🚀 LiveKit Cloud Plugin

CLOUD LIVEKIT

bitHuman LiveKit Cloud Plugin Integration

Use existing bitHuman agents in real-time applications with our cloud-hosted LiveKit plugin featuring Essence (CPU) and Expression (GPU) models.

🚀 Quick Start

1. Install Cloud Plugin

```
# Uninstall existing plugin
uv pip uninstall livekit-plugins-bithuman

# Install cloud plugin from GitHub
GIT_LFS_SKIP_SMUDGE=1 uv pip install git+https://github.com/livekit/
agents@main#subdirectory=livekit-plugins/livekit-plugins-bithuman
```

2. Get API Credentials

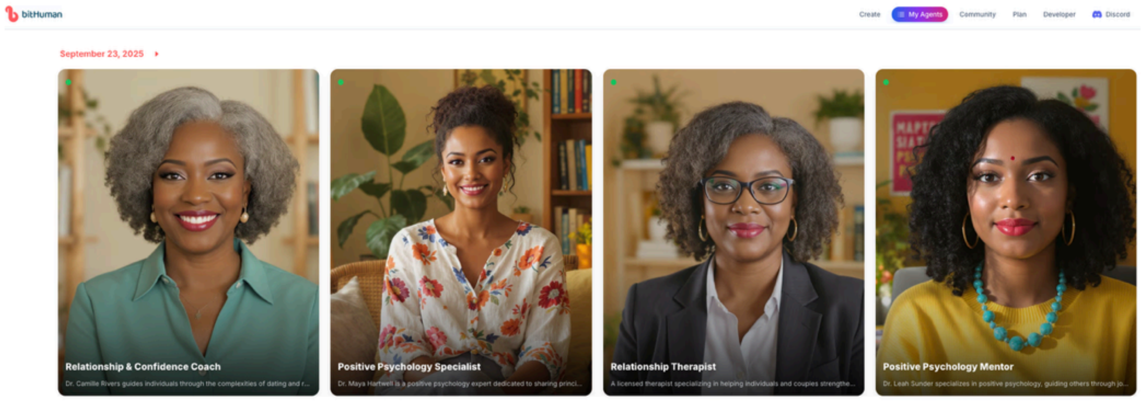
- API Secret: imaginex.bithuman.ai

3. Find Your Agent ID

To use an existing avatar with the Expression Model, you'll need to locate your agent ID from the bitHuman platform.

Step 1: Select Your Agent

Navigate to your imaginex.bithuman.ai dashboard and click on the agent card you want to use.



Relationship & Confidence Coach
Dr. Camille Rivers guides individuals through the complexities of dating and r...

Positive Psychology Specialist
Dr. Maya Hartwell is a positive psychology expert dedicated to sharing princi...

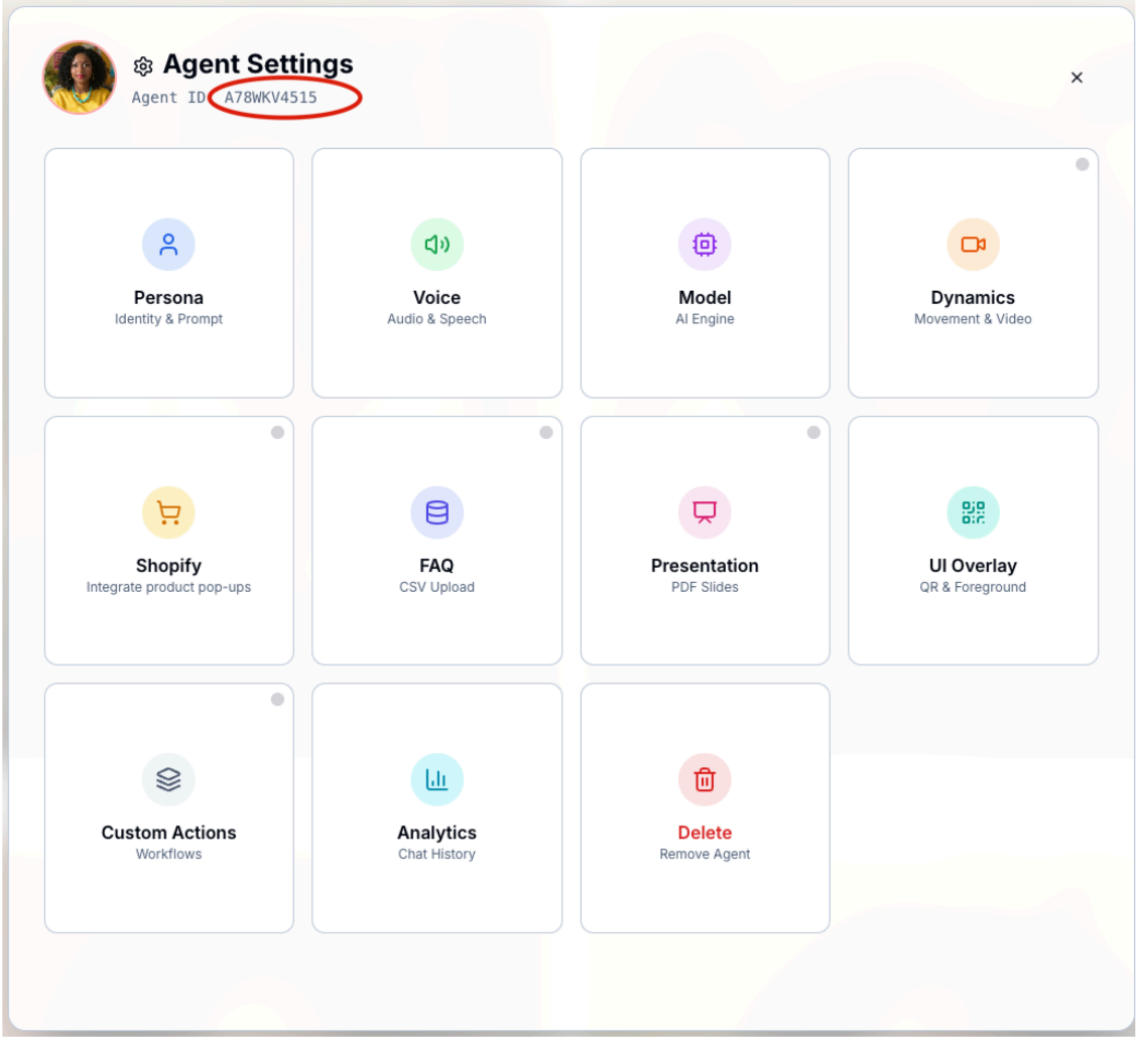
Relationship Therapist
A licensed therapist specializing in helping individuals and couples strengthe...

Positive Psychology Mentor
Dr. Leah Sunder specializes in positive psychology, guiding others through jo...

Click on the agent card you want to use for integration

Step 2: Access Agent Settings

Once you click on the agent, the Agent Settings dialog will open, displaying your unique Agent ID at the top.



Agent Settings

Agent ID A78WKV4515

Copy the Agent ID from the Agent Settings dialog

💡 Tip: The Agent ID (e.g., `A78WKV4515`) is a unique identifier for your specific avatar. You'll use this as the `avatar_id` parameter in your code.

4. Set Environment

```
export BITHUMAN_API_SECRET="your_api_secret"
```

Usage Examples

**Essence Model (CPU) **

For standard avatar interactions with built-in personalities:

```
import bithuman

# Create avatar session with essence model
bithuman_avatar = bithuman.AvatarSession(
    avatar_id="your_agent_code",
    api_secret="your_api_secret",
)

# Start conversation
response = bithuman_avatar.generate_response("Hello, how are you?")
```

Expression Model (GPU) - Agent ID

For custom avatars created through the platform (see Find Your Agent ID above for instructions):

```
import bithuman

# Create avatar session with expression model
bithuman_avatar = bithuman.AvatarSession(
    avatar_id="your_agent_code",
    api_secret="your_api_secret",
    model="expression"
)

# Generate avatar response
response = bithuman_avatar.generate_response("Tell me about yourself")
```

Expression Model (GPU) - Custom Image

For dynamic avatar creation using custom images:

```
import bithuman
import os
from PIL import Image

```

```
# Create avatar session with custom image
bithuman_avatar = bithuman.AvatarSession(
    avatar_image=Image.open(os.path.join("your_image_path")),
    api_secret="your_api_secret",
    model="expression"
)

# Process custom image and generate response
response = bithuman_avatar.generate_response("Describe what you see")
```

Configuration Options

Avatar Session Parameters

Parameter	Type	Required	Description
avatar_id	string	Yes*	Unique identifier for pre-created avatar
avatar_image	PIL.Image	Yes*	Custom image for dynamic avatar creation
api_secret	string	Yes	Authentication secret from bitHuman platform
model	string	No	Model type: "essence" (default) or "expression"

*Either avatar_id or avatar_image is required, not both.

Model Types

Essence Model:

- Pre-trained personalities and behaviors
- Optimized for conversational AI

- Faster response times
- Supports full body and animal mode

Expression Model:

- Dynamic facial expression mapping
- Image-based avatar generation
- Supports only face and shoulder & above
- Do not support animal mode at the moment

🌐 Cloud Advantages

✅ No Local Storage - No need to download large model files

✅ Auto-Updates - Always use the latest model versions

✅ Scalability - Handle multiple concurrent sessions

✅ Performance - Optimized cloud infrastructure

✅ Cross-Platform - Works on any device with internet

⚒️ Advanced Integration

Session Management

```
import bithuman

class AvatarManager:
    def __init__(self, api_secret):
        self.api_secret = api_secret
        self.sessions = {}

    def create_session(self, session_id, avatar_id, model="essence"):
        self.sessions[session_id] = bithuman.AvatarSession(
            avatar_id=avatar_id,
            api_secret=self.api_secret,
            model=model
        )
        return self.sessions[session_id]

    def get_response(self, session_id, message):
        if session_id in self.sessions:
            return self.sessions[session_id].generate_response(message)
        return None

# Usage
manager = AvatarManager("your_api_secret")
session = manager.create_session("user_123", "avatar_456")
response = manager.get_response("user_123", "Hello!")
```

Error Handling

```
import bithuman

try:
    avatar = bithuman.AvatarSession(
        avatar_id="your_agent_code",
        api_secret="your_api_secret"
    )

    response = avatar.generate_response("Test message")

except bithuman.AuthenticationError:
    print("Invalid API secret. Check your credentials.")

except bithuman.QuotaExceededError:
    print("API quota exceeded. Upgrade your plan.")

except bithuman.NetworkError:
    print("Network connectivity issues. Check internet connection.")

except Exception as e:
    print(f"Unexpected error: {e}")
```

Monitoring & Debugging

Enable Logging

```
import logging
import bithuman

# Enable debug logging
logging.basicConfig(level=logging.DEBUG)
```

```
logger = logging.getLogger('bithuman')

avatar = bithuman.AvatarSession(
    avatar_id="your_agent_code",
    api_secret="your_api_secret",
    debug=True
)
```

Performance Metrics

```
import time
import bithuman

avatar = bithuman.AvatarSession(
    avatar_id="your_agent_code",
    api_secret="your_api_secret"
)

start_time = time.time()
response = avatar.generate_response("Performance test")
response_time = time.time() - start_time

print(f"Response generated in {response_time:.2f} seconds")
```

🚨 Common Issues

Authentication Errors:

- Verify API secret from imaginex.bithuman.ai
- Check environment variable is properly set

Network Timeouts:

- Ensure stable internet connection
- Consider implementing retry logic for production use

Model Loading Issues:

- Verify avatar_id exists in your account
- For expression model, ensure image format is supported (PNG, JPG, WEBP)

Plugin Installation:

- Use `uv` package manager as shown in installation
- Ensure `GIT_LFS_SKIP_SMUDGE=1` flag is included

🎯 Perfect for

✅ Production Applications - Reliable cloud infrastructure

✅ Scalable Solutions - Handle thousands of concurrent users

✅ Mobile Applications - No local storage requirements

✅ Enterprise Integration - Professional-grade API

✅ Rapid Prototyping - Quick setup without model management

📊 Pricing & Limits

Visit imaginex.bithuman.ai for current pricing and usage limits.

Free Tier Includes:

- 199 credits per month
- Community support

Pro Features:

- Unlimited credits
- Priority support
- Custom model training

➡️ Next Steps

API Documentation: Agent Generation API
Local Examples: Examples Overview
Community Support: Discord