

Agregue los nuevos botones de acción en mis servidores

Contenido

[Introducción](#)

[Antes de comenzar](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Agregar los nuevos botones a mi servidor](#)

[Diseñador IAC y del servicio](#)

[Agregar un nuevo botón](#)

[Información Relacionada](#)

[Introducción](#)

Este documento explica cómo agregar un nuevo botón en mi popover de los servidores y atarlo con un nuevo servicio.

[Antes de comenzar](#)

[Requisitos](#)

No hay requisitos específicos para este documento.

[Componentes Utilizados](#)

La información en este documento se basa en estas versiones de software y hardware.

- La configuración siguiente se aplica a las versiones 4.0 IAC.

[Agregar los nuevos botones a mi servidor](#)

Crear un nuevo servicio o la modificación de un servicio existente está fuera del alcance de este documento.

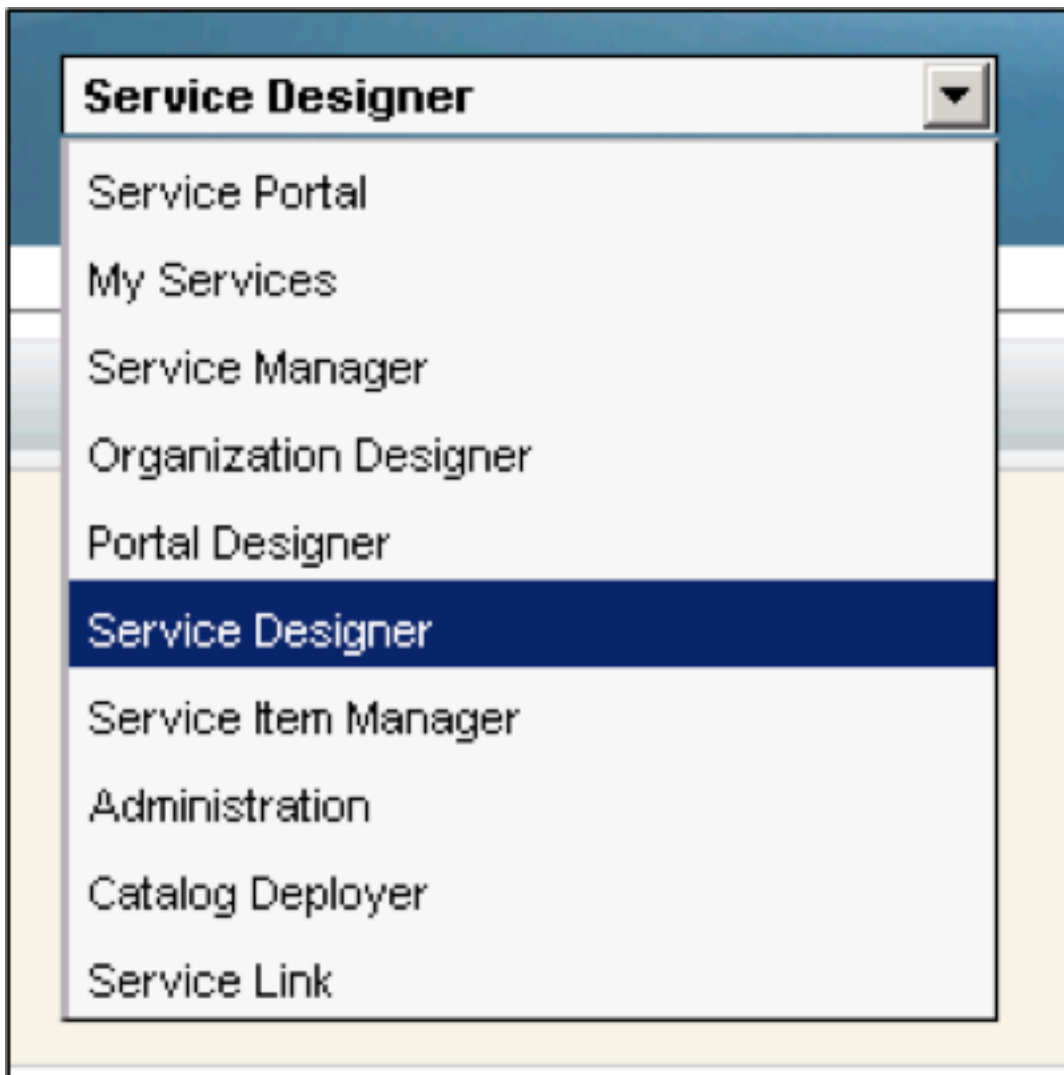
[Diseñador IAC y del servicio](#)

El procedimiento para crear los nuevos botones en mis servidores requiere el uso del diseñador del servicio.

Agregar un nuevo botón

El siguiente ejemplo ilustra los pasos para agregar un nuevo botón “para liberar la dirección IP flotante.”

Paso 1: Entre al diseñador del servicio como se muestra abajo en el diagrama 1.



Paso 2: Cree un servicio y salve su nombre para uso futuro.

- En el diagrama 2 mostrado abajo, el nombre “dirección IP flotante de la versión” se asigna al nuevo botón y es con diferenciación entre mayúsculas y minúsculas.

Paso 3: Localice el directorio del despliegue IAC y abra el archivo “my-servers.html” para editar.

- “\ RequestCenterServer \ implementaciones \ RequestCenter.war \ aduana \ IAC \ portlets \ my-servers.html”

Paso 4: Agregue la línea nueva al final de la sección de ServiceIds como se muestra abajo en el diagrama 3.

NOTA: El nombre del servicio es con diferenciación entre mayúsculas y minúsculas, y utiliza el “serviceId” como segundo argumento.

```
223     serviceIds.push(getServiceProperties("Power Down Physical Server", "serviceId"));
224     serviceIds.push(getServiceProperties("Power Cycle Physical Server", "serviceId"));
225     serviceIds.push(getServiceProperties("Decommission Physical Server", "serviceId"));
226
227     serviceIds.push(getServiceProperties("Allocate Floating IP Address", "serviceId"));
228     serviceIds.push(getServiceProperties("Delete VM Firewall Rule", "serviceId"));
229     serviceIds.push(getServiceProperties("Release Floating IP Address", "serviceId"));
230
```

Paso 5: Abra el archivo “IACServersPopover.js” para editar eso está situado en “\ src \ aparato \ iac \ IACServersPopover.js”

- Encuentre la declaración de variable: “buttonHTML”. Debe ser el número de línea cercano 160 como se muestra abajo en el diagrama 4.

```

159
160 var buttonHTML = '';
161 if(status == "INACTIVE") {
162
163     buttonHTML = "<p style=\`font-weight:bold; color: #455A73; margin: 5px 0 0 10px;\`>This "+serve
164 }
165 else {
166
167     if(serverType == "Virtual Machine (VM)" && powerState == "Off") { Start of conditional block
168
169         if(podType == "OpenStack Cloud Manager"){
170             serviceNames = ["Power Up Virtual Machine", "Modify Configuration", "View Snapshots",
171                 serviceIDs = [idsArray[0], idsArray[3], idsArray[4], idsArray[5], idsArray[6], idsArray
172                 cssClassNames = ["powerUp", "modifyConfiguration", "viewSnapshots", "takeSnapshot", "
173         }
174         else if(podType == "Amazon EC2") {
175             serviceNames = ["Power Up Virtual Machine", "View Snapshots", "Take Snapshot", "Revert
176             serviceIDs = [idsArray[0], idsArray[4], idsArray[5], idsArray[6], idsArray[7], idsArra
177             cssClassNames = ["powerUp", "viewSnapshots", "takeSnapshot", "revertSnapshot", "deleteS
178         }
179         else {
180             //All actions except Power Down, Power Cycle
181             serviceNames = ["Power Up Virtual Machine", "Modify Configuration", "View Snapshots", "
182             serviceIDs = [idsArray[0], idsArray[3], idsArray[4], idsArray[5], idsArray[6], idsArray
183             cssClassNames = ["powerUp", "modifyConfiguration", "viewSnapshots", "takeSnapshot", "re
184             if(podType == "VMware vCenter Server") {
185                 serviceNames.push("Convert VM to Template");
186                 serviceIDs.push(idsArray[10]);
187                 cssClassNames.push("convertVMtoTemp");

```

- Después de la declaración del "buttonHTML" son las condiciones adicionales dispuestas por el tipo, el estado y los rol del usuario VM.

Paso 6: Inserte el código siguiente con las condiciones apropiadas para mostrar la acción para cada papel específico y el tipo PE, antes de llamar el código de los "generateActionButtons" (situado antes de la línea no.: 198) como se muestra abajo en el diagrama 5.

```

201
202     serviceNames.push("Release Floating IP Address");
203     serviceIDs.push(idsArray[20]);
204     cssClassNames.push("releaseFIP");

```

- En los serviceNames, avance el nombre del servicio, en este ejemplo "dirección IP flotante de la versión."
- En los serviceIDs, avance el valor del índice apropiado de idsArray, en este ejemplo que es el índice 20, la búsqueda más reciente del servicio de los "getServiceProperties" en my-servers.html.
- En los cssClassNames, avance la clase CSS de botón.

Paso 7: El abrir archivo "IACGenerateActionButtons.js" para editar y encuentra la función: "generateMyServersGroupButtons".

- En el extremo de este loop de la función mostrado abajo en el diagrama 6, cambie los valores resaltados según su servicio:

```
else if (serviceNames[i] == "Release Floating IP Address" && add
extra validations if required){
    networkServicesGroup+="<div class=\"popover_main\"
id=\"releaseFIPCont\"><span class=\"popover_sub\"><button type=\"button\"
id=\"releaseFIP\" class=\"\" + cssClassNames[i] + \"\" title=\"\" + serviceNames[i] +
\"\"+ alt=\"\" + serviceNames[i] + \"\"></button></span><span class=\"popover_label
popover_content\" id=\"releaseFIPLabel\">"+serviceNames[i]+</span></div>;
}
```

Usted puede cambiar el networkServicesGroup a uno del siguiente basado en sus requisitos:

- lifecycleManagementGroup
- networkServicesGroup
- serverOpsGroup

Paso 8: Se crea la estructura, se ejecuta y despliega para verificar el nuevo botón.

[Información Relacionada](#)

- [Cisco Intelligent Automation for Cloud](#)