



研華—WISE-PaaS/EnSaaS 物聯網雲平台 DevOps 實踐

案例簡介

(一) 案例背景

在物聯網的帶動下，製造業正迎來新一輪變革浪潮，雲計算、大數據、人工智慧等新技術正在加速與工業領域的全方位融合。縱觀全球市場，工業 4.0 趨勢所向，各國的企業都在製造領域中尋找新的經濟成長契機。自從 2015 年大陸公佈《中國製造 2025》計畫以來，大陸的工業轉型正迎來大突破、大升級。《中國製造 2025》的核心關鍵是為了打造智慧化、網路化生產系統的「智慧工廠」。物聯網、大數據、人工智慧和實體經濟深度融合，在工業的應用下已迎來蓬勃發展時期。

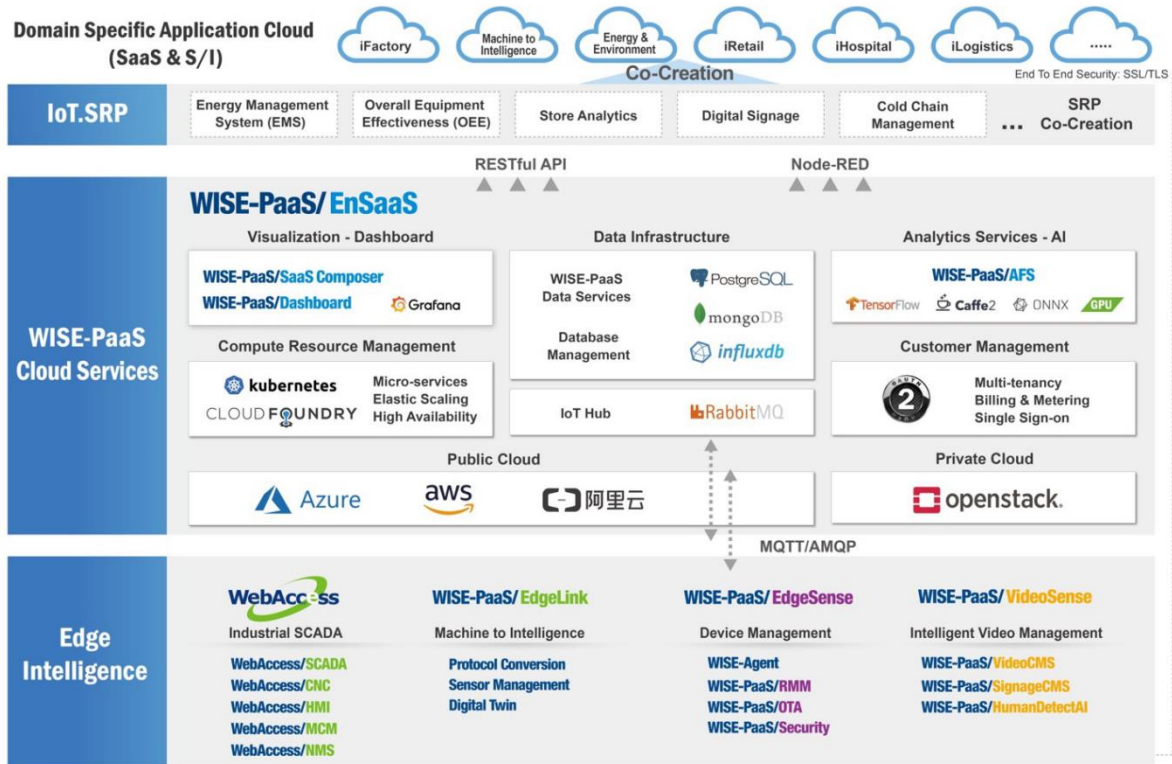
作為最早工業電腦行業的廠商之一，研華意識到物聯網的發展趨勢將給世界帶來巨大的變化，致力於充當推手的角色來推動物聯網產業的發展。研華為建構工業物聯網之完整價值鏈，自 2017 年起與資策會協同開發 IoT PaaS 物聯網雲平台，並導入 DevOps 的開發流程，以加速從端到雲 (Edge, PaaS, SaaS) 之串接與維運服務。雙方將透過研華在 Edge 端既有的硬體系統優勢、共同開發之 WISE-PaaS 2.0 資料分析平台，以及開放各垂直領域第三方單位之 SaaS 服務於該平台上進行開發，以布建完整工業設備聯網之維運雲端服務平台。

(二) 案例特點

WISE-PaaS 工業物聯網雲平台，以下簡稱為「WISE-PaaS 雲平台」或「雲平台」，是一個整合的物聯網服務平台，旨在從邊緣到雲端提供可操作的洞察力。讓使用者能夠輕鬆安全地連接、管理和吸收大規模的物聯網數據，即時處理和分析/視覺化數據。憑藉全套開發工具，WISE-PaaS 簡化了物聯網解決方案佈署，使資源可集中在關注的專業領域。WISE-PaaS 2.0 平台架構如圖 3-1 所示。



WISE-PaaS 2.0 Architecture



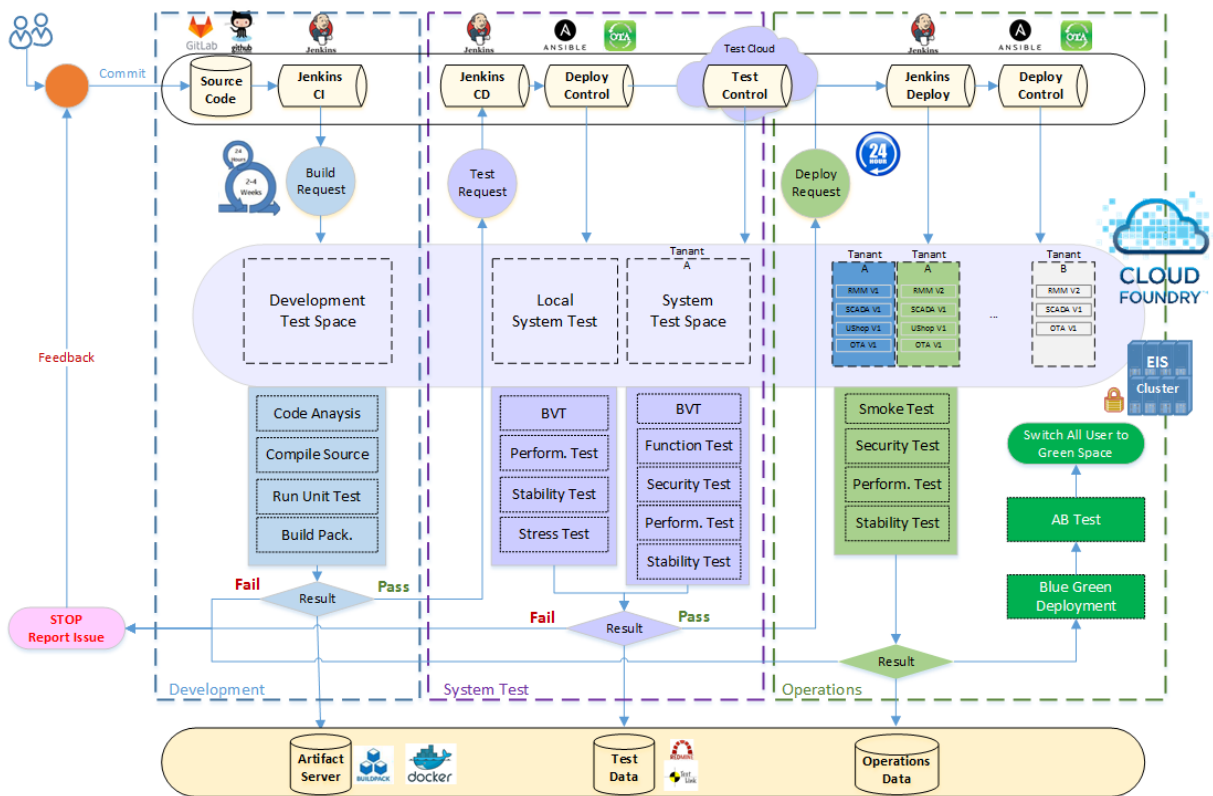
WISE-PaaS 2.0 平台架構圖

需求分析

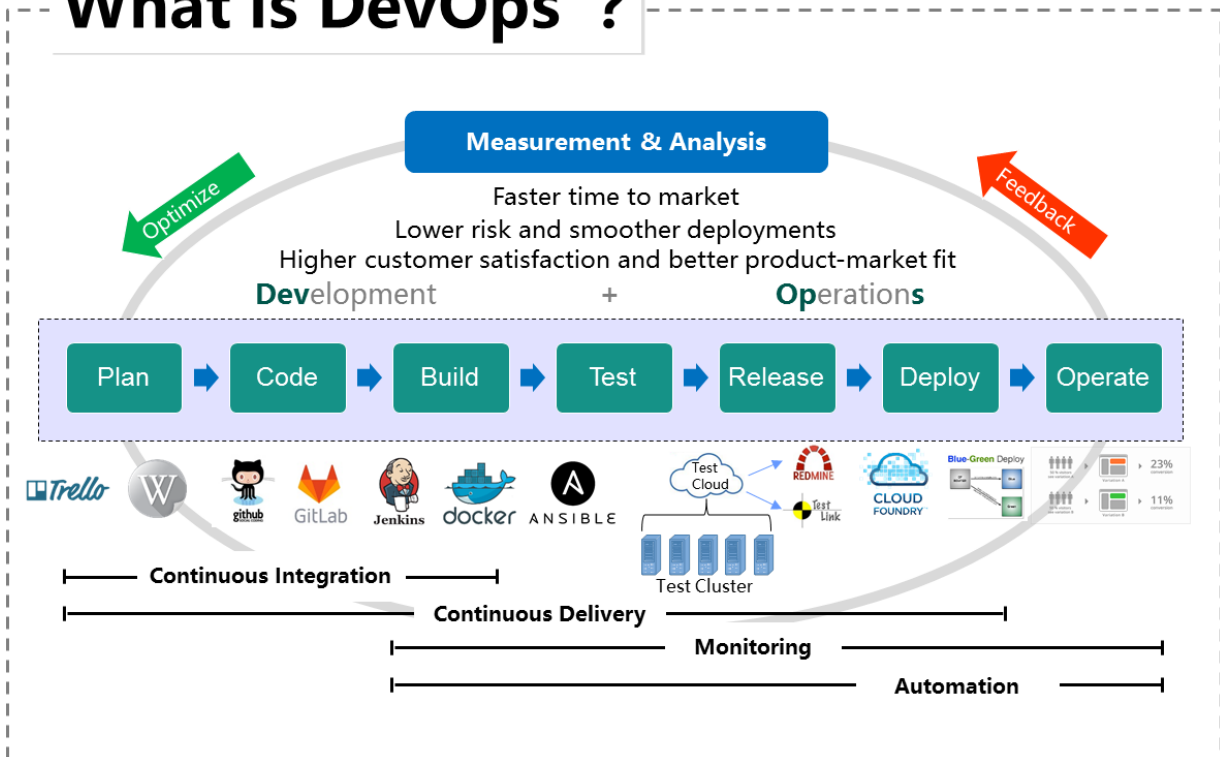
軟體交付需要經過建構、測試、佈署等複雜過程，如果主要依賴人工去完成這個流程需要花費很多時間，延誤產品的上線發佈。

為了實現軟體的快速交付，越來越多的企業開始遵循 DevOps 軟體交付理念和方法，DevOps 集文化、實踐和工具於一身，以開發團隊和維運團隊的密切合作為核心，透過工作實踐將交付過程打造成一條包含開發、建構、測試、發佈、佈署、維運等步驟的標準化流程，並用各種工具將其自動化，最終實現產品的快速、高品質交付，並提供 7*24 小時不間斷服務，如圖 3-2 所示，上圖為研華 WISE-PaaS 架構流程與工具，下圖為打造的交付作業標準化流程。





What is DevOps ?



DevOps 打造交付標準化流程



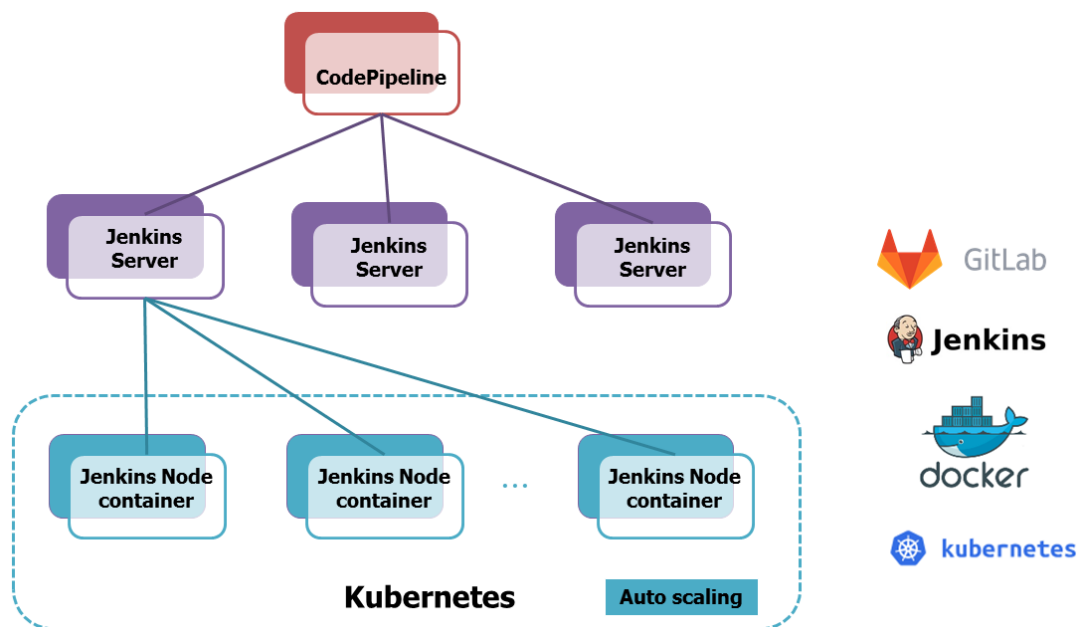
解決方案

(一) 總體技術架構

WISE-PaaS 平台解決方案如圖 3-3 所示。解決方案的最上層提供 CodePipeline 服務，它是正在開發的一款具有持續整合/持續交付能力，並能相容 Jenkins 的 SaaS 化產品。透過使用 CodePipeline，可以使客戶方便的在雲端實現從源碼到應用的持續整合和交付，方便客戶快速的對產品進行功能迭代和推進。

整個解決方案的核心是 Jenkins，Jenkins 提供了軟體發展的持續整合服務，它透過 Master/Agent 架構來實現分散式建構，將不同的任務下發到多台機器(Jenkins Node) 執行，提高處理性能。

解決方案的最下層透過 Kubernetes 來管理 Jenkins 的節點，當有建構任務時會自動創建一個 Docker Container 來完成建構任務，當任務結束後 Container 會自動銷毀，資源動態使用動態銷毀，避免資源浪費，無需擔心源碼或建構物外洩。



WISE-PaaS 平台解決方案

(二) 具體技術方案

WISE-PaaS 平台技術方案可從兩方面闡述，包括持續交付流程以及 Jenkins Pipeline，如圖 3-4 所示。

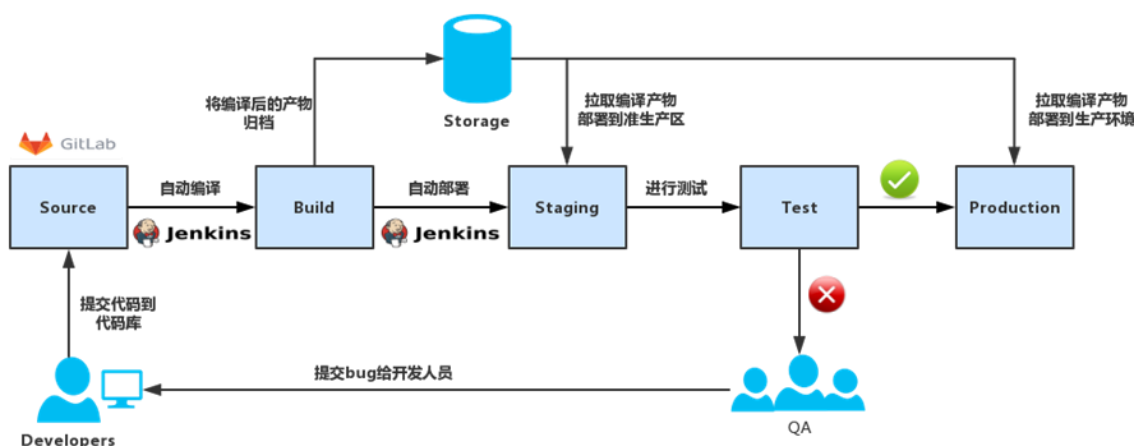


圖 3-4 WISE-PaaS 持續交付流程

1) WISE-PaaS SRP (Solution Ready Package) 持續交付流程

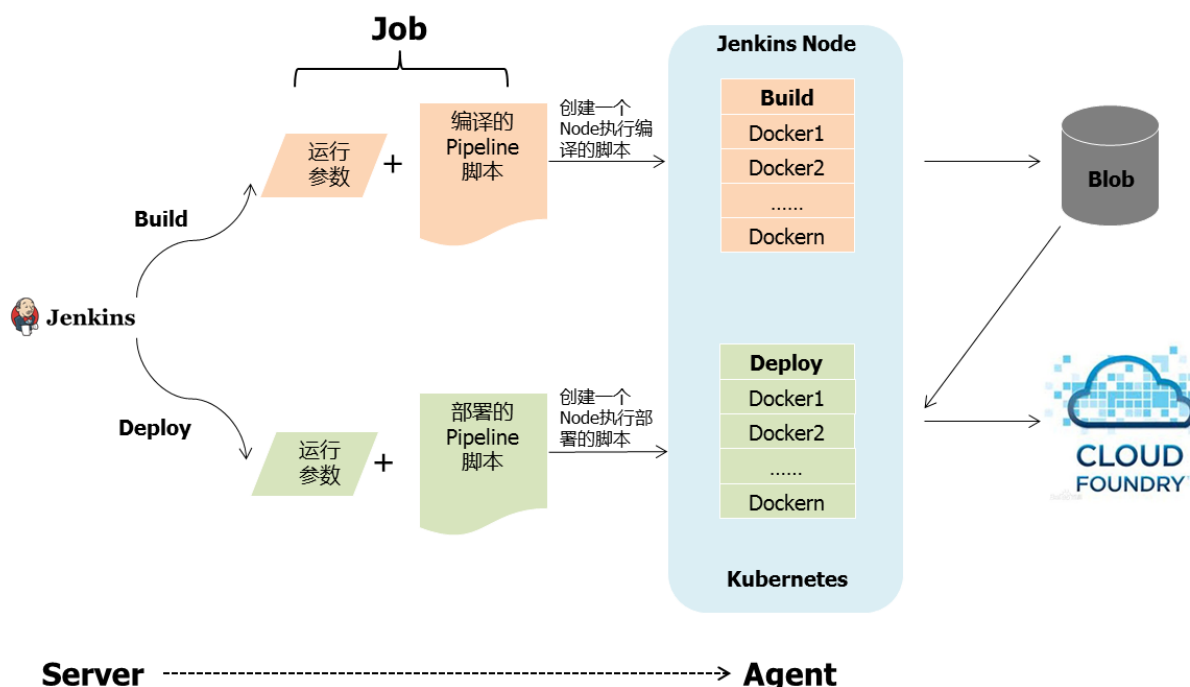
- 開發人員提交源碼到源碼倉庫；
- 倉庫打 tag 後透過 Git Webhook 觸發 Jenkins 上面自動編譯的 Pipeline；
- 編譯後將產物儲存到 storage，例如 blob；
- 觸發 Jenkins 上面自動佈署的 Pipeline 從 storage 拉取編譯產物佈署到準生產區；
- QA 在準生產區進行自動化及人工測試，包括功能測試、性能測試、壓力測試和穩定性測試。
- 測試通過後觸發 Jenkins 上面自動佈署的 Pipeline 將編譯產物佈署到生產區。

2) Jenkins Pipeline

Pipeline 是 Jenkins 的一系列外掛程式的組合，透過這些外掛程式可以將持續交付管道化流程在一個 Jenkinsfile 中實現，將複雜的交付流程轉化為 code，即「Pipeline as Code」，並且可以將 Jenkinsfile 放入專案的源碼管理中，像管理其它源碼一樣來管理

pipeline 的 code。jenkins 中 pipeline 腳本是基於 groovy 編寫的，可實現靈活、可擴展的持續發佈（CD）工作流。

WISE-PaaS 平台 App 的自動建構和自動佈署都是透過編寫 Pipeline 腳本來實現的。在 Jenkins 上面創建建構和佈署 Job，並將建構和佈署的步驟在 Pipeline 腳本中實現，然後配置各種運行參數，便可實現建構和佈署的自動化。Job 被觸發後，Jenkins Server 會將 Pipeline 腳本發送到腳本中指定的 Node 上去執行，最終完成建構和佈署任務，如圖 3-5 所示。



WISE-PaaS 平台實現自動建構和自動佈署

3) Kubernetes+Jenkins

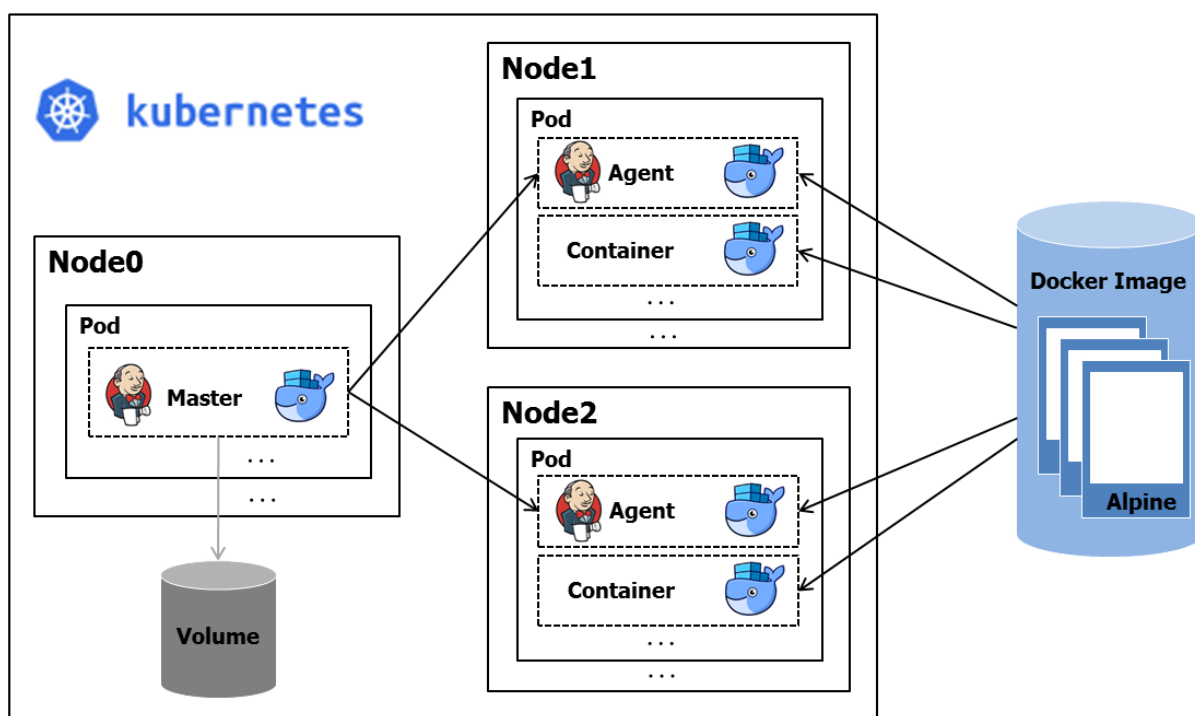
傳統的 Jenkins Master/Agent 方式可以說明使用者實現分散式建構，提高處理性能，但是在使用時還是會存在很多缺點，例如：

- 當 Master 節點發生故障時，便無法再進行任何建構任務；
- 為了完成不同語言的編譯打包等任務，會創建很多 Jenkins Node，但是這些

Node 的環境又很難複製，導致管理和維護都很困難；

➤ 資源配置不均衡，有些 Node 使用率比較高，會出現 job 排隊的情況，但有些使用率比較低的 Node 卻很多時候又處於空閒，導致資源的浪費。

為了解決以上種種問題，需要尋找一種更可靠更高效的方式來完成 CI/CD 流程，使用 Kubernetes 搭建 Jenkins 集群的架構便解決了這些問題，如圖 3-6 所示。



以 Kubernetes 搭建 Jenkins 集群完成高效 CI/CD 流程

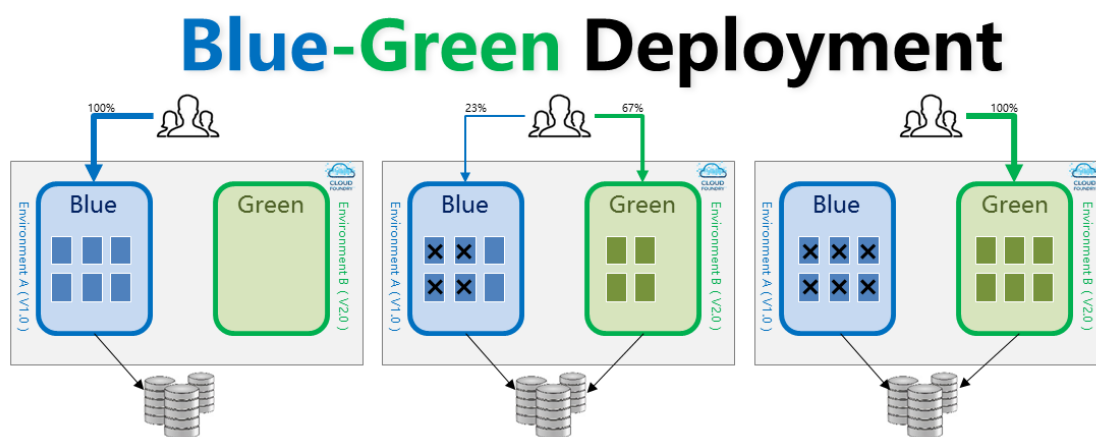
在這種架構中，Jenkins Master 和 Jenkins Agent 以 Docker Container 形式運行在 Kubernetes 集群的 Node 上，創建一個持久化的 Volume 用來儲存 Jenkins 服務的資料，當 Master 出現故障時，可以保證資料不會丟失。創建 Jenkins Agent 使用的 Docker Image 保存在 Docker 儲存服務中(例如 Docker Hub)，便於管理和複用。Jenkins Agent 會根據需要拉取 Docker Image 動態創建和銷毀，不會一直佔用資源。

具體實現過程：Jenkins 提供了 Kubernetes 的外掛程式，安裝外掛程式並配置好 Kubernetes 連接資訊後，就可以在 Pipeline 腳本中調用 Kubernetes 的介面自動啟動一

個 Pod 和多個 Docker Container，其中一個 Container 作為 Jenkins Agent 註冊到 Master 上面，建構任務可以指定在其他 Container 上面完成，當所有操作都完成後，Jenkins Agent 會被註銷，並且釋放掉所有 Docker Container 的資源。

4) 藍綠佈署

傳統模式下，如果要更新應用，基本上無可避免有停機時間。在 WISE-PaaS 上，透過藍綠佈署實現不間斷服務的更新。藍綠佈署是軟體佈署模式的一個術語，藍色是現在正在運行的當前版本，綠色是更新的版本，先佈署綠色版本，然後對綠色版本執行煙霧測試。通過測試後，才將應用流量逐步切換到綠色版本，然後監控綠色版本，一旦異常，立刻回滾到藍色版本。整個過程高效迅速，可保證零宕機升級，服務不間斷，如圖 3-7 所示。

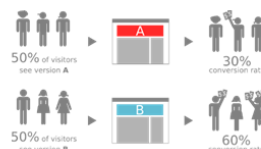


藍綠部署流程

- 从已有的版本开始
- 部署新的版本
- 测试绿版本
- 逐步切换用户流量
- 监控绿版本
 - 异常，回滚蓝版本
 - 正常，删除蓝版本

藍綠部署益处

- 全自动化部署，高效迅速
- 零宕机
- 易于回滚
- 逐渐增加负载，降低容量风险



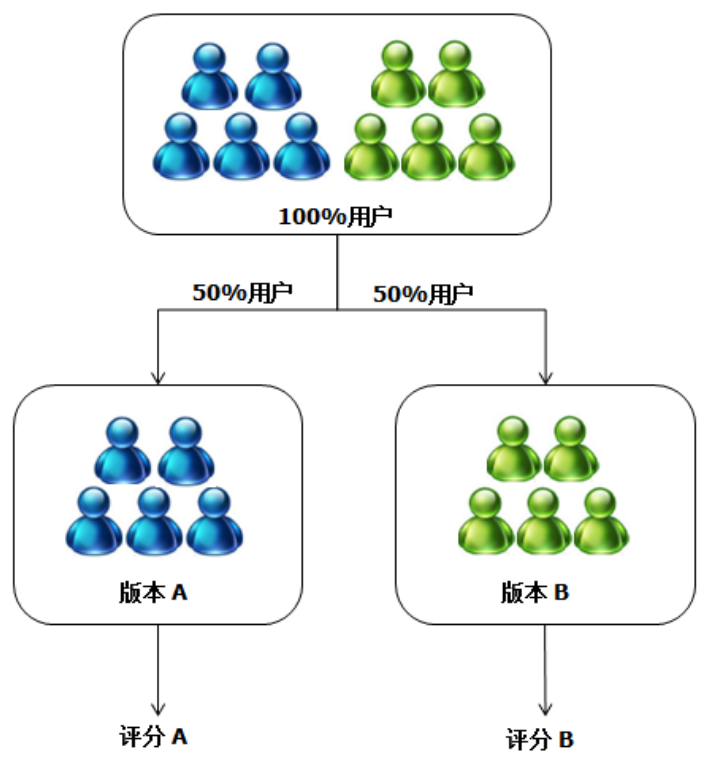
透過藍綠佈署實現不間斷服務更新



WISE-PaaS App 能夠實現藍綠佈署，主要是因為平台支援 App 透過 Scale 實例個數來分擔流量，比如藍的版本有 6 個實例，目前使用者訪問的所有流量都集中在藍的版本上。需要升級時，先佈署只有一個實例的綠的版本，測試通過後，再將其 Scale 到 4 個實例，並將使用者使用的 Route Map 到綠的版本上，並且將藍的版本 Scale 為 2 個實例，這個時候，綠的版本正式被用戶所用，並且分擔了 67% 的流量，然後再繼續將綠的版本 Scale 到 6 個實例，並且將藍的版本 Stop，最終所有流量成功導入到綠的版本，升級過程中不會出現宕機的狀況。

5) 煙霧測試與 A/B 測試

QA 為每個 APP 都編寫了自動化的 SmokeTest 腳本，每次佈署之後都會自動運行 SmokeTest 腳本，保證 APP 佈署以及基本功能的正確性，如圖 3-8 所示。



结论:

煙霧測試與 A/B 測試確保功能正確性

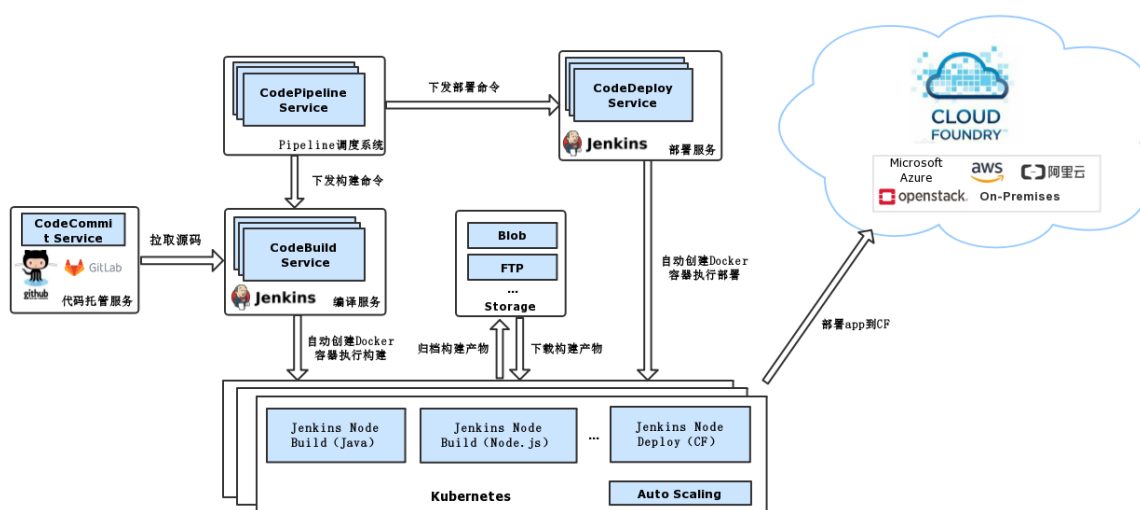


WISE-PaaS 平台的 APP 還支援 A/B 測試。在不斷引進持續交付的流程後，產品更新越來越快，不同的方案設計有可能會帶來不同的用戶使用效果，A/B 測試提供了一個有價值的方式來評估新特性對客戶行為的影響。具體來說，就是為同一個目標制定 2 個方案（例如 2 個頁面），讓一部分使用者使用 A 方案，另一部分用戶使用 B 方案，對 2 種方案進行用戶轉化率、點擊量等指標的統計，以判斷 2 種方案的優劣並進行決策。

WISE-PaaS App 支持 A/B Test，主要是因為平台支持同一個 Route Map 到不同的 App，這樣就可以讓用戶透過同一個 url 訪問到不同版本的 APP。實際測試時，同時佈署 2 個版本的 App，保持相同的實例個數，並且 Map 相同的 url，就可以將使用者流量分別導向 2 個 App，然後透過統計用戶對不同版本的 App 的使用情況，最終確定選擇哪個方案更好。

6) CodePipeline 服務

CodePipeline 是在 Kubernetes + Jenkins 的基礎上開發的一個 SaaS 化的產品，提供視覺化的動作頁面，協助用戶簡單快捷地實現持續整合與持續交付的流程，服務結構設計如圖 3-9 所示，而 CodePipeline 服務提供的視覺化動作頁面如圖 3-10 所示。



CodePipeline 服務結構設計

The screenshot displays the CodePipeline console interface. At the top, there's a navigation bar with the CodePipeline logo and a user profile icon. Below that, a dropdown menu shows 'El-Workspace / All' and a '+ Add' button. A table lists pipelines with columns: Active, Name, Project, Status, Duration, and Operate. The 'Active' column has checkboxes, and the 'Operate' column has icons for refresh, edit, and delete. Below this, the 'BuildHistory' tab is selected, showing a 'Pipeline Builds' table with columns: Index, Status, CreateTime, Duration, Domain, and Space. The table contains five rows of build data.

Index	Status	CreateTime	Duration	Domain	Space
322	failed	2018-06-26 09:32:51	1 minute 14 seconds	wise-paas.com	ifactory
323	started	2018-06-26 09:34:34	N/A	N/A	pipeline
324	failed	2018-06-26 11:00:50	43 seconds	wise-paas.com	Production
325	succeeded	2018-06-26 11:05:01	3 minutes 7 seconds	wise-paas.com	Production
326	succeeded	2018-06-26 16:11:17	3 minutes 42 seconds	wise-paas.com	Production

CodePipeline 服務提供的視覺化動作頁面


(三) 解決方案特點

使用 Kubernetes + Jenkins 作為基礎架構，提高了服務可用性，並且實現資源按需使用，避免浪費；上層又封裝了 CodePipeline 服務，讓使用者打開服務即可使用，不需要自己搭建編譯和佈署等複雜的環境，也無需用戶維運：

- 服務高可用：當 Jenkins Master 出現故障時，Kubernetes 會自動創建一個新的 Jenkins Master 容器，並且將持久化的 Volume 分配給新創建的容器，保證資料不丟失，從而達到集群服務高可用性。

- 資源按需使用、動態生成、動態銷毀：Jenkins Agent 只有在 Job 運行時才會被創建，Job 運行結束後會自動銷毀以提高資源使用率，而且還可以防止源碼或者建構物外洩。





- 服務打開即用，無需額外的配置：打開 CodePipeline 服務連結，即可在上面透過配置完成 CI/CD 的過程，不需要自己搭建編譯和佈署等複雜的環境。

- 使用者無需關心整個交付系統的維運，由平台統一管理。

總結

(一) 經濟/社會效益

目前研華內部人員已經開始使用 CodePipeline 服務來完成部分 SRP 的持續交付過程，加速了軟體產品的交付流程，降低佈署過程出問題的風險，提高了客戶滿意度和產品的市場競爭力。

- 自動化軟體發佈流程
- 提高開發人員的工作效率
- 更快發現並解決缺陷
- 更快交付更新

(二) 用戶評價回饋

目前 CodePipeline 還有很多需要完善的地方，使用者在使用過程中也發現了一些 bug，目前正在修正中。系統本身也有很多功能需要完善，比如需要設計更完善的使用者操作許可權，整合 WISE-PaaS 雲平台的 SSO 系統，在使用方面也希望能增加一些更簡單的視覺化配置，讓使用者不用寫源碼就能快速完成編譯和佈署的工作。

(三) 小結

在快速發展的雲時代，軟體產品層出不窮，為搶奪先機，第一時間將產品上市，快速發佈和保證品質成為產品取勝至關重要的因素，而 DevOps 就是順應時代的潮流產生的一種軟體交付理念和方法，遵循 DevOps 的交付理念，可幫助開發人員第一時間發現軟體中的缺陷，保證產品快速和高品質上線，並提供不間斷的服務，提高客戶的滿意度





和產品的市場競爭力。所以 DevOps 已經成為軟體驅動型企業在雲時代取得成功的關鍵。

