxm5768ulcnmkrrn70abis3wq3la-d22bg3tgt9hrf8f85j8vrwfn3 192.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

PS D:\project\project93> █
```

- จากนั้นใช้คำสั่งนี้เพื่อสร้าง token สำหรับให้ worker node เข้าร่วมกลุ่ม

```
PS D:\project\project93> docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3go2iewmuqgg2h12tj3ebelxm5768ulcnmkrrn70abis3wq3la-d22bg3tgt9hrf8f85j8vrwfn3 192.168.65.3:2377

PS D:\project\project93> █
```

- ใช้คำสั่งนี้เพื่อสร้าง token สำหรับให้ manager node เข้าร่วมกลุ่ม

```
PS D:\project\project93> docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3go2iewmuqgg2h12tj3ebelxm5768ulcnmkrrn70abis3wq3la-72z60g59rofc0nx6jefef0upbp 192.168.65.3:2377

PS D:\project\project93> █
```

- จากนั้นใช้คำสั่งเพื่อดูรายละเอียดของ swarm

```
PS D:\project\project93> docker node ls
ID                HOSTNAME          STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
mrpoahkcc17ucz11ur1fbj41v * docker-desktop    Ready     Active           Leader             27.1.1
PS D:\project\project93> █
```

การสร้าง Swarm










```
PS D:\project\project93\test> docker stack deploy --compose-file docker-compose.yml project93
Ignoring unsupported options: restart

Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating service project93_phpmyadmin
Creating service project93_db
Creating service project93_wordpress
PS D:\project\project93\test> docker service ls
```

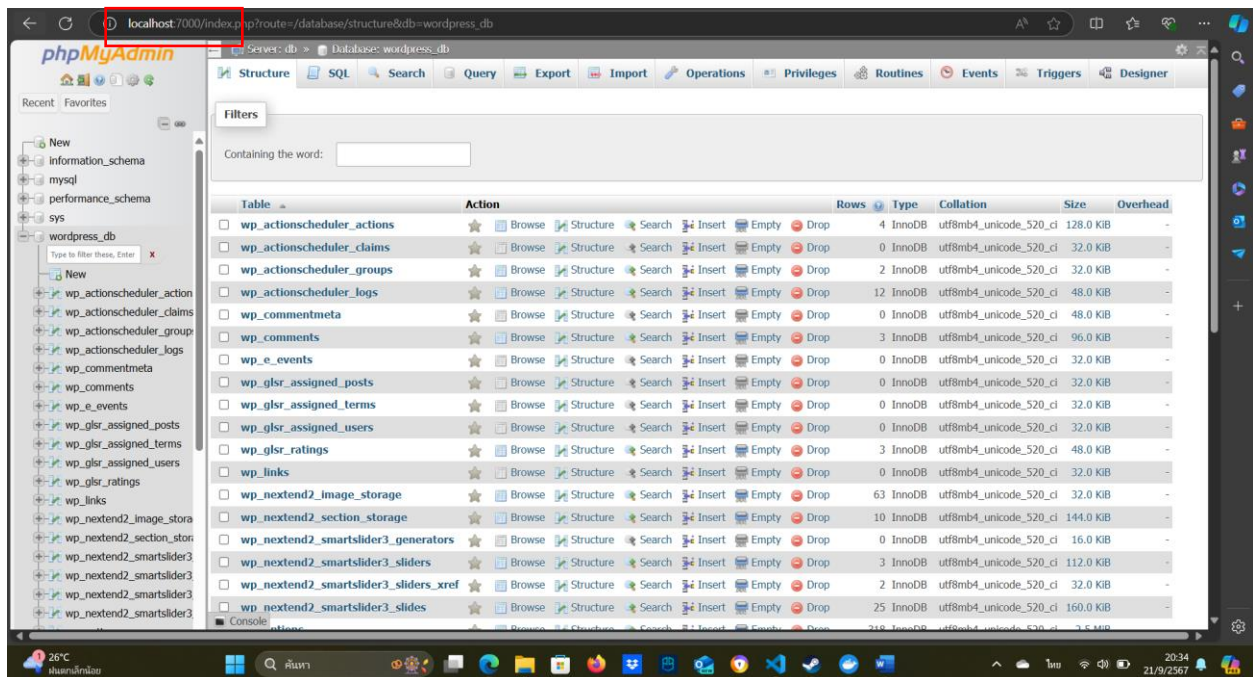
- ทำการสร้างและจัดการบริการ (services) ที่กำหนดไว้ในไฟล์ docker-compose.yml ของคุณในสแตกที่ชื่อว่า project93 โดยคำสั่งนี้มีการทำงานดังนี้:
- Creating service project93_phpmyadmin: บริการ (service) ที่ชื่อว่า project93_phpmyadmin ซึ่งน่าจะเป็นบริการสำหรับ phpMyAdmin
- Creating service project93_db: กำลังสร้างบริการที่ชื่อว่า project93_db บริการฐานข้อมูล (เช่น MySQL หรือ MariaDB)
- Creating service project93_wordpress: กำลังสร้างบริการที่ชื่อว่า project93_wordpress เป็นบริการของ WordPress
- แสดงรายละเอียดของ Service

```
PS D:\project\project93\test> docker service ls
ID                NAME                MODE                REPLICAS    IMAGE                                  PORTS
ntcwbcvk364p      project93_db         replicated          1/1          mysql:5.7                             *:7770->80/tcp
2jxo6ulqn6s4      project93_phpmyadmin replicated          1/1          phpmyadmin:latest                    *:7770->80/tcp
kvyra5rjlrnx      project93_wordpress  replicated          1/1          wordpress:6.6-php8.3-apache          *:8800->80/tcp
PS D:\project\project93\test>
```

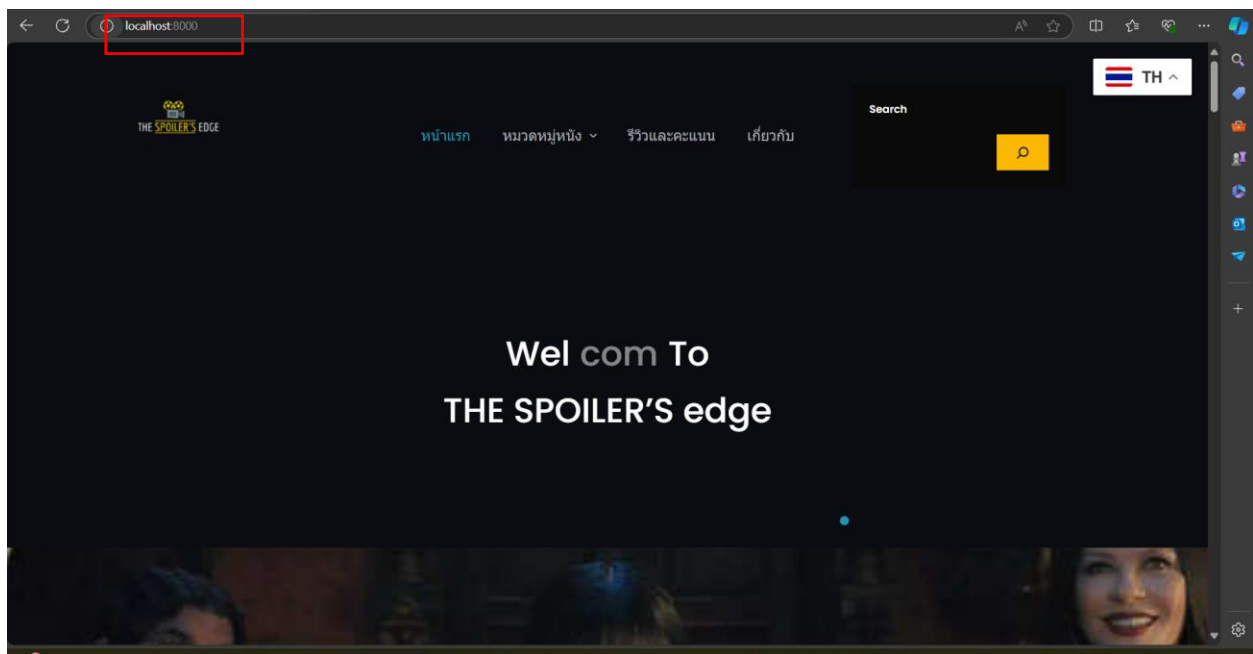
```
Individual Containers
> wordpress:6.6-php8.3-apache@sha256:12c9e34503ef8c6629683c0732a29ee7b46161aaf0aec98978f3770aac8d52f9 project93_wordpress.126ev9zd5um2z6h5dij4uho6iu - Up 40 minutes
> mysql:5.7@sha256:4bc6bc963e6d8443453676cae56536f4b8156d78bae03c0145cbe47c2aad73bb project93_db.1k3dr722m5f8gt76ok88xf0nqj - Up 40 minutes
> phpmyadmin:latest@sha256:8979d7219511e94fa8fc88afdb45c62392c6e09ae9b28bce97a4bb7a078bebb5 project93_phpmyadmin.1.wp4ilj2sgoawkq7dx3m6r3ky3 - Up 40 minutes
```

<input type="checkbox"/>	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	 project93_f3338c51171	phpmyadmin:late	Running		0%	41 minutes ago	<input type="checkbox"/>  
<input type="checkbox"/>	 project93_6b5ece82b1	mysql:5.7	Running		0%	41 minutes ago	<input type="checkbox"/>  
<input type="checkbox"/>	 project93_f7863a5f641	wordpress:6.6-php8.3-apache	Running		0%	41 minutes ago	<input type="checkbox"/>  

หน้า phpMyAdmin <http://localhost:7000/>



หน้า Wordpress <http://localhost:8000/>



การทำงานของ Kubernetes

1. mysql-deployment.yml

1.1. PersistentVolume (PV):

- สร้าง PersistentVolume ชื่อ "mysql-pv-volume" ที่ใช้ storageClassName "manual".
- มีความจุ 1Gi และ accessModes เป็น ReadWriteOnce (สามารถเขียนได้จาก node เดียว)
- ใช้ hostPath เพื่อเก็บข้อมูลใน "/mysql"
- กำหนด persistentVolumeReclaimPolicy เป็น Recycle

1.2. PersistentVolumeClaim (PVC):

- สร้าง PersistentVolumeClaim ชื่อ "mysql-pv-claim" ที่ใช้ storageClassName "manual".
- มี accessModes เป็น ReadWriteOnce และขอ storage 1Gi.

1.3. Service:

- สร้าง Service ชื่อ "wordpress-mysql" ที่ใช้ port 3306.
- เลือก pods ที่มี label "app: wordpress-mysql".

1.4. Deployment:

- สร้าง Deployment ชื่อ "wordpress-mysql" ที่ใช้ pods ที่มี label "app: wordpress-mysql".
- กำหนด strategy เป็น Recreate (จะสร้าง pods ใหม่และลบ pods เก่า).
- กำหนด container ใน pod โดยใช้ image "mysql:5.7" และกำหนด environment variables สำหรับการกำหนดค่า MySQL.
- เปิดพอร์ต 3306 ใน container.
- ใช้ PersistentVolumeClaim "mysql-pv-claim" ที่ได้รับจาก PVC เพื่อเก็บข้อมูลฐานข้อมูล MySQL ใน "/var/lib/mysql".

2. myadmin_deployment.yml

2.1. Deployment:

- สร้าง Deployment ชื่อ "phpmyadmin-deployment" ที่มี 1 replica.
- เลือก pods ที่มี label "app: phpmyadmin".
- กำหนด container ใน pod โดยใช้ image "phpmyadmin".
- เปิดพอร์ต 80 ใน container.
- กำหนด environment variables สำหรับกำหนดค่าใน phpMyAdmin:

PMA_HOST: ชื่อของ MySQL service (wordpress-mysql)

PMA_PORT: พอร์ตที่ MySQL service ใช้ (3306)

MYSQL_ROOT_PASSWORD: รหัสผ่านของผู้ใช้ root ใน MySQL

UPLOAD_LIMIT: ขนาดสูงสุดของการอัปโหลดไฟล์ (100MB)

2.2. Service:

- สร้าง Service ชื่อ "phpmyadmin-service".
- ใช้ประเภท NodePort เพื่อให้บริการเข้าถึงจากภายนอก cluster.
- เลือก pods ที่มี label "app: phpmyadmin".
- เปิดพอร์ต 80 ใน service และเชื่อมโยงกับพอร์ต 80 ของ container.

3. wordpress-deployment.yml

3.1. PersistentVolume (PV):

- สร้าง PersistentVolume ชื่อ "wp-pv-volume" ที่ใช้ storageClassName "manual".
- มีความจุ 1Gi และ accessModes เป็น ReadWriteOnce (สามารถเขียนได้จาก node เดียว)
- ใช้ hostPath เพื่อเก็บข้อมูลใน "/wordpress"

- กำหนด persistentVolumeReclaimPolicy เป็น Recycle

3.2. PersistentVolumeClaim (PVC):

- สร้าง PersistentVolumeClaim ชื่อ "wp-pv-claim" ที่ใช้ storageClassName "manual".
- มี accessModes เป็น ReadWriteOnce และขอ storage 1Gi.

3.3. Service:

- สร้าง Service ชื่อ "wordpress" ที่ใช้ port 80.
- เลือก pods ที่มี label "app: wordpress" และ "tier: frontend".
- เปิดพอร์ต 80 ของ container.
- กำหนด NodePort เป็น 30002 และประเภทของ Service เป็น LoadBalancer.

3.4. Deployment:

- สร้าง Deployment ชื่อ "wordpress" ที่มี 2 replicas.
- เลือก pods ที่มี label "app: wordpress" และ "tier: frontend".
- กำหนด strategy เป็น Recreate (จะสร้าง pods ใหม่และลบ pods เก่า).
- กำหนด container ใน pod โดยใช้ image "wordpress:5.9.3-php7.4-apache" และกำหนด environment variables สำหรับการกำหนดค่า WordPress.
- เปิดพอร์ต 80 ใน container.
- ใช้ PersistentVolumeClaim "wp-pv-claim" ที่ได้รับจาก PVC เพื่อเก็บข้อมูล WordPress ใน "/var/www/html".

การทำงานของ Kubernetes

1. mysql-deployment.yml

```
! mysql-deployment.yml
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: mysql-pv-volume
5    labels:
6      type: local
7  spec:
8    storageClassName: manual
9    capacity:
10     storage: 1Gi
11    accessModes:
12     - ReadWriteOnce
13    hostPath:
14     path: "/mysql"
15    persistentVolumeReclaimPolicy: Retain
16
17  ---
18  apiVersion: v1
19  kind: PersistentVolumeClaim
20  metadata:
21    name: mysql-pv-claim
22  spec:
23    storageClassName: manual
24    accessModes:
25     - ReadWriteOnce
26    resources:
27     requests:
28       storage: 1Gi
29
30  ---
```

```
! mysql-deployment.yml
30  ---
31  apiVersion: v1
32  kind: Service
33  metadata:
34    name: wordpress-mysql
35    labels:
36     app: wordpress-mysql
37  spec:
38    ports:
39     - port: 3306
40       targetPort: 3306
41    selector:
42     app: wordpress-mysql
43
44  ---
45  apiVersion: apps/v1
46  kind: Deployment
47  metadata:
48    name: wordpress-mysql
49    labels:
50     app: wordpress-mysql
51  spec:
52    selector:
53     matchLabels:
54       app: wordpress-mysql
55    strategy:
56     type: Recreate
57    template:
58     metadata:
59       labels:
60         app: wordpress-mysql
61     spec:
```

```
61     spec:
62       containers:
63         - image: mysql:5.7
64           name: mysql
65           env:
66             - name: MYSQL_DATABASE
67               value: wordpress_db
68             - name: MYSQL_ROOT_PASSWORD
69               value: password
70             - name: MYSQL_USER
71               value: wordpress
72             - name: MYSQL_PASSWORD
73               value: wordpress
74           ports:
75             - containerPort: 3306
76               name: mysql
77           volumeMounts:
78             - name: mysql-persistent-storage
79               mountPath: /var/lib/mysql
80       volumes:
81         - name: mysql-persistent-storage
82           persistentVolumeClaim:
83             claimName: mysql-pv-claim
84
```


1.1 PersistentVolume

- apiVersion: ระบุเวอร์ชัน API ที่ใช้
- kind: ประเภท PersistentVolume
- metadata: ข้อมูลเกี่ยวกับวัตถุ
- name: ชื่อ PersistentVolume (mysql-pv-volume)
- labels: แท็กประเภท (local)
- spec: รายละเอียดการตั้งค่า
- storageClassName: ชนิดของ Storage Class (manual)
- capacity: ขนาดพื้นที่ที่จัดเก็บ (1Gi)
- accessModes: โหมดการเข้าถึง (ReadWriteOnce)
- hostPath: ที่อยู่จัดเก็บบนโฮสต์ (/mysql)
- persistentVolumeReclaimPolicy: นโยบายคืนค่า (Retain)

1.2 PersistentVolumeClaim

- apiVersion: ระบุเวอร์ชัน API ที่ใช้
- kind: ประเภท PersistentVolumeClaim
- metadata: ข้อมูลเกี่ยวกับวัตถุ
- name: ชื่อ PVC (mysql-pv-claim)
- spec: รายละเอียดการตั้งค่า
- storageClassName: ชนิดของ Storage Class (manual)
- accessModes: โหมดการเข้าถึง (ReadWriteOnce)
- resources: กำหนดทรัพยากรที่ต้องการ
- requests: ต้องการพื้นที่จัดเก็บ (1Gi)

1.3 Service

- apiVersion: ระบุเวอร์ชัน API ที่ใช้
- kind: ประเภท Service
- metadata: ข้อมูลเกี่ยวกับวัตถุ
- name: ชื่อ Service (wordpress-mysql)
- labels: แท็กสำหรับแอปพลิเคชัน (wordpress-mysql)
- spec: รายละเอียดการตั้งค่า
- ports: กำหนดพอร์ตที่ Service เปิดเผยแพร่

- port: พอร์ตที่ Service ใช้ (3306)
- targetPort: พอร์ตที่ Pod ฟังอยู่ (3306)
- selector: เชื่อมโยงกับ Pod ที่มี label (app: wordpress-mysql)

1.4 Deployment

- apiVersion: ระบุเวอร์ชัน API สำหรับ Deployment
- kind: ประเภท Deployment
- metadata: ข้อมูลเกี่ยวกับวัตถุ
- name: ชื่อ Deployment (wordpress-mysql)
- labels: แท็กสำหรับแอปพลิเคชัน (wordpress-mysql)
- spec: รายละเอียดการตั้งค่า
- selector: เลือก Pod ด้วย label (app: wordpress-mysql)
- strategy: ใช้กลยุทธ์การอัปเดต (Recreate)
- template: รูปแบบของ Pod
- metadata: ข้อมูลเกี่ยวกับ Pod
- labels: แท็กสำหรับ Pod (wordpress-mysql)
- spec: รายละเอียดของ Pod

1.5 containers: รายการของ container ที่จะรัน

- image: ใช้ Docker image (mysql:5.7)
- name: ชื่อ container (mysql)
- env: กำหนด environment variables สำหรับ MySQL
 - MYSQL_DATABASE: ชื่อฐานข้อมูล (wordpress_db)
 - MYSQL_ROOT_PASSWORD: รหัสผ่าน root (password)
 - MYSQL_USER: ชื่อผู้ใช้ (wordpress)
 - MYSQL_PASSWORD: รหัสผ่านผู้ใช้ (wordpress)
- ports: กำหนดพอร์ตที่ container จะฟัง
 - containerPort: พอร์ตที่ MySQL ฟังอยู่ (3306)
 - name: ชื่อพอร์ต (mysql)
- volumeMounts: กำหนด mount volume
 - name: ชื่อ volume ที่ mount (mysql-persistent-storage)
 - mountPath: ตำแหน่งที่ mount ใน container (/var/lib/mysql)

1.6 volumes: รายการ volume ที่ใช้ใน Pod

- name: ชื่อ volume (mysql-persistent-storage)
persistentVolumeClaim: ใช้ PVC ที่อ้างถึง
claimName: ชื่อ PVC (mysql-pv-claim)

2. phpmyadmin_deployment.yml

```
! phpmyadmin-deployment.yml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: phpmyadmin-deployment
5    labels:
6      app: phpmyadmin
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: phpmyadmin
12    template:
13     metadata:
14       labels:
15         app: phpmyadmin
16     spec:
17       containers:
18         - name: phpmyadmin
19           image: phpmyadmin
20           ports:
21             - containerPort: 80
22           env:
23             - name: PMA_HOST
24               value: wordpress-mysql
25             - name: PMA_PORT
26               value: "3306"
27             - name: MYSQL_ROOT_PASSWORD
28               value: password
29             - name: UPLOAD_LIMIT
30               value: 100M
31
```

```
32  ---
33  apiVersion: v1
34  kind: Service
35  metadata:
36    name: phpmyadmin-service
37  spec:
38    type: NodePort
39    selector:
40      app: phpmyadmin
41    ports:
42      - protocol: TCP
43        port: 80
44        targetPort: 80
45
```

2.2 Deployment

- apiVersion: ระบุเวอร์ชัน API ที่ใช้สำหรับ Deployment
- kind: ประเภทวัตถุเป็น Deployment
- metadata: ข้อมูลเกี่ยวกับ Deployment
- name: ตั้งชื่อ Deployment ว่า phpmyadmin-deployment
- labels: แท็กสำหรับการจัดกลุ่ม
- app: ระบุชื่อแอปว่า phpMyAdmin
- spec: สเปคของ Deployment
- replicas: จำนวน Pod ที่ต้องการ (1)
- selector: ระบุการเลือก Pod ด้วย label
- matchLabels: เลือก Pod ที่มี label app: phpMyAdmin

- template: รูปแบบของ Pod ที่จะสร้าง
- metadata: ข้อมูลเกี่ยวกับ Pod
- labels: แท็กสำหรับ Pod (phpmyadmin)
- spec: สเปคของ Pod
- containers: รายการ container ที่จะรัน
 - name: ชื่อ container (phpmyadmin)
 - image: ใช้ Docker image ของ phpmyadmin
 - ports: กำหนดพอร์ตที่ container จะฟัง
 - containerPort: พอร์ตที่ MySQL ฟังอยู่ (80)
- env: กำหนด environment variables สำหรับ phpMyAdmin
 - PMA_HOST: ชื่อ host ของ MySQL ที่จะเชื่อมต่อ (wordpress-mysql)
 - PMA_PORT: พอร์ตของ MySQL (3306)
 - MYSQL_ROOT_PASSWORD: รหัสผ่านของ root สำหรับ MySQL (password)
 - UPLOAD_LIMIT: ขนาดสูงสุดของไฟล์อัปโหลด (100M)

2.3 Service

- apiVersion: ระบุเวอร์ชัน API ที่ใช้สำหรับ Service
- kind: ประเภทวัตถุเป็น Service
- metadata: ข้อมูลเกี่ยวกับ Service
- name: ตั้งชื่อ Service ว่า phpmyadmin-service
- spec: สเปคของ Service
- type: ประเภทของ Service คือ NodePort
- selector: เชื่อมโยงกับ Pod ที่มี label app: phpMyAdmin
- ports: รายการพอร์ตที่ Service จะใช้
- protocol: โพรโทคอลที่ใช้ (TCP)
- port: พอร์ตที่ Service เปิดเผยแพร่ (80)
- targetPort: พอร์ตที่ Pod ฟังอยู่ (80)

3. wordpress-deployment.yml

```
! wordpress-deployment.yml
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: wp-pv-volume
5    labels:
6      type: local
7  spec:
8    storageClassName: manual
9    capacity:
10     storage: 1Gi
11    accessModes:
12     - ReadWriteOnce
13    hostPath:
14     path: "/wordpress"
15    persistentVolumeReclaimPolicy: Retain
16
17 ---
18 apiVersion: v1
19 kind: PersistentVolumeClaim
20 metadata:
21   name: wp-pv-claim
22 spec:
23   storageClassName: manual
24   accessModes:
25     - ReadWriteOnce
26   resources:
27     requests:
28       storage: 1Gi
29
```

```
! wordpress-deployment.yml
30 ---
31 apiVersion: v1
32 kind: Service
33 metadata:
34   name: wordpress
35 spec:
36   ports:
37     - port: 80
38       targetPort: 80
39       nodePort: 30002
40   selector:
41     app: wordpress
42   type: LoadBalancer
43
44 ---
45 apiVersion: apps/v1
46 kind: Deployment
47 metadata:
48   name: wordpress
49   labels:
50     app: wordpress
51     tier: frontend
52 spec:
53   replicas: 2
54   selector:
55     matchLabels:
56       app: wordpress
57       tier: frontend
58   strategy:
59     type: Recreate
60   template:
```

```
61     metadata:
62       labels:
63         app: wordpress
64         tier: frontend
65     spec:
66       containers:
67         - image: wordpress:5.9.3-php7.4-apache
68           name: wordpress
69           env:
70             - name: WORDPRESS_DB_HOST
71               value: wordpress-mysql
72             - name: WORDPRESS_DB_NAME
73               value: wordpress_db
74             - name: WORDPRESS_DB_USER
75               value: wordpress
76             - name: WORDPRESS_DB_PASSWORD
77               value: wordpress
78           ports:
79             - containerPort: 80
80           volumeMounts:
81             - name: wordpress-persistent-storage
82               mountPath: /var/www/html
83       volumes:
84         - name: wordpress-persistent-storage
85           persistentVolumeClaim:
86             claimName: wp-pv-claim
87
```

3.1 PersistentVolume

- apiVersion: ระบุเวอร์ชัน API ที่ใช้สำหรับ PersistentVolume
- kind: ประเภทวัตถุเป็น PersistentVolume
- metadata: ข้อมูลเกี่ยวกับ PersistentVolume
- name: ตั้งชื่อ PersistentVolume ว่า wp-pv-volume
- labels: แท็กประเภท (local)
- spec: รายละเอียดการตั้งค่า
- storageClassName: ชนิดของ Storage Class (manual)
- capacity: ขนาดพื้นที่ที่จัดเก็บ (1Gi)
- accessModes: โหมดการเข้าถึง (ReadWriteOnce)
- hostPath: ที่อยู่จัดเก็บบนโฮสต์ (/wordpress)
- persistentVolumeReclaimPolicy: นโยบายคืนค่าคือ Retain

3.2 PersistentVolumeClaim

- apiVersion: ระบุเวอร์ชัน API ที่ใช้สำหรับ PersistentVolumeClaim
- kind: ประเภทวัตถุเป็น PersistentVolumeClaim
- metadata: ข้อมูลเกี่ยวกับ PersistentVolumeClaim
- name: ตั้งชื่อ PVC ว่า wp-pv-claim
- spec: รายละเอียดการตั้งค่า
- storageClassName: ชนิดของ Storage Class (manual)
- accessModes: โหมดการเข้าถึง (ReadWriteOnce)
- resources: กำหนดทรัพยากรที่ต้องการ
- requests: ต้องการพื้นที่จัดเก็บ (1Gi)

3.3 Service

- apiVersion: ระบุเวอร์ชัน API ที่ใช้สำหรับ Service
- kind: ประเภทวัตถุเป็น Service
- metadata: ข้อมูลเกี่ยวกับ Service
- name: ตั้งชื่อ Service ว่า wordpress
- spec: รายละเอียดการตั้งค่า
- ports: กำหนดพอร์ตที่ Service เปิดเผยแพร่
- port: พอร์ตที่ Service ใช้ (80)

- targetPort: พอร์ตที่ Pod ฟังอยู่ (80)
- nodePort: พอร์ตที่เปิดบนโหนด (30002)
- selector: เชื่อมโยงกับ Pod ที่มี label app: wordpress
- type: ประเภทของ Service คือ LoadBalancer

3.4 Deployment

- apiVersion: ระบุเวอร์ชัน API ที่ใช้สำหรับ Deployment
- kind: ประเภทวัตถุเป็น Deployment
- metadata: ข้อมูลเกี่ยวกับ Deployment
- name: ตั้งชื่อ Deployment ว่า wordpress
- labels: แท็กสำหรับแอปพลิเคชัน
- app: wordpress
- tier: frontend
- spec: รายละเอียดการตั้งค่า
- replicas: จำนวน Pod ที่ต้องการ (2)
- selector: เลือก Pod ด้วย label
- matchLabels: ต้องมี label app: wordpress และ tier: frontend
- strategy: ใช้กลยุทธ์การอัปเดต (Recreate)
- template: รูปแบบของ Pod
- metadata: ข้อมูลเกี่ยวกับ Pod
- labels: แท็กสำหรับ Pod
- spec: สเปคของ Pod
- containers: รายการ container ที่จะรัน
 - image: ใช้ Docker image ของ WordPress (wordpress:5.9.3-php7.4-apache)
 - name: ชื่อ container (wordpress)
- env: กำหนด environment variables สำหรับ WordPress
 - WORDPRESS_DB_HOST: ชื่อ host ของ MySQL (wordpress-mysql)
 - WORDPRESS_DB_NAME: ชื่อฐานข้อมูล (wordpress_db)
 - WORDPRESS_DB_USER: ชื่อผู้ใช้ (wordpress)
 - WORDPRESS_DB_PASSWORD: รหัสผ่านผู้ใช้ (wordpress)
- ports: กำหนดพอร์ตที่ container จะฟัง

- containerPort: พอร์ตที่ WordPress ฟังอยู่ (80)
- volumeMounts: กำหนด mount volume
- name: ชื่อ volume ที่ mount (wordpress-persistent-storage)
- mountPath: ตำแหน่งที่ mount ใน container (/var/www/html)
- volumes: รายการ volume ที่ใช้ใน Pod
- name: ชื่อ volume (wordpress-persistent-storage)
- persistentVolumeClaim: ใช้ PVC ที่อ้างอิง
- claimName: ชื่อ PVC (wp-pv-claim)

การใช้งานคำสั่ง kubectl เพื่อสร้าง Deployment ใน Kubernetes

```
PS D:\project> kubectl apply -f mysql-deployment.yml
persistentvolume/mysql-pv-volume created
persistentvolumeclaim/mysql-pv-claim created
service/wordpress-mysql created
deployment.apps/wordpress-mysql created
PS D:\project> kubectl apply -f phpmyadmin-deployment.yml
deployment.apps/phpmyadmin-deployment created
service/phpmyadmin-service unchanged
PS D:\project> kubectl apply -f wordpress-deployment.yml
persistentvolume/wp-pv-volume configured
persistentvolumeclaim/wp-pv-claim unchanged
service/wordpress unchanged
deployment.apps/wordpress created
```

การใช้คำสั่ง kubectl get pods เพื่อตรวจสอบ pods

```
PS D:\project> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
phpmyadmin-deployment-58bdc6f899-cvxb2	1/1	Running	0	2m8s
wordpress-c6dfff598-4r7xs	1/1	Running	0	35s
wordpress-c6dfff598-r2v67	1/1	Running	0	35s
wordpress-mysql-84fb4f6d9b-hg6gb	1/1	Running	0	2m27s

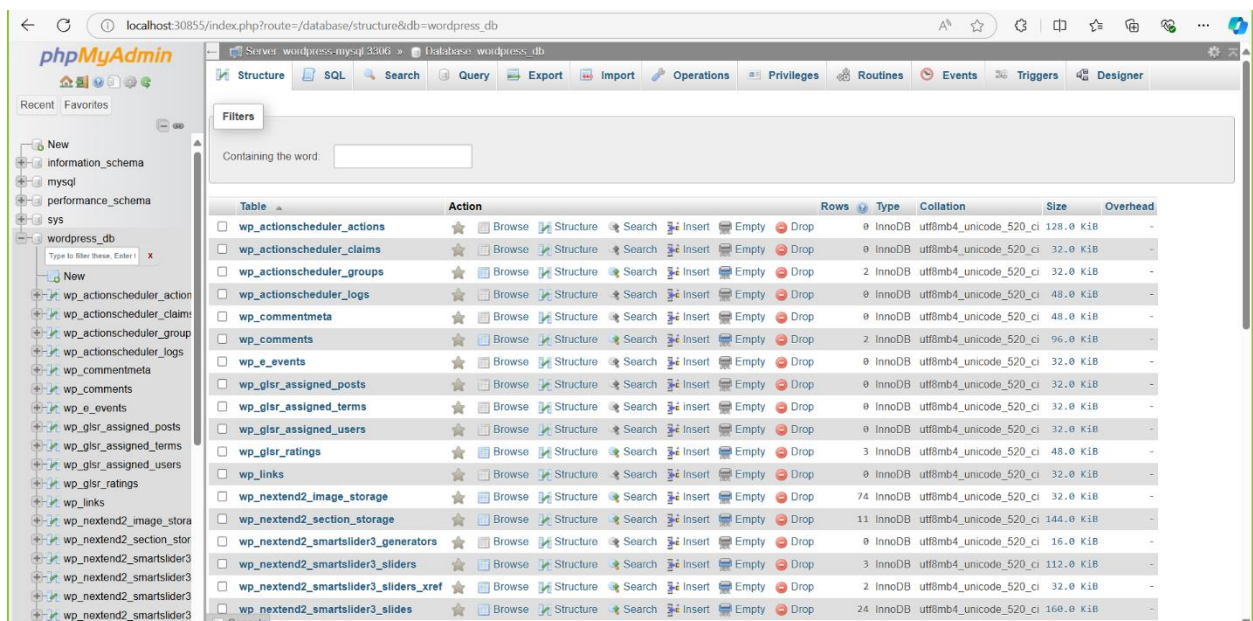
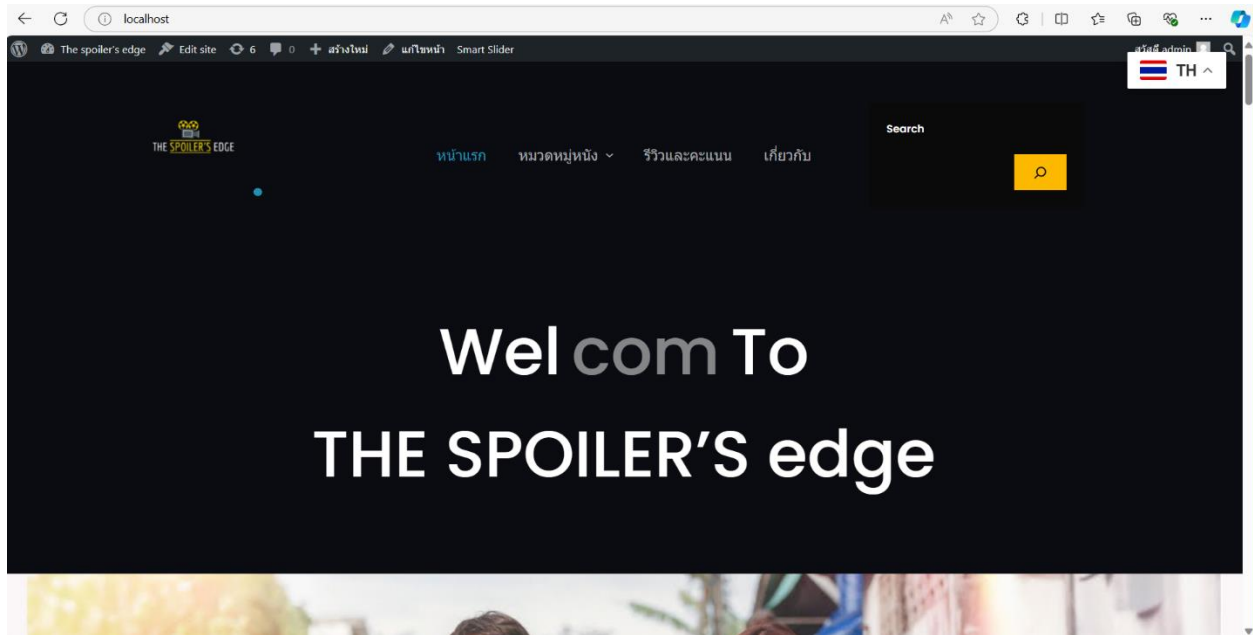
การใช้คำสั่ง kubectl get service เพื่อตรวจสอบ service

```
PS D:\project> kubectl get service
```

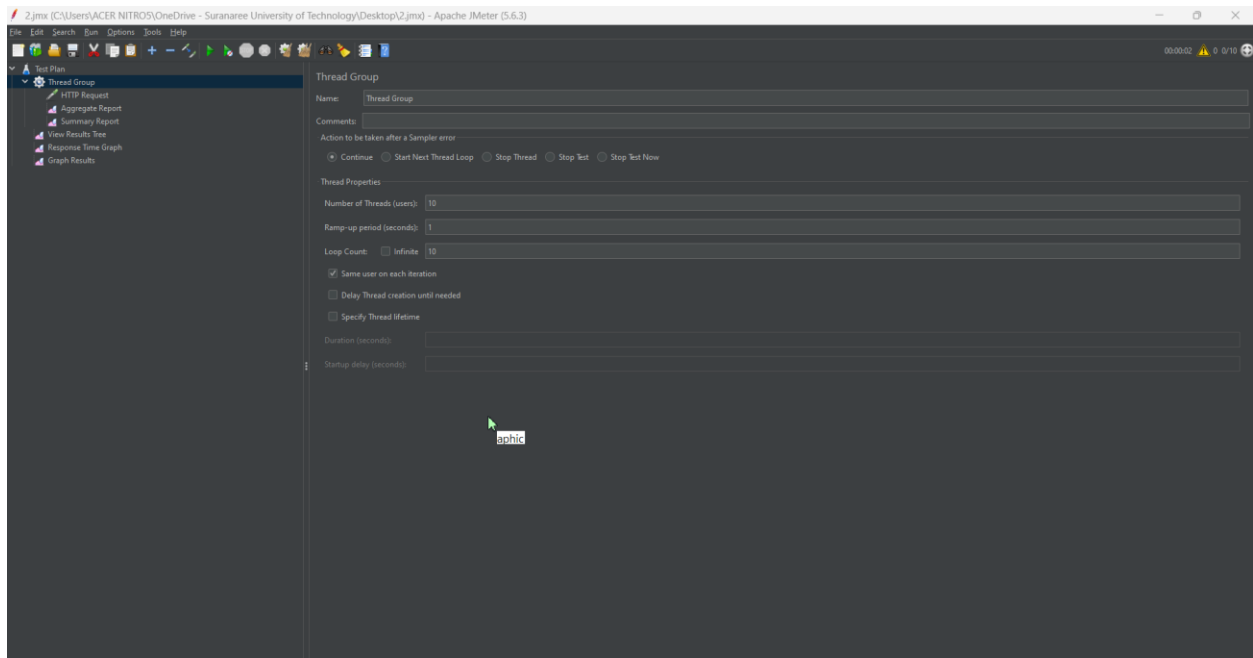
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	68m
phpmyadmin-service	NodePort	10.103.227.99	<none>	80:31278/TCP	61m
wordpress	LoadBalancer	10.108.146.187	localhost	80:30002/TCP	57m
wordpress-mysql	ClusterIP	10.107.27.175	<none>	3306/TCP	37m

```
PS D:\project> 
```

รายละเอียดตัวเว็บ Wordpress / phpMyAdmin หลังจากการทำ Kubernetes



ทดสอบการรันของ wordpress by JamesMiner



replicas: 1

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	100	101	75	169	17.48	0.00%	37.0/sec	7432.11	4.12	205635.1
TOTAL	100	101	75	169	17.48	0.00%	37.0/sec	7432.11	4.12	205635.1

replicas: 10

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	100	135	91	185	18.12	0.00%	45.5/sec	9132.13	5.06	205635.1
TOTAL	100	135	91	185	18.12	0.00%	45.5/sec	9132.13	5.06	205635.1

แผนการดำเนินงานของโครงการ

กิจกรรม	สัปดาห์ที่											
	1	2	3	4	5	6	7	8	9	10	11	12
1. ศึกษาข้อมูลเกี่ยวกับโครงการและหาหัวข้อในการทำโครงการ												
2. จัดทำเอกสารในการรายงานข้อเสนอโครงการ												
3. เสนอโครงการที่จะทำการศึกษา												
4. ศึกษาหาข้อมูลเพื่อเตรียมทำเว็บไซต์												
5. เริ่มต้นการทำเว็บบน xampp												
6. ใส่ข้อมูลต่างๆที่รวบรวมมาและตกแต่งเว็บ												
7. นำเว็บไซต์จาก Xampp มา Run บน Docker Compose												
8. เริ่มทำเว็บบน Docker Swarm												
9. จัดการ Web บน Docker Kubernetes												
10. ปิดโครงการ												

สรุปการทำงาน

จากการพัฒนาเว็บไซต์ด้วย Wordpress โดยใช้ Kubernetes ในการ deployment โดยตัว Kubernetes จะช่วยในการสร้าง, จัดการ, และปรับขนาด containers ของ Wordpress โดยอัตโนมัติ ซึ่งทำได้ง่ายต่อการจัดการ resource และการทำ auto-scaling เมื่อมีการบริการมากขึ้นหรือน้อยลง และมีประโยชน์ต่อผู้ใช้งาน