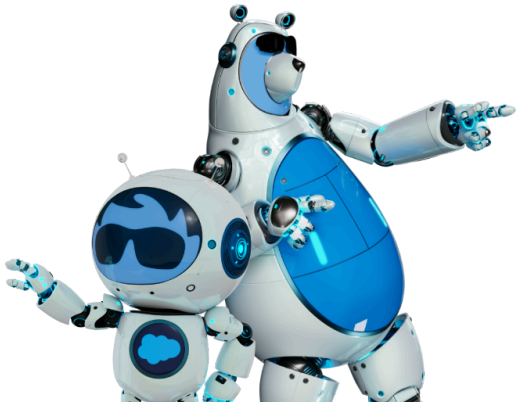
Agentforce Vibes Hands on Workshop

Agentforce Vibes is how we reimagine development in the Agentic era. Instead of juggling tools, boards, and terminals, developers can open their Salesforce Org, launch Agentforce Vibes IDE directly inside it, and start building with Agentforce.

The flow is conversational and governed:

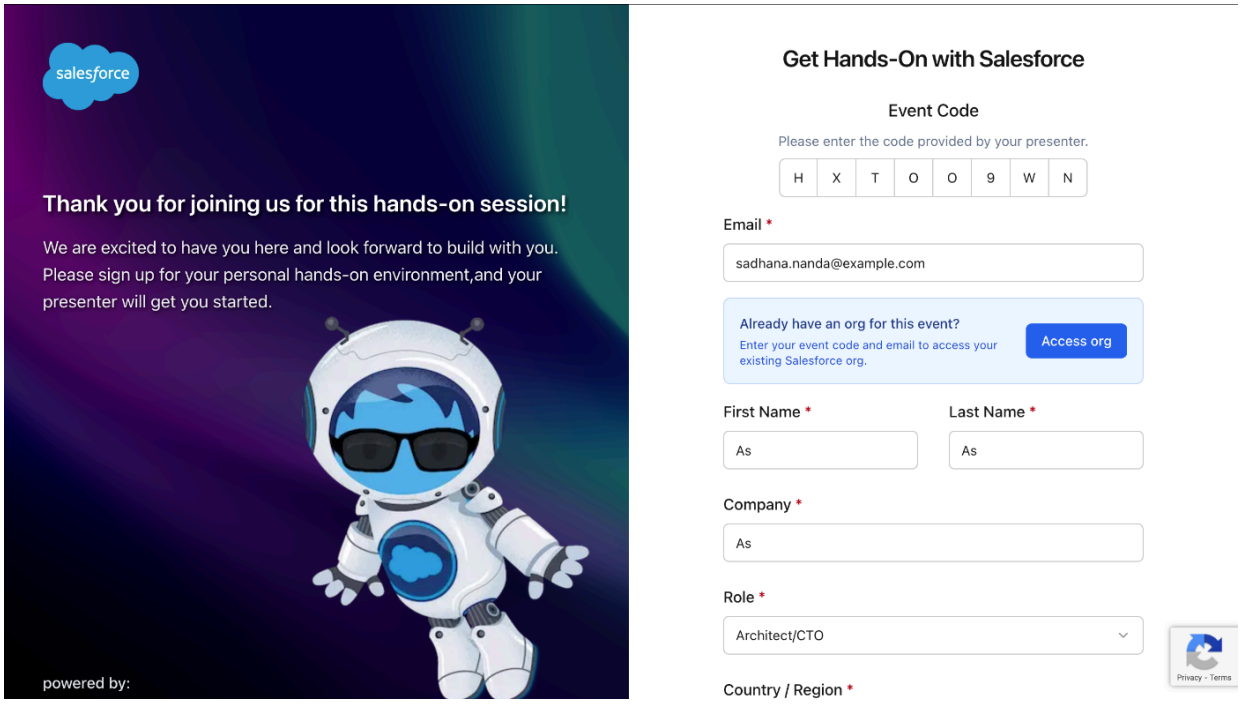


- Begin in your Org—your place to build.
- Use vibe coding with Agentforce to scaffold functionality.
- Test and run quality checks in the same loop.
- Deploy to the Org for verification.

1. PREP (5 MIN)

Sign Up:

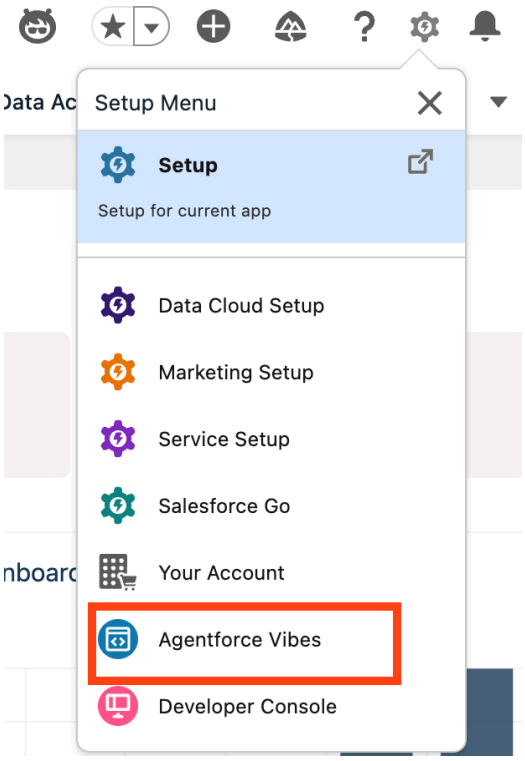
1. Sign up using the url, add in the event code and your email address. You will be receiving the org credentials on this email.



You should now see a signup successful message with a link to open the org directly. Alternatively you can also check your email for the sign up link.

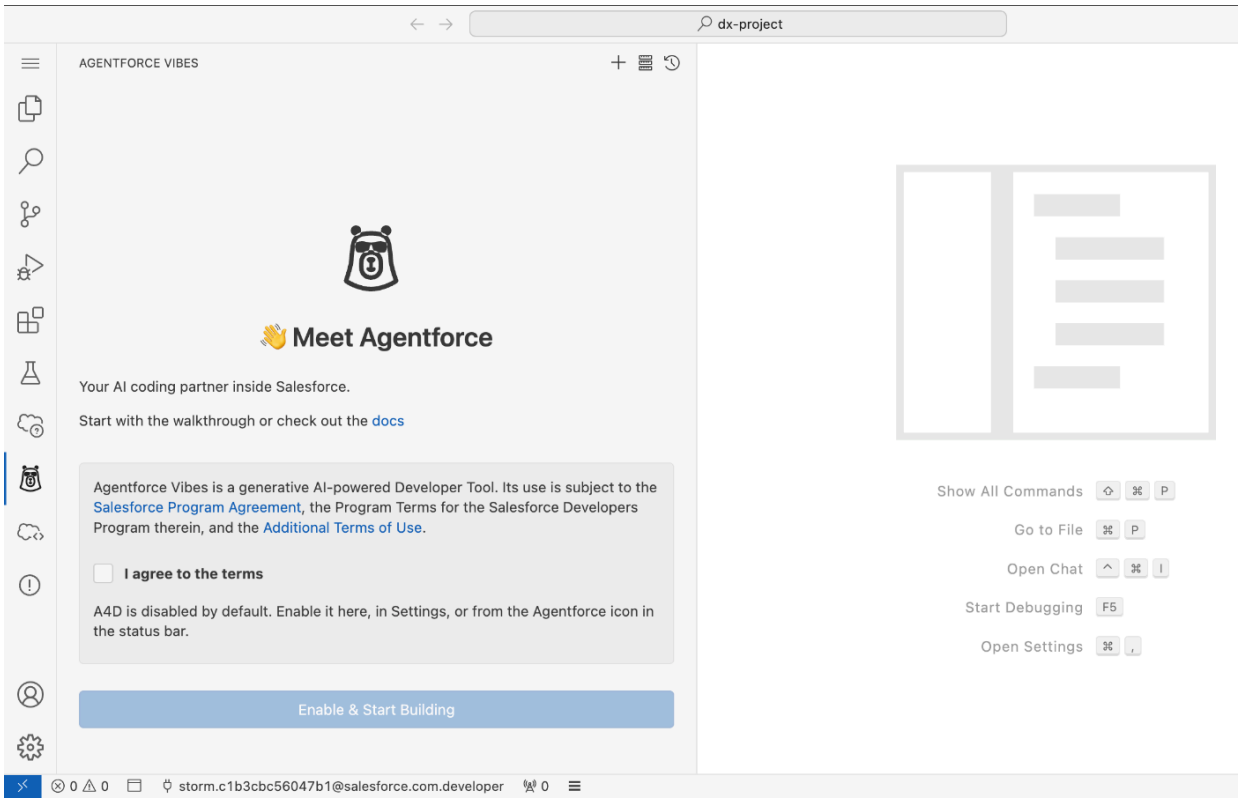
Hands On

From the Setup (gear) icon select Agentforce Vibes.

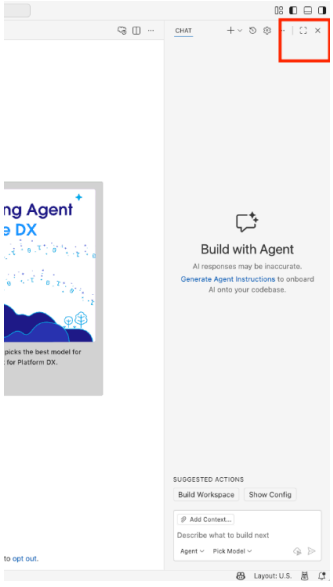


Once loaded you will see the Agentforce Vibes IDE(formerly Code Builder) has already authenticated to the org and is ready to go.

- Agentforce should automatically open, select on I agree to the terms and click on Enable & Start Building.



Additionally, make sure to close the Github Copilot from the left side pane by clicking on the X.



2) Agentforce Vibes Walkthrough (5 minutes)

Learning Moment

Inside Agentforce Vibes IDE, you'll see Agentforce — your conversational partner for vibe coding. Agentforce represents the evolution of Salesforce's AI-powered Agentforce. While Agentforce provided helpful code suggestions and explanations, Agentforce goes further by enabling true agentic interactions. When you chat with Agentforce, you're not just getting code suggestions—you're getting a coding partner that can understand your project context and execute tools on your behalf, and help you complete entire development workflows through natural language conversations.

The way Agentforce does this is through **Model Context Protocol, or MCP**. Think of MCP like the USB standard for AI agents: just as USB gives your computer a consistent way to connect to printers, cameras, or drives, MCP gives Agentforce a consistent way to connect to different development tools.
Here's how it works in practice:

- **Tool Discovery:** Agentforce knows what tools are available in your environment.
- **Intelligent Selection:** When you make a request, it determines which tools it needs.
- **Execution:** It runs those tools in the right order.
- **Results Integration:** It then brings the results back into a single, coherent response.

With MCP, Agentforce doesn't just generate code — it scaffolds metadata, LWCs, Apex, and UI assets while staying aligned with Salesforce development patterns. The out-of-the-box MCP servers are designed specifically for Salesforce, so the tools understand your org's metadata, operate securely within Salesforce's trusted boundary, and optimize for common development workflows.

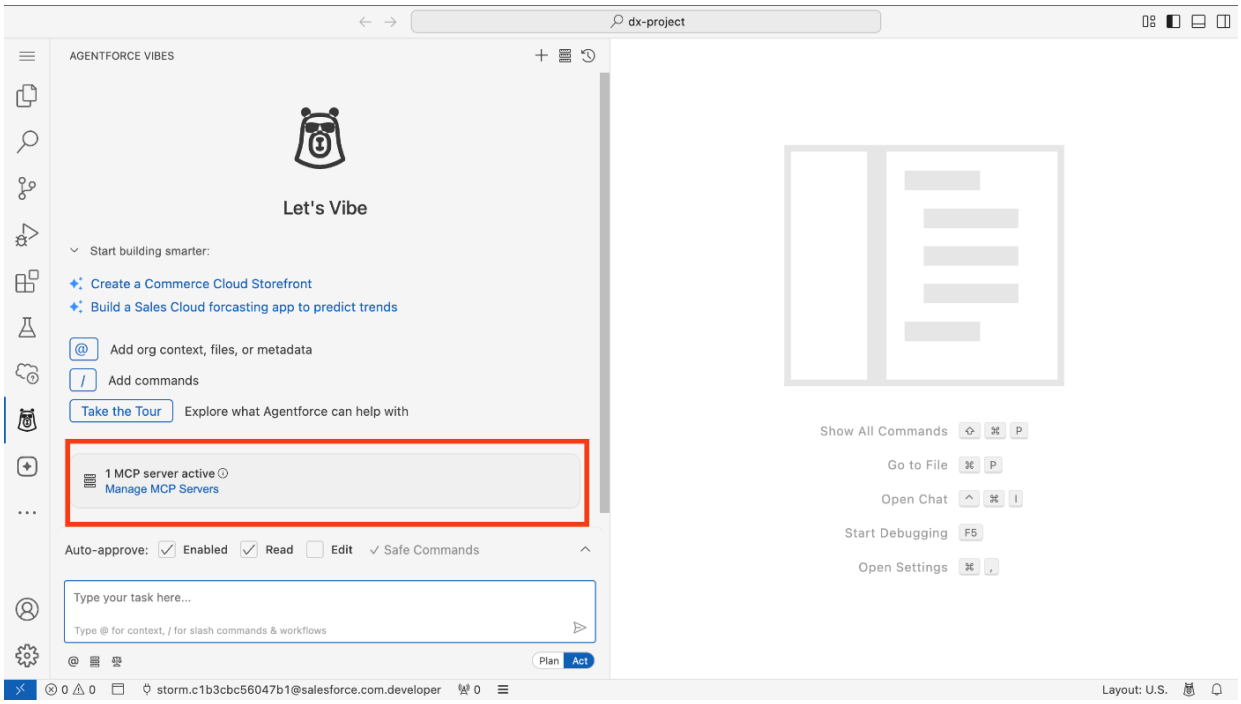
Finally, Agentforce always works in two phases:

- **Plan** — explains what it intends to do and why.
- **Act** — executes those steps under guardrails.

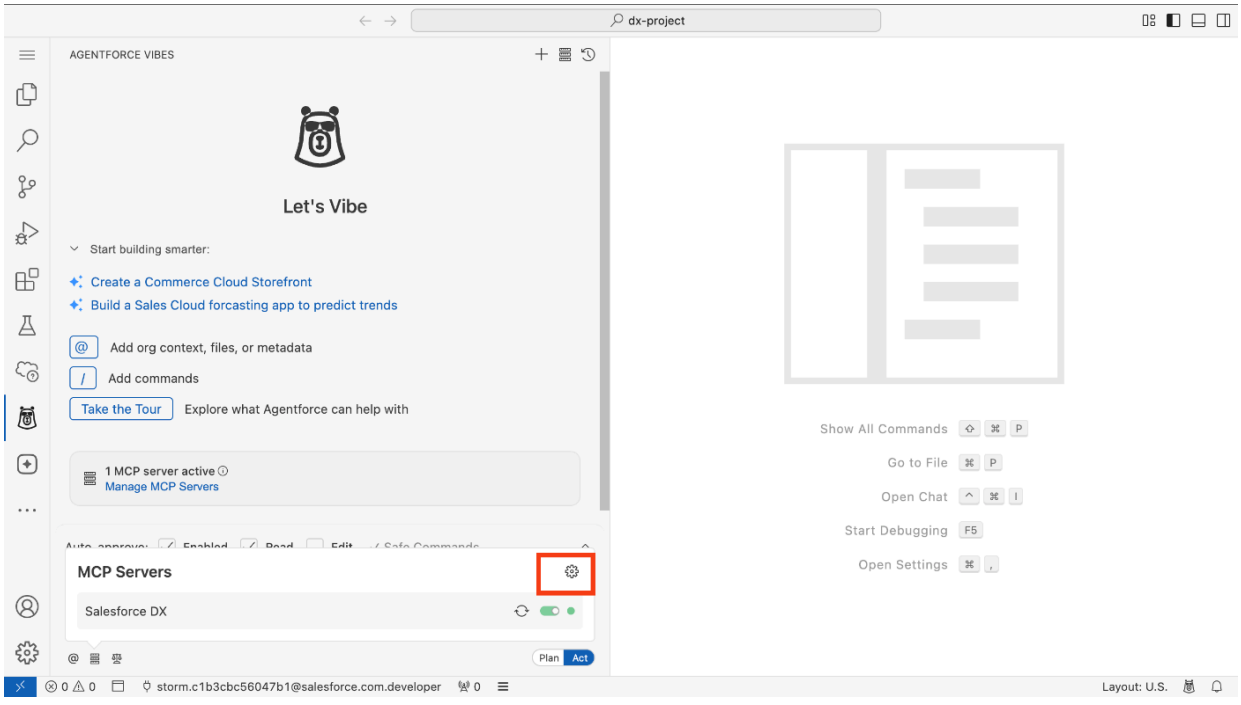
Hands On

Lets explore the MCP Server.

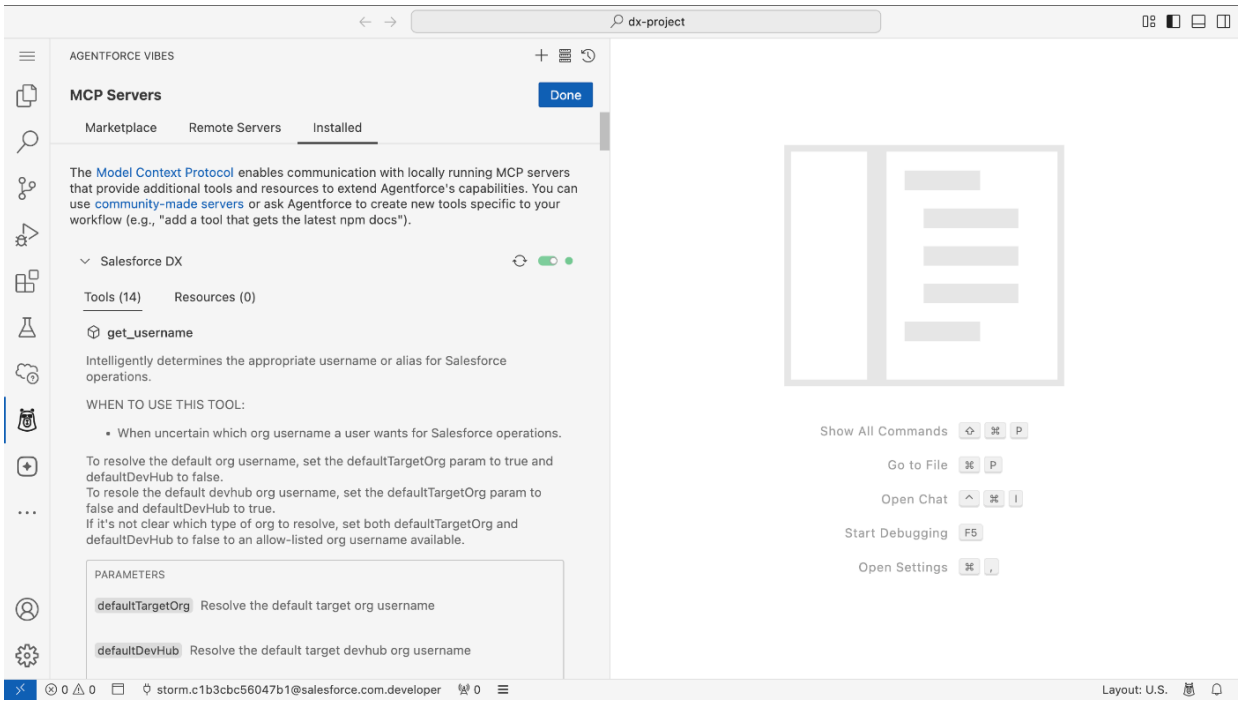
- Click **"Manage MCP Servers".**



- Notice that the Salesforce DX MCP Server is already added by default. Click on the gear icon to explore the MCP Server and the tools it supports.

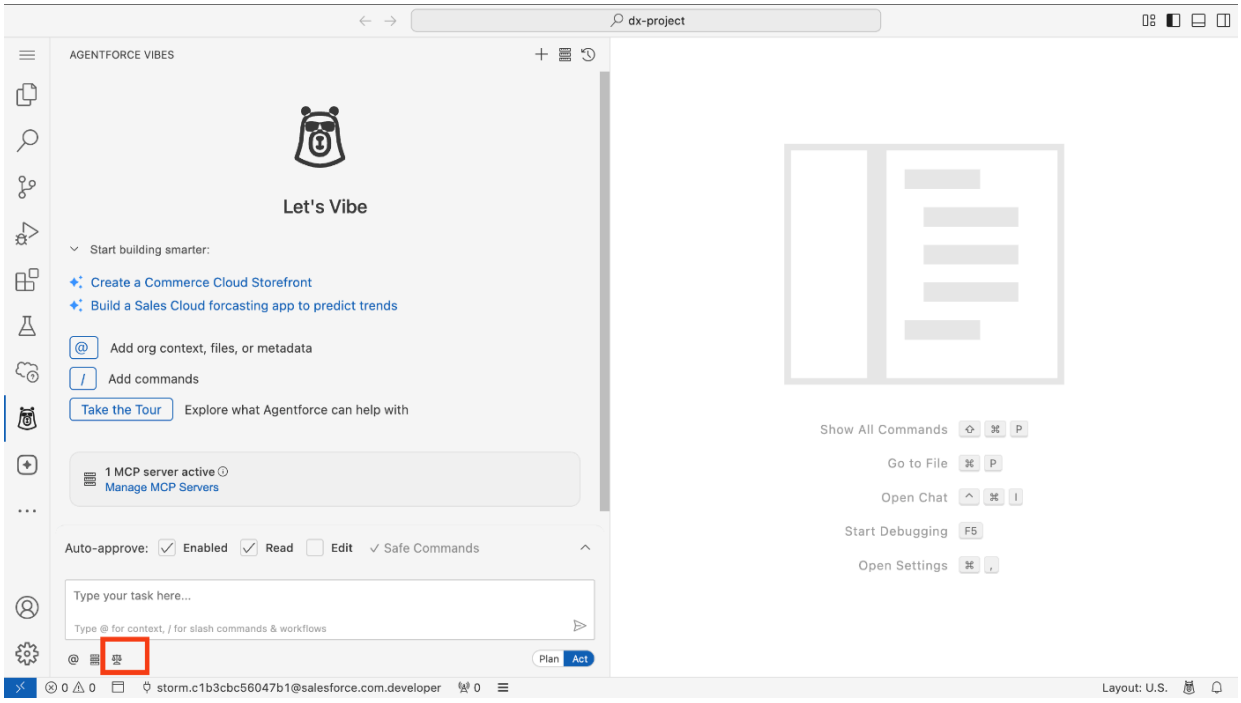


- Here you will see a list of supported tools, scroll to explore the various tools. Click on Done to exit out of this view.

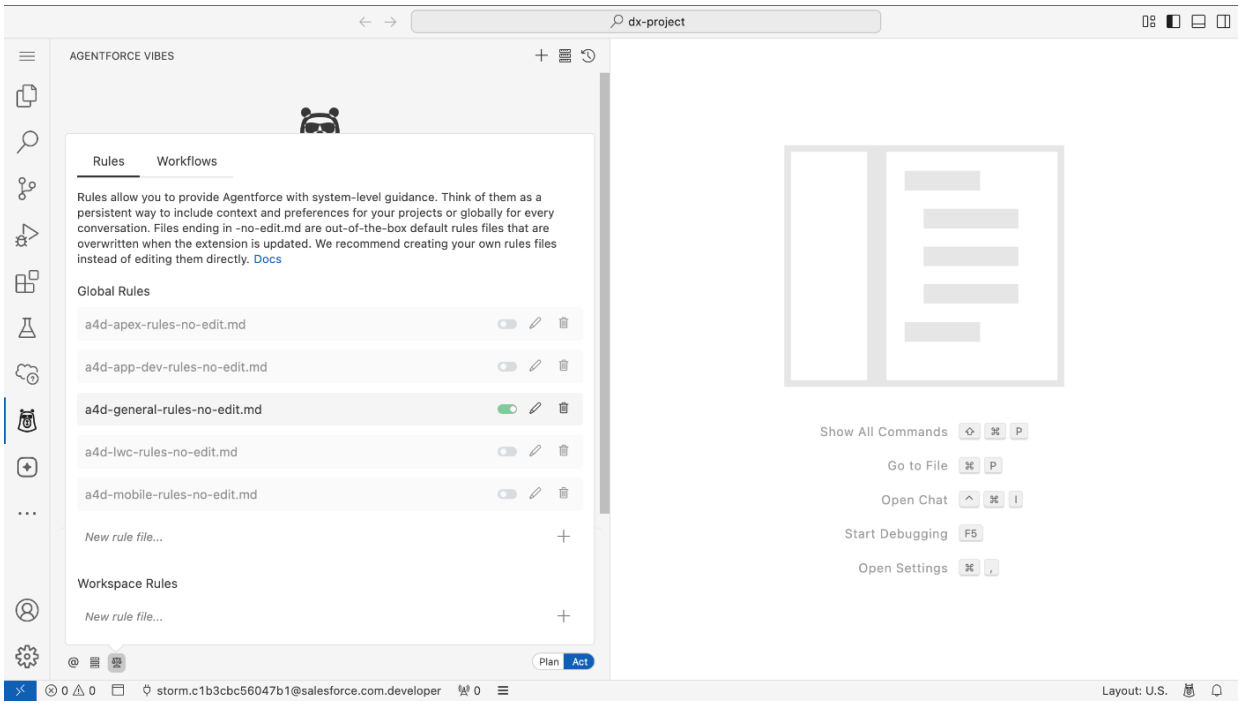


Agentforce also supports **Rules** — persistent instructions that establish your team's standards and deployment guardrails. Unlike one-time prompts, rules remain active, so Agentforce always follows your patterns and quality checks. By default Agentforce comes with a bunch of rules to promote best practices.

- Click on the Manage Agentforce Rules & Workflows icon on the bottom pane

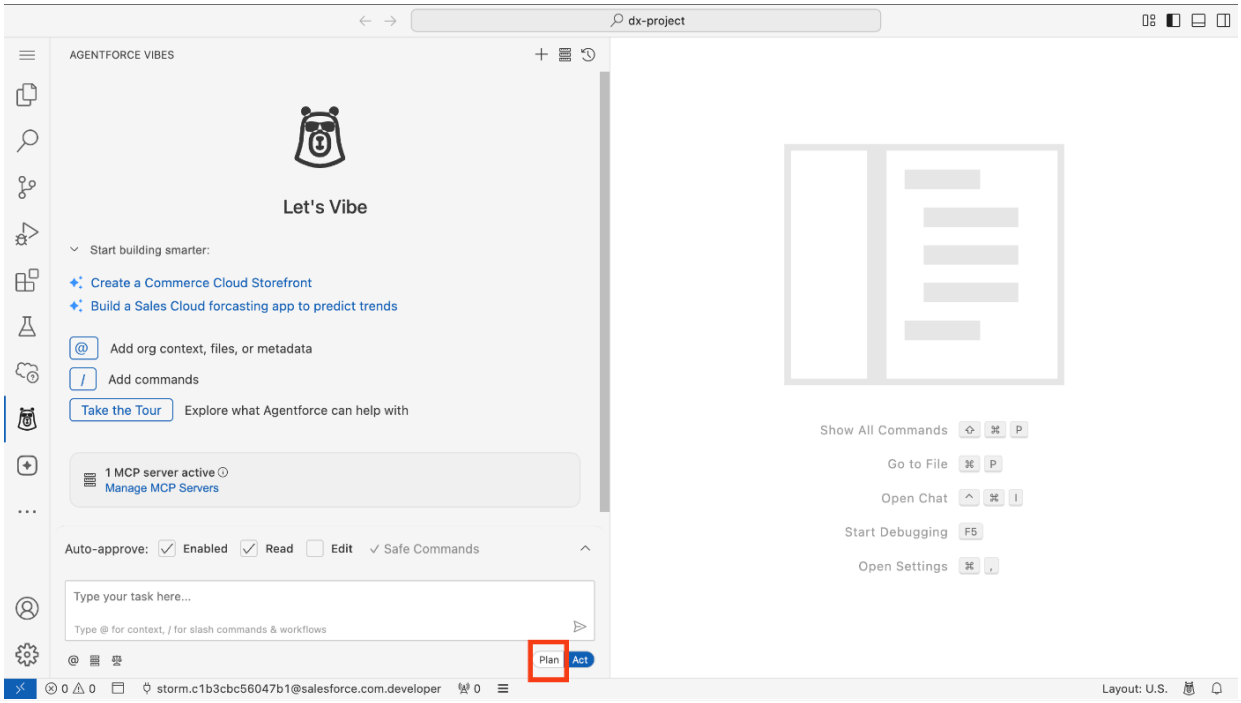


- Inspect the various rules, you can click on the pencil icon on any of the rules to explore further.



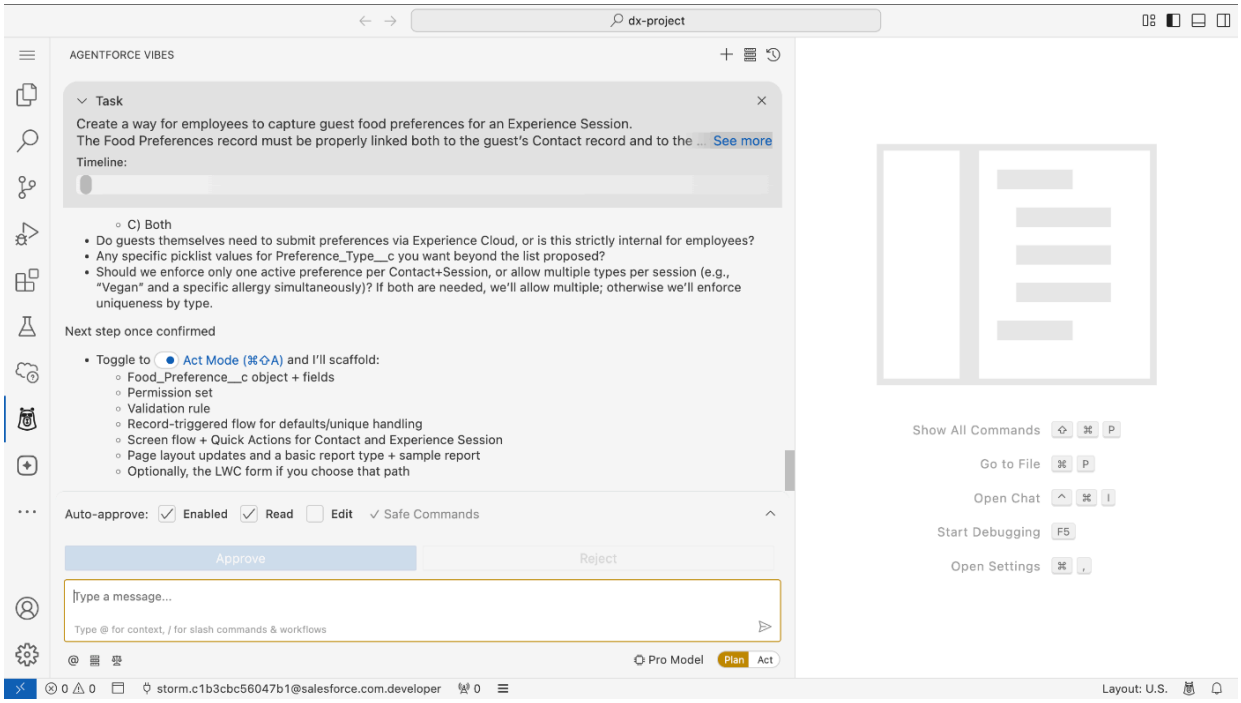
3) Scaffold Preview

- Next lets take the Agentic Chat for a spin. From the bottom right section of the Agentforce, Click on **Plan.** Now provide the following prompt.



```
Create a way for employees to capture guest food preferences for an Experience
Session.The Food Preferences record must be properly linked both to the guest's
Contact record and to the Experience Session record, so we always know who
the preference belongs to and which session it applies to.
```

- **Plan mode focuses on analysis and strategy without making changes to your files or org.** Examine the plan generated, notice it identifies all the metadata required for the plan including custom objects, apex methods, relationship fields, LWC and even tests.



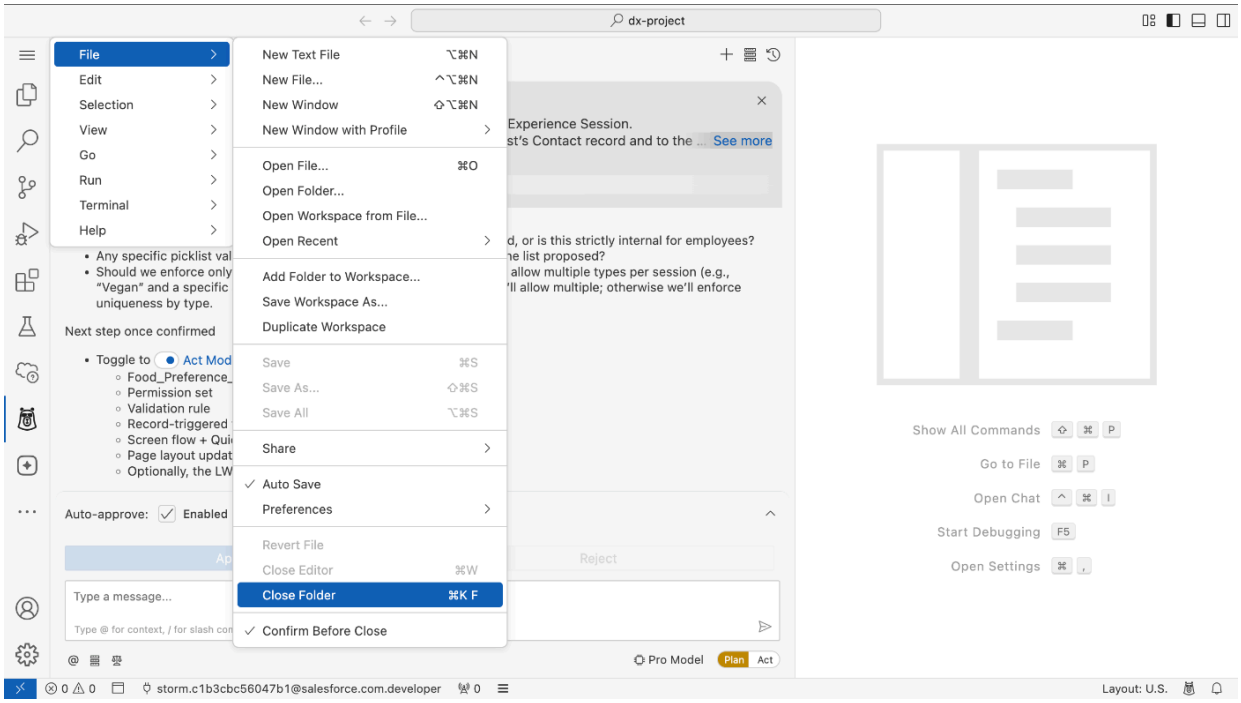
Vibe coding is great for rapid prototyping and cutting boilerplate, but enterprise teams also need trust and governance. That's why Agentforce Vibes balances both — fast scaffolding with Plan and Act, but always under developer review.
Since we only limited time in this workshop, we'll preview scaffolding in Plan mode, then fast-forward into a curated project so you can experience the full loop — testing, deploying, and iterating in the org.

---

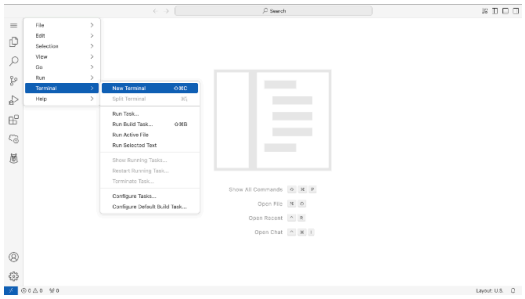
4) Fast-Forward to Prebuilt Project (8 minutes)

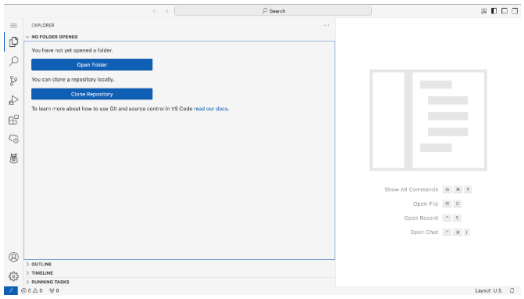
To make sure everyone can complete the full loop today, we'll now fast-forward to a curated repository. It contains the same functionality Agentforce would scaffold — metadata, a Food Preferences LWC, and test classes — so we all start from the same working baseline.

- In Agentforce Vibes IDE click on the Menu icon from the left pane → File → Close Folder



- Next go to Menu from the left pane → Terminal → New Terminal



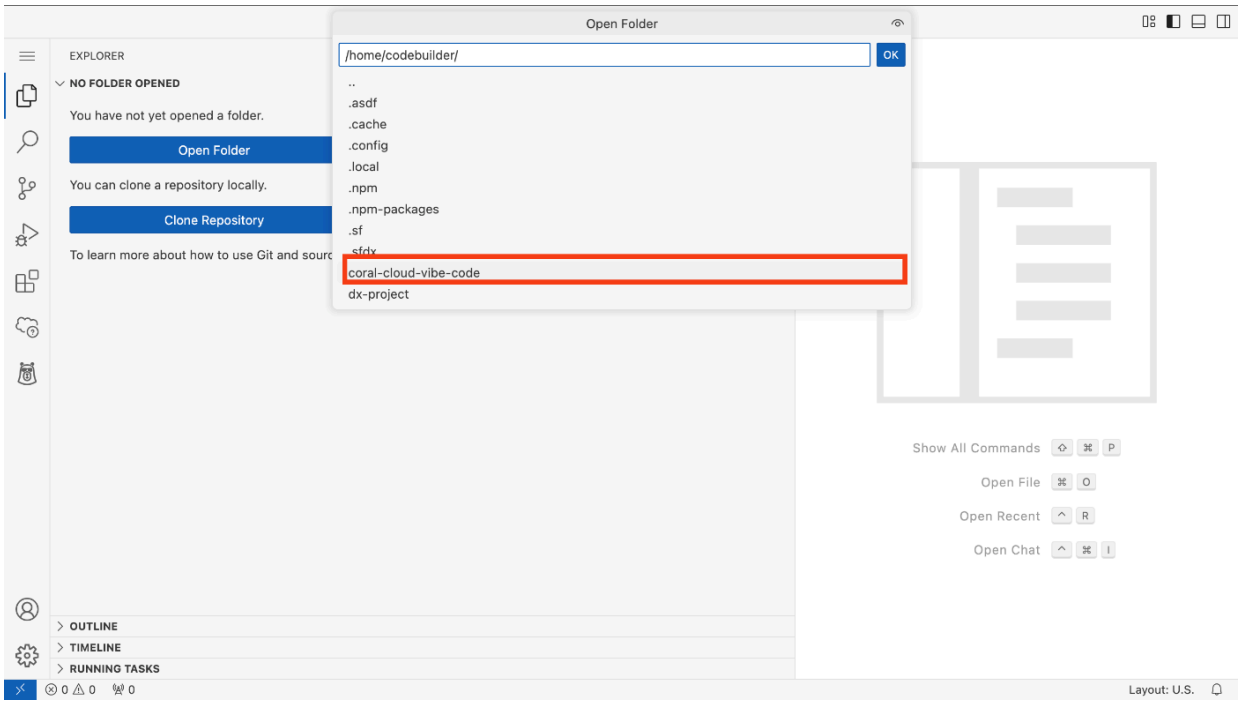


- In the terminal paste the following git command; if asked to allow pasting on terminal window, click allow;

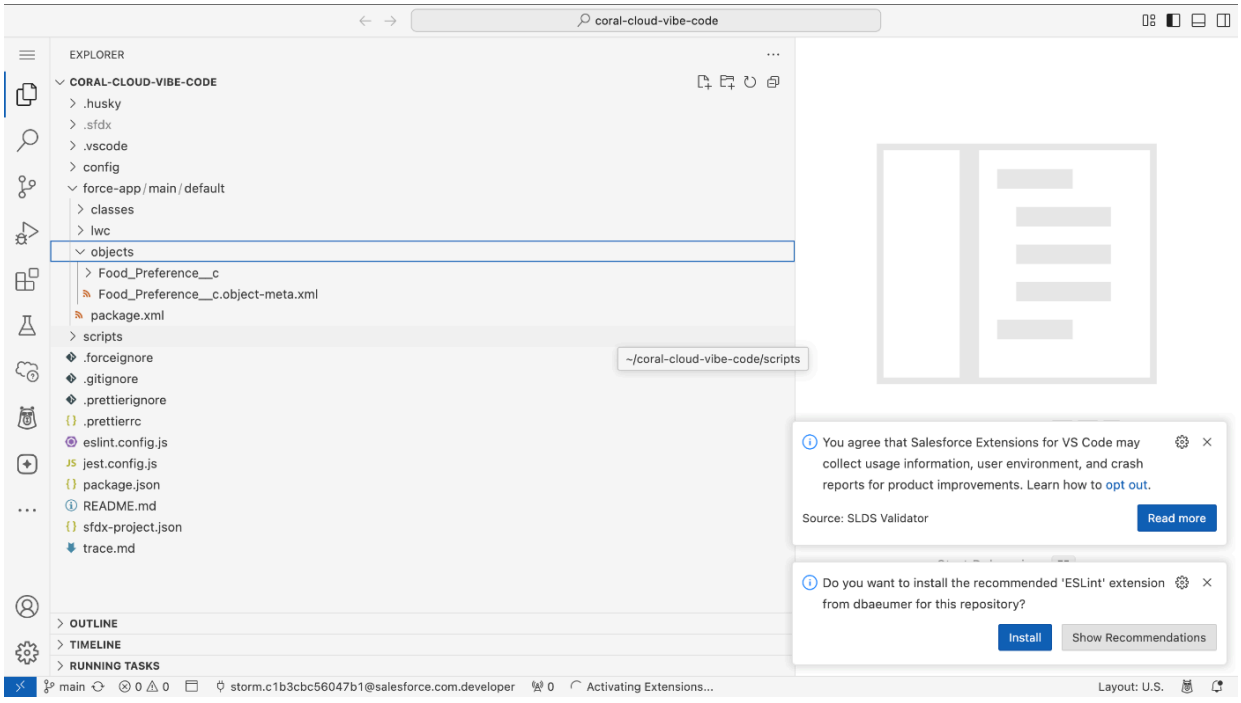
```
cd ..
cd codebuilder/
git clone https://github.com/sadhananandakumaralmny/coral-cloud-vibe-code
```

- Now lets open this folder, click on the File icon from the left pane and click on Open Folder

- Choose the folder coral-cloud-vibe-code and click ok.

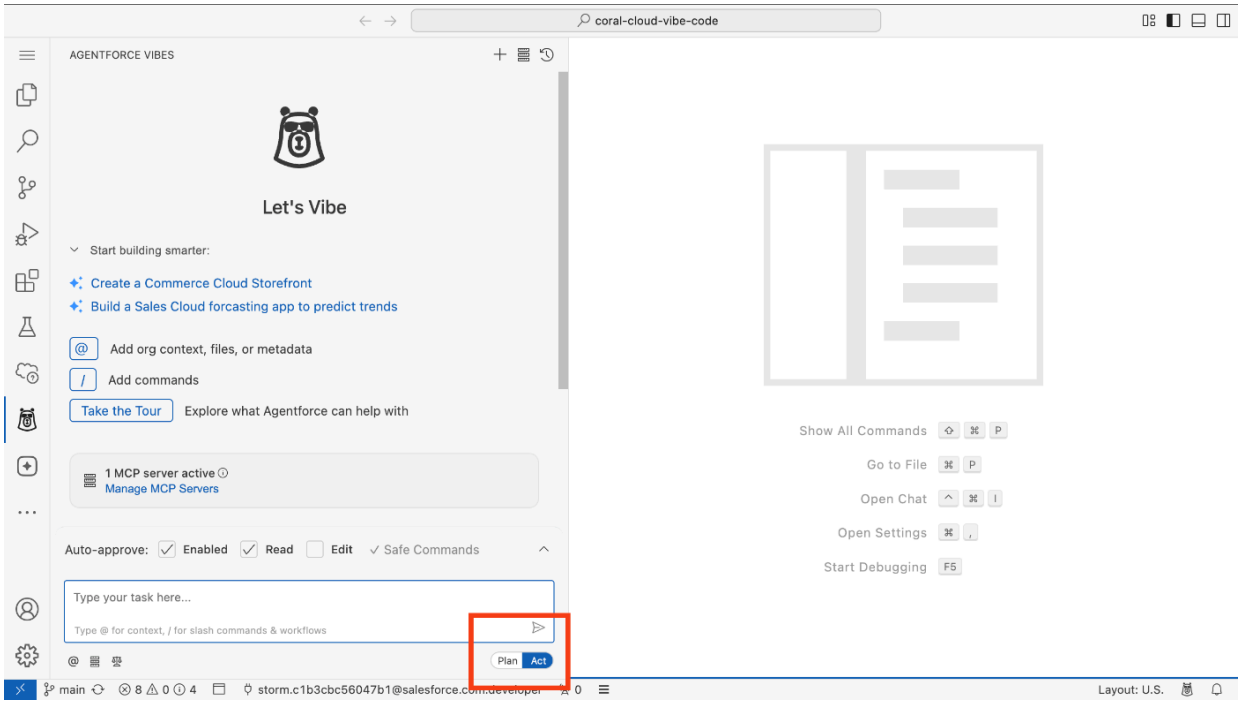


- The project is now available for us to use, you can inspect the folder structure to see all the metadata related to the Food Preference feature.



Notice how the curated repository already includes everything Agentforce would scaffold: the metadata, the Food Preferences LWC, and **test classes**. There are MCP tools that can run tests for both agents and Apex methods, and you can also keep code quality in check using the **Code Analyzer**—just right-click a file to analyze it.
With that foundation in mind, let's move directly into the **deployment portion** of our workflow.

Finally also make sure you flip the switch from Plan to Act so we can deploy in our next chapter.



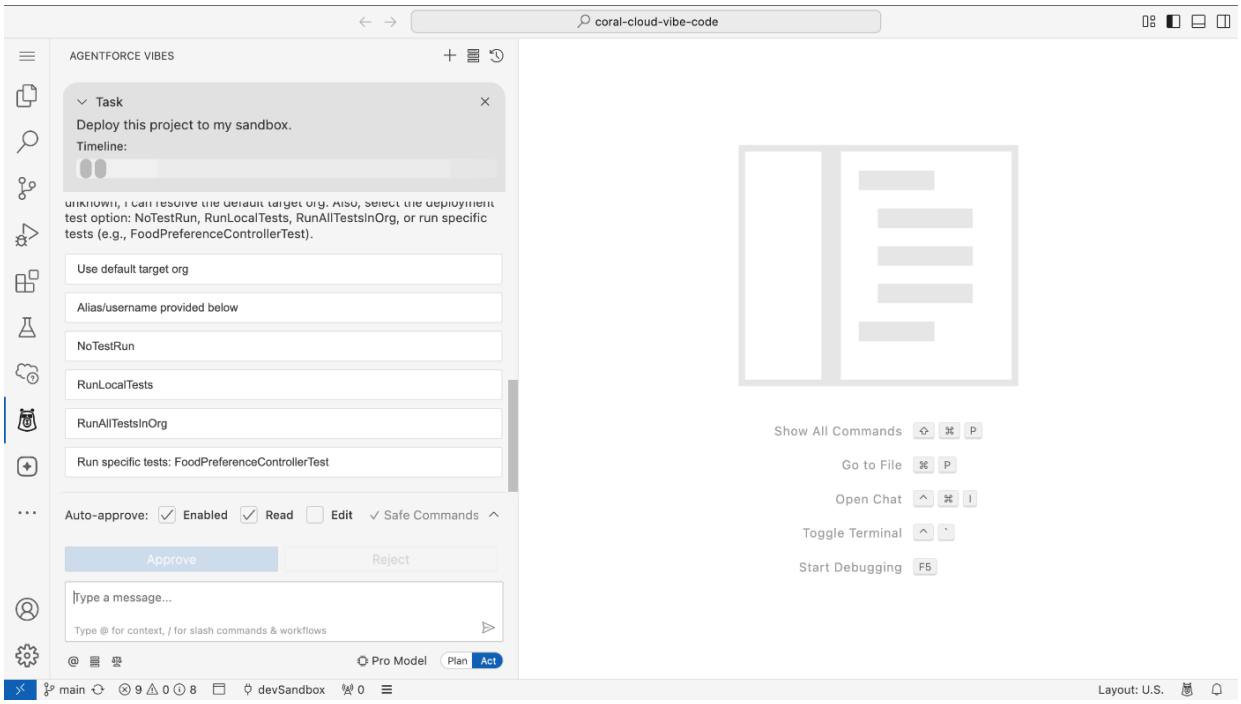
5) Deploy & Verify (10 minutes)

Now let's deploy the functionality into our org. Once complete, we'll flip to the Salesforce UI and validate: inside Coral Cloud Resorts, and we will discover and add the Food Preferences component to our home page.

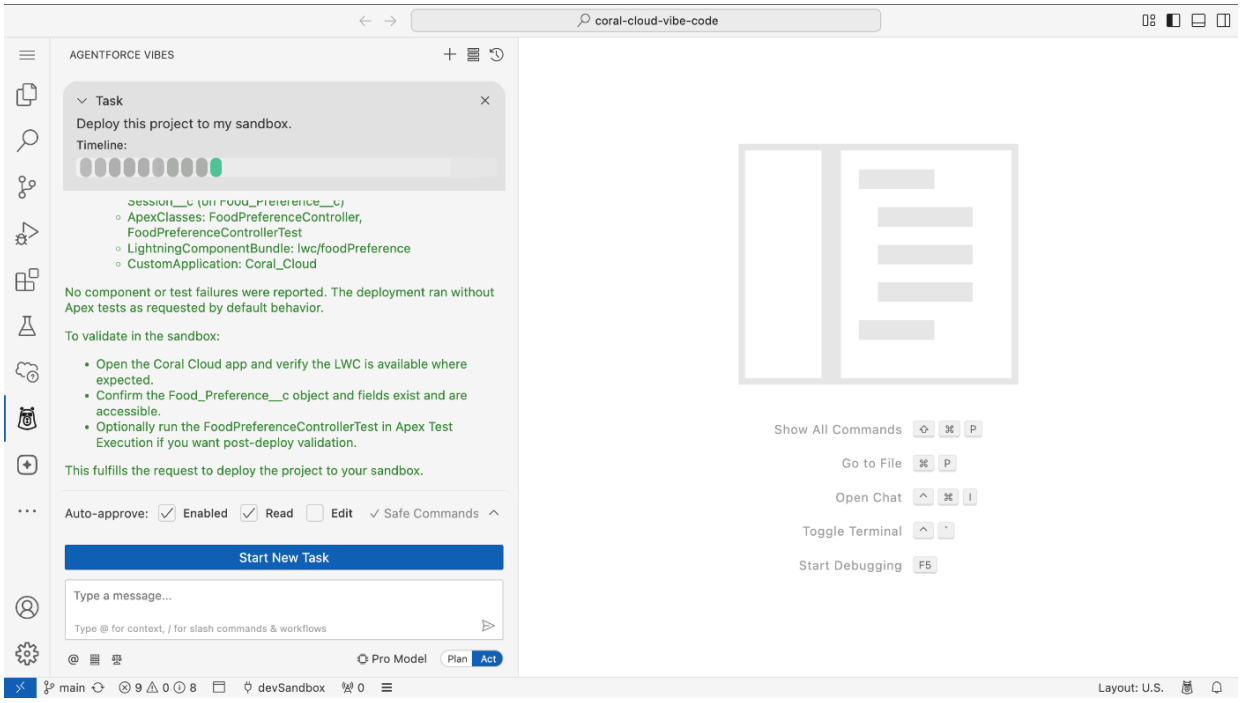
- In Agentforce, conversation pane, paste the following prompt

```
Deploy this project to my org.
```

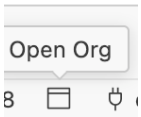
- Notice that it asks for a confirmation on which org to use, lets choose the default target org and proceed.



- Behind the scenes, MCP tools are used get the username and perform the deployment. Once done inspect all the changes deployed. Once we have a clean run we can proceed to the next step, Agentforce asks for confirmation as it performs the various actions.

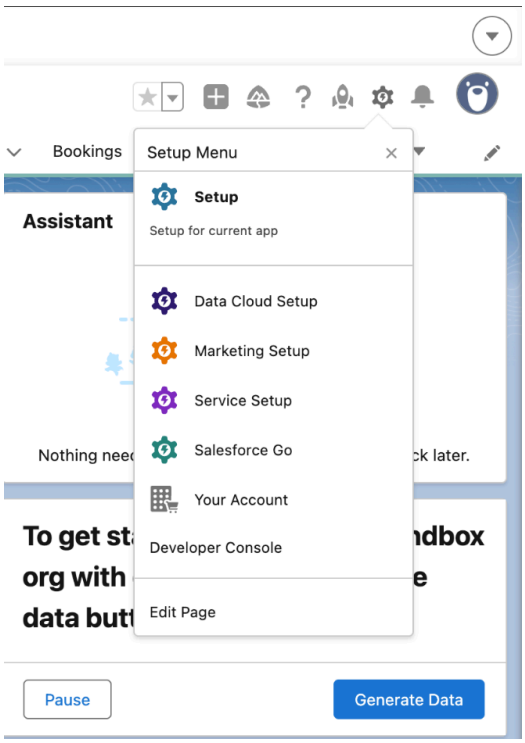


- We will now verify the changes. Click on the window icon from the bottom pane to open the org.

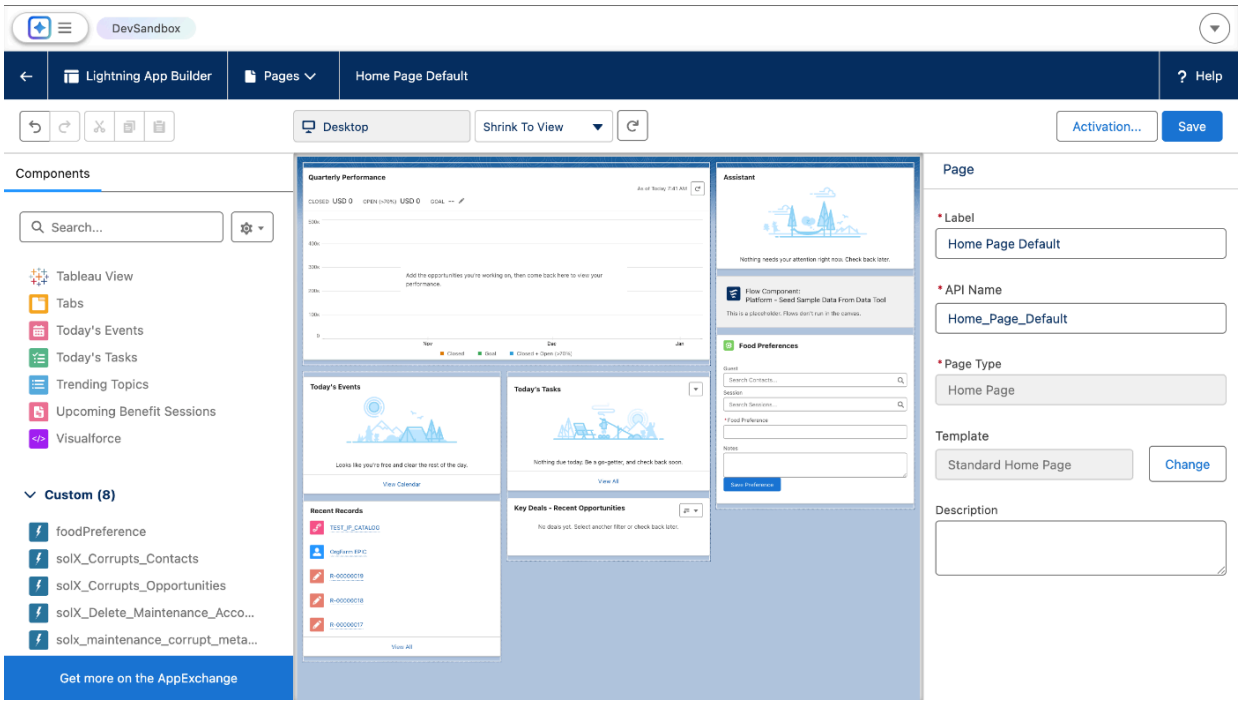


- From App Launcher go to Coral Cloud Resorts app → On the Home Page → Click on the gear icon on the top of the screen and click on Edit Page.

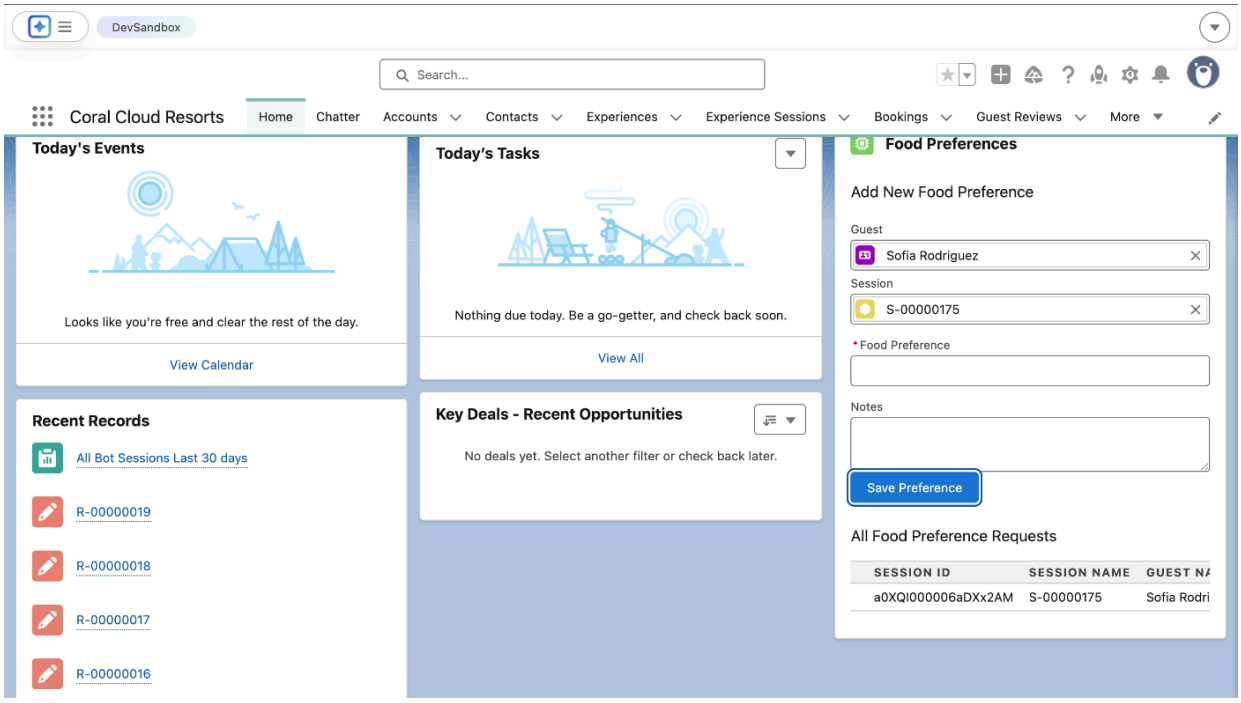
TIP: It might take a second/two for the custom components to load.



From the custom components choose drag foodPreference and drop it anywhere on the page, click on Save.



- Now Test the component, by providing Guest (choose from lookup), Session (Choose from lookup) and specify a food preference (Vegan, Vegetarian etc.,) and click on Save Preference. You will see the food preferences updated in a table below.



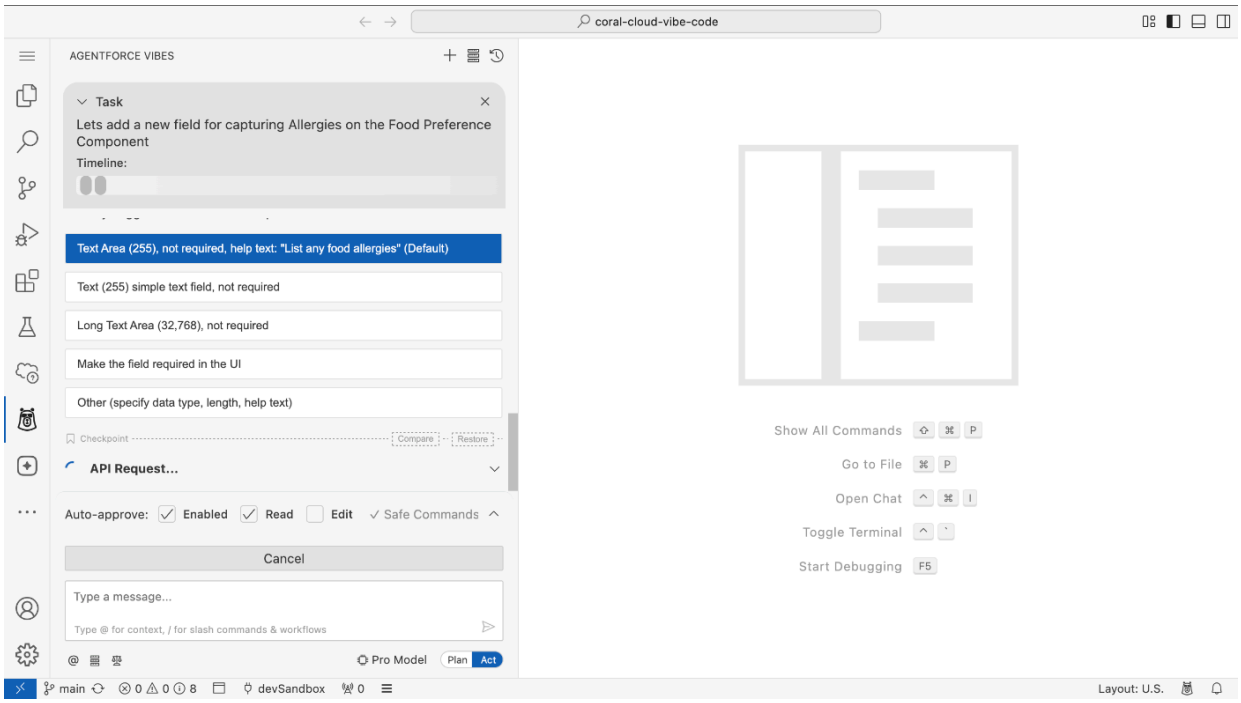
6) Make a Small Change & Redeploy (13 minutes)

Development is iterative. Let's make a change: add a Special Instructions field to the Food Preferences component. This demonstrates the EVC loop: instruct → scaffold → deploy → verify. Every iteration lives in the org until it's validated.

- In Agentforce, conversation window paste the following prompt:

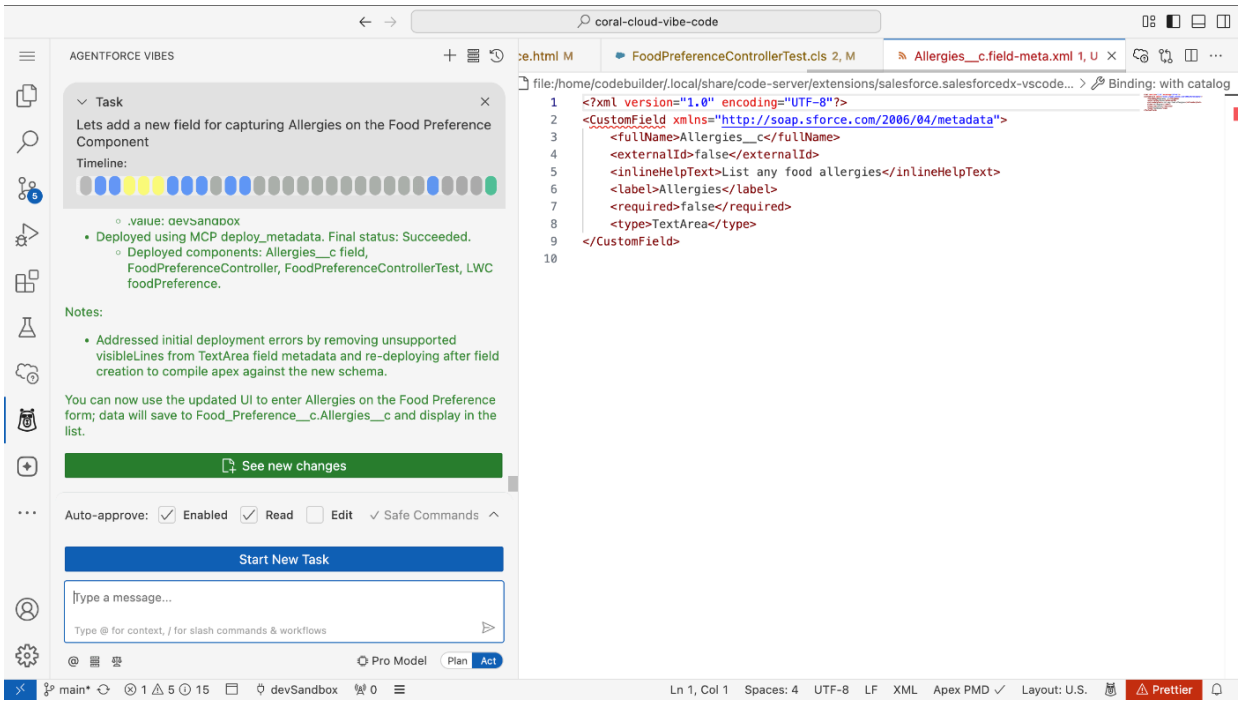
```
Lets add a new field for capturing Allergies on the Food Preference Component
```

- Notice how the Agentforce asks for clarification of the best approach to follow. The user can choose the option to proceed with the development.

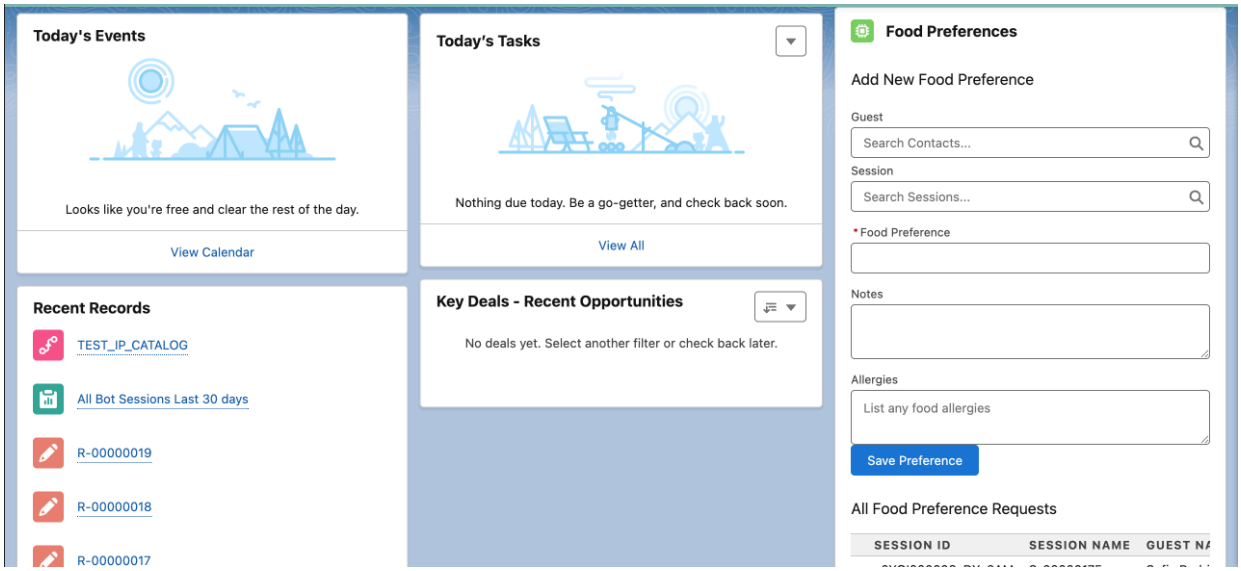


TIP: Agents are non determenistic, you might not see the same response as above! You might notice issues, but the agent iteratively fixes the issues until eventually getting to a clean deployment.

- At every stage Agentforce asks for permission to make file changes and dynamically updates to fix errors as it goes until completion and even continue to deploy the changes.



- The changes are now deployed and we can now verify the dev env
- Refresh the org to see the new field added (you might need to hard refresh)



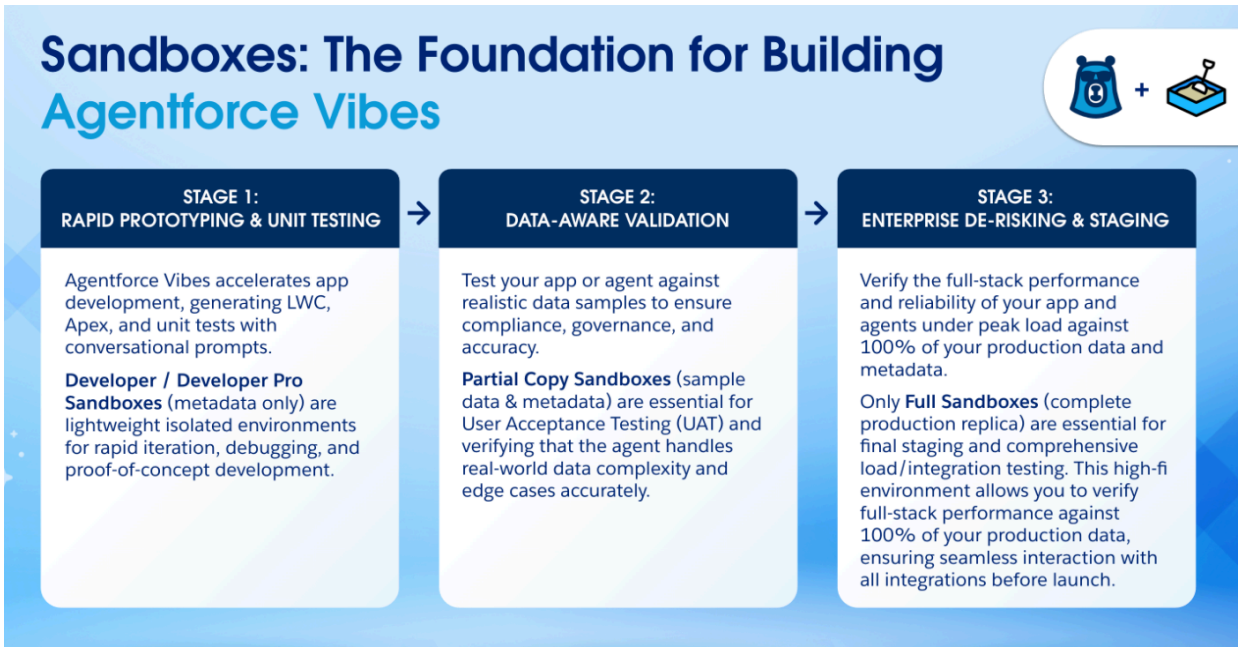
7) Beyond this workshop

Learning Moment

What we experienced today was the rapid, iterative cycle of vibe coding: building, testing, quality checks, deployment, and verification.
While we worked directly in the Org today for the purpose of this workshop, the **ideal best practice** for enterprise development is to execute this entire loop inside a **Sandbox**. This ensures you can experiment and iterate without any risk to your production data or configuration.
The Ideal Workflow: In a real-world scenario, the process follows this safe path:

- **Begin in a Sandbox:** Your safe, isolated environment for development.
- **Launch Agentforce Vibes:** Open the IDE directly inside the sandbox.
- **Scaffold & Iterate:** Use Agentforce to build features conversationally, applying rules and guardrails via MCP.
- **Deploy & Validate:** Deploy changes to the sandbox first to verify functionality.
- **Promote:** Only once changes are validated do you commit them and use your CI/CD tool (DevOps Center, GitHub Actions, or Jenkins) to move code into production.

This iterative cycle—from scaffolding to validation—is the core of the Agentforce ALM strategy. Although we used a developer Sandbox for the purpose of this workshop, to make this process secure and repeatable for every feature, we rely on a specific environment at every step (eg a separate **Full Copy Sandbox** for testing agents and complex apps). This slide illustrates exactly which type of Salesforce Sandbox provides the necessary data and fidelity for each stage of the AI agent lifecycle.



8) 🧠 Take Home Prompts

Keep building your prompt-engineering skills beyond this workshop. Use these examples as a starting point for your own experimentation. The more you iterate, the better you'll understand how prompt phrasing, context, and grounding affect the quality of responses. Keep refining your approach — effective prompt writing is a skill that improves with practice.