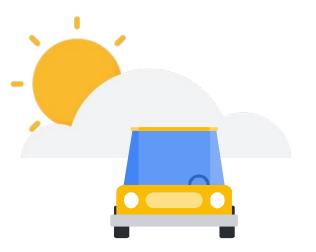
Carolyn Huynh | Google Cloud Security

# Self Driving Cloud

Troubleshooter



careOhlyn.github.com

### Carolyn Huynh | Google Cloud Security



### Lead UX Designer

### Role

I was the first designer to join the Self Driving Cloud team and ever since, I've been the Lead UX Designer and thought leader behind the suite of SDC products three years now.

Troubleshooter is the second product within the self-driving cloud initiative that I have lead from inception to launch.

# **The Problem:**

Troubleshooting 403 error messages is hard. It's done through lots of trial & error.

## Admins troubleshoot by giving more permissions, which makes it hard to maintain least privilege.

### What are we building?

Troubleshooter is the second product launch within the Self Driving Cloud suite of products. This product is meant to help over granting permissions by helping admins debug denial error messages.

Built on the IAM Recommender API, it was a sophisticated backend that needed to be unpacked simply in the UI.



### **Example User Scenario**

Anita is a developer in your organization and gets an error message informing her that she *does not* have the relevant permissions when she tries to delete images in a bucket.

Anita goes to Mike, her security admin for help, who has to manually check the policy himself, and will end up expediting the process by just granting Anita a bunch of unnecessary roles.



### Translating the API to GUI

The engineers had approached me with a predetermined sketch in mind.

After going through many rounds of the engineers' design solution, I was convinced that their way was not the way to go. Through lots of research, I changed product direction by designing a different way to display IAM bindings by showing raw json policy so that advanced users would be able to click through bindings via a tree hierarchy.

I was able to change product direction with multiple stakeholders with design thinking and creating a more visual tool for developers.

## ...FYI, here was the engineer's idea

	Project ▼ Q.	5 🏚 🛛 🗊 i 🌘
IAM & admin	Troubleshooting	
IAM	Check on anita@google.com's bucket.update permission on bucket/goo1	
Troubleshooting		
Organization policies	A You don't have permission to check the role and one of the groups here.	
Service accounts	- Filter table	× C III
Labels	Resource ↑ Role check ↑ Principal check ↑ Conditions	
GCP Privacy & Security	▶ Ba Foogle	1
Settings	► folder/foo2	:
	folder/foo2	:
Roles	▶ 🐤 project/bar2 🔒	:
	bucket/goo1  Check	÷
	🕒 🖶 bucket/goo1 🕕 No check 🤡 Check	÷

## ...another example.

Google Cloud Platform	🐉 Project ,	Q					2		9 Ø	)
IAM & admin	Helpdesk	VIEW TICKET								
IAM	Week of Februa	ary 5, 2018 - Februa	ry 11, 2018							
Helpdesk Organization policies	Total events tl 5	his week	Resolved 3	Pending 2		Recommendations				
Service accounts		vents for the past 7 day	T. and	2						
Labels GCP Privacy & Security	Filter log		<b>y</b> 5					I		
Settings	Status	Date 个	User ↑	Email address	Time	Permission	Request Attributes			
Roles	0	Feb 3, 2018	Anita Roy	anita@company.com	9:50 PM	Bucket.write	IP Address Time of day		:	
		Feb 3, 2018	Carolyn Huynh	carolyn@company.com	8:00 PM	Bucket.stuff	Time of day		:	
		Feb 2, 2018	Sarah W.	sarah@company.com	11:00 PM	Foo.bar	Another reason		:	
		Feb 1, 2018	Vandy Ramadurai	vandyr@company.com	4:00 AM	Other.stuff	Foo bar reason		:	

### ...annnd one more.

IAM & admin	Helpdesk	VIEW TICKET		× Ticket #000							
IAM	Week of Februa	ary 5, 2018 - February	11, 2018	View policy heirarchy	Export all p	olicies 🏦	Deny incident				
Helpdesk			_	Org name (Google	)	1 ~	User	anita@company.com			
	Total events th	is week	Resolved				Permission	Bucket.write			
Organization policies	5		3	Member/Group	Storage Admin Group		Status	Denied			
Service accounts				Permissions	Bucket Manager - custom role		Time	9:50PM			
	Log of all deny ev	ents for the past 7 days		Conditions	none		Permission	S2			
Labels				115			Outcome of check	Denied		Request Attributes	IP Address
GCP Privacy & Security	= pending	8						Time of day			
Settings	Status	Date ↑	User ↑				GRANT ACCESS				
Roles	<b>0</b>	Feb 3, 2018	Anita Roy	Folder-A Name		/ ^					
				Member/Group	Anita Roy						
		Feb 3, 2018	Carolyn Huy	Permissions	Folder Admin Role						
				Conditions	SFO Office IP						
				Outcome of check	Denied						
				Project Name		<i>i</i> ~					

No matter how many versions of tables I designed, the problem was that developers still couldn't understand and debug readable text.

## ...jk, last one.

Google Cloud Platform	Q					
IAM & admin	$\leftarrow$ Troubleshooting					
IAM	Check on anita@googl	e.com's bucket.c	delete permission on b	ucket/goo		
Organization policies Service accounts	Checker status Conditional					
Labels	- Filter table					× C III
GCP Privacy & Security	Checker 🛧	Resource 个	Role	Principal	Conditions	
Settings	<ul> <li>Denied</li> </ul>	Foogle.com				:
les			🤣 Owner	😑 group/admin		
			🤣 Owner	group/boss		:
			🧭 Editor	group/exec		:
	<ul> <li>Conditional</li> </ul>	folder/dev	📀 Owner	group/admin		i
			Security	group/watchers		
			Editor	🥑 group/dev-eng	From 9AM - 5PM	
	🝷 😑 Denied	folder/dev-SF0				1
			🥑 Owner	😑 group/sfo-admin		
			Security reviewer	group/sfo-watchers		
	T A Donied	- project/demo				

```
"bindings": [
   "members": [
     "user:jie@example.com"
   ],
   "role": "roles/resourcemanager.organizationAdmin"
 },
   "members": [
      "user:divya@example.com",
     "user:jie@example.com"
   ],
   "role": "roles/resourcemanager.projectCreator"
"etag": "BwUjMhCsNvY=",
"version": 1
```

### Research

Research had shown that the original idea behind the engineer's mocks for the product had one consistent theme:

Nobody could tell that each row in the table represented a binding within an IAM policy.

So, I literally went through the API and the subsequent raw json policy and began scrubbing it, in an attempt to find UI components that would expliciting show what each section of the policy meant. I decided to be as <mark>explicit as possible</mark> and expose raw json policy, just like how developers see it today.

I took a look at some of the most competitive products on the market and <mark>understood a</mark> rising trend in policy-as-code. I decided to mimic the policy-as-code trend as best as I could, while understanding the constraints of the API.

We couldn't provide a way to version control policy, but we **could** provide a way for customers to integrate with **3rd party tools** that will allow them to.



## A preview of the end solution I designed.

	orm ۹		<b>2</b> • 0 0 : (
IAM & admin	← 11 bindings are affecting buc	ket/goo throughout the resource hierarchy	
IAM	Bindings by resource	Access granted if condition is true for API call for anita@foogle.com, bucket.delete, bucket/goo.	CONTACT SUPPORT
Troubleshooter	Foogle's policy (3)	For further information and assistance, contact your support team.	
<ul> <li>Organization policies</li> <li>Service accounts</li> <li>Labels</li> <li>GCP Privacy &amp; Security</li> </ul>		anita@foogle.com is not in group by helet delete	
Settings Roles	temo's policy (2)     goo's policy (1)  EXPAND ALL BINDINGS	OUCKEY: Call         10         "role': "role'security",           permission is not in role         11         "members": [           anita@foogle.com is not in group         12         "group:admin@foogle.com"           anita@foogle.com is not in group         13         ]           16         *cole": "roles/editor", 17         "members": [           18         "group:deveng@foogle.com",	
		<pre>19 ], There is a condition in this binding 21 condition": { 21 condition ": { 22 expression: 'resource.type == "iam.roles" 23 &amp;&amp; request.time &gt; timestamp("%s2")' 24 } 25 } 26 ] 27 "etag": "123457" 28 }</pre>	



### **Research 2.0**

My researcher ran with my idea of exposing raw json (something that hadn't been done before within GCP), and we ran lots of customer sessions, often times changing up the design on the spot.

The PM of Troubleshooter was impressed by the amount of praise the new UI was getting from customers, and began to take the idea of exposing the raw policy and reusing the pattern in other areas of self driving cloud.



# Expert interface. Looks just like an API response.

Customer on GCP Name and Company redacted

(I swear they said this).

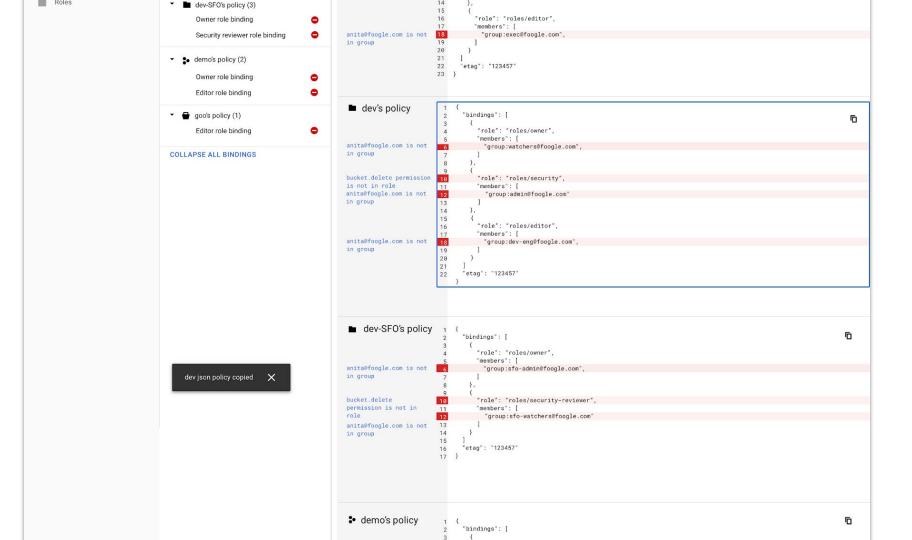


# Now, let's walk through the whole product from start to finish.

≡	Google Cloud Platform		۹			5 🖡 0 🗩 E 🧕
	IAM & admin	Troubleshooter				
	IAM	IAM Troubleshooter	r assesses if a specific API call	will grant the identity	access to a resource.	
	Troubleshooting	Troubleshooter 💿				
	Organization policies	If you have data acces	ss logs turned on, you can retrieve th	em from Stackdriver		
	Service accounts	Principal		0		
	Labels	Enter email address su	uch as user@company.com			
	GCP Privacy & Security	Permission		0		
	Settings	Enter the permission s	such as "storage.buckets.getlamPolicy			
	Roles	Resource		0		
		Enter the name of the	resource on the API call			
			CLEAR	CHECK API CALL		

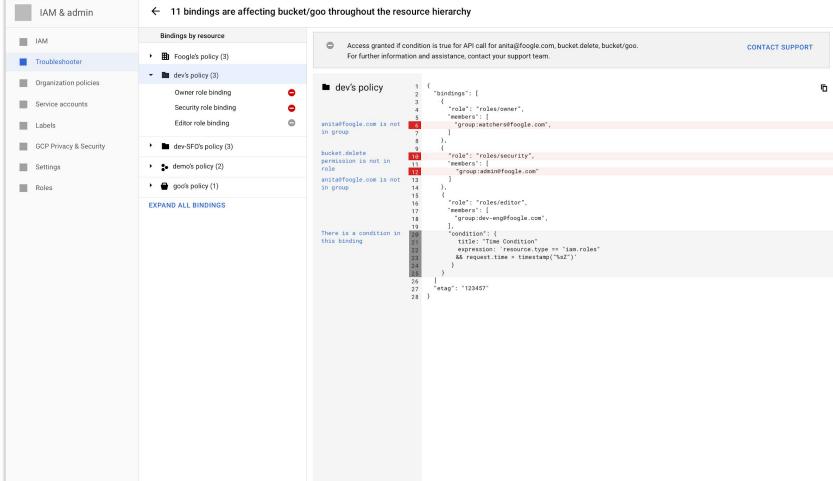
		2 🖡 0 🛿 : 🧕
IAM & admin	Troubleshooter	
IAM	IAM Troubleshooter assesses if a specific API call will grant the identity access to a resource.	
Troubleshooting	Troubleshooter ⊘	
Organization policies	If you have data access logs turned on, you can retrieve them from <u>Stackdriver</u>	
Service accounts	Principal anita@foogle.com ⑦	
Labels	Enter email address such as user@company.com	
GCP Privacy & Security	Permission bucket.delete	
Settings	Enter the permission such as "storage.buckets.getlamPolicy"	
Roles	Resource bucket/goo ⑦	
	Enter the name of the resource on the API call	
	CLEAR CHECK API CALL	

	m Q		2 🜲 0 D i 🌘
IAM & admin	← 11 policy bindings are affect	ing bucket/goo throughout the resource hierarchy	
IAM	Bindings by resource	<ul> <li>Access denied for API call for anita@foogle.com, bucket.delete, bucket/goo.</li> <li>For further information and assistance, contact your support team.</li> </ul>	CONTACT SUPPORT
Troubleshooter Organization policies	Owner role binding Viewer role binding	Foogle's policy 1 ( "bindings": [	
Service accounts Labels GCP Privacy & Security	✓ ■ dev's policy (3)     Owner role binding	anita@foogle.com is not in group	
Settings Roles	<ul> <li>dev-SFO's policy (3)</li> <li>Owner role binding</li> </ul>	<pre>11 "members": [ anita@foogle.com is not 12 "group:boss@foogle.com" 13 ] 14 }, 15 { 16 "role": "roles/editor", 17 "members": [ anita@foogle.com is not 18 "group:exc@foogle.com",</pre>	
	demo's policy (2)     Owner role binding	in group 19 ] 20 } 21 ] 22 "etag": "123457" 23 }	
	<ul> <li>➡ goo's policy (1)</li> <li>Editor role binding</li> </ul>	<pre>dev's policy 1 {</pre>	
	COLLAPSE ALL BINDINGS	<pre>in group 7 ] 8 } 9 { bucket.delete permission 10 "role": "roles/security", is not in role anit@foogle.com is not 12 "group:admin@foogle.com" 13 ] 14 }, 15 { 16 "role": "roles/editor", 17 "members": [</pre>	
		anita@foogle.com is not 12 "group:dev-eng@foogle.com", in group 19 ] 20 } 21 ] 22 *etag": "123457" }	



≡	Google Cloud Platform	ı	

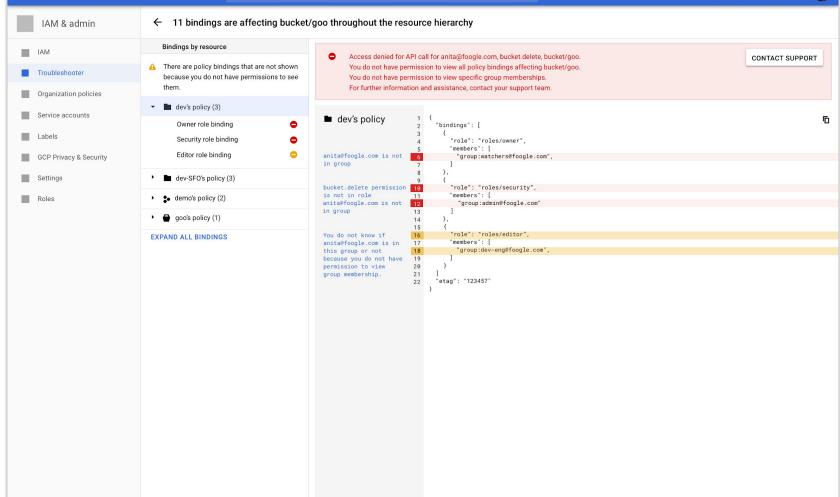
#### 🗵 🌲 🛛 와 E 🗕



#### ≡ Google Cloud Platform

#### Q

### 🛛 🔈 😰 😰 E 🌘



# Debuted on stage at Google Cloud NEXT.

# One of the top most used products within GCP, making close to a [redacted #] API calls a month.

# **Key Takeaways & Reflections**

As the Lead Designer (and only designer) to ship a second self driving cloud product...

0 0

### 1

### Accessibility

Since this is highly visual tool that really (and literally) highlights areas that helps developers debug 403 messages, in a mad rush to ship, I had forgotten the most *foundational* part of shipping any visual tool: accessibility.

I hadn't taken into account those were were colorblind.



2

### Accessibility

I knew my mistake immediately after we shipped.

Unlike role recommender which had the traditional +/- pattern, (which by the way, is the standard way for those who are colorblind to go through a code diff), Troubleshooter didn't have that pattern.



### Accessibility

I knew that icons were needed, along with a hidden tag within each icon that would describe what each denial message meant.

I deeply regretted this error, and as someone who prides myself n being an inclusive designer, I knew I had excluded a large portion of users.. Though I talk a lot about inclusivity in design, I hadn't even held myself accountable, and had prioritized shipping > doing right by the user. It ended up creating tech debt to fix the problem.

I will do much, much better next time.

As we all should.



# **Thank You**