

# Добавьте новые командные кнопки в моих серверах

## Содержание

[Введение](#)

[Перед началом работы](#)

[Требования](#)

[Используемые компоненты](#)

[Добавление новых кнопок к моему серверу](#)

[IAC и сервисный разработчик](#)

[Добавление новой кнопки](#)

[Дополнительные сведения](#)

## [Введение](#)

Этот документ объясняет, как добавить новую кнопку в Моей легкой сдобе Серверов и связать ее с новым сервисом.

## [Перед началом работы](#)

### [Требования](#)

Для этого документа отсутствуют особые требования.

### [Используемые компоненты](#)

Сведения в документе приведены на основе данных версий аппаратного и программного обеспечения.

- Следующая конфигурация применяется к версиям IAC 4.0.

## [Добавление новых кнопок к моему серверу](#)

Создание нового сервиса или изменение существующего сервиса выходят за рамки этого документа.

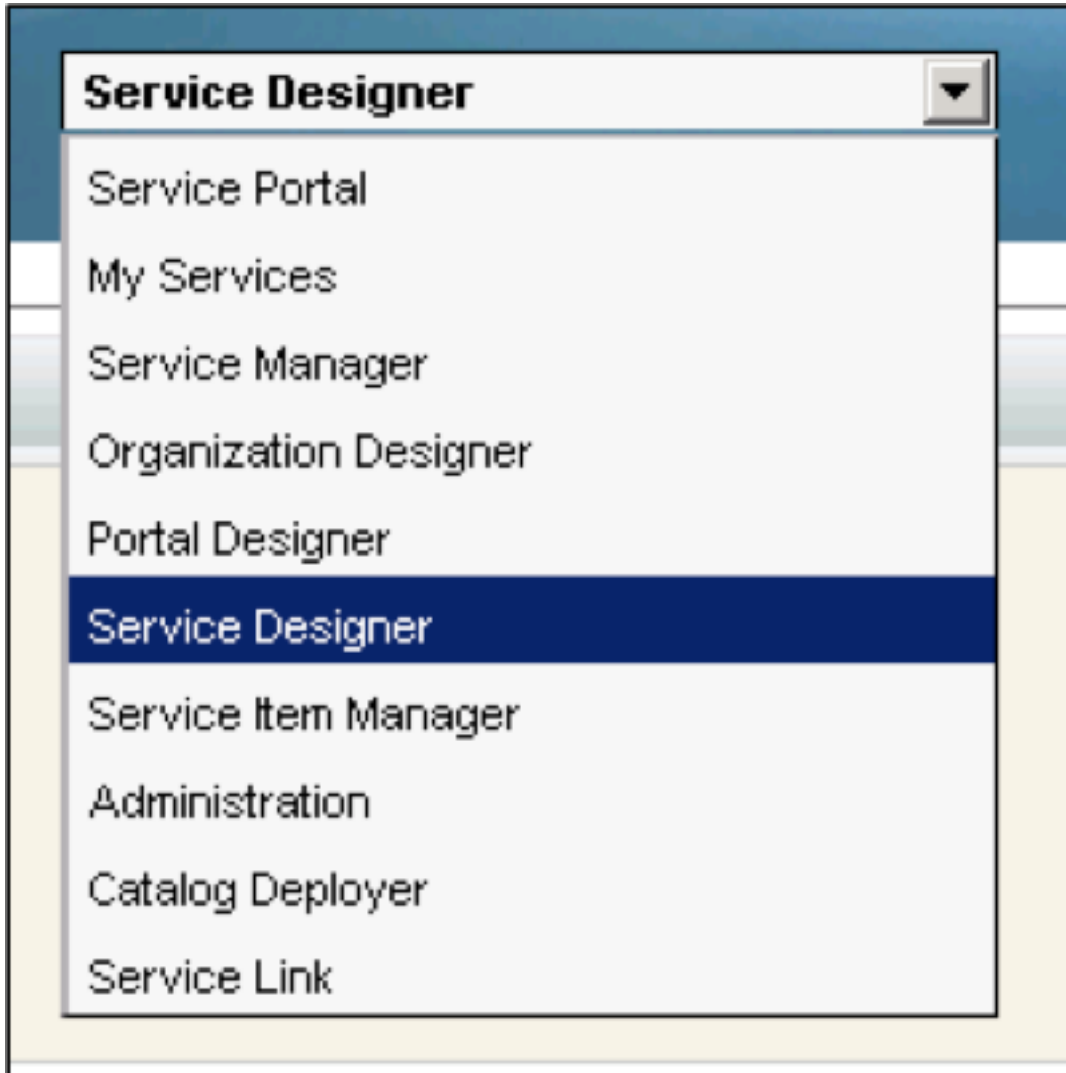
### [IAC и сервисный разработчик](#)

Процедура для создания новых кнопок в Моих Серверах требует использования Сервисного Разработчика.

## Добавление новой кнопки

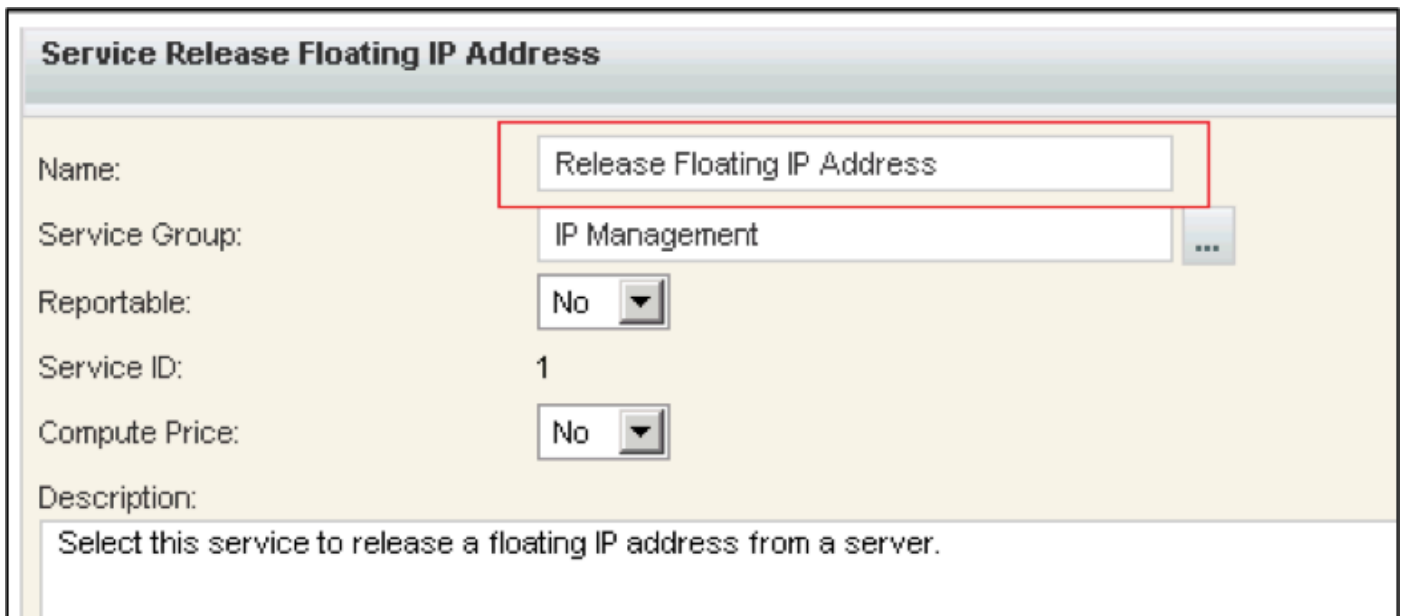
Следующий пример иллюстрирует шаги для добавления новой кнопки для "Выпуска Плавающего IP-адреса".

Шаг 1: Перейдите к Сервисному Разработчику как показано ниже в Схеме 1.



Шаг 2: Создайте сервис и сохраните его название на дальнейшее использование.

- В Схеме 2, показанной ниже, название "Плавающий IP-адрес Выпуска" назначено на новую кнопку и учитывает регистр.



**Service Release Floating IP Address**

Name: Release Floating IP Address

Service Group: IP Management

Reportable: No

Service ID: 1

Compute Price: No

Description: Select this service to release a floating IP address from a server.

Шаг 3: Найдите каталог развертываний IAC и откройте файл "мой-servers.html" для редактирования.

- "\\RequestCenterServer\deployments\RequestCenter.war\custom\IAC\portlets\my-servers.html"

Шаг 4. : Добавьте новую линию до конца раздела ServiceIds как показано ниже в Схеме 3.

Примечание: Имя сервиса учитывает регистр и использует "serviceId" в качестве второго аргумента.

```

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

```

Шаг 5. : Откройте файл "IACServersPopover.js" для редактирования, которое расположено в "\\src\widget\iac\IACServersPopover.js"

- Найдите объявление переменной: "buttonHTML". Это должен быть близкий номер строки 160 как показано ниже в Схеме 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");

```

- После "buttonHTML" объявления дополнительные условия, расположенные Типом VM, Состоянием и Ролями пользователя.

Шаг 6: Введите следующий код с соответствующими условиями показать Действие для каждой определенной Роли и Типа PE, прежде, чем вызвать код "generateActionButtons" (расположенный перед номером линии: 198) как показано ниже в Схеме 5.

```

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

```

- В serviceNames выдвиньте имя сервиса в данном примере "Плавающий IP-адрес Выпуска".
- В serviceIDs выдвиньте надлежащее значение индекса idsArray в данном примере, в котором это является индекс 20, последняя сервисная выборка из "getServiceProperties" мой-servers.html.
- В cssClassNames выдвиньте класс CSS кнопки.

Шаг 7: Открытый файл "IACGenerateActionButtons.js" для редактирования и находит функцию: "generateMyServersGroupButtons".

- В конце этой функциональной петли, показанной ниже в Схеме 6, измените выделенные значения согласно своему сервису:

```
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>;
    }
```

Можно изменить networkServicesGroup на один из следующих на основе требований:

- lifecycleManagementGroup
- networkServicesGroup
- serverOpsGroup

Шаг 8: Сборка, работайте и разверните, чтобы проверить, что создана новая кнопка.

## [Дополнительные сведения](#)

- [Cisco Intelligent Automation for Cloud](#)