SESSIONS

Hosted GPU Access, AI Image Generation, LLM Driven NPC's in Games, and More! - April 3, 6:30 PM | 9:19 PM

Colin McNamara - 6:30 PM
Thank you.

Colin McNamara - 6:32 PM
Hey all, Ricky is getting all set up and started. If you can log on in and chat and say where you're at, what you're looking for. We'll be on shortly.

Ricky Pirruccio - 6:33 PM
There you go, there you go. . . . . .

Ricky Pirruccio - 6:34 PM
. . . . . .

Ricky Pirruccio - 6:36 PM
. . . . . .

Ricky Pirruccio - 6:37 PM
. . .

Ricky Pirruccio - 6:41 PM
Right here.

Ricky Pirruccio - 6:41 PM
If you also just Google search Austin Lain Chain on Google, Austin Lain Chain GitHub and go there, that could work. Austin Lain Chain.

Ricky Pirruccio - 6:42 PM
Okay, so we're going to briefly go over accounts that you will need to set up here, then we'll dive into setting up environment variables, and that will pretty much be it. And then Docker, briefly, as well. So, okay, accounts. Does

Ricky Pirruccio - 6:42 PM
everyone heard of OpenAI? I assume you have, until this point. But if you haven't, that's where we're, that's mostly the models that we work with. If you go to this account right here, to this link right here, you will, it will prompt you to create an account. If you don't have one, and you need an account, because you need to use...

Ricky Pirruccio - 6:43 PM
Does anyone not have an OpenAI account? It's a free one. You will have to pay for your own tokens. I believe they give you a certain amount of tokens for free when you sign up. But generally, it's recommended to pay for it. It's basically, you load it up with a couple bucks.

Ricky Pirruccio - 6:44 PM
Once you're there, you're going to want to go to the API keys tab, click on that, and then you'll want to create a new secret key. That's what it's going to look like.

Ricky Pirruccio - 6:44 PM
Does everyone have an API key for OpenAI? Awesome.

Ricky Pirruccio - 6:45 PM
The next step is Langsmith. Who has not heard of Langsmith? Awesome, makes sense. So, Langsmith is sort of a tool developed by Langchain to give visibility into what's happening under the hood as you run your AI app.

Ricky Pirruccio - 6:45 PM
It kind of gives you a trace of everything that's going on from the call, from where the user started the query, to the LLM, the processes, and all the various components along the chains. So, it's a very useful tool for that. Basically, testing and debugging. You can kind of think of it as that. So, you'll want to go to...

Ricky Pirruccio - 6:45 PM
this link right here. Right here. Langsmith account signup. Click on that. It'll lead you here. It will look something like this once you have an account and you're logged in. But, what you want to do is basically get an API key for this as well.

Ricky Pirruccio - 6:46 PM
So, then you'll want to go to settings. Actually, I'll wait for anyone. Has everyone created an account? Langsmith. I just finished. But, I'm going to give up and I'll just watch. Just let me know. The pace is too quick for me. It's fine. Keep going. That's... Sure. It's...

Ricky Pirruccio - 6:47 PM
The link is right here, if you want to follow along. No problem. So, you'll want to go to settings. And you will create a...

Ricky Pirruccio - 6:47 PM
Click on this button right here, create API key. That will give you a key. Just copy it, put it somewhere safe for now. So, at this point you should have your OpenAI key and your Langsmiths key.

Ricky Pirruccio - 6:48 PM
anyone still

Ricky Pirruccio - 6:48 PM
I have a question. Is there any option like open source option that does kind of the same? I'm talking about the lengthy parts.

Ricky Pirruccio - 6:49 PM
It's free right now. There's also Langflow, which you can run on your own computer. There's a few more options. You can also debug locally. Yeah, so Langsmith is completely free. Right now. Right now. At the moment.

Ricky Pirruccio - 6:49 PM
So to echo, Langsmith right now is a paid platform. The other, you can always, you can always output, you can also, excuse me, you can debug it to a stream and see the debugs going through. There's also a project called Langflow at a Yahoo Combinator.

Ricky Pirruccio - 6:50 PM
Right now you can run it on a Docker image and you can get a very similar interface. Langflow is offering the first, I don't know, small amount of messages free. We expect the Langchain AI team to effectively have a parallel to them. The Langsmith is kind of an integrated debugging interface for the project. It's good stuff. It works pretty well.

Ricky Pirruccio - 6:50 PM
Cool. So next thing we're gonna do is go to Google Collab. Who has not heard of Collab? Awesome. So have you guys heard of Jupyter Notebooks? Awesome. So yeah, that's basically Collab. It's Jupyter Notebooks hosted by Google. By default when you have a Jupyter Notebook on a GitHub repo, you automatically get a link.

Ricky Pirruccio - 6:51 PM
for a Collab Notebook. That basically lets you just run a Jupyter Notebook just being logged in the web. All you need is a Google account. Does everyone have a Google account? Alright, cool. But if you don't have one, you can go click this link right here and you can create a Google account.

Ricky Pirruccio - 6:51 PM
And if you have a Google account, you should just see this right away. There shouldn't be any more any other tinkering with

Ricky Pirruccio - 6:52 PM
it." So if you click on this link right here, LinkChain 101, that will lead you to a previous notebook that we have in our repo. And if you see that, you're in good shape. You have Colab. It's working for you.

Ricky Pirruccio - 6:52 PM
Everyone able to get here so far? Awesome. So now in Colab, what you want to do is, you'll want to set up environment variables. So we work a lot with Colab because it's really easy to put your environment variables there and share code and run it in sections and blocks.

Ricky Pirruccio - 6:53 PM
So you can see exactly what's happening in every section of the code. So with environment variables, Colab does this very cool thing, where you can store environment variables in basically your account, and then that persists throughout all the notebooks that you run. So super useful. It's very easy.

Ricky Pirruccio - 6:53 PM
You go to this key icon right here, click on that, and you'll want to store your keys right here. So you'll basically do add a new secret key. You just type in the key name right there, and then here you'll put in your actual key. So for the OpenAI API key, it's just this, OpenAI underscore API underscore key.

Ricky Pirruccio - 6:54 PM
And then the Lang chain is going to be, that's for the LangSmith, LangChain underscore API underscore key. So those keys that you got earlier from OpenAI and LangSmith, I want you to put them here.

Ricky Pirruccio - 6:54 PM
So far, so good. Cool. Now there's another way to, by the way, this is how you would use it in your code. If you're wondering, if you ever need to reference this, you can just go to this MD file, and you can see it here. Google also tells you how to use it here, right under where you declare your environment.

Ricky Pirruccio - 6:55 PM
And you can also store environment variables in your own machine. OpenAI has some pretty good docs on how you do that, if you go here. You don't really need to do that for today. We're just going to be running lab notebooks. But if you ever need this, you can just go to our MD file in GitHub, and you can see yourself.

Ricky Pirruccio - 6:56 PM
The final piece of this onboarding is setting up docker. Who has not heard of docker? Awesome, okay. Everyone's heard of docker. Super useful tool. Very good for us to run our models, especially the ones that code.

Ricky Pirruccio - 6:56 PM
Do some harm to your system where you're running it. You can run your chains or models in these sandbox environments. Docker is also really cool for just sharing code with people. So the same code works on different environments.

Ricky Pirruccio - 6:57 PM
It's super cool, super powerful. We use it a lot here. So if you go to this intro to Docker tutorial, so that there is a link here, intro to Docker, it would actually lead you to a previous tutorial that we made here on how to set up Docker. And all you really need right now

Ricky Pirruccio - 6:57 PM
is to install it on your desktop. It's really easy to do. So you just go to Docker desktop right here, this link. You can also just Google Docker desktop, uh, download. Not necessarily. Yeah. You just need to download Docker. Um,

Ricky Pirruccio - 6:58 PM
does everyone have Docker downloaded or anyone not download Docker?

Ricky Pirruccio - 6:58 PM
Well, we're not going to be using Docker today, it's just going to be Collab notebooks. So if you were able to get your environment variables and you put it into Collab, you're in good shape. Cool, let's see how we're doing in sessions. Is there anyone here? We've got 16 participants. Oh wait, there's people here.

Ricky Pirruccio - 6:58 PM
So the people on that line, is everyone connected? Yeah, there's also people remote. Awesome. I think that is it. Does anyone have any questions? Do you want to go through the first notebook? Yeah.

Ricky Pirruccio - 6:59 PM

We had news and updates first. Oh, I thought you were going to go through that first short update, the first getting started notebook that you had launched. Yeah, it was this one. Right. Okay. Cool. Okay.

Ricky Pirruccio - 6:59 PM
I'm sorry, is that a calling tab for us? Calling that McNamara. The GitHub? Oh. Is that what you're trying to look for? Yeah, this is a repo.

Ricky Pirruccio - 7:00 PM
Austin link chain. If you just look for that on Google, you'll find it. Come over on this side and Ricky's just going through or you're gonna go through

Ricky Pirruccio - 7:01 PM
it. You should've gotten a link in your email for the sessions interface. Let me see. We have this set up for whoever's speaking. And then the AI mode? Yeah.

Colin McNamara - 7:02 PM
I'll just give it one moment, we're going to switch over to our next speaker.

Ricky Pirruccio - 7:02 PM
Thank you so much for bearing with us and learning with us. We are experimenting each month in a new and interesting way.

Karim Lalani - 7:04 PM
It's an art tool that I use in my work and I was able to get like a

Colin McNamara - 7:08 PM
Okay, maybe if I unmute myself, this will help. And again, this is free. There are a lot of people trying to sell everything on earth. That is not us. No, I don't work for LinkChain. I'm a managing partner for engineering and finance for local consumer packaged goods companies called Always Cool Brands.

Colin McNamara - 7:08 PM
We make products for retailers and grocers, like juice boxes, gummy bears, and we take bad stuff out of it and replace it with good stuff. Thank you. It's a great job. We have some gummy bears going into 400 stores pretty soon that have replaced Red 40 with Beets and stuff like that.

Colin McNamara - 7:09 PM
A few years ago, I ran into LangChain because we were working with manufacturers who are really our clients to take a beverage product to market and I ran into the child data challenge from the LangChain project and used it to create pitch decks for investors and stuff like that. And what it actually uncovered was a fraud in my supply chain. It saved me from an SEC violation.

Colin McNamara - 7:09 PM
And I was like, holy smoke, this is a rad project. I need to share this with people around me. I need to connect people who are using it. And over the past probably six months of this group, we've been seeing awesomeness happen as we're all just learning in the open. So yeah, I got nothing to sell you, man. But hey, when the gummy bears go to market or some of the juice boxes, I'm all happy. You want to drink some juice, get some gummies, I'll hook you up for free. Okay, so Ricky gave a little bit of a 101.

Colin McNamara - 7:10 PM
One quick start going coming on, they'll kind of get you prepped. And this is our AI imagery and gaming session that we're doing today on 4.3 in Austin, Texas. The QR code here, this is our repo, we have a basically a hacking repo where we put presentations, we put

Colin McNamara - 7:10 PM
all sorts of fun things. So if you go to github.com slash column McNamara slash Austin underscore lang chain, you'll see some folders inside of here, right? So here you have labs. What's inside of labs? Well, we have our meetings, our in-person meetings just like this, our first one, our second one, our third one, our fourth one, our fifth one, and this is our prepping for the one the next one.

Colin McNamara - 7:10 PM

Coming up is one of six, but we're in one of five right now. But there's a whole bunch of labs that you can go through and notebooks. And for those of you that got set up with your keys, you're new to this project, maybe you go to one-on-one and you go to Streamlit streaming, and you click open in cloud. Well, you can start following along and literally just press play and run anyway. And you can start coming through that you can start running through these and start playing with it on your own.

Colin McNamara - 7:11 PM
So we'll go ahead and host like a full three-hour one-on-one quick start, you know, every once in a while, we got to figure out when the next one is scheduled. Cool, again, this project and the people inside of it had terribly impressed me. The people that we connect with here in Austin continue to impress me. And people are using this project are really staying on the cutting edge. And we're finding for those of us in the core group, who work together.

Colin McNamara - 7:11 PM
They're contributing code. We're finding that, even though I feel stupid all the time, all the time, that we tend to be on the cutting edge. Just by the fact there's a new release, there's a new thing, we're playing with it. And some of what we're talking about here today is on the cutting edge of AI technology here in Austin. How weird. Okay, so we're going to go message, give some news and announcements, and then we're going to go and talk some labs.

Colin McNamara - 7:12 PM
The first one will be integrating AI with web applications with Paperspace. James behind me? James, right? Yes, okay. Next, and why James is going first, one, it's cool, but two, it allows you to basically access hosted GPUs. If you don't have a 4090 card at home or GPU at home, like, my only GPUs are on my Mac.

Colin McNamara - 7:12 PM
You know, you have to rent GPUs from somewhere else to run some notebooks, run some labs. So, this will get you set up for the second talk we have today. And again, you might not be able to follow along quickly all the time, but it's recorded, all the resources are there, all the stuff's in the repo, so on and so forth. Next, we're going to follow up with a thing that was really cool. At our Hacky Hour, two weeks, four weeks ago, excuse me, the last Hacky Hour.

Colin McNamara - 7:13 PM
I was hanging over Kareem's shoulder, being like, dude, what is on your screen? Our Hacky Hour is where we come and we bring our laptops to a local bar, we hack on code, we hack on labs, we hang out, we drink beer, eat pizza, whatever. And Kareem was showing me some really cool stuff with Stable Diffusion, how he had this really cool image workflow. So, what Kareem did was he integrated it with LangChain. And so now, LangChain, this Python project we use ...

Colin McNamara - 7:13 PM
he's actually like making images and stuff like that. So he's going to go over and do a showcase on that. And then, last, we have D Manning, are you here? Sweet. Okay, please, with your laptop, before it comes to you, get all set up on the sessions interface, so we can make you a presenter. Oh, and did you get the email with your login information? Okay, cool. It may have come like a last minute, we might have to fudge ...

Colin McNamara - 7:13 PM
around, is Don Manning, right? Dan? Dan Manning, okay. Dan, local Austinite game developer, made a really cool application where NPCs in games are interacting with each other in a virtual world, all in LangChain, and you can interact with it. Kind of neat, kind of blows my mind. So again, those are the talks and labs we have going over today. I'm really impressed by everyone and everything.

Colin McNamara - 7:14 PM
The password for the Wi-Fi is on the clear thing that's floating around. Thank you very much for asking for that, and my apologies for not... One, two, zero, four, yes, there it is. Well, thank you. And I appreciate everyone's help. It's hard when you're facilitating to keep everything straight all the time. So, if I go off on a rail or forget something, hey.

Colin McNamara - 7:15 PM
We're local in Central Texas. We do have people connecting from all over the world on our Discord and via our virtual interface on meet.amug.org. So, our Discord, the link is there. I encourage you to join it. We have some chatter that happens on there. It's where we do our community call, excuse me, it's where we do our office hours.

Colin McNamara - 7:15 PM
For example, we have someone in Brazil right now that I think Ricky's helping out or Kareem's helping out. Excuse me, he's using Ricky's code for your Google Drive code that went and fit my G-Drive. And Kareem's helping him out

on there. So, running through labs, you run into problems, hop in there. We're here to help. This is a community organization. It's for public good. It is a public good. We have, again, the GitHub on there.

Colin McNamara - 7:16 PM
It's under my account right now. Eventually, I'll move it up to an organization account or something like that. As this group is growing, I think we have 350 people or 360 people in the meetups. It's kind of getting scary big. Our meetup is right there. You can go to meetup.com. I do announce on Twitter at Austin Langchain. Usually, just like what we're doing, when we're doing type of thing. If Twitter's your...

Colin McNamara - 7:16 PM
thing, we have a YouTube channel where the videos from today will get produced on there and edited out. If there's anything we got out, you know, someone like accidentally slipped a key or something like that, and we'll be there for your review. And again, if you're in the sessions interface, you have access to the transcripts, you have access to the recordings, you have access to all the notes and stuff like that.

Colin McNamara - 7:16 PM
So hopefully you can make stress of it. We host Austin Lang Chain, generally monthly in person like this. And then we do in the middle of the month, we've been doing hacky hours, where we get together at local bars, explore the cool things Austin has to offer, hang out with each other, and show off our code. We've been doing, we are consolidating the virtuals, because honestly, it is a ...

Colin McNamara - 7:17 PM
It's a lot to do a virtual from home, do an in-person here and do a hacky hour. And let me see, but our focus is low stress, right? So we want to learn and share, and again, it's low stress learning and sharing. We want to connect with other early adopters that are inside here. There's a lot of doers, a lot of makers, a lot of people that there's business people that are effectively integrated. Rob's a great example of that.

Colin McNamara - 7:17 PM
You know, for your energy investing and stuff like that. Where's my mind? What he's been able to do from sitting here, you know, I remember you're like, I'm not technical. I'm not technical. You're like, oh, yeah, I get Lengchain, gives me a report every morning of all this cool investment stuff that I'm doing. Maybe not so technical according to you, but you're using it for business use, and you're doing stuff. I think it's rad. Again, our focus is learning, sharing, and growing.

Colin McNamara - 7:18 PM
We have a very simple code of conduct. Be cool. Like, don't be uncool. Be cool to each other. Don't be gross. We know what that means. If anyone doesn't understand that or has a problem with someone being uncool or being gross, come to me. I'm a Marine. I'm a firefighter. I'll be dealt with. And the most important thing about being

Colin McNamara - 7:18 PM
cool is learning. I've done this eight times in my career where I go from being like an expert in hyperscale or an expert in global voice or global storage, and I go down to something I'm really stupid at. And we become toddlers. We really become insecure. We become unsure. We become really vulnerable. So really embrace. Be cool to each other. You know, just because, you know, if any of us are super smart or in different areas,

Colin McNamara - 7:19 PM
we're all learning, right? So give each other some grace. I want to make time to thank our supporters. Meaning Cannot allows us to use this amazing space for free. Amazing. Thank you. If you're ever in trouble with the law, please use them. Or if you plan on getting in trouble with the law and you need an opinion, use them. I want to thank KeyChange who focuses on like...

Colin McNamara - 7:19 PM
early venture business growth for being there and our name tags and trials and being a positive energy and helping facilitate a lot of really great stuff in the community here in Austin. I want to thank my partners at Always Cool Brands for allowing me to spend so much time fostering this community, but also for our sessions interface and other things like reserving tables

Colin McNamara - 7:20 PM
and hack hours and stuff like that. Most importantly, I want to thank our contributors for our code, for our content, but also consuming it and contributing our community. So, news and announcements. Does anyone have any questions about what we're doing, how we're organized? We are a public good. We are learning in the open.

Colin McNamara - 7:20 PM
Occasionally, people who work from companies may present and talk about that. But, you know, it's mainly because we think they can add value to the community. We do not take compensation for what we're doing here. Okay, news and announcements. All of our webinars are scheduled in two weeks at Central MachineWorks. That is on the east side, on like 5th, kind of near ... Huh? Am I ...

Colin McNamara - 7:20 PM
Turn around. That way. Okay. Cool. I live on the east side. I should know this. You do, too. That's my favorite bar, and it happens to be where I posted a thing on Meetup, and Charles was like, Yo, what's up? And I'm like, I got an idea. I can get you through some ... I want to share what I've been able to do for myself. And I thought it would take like an hour, and we're there all night. But, you know, as you know, sometimes these things go wrong.

Colin McNamara - 7:21 PM
I'm just letting you know. So anyways, we have a hacky hour on the 17th at Central MachineWorks. It is no stress. Bring your laptop, hack on labs, hang out. I got three tables reserved. We have plenty of space. Be cool to your servers. Grab a beer. If you're not into beer, grab something non-alcoholic. The pizza's great, so are their nachos. Let me see. What's up?

Colin McNamara - 7:22 PM
So to echo back for people that couldn't hear online, or who weren't clear, we're throwing beers for people to do cool stuff. Right? And if you don't drink non-alcoholic beverages or whatever, I don't care. I think it's cool to just give someone a high five. There's stuff they respect. There's things that are really interesting to us. I've merged somewhere in there. I think on the Discord.

Colin McNamara - 7:22 PM
We've got some really some interesting code examples for code to control PII escape to be really cool. And I'm personally working on some stuff. But the whole idea is that we can start throwing some positive reinforcement around for people that are creating patterns and then distributing them out in the community. I think it's really, really neat. Beer, pizza, I don't know. We'll see where it goes. If anyone else wants to join us in throwing up, throwing some beers in the community.

Colin McNamara - 7:23 PM
Kitty, for people showing off cool stuff, you know, talk to me, talk to Charles, hop on the Discord. Okay, let me see. Anything else we can cover about hacky hours? It seems to be a structure that's happening. One time in here, doing formal stuff, presentations, recording, whatnot. And another time out on town enjoying this wonderful town that we live in. And let me see. So office hours, we've been experimenting.

Colin McNamara - 7:23 PM
The last two weeks, we're going to continue, I think, yeah, at two o'clock on Tuesdays, hopping on the Discord voice and video channel and doing office hours. It's been really great. We've had people pop in asking for help, for having discussions, we've been hanging out. I know that I've gotten stuff figured out inside of there. It's really great.

Colin McNamara - 7:23 PM
I'm having a break midday on Tuesdays and chatting with people and luckily Kareem, not only is he really, really smart, so is Ricky, but both of them are really, really cool. And there's some really cool discussions that are happening inside of there. Our community calls are at 2.30 on Thursdays. Generally, those are more structured meetings where we're planning our events and stuff like that. So if you are a person who adds value to a community by helping make things happen,

Colin McNamara - 7:24 PM
you don't have to be like a super cool hacker person, right? Hop on! The more, the merrier. Other announcements, we posted a video exploring LandGraph from our last virtual session. As we got a good recording, I think our in-person weren't able to capture recording. So if you want to review any of the LandGraph stuff, it's up there on our YouTube channel. And then last announcement ...

Colin McNamara - 7:25 PM
And to echo for those that are ... Sorry, what? So to echo, Catherine, I'll be there. I've been bringing in Plus One. I'm going to go shoot some nerds. I say this lovingly. I am a nerd, right? We recognize ourselves, but y'all going to get shot with some laser. I'm going to do like a combat role. You'll probably shoot me in the back. It'd be cool. So again, that's at 12 noon on Sunday.

Colin McNamara - 7:25 PM
There's people in the core group who do the work, that write notebooks, that create code. Really happy today that we have people coming in from the outside to talk about ... new people to the group, bringing from Paper Space, giving talks, from a local community, and I'm forgetting the name of your video game company. But please, if you want to talk about something, if you've done something ...

Colin McNamara - 7:26 PM
if you want to showcase something that's very ... you think might be a little too simple, we want to give you the visibility, we want to give you the stage, we want to give you the support. So please hop on the Discord, hop on the community call, we'll coach you through it. It is a safe place to be a nerd and show off what you're doing. This is not the call-in show. This is the show for our group. Okay, we talked about our last meeting. If you want to review it, it's right there.

Colin McNamara - 7:26 PM
So again, everyone log into Sessions, and then I'm going to hand it over to James. So James, are you on the Sessions interface? Okay, I'm going to put my glasses on so I can see you, and we're going to flip things on over. I'm going to sit back and learn some stuff. James, I don't ... let me give you a moment here, I'll figure some stuff out. Make assistant. James Skelton.

Colin McNamara - 7:27 PM
I don't have any hold music, but we can imagine some like Girl From Ipanema going. Let me make you.

James Skelton - 7:27 PM
I'm going to put myself on mute real quick. Okay. So we can all see this now? Yeah. That didn't break it? Good. I was worried. All right. Hi, everybody. Thank you so much, Colin, Kareem, all of you, for having me. I really appreciate it.

James Skelton - 7:28 PM
And thanks so much for your time today listening. I'm James Skelton. I'm a technical evangelist for Paperspace, now by DigitalOcean. That's a recent thing. I'm still not used to it. And today I'm going to talk about integrating AI web into your web applications with Paperspace. Got a nice little session prepared for you all here. I'm going to start by introducing

James Skelton - 7:28 PM
Paperspace. Just kind of, you know, give everybody a quick taste of what it is before we dive into it. Then talk about developing with language models on Paperspace. Talk about our GPU offerings. Then I'm going to talk about the Surge application, which is what I'm going to be showing today. It's a fast API and landchain-enabled... I guess you could say...

James Skelton - 7:29 PM
...serving application for different GPT models. Talk about how it uses landchain. And then we're going to talk about how to actually use Surge with Paperspace. So creating a deployment through the UI and interacting with your deployment through the API endpoint. Then we're going to actually look at the demo itself. I've got one running, and you can go along with me if you so choose. Don't totally recommend it right now.

James Skelton - 7:29 PM
I'm going to be running on an A100, and that stacks up if you're not paying attention. And at the very end, I'm just going to talk about what I would like to talk about next time I CEO, which is deploying with LangServ and Paperspace. But I didn't have time to develop a whole proprietary application this time around. You know, I just realized they can't see me. Does that matter? Oh, you can. Okay, cool.

James Skelton - 7:29 PM
Alright, so let me give you all my Paperspace pitch. So what do we do? Who are we? In short, Paperspace is an MLOps and Cloud VM platform specifically focusing on giving GPU power for AI model training and deployment at scale. Specifically, we have two main products to know about. They are virtual machines on core and our

James Skelton - 7:30 PM
MLOps platform on Gradient. This is mostly talking about Gradient here, but with us it's pretty easy to launch an IPython notebook with our notebooks, do your training with a core machine, and then deploy using the deployments. It's definitely built for AI developers specifically. I should know, I torture myself trying other tools all the time.

James Skelton - 7:30 PM

It's got a really nice wide variety of GPUs, which is really my favorite thing about it. So if you need to use an H100 for your task, which is coming up more and more these days as these language models get bigger and bigger, we have them, but we've also got as small and cheap as Maxwell M4000s, which can be really nice if you're running something lightweight. We've also got pre-configured deep learning containers to make it easy.

James Skelton - 7:31 PM
So if you're ever starting an IPython notebook with our notebooks product, the docker container has everything pre-installed. So you don't need to go through a lot of the setup you see in stuff like Colab. Gradient is the name of the ML platform itself, it's got all the tools in there, and it comes with a lot of really nice in-built features for collaboration and sharing. In fact, I had every intention...

James Skelton - 7:31 PM
of offering credits for y'all today. To make a long story short, we're recently acquired, integration is going interestingly, and I have to manually enter it, so we're gonna talk about it later, but the other option is I can make a team and just, you know, give everybody free GPUs for a week, so we'll get there when we get there. It's an option for collaboration and sharing to the extreme, is what I'm saying.

James Skelton - 7:31 PM
Yeah, this is just a little bit more about those. So the notebooks, those are gonna be really familiar to all of you that have used stuff like Colab or Kaggle, AWS SageMaker, IPython notebooks through Jupyter. Machines, those are just virtual machines that are connected to our GPUs, they can be run on Linux or Windows, and you can get either just direct SSH access, or you can get like the entire...

James Skelton - 7:32 PM
thing in your web browser, which I quite like to use from time to time. I play PS3 games on my core machine. And finally, deployments, which is what we're going to talk about today, which is converting the models into scalable API endpoints, and it's got nice in-built features like security, health checks, everything you need to really take your application to scale. I'm not gonna

James Skelton - 7:33 PM
even take that work, put it in a docker container. I prefer to do that on core, but you can do that wherever. And then deploy the application with Paperspace deployments. You can access the 8xH100 or A100 GPU, multi-GPU machines on our VM platform. So if you're ever doing training or fine-tuning, that is 100% what I would recommend. And integrations with Hugging Face, AWS.

James Skelton - 7:33 PM
And DigitalOcean Spaces make it straightforward to plug and play with your different models if you build a templatized application. Question? So is the idea with Paperspace that you can deploy a large open source model, but it's your own private deployments? Precisely. And how do you figure out how many nodes you need, how many GPUs?

James Skelton - 7:34 PM
And how do you synthesize your deployment to match the compute needs, and then match the machines within VLink? So it keeps things smooth as you need to scale up, which is a really nice feature. But yeah, I'm going to show the deployments tool itself in a lot of detail.

James Skelton - 7:34 PM
So whenever you're using an OpenAI API call, it's doing all the compute on their end. If you wanted to set something up with an open source model where you're controlling the entire...

James Skelton - 7:35 PM
environment, this would be a good solution. Honestly you could run this on a CPU and do API calls too, if you need to, and then just abstract away the GPU problem. But there's plenty of services that do that. I would like to talk about what makes this special today, right? This is just a full list of our GPU offerings. I recommend perusing it from time to time, just see as we grow what else is there.

James Skelton - 7:35 PM
Hopefully H200 is coming soon, I can't promise that, but we've got everything from Pascal to Hoppers, which is really nice when you're not trying to break the bank, just testing out a new model. Fire away. I don't know anything about GPUs, like that whole list of 3100X, I don't know anything about it, Matt. I want to learn though, because I think it's important.

James Skelton - 7:36 PM
So if anyone has a good resource, maybe put it in the chat. I will. I can definitely do that. That's our whole, yeah, paper space blog got yet. I want to know which one to choose, and I want to figure out how to do it. I mean, quick nutshell, M4000s are about 12 years old, Pascal 4000s are, I don't know, 6 or something, I think. Then RTX 5...

James Skelton - 7:36 PM
or 4, A3, Voltas are actually older, but yeah, the terms stand for microarchitectures, they're all named after different mathematicians and scientists. That's what NVIDIA likes to do, I guess. The new ones are Blackwell, for what it's worth. But yeah, I will share a resource on what this all means, absolutely. Okay, so

James Skelton - 7:36 PM
what is the Surge application, what is the thing I am presenting on today? I appreciate y'all listening about paper space, let's talk about something a little more fun. This is just a quote directly from their GitHub repo, but Surge is a chat interface crafted with Lama.tpp for running GGUF models. It requires no API keys and is entirely self-hosted. Kind of hitting on what you just asked, coincidentally.

James Skelton - 7:37 PM
It's an application for serving large language models with a Svelte kit frontend, Redis chat history and storage management, that's actually done by LangChain, and a FastAPI plus LangChain API to wrap calls to Lama.tpp using Python bindings. In practice, this just lets you speed everything up while you're running on your GPU, and it's packaged all in one tight little application.

James Skelton - 7:37 PM
You can get everything together in one spot. I'll share links to this later, but they're also in the slides. But this is the GitHub repo for Surge. It's a really cool project, please star it. If y'all like this presentation, I would love to support them some more. I screwed that up. That's good enough.

James Skelton - 7:38 PM
So, how does Surge actually use LangChain? I went into the application files, digging in, trying to find where they were actually implementing some LangChain features. These were namely in the chat.py and stream.py, which makes sense if you think about it. And we can click in there, using those links to see them.

James Skelton - 7:38 PM
And I realize this is far too small to see from there, but LangChain.memory, import Redis chats, message history, and then it's using LangChain.schema to do the system message, messages to dict, AI message, and human message. So what do these do? Well, the Redis chats history is for memory management of the chat, it organizes and stores all of your different chats using Redis.

James Skelton - 7:39 PM
You can see it across the application, which is really nice and useful for managing a bunch of different chats. Here is the actual application itself. You can see right here, I've got two chat windows open, but I think it can support as many as your storage can handle, which is really, really useful. And the schema-related functions, like AI message, human message, and system message, they do the organization for your chat history.

James Skelton - 7:39 PM
So if the LLM is sending a response, it'll be AI message, human message is your inputs, and system message is logs, basically. These really help to organize the entire conversation and make the application run more smoothly and more efficiently. They also functionally log the interactions between the user system and model, enable quick access to previous chats.

James Skelton - 7:39 PM
They log by the Redis database and enable a ton of different quality of life features like history purging in an application. It really simplifies a lot of the things that go into managing a chatbot, which I really like about the application, and we owe it to LangChain here. So I wanted to make sure we gave that credit there. Next, I'm going to dig into how to actually deploy...

James Skelton - 7:40 PM
search with Paperspace. So in the GitHub repo for the Lab 5, there's a readme in the Paperspace subdirectory. So Austin, LangChain Labs, LangChain 105, there is a subdirectory called Paperspace, and I put in a readme with the instructions.

James Skelton - 7:40 PM
This is a bit of an experiment, I have not put this into text format before, I've just talked it out with people or shown schema. So if this isn't particularly clear, give me feedback and I'll update it for sure. But I'm going to walk through the entire process here. So, how to actually launch the deployment. I'm going to sit for this part.

James Skelton - 7:41 PM
So, the first thing we will do is go to console.paperspace.com. If you don't already have a Paperspace account, this is where you can sign up. We do have free accounts, just so y'all know, but if you're ever using a GPU it will cost money, but there's never any hidden fees in my experience. And I've been using it for three years.

James Skelton - 7:41 PM
Once you're in there, this is just one of those teams I was talking about where you can do different collaboration. I'm just going to jump into this project. This is just where I share a lot of the work. I have my GTC demos here, for example. And then go over to the deployments tab.

James Skelton - 7:41 PM
Now that we're in the deployments tab, we can go right here in the middle and hit create to actually create the deployment. And then, actually, I've put all the things you need for this setup right here in the demo slide.

James Skelton - 7:42 PM
So just a heads up, that can make it a little bit easier. We can just go through here, quickly select our machines, and set. I'm going to use an A100 80 gig GPU for this. This is the most powerful available on Gradient right now, although the H100s are coming online pretty soon for Gradient. Again, I don't recommend it right now because it's 318 and out.

James Skelton - 7:42 PM
P4000s are 50 cents and they will run this just as well, just probably not as fast. Do you have any guidelines on GPU sizing for workloads? How do you figure out whether you want to go for a 4 or 5 year old GPU for 50 cents an hour versus an A100 or a H100?

James Skelton - 7:43 PM
Yeah, general answer is look at the VRAM and look at the bandwidth and look at the CUDA cores. Those are three values for GPUs that kind of correspond to strength. CUDA cores. Yeah, so CUDA the software layer for NVIDIA GPUs but they have...

James Skelton - 7:43 PM
specialized cores that are like... they're the, you know, they're the mini computational modal in the GPU itself. I know that's not the right word and I apologize. Hardware is not my thing. But core. The CUDA cores themselves are the tiny individual units that are doing the computations. So with each successive generation...

James Skelton - 7:44 PM
the CUDA cores advance and they were able to use that and a few other less intuitive things to increase the bandwidth as well. But generally, so like Hopper has the same VRAM, like an H100 and an A100 both have 80 gigabytes of VRAM. But Hopper has significantly higher bandwidth and significantly higher CUDA cores.

James Skelton - 7:44 PM
So if you are trying to do something more quickly it'll be more efficient. But I think Hopper for us is 594 and A100s are 318. So you have to kind of do the calculus there about time, how much time you're gonna need to actually use your thing. Really the question honestly is are you training or are you

James Skelton - 7:44 PM
doing inference. If you're training the answers almost always go as big as possible. If you're doing inference serving a model then things get a little trickier. Yeah that's a good question. It's not a solved problem that's for sure.

James Skelton - 7:45 PM
Yeah basically if you overwhelm the number of requests it's able to handle it'll attach another machine. Everything on Gradient specifically is...

James Skelton - 7:46 PM
I'm worried I'm going to say something wrong here. It's okay. Remember, we're cool. Everyone has to be cool. It's okay. I've said some stupid shit. And what's cool is everyone's like, hey... I see you back there, by the way. Sorry, no, go on. So you asked about vertical scaling on GPUs. Uh, you didn't.

James Skelton - 7:46 PM
Let's see. Let's just go to the docs. Uses Kubernetes horizontal pod autoscaler. So it looks like it's horizontal, not vertical. Well, okay, language is getting...

James Skelton - 7:46 PM
confusing there. This is not my...I make cool demos. It's out of my wheelhouse, I'm sorry.

James Skelton - 7:47 PM
I'd be surprised if they gave you a bigger instance. I think it's probably...

James Skelton - 7:47 PM
you know, not enormously so, but enough that we don't bother right now." Okay. Where was I? I was in here. Creating the GPU. Yeah. Oh, I already filled in this one. Okay, cool, I don't have to go back and forth now. For the image... I'm sorry, the name, you can name it whatever you want, I'm just going to call this

James Skelton - 7:48 PM
For the image, we're just going to use something that's on a public container registry. This is on NV... I get this wrong every time. NVCR. NVIDIA Container Registry. And we don't need to add a default command, it's automated already in the Docker container. Next is the resources section. Here, if we...

James Skelton - 7:48 PM
wanted to change our port or add another available port, we can do that. For this example, we're going to use 8008. I believe it defaults same as Jupyter to 7860, so if you don't feel that, just heads up. Next is the replicas and autoscaling. Here, we can initially set the number of replicas we want for our machine at the start, and then...

James Skelton - 7:48 PM
we can set parameters for how autoscaling will work. So a maximum number of replicas, the request duration, CPU utilization, and memory utilization thresholds we want to cross before it triggers the autoscaling to start. It's a little more descriptive than what I had before. Here are just some of the quick integrations. So if you wanted to do something like mount a, in this case, GPT to me...

James Skelton - 7:49 PM
This is the medium they have as their example model. It'll put this in the mount path that you set here. This can be really useful for, like I said, plugging and playing different models with your application. DigitalOcean Spaces and S3 Buckets are two different storage solutions you can use. I'm very new to DigitalOcean, but I quite like their spaces. They're about as convenient as S3.

James Skelton - 7:49 PM
If you haven't tried it, I recommend it. Environmental Variables. Let's say you're running a Gradio application, and you need to set the Gradio server port, Gradio server name. That's a very common problem when dealing with the cloud and Gradio applications. You would do that here, manually. And finally, we can do things like set health checks for our startup time, liveness, and readiness.

James Skelton - 7:50 PM
To tell us when we are, you know, verify whether our application has started, whether it's in a healthy state, and whether it's ready to serve requests. Finally, endpoint security. We can add a deployment key secret if we would like to add some degree of security to our deployment endpoint. Seems very familiar. Looks like it's a front-end, GraphQL front-end for Kubernetes deployment. Pretty much.

James Skelton - 7:50 PM
This whole thing is also... we have a CLI that does the entire same thing, and I'm pretty sure it just wraps around that. And finally, price summary. Because I chose 14 A100s, 44.52 per hour. Depending on the scale, that can go up to 47.7, so it gives you the range. And, like I said,

James Skelton - 7:51 PM
I've had bad experiences with billing, and this one is pretty intuitive. I've been very happy with it. Once that's all done, you can just hit deploy, and it will take you to the deployment homepage. Going back here. I've already got one running for the sake of time. And here on the deployment homepage, we can see insights like...

James Skelton - 7:51 PM

how many replicas we're running, how much traffic we're getting, CPU percentage, RAM used, etc. We've got our endpoint URL here, and some information about the replicas we've got running. In this case, it's just one. It's healthy. I can jump over to the logs and see what's going on underneath the hood here. And if we're iterating...

James Skelton - 7:51 PM
on the application for whatever reason, we've also got a history feature. So you can download previous configurations, which are all stored in JSON or YAML format, and then relaunch it. Alright, now that that's all finally gone through, let me just make sure I don't have one more thing to cover. Cool. If we want to interact with our deployment through the API endpoint, all we need to do is click the...

James Skelton - 7:52 PM
URL in the top right corner of the deployment details page. From there, you're going to need to download some models. There's a pretty wide variety there. I was kind of just made to see, unlike my other favorite tool, LlamaBoard, I want to shout them out, and I will again later, that you can't just plug in a URL from HuggingFace and download it, but it's got a pretty robust list.

James Skelton - 7:52 PM
And from there, you set your parameters and begin chatting. And I've got some more detailed instructions on how to do that setup in here. But it'll be very straightforward to all of you, I think. It's pretty traditional LLM interface. Yeah, say hi to Surge. It was originally designed as an easy way to chat with Llama-based models, but since

James Skelton - 7:53 PM
it's grown a fair bit. Alfred, Code, Code Llama, Falcon, Open Llama right here. It's a favorite. Yeah, pretty good list. Today, for this session, I downloaded Mixed Drill, 8x7B, Dolphin, and Phi2. Just kind of give examples of two different

James Skelton - 7:53 PM
size models running on here, and just showing what we can do with them. To start a new chat, you can use this little sidebar here on the left, and put in your settings. So we've got temperature, which controls how random the generated text is, maximum number of generated tokens, goes up to 37768. That's pretty cool.

James Skelton - 7:53 PM
Context length is at a maximum of 2048, which makes sense. Number of layers put on the GPU, so if you are running a CPU optimized... I'm worried I'm going to get my acronyms wrong here. Is that GGML or the... GGUF? OK, thank you. So if you're running something GGUF and you've got a...

James Skelton - 7:54 PM
We actually do offer crazy CPU machines, so if you wanted to do that on a CPU machine, that is very much an option. And then, you know, other things like repeat last-in, repeat penalties, number of threads to run LOM on, etc. Here's the model choice. I'm going to do... Let's just do Phi2. And finally, the prompt template. So this is the template it uses to handle...

James Skelton - 7:54 PM
your requests. It's kind of defaulted to an instruct response setting, but you can put whatever you want in here as a way to sort of prime the model. You know, because it's generating each token as it goes, that initial context will really affect the output. Once you got that all set up, we can start a new chat.

James Skelton - 7:55 PM
My new favorite test is to ask it, tell me about the mathematician Blackwell. I feel like caps matters. I already asked Mixedrel this a while ago and it didn't get it right. This looks right! Oh my goodness!

James Skelton - 7:55 PM
Oh, but it did just dump a bunch of empty characters at the end. Yeah, so Phi2 is a 1.6 gigabyte model, I think. 8. So it's remarkably small and we're running on an 8100 right now. Phi is a really cool project by the way, I mean, you know, I don't love to shout out my...

James Skelton - 7:56 PM
but shout out Microsoft. So they're not in memory, they have to be loaded into the pipeline, and that's what it's part of what's going on under the hood when you start a new chat. Yeah, so we can just, I'll do a quick example right now, do that

James Skelton - 7:56 PM

mixture one, start a new chat. So the window comes up instantly, but, did I spell that right? Yeah. It's gonna take a little bit longer to get it all together, and it's gonna take a little longer to do the actual setting, and we can see that in the logs if it will refresh.

James Skelton - 7:56 PM
Is that? No, it is active. Okay, so it looks like this is what it says while it's loading the model up. So it doesn't look like this is good. Yeah. Oh, no. 56. It may have just loaded incredibly quickly. I may have to eat my words here. Yeah. Also,

James Skelton - 7:57 PM
is it David Harold Blackwell? Any mathematicians know the answer here by any chance? Anyone? I don't actually know. It's very confidently correct. Incorrect. Yeah. Okay. No. Oh, no. Nevermind. He was African American.

James Skelton - 7:57 PM
I thought it said African American studies, and I was like, that's very not right. But yeah, while that's going, last thing I wanted to show is you can, the line chain built-in features allow you to do all the database management for the chat, so we can go back and look at old ones.

James Skelton - 7:58 PM
Like this fee one, I asked hi earlier, oh look, a bug, it's still printing from the previous one. Well, I didn't make this application. But yeah, that's the fun of it. I loaded a 2048 context length, so this is going to be a second.

James Skelton - 7:58 PM
Oh look, it's repeating. Well, not everything is perfect, is it? But especially live demos, right? Let's clear the chats. That'll wipe everything. Like I said, Redis makes that really, really easy, so thank you for LangChain for that. Helping me deal with my buggy situation. Are you fixed?

James Skelton - 7:59 PM
I think with that, I am pretty much finishing up here now. I've been answering... Oh, I'm loaded in the Lana 13v model, so it's going to take a little bit longer to actually respond. With that, I'm pretty much done here. The demo instructions, everything that I did to actually launch it that...

James Skelton - 7:59 PM
Kubernetes setup, as Karim pointed out, is all right here. Those are the only values you have to paste in and change, so it should be very straightforward if you wanted to copy this. The only thing I recommend is changing that A100 and the machine codes are linked right there, so it'll be easy to change it. Next time I'm here, and I would really like there to be a next time, I hope...

James Skelton - 7:59 PM
I hope you all enjoyed this. I'm going to try and develop a LangServ application. I was digging into it a little bit. It looks really... I like FastAPI, so something that works with FastAPI, I'm a fan. And the invoke, batch, and stream endpoints particularly caught my eye. And I'm very curious if I can build something with JavaScript, too, to play around with it again.

James Skelton - 8:00 PM
JavaScript developers, you may hear from me, too, if anybody wants to collab. Yeah, thank you guys so very much. All the links are in here, in the readme, so the slides and the JSON. So we should all be good to go. Thank you all so much for listening. Awesome.

Karim Lalani - 8:02 PM
lot of cool stuff out of it." So, I'm going to let Kareem talk. Kareem is getting his stuff figured out. Kareem is one of the board members. He's contributed a lot of cool code. Definitely I learn from Kareem every day. And Kareem is ready to go. So again, this is, shall I let Kareem talk.

Karim Lalani - 8:03 PM
Thank you for offering this platform, bringing everybody together, for learning in the open. And thanks, James, for the introduction to PaperSpace, because that's what we'll be using for this demo, because for this one, I had to load three models and do some things that Colab doesn't.

Karim Lalani - 8:03 PM
Colab doesn't support out-of-the-box, so yeah, so for this, so this is going to be a land graph agent, there's going to be function calling, we're going to be loading multiple LLMs, we are going to get, you know, read context from images,

this is what we did in the previous one, but we're going to take it a step further.

Karim Lalani - 8:03 PM
and we're also going to do image generation using automatic 11.11. So let's get started. Like I said, so we'll be using, loading local LLMs on this one, no OpenAI, no Anthropic, it's all going to be local, like I said, multiple language models. We'll use Baklava model to read the image, context RFA.

Karim Lalani - 8:04 PM
Image, essentially image to text, automatic 1111 for image generation from text, and for simple question answering, we'll use the LLAMA2 model. Because this exercise needed a bit of function calling, and because I was working with local models, I had...

Karim Lalani - 8:04 PM
I had to employ a fine-tuned LLM, and I'll talk about that a little bit as well. And we'll have two custom tools, and all of the code is in the repo. And I just put the link to your code in the chat, as well as your book agenda, so maybe I can help you with that.

Karim Lalani - 8:05 PM
Okay, so Automatic 11.11, most of you might have heard it before, stable diffusion. It's basically a front-end that allows you to load stable diffusion models and do image generation against those models. Here's a screenshot of what it looks like.

Karim Lalani - 8:05 PM
This one, I had loaded a DreamShaper XL model, gave it a prompt, you know, an astronaut drifting away in space, and I don't know if you can see it, but this is what it generated for me. Now, to generate a model, it's, I mean, yes, I only gave it the text, you know, a small line of text.

Karim Lalani - 8:06 PM
But there's a whole lot of other data that needs to be provided to the model. And some of that you can piece together by looking at this nice little UI here. There's the sampling method, sampling steps, whether I need high-res on this one, whether I'm using a refiner, what is the resolution, what's the batch count in each batch, how many images I want generated, what is the CFG scale. And there's...

Karim Lalani - 8:06 PM
There's a few more of the things that go in here. Now in order to use that, Automatic 11.11 does allow you to launch in API mode, which means that you can not only interact with it using the web interface, but you can also make web calls to it. If you know the schema.

Karim Lalani - 8:06 PM
You can provide all of that information to it, and it will return back the images that it generates out of those. Now because we are making calls to an external system, we need something that you might have heard before called function calling. OpenAI popularized it with their models. Lanchain, some of the most interesting demos that I've seen with Lanchain.

Karim Lalani - 8:07 PM
Lanchain involves function calling. Because without functions, you're just chatting with a language model, and you can only do so much. In order to go beyond that, you would have to essentially give access to external tools. And in this case, the tool happens to be automatic 11.11 through the API.

Karim Lalani - 8:07 PM
I use Mistral for my stuff when I'm running locally, 7 billion models because I don't have a very powerful GPU, and out of the box, the pre-trained model or the instruct model from Mistral turns out did not have any function calling in its training dataset. So what I mean by that.

Karim Lalani - 8:08 PM
If I ask it, if I give it a function definition in the form of, okay, you're a helpful assistant, use this functions if required, and it's a calculate tip function. It takes in the parameters of bill amount, which is a number, tip percentage, and both of those values are required. And if I say I need help calculating the tip, my bill amount is $50, I want to give a tip of 15 percent.

Karim Lalani - 8:08 PM

I expect, in return, something that'll take those two values and structure it in such a way that is consistent so that I can pass it to an external tool to work with. What I mean by that is, I need something to come back, say, oh, you need a function called calculate tip, and these are the arguments. This is what my ...

Karim Lalani - 8:08 PM
expectation is. But really what I get back when I give it all of that is, oh, here's some JavaScript code. I mean, it does a good job of piecing together, okay, you might want to call calculate tip function, pass it to this, but this is not what I'm asking. I need it to give me a JSON construct that defines the function I want to call, not to give me instructions on how I could do it in JavaScript.

Karim Lalani - 8:09 PM
Instead of Python. So for that, I, because, you know, so there's fine tunes available that you could use, Dolphin or Open Hermes. I don't remember which one of those does a good job of function calling. But in this case, I just went there and I created a fine tune on Python.

Karim Lalani - 8:09 PM
With the data set that I found online on Hugging Face, and there might be a lab in the future where I can go. I hope there's a lab in the future where you show us your fine tunes, it's super duper exciting. Yeah. So now going on to tools. Like I mentioned, there's two tools that we're going to use, text-to-image and image-to-text. How do you do, how do you create a...

Karim Lalani - 8:10 PM
What is your tool to use in Langstein? You define a function, and you decorate it with the tool decorator, as simple as that. Now there's stuff that you're not seeing here, text-to-image input is a pedantic schema that is elsewhere. But here I'm defining in the doc string, I'm giving it a prompt essentially, of what I

Karim Lalani - 8:10 PM
was expecting when I'm giving it an input. And there's some custom Python code that I created for this one, and I'll go over that in a little bit. But basically, I'm creating a config, passing the prompt to it, passing any additional keyword arguments that I might pass, in this case, you know, I'm passing it along. And if a seed is not specified, in this case it won't be.

Karim Lalani - 8:11 PM
I'm going and initializing a random seed. And once I have that config, I am making a request to the API that I am running for automatic 11.11 to this endpoint, which is what it exposes. And I'm passing the config, basically, as a JSON string. So basically, what config...

Karim Lalani - 8:11 PM
config is, it's really the request structure that automatic 1111 is expecting for this text-to-image endpoint. I've just created a parentic object to make it easy to work with. Whatever response I get back, I take the JSON body of that and return it. That's the first tool. The second tool is image-to-text. This is something that we saw in...

Karim Lalani - 8:11 PM
the previous demo, but basically, again, a function that takes in two parameters. First is the spring, the prompt. Second one is the base64 encoded image that I want to use as context. If there is no image provided, and we end up here, I want to return the error message, the canned error message.

Karim Lalani - 8:12 PM
And otherwise, I will basically take the LLM, bind the images that I have, and then I'll pass the prompt to it. And whatever response I get back, I have this little bit of, you know, I call the function strip on it, because for whatever reason, the response I get back, I don't know, I don't know what the response is.

Karim Lalani - 8:12 PM
What I get back includes either a new line before it or after it or some extra spaces. So just to keep it a little bit cleaner, I strip those away. Here is our, how the agent, line graph agent looks like. We start from the left and we just follow the arrows. We give it the prompt, you know, either ask it a question or, you know, ask it to generate an image or describe an image.

Karim Lalani - 8:13 PM
It'll, you know, it'll create the initial, you know, it'll take that prompt, put it into the state, send it to, you know, send it to the function identification tool. It'll, basically, it'll do the first LLM call over there. So where you're seeing the circle,

these are LLM calls that are happening inside the agent.

Karim Lalani - 8:13 PM
First call is essentially, look at the question and figure out what I'm asking, you know, whether I want you to answer this question immediately or whether I want you to use a tool to answer this. Based on that, it'll come up with one of two values, you know, outputs, either it's, you know, the question is something, it's a human question or, you know, it's a function.

Karim Lalani - 8:13 PM
question call. If it's a human question, then it just makes, you know, passes it to another LLM, the prompt itself, and then the LLM will give me an answer. We'll send it back. That's the end of that loop. If it's a function, then it'll invoke that function with the parameters and it'll give you the response. This is how the app looks like. It's a screenshot

Karim Lalani - 8:14 PM
The first thing I did was ask it its name. Again, not asking you to generate anything, just asking it a simple question. And it correctly identifies that, oh, this is a human interaction. So let me just respond with it. And because I'm using LLAMA2 as the model to answer that, it's responding it as such. Now, my function calling model is a fine tune of Mistral. The human interaction model is LLAMA2.

Karim Lalani - 8:14 PM
The image-to-text model is Baklava. So again, three different models, two of them built on the Mistral architectures, but again loaded as two separate models here. Then I gave it another prompt, said, generate a cartoon of an astronaut planting a US flag on Mars. This is what it came up with.

Karim Lalani - 8:15 PM
Now, the thing with stable diffusion and image generation is, you can give it the same prompts, pass it multiple times, and you'll get different responses. How do you get consistent response? Well, how do you get the same response out of it? That's where the seed comes in, because seed goes in as a parameter, and if you pass in the same arguments and you supply the same seed, you will get the same result.

Karim Lalani - 8:15 PM
How do you get the same image consistently? So, I didn't want the demo to just end here. It's like, what if you want to generate this image on your web UI using automatic 11.11? So you could look at the configuration, and you could say, okay, pass this prompt, here's a seed for that, and all of the other parameters that are going in to create this. If you take this, and you fill in the right...

Karim Lalani - 8:16 PM
you know, text boxes and sliders in automatic 11.11, load the right model, you will get the same image that was generated here. The other thing that will happen is, I gave it the message of, generate a cartoon of an astronaut planting a U.S. flag. But if you recall, the tool description said, you know, had a little bit of a docstring. Basically what it said was, look at the image, look at the prompt.

Karim Lalani - 8:16 PM
And convert it into something that I would pass to mid-journey, or something similar. If you look at the prompt here, it just says, an astronaut in a spacesuit, holding a U.S. flag, standing, and it's got something else. It constructed that from this prompt that I gave it. So, you know, even that's happening as part of your tool definition. You can, your tool definition has the ability to transform.

Karim Lalani - 8:16 PM
Transform your prompt. You know, all the resources that we use here, again, I'm using Olama for, you know, James used Llama CPP in his demo. I'm using Olama. They both, Olama uses Llama CPP under the hood, but it gives you an API interface. Llama CPP, on the other hand, is a library. So you have to construct that on top of that, which is what Surge essentially is.

Karim Lalani - 8:17 PM
It's using Llama CPP under the covers and it's giving you a nice user interface. Olama is using Llama CPP under the covers, but it's giving you an API front end for that. Automatic 11.11, of course, is the link for that. For the Olama model, I mentioned that I'm using FineTunes for function calling. The function calling model is hosted on Olama.

Karim Lalani - 8:17 PM

You can use it for your own, if you want to use the same format, you'll be able to use that for your own projects and stuff. The template is hosted on Lanchain Hub. So Lanchain Hub is one of those things that we don't talk about as much. It is also something that Lanchain offers. A lot of what we do with AI boils down to repeatability. We want to be able to use the same thing over and over and over and over again.

Karim Lalani - 8:18 PM
And Lanchain Hub is basically where you could store, publish your prompt templates for others to be able to use. So it makes my code a little bit cleaner because I don't have a lot of this text prompt in there. I'm just making a call to Lanchain Hub with my identifier.

Karim Lalani - 8:18 PM
And anybody else who would like to, you know, just use, who just needs a template, doesn't care about the code, what I'm doing, but they're interested in just the function calling bit, well, they can just go there and they can just use that bit. With that, let me just give you a quick run through of how you would run, you know, run this. Okay.

Karim Lalani - 8:18 PM
So, on the GitHub repo, lankean105, I am, am I not, okay, let's do sessions I believe, okay, so there is a notebook that outlines all the steps that you could run.

Karim Lalani - 8:19 PM
on your GPU, in a machine with access to GPU. There are some disclaimers here, again, you know, we are running three quantized models, which means that you need enough VRAM to be able to load those models, or you might want to have different instances because you're running automatic 11.11, which is hosted as a service, Olama, which exposes a service, so that could be running on a different machine as well.

Karim Lalani - 8:19 PM
But if you're running it on one machine, you need at least 20 GB of VRAM. I am, you know, I tested this on paper space, using the P6000 instance, which runs about, I think, 50 cents, or is it a dollar an hour, I think. Yeah. Now, the default model, the default container that it uses...

Karim Lalani - 8:20 PM
works for Lanchain, but uses an older version of Python, which doesn't work for Automatic 11.11. So, but paper space also hosts a whole bunch of other variations of the containers, and this is the container that I ended up using for that. And the nice thing is, there's this run on Gradient button, so if you do end up, you know, signing up with Gradient, and if you want to run...

Karim Lalani - 8:20 PM
this, or follow along these steps, if you click this, all of this information is pre-fed into this, so it will launch a P6000 instance on this container, and it'll load the repo in its entirety. Quick thing, that image is not cached, so it will take a minute. Yeah. It will not be a very fast setup, unfortunately. Which is why I actually...

Karim Lalani - 8:21 PM
Which is what, you know, it all adds up to about 20 gigs. And downloading that does take a little bit. Which is why, you know, as much as I would have loved to sort of walk through the steps, it would have taken far longer. So that's why I just went ahead and... But I will walk you through the steps that I did, so you can sort of see. That's great. I wish I looked at this before.

Karim Lalani - 8:22 PM
No worries. Yeah. So, again, this is the running instance. This is running on Paperstress at the moment. Again, I asked it, what's your name? And again, in a similar fashion as it was in the screenshot, it responded with, yeah, I'm Lama, you know, from MetEye, AI and all that stuff. I asked it, okay, portrait of an astronaut riding a motorbike on Mars.

Karim Lalani - 8:22 PM
It gave me this hideous looking image. But, you know, again, here are the parameters. Like I said, I asked it a portrait of an astronaut and it shortened it to just astronaut riding a sports bike on Mars. Here are the parameters. If you like this image and if you want to generate it on automatic 11.11, here are the details. And then...

Karim Lalani - 8:23 PM
Just to flip things over, the way this project works is, if there is an image already loaded, you can query that image. You can ask it questions and then what it'll do is it'll take this image and it'll pass it to Baklava with the prompt. And

what you're seeing here, you know, I asked, okay, describe this image. This is a response from a language model, which was passed this image and this text. And it came up with...

Karim Lalani - 8:23 PM
It's a 3D rendered image of a man in an astronaut suit riding a motorcycle on a desert planet. And it talks about, you know, a few more details about it. So that's the demo. Let's go quickly into paper space. So when you click, when I click that link, you know, it loaded up that, you know, it'll be something like this.

Karim Lalani - 8:23 PM
It'll clone the repo. So here's the pre-cloned repo. And it'll drop you on screen like this. Now you could go in and you could launch the lab and you could just run this, you know, step at a time. And that's, that's all fine. But see, we already have the code here. So, you know, the code is also present.

Karim Lalani - 8:24 PM
And this is in this folder. So what I ended up doing was there on the left-hand side, there's this terminal tab. I created a bunch of terminals here. On the first terminal, I basically went in and I created a virtual, you know, environment for Python. We, you know, I, you know, we do ship with the requirements TXT with all these versions. This will ensure that

Karim Lalani - 8:24 PM
that if you run this a year from now, a lot of these underlying tools would have changed to different versions. Without this requirements TXT file, the code is 99% likely not to run. But this is giving, you know, telling Python, you know, I only want these specific versions. And in order to sort of, you know, install that, the code...

Karim Lalani - 8:25 PM
the command basically is Python install, you know, we do... not Python install, pip install. This is an incorrect command. We do pip install at the names of the libraries, but what I'm saying is use the requirement, read them from this requirements TXT, and it's basically just feeding everything that I have here to, you know, pip, and it's installing all of those things.

Karim Lalani - 8:25 PM
Okay, we need to download and run Olama, we need to download the models, so we do that next. Basically just copying, so yeah, I just ran those, you know, basically copied this section, ran it in the terminal. Olama serve is a long-running process because it's launching an API, which means once you do this, unless...

Karim Lalani - 8:26 PM
unless you're backgrounding it, you can't, you know, the terminal is blocked, which is fine, I want it that way, because then I know that Olama is running successfully. Once I am, once Olama is running, then I run this in a separate terminal, and it's downloading each of these models from Olama's registry, and each of these, like I said, is a four and a half, five gigabyte model.

Karim Lalani - 8:26 PM
This output is essentially coming from there. After that, I did, you know, the last command is olamalist, which gives me, okay, how many models are you seeing that are local, that have been downloaded locally, and this is a confirmation that yes, I downloaded this, I downloaded this, I downloaded this, okay, good. This is the command to, going down, okay, this is, this is, this is, this is, this is,

Karim Lalani - 8:27 PM
okay. The next bit is to download the installer for automatic 11.11 itself. Again, I open up a new terminal, run this, this will install those scripts, and then, then we install the DreamShopper model itself, which is, again, I think, about six.

Karim Lalani - 8:27 PM
six gigs, if I'm not mistaken, and the, there's, yeah, it's, it's called the sampler, sampler, yeah. So, so run that, once, once it's done downloading.

Karim Lalani - 8:28 PM
Make the web UI that we downloaded, you know, the script that we downloaded a step prior, make it executable and run it. Now, if you're running it locally, you don't need the hyphen F, but because we're running in paper space and I wanted to keep the instructions tight, paper space will launch the container as, you'll drop in a terminal as root.

Karim Lalani - 8:28 PM

And so you need to provide this hyphen F. So, the web UI will not complain about you running it as root and it'll just, okay, you know what you're doing, I'll just run this. And this will take another few seconds to a couple of minutes because now what it's doing is it's installing automatic 11.11 locally on that paper space instance or on the computer that you're running. And once it's installed, it's, you know, we're passing it the path to the, to the...

Karim Lalani - 8:29 PM
7860, but because I didn't change that, I, you know, I'm using our handy-dandy trick of using local tunnel to expose a URL. This step is not needed on paper space because paper space will give you a publicly accessible IP. So, but again, you know, you can still do that because it's easier to copy because what paper space does is it gives you a public IP address. So, you know, you can still do that because it's easier to copy because what paper space

Karim Lalani - 8:29 PM
does is it gives you a publicly accessible IP address. So, you know, paper space will give you, you know, it'll be the output of Streamlit and it'll be the whole HTTP colon forward slash forward slash IP address. And so you'll have to do a little bit of, you know, stripping out the stuff. But once you have done that, that's when you can just, you know, click that link that local tunnel gives you and you will have this interface.

Karim Lalani - 8:30 PM
So, some of the things I'm super impressed with when Kareem has shown me his automatic local number workflows. For those of you that are kind of new to us, in our labs over the past, I guess, six months, in one-on-one labs, in one-on-twos, we introduced the concept of using Streamlit for ST inside of Python, a lightweight Python web interface, to create front ends to our link chain code.

Karim Lalani - 8:30 PM
We've introduced the concepts of, when I say we, we as a user group have introduced the concepts of using local models like Obama and others in conjunction with externals, but also internal on your own, so if you want to run like your AI code, which is not breaching your organization externally, not breaching boundaries, maybe you just want to play with running separate models.

Karim Lalani - 8:31 PM
I think this is really interesting how you built on all those concepts of using your local models, creating functions that are called within your streamlink code, you haven't shown us how we use fine-tunes, but you shared our fine-tunes, as well as integrated your link chain hub, your front proxy link.

Karim Lalani - 8:31 PM
And so, it's really cool to see a local user member out of central Texas, in Austin, put this out here. But also cool to see a lot of the concepts we've gotten over the past six months, kind of starting to congeal in something I think is really super-duper neat. Thanks. Thanks, Colin. So, all of that code, again, is here. You know, it's in the notebook, but there's also this folder which has the

Karim Lalani - 8:32 PM
you know, say, if you want to take this and just deploy. You know, it's all, it's here, and that's what, you know, I mean, it's got the requirements, you know, txt file. Same code that's in the notebook, in those blocks, is here, you know, and then some instructions down here. But the streamlined interface is in the app py, and the land graph agent itself is in graph py, which means that...

Karim Lalani - 8:32 PM
if you want to create application with a different front-end, not, and you don't want, and this is what we were talking about, I think, in our, you know, one of our calls was when we were having that runnables discussion, is this could all have been in part of a single py file, and it would have been fine. But say I want to expose a Lancer application, you know, I want to expose this application as a Lancer FastAPI service.

Karim Lalani - 8:32 PM
Then I would have to strip out, you know, sort of tear out the logic from that and then create another application. Or I could just structure my code in such a way that the reusable portions are self-contained, isolated, and, you know, now I could create a Lancer, you know, application and just include the graph py in it, and it has the agent, everything in it.

Karim Lalani - 8:33 PM
And then the Lancer portion would just expose the right API endpoints, the additional API endpoints that I might need to go with. But just a quick look through this again. There's about less than 300 lines of code here, but some interesting bits. It looks at, you know, automatic 1111 base URL. If you don't specify one, if it's not

Karim Lalani - 8:33 PM
in the environment variables, it'll use the localhost. Same thing for Olama. If you don't specify it, it'll use localhost. Image LLM is the one that's using Baklava model. Text LLM is the Lama model. And the function calling is the fine-tuned one that I mentioned. And this is hosted on Olama AI.

Karim Lalani - 8:34 PM
This is the config object that I mentioned that we're passing, you know, we're basically using this to collect all the parameters that we then pass to the API. Sorry? Is that your state? No, that's not my state, no. This is just a, just so that it's easy to work with.

Karim Lalani - 8:34 PM
Because I need to, all of these are, think of this as a structured request object for image generation, for, you know, package into a single class. This is the text to image input again. This is the field description for that. This is the code we, I shared on the slide.

Karim Lalani - 8:35 PM
Okay, which takes the prompt, the doc string becomes part of the prompt itself. This is what the language model will read and it'll use it to transform the, you know, if it needs to. So, here I have an image generation tool that takes a prompt as a string and returns a JSON response with images encoded as base64 string. The prompt is transformed from simple English to a

Karim Lalani - 8:35 PM
comma-separated mid-journey image generation prompt. And this is where you would, you know, tinker with the prompt if you need to make adjustments to it. And this is where I'm passing it the, you know, I'm filling in the prompt here and then I'm passing that prompt object, the contents of that as JSON to automatic1111. That's all this tool is doing. It's basically a runnable that lets you...

Karim Lalani - 8:35 PM
make API calls to automatic1111. This is, again, just one of the API calls. Automatic1111 exposes a ton. There is an API call that gives you a list of all the models, for example, you know, that you have. So you, you know, you could imagine having another tool that says, okay, that queries, okay, give me all the models that I have. And that gives you the response. And then you can say, you know, you can say that...

Karim Lalani - 8:36 PM
in the prompt, the tool also say that I'm going to pass you a prompt and maybe also the name of the model to use. So, you know, it would then pick up that value and then it would just push it through. Again, same thing here. This is the image to text.

Karim Lalani - 8:36 PM
And then this is where the tool to definition takes the tool definition and it creates a JSON representation of it. This is what gets injected into the prompt itself. So if I go back here, it'll take my tool definition and it'll...

Karim Lalani - 8:37 PM
and then everything after that is basically just, you know, okay so here's the prompt being pulled from LangChain Hub. We're passing in the tool description as a partial there and then we're getting the JSON output from it. The agent...

Karim Lalani - 8:37 PM
state, Ricky, to your question, this is the agent state. Again, it keeps it keeps track of all the messages and it stores one in the most recent image that was either uploaded or generated. It just hangs on to that. And then, you know, this is the everything after that. Again, in the interest of time, I won't go over the code.

Karim Lalani - 8:38 PM
But this is graph.py, this is your LanGraph agent that is powering the streamlet application that I just showed. And with that, I think I have, I'll have, okay, here you go.

Colin McNamara - 8:38 PM
I'll stop the share real quick. Let me share Dan's screen. Dan, do you want to talk to me? Yeah, that would be great. I think that would make sense. Can you hand me that real quick while Dan gets all set up? Thank you. Yeah, thank you so much, Karim. I am consistently invested in every day where I eat.

Karim Lalani - 8:38 PM

Dan, do you want to start sponsoring this? Oh yeah, I did. I didn't think that would make sense. Here, can you hand me that? Well, I think it's all set up. A very impressive presentation. Thank you. Yeah, thank you so much for... Yeah. I am consistently impressed every day where I demonstrate your knowledge, but also...

Colin McNamara - 8:39 PM
You demonstrate your knowledge, but also, can someone kill it, but also how cool you are about it. You're definitely willing to share, which I'm smarter every day, listening to you, watching what you're doing. Can you hand me that? Cool. Are you in the sessions interface? You are? Oh, cool. Yeah. Oh, you just joined. Okay, cool. And then, can you pop this on your thingy?

Colin McNamara - 8:39 PM
We're going to make you a How? Why can't I make you in us? Had the spotlight. Hold on. Remove from spotlight. One moment. Stop spotlight.

Dan Manning - 8:40 PM
Click that right there. Video? Let's do that. Let's take that. Can you share your screen? Is that available to you? A little share screen thing here. Cool. And then... This is going to get you on the projector.

Colin McNamara - 8:40 PM
Video. Let's do that. Let's check that. Can you share your screen? Is that available for you? Cool. And then, this is going to get you on the projector.

Dan Manning - 8:40 PM
I'm going to kill my microphone. I don't think your video is being shared. Share my screen.

Dan Manning - 8:42 PM
Like, why do I spend so much time on the link chain for work? This gives me an opportunity to define agents. It gives me an opportunity to define workflows, scripts, processes. Where things that I would do, that would take time out of my day, if I think back to my life on a hyperscaler, my life, and my team's, and my organization's lives were improved by finding those 5 minutes that we can get back to each other. 5 minutes I can take out of my day …

Dan Manning - 8:43 PM
5 minutes that I could save our support organizations. And so, what's interesting with Dan's code is these agents are interacting with themselves in a shared model. But, if you think about the agents we write for ourselves, as our scripts, as our automations become more autonomous, maybe there's some patterns that we'll see in Dan's video games that we can use in our …

Dan Manning - 8:43 PM
in our lives, in our businesses, in our funds, whether it's managing a social profile or whether it's responding to a support request. I see that Dan is sharing, and I'm going to stop talking and let you talk. Hey. Yeah, thanks again for having me. I'm here today to share a project I'm working on, Pixel Valley. Kind of the point of this talk is I'm just going to briefly do it like a high level. I'm not going to go too deep into anything.

Dan Manning - 8:43 PM
I've got a lot to cover, and basically what this is about is that I am building an application on top of existing stuff. This is all off-the-shelf kind of software, and I'm just kind of pulling things together and building this tool. So who am I? I'm a software architect in Austin for about 15 years, game designer. I'll have some content info at the end if anybody wants to follow me on GitHub.

Dan Manning - 8:44 PM
You can hit me up on LinkedIn. I'm a green banner, so that would be great. So what am I building? I'm building a Pixel Valley. I'm just a software architect. I don't have any machine learning experience. I don't have any data science or anything. I'm just a guy who builds things, right? So what is Pixel Valley? This was laid off in October.

Dan Manning - 8:44 PM
And this is an idea I had. I want to learn all the AI technology. And so what better way to do it than build a little game? So what I'm building is a user can enter a little prompt, and it will generate an entire AI-driven village for the user. So it's like, it'll do all the buildings, all the characters, all the tools, things the building characters can use. It generates using stable diffusion, all the game.

Dan Manning - 8:45 PM

content. And then after it's all created, the player can drop in and interact with the characters. And they're all autonomous. They have their own hopes and dreams and stuff. And they're fun to talk to. So that's a lot. There's a lot of challenges involved there, what I just said. So the first one is procedural generation. How do you generate a village? How do you get...

Dan Manning - 8:45 PM
Game content. If you don't know what the user is going to type in there, you don't know what to create with Stable Diffusion, I think. So how do you do that? There's a problem with the autonomous generative agents. Like, how do you get these agents that work and talk to each other and make them behave in a way that's believable and fun? Which brings me to the next one, which is game design. I'm not going to get into that too much today.

Dan Manning - 8:46 PM
There's some game design problems involved there. It's an AI village. There's not really any goals. Words of fun, right? So there's a lot of challenges there. And then the DevOps and MLOps, which previous speakers have kind of touched on a little bit. I haven't figured that out yet, so I'm not going to talk about that. I'm still having to figure out where to put all this stuff in the cloud. So that's a lot of challenges. I'm just one guy.

Dan Manning - 8:46 PM
Is it possible? And it totally is. So I've been kind of hacking on this. This is kind of what I've come up with. So this is from the prompt Pirate Village. It's generated all these different... First it generated all the types of buildings that would go in there. Created all the artwork in Stable Diffusion. And then this is all stitched together in Python. You can see we got the little character heads that shows who's where, where they're hanging out, what they're doing.

Dan Manning - 8:46 PM
So this was just from the prompt Pirate Village. It's all AI generated. This is an ancient Greek city. As you can see it's got more of a kind of a ancient Greece theme. We've got like some little Greek people kind of running around and stuff. This is the High Elves. This is an experiment I was doing because...

Dan Manning - 8:47 PM
When it generates the characters it gives them all like a name, an age, gender, and I wanted to see if I said hey these are elves and yeah it worked because these are all hundreds of years old is what it decided. They're all extremely old and this is like inside of a building. So this is what it came up with. These are three characters that are in the blacksmith and you can't really tell by looking at this but one of these characters has decided she's going to be the

Dan Manning - 8:47 PM
world's greatest swordsman and to do that she needs to create, to get to that goal, she made a short-term goal of create the world's greatest sword and she has talked to these other two characters and they are all working together to forge a sword. They're in the blacksmith shop talking to each other using tools and stuff. This is, I set up a really quick

Dan Manning - 8:48 PM
chat interface just to make sure everything was working. I put this together in a few hours with Streamlit and Lankchain just to see if it was working and it totally is. So this is I'm talking to a pirate, have you talked to anyone lately? And the summarize with the response with some conversations that she had previous and these are all real answers. She did actually have those conversations. So this worked really cool and I was really impressed with it and again Lankchain and Streamlit

Dan Manning - 8:48 PM
makes this super easy like just if you need to hack something together really quick it's very easy with Streamlit and Lankchain. So let's talk about the first part is the procedural generation. So user types in a prompt, how are you going to get that prompt and generate all this stuff. So here's some of the tools that I'm using. Using Lankchain, using a large language model. I'm using GPT 3.5 turbo. It's fast, cheap, and super easy.

Dan Manning - 8:49 PM
I'm using the AI function calling that we talked about previously, which is supported by GPT 3.5 turbo, so I didn't have to do any fine-tuning or anything. Python, Streamlit. So Lankchain, I'm assuming you've all touched on it, we all know a little bit about it. Super easy to use, flexible, can swap out components and stuff. So this is an example of where I might be using this.

Dan Manning - 8:49 PM

So users type in a pirate village, and I say, you know, system message, expand the description of that scenario. I want, like, a big description. I can't just say it's a pirate village. It's got to be long and fun, right? So, you know, this is kind of what it comes out with, a big, long thing, crystal clear, turquoise waters, you know, that's much better. So we can get a lot more out of this. I put this slide in here.

Dan Manning - 8:49 PM
Recommended reading. These are the two books that I used to learn linkchain. I very much definitely recommend the Greg Lim book was really good. The ML Bear book was okay. I think that he's a machine learning guy, and you can tell I think he used it to create this book. It's a little... It reads like a machine learning. It was written by Chachibiti. Greg Lim is really good. I recommend all of his stuff. I read all of his.

Dan Manning - 8:50 PM
But the ML Bear, it's useful, but it's also like a little Chachibiti-ish. And AI function calling. This is a tool that I use. This is... Yeah, it's just so basically this is letting the LLM call into your own code, and then specifying like an interface for that.

Dan Manning - 8:50 PM
So, it doesn't actually call into your code, it just sends back the JSON of what the call should look like, and all the parameters and stuff that it should pass in. And I'll show you an example. So why should we use this? So here's an example of like, if you're just using like Chachibiti or just like a text LLM, so this is an example somebody showed me of how they're doing it. You're a thief. You must choose your next action. You can be the only one.

Dan Manning - 8:51 PM
One of the following actions, he's got some enumerated values, only output JSON, and nothing else. So he's described what he wants his output to be. And then in order to take a text output and do something with it, it's got to be in some sort of consumable format. And so this is what he got. And that is consumable. You know, it's thinking, it's got the action. So it's what he described. The problem with that is that it's a very...

Dan Manning - 8:51 PM
simple response. And if you want to do something more complicated, like, I don't know how... So you do something like AI function calling. So I've got a scenario, and I need a whole list of buildings. I need an array of them. I need names for them. I need a description. I want to know if it's a building. I'm going to go back in and do rooms. So I can use recursion, actually, in AI function calling.

Dan Manning - 8:52 PM
And it's... So I couldn't even describe how to do this in plain text. I don't know how you would describe this and get a usable answer. So this just gives you, like, a data contract that you can pass to the large language model, and you get something usable back. And, yeah, it gets some great results with this. So, like, here's an example using AI function calling.

Dan Manning - 8:52 PM
You can see it's got, like, the name, it's got the description. And then I went back into stable diffusion and kicked out some assets for that. And then the recursion. You've got to make sure you add a break condition, though, if you're going to do that. I ran into, like, an example where it generated a castle, and I went and got a cup of coffee and came back in, and it's still running. I'm like, what is it thinking so hard about? It had created, like, a fractal labyrinth underneath the castle.

Dan Manning - 8:52 PM
I didn't have a break condition. And it was just, like, all these rooms, and it was just going and going and going. Okay, you've got to stop. It's costing me a fortune, right? So yeah, so the next part of procedural generation, once you've got all that text, I've got a scenario, I've got all the buildings I want, all the rooms I want to build, all the characters and stuff, kick it over to stable diffusion, which is kind of a...

Dan Manning - 8:53 PM
art form of itself. So you've got to find the best model, you've got to, there's things called LORAs, Learn On Reconstruction Intention, it's basically like a Photoshop filter for stable diffusion, where you can kind of fine tune your results a little bit. Prompting, negative prompting, those are super important. And then I put yikes. Stable diffusion is a very...

Dan Manning - 8:53 PM

active community. You've been warned. So first, finding the model. I think you mentioned DreamShaper. This is like super easy to use. Get great results with DreamShaper. This is one that I'm using. It's very good at doing characters. Models are very specialized though. DreamShaper is great at doing characters. It sucks at everything else. You can get awesome characters very easily.

Dan Manning - 8:54 PM
But you'll never get a building out of it. Some of them are easier to use than others. And then you can use Loras to refine results. They have different Loras you can use. Like this one right here. This is from a Lora. I said, you know, I need a theater. And then

Dan Manning - 8:54 PM
it does like this isometric 30-degree thing, kind of, and shapes it. And then that's how I'm able to get like these consistent results with Stable Diffusion. It's like a big win, getting consistent, reusable results out of Stable Diffusion that I can put in the game. It's very difficult. It's very hard. A lot of people are trying to do that right now. Prompting and negative prompting. So basically what I'm doing is...

Dan Manning - 8:55 PM
You know, I've got the scenario. I've got the location name. So I've got this whole prompt set up beforehand. And then just attach the user input at the end over there. So I've got the prompting. But then also the negative prompting is super important. What you don't want your results to look like. So you'll see a lot of things people put like bad quality, low quality, watermarks, signatures. You can put parentheses around things to emphasize.

Dan Manning - 8:55 PM
That's super important. Like I want this to be, all my buildings to be isometric at 30 degrees. So I've got three sets of parentheses around that. And you can see this is the lower right there where it's got the 4 at the end. I'm really leaning on that really hard to get those results. So yeah, so at that point we've got the whole city generated. Got all our characters in there. But now how do we interact with them?

Dan Manning - 8:56 PM
We come to the generative agents. This is based on a famous paper from last year. It leans on a technique called Retrieval Augmented Generation, which is very popular right now. Rerank, which is a technique for getting better results out of your REG pipeline.

Dan Manning - 8:56 PM
I kind of want to touch on Hallucinations a little bit, where it applies to generative agents in video games in particular. Everything that we're building is sitting on the shoulders of giants. These are some of the papers that I've been reading, that I've been building stuff on. The big one was generative agents, which is basically where I got this whole idea from. You can see this is their village right here.

Dan Manning - 8:56 PM
It's kind of pre-built, kind of like an old-school RPG game, and then there's some other cool papers that people have written about generative agents and things like that. So what are we trying to achieve with these agents and these NPCs? So basically we want them to have memory, we want them to remember what we talk to them about, what they do. They want to have memory.

Dan Manning - 8:57 PM
They want to have long-term goals, short-term goals. I want them to use those goals to, in their memories, to do like a daily plan when they wake up in the morning, what are they going to do, decide what to do based on all that. They can use tools, they can talk to each other, they can talk to the player, and they can all work together to do stuff and accomplish all these goals and stuff. And this is basically like the pipeline that we're going to use.

Dan Manning - 8:57 PM
So the agents see something, it gets put in this memory stream, and then when we want to do something, we pull memories out based on what we're trying to do. So if you ask them, like, where do you want to go, it'll say, like, well, I'm trying to make a sword. It'll pull out all the memories related to building a sword and then decide where to go from there.

Dan Manning - 8:58 PM
And that's all done using retrieval-automated generation. So basically the whole memory stream is all stored in, I'm using MongoDB, there's a bunch of them. Here we go. So yeah, it's basically...

Dan Manning - 8:58 PM
Let me see. So it's a technique, instead of fine-tuning your model, it's where you pass all your data into a request as context, and a large language model uses that to make a decision and get a response back. And so how we're going to do that is, first you need like an embedding. So what that is, is that it's taking a piece of text, and it's assigning a huge string of numbers to it. And then you store that in some...

Dan Manning - 8:59 PM
something called a vector storage. It could be like a Weave8 or Pinecone. I'm using MongoDB, they have a vector embedding storage. And then basically from there, when you want to get a query, it'll take your query, it'll convert that with the embedding, get another long string of numbers, it does a piece of math to find pieces of...

Dan Manning - 8:59 PM
...texts that are adjacent to each other, and then that's what it uses to return that, and then you pass that as context to your log-jargon model request. That's like a super high level of how that works. It's actually very pretty easy to do with Langeng. So doing a re-rank, so part of this, part of the rank pipeline is...

Dan Manning - 8:59 PM
making sure that it's passing the best data to the context, if that makes sense. And so it's a difference between a memory versus data. With these agents, it's not just a piece of data that's being sent in, it's a memory, and those have specific... it's different. Memories can be important, they can be recent. And so we have to do some kind of calculation.

Dan Manning - 9:00 PM
Of what we want... when an agent's trying to make a decision, what do we want to use to make the decision? And so this is like a real basic algorithm. It's adding up the recency, the importance, and the relevance, and then using that to use a score to re-rank and bubble up the most important things every time. This is what we're trying to accomplish. So it says, what are you looking forward to most right now? It's going through the memory stream.

Dan Manning - 9:00 PM
Finding the most important... doing that thing, and then it's down here. It's looking forward to a valentine's day. So this is... I thought I'd talk about this hallucination. Everybody says it's super bad. Sometimes it's actually not that bad. So, you know, we're doing video games here. So in LLMs, the control value is the temperature, and it's like a porch too hot.

Dan Manning - 9:01 PM
Porch too cold kind of situation. And is it always bad? I don't think it is. So the porch is too cold. This is if you run your LLM queries all the way, as cold as possible. The upside of that is that you get concise answers only from the context. You don't get very little hallucination from that. The cons for that... I mean, if you're doing something in a business...

Dan Manning - 9:01 PM
That's probably good. But if the cons are, it gives really lifeless answers. You get this as an AI language model. I can't answer that or whatever. It gives up. And then it repeats itself a lot. Kind of a joke there. Also, porch too hot. So that was boring. So I started running my queries all the way to the other end of the temperature. I get some great answers.

Dan Manning - 9:02 PM
But the problems there are, one, it breaks AI function calling. So it's got to be, using that like, depending on if it's valid JSON when it comes back. If you let the ALM go too wild, it won't return valid JSON sometimes anymore. So you have to dial it back. And then also, you get invalid game state. So like, you talk to a pirate and say, like, what are you doing today?

Dan Manning - 9:02 PM
Say, I'm sailing to the island with Captain Blackbeard, and we're going to fight the dragon and get the treasure, and that's going to be our day. And it's like, wow, that sounds awesome, except the only problem is that none of those things exist in this game. That's great, but you just broke the game, you can't do any of that. So you've got to dial it back. I found it's kind of an art of trial and error, you know.

Dan Manning - 9:02 PM
So yeah, so we're just, you know, and then the poor is just right. We're making a video game. This isn't rocket surgery, you know. So if they, you want to run it a little hot, so they stay in character, like if you're talking to a Viking, you want them to talk like a Viking, like not like a large language model. And also it's fun to let them hallucinate a little bit,

because like pirates lie. Like if he says he's the best pirate in the world, it's like, okay, bud, sure you are, you know, it's whatever, like just let him.

Dan Manning - 9:03 PM
Let him lie. So, yeah, there's some game design challenges. You can see these two guys are playing this game very hard. This is definitely not the game. They're not playing Pixel Valley. Like this is not an intense game. It's probably not for those guys. You know, where's the fun in this game? You know, who's the game for?

Dan Manning - 9:03 PM
Who's going to play a game about chatting with pirates and Vikings and stuff? I don't know. I think it's cool. I'm going to try to get some streamers to play it. I think it would be fun to do like a stream of this. Those are all things I'm working on. And then the bar feature goals. So I'm building this like this is a project that I'm building, like a video game. Eventually like doing NPCs as a service, maybe productize.

Dan Manning - 9:04 PM
The generative agent part of it and do like NPCs as a service like some other game developer wants to have. Some agents running around for flavor. They could just do like an API call and drop it in. That's something I'm thinking about, trying to get funded. It's hard. I'm just, you know, this is all bootstrap right now. Text-to-voice. I have some interesting...

Dan Manning - 9:04 PM
Well, there's interesting problems there because, you know, a lot of the text-to-voice stuff right now is you need to have a voice actor, you need to fine-tune your model every time. But for something like this, like we don't. Someone says, okay, it's a pirate. I don't have a pirate voice actor. So, like, how do you do that? I don't know. It's kind of an interesting problem that nobody's really working on, I don't think. I'd like to. I'd like to see somebody...

Dan Manning - 9:05 PM
I've got to figure that out. And then finite state machines. This is a useful question mark. I don't know. This is something cool that I've found that LLM could do, and I couldn't figure out what to do with it. So I was like, I just kind of tore it out. Yeah, and that's pretty much, that's my talk. That was kind of super high level. I don't know if anybody has any questions. I got my contact info there. Yes.

Dan Manning - 9:05 PM
That's a great question. It's not actually stood up anywhere yet. I'm sorry. I'm working on it. Soonish? Probably like next week. I got the GraphQL stood up like this week. So the API stood up. I need to stand at the front end somewhere. I'm working on it.

Dan Manning - 9:05 PM
Yes? How did you find out about this project? Right now it's just on my github. dmanu23, Pixel Valley. When it's live, can you post it to the Discord? Yeah, absolutely. I was going to go in there anyway too. If you go on my github, I have hundreds of repos on there. I'm always just hacking on stuff.

Dan Manning - 9:06 PM
I have like a dozen Lanchain projects on there that I could drop in the Discord, doing different things like a lot of Stablefusion projects, Lanchain and Stablefusion, different chatbots, cover letter creators, resume creators, things like that. I could drop a lot of stuff in there too. You mentioned the word finance fake machines, but I didn't hear in the context, did you say LLN?

Dan Manning - 9:06 PM
So I put together this big AI function call, so basically there's a lot of items in this game, like a coffee pot or something, and I want the characters to be able to use them, so a finite state machine, what it is, is basically like, if you're not familiar with it, it's got states, it's got messages, and then it responds to messages and changes state depending on what state it is and what message it gets, and I wrote

Dan Manning - 9:07 PM
the AI function call where the LLM can actually create those convincingly, so like you'd say convincingly, yeah, so you can ask GPT-3-5 like, I need a coffee pot, and it'll say like it's got like off, on, full of water, brewing, it's just like, just like plain text, I mean, so it basically is like, say it's an AI function call, so it's coming back as like a JSON.

Dan Manning - 9:07 PM
JSON, and then various states are shown in JSON. Yes, yeah, exactly. But not the, not the computer program that would run through it. Well that's, that's the problem too, is that like the LLM, I would say like, okay, you've got this

coffee pot, you know all the states, all the messages that you can do with it, all the transitions that go, brew a cup of coffee, and it can kind of struggle. So

Dan Manning - 9:08 PM
it's, it's there though. I think the code is still up there, I can share that, how to generate a finite state machine with GPT 3.5. It was cool, I just couldn't, I couldn't get it to work, I couldn't get the LLM to properly interact with it, and sometimes it gave goofy results. What language is your game written in? What programming language? Oh, interesting, great question. It's mostly Python, Node.js.

Dan Manning - 9:08 PM
On the front end, let's see, GraphQL, a lot of GraphQL stuff, yeah, Python, most of Python, yeah. Any other questions? Is the goal, so at this point, it's almost like watching an aquarium, right? Yes, yeah. At some point, we allow people to come in and find their own characters, and then you just come in and check in on them, like see how the autonomous agent is doing.

Dan Manning - 9:08 PM
Exactly, yeah. Yeah, look, I could, and like, and sometimes, like, even though it's like kind of, I think I've got it running somewhere. Yeah, yeah. Oh, thank you, thank you. I can almost imagine someone in a black hat doing a version of this.

Dan Manning - 9:09 PM
That's where they're trying to hack each other. Yeah, yeah. It's like a capture-the-flag almost. Just a totally different, like, genre of game, but same framework, same essence. Yeah. So, let me see if it's running somewhere. The demo gods are harsh, so if this blows up, no one's gonna... Yeah, do a demo, lose the sale, right? Yeah. Let me see, I'm gonna need... So here's kind of a cool one.

Dan Manning - 9:09 PM
It should be running. It is not running. This concept of character building is so neat. My brother, who's the main attorney here behind the firm, he held a costume party in New Orleans that was, like, based in 1920s, and he took a while to get the prompting right, like, took all his favorite authors and all this, but now he's got this prompt where he can tell it, like...

Dan Manning - 9:10 PM
You know, here's Pevrim, and a couple of things, like, make him into a 1920s character, but it's just the prose that comes out of it, the description of what you wear, and, like, the linguistics that you would use at the time, it's just so awesome, like, you know, Dan just took this to the video game world. Yeah. Yeah. Uh, so here's, like, a Viking village. Um... Uh, let me see, uh... Does time work? Is it turn-based? Like, does each character...

Dan Manning - 9:10 PM
Does each character get a chance to make a move? Yeah, so basically it's just, uh, time just runs it, there's a microservice, just a Python microservice on the backend, uh, that just kicks off 15-minute increments, uh, and whenever it flips over, uh, it just updates the entire game state. Um, uh, so it's not, like, real-time, it does take quite a while to, like, it's gotta go back and forth to the LLM, all that stuff. But then these things will move.

Dan Manning - 9:11 PM
If you stay here and watch it. Yes. And how do you, like, you know, how do you look at, like, their conversations that are happening, or, like, introspect what's going on? Uh, uh, well, that, I'm still working on. Uh, so, like, here's Ingrid the healer, uh, you can see some information about her, um, uh, you can get, like, a description, this is, like, what it came up with, which was passed to Stable Diffusion to, uh, create

Dan Manning - 9:11 PM
all the character artwork. Um, uh, eventually users will be able to, like, like, come in here and tweak this a little bit if they don't like something about the character. Uh, uh, you can see all the goals that she's come up with. She wants to learn about medicinal plants. Uh, she wants to teach an apprentice, create Woodruff's survival kit, uh, all this stuff, right? And then this is what she's decided to do with her day to, uh,

Dan Manning - 9:12 PM
uh, accomplish all those goals, right? Um, she wants to explore the architundra, uh, teach healing methods, uh, doing all this stuff. Um, Do you ever consider piping in, like, business data and having a project manager fill this role? Or have this one fill the role of a project manager? Yes, I've thought about, like, having, um, having them be like, uh, microservices that you program with plain text. Like, um,

Dan Manning - 9:12 PM
So, spin up an agent and say, you sit here, you watch this end point, when a piece of data comes in, you transform it and kick it out this other end point. Uh, so you don't have to program them, right? And then, uh, if something goes wrong, instead of blowing up your page of duty, they, they come chat with you and they say, like, hey, something's going wrong. I need to go talk to this other agent. Or they can talk to the other guys in the pipeline and figure out what's going on. Yeah. Something like that would be cool.

Dan Manning - 9:12 PM
Uh, yes. Are your agents only characters? Uh. In other words, do you have any other types of agents? Like a master agent? Or did you model time itself as an agent? So that things would move along depending on how that agent would talk about how things should be in the game. And then you'd do a master agent? Yeah. Agent control. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah.

Dan Manning - 9:13 PM
You can nudge characters to do certain things that the plot's getting a little slow or something. Cause floods, drop hurricanes, this kind of thing. You can model the weather itself as an agent. Yeah, okay. Yeah, that's interesting. Yeah. I'm gonna write that down. Or the environment. Animals. Chaos monkey that goes and just...

Dan Manning - 9:13 PM
Model the plot of the story as an agent, so it can have all its own idea of itself. Yeah, like it has a storyline behind it that it's trying to guide you. You may not know it at that time, but you can think about it and keep working on it as it moves. And you can give it access to all the memories of all the characters, so it's a mission. Yeah. Okay. Yeah, I totally get it. Yeah, that'd be awesome.

Dan Manning - 9:14 PM
Okay, I see what's going on. I mean, that's interesting though, because like, yeah, this is kind of, you know, they run around and do stuff, but it's... There's really no goal to the game, you know, so maybe that would be a way to introduce some sort of goal or something. But I showed this one because like, even though it's like there's no goal or anything, there is kind of...

Dan Manning - 9:14 PM
The type of mystery is that all of these agents have decided this is Atlantis and they're all in the Temple of Poseidon reading books. I have no idea why, like, so maybe I could go in there and talk to them, like, why are you guys all in here reading books together, like, what is going on? You're a game designer, right? Yes. This is really interesting, this is like, you could do like, this is like Monte Carlo simulation for any scenario. Yeah.

Dan Manning - 9:15 PM
I think that's fascinating, like, you know, Petra and her brother Nima, you know, this would be really interesting to, like, load up characters with, like, affidavits and you, like, simulate how a court case is going to go based on, like, the jurors' attitudes and, like, this is a really just fascinating way to choreograph, like, strategic actions in, like,

Dan Manning - 9:15 PM
complex environment and just to see all the different ways that it can play out." Yeah, that's a, that's fascinating. Yeah, there's a lot of possibility and there's not a, there's some people working on this, but this is all brand new. Like, the papers were all from last year, so, like, this is all, um, nobody's working on this yet. Nobody's really thinking about it. I mean, people are thinking about it, but, like, there's not a lot of work being done with agents and games, like, just possibly.

Dan Manning - 9:16 PM
There's a lot, there is, the work that I'm seeing being done is all towards traditional games, like, how do we do a, a big, you know, the kids that were sitting there mashing on the controllers, how do we make games with this for those guys, and, like, I don't really know if it's really a good fit, you know. This is more like a storytelling thing, not like an action game, like, you know, LLMs are like a tool in your tool belt.

Dan Manning - 9:16 PM
And when you're making a game, if it's a fast-paced, goal-oriented game, maybe it's not a tool you control very often. Or something like this, like, like a Pride and Prejudice LARP, this is, they do this in Austin, I didn't, something I knew, found out about, and so I thought, maybe this would be something fun for those kind of people, like, you can come here, and, I think I typed in the Jane Austen village, and this is what I came up with, it's all the Pride and Prejudice characters doing stuff, talking to each other.

Dan Manning - 9:16 PM
If I was a big Pride and Prejudice fan, I could come in here, and, you know, set them up against each other, and, you know, mess with them, kind of fun. Wow. It would be fun to run, like, classic fairytales through this, and you can, like, check in after the half-live, or after. Yeah. Whatever happened with Cinderella and the Prince? It was a chance to force after seven years.

Dan Manning - 9:17 PM
Cool. Are you taking notes? Yes. Let the characters run for 10,000 years, and see where their societies develop. Yeah. Well, I mean, I think at that point, like, my MongoDB bill would, like, explode. Awesome. Well, at this point, I want...

Dan Manning - 9:17 PM
I want to go ahead and, of course, we can all continue chatting afterwards. Perfectly fine. I do want to be respectful of everyone's needs to get to bed, and what not. Intermine people. This is an ongoing thing. Our point here is to connect people, to inspire each other, to improve our skills in what we're doing, and to enjoy this really cool project. Dan, thank you so much. Hopefully, we'll see you next time.