

CallVU Digital Engagement Platform

**AWS Visual Connect QuickStart Guide**

The CallVU Digital Engagement Platform augments the customer experience of Amazon Connect user by providing a visual in-call experience, empowering caller’s with self-service capabilities typically not available in traditional phone interactions.

The Amazon Connect IVR drives both the voice and the digital/visual interactions as a single contact flow.

This document focuses on the post-deployment implementation stages for creating visual interactions. These stages are:

1. Creation of ‘screens’ for in-call, visual interaction
2. Integration of CallVU visual interaction capabilities in an Amazon Connect callflow.

# CallVU Screen Creation for Visual Sessions on AWS

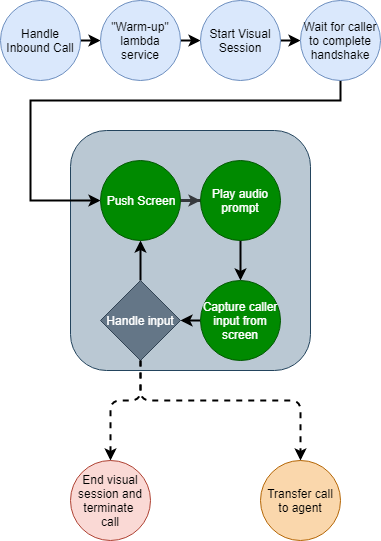
During this stage of implementation, you will create the screens that will be displayed to the caller at various stages of a phone call through Amazon Connect. CallVU provides a Screen Builder utility on one of the EC2 servers created during cloud deployment. The following steps will allow you to access this utility:

1. Using Remote Desktop (RDP), connect to the “external” EC2 server created during the cloud deployment.
   1. When prompted for a password, use the password you provided during cloud formation.
   2. If you did not provide a password during cloud formation, you can login with a default password of: **P@ssword**
2. Launch Google Chrome from within the remote desktop session.
3. Browse to the following URL: <http://localhost/cv-screen-builder>
4. You should now see the CallVU Screen Builder in your browser. Please see this document to get you started on building your Visual IVR Screens:   
   [https://callvu.net/documentation/Screen Builder Manual.pdf](https://callvu.net/documentation/Screen%20Builder%20Manual.pdf)

Once you have created a few screens using the CallVU Screen Builder, proceed to the next section of this document to integrate your screens with your Amazon Connect callflow.

# Integrating CallVU Visual Interactions in an Amazon Connect Callflow

Integrating the CallVU Visual IVR platform with Amazon Connect consists of communication between Amazon Connect and CallVU via web services. These web services are brokered by lambda functions provided by the cloud formation. The integration of these technologies provides the caller with what we will henceforth refer to as a “Visual Call”. The typical IVR control flow for a visual call is as follows:



To implement the actions in the control flow illustrated above, CallVU provides a lambda function called **SetDynamicDisplay**, which has the following methods:

HealthCheck  
This method simply returns a string with the current CallVU platform status. In Amazon Connect implementations, this method is used to ‘warm-up’ the lambda service, which is necessary to correctly start a Visual Call.

SET  
The SET method tells CallVU to push a given screen to the caller during a visual session. This method is also used to initiate the Visual Call the first time it is used in a callflow.

ISWEBCLIENTSTARTED  
This method returns a Boolean true/false depending on whether or not the caller has begun a Visual Call (in other words, if the user tapped the link provided via SMS or not).

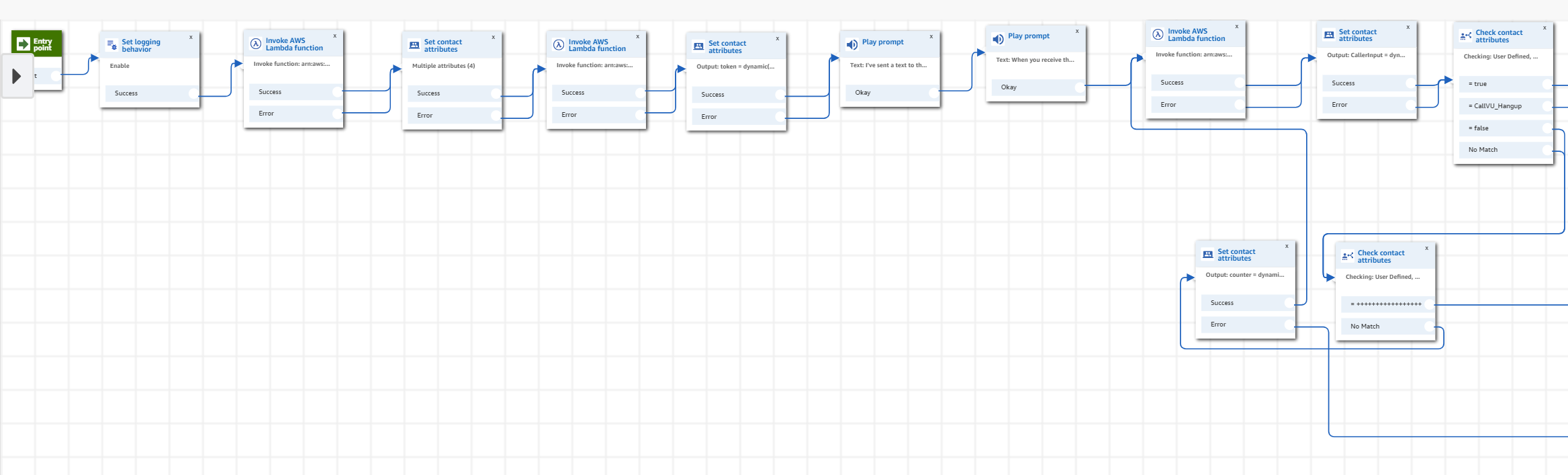
RCV  
This method returns input from the caller provided to the Visual Call screen. There are four possible return values for the RCV method:

* String containing a token and the caller input from the screen. Separate values are delimited with a semicolon.
* If the visual session has terminated, the return value is Boolean **false**.
* If the caller tapped the hangup button on a Visual Call screen, the return value is the string **CallVU\_Hangup**.
* If the RCV method times-out with no input from the caller, the return value is the string **NoInput**.

## Getting Started with a Sample Visual Callflow

Let’s start with a sample project to illustrate how to use the aforementioned CallVU methods in an Amazon Connect Callflow. To do this:

1. Download this sample callflow to your computer:  
   <https://callvu.net/documentation/SampleFlow.txt>
2. In Amazon Connect, create a new Callflow.
3. Open your new callflow.
4. Import the callflow you downloaded in step 1 into this callflow.
5. You should now see a pre-build diagram that looks like this:



Now that you have the sample visual flow imported into your Amazon Connect instance, let’s look at how this flow is configured.

### Warming up the lambda function

Click on the first “Invoke AWS Lambda function” box to display its properties. You should see that the function input parameters consist of a key=value pair where the key is set to *method* and the value is set to *HealthCheck*. Additionally, you should notice that the function being invoked is currently a placeholder for the lambda function provided in your cloud formation deployment.

Replace the string *SetDynamicDisplay ARN HERE* with the actual ARN of the SetDynamicDisplay lambda function in your AWS environment.

The invocation of this HealthCheck function performs a lambda ‘warm-up’. This is a sacrificial web service call that will always fail. However, this action must occur for any subsequent lambda calls to succeed in the callflow. For this reason, you must perform this step at the beginning of every visual callflow.

### Instantiate Visual Call Variables

Before the Visual Call can be established, we must first instantiate some variables that will be used throughout the callflow. Click on the first “Set contact attributes” block in the callflow. Here, we have set the following variables:

CallerInput  
This variable will be used as a temporary location to store any caller input from the Visual Call screens once it has been submitted. It is instantiated to a default value of **NoInput**.

token  
This variable will hold the CallVU session token for the Visual Call, which is returned every time a screen is pushed to the caller. However, it is instantiated with a special value – **starthtml5**. This value tells CallVU to start the Visual Call when the first screen ‘push’ is made in the callflow.

number  
This variable holds the inbound caller’s phone number. This is necessary for all invocations of CallVU web methods.

counter  
This variable contains a string that indicates the number of web service retries that have been performed in any of the loops in the callflow. We will look at this in more detail later. It is instantiated to the plus character **+**.

### 