# ECPaaS PVC Clone User Guide

## PVC Clone功能說明

### 功能說明

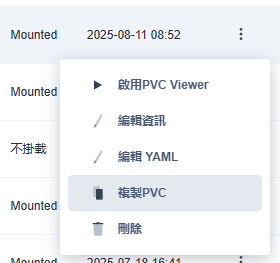
使用者可以對集群中的PVC進行複製，複製的方式是新建一個PVC，並將來源PVC中的檔案全數複製到新建的PVC複製體當中。

送出複製請求後，會建立兩個容器組和一個服務，這些資源的目的是為了將來源PVC中的檔案複製到複製PVC當中。一旦複製結束，不論成功或失敗，都會將結果記錄在複製PVC詳細資料頁面中的「Metadata」分頁的註釋(Annotation)當中，會以「ecpaas.io/pvc-clone-result」為key來紀錄複製結果，來源PVC或是其他方式建立的PVC就不會有這一筆註釋。過程中建立的兩個容器組和服務在複製結束之後都會自動被刪除。

此功能可以複製任何儲存類型(StorageClass)建立的PVC，不須透過建立PVC的儲存類型提供的Clone功能來複製。如果建立目標PVC的儲存類型支援Clone功能，也依然可以選擇使用此功能來複製PVC。

### 設定位置說明

在ECPaaS的「儲存」的「Persistent Volume Claims」頁面中，在PVC列表中找到想要複製的來源PVC，點擊PVC右側的「…」按鈕，可以在清單中找到「複製PVC」選項：



或是進入來源PVC的詳細資料頁面，點擊「更多操作」，也可以在清單中找到「複製PVC」選項：

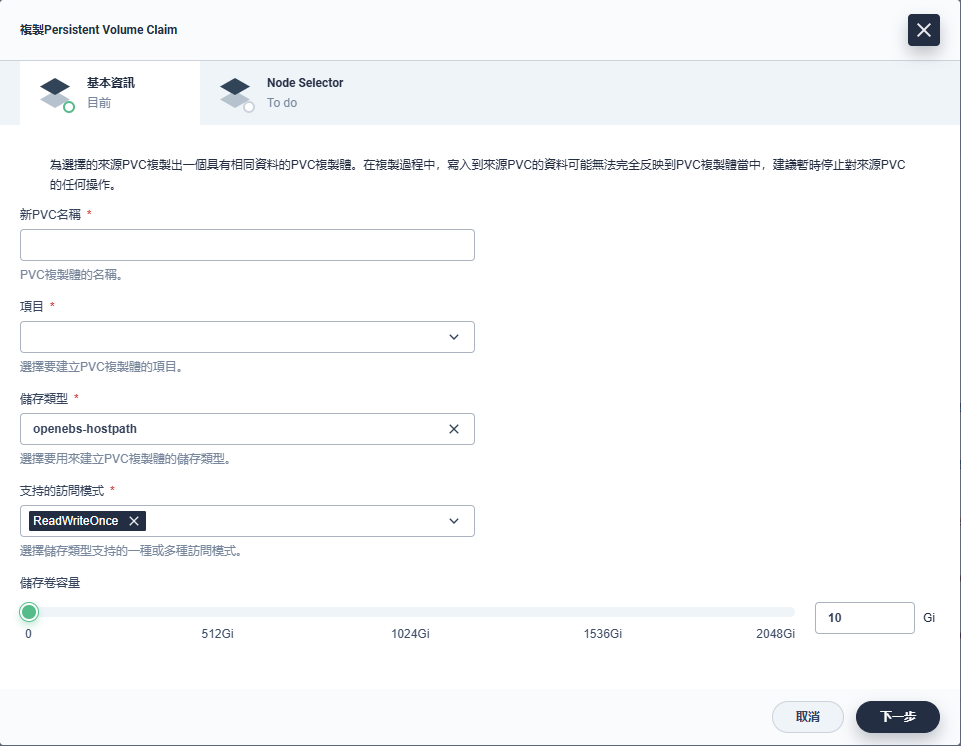


只有Bound狀態的PVC可以進行複製，所以如果選擇的PVC還未進入Bound狀態，在這兩個地方都會看不到複製PVC選項。

## 操作方式說明

### 欄位及頁面說明

透過兩種方式找到複製PVC按鈕之後，點擊後會出現複製PVC頁面：

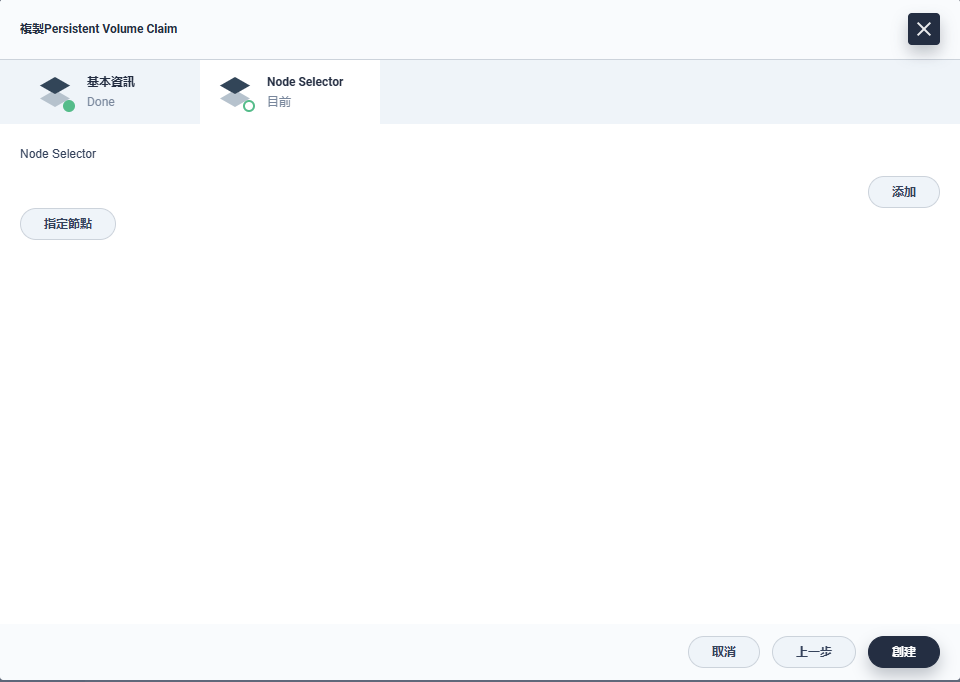


頁面中的設定和創建PVC時大致相同，但要注意在這裡設定的**新PVC名稱(New PVC Name)**與**項目(Project)**是指複製出來的PVC名稱以及所在的項目。複製PVC可以建立在與來源PVC不同的項目中。

**儲存類型(StorageClass)**：可以從集群中有安裝的儲存類型來選擇，這個欄位是一個下拉式選單，點擊後會自動列出所有可以選擇的儲存類型。複製PVC可以和來源PVC使用不同的儲存類型。

**支持的訪問模式(AccessModes)**：可以設定複製PVC所綁定PV的訪問模式，可以由以下三種模式中選擇：ReadWriteOnce、ReadOnlyMany、ReadWriteMany。可以設定一種或多種的模式。ReadWriteOnce模式代表PVC只能由一個節點讀取與修改。ReadOnlyMany模式可以同時被多個節點讀取，且只能讀取。ReadWriteMany模式則可以同時被多個節點讀取與修改。

**儲存卷容量(Volume Capacity)**：設定複製PVC的容量大小，這個大小可以依照實際需求來設定，但要注意如果設定的比來源PVC中的合計檔案大小還小的話，複製可能會失敗。可以透過移動調整桿來設定，或是在右側的輸入框中直接輸入數字。可以設定的範圍是1 ~ 2048的整數，單位是GiB。



第二頁是**Node Selector**設定頁面，如果使用者希望和複製PVC綁定的PV建立在特定節點上，可以使用Node Selector來設定。但是實際上會與PVC綁定的PV的建立位置還是會以儲存類型以及Kubernetes本身來決定，如果設定的節點無法建立PV，那複製PVC可能就會一直卡在Pending狀態。

可以點擊「添加」自行設定Key與Value，或是點擊「指定節點」按鈕，在出現的視窗中點選集群中的節點來快速設定：



選擇節點並點擊確定後，會以節點名稱做為Node Selector的內容：



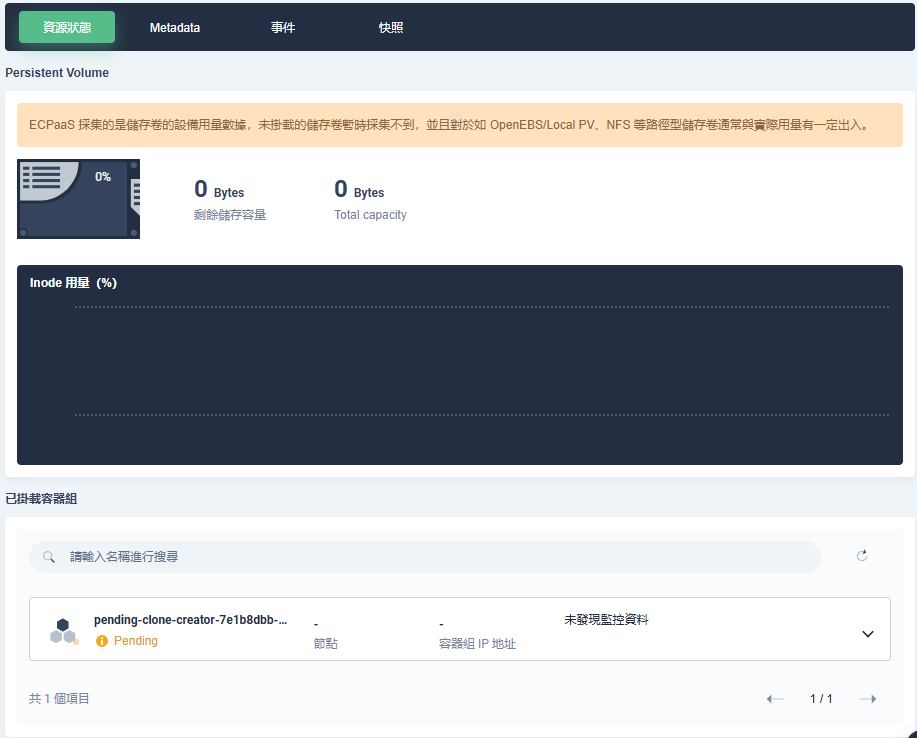
設定完需要的欄位後，點擊右下角的創建按鈕就可以建立複製PVC，並且會自動開始複製來源PVC中的檔案到複製PVC當中。

### 確認複製結果

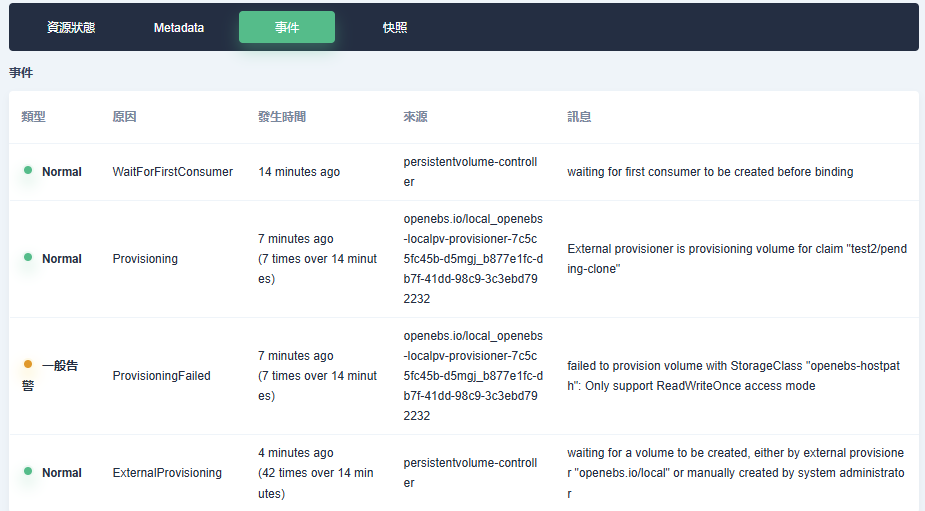
如果複製過程順利，可以在複製PVC的Metadata註釋中看到複製指令的執行狀況，如：



根據來源PVC中的檔案數量及大小，可能會需要更長的時間來完成，如果一直無法在註釋中看到「ecpaas.io/pvc-clone-result」紀錄出現，可以到「資源狀態」和「事件」分頁中檢查容器組的執行狀況，如以下例子：



在複製PVC的「資源狀態」分頁，已掛載容器組區塊中可以看到有一個容器組處在Pending狀態，這個容器組是用來將資料複製進PVC，複製資料用的容器組名稱格式會是「<複製PVC名稱>-creator-<UID>」，pending狀態代表複製尚未開始。導致容器組無法建立的原因有很多種，如果是有設定了Node Selector的情況，可以看看是否設定的Node Selector在集群中找不到對應的節點，這個可以在容器組的詳細資料中查看。另外可以在複製PVC的「事件」分頁中，查看設定的儲存類型與訪問模式是否有問題，如：

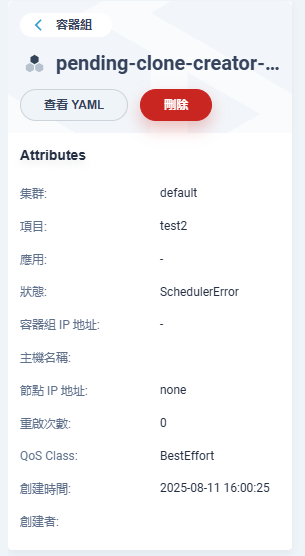


在這裡可以看到因為設定了不支援的訪問模式，導致PV無法建立，最終導致複製用的容器組無法建立。

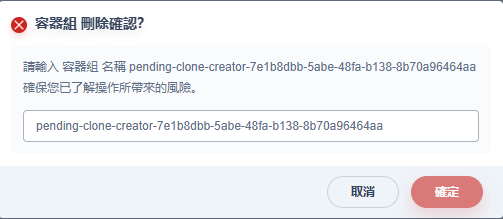
### 中止操作方式說明

如果因為設定或是集群中的問題導致的複製過程無法順利執行，或者只是單純想要中止複製PVC過程的話，使用者可以透過刪除複製PVC上的複製用容器組來達成。在建立複製PVC之後，進入複製PVC的詳細資料頁面，「資源狀態」分頁中的「已掛載容器組」區塊找到複製用的容器組，點擊名稱可以進入容器組的詳細資料頁面：

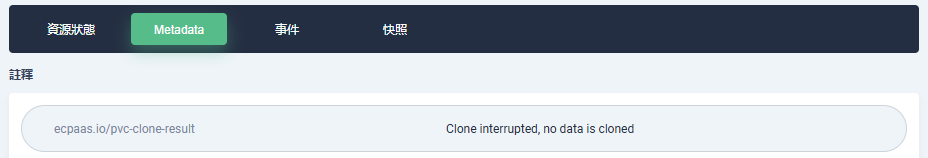




點擊左上角區塊中的紅色刪除按鈕，輸入名稱，並點擊確定，即可開始刪除複製用的容器組：



刪除容器組可能會需要一點時間，回到複製PVC的詳細資料頁面，等待複製用的容器組從「已掛載容器組」區塊中消失後，複製過程就會被中止。進入「Metadata」分頁，可以看到註釋中出現了新的內容，說明複製過程被中止，且沒有任何資料被複製：



Mayastor卡住在mayastor-single-replica的儲存體

**检查并强制删除 Pod（如果存在）**

首先，确认是否有 Pod 还在使用这个 PVC。即使 Pod 显示为已删除，有时其引用可能残留。

1. **检查关联 Pod**：

bash

kubectl get pods --all-namespaces -o=jsonpath='{range .items[?(@.spec.volumes)]}{.metadata.namespace}{"/"}{.metadata.name}{"\n"}{range .spec.volumes}{}{"\t"}{.persistentVolumeClaim.claimName}{"\n"}{end}{end}' | grep -i "<your-pvc-name>"

root@ecpaas42221:/home/ecpaas# kubectl get pods --all-namespaces -o=jsonpath='{range .items[?(@.spec.volumes)]}{.metadata.namespace}{"/"}{.metadata.name}{"\n"}{range .spec.volumes}{}{"\t"}{.persistentVolumeClaim.claimName}{"\n"}{end}{end}' | grep -i "<your-pvc-name>"

root@ecpaas42221:/home/ecpaas# kubectl get pods --all-namespaces -o=jsonpath='{range .items[?(@.spec.volumes)]}{.metadata.namespace}{"/"}{.metadata.name}{"\n"}{range .spec.volumes}{}{"\t"}{.persistentVolumeClaim.claimName}{"\n"}{end}{end}' | grep -i "d1"

kubevirt/d1-creator-b6f6a1f9-d5d7-4465-acfd-0971c35674ab

[{"name":"target","persistentVolumeClaim":{"claimName":"d1"}},{"name":"kube-api-access-j7gwd","projected":{"defaultMode":420,"sources":[{"serviceAccountToken":{"expirationSeconds":3607,"path":"token"}},{"configMap":{"items":[{"key":"ca.crt","path":"ca.crt"}],"name":"kube-root-ca.crt"}},{"downwardAPI":{"items":[{"fieldRef":{"apiVersion":"v1","fieldPath":"metadata.namespace"},"path":"namespace"}]}}]}}]

root@ecpaas42221:/home/ecpaas# kubectl delete pod d1-creator-b6f6a1f9-d5d7-4465-acfd-0971c35674ab -n kubevirt --force --grace-period=0

Warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.

pod "d1-creator-b6f6a1f9-d5d7-4465-acfd-0971c35674ab" force deleted

root@ecpaas42221:/home/ecpaas# kubectl get pods --all-namespaces -o=jsonpath='{range .items[?(@.spec.volumes)]}{.metadata.namespace}{"/"}{.metadata.name}{"\n"}{range .spec.volumes}{}{"\t"}{.persistentVolumeClaim.claimName}{"\n"}{end}{end}' | grep -i "d1"