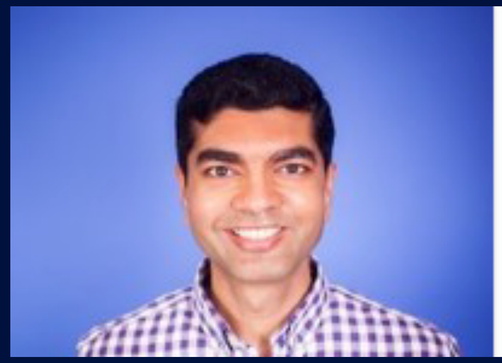
# TRAINING DEEP LEARNING MODELS AT SCALE USING KUBERNETES

**Mitul Tiwari and Deepak Bobbarjung**

Passage AI

Passage AI

# Introductions



## Mitul Tiwari

CTO and Cofounder at Passage AI

San Francisco Bay Area | Computer Software

| | |
|---|---|
| Current | Passage AI, Forbes Technology Council |
| Previous | LinkedIn, Kosmix, Google |
| Education | The University of Texas at Austin |

## Deepak Bobbarjung

Founding Engineer at Passage AI

San Francisco Bay Area

Information Technology and Services

| | |
|---|---|
| Current | Passage AI |
| Previous | EMC, Maginatics, Inc (Acquired by EMC), VMware |
| Education | Purdue University |

# Outline

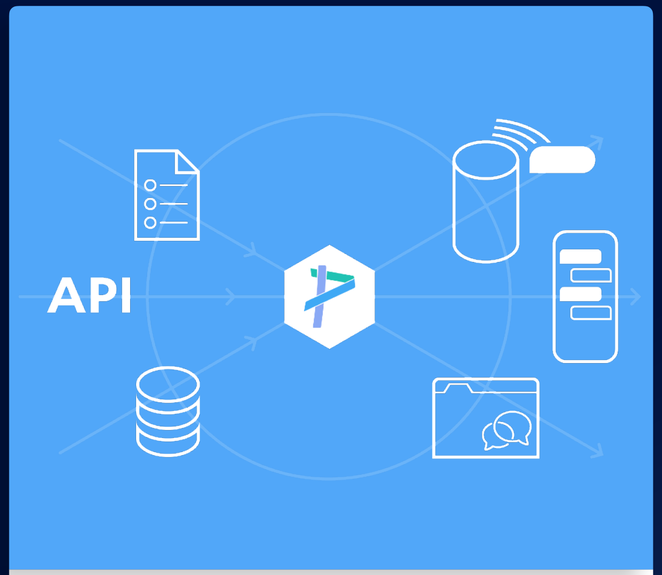
Conversational AI and Deep Learning

Need for a Jobs framework on Kubernetes

Our Jobs architecture

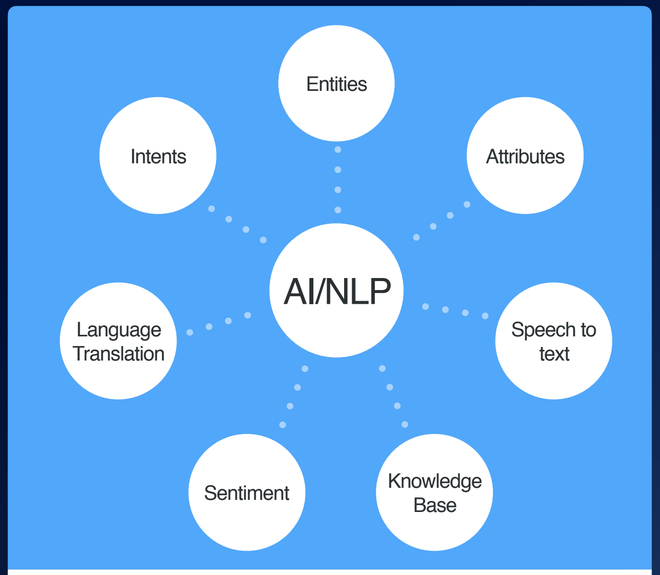
Passage AI

# Our Conversational AI Platform



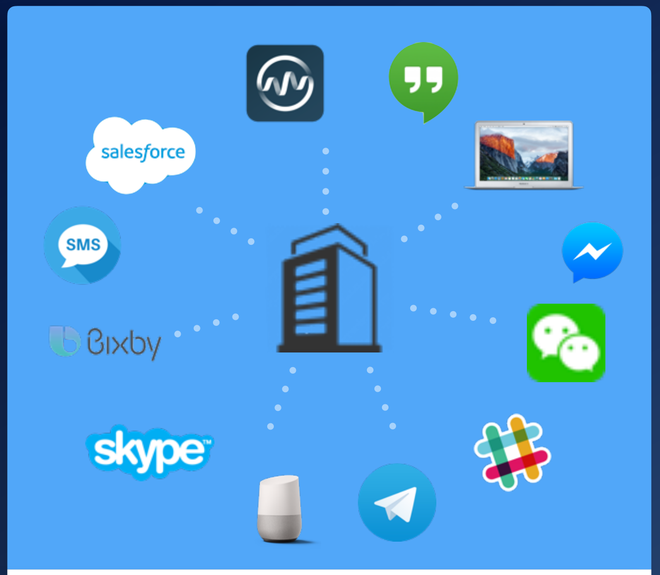

## Bot Builder

No Coding Required.




## Bot Training

#1 AI/NLP Model.




## Bot Deployment

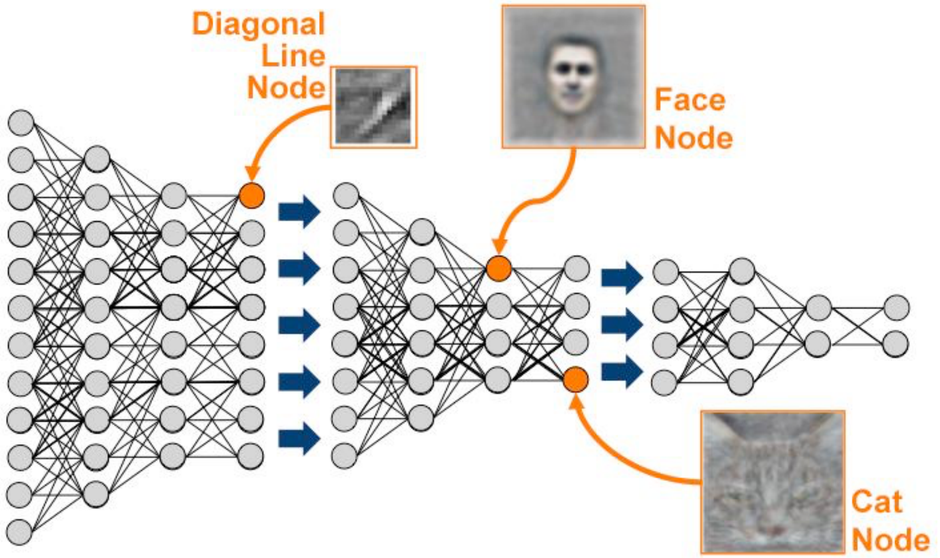
Build Once, Deploy Everywhere.

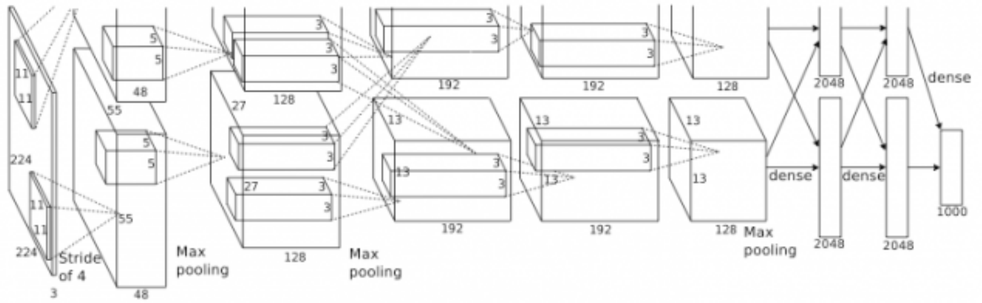
# How to Make a Bot Intelligent?

- Natural Language Understanding
  - Information Extraction
  - Entities
  - Intents
  - Actions
- Natural Language Generation
  - Generating Response

# Deep Learning

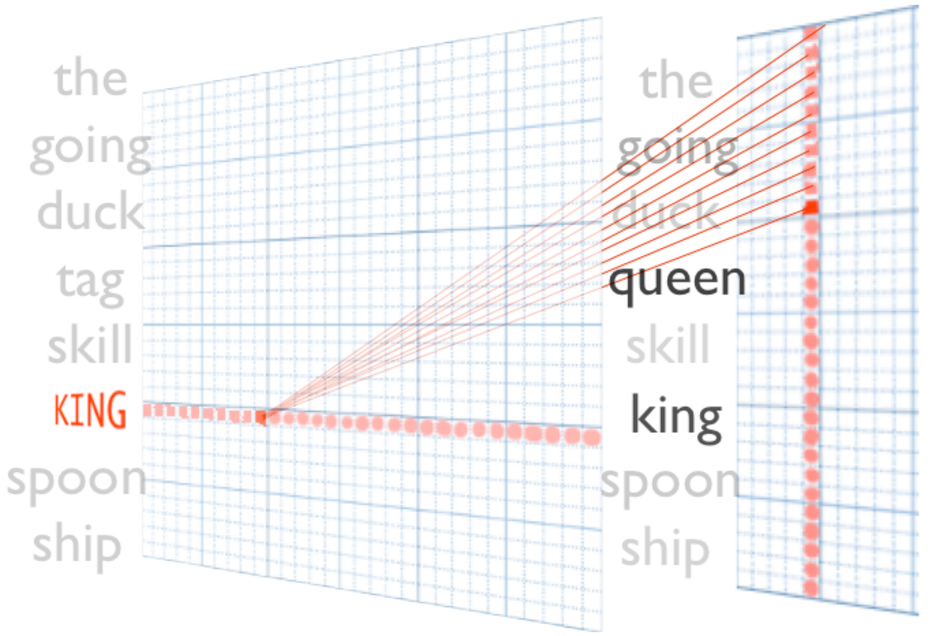
- Traditional Machine Learning
  - Human designed features and representations
  - Optimize weights to combine
- Deep Learning
  - Deep Neural Network
  - Learn good features and multiple levels of representations

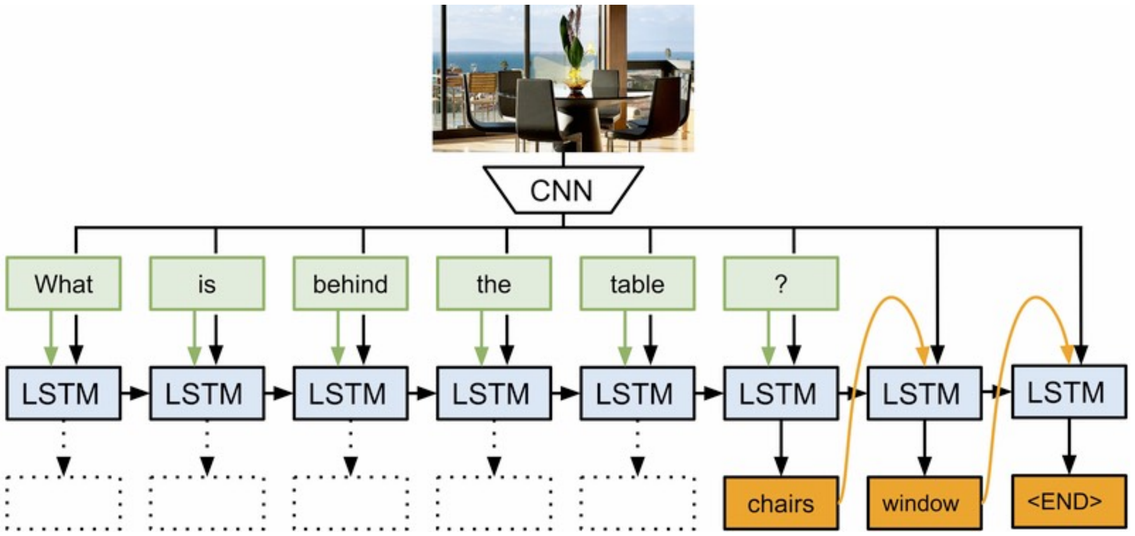


# Deep Learning for NLP

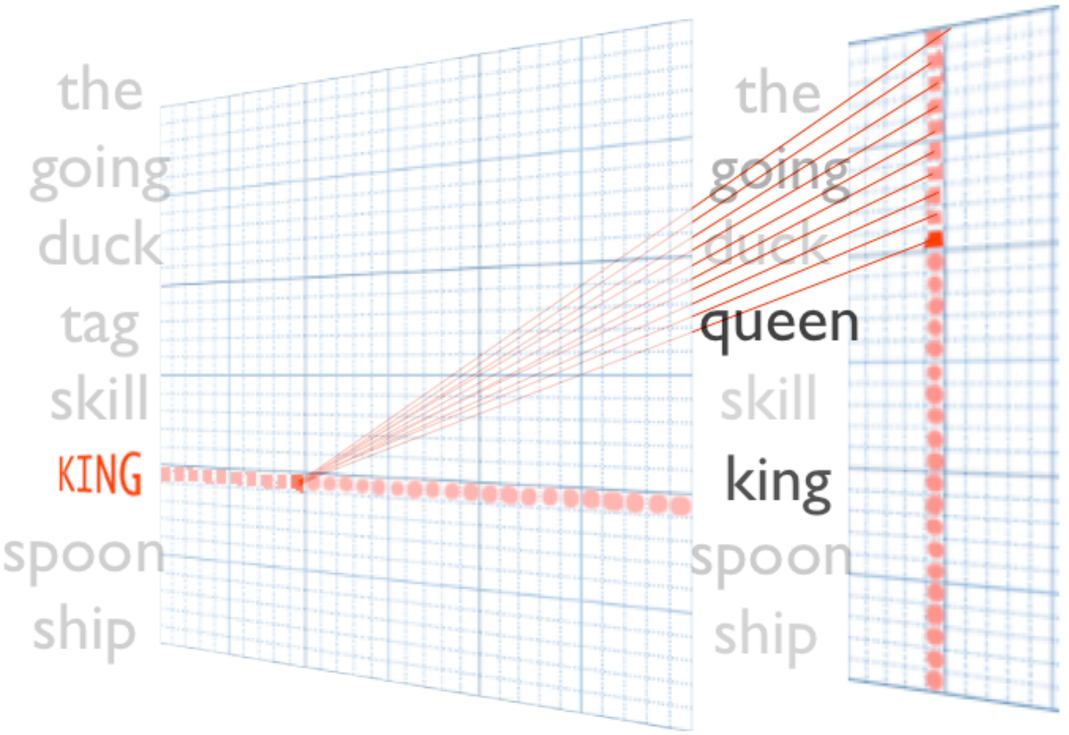
- Language Translation
- Image Captioning
- Text Summarization
- Parts-of-speech Tagging
- Named Entity Recognition
- Natural Language Generation
- Question-Answering
- Optical Character Recognition
- Speech Recognition
- Machine Reading Comprehension



# Neural Network for Word Embedding

- Word Embedding: Word2Vec
- Embed words in continues vector space
- Semantically similar words are mapped to nearby points
- Enables powerful operations
- “King”-“Man”+”Woman” -> “Queen”

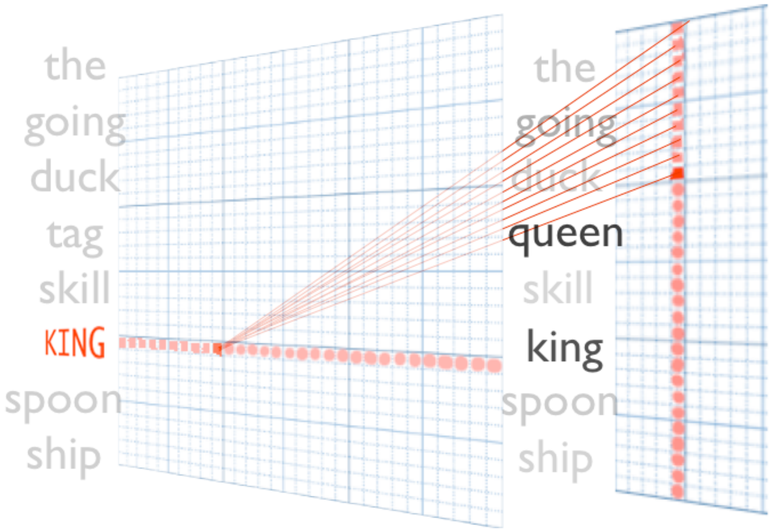


# Bag of Words - Curse of Dimensionality

- Before word embeddings - Bag of words
  - Dictionary of words & counts in the text
  - Easy feature generation technique
- Limitations
  - Hard to capture order of words
  - Curse of dimensionality - limited vocabulary - similar words don’t match

# Word Embeddings Cont'd

- Word Embedding: mapping words to a higher dimensional space, typically 200-500, e.g.,
  - W('King') = (0.2, -0.4, 0.9, …)
  - W('Queen') = (0.1, -0.3, 0.8, …)
- Learn representations of words
- How:  two layer NN to learn word representations by predicting validity of phrases

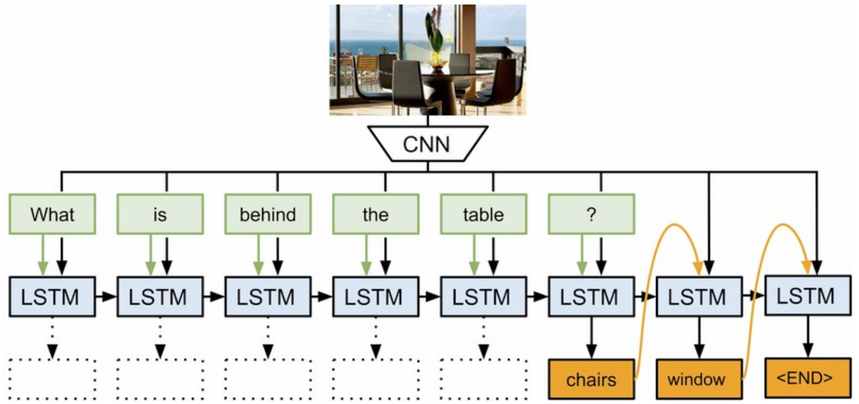


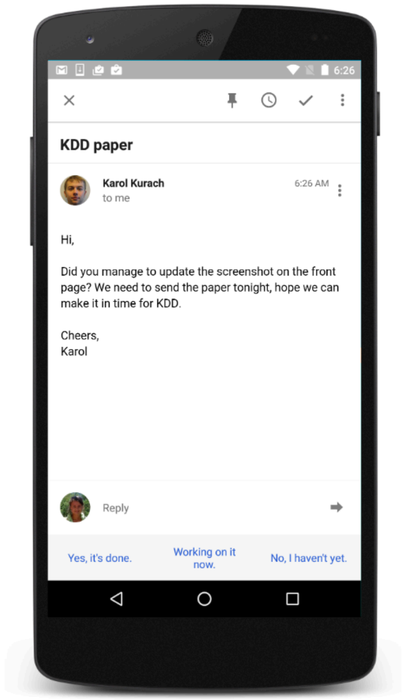

| Relationship | Example 1 | Example 2 | Example 3 |
|---|---|---|---|
| France - Paris | Italy: Rome | Japan: Tokyo | Florida: Tallahassee |
| big - bigger | small: larger | cold: colder | quick: quicker |
| Miami - Florida | Baltimore: Maryland | Dallas: Texas | Kona: Hawaii |
| Einstein - scientist | Messi: midfielder | Mozart: violinist | Picasso: painter |
| Sarkozy - France | Berlusconi: Italy | Merkel: Germany | Koizumi: Japan |
| copper - Cu | zinc: Zn | gold: Au | uranium: plutonium |
| Berlusconi - Silvio | Sarkozy: Nicolas | Putin: Medvedev | Obama: Barack |
| Microsoft - Windows | Google: Android | IBM: Linux | Apple: iPhone |
| Microsoft - Ballmer | Google: Yahoo | IBM: McNealy | Apple: Jobs |
| Japan - sushi | Germany: bratwurst | France: tapas | USA: pizza |

Example of similar word vectors

# Sequence Learning: Response Generation

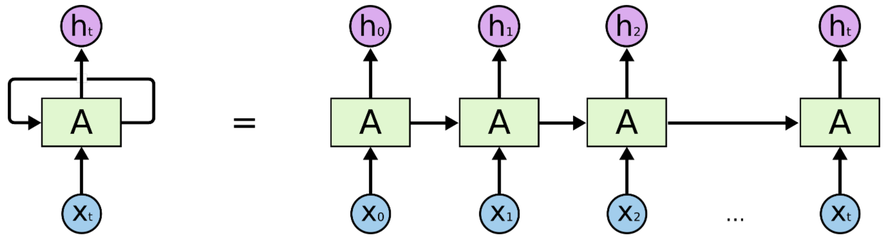
- Automated Response Generation
  - Sequence 2 Sequence Model
  - Recurrent Neural Network (RNN)
  - Long Short Term Memory Network (LSTM)
- Example: GMail Smart Reply
  - Automated Response Suggestions

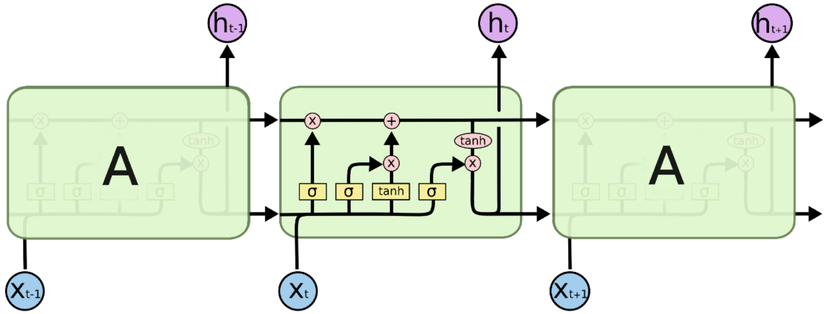


# Sequence Learning: RNN And LSTM

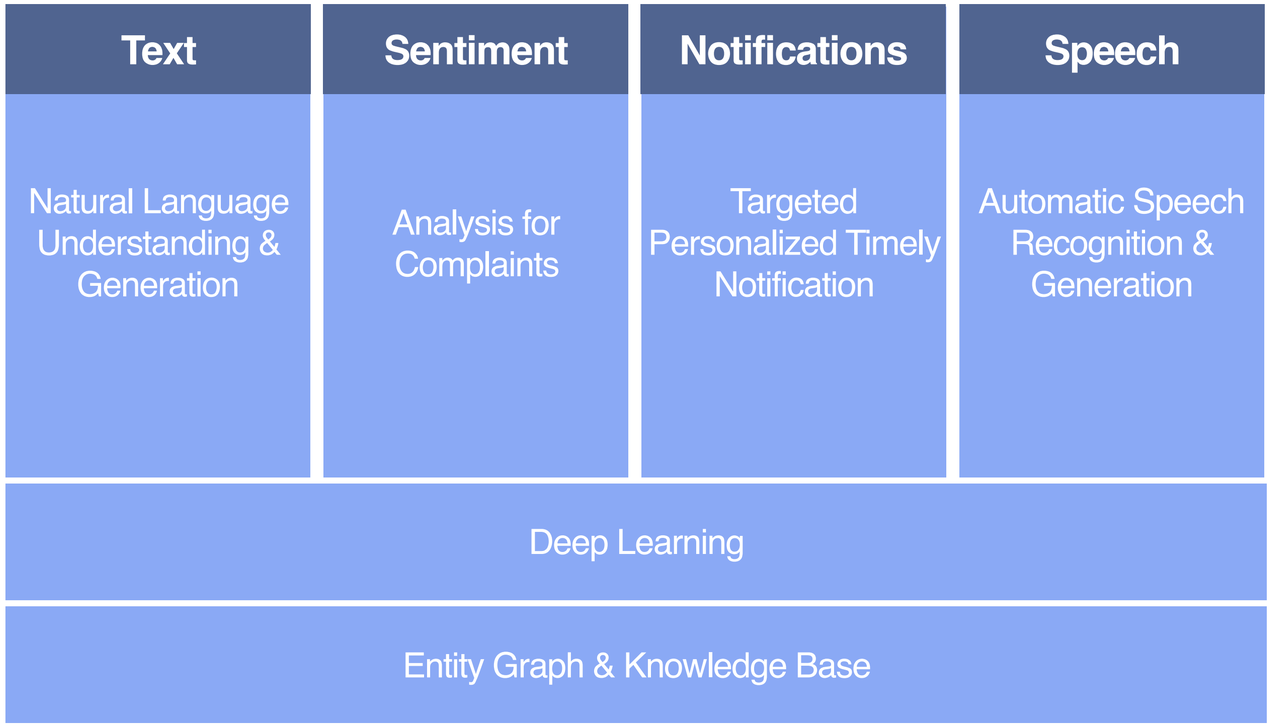
- Recurrent Neural Network
  - Output of a module go into a module of same type (recurrent)
  - Good for capturing a sequence
- Long Short Term Memory Network
  - Long running cell state: forget & add new values
  - Output: combination of cell state, previous output, and new input



# Training Deep Learning Models for NLP

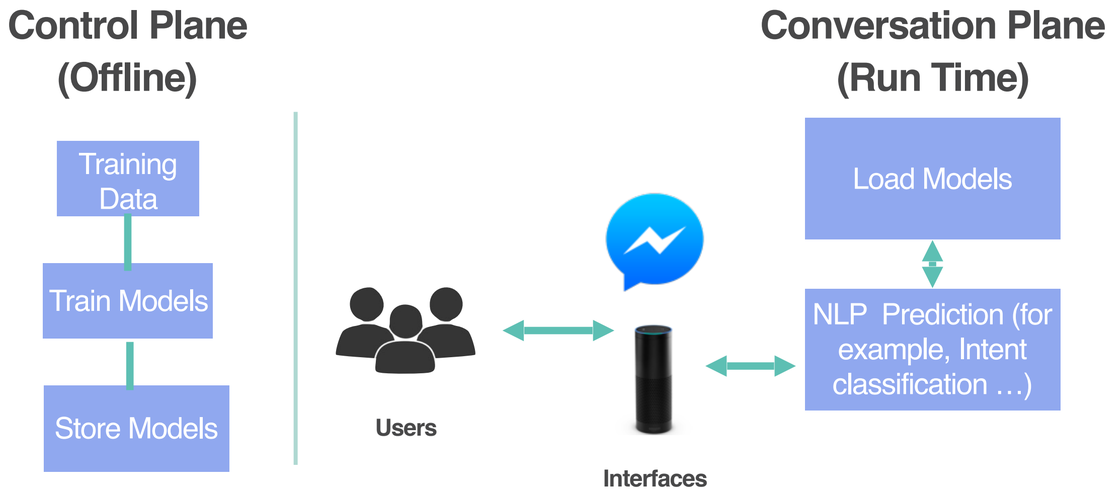
- Intent Classification
  - Deep Learning — LSTM
- Information Extraction
  - Named Entity Recognition (NER)
  - Slot attributes
- Sentiment and Complaint Classifier
- Knowledge Base & Semantic Search
- Machine Reading Comprehension



Passage AI

# Scaling Training Deep Learning Models For NLP

- Off line: Started with a script for training models
- Run Time: A service for prediction during runtime
- However, the number of models are reaching in thousands
- Hard to manage model training script for each of the bot



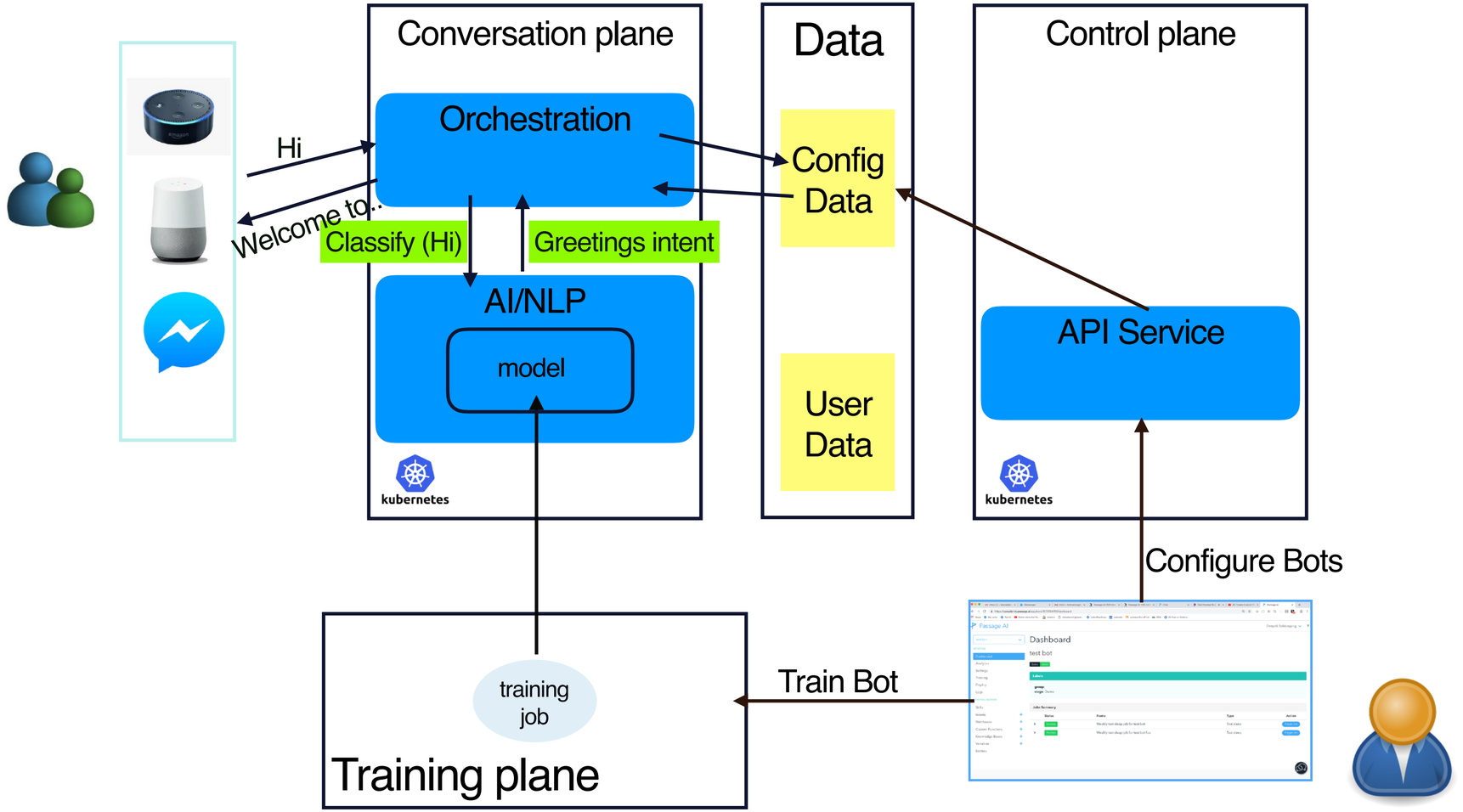
# Outline

Conversational AI and Deep Learning

Need for a Jobs framework on Kubernetes

Our Jobs architecture

# Passage AI Architecture



Passage AI

# Passage AI Architecture

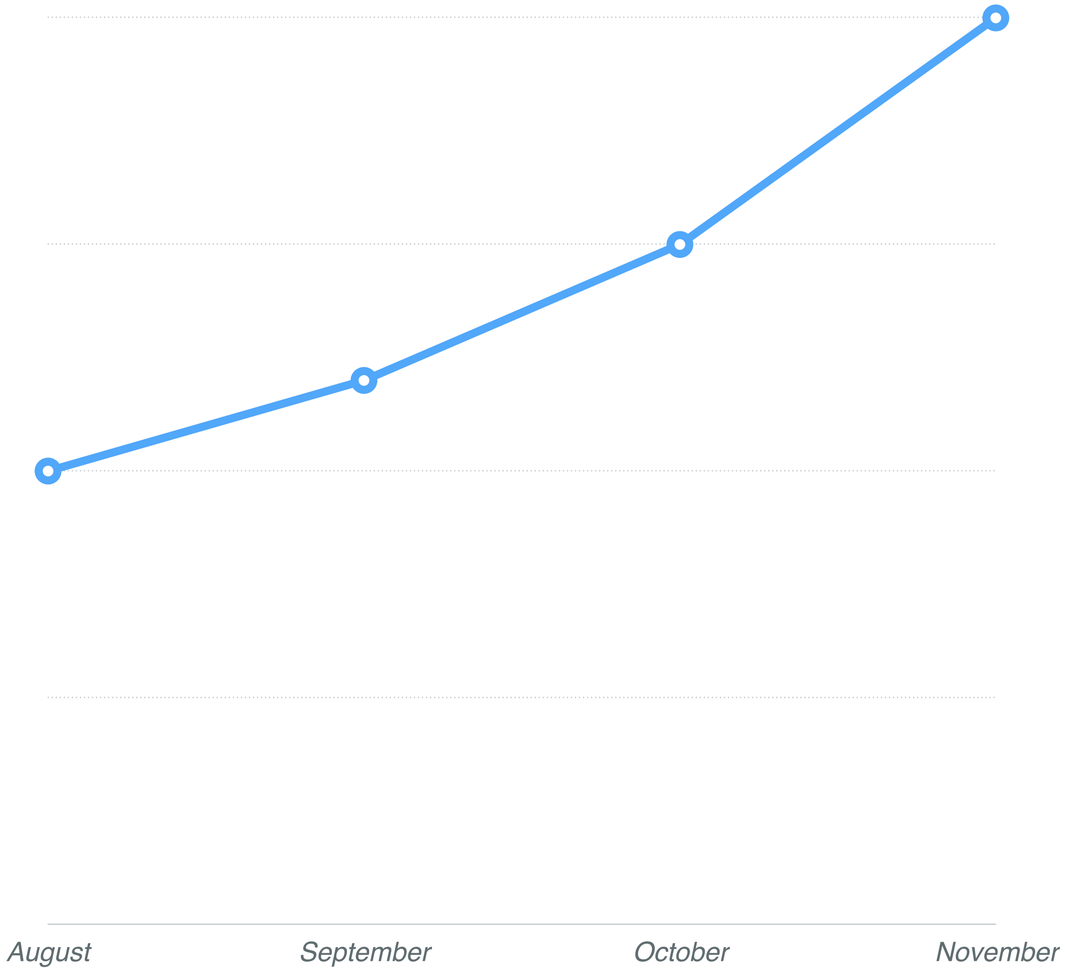

(Jobs)

Training plane

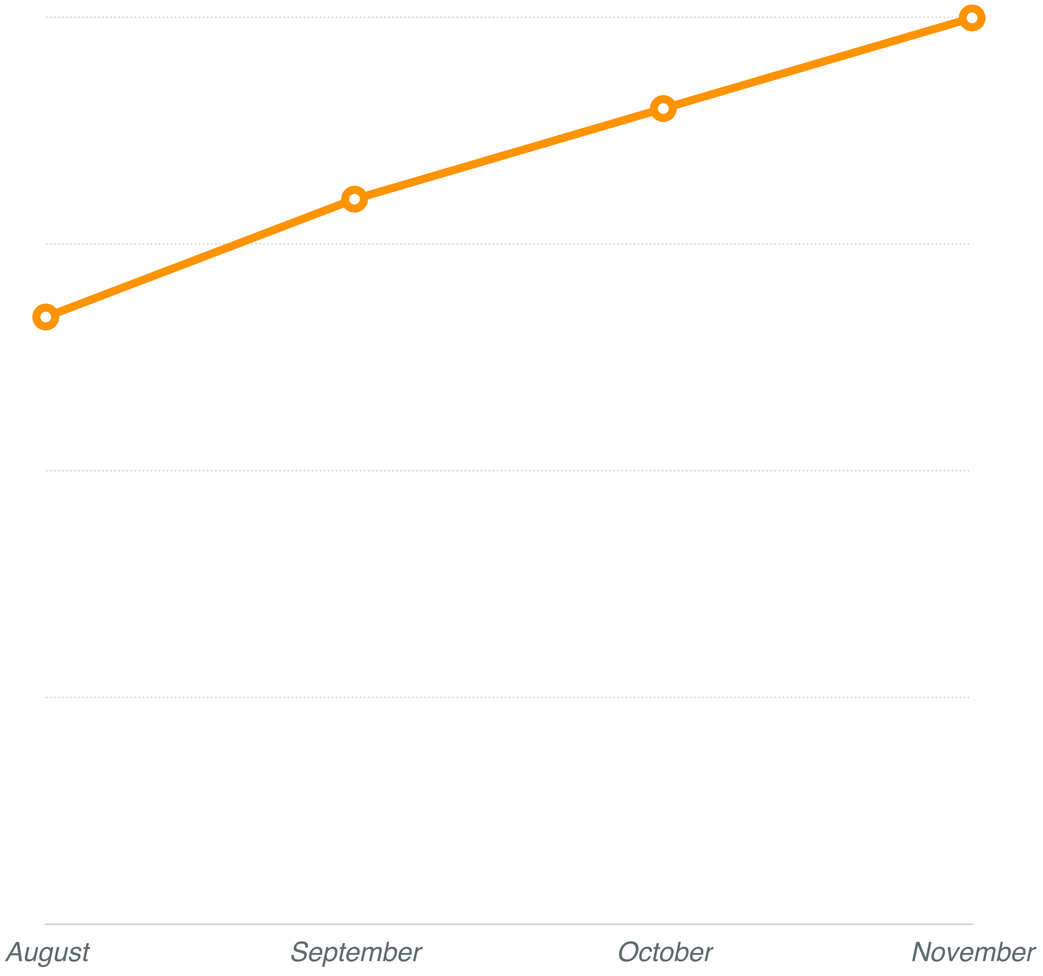
# When do we train a new model for a bot ?

- When a new bot is created
- When a bot is changed
  - utterances are added or modified
  - New training data is available

## # of Bots

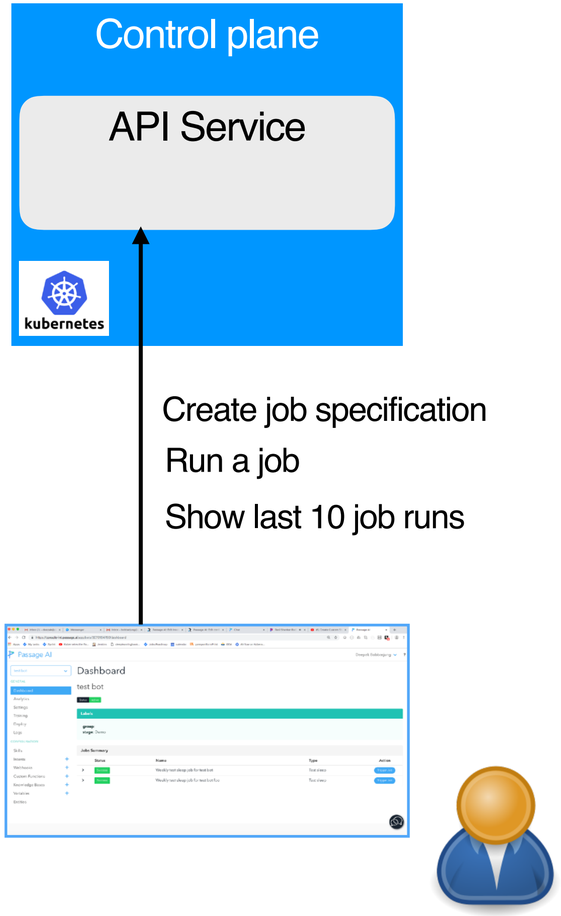



## # of Bot changes per day



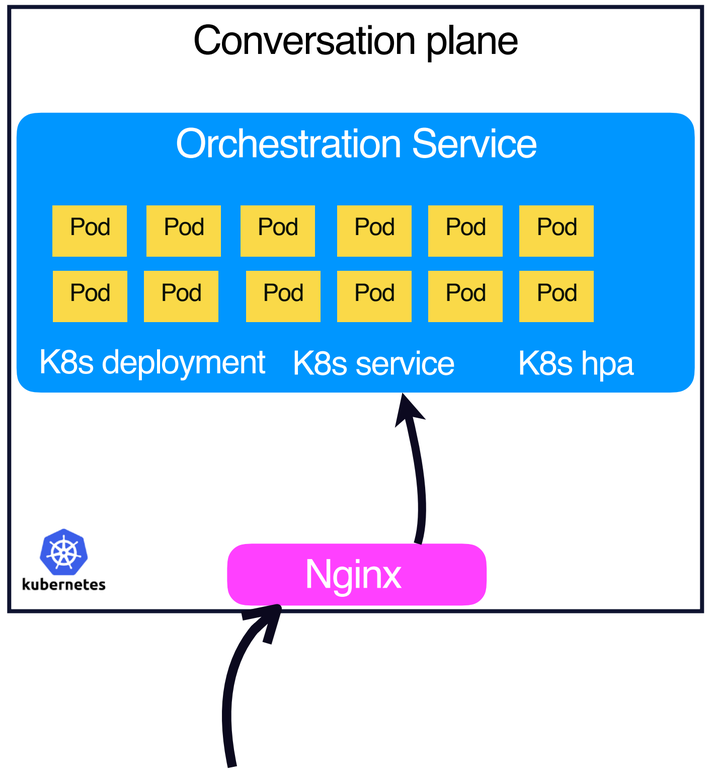
# Why do we need a Jobs framework?

- Run jobs at scale
- Eliminate out of band scripts that tend to become ‘tribal’.
- APIs and UI for exposing jobs to our customers in our Bot Builder UI.
- Reporting and auditing around jobs.



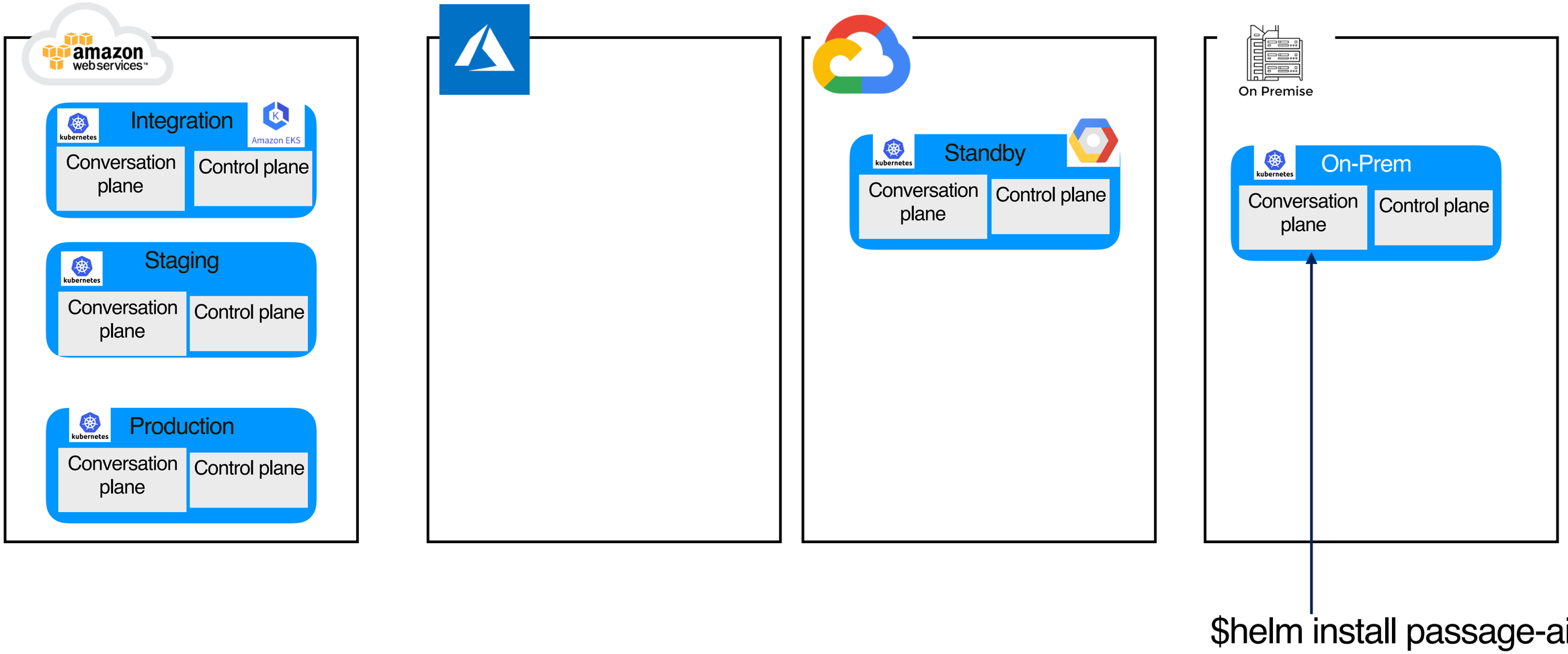
# Why Kubernetes (K8S) For Our Microservices?

- Scale and availability of our microservices



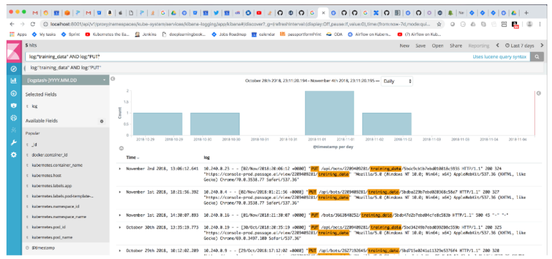
# Why Kubernetes?(Contd)

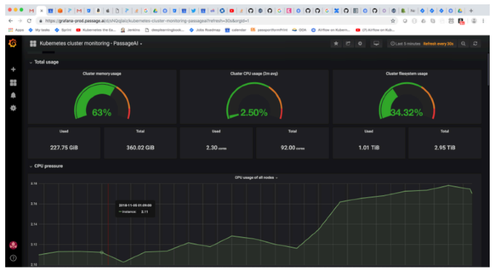
- Cloud Agnostic and On-prem ready

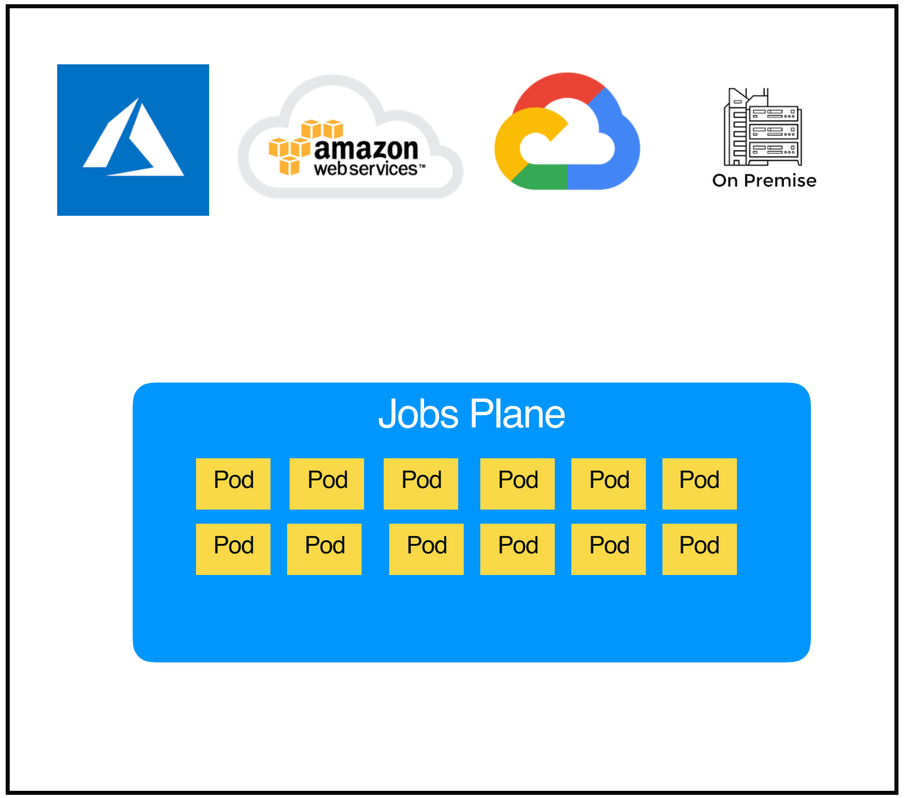


# Why Create The Jobs Framework In Kubernetes?

- Jobs should also be cloud-agnostic and on-prem ready
- Handle scale and availability in the same way as our microservices
- Same set of tools for monitoring, logging and auditing.







# Outline

Conversational AI and Deep Learning

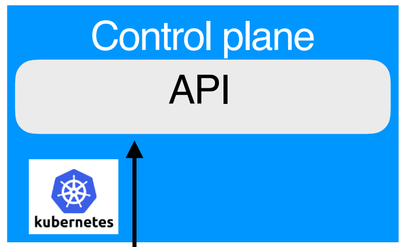
Need for a Jobs framework on Kubernetes

Our Jobs architecture

# Example Job types in our system

- Training deep learning models
- Extracting and indexing knowledge base articles
- Nightly testing of our bots

# Job Specification




Create a job spec

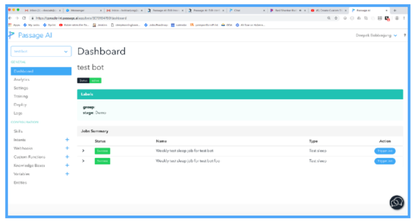
Job type: “Training”

Schedule: “Every Monday at 1 AM”

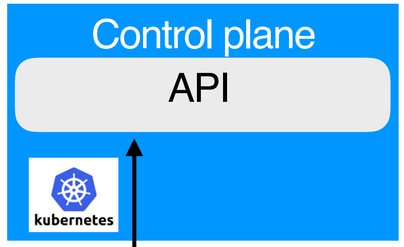
Training specific params:

- bot ID, training data

priority




# Job Object




Trigger a job from a job spec

job_spec_id: <id>

progress: 25

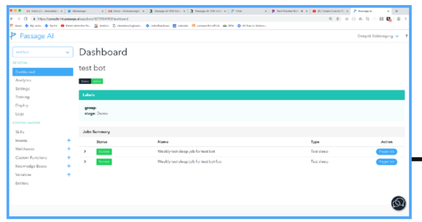
state: in_progress

- data: < confusion matrix>

description: performing training

Passage AI

# Jobs Architecture




Trigger training job from job spec

Control plane

API Service

jobs

job_specs

Jobs plane

Create a Job (params)

Jobs Service

Update job progress

Add item on Q

training Q

KB Index Q

Bot Testing Q

Pickup item

Pod1

gpu

Pod2

Pod3

gpu

Pod4

K8s deployment of training pods

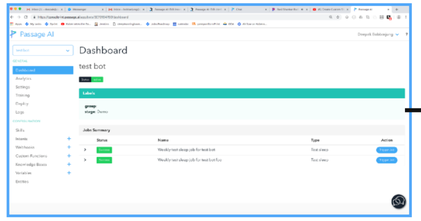
Pod1

Pod2

Pod1

Pod2

# Jobs Architecture (scheduled jobs)




Get job status

Control plane

API Service

jobs

job_specs

Jobs plane

Trigger job from job spec

Update job progress

Jobs Service

CronJob

Scheduler

kubernetes

trainingQ

KB Index Q

Bot Testing Q

Pod1

Pod2

Pod3

Pod4

K8s deployment of training pods

Pod1

Pod2

Pod1

Pod2

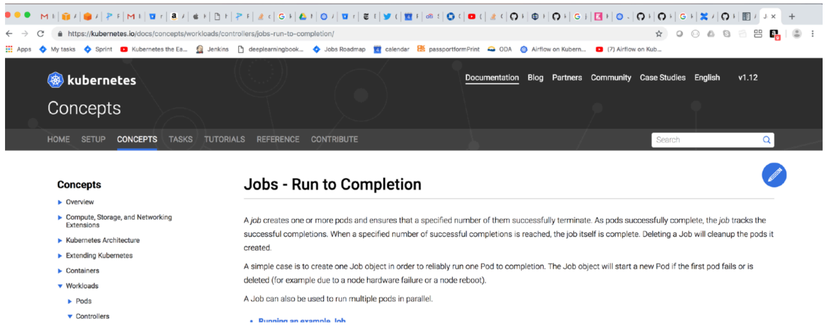
kubernetes

# Alternatives that we considered

Apache Airflow

Kubeflow

Azkaban



# Thank You

Passage AI


Mitul Tiwari

mitul@passage.ai

@Mitultiwari

Deepak Bobbarjung

deepak@passage.ai

@Bobbarjung