

Langflow Exercise: Build a Multi-Agent Content Creation Pipeline

☐ Problem Statement

You are building a content creation pipeline using multiple collaborating agents in Langflow's visual UI. The pipeline should:

- Research a topic
 - Develop a strategy
 - Generate content
 - Review and polish it
-

Step 1: Plan Agents and Tasks

Define Agents

Plan at least **3–5 agents**, such as:

Agent Name	Role	Goal
ResearchBot	Topic Researcher	Gathers relevant insights on a given topic
StrategyBot	Marketing Strategist	Forms a communication strategy based on research
CopyBot	Content Generator	Writes content pieces (ad copy, blog, etc.)
ReviewBot	Quality Checker	Reviews and polishes content
DirectorBot	Decision Maker	Makes final approval decisions

Step 2: Create Agents in Langflow

1. Drag in **nodes** (e.g., `ConversationalAgent`, `SimpleAgent`) for each role.

2. For each agent, set:

- **System Prompt:** Include role, goal, and tone (e.g., "You are a skilled marketing strategist...")
 - **LLM Node:** Attach an OpenAI-compatible LLM (like [OpenAI](#), [ChatOpenAI](#), etc.)
-



Step 3: Define Tasks (using Chat or Function Chains)

1. Create a **chain of agents** using:

- [ChatBox](#) for simple user inputs
- [PromptTemplate](#) + LLM + [TextOutput](#) for step outputs
- [Memory](#) or [Variable](#) nodes to pass project context

2. Connect agents using:

- [Output of Agent A](#) → [Input to Agent B](#)

3. Use variables like [{project_topic}](#), [{target_audience}](#) across the flow.



Step 4: Run and Iterate

1. Add a **Markdown Display Node** or [TextOutput](#) node to show final content.
 2. Test the pipeline with various inputs.
 3. Refine agent prompts and flow connections based on results.
-



Tips

- **Keep prompts concise** but role-specific
- Use **variables** to pass shared data like [{project_description}](#)
- Use **Markdown Display** to show results nicely
- Avoid loops—Langflow works best in sequential or hierarchical flows