Docker for Data Scientists

Jacqueline Nolis
Chief Product Officer

(disclaimer: I am not a professional engineer)

My dream as a young data scientist:

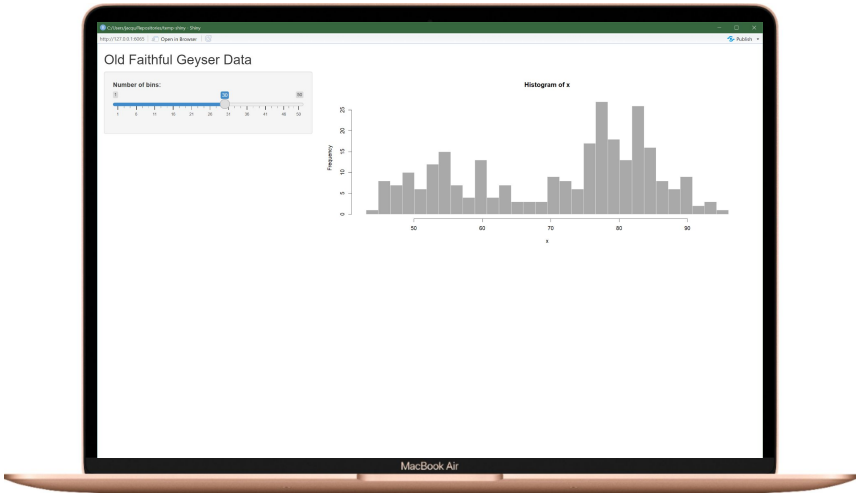
That I could write code and deploy it for other people to use

THIS HAS BEEN HARDER THAN I WOULD HAVE EXPECTED

(Spoiler: Docker makes this easier)

You have wrote some data science code on your work machine that you want to run somewhere else.

- **An API**
  - Python: Flask, Django, FastAPI, ...
  - R: Plumber
  - Julia: Genia
- **A dashboard**
  - Python: Streamlit, Dash, Voila, ...
  - R: Shiny
  - Julia: Dash
- **An arbitrary script** that runs on a schedule



Could be publicly for everyone, internally within your company, as part of your companies product, or somewhere else

SaturnCloud

Making your code run somewhere else

Want to run it on a server

Servers are just computers that are running in a centralized location.

If you choose they can have these benefits:

- Always on
- Restart gracefully
- Are managed by someone else
- Can have more memory, CPUs
- Can spin up a lot of them if you have high traffic

They an also literally just be a computer somewhere else

TECHNICALLY A SERVER

SaturnCloud

Servers host virtual machines

A virtual machine is an emulation of a computer.

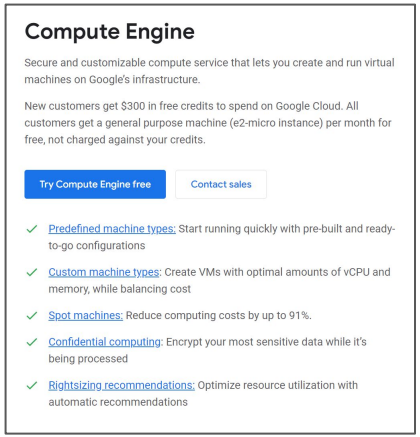
Don’t buy hardware for every machine you need, just buy one big machine and run lots of pretend machines on it.

You can run virtual machines on laptops too!

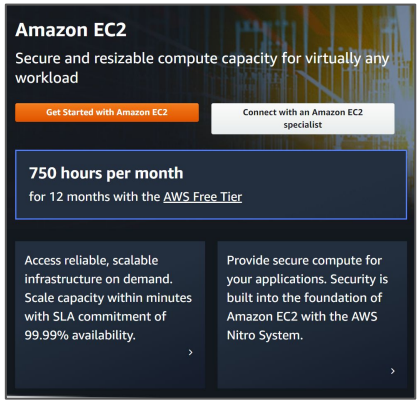
SaturnCloud

Get your code on a server V1: manually set up a virtual machine

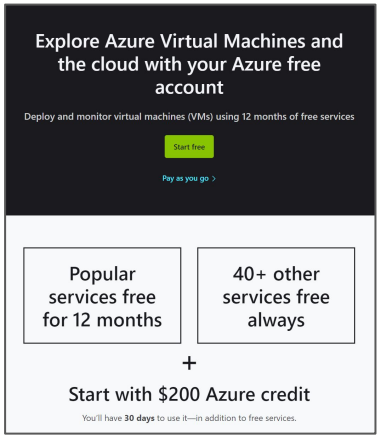
GOOGLE CLOUD PLATFORM
VMs



AMAZON WEB SERVICES
EC2



AZURE
VMs

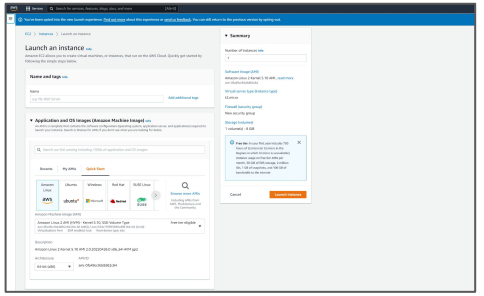


All cloud providers let you rent a virtual machine by the hour!

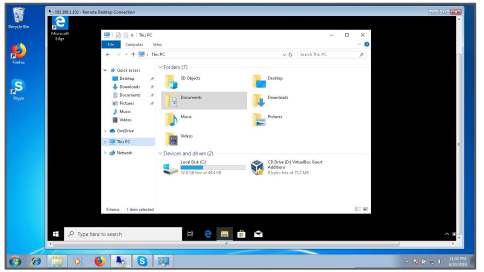
SaturnCloud

V1: set up a virtual machine

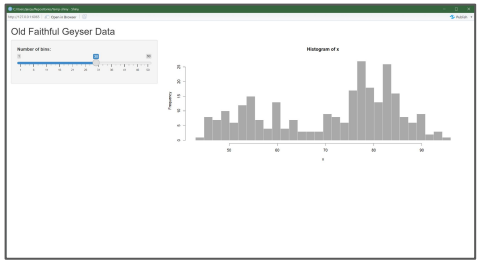
1. **Get a server** on AWS/GCP/Azure
   a. Select the amount of memory/CPUs/disk
   b. Select the operating system
   c. Select who can connect to it
2. **Start the server**
3. **Remotely connect** to the server (ssh or remote desktop)
4. **Install**
   a. Required OS libraries
   b. R/Python/Julia
   c. Packages (Flask, Shiny, whatever)
5. **Start your code**
   a. Run the API/Dashboard
   b. Set the schedule for the script



SETTING UP AN AWS EC2 INSTANCE



VIRTUALLY CONNECTING AND INSTALLING PACKAGES



VM THEN HOSTS THE CODE

SaturnCloud

Setting up a virtual machine (V1): Pros and cons

Pros:

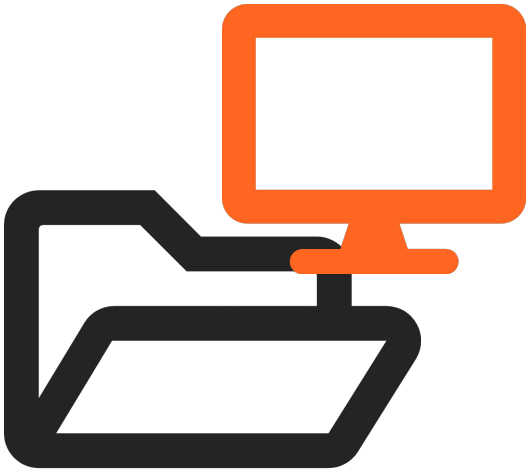
- Your code is now running somewhere else.
- The goal is achieved; the problem is solved.
- This feels like magic.

Cons:

- You have to reinstall everything on a new machine each time you set one up.
- Messing up one step can ruin everything.
- If your VM is destroyed you have to do everything again.

Get your code on a server V2: VM Snapshots

- “Virtual” Machines are still programs themselves, what if we made a copy of that program?
- Now if the VM is destroyed we can just load up the snapshot



(This is where computers become programs, programs become data, and the boundaries of identity become meaningless)

SaturnCloud

Snapshots (V2), pros and cons:

Pros:

- You no longer have to worry about a virtual machine being destroyed
- You can run duplicates of the virtual machine

Cons:

- You have no record of the steps you took to set up the virtual machine.
- Snapshots are really big since they capture the entire virtual machine.
- You can’t undo steps you did in machine the virtual machine.
- VMs that are nearly the same don’t have any benefits

SaturnCloud

Getting your code on a server v3

What if we had a way of:

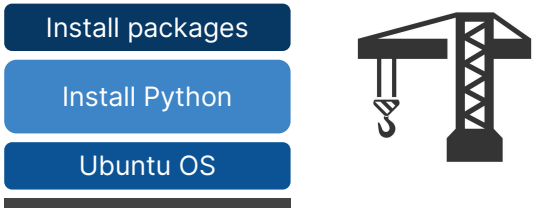
- Keeping track of each step used to set up a server
- Taking a snapshot after each step and storing them separately
  - So that if two applications shared mostly the same setup steps you didn't have to store it all twice
- Running snapshots extremely efficiently

Containers!!!

SaturnCloud

Containers: basic idea

- A Docker image is a snapshot that you can run
  - Images are built on top of other images
- A Dockerfile specifies how to build the image
  - The program Docker lets you build the images
  - The files specify the command to run when the image starts
- A Docker container is when you run the image
  - Docker (the program) can run containers
  - Kubernetes is another (often better) platform for running Docker containers
- Docker can refer to
  - This whole framework (but people are moving away from saying Docker)
  - The specific program you can use to build images and run containers



BUILD A NEW IMAGE BY TAKING ANOTHER IMAGE AND ADDING LAYERS TO IT

RUN THE IMAGE ALONGSIDE OTHERS USING TOOLS LIKE DOCKER OR KUBERNETES

SaturnCloud

Example Dockerfile

This is an example of a Dockerfile. Each Docker command adds new layer, meaning it will create a new image on top of the previous one.
You can build the image from the Dockerfile using `docker build -t image-name .`

```
FROM ubuntu:20.04

RUN apt-get update \
    && apt-get install -y --no-install-recommends wget \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/

RUN wget
"https://repo.anaconda.com/miniconda/Miniconda3-py39_4.11.0-Linux-x86_64.sh"&& \
    bash Miniconda3-py39_4.11.0-Linux-x86_64.sh -b

RUN pip install tensorflow

ENV PYTHON=3.9

COPY my_python_script.py /

ENTRYPOINT ["python", "my_python_script.py"]
```

FROM indicates what the starting image is. All images start on top of another image. Here we are using an empty Ubuntu image. You can use other people's images as your starting point!

RUN indicates a Linux command to run. Here we are installing wget to download a file. Many common programs don't come installed in images to keep them lightweight, so you have to install yourself.

Next we download and install miniconda to get Python. Because this is a separate RUN command, it will make a new intermediate image.

ENV specifies an environment variable that should exist both when building the image and running it as a container. **ARG** is used for variables only needed during building.

COPY will copy a file or folder from the location of the Dockerfile into the Docker image.

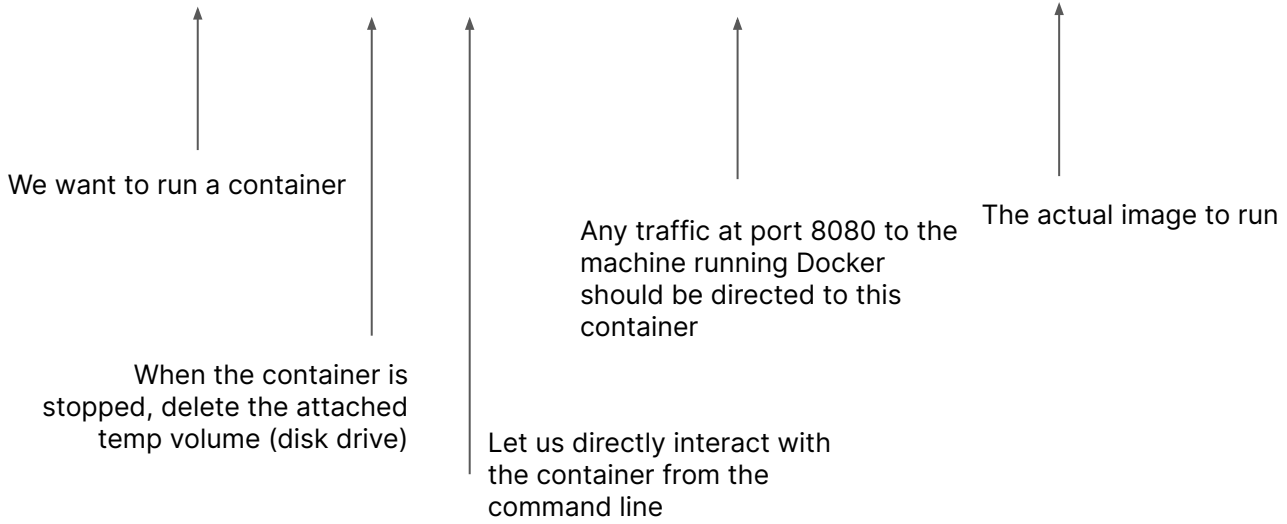
ENTRYPOINT specifies the program that should run when the container starts. **CMD** is a similar command to specify the same thing. You can specify a script to run on start an interactive shell for your language.

SaturnCloud

Running a container

Once you have built an image, you can run it with a command like

```
docker run --rm -it -p 80:80 image-name
```



SaturnCloud

Live demo!

Aside: data science use cases

FOR PRODUCTION

1. Take your code and package it
2. Hand it to engineers to deploy

(what we've been talking about)

FOR ANALYSES

1. Take your code and package it
2. It is now fully reproducible by other data scientists

(also a totally valid use case)

SaturnCloud

More container concepts

Debugging containers

- If you have a running container you can also directly enter it
  - `docker exec -it image-id /bin/bash`
  - This can be useful to figure out why a container isn't doing what you expect
  - This does not help you if your image doesn't start
- You can also run intermediary images made during the build process
- Use `-d` to run the image in detached mode in the background (make sure you stop it!)

[Another live demo]

Storing images

- Your computer will store the images you build
- This includes intermediary images
- You can also send your images to a registry
  - Dockerhub - popular service for hosting public images
  - AWS ECR, GCP Artifact Registry, Azure Container Registry - enterprise cloud image stores you can use
- Use `docker push` to push an image to a registry

```
jnolissaturncloud/docker-demo:2022.06.01
```

Org or user who owns the image

Image name

Tag (the version of the image). Use `latest` for most recent

SaturnCloud

Building from other images

- There are lots of publicly available images, including
  - With just an OS
  - With an OS super slimmed down
  - With R/Python/Julia installed
  - With lots of data science packages installed
- If you can use someone else's image to start you can save lots of time

Image size

- The bigger the image, the harder it is to use
  - Longer to download
  - Longer to start
- The more you can remove from an image, the better
  - <1GB - fast!
  - <5GB - maybe okay
  - >5GB - danger zone
- If you use another person's image as a base, it might be bigger than you need

Attaching a volume

- Docker containers do not inherently have attached drives
- Stopping a container deletes everything saved on it
  - Often super helpful! Sometimes disastrous!
- You can attach a local drive to the container with `--mount`

SaturnCloud

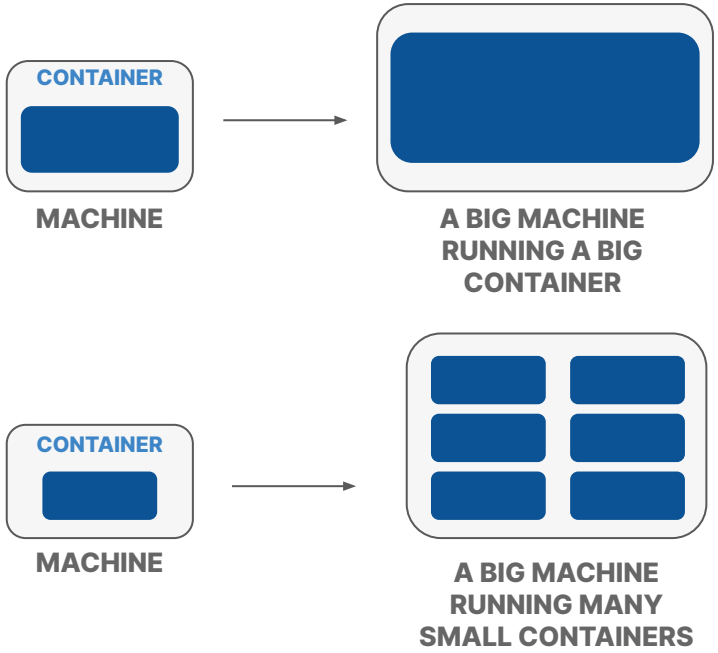
Operating systems

- Docker images originally only supported Linux distributions
- Microsoft and Docker spent lots of effort to make Windows images
  - Fewer Windows programs work from the command line
  - Somewhat less stable
- There aren't really macOS Docker images

Image	Computer running Docker	Details
Linux	Linux	Just works
Linux	Windows	Use Docker Desktop or Docker on Windows WSL2
Linux	macOS	Use Docker Desktop
Windows	Windows	Use Docker on Windows Server or Docker Desktop
Windows	Linux	Cannot be done
Windows	macOS	Cannot be done (cursed)

Scaling

- Need more CPU/RAM? Run the container on a larger machine.
- Need to handle lots of requests or parallelize your code? Run many of the same containers at once.
- Is one physical server not enough scale? ...



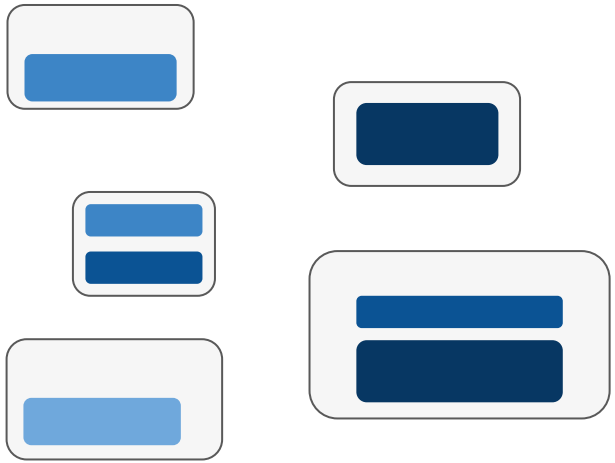
SaturnCloud

Running containers at scale

Running containers

- Running containers on a single machine still has same problems as using a laptop
- What if we had a way of orchestrating many computers at once to run containers?
  - Hello Kubernetes
- Kubernetes can
  - Automatically scale up/down hardware running containers
  - Load balance traffic
  - Gracefully deploy new versions of images
  - Handle security (https)
  - Run scheduled containers
  - So much more!

kubernetes



MANY MACHINES RUNNING MANY CONTAINERS

SaturnCloud

Options for using Kubernetes

- Deploy your own Kubernetes cluster on your cloud service
  - Waaaay out of my pay grade
- Use a cloud service provider's turnkey Kubernetes deployment
  - Amazon Elastic Kubernetes Service (EKS)
  - Waaaay out of my pay grade
- Use a cloud service provider solution that entirely abstracts Kubernetes away
  - Cloud Run on Google Cloud Platform
  - Actually pretty cool I'll demo that
- Use a data science specific tool which entirely abstracts Kubernetes
  - Saturn Cloud
  - Actually pretty cool I'll demo that

Cloud Run (on Google Cloud Platform)

You give Cloud Run a Docker image and

- Tell it how much memory you need
- Tell it the max number of concurrent instances allows

You get

- An endpoint!

It will start the container when someone calls your code

- Great and cheap for deploying stateless production code (ex: model serving)
- Not good for long running complex tasks (data pipelines)

SaturnCloud

Saturn Cloud

Saturn Cloud is a platform for data scientists

- Do ad hoc analyses & train models with JupyterLab, RStudio, or SSH
- Run code on a schedule (jobs)
- Deploy dashboards and APIs (deployments)

All Saturn Cloud resources start with Docker images

- Use our premade ones
- Upload your own

SaturnCloud

Takeaways

Docker images are a way to create environments for running programs

- Images capture the state of a machine
- Images are built on other images
- Build an image with `docker build -t image-name .`

Dockerfiles give the blueprints for images

- Dockerfile in a folder describes how to build
- Some basic commands
  - `RUN` - execute a Linux command
  - `COPY` - copy a file
  - `ENTRYPOINT` - what starts when the container starts

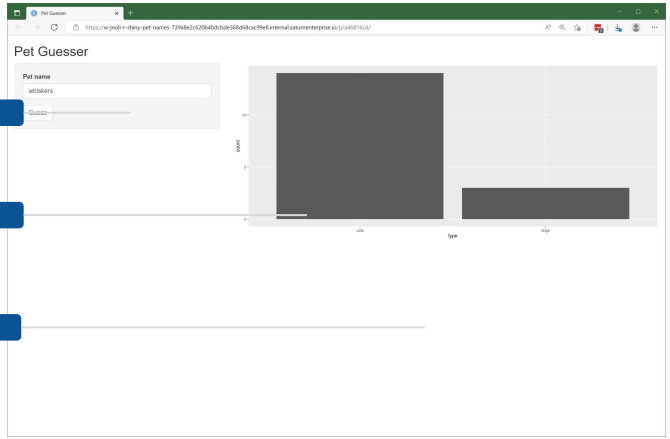
Containers

- When an image is run that’s a container
- A container can be scaled on different hardware
- Stopping a container loses what’s on it
- Run a container with `docker run -it image-name`

Managing containers

- Docker can run containers for you
- To run many containers across multiple machines at scale, people use Kubernetes
- Kubernetes can load balance, autoscale, and more
- Not easy to manage yourself
- Solutions like Google Cloud Run (or Saturn Cloud) let you use containers more easily


SaturnCloud



Boring

Sad

Soooo *default*

BEFORE

Favicons

Images

Custom CSS

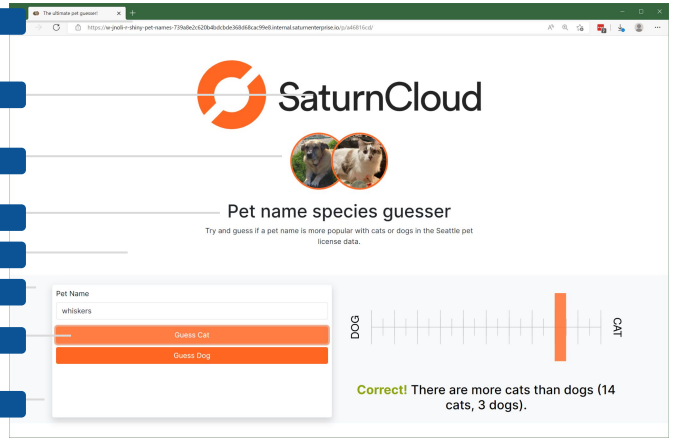
Google & Other Fonts

Mobile layouts

Backgrounds

Theme colors

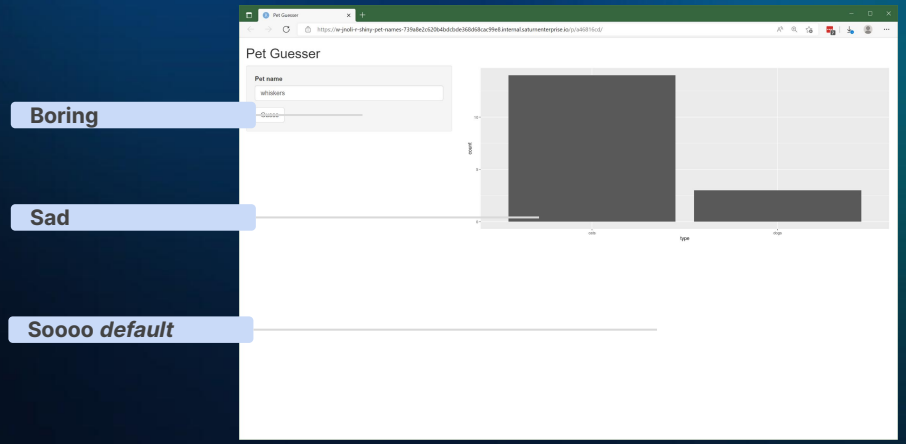
Shadows



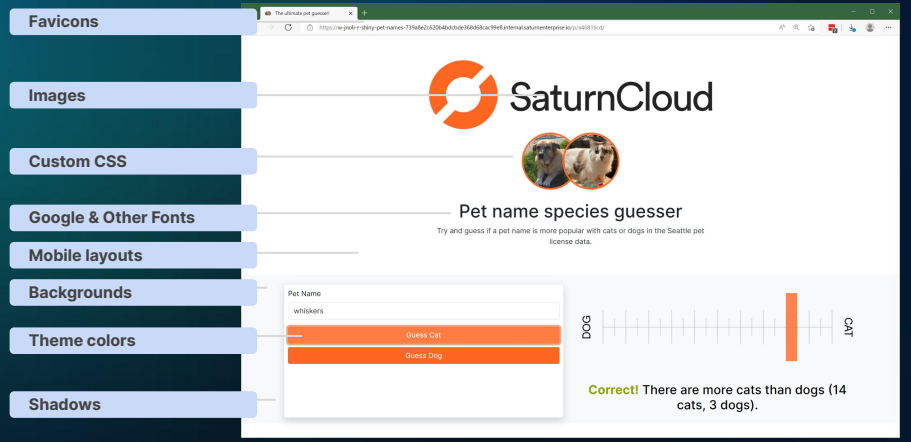
AFTER

SaturnCloud

Upcoming webinar: Styling R Shiny Apps
June 22nd @ 2PM ET
https://scld.io/workshop/shiny



BEFORE



AFTER

Other upcoming webinars:

Introduction to Julia
Getting Started with Saturn Cloud

Thank you!

Jacqueline Nolis
Chief Product Officer
jacqueline@saturncloud.io