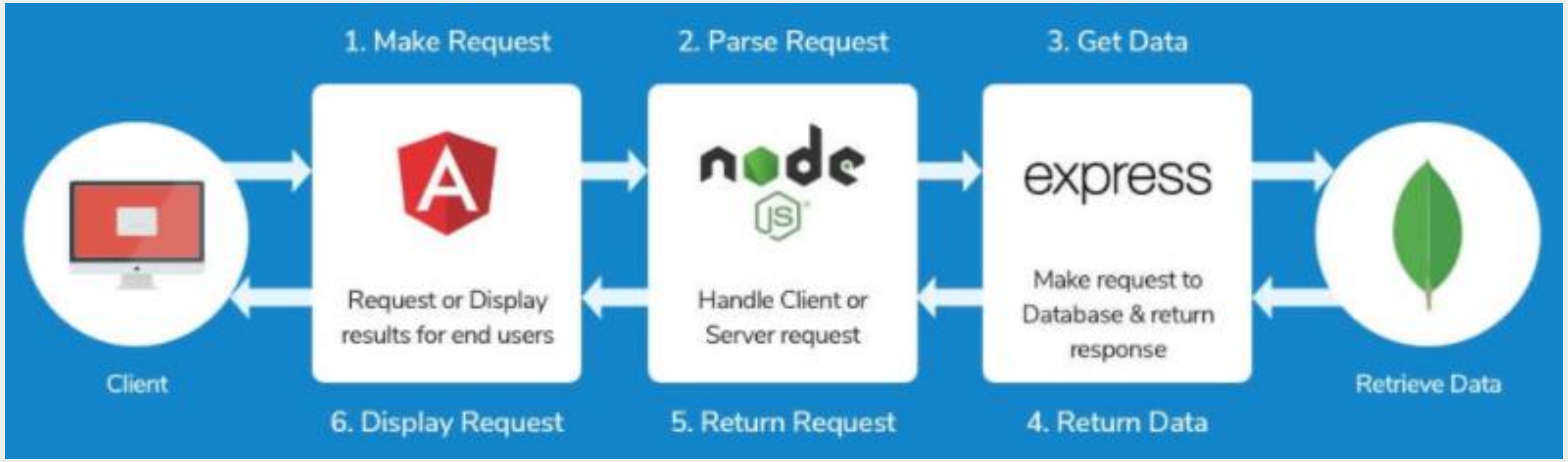
05

Lab intro

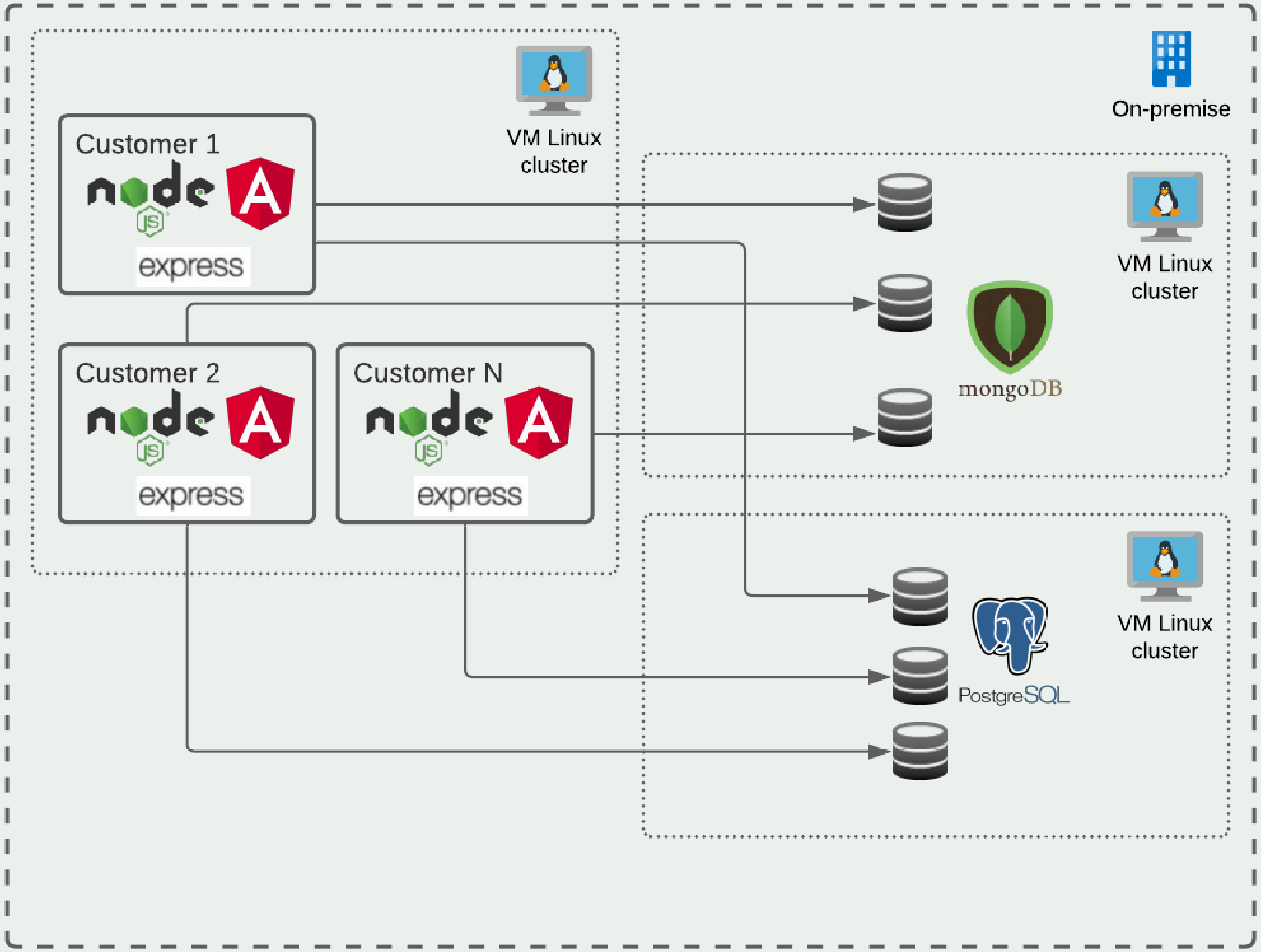
As-is

Conference site solution

- Fabrikam Medical Conferences provides conference web site services tailored to the medical community. They handle over 100 conferences per year and growing.
- The conference sites are typically low budget websites for conferences usually between 100 and 1500 attendees.
- Today the sites are hosted on-premise.
- The conference web sites are built with the MEAN stack (Mongo, Express, Angular, Node.js).



As-is
Current architecture



To-be
Customer business objectives

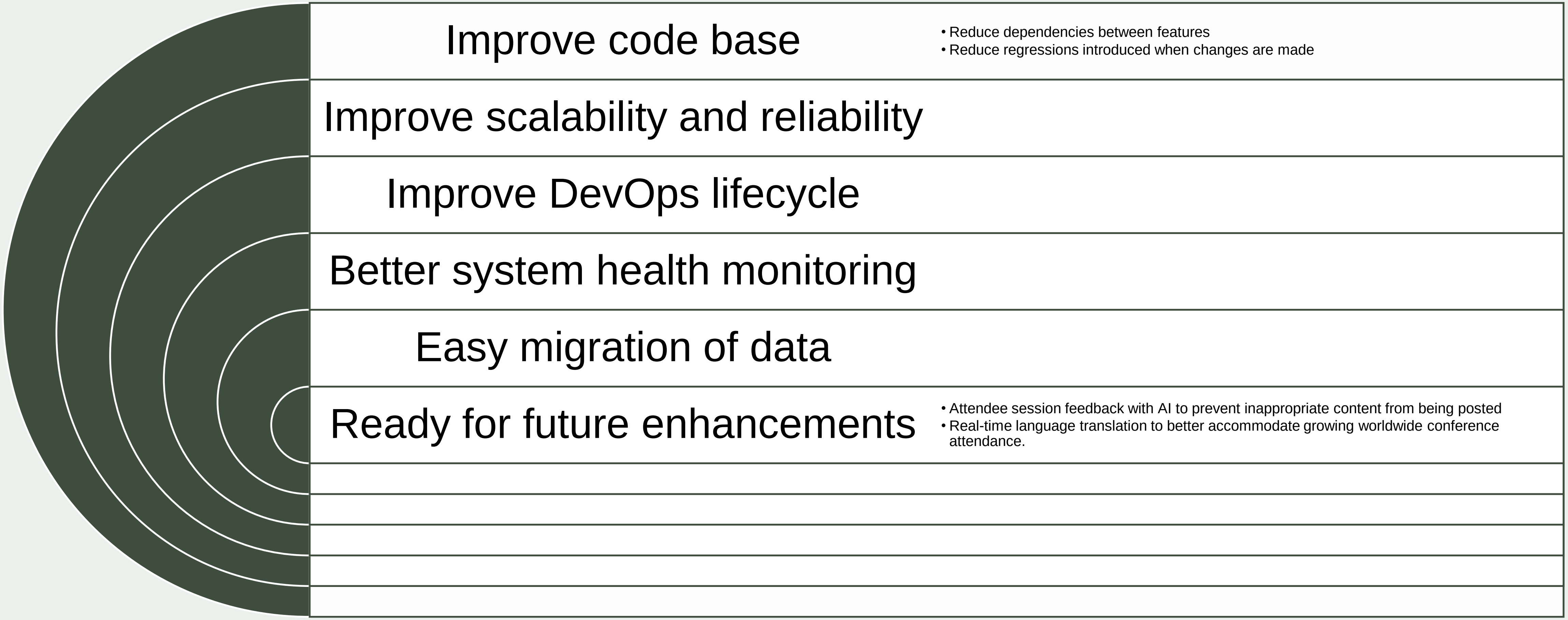
Lower costs

Improve customer satisfaction

Expand: Global reach and larger conferences

Be ready for the future

To-be
Customer technical objectives



Improve code base

- Reduce dependencies between features
- Reduce regressions introduced when changes are made

Improve scalability and reliability

Improve DevOps lifecycle

Better system health monitoring

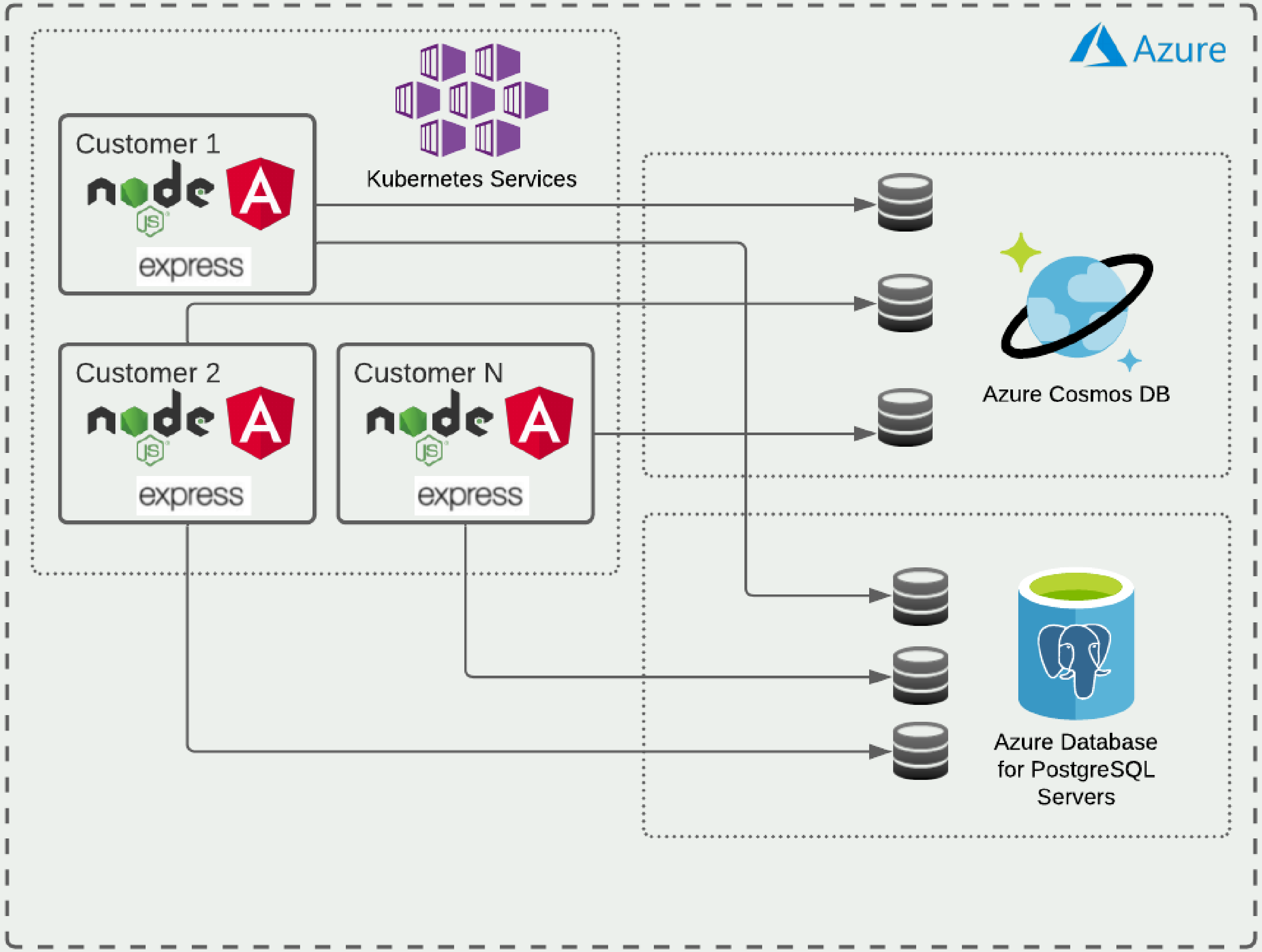
Easy migration of data

Ready for future enhancements

- Attendee session feedback with AI to prevent inappropriate content from being posted
- Real-time language translation to better accommodate growing worldwide conference attendance.

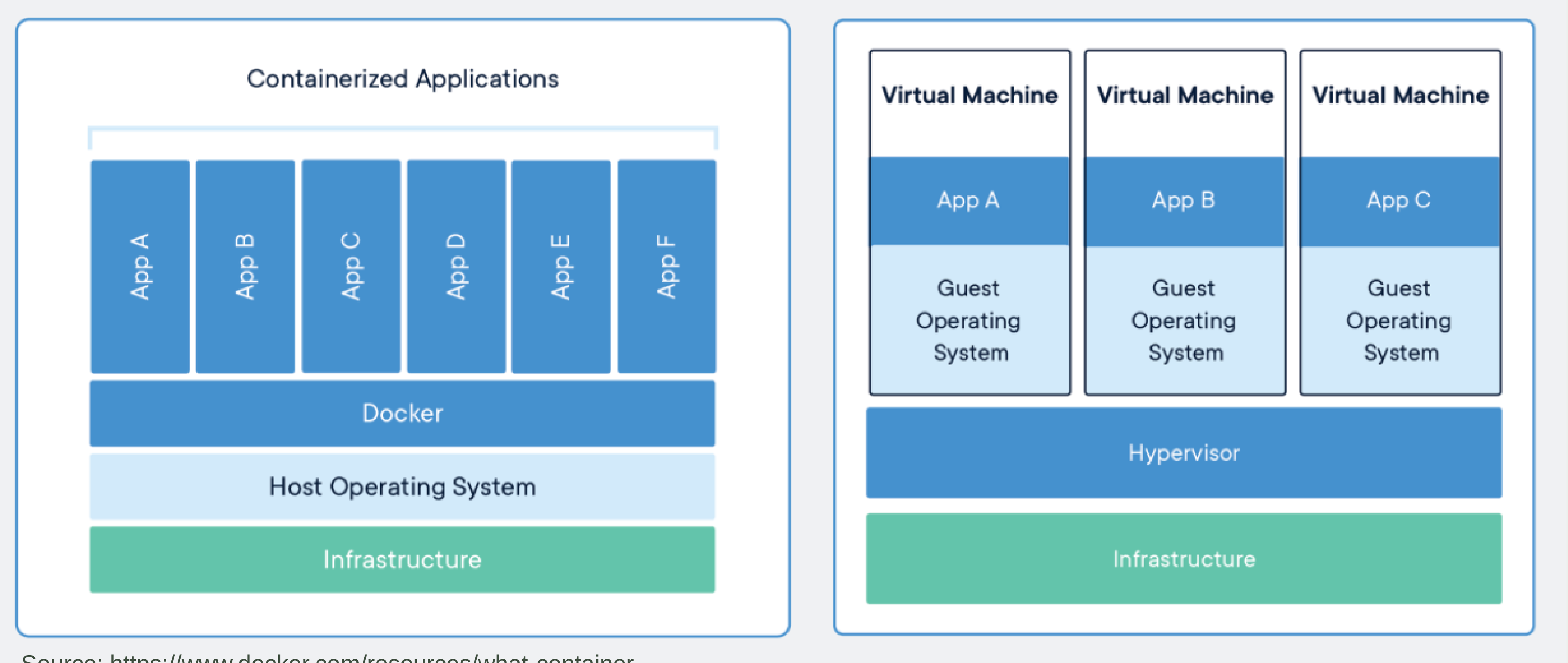
To-be

Future architecture



To-be

“Containerize”



Source: https://www.docker.com/resources/what-container

To-be

Virtual machine vs. Docker

	Docker Container	Virtual Machines (VMs)
Boot-Time	Boots in a few seconds.	It takes a few minutes for VMs to boot.
Runs on	Dockers make use of the execution engine.	VMs make use of the hypervisor.
Memory Efficiency	No space is needed to virtualize, hence less memory.	Requires entire OS to be loaded before starting the surface, so less efficient.
Isolation	Prone to adversities as no provisions for isolation systems.	Interference possibility is minimum because of the efficient isolation mechanism.
Deployment	Deploying is easy as only a single image, containerized can be used across all platforms.	Deployment is comparatively lengthy as separate instances are responsible for execution.
Performance	Limited performance	Native performance

Kubernetes Cluster

Nodes

Basic machinery where pods can be wound up. In practice, it is an Azure VM in AKS. Each node has capacity in terms of cpu, ram, and hard disk.

Pods

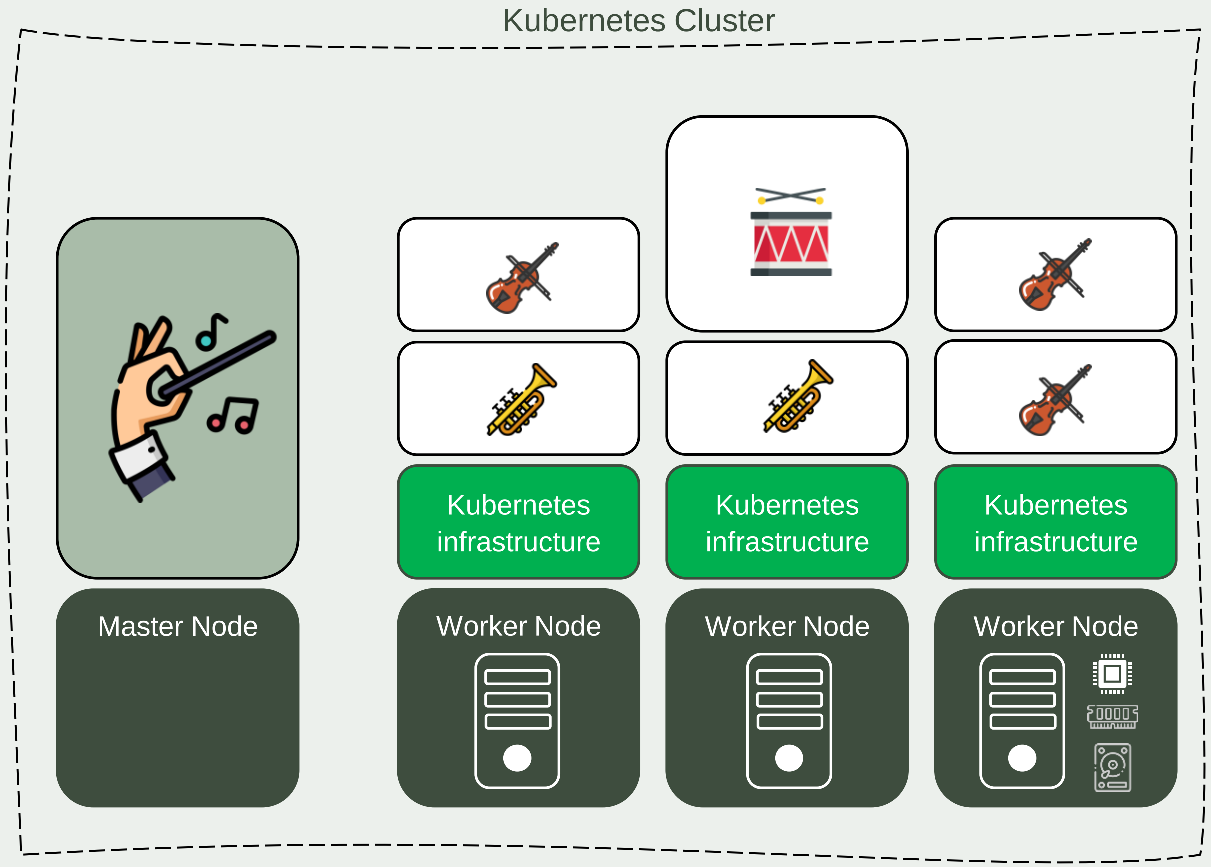
Container instance that executes an application, each pod uses a share of the underlying node's capacity.

Kubernetes infrastructure

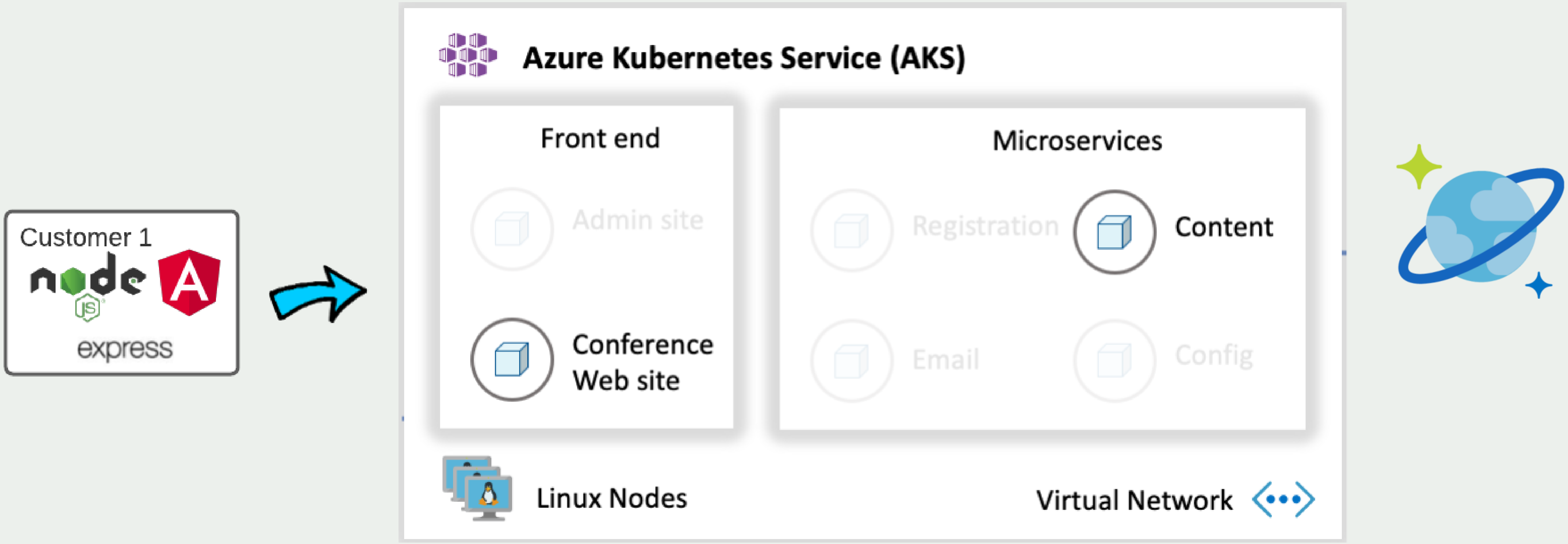
Infrastructure that ensures that each node is part of the k8s cluster and that the node can run pods

Master/master node

Responsible for running the processes that control the cluster


Kubernetes Cluster
Kubernetes infrastructure
Kubernetes infrastructure
Kubernetes infrastructure
Master Node
Worker Node
Worker Node
Worker Node

Migrate to microservice architecture



Lab

- **Development setup**
  - Get the Fabkrikam Medical source code and commit to git
  - Create a Github account and a git repository
  - Clone the git repository to the development build agent.

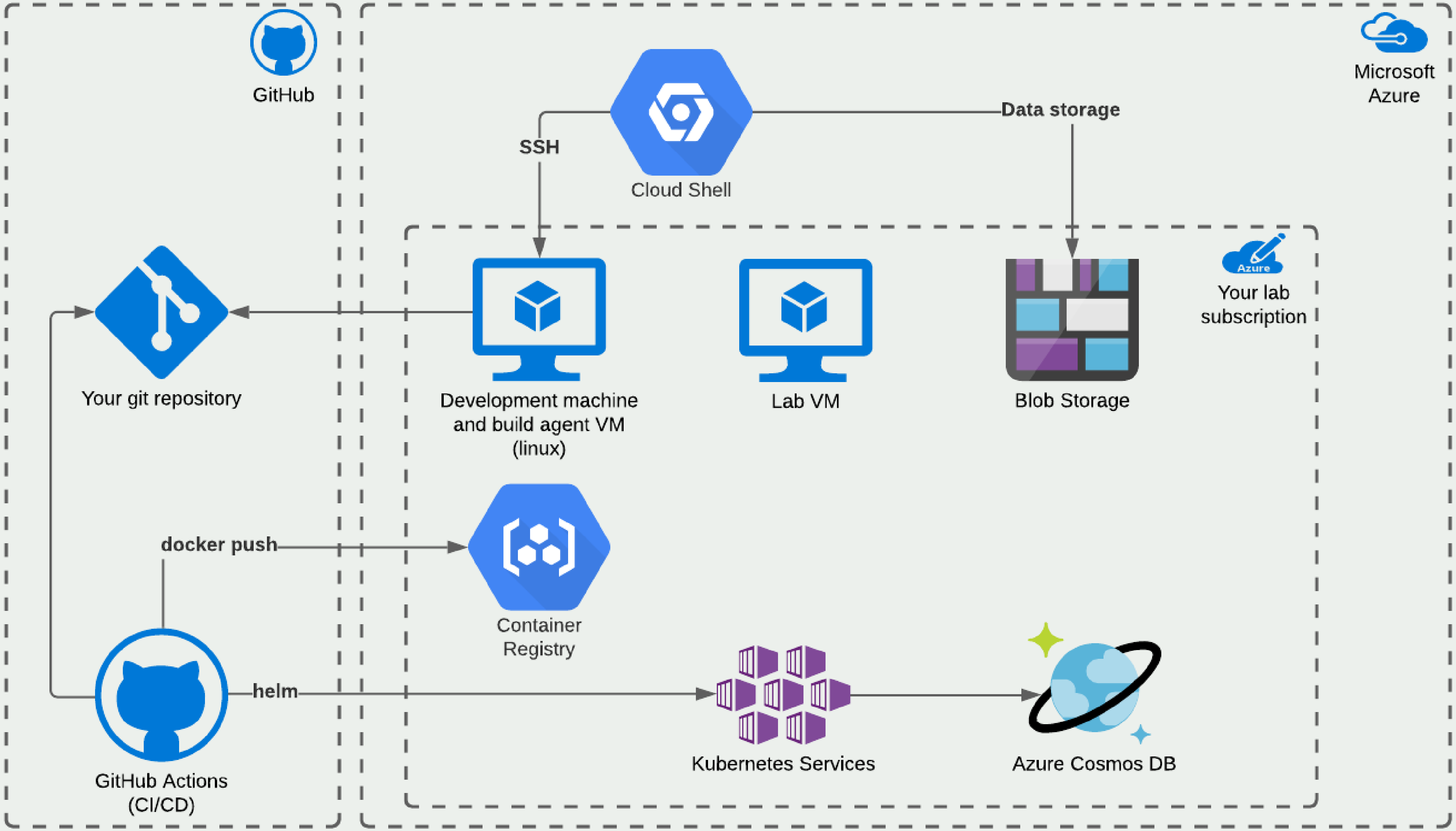
- **Exercise 1**
  - Create a Conference site (content-web) with the Content microservice (content-api) and Mongo DB on Linux (build agent).
  - “Dockerize” content-api
  - Setup and run the components in a Docker “containerized” setup
  - Upload Docker images to Azure Container Registry
  - Setup CI pipeline (Github Actions) to build and upload Docker images.

- **Exercise 2**
  - Migrate the existing Mongo DB data to Cosmos DB

- **Exercise 3**
  - Deploy content-web and content-api to Azure Kubernetes Services (AKS).
  - Setup CD pipeline for automated deployment to AKS.

- **Exercise 4**
  - Setup and test scaling of AKS
  - Setup and test scaling of Cosmos DB

- **Exercise 5**
  - Improvements and fine tuning of scaling.
  - Run a “rolling” update of content-web
  - Setup routing of trafic into the AKS cluster
  - Introduction to multi-region load-balancing with Traffic Manager


GitHub
Microsoft Azure
Cloud Shell
SSH
Data storage
Your lab subscription
Your git repository
Development machine and build agent VM (linux)
Lab VM
Blob Storage
docker push
Container Registry
helm
Kubernetes Services
Azure Cosmos DB
GitHub Actions (CI/CD)

Lab

Get started

1. Sign-up link https://bit.ly/2OLTVuk
2. Activation code ACTIVATE12903
3. Launch lab
4. Follow lab instructions

Tips

- Open Cloud Shell (https://shell.azure.com) in your local browser.
- Take it easy. Learn and understand.
- The Environment Details tab in the lab contains valueable variables and references.

Lab
Help

Write on Teams

We will try to help as quickly as possible.

While you wait

1. Browse our reference repository: https://github.com/immeorfj/Fabmedical2021
2. See example files. Remember to update environment specific references, if you use the files. https://github.com/immeorfj/Fabmedical/tree/master/help/files

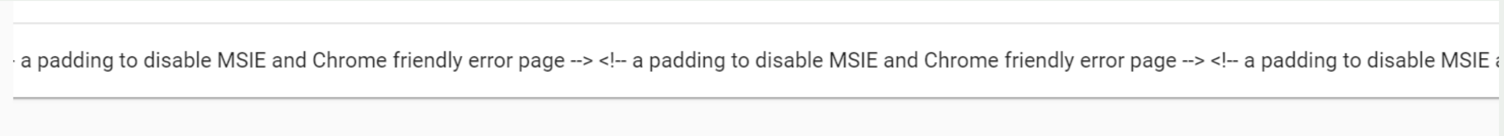
Problems with yaml formatting: http://yamllint.com

Technical issues with the lab

Teams: http://bit.ly/cloudlabs-support
Email: cloudlabs-support@spektrasystems.com

Lab
Known issues in the lab (1/2)

- General
  - SUFFIX = [DeploymentID] on the Environment Details tab.
- Exercise 2
  - Creation of the Azure Database Migration Service ressource may take some time (+15 min). Take a break and pre-start on Exercise 3.
  - Step 4. It is the **Private IP** not the public IP, which must be used. The find the Private IP got to https://portal.azure.com and search for fabmedical. Choose the fabmedical (Virtual machine). Choose Networking. The Private IP is the *NIC Private IP e.g.* 172.16.0.5
- Exercise 3
  - Task 1
    - If the Kubernetes dashboard shows an error like



```
- a padding to disable MSIE and Chrome friendly error page --> <!-- a padding to disable MSIE and Chrome friendly error page --> <!-- a padding to disable MSIE a
```

    - Press the different menu items in the left menu. This will usually fix the issue. Alternatively close the window and run `az aks browse --name fabmedical-SUFFIX --resource-group fabmedical-SUFFIX` from Cloud Shell
  - Task 2.
    - Step 16: If there are issues with exporting a valid api.deployment.yml file copy the contents from the following link: https://raw.githubusercontent.com/immeorfj/Fabmedical2021/master/help/files/api.deployment.yml. Replace 226960 with your own SUFFIX.

Lab
Known issues in the lab (2/2)

- Exercise 3
  - Task 3
    - Step 14: Press Speakers first and validate that data is shown. Then press Sessions. If Sessions do not show any content it is probably because an error has occurred, when calling the api service. Beware that the web application will crash and the site will become unavailable as a result of the error. You can check if it is a api problem by running the following command in Cloud Shell

      ```
      kubectl get pods
      ```

      Copy pod NAME and then with NAME replaced run

      ```
      kubectl logs NAME
      ```

      The error is due to a missing ndex in the sessions collection in Cosmos DB.

      Open https://portal.azure.com and search for Cosmos

      Open the Cosmos resource.

      Choose "Data Explorer”

      Expand contentdb and then expand sessions

      Choose "Scale & Settings”

      Press "Indexing Policy"

      Below "Current index(es)" specify a new

      Name: startTime

      Type: Single Field

      Press Save in the top menu

      Go back to Cloud Shell

      Run `kubectl get pods` and copy pod NAME for web

      Run `kubectl delete NAME`

      Run `kubectl get pods` until both the api and web pods have status Running and 1/1 under Ready.

      Reload the Sessions page in the browser and validate that data is shown.

Lab
If Kubernetes dashboard does not work

Close the dashboard window and restart it by running

```
az aks browse --name <aks name> --resource-group <resource
group name>
```

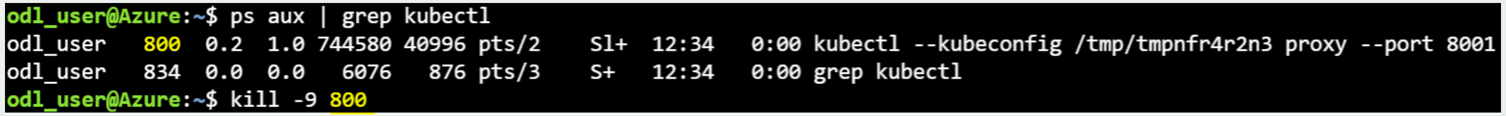
Address already in use error

Find proces id for kubectl –kube-config using:

```
ps aux | grep kubectl
```

Kill the process using:

```
kill -9 <procesid>
```



```
odl_user@Azure:~$ ps aux | grep kubectl
odl_user    800  0.2  1.0 744580 40996 pts/2    Sl+  12:34   0:00 kubectl --kubeconfig /tmp/tmpnfr4r2n3 proxy --port 8001
odl_user    834  0.0  0.0   6076   876 pts/3    S+   12:34   0:00 grep kubectl
odl_user@Azure:~$ kill -9 800
```

Alternatively, to the Kubernetes dashboard use kubectl commands

See https://github.com/immeorfj/Fabmedical2021/wiki/Kubernetes-dashboard-does-not-work

Other useful kubectl commands

Scale: `kubectl scale --replicas=3 deployment/api`
List replica sets: `kubectl get rs`
List deployments: `kubectl get deployments`
Edit deployment: `kubctl edit deployment api`
List services: `kubectl get services`
List pods: `kubectl get pods`
Delete pod: `kubectl pod <pod name>`
Apply changes to a resource: `kubectl apply -f <yaml file path>`

Kubernetes cheat sheet

https://kubernetes.io/docs/reference/kubectl/cheatsheet/