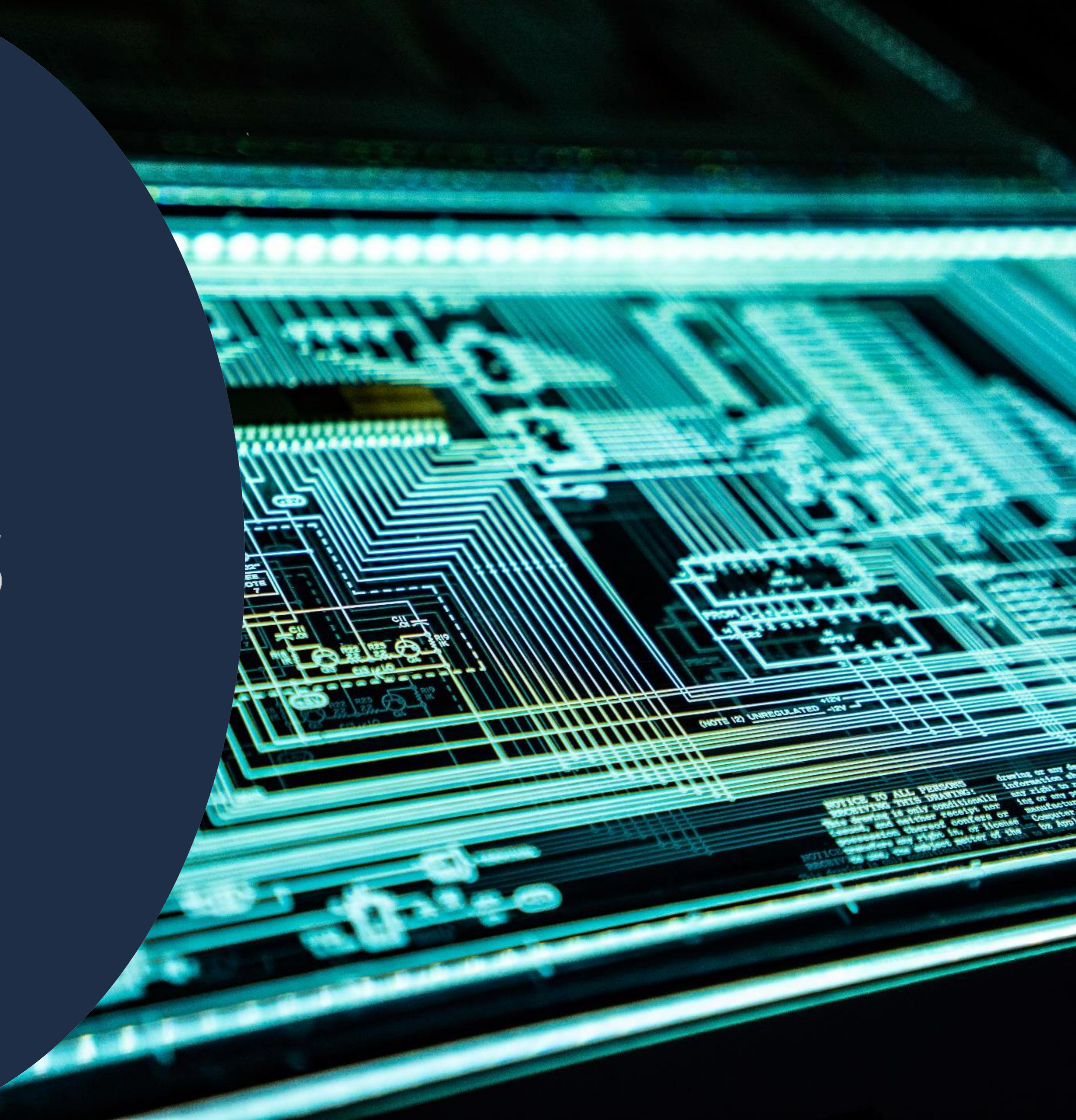




CLOUDBITES

Hands-On Tutorial




NOTICE TO ALL PERSONS RECEIVING THIS DRAWING: This drawing is only conditionally loaned, and neither receipt nor possession thereof confers or conveys any right in, or license to copy, reproduce, or otherwise use the information shown hereon for any purpose other than that for which it was originally prepared. The manufacturer of the product shown hereon is not responsible for any errors or omissions in this drawing.





☰ README.md ✎

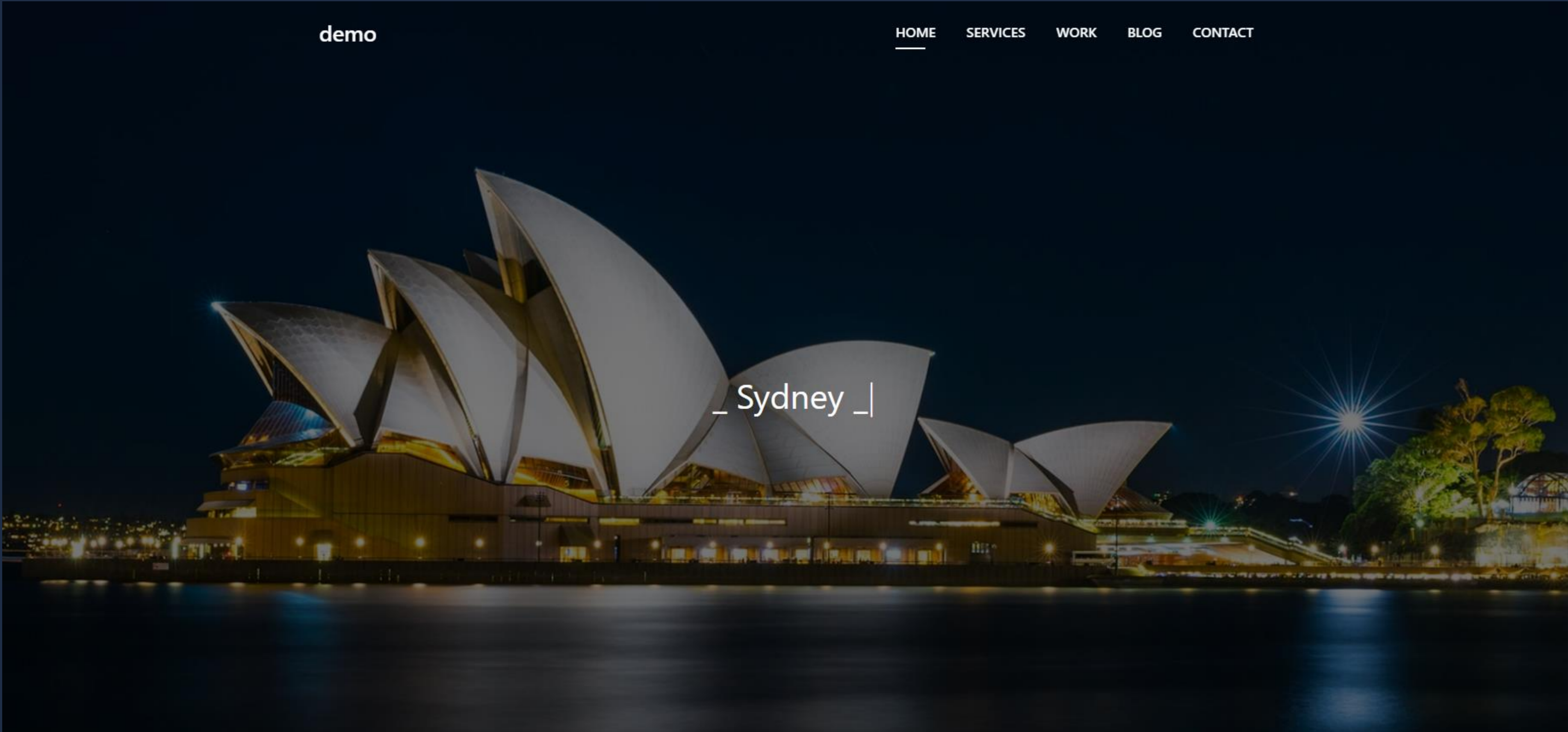
Cloud Bites Tutorial



The idea of the tutorial is to give an introduction to the basics of application deployment in a cloud native environment. A simple website serves as a sample application, this website can be run as a container in various cloud environments. The source code of the sample application is part of the tutorial. The idea is to continuously develop this tutorial. In a first chapter, the basics of container technology, deployment and management of container applications are outlined. This first chapter consists of several parts that build on each other. In a second chapter, the idea is to have a broader look at the ecosystem e. g. various projects from the [Cloud Native Computing Foundation](#) cosmos are going to be discussed.

For simplicity make sure not to be behind a corporate firewall. Comments, additions and collaboration are welcome.

https://github.com/smichard/cloud_bites_tutorial



Agenda

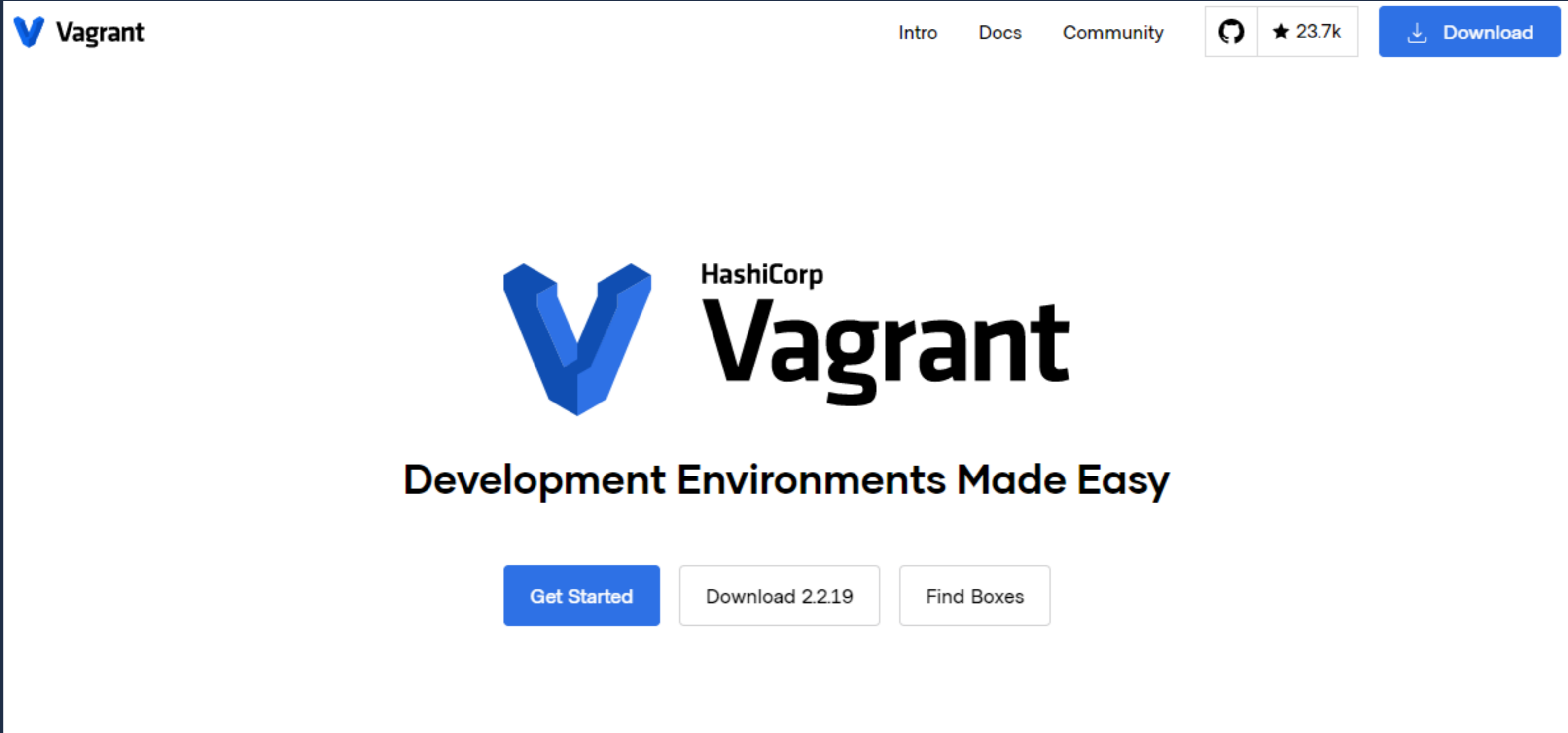
1. Setup of local environment
2. Build and run docker container locally
3. Deploy a local Kubernetes cluster using k3d
4. Basic operations to handle Pods and Deployments
5. Deploy a remote Kubernetes cluster on Google Cloud – part 1
6. Deploy a remote Kubernetes cluster on Google Cloud – part 2
7. Deploy a remote Kubernetes cluster leveraging terraform
8. Visualize Kubernetes workloads with VMware Octant

Setup of local environment

1. Vagrant
vagrantup.com
2. Virtual Box
virtualbox.org
3. Visual Studio Code
code.visualstudio.com




Setup of local environment



vagrantup.com

Setup of local environment



The screenshot shows the VirtualBox.org homepage. On the left is a navigation menu with links for About, Screenshots, Downloads, Documentation (with sub-links for End-user docs and Technical docs), Contribute, and Community. The main content area features the VirtualBox logo, a welcome message, and a large blue button that says "Download VirtualBox 6.1". Below this, there is a "Hot picks" section with three bullet points: "Pre-built virtual machines for developers at Oracle Tech Network", "Hyperbox Open-source Virtual Infrastructure Manager project site", and "phpVirtualBox AJAX web interface project site".

virtualbox.org

Setup of local environment

The image shows a composite of two screenshots. The top-left screenshot is the Visual Studio Code website landing page, featuring the text "Code editing. Redefined." and a "Download for Windows" button. The top-right screenshot shows the Visual Studio Code interface with the Extensions Marketplace on the left, a code editor in the center, and a terminal at the bottom. The code editor displays JavaScript code for a service worker, and the terminal shows the output of a command to start a development server.

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn

Search Docs Download

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows Stable Build

Other platforms and Insiders Edition

By using VS Code, you agree to its [license](#) and [privacy statement](#).

EXTENSIONS: MARKETPLACE

- Python 2019.6.24221 54.9M 4.5
- GitLens — Git sup... 3.8.5 23.1M 5
- C/C++ 0.24.0 25M 3.5
- ESLint 1.9.0 21.9M 4.5
- Debugger for Ch... 4.11.6 20.6M 4
- Language Supp... 0.47.0 18.7M 4.5
- vscod... 8.8.0 17.2M 5
- Vetur 0.21.1 17M 4.5
- C# 1.21.0 15.6M 4

```
JS serviceWorker.js > register > window.addEventListener("load") callback
39
40
41
42 // Add some additional logging to localhost, p
43 // service worker/PWA documentation.
44 navigator.serviceWorker.ready.then(() => {
45   product
46   productSub
47   removeSiteSpecificTrackingException
48   removeWebWideTrackingException
49   requestMediaKeySystemAccess
50   sendBeacon
51   serviceWorker (property) Navigator.serviceWorke...
52   storage
53   storeSiteSpecificTrackingException
54   storeWebWideTrackingException
55   userAgent
56   vendor
57 }
58 function registerValidSW(swUrl, config) {
59   navigator.serviceWorker
60   .register(swUrl)
61   .then(registration => {
```

TERMINAL ... | node

You can now view create-react-app in the browser.

Local: http://localhost:3000/
On Your Network: http://10.211.55.3:3000/

Note that the development build is not optimized.


Ln 43, Col 19 Spaces: 2 UTF-8 LF JavaScript

code.visualstudio.com

Setup of local environment

```
☰ README.md
```

Cloud Bites Tutorial



The idea of the tutorial is to give an introduction to the basics of application deployment in a cloud native environment. A simple website serves as a sample application, this website can be run as a container in various cloud environments. The source code of the sample application is part of the tutorial. The idea is to continuously develop this tutorial. In a first chapter, the basics of container technology, deployment and management of container applications are outlined. This first chapter consists of several parts that build on each other. In a second chapter, the idea is to have a broader look at the ecosystem e. g. various projects from the [Cloud Native Computing Foundation](#) cosmos are going to be discussed.

For simplicity make sure not to be behind a corporate firewall. Comments, additions and collaboration are welcome.

https://github.com/smichard/cloud_bites_tutorial



Agenda

1. Setup of local environment
- 2. Build and run docker container locally**
3. Deploy a local Kubernetes cluster using k3d
4. Basic operations to handle Pods and Deployments
5. Deploy a remote Kubernetes cluster on Google Cloud – part 1
6. Deploy a remote Kubernetes cluster on Google Cloud – part 2
7. Deploy a remote Kubernetes cluster leveraging terraform
8. Visualize Kubernetes workloads with VMware Octant

Build and run docker container locally



```
FROM nginx:alpine  
  
COPY website /usr/share/nginx/html  
  
EXPOSE 80
```

Commands

Build local container image:

```
docker build -t <image_name> .
```

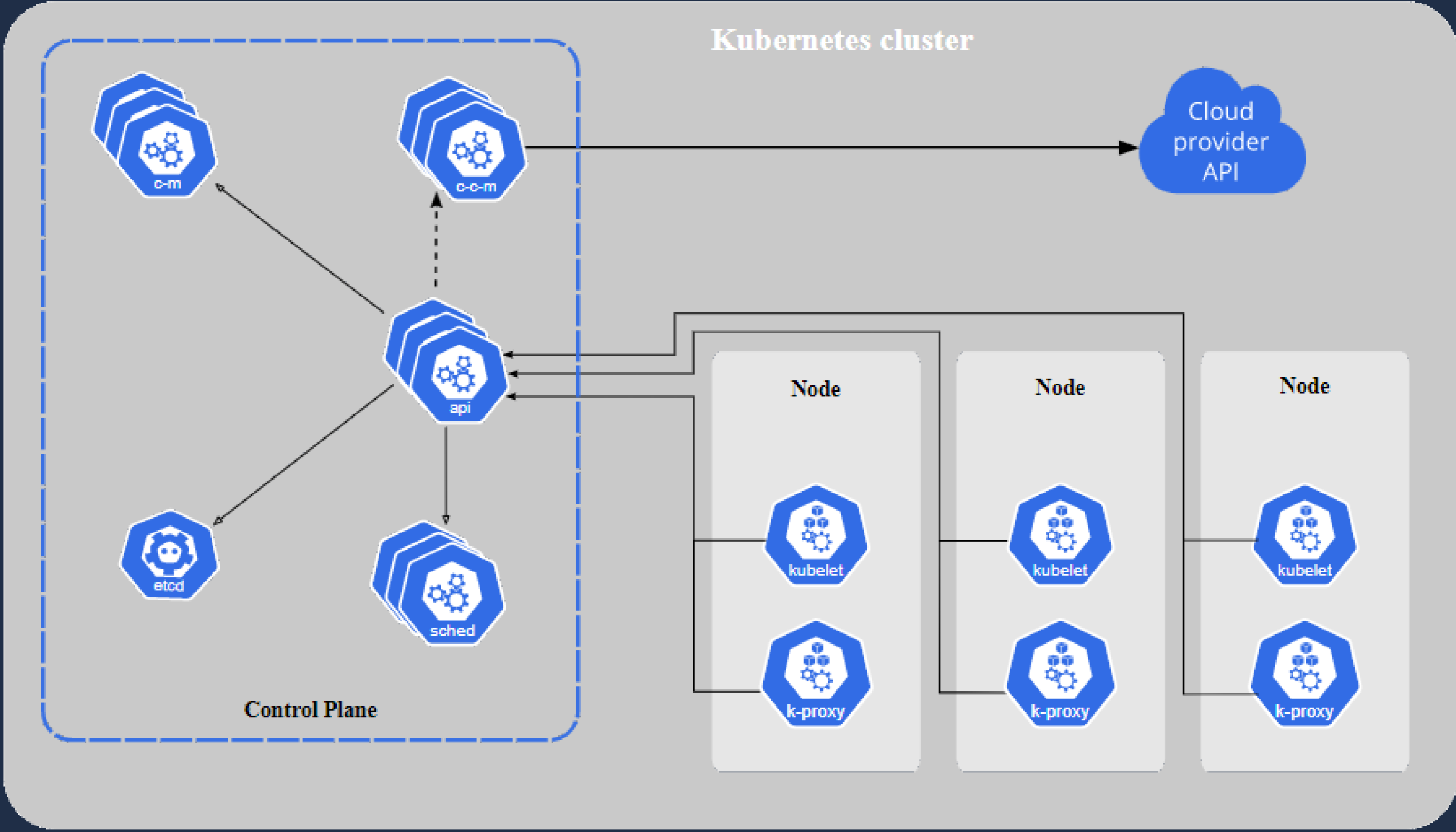
Run docker container locally:

```
docker run -d -p 8080:80 <image_name>
```


Agenda

1. Setup of local environment
2. Build and run docker container locally
- 3. Deploy a local Kubernetes cluster using k3d**
4. Basic operations to handle Pods and Deployments
5. Deploy a remote Kubernetes cluster on Google Cloud – part 1
6. Deploy a remote Kubernetes cluster on Google Cloud – part 2
7. Deploy a remote Kubernetes cluster leveraging terraform
8. Visualize Kubernetes workloads with VMware Octant

Introduction to Kubernetes



<https://kubernetes.io/docs/concepts/overview/components/>

Deploy a local Kubernetes cluster using K3D



Commands

Deploy local k3d cluster:

```
k3d cluster create local-cluster --servers 3 --agents 3 -p "8080:80@loadbalancer"
```

Deploy app to local cluster:

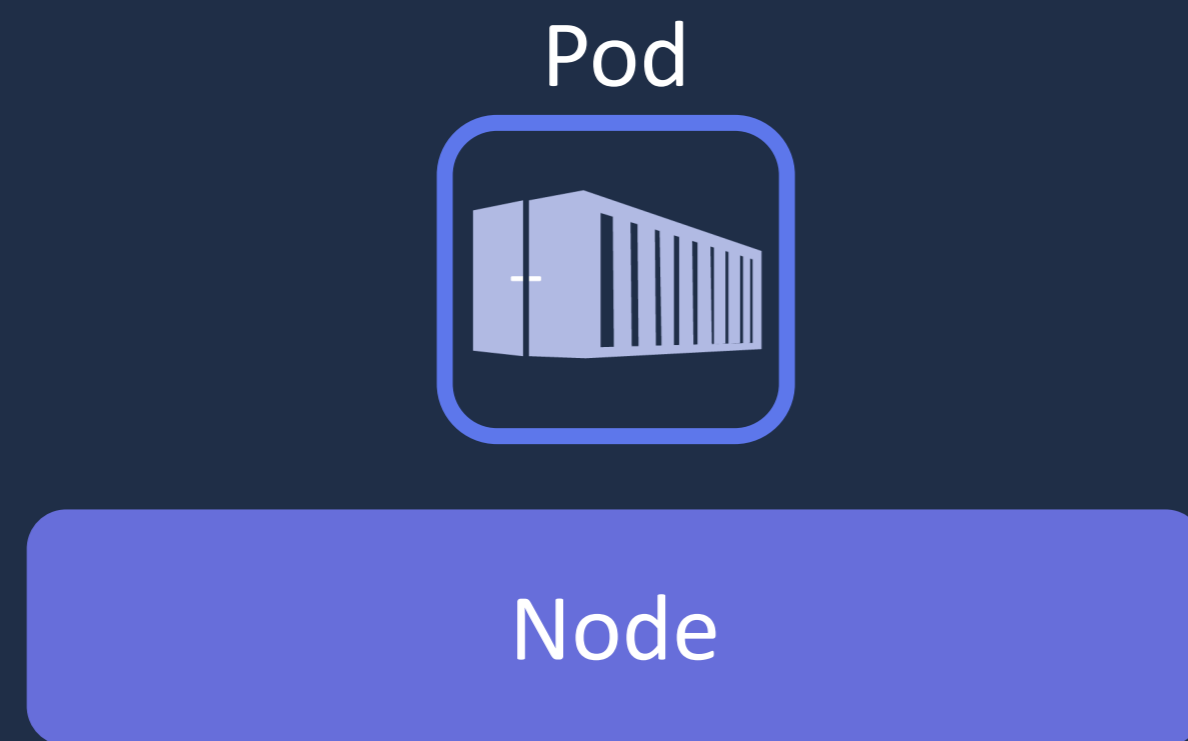
```
kubectl apply -f 1_3_local_deployment/deployment.yml
```

Get status of the deployed resources:

```
kubectl get deployments,replicasets,pods,services
```

Agenda

1. Setup of local environment
2. Build and run docker container locally
3. Deploy a local Kubernetes cluster using k3d
- 4. Basic operations to handle Pods and Deployments**
5. Deploy a remote Kubernetes cluster on Google Cloud – part 1
6. Deploy a remote Kubernetes cluster on Google Cloud – part 2
7. Deploy a remote Kubernetes cluster leveraging terraform
8. Visualize Kubernetes workloads with VMware Octant



A Pod is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers.

Basic operations to handle Pods and Deployments



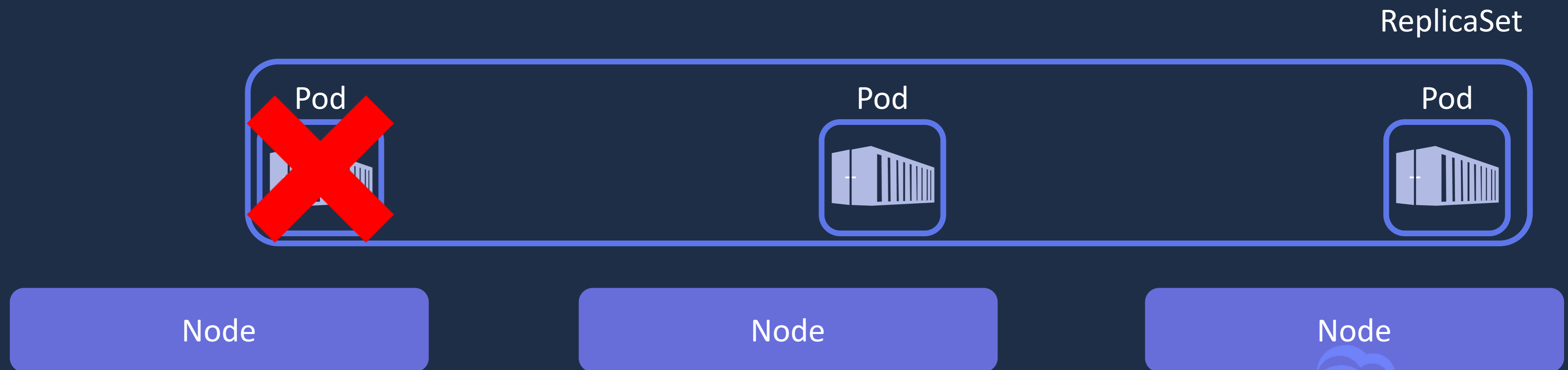
Pod

Node

Node

Node

Basic operations to handle Pods and Deployments



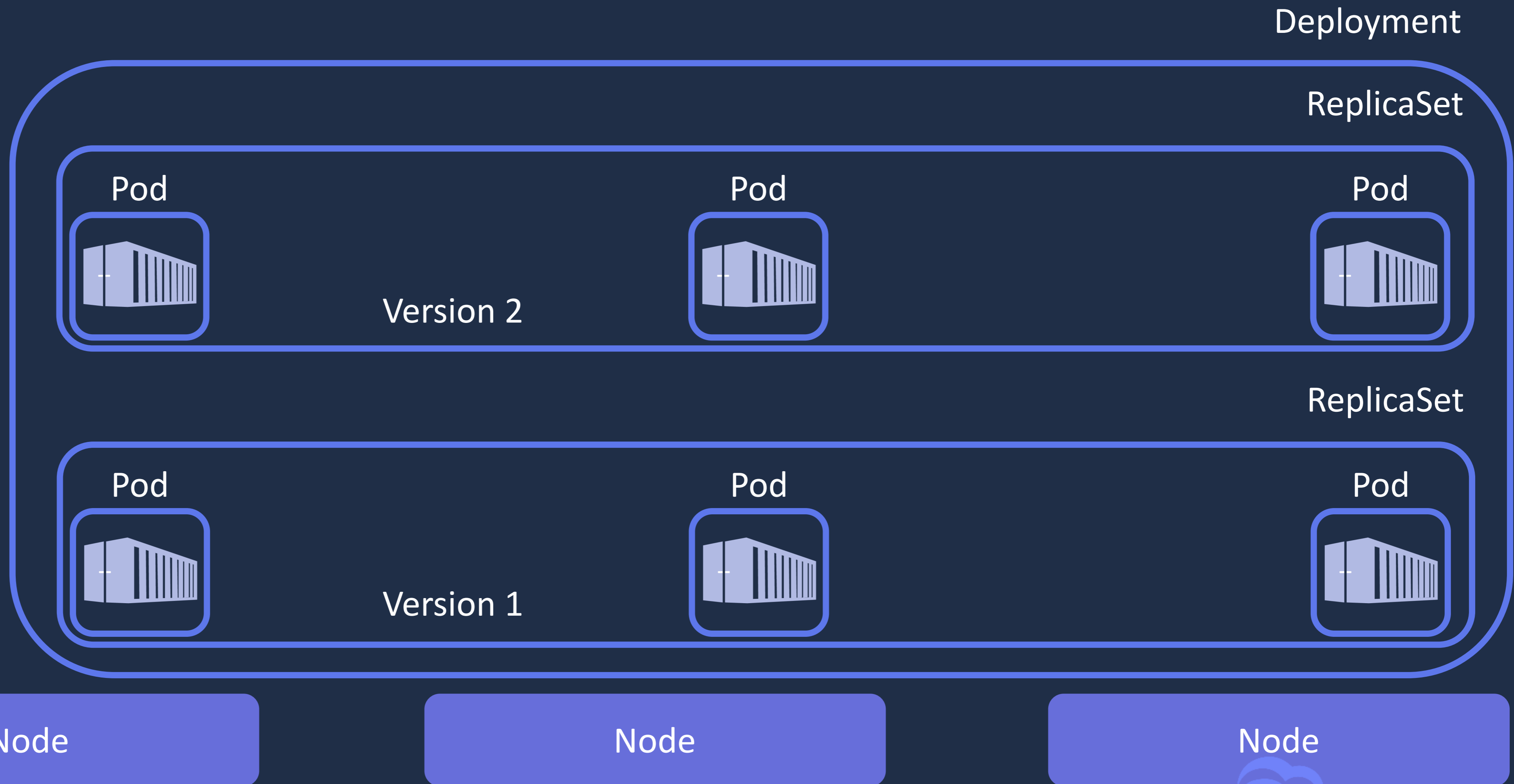
Basic operations to handle Pods and Deployments



Basic operations to handle Pods and Deployments



Service



Pod definition file

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-name
  namespace: default
  labels:
    app: pod-label
spec:
  containers:
  - name: demo-container
    image: gcr.io/neat-responder-349115/demo-app:sydney
    ports:
    - containerPort: 80
```

Replicaset definition file

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replicaset-name
  namespace: default
  labels:
    app: replicaset-label
spec:
  replicas: 3
  selector:
    matchLabels:
      app: pod-label
  template:
```

```
  metadata:
    labels:
      app: pod-label
  spec:
    containers:
      - name: demo-container
        image: gcr.io/neat-responder-349115/demo-app:sydney
```

Pod definition

Deployment definition file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-name
  namespace: default
  labels:
    app: deployment-label
spec:
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
    type: RollingUpdate
```

```
replicas: 3
selector:
  matchLabels:
    app: pod-label
template:
  metadata:
    labels:
      app: pod-label
spec:
```

Replicaset definition

```
containers:
- name: demo-container
  image: gcr.io/neat-responder-349115/demo-app:sydney
  imagePullPolicy: Always
  ports:
  - containerPort: 80
dnsPolicy: ClusterFirst
```

Pod definition

Service definition file

```
apiVersion: v1
kind: Service
metadata:
  name: service-name
  namespace: default
  labels:
    app: service-label
spec:
  ports:
  - name: 80-80
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: pod-label
  type: ClusterIP
```

← has to match Pod label

Basic operations to handle Pods and Deployments - links

29

<https://kubernetes.io/docs/concepts/workloads/pods/>

<https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller/>

<https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

<https://kubernetes.io/docs/concepts/services-networking/service/>

Commands

Scale deployment:

```
kubectl scale deployment demo-1 --replicas=10
```

Display rollout history:

```
kubectl rollout history deployment demo-1
```

Rollback deployment to previous state:

```
kubectl rollout undo deployment demo-1
```

Agenda

1. Setup of local environment
2. Build and run docker container locally
3. Deploy a local Kubernetes cluster using k3d
4. Basic operations to handle Pods and Deployments
- 5. Deploy a remote Kubernetes cluster on Google Cloud – part 1**
6. Deploy a remote Kubernetes cluster on Google Cloud – part 2
7. Deploy a remote Kubernetes cluster leveraging terraform
8. Visualize Kubernetes workloads with VMware Octant

Deploy a remote Kubernetes cluster on Google Cloud

32



Commands

Create Google Cloud Project:

```
gcloud projects create <project_id>
```

Create Google Cloud Project:

```
gcloud config set project <project_id>
```

Create Kubernetes cluster:

```
gcloud beta container clusters create <cluster_name>  
--zone <compute_zone>
```

Fetch credentials for running cluster, updates kubeconfig file:

```
gcloud container clusters get-credentials <cluster_name> --zone <compute_zone>
```

Agenda

1. Setup of local environment
2. Build and run docker container locally
3. Deploy a local Kubernetes cluster using k3d
4. Basic operations to handle Pods and Deployments
5. Deploy a remote Kubernetes cluster on Google Cloud – part 1
- 6. Deploy a remote Kubernetes cluster on Google Cloud – part 2**
7. Deploy a remote Kubernetes cluster leveraging terraform
8. Visualize Kubernetes workloads with VMware Octant

Deploy a remote Kubernetes cluster on Google Cloud



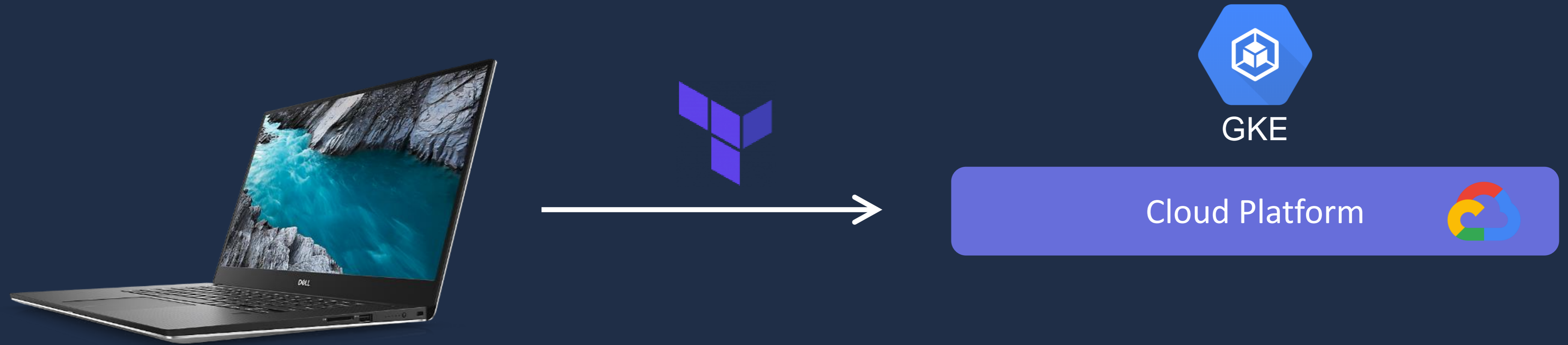
The screenshot shows the Google Cloud Platform (GCP) dashboard for a project named 'webapp'. The interface includes a top navigation bar with the GCP logo, project name, and search bar. A left sidebar contains navigation links for Home, API Manager, Billing, Cloud Launcher, Support, IAM & Admin, and a 'COMPUTE' section with links to App Engine, Compute Engine, Container Engine, Cloud Functions, and Networking. The main dashboard area is divided into several widgets: 'Project info' (showing project ID and name), 'APIs' (showing a request rate chart with no data), 'Google Cloud Platform status' (reporting all services normal), 'Resources' (showing 2 Cloud Storage buckets), 'Billing' (showing \$0.00 in charges), and 'Trace'. A 'CUSTOMIZE' button is located in the top right of the dashboard area.

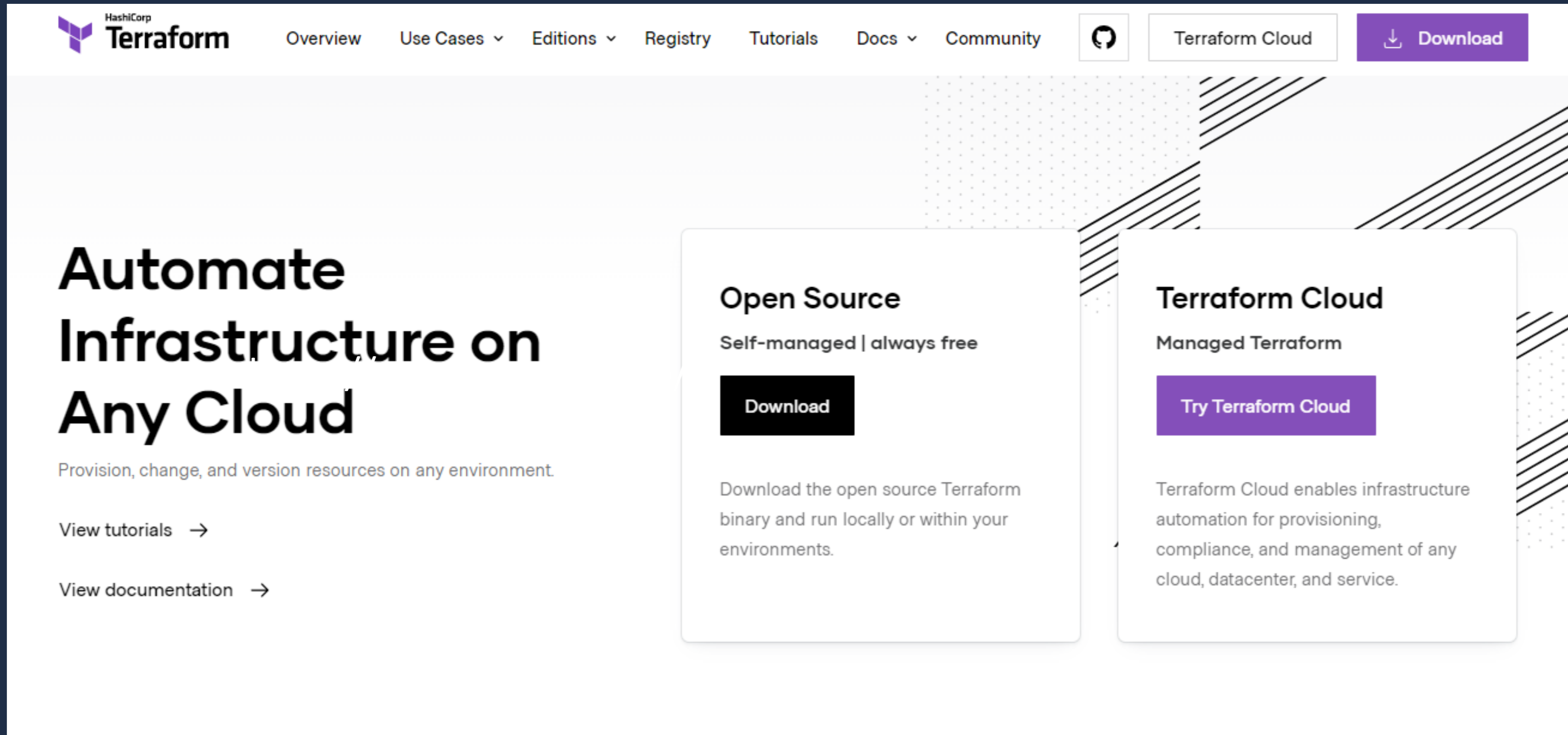
Agenda

1. Setup of local environment
2. Build and run docker container locally
3. Deploy a local Kubernetes cluster using k3d
4. Basic operations to handle Pods and Deployments
5. Deploy a remote Kubernetes cluster on Google Cloud – part 1
6. Deploy a remote Kubernetes cluster on Google Cloud – part 2
- 7. Deploy a remote Kubernetes cluster leveraging terraform**
8. Visualize Kubernetes workloads with VMware Octant

Deploy a remote Kubernetes cluster on Google Cloud

38





terraform.io

Commands

Initialize a new Terraform working directory:

```
terraform init
```

Generates a speculative execution plan:

```
terraform plan -var project_id=${PROJECT_ID}
```

Creates or updates infrastructure according to Terraform configuration:

```
terraform apply -var project_id=${PROJECT_ID}
```

Destroy Terraform managed infrastructure :

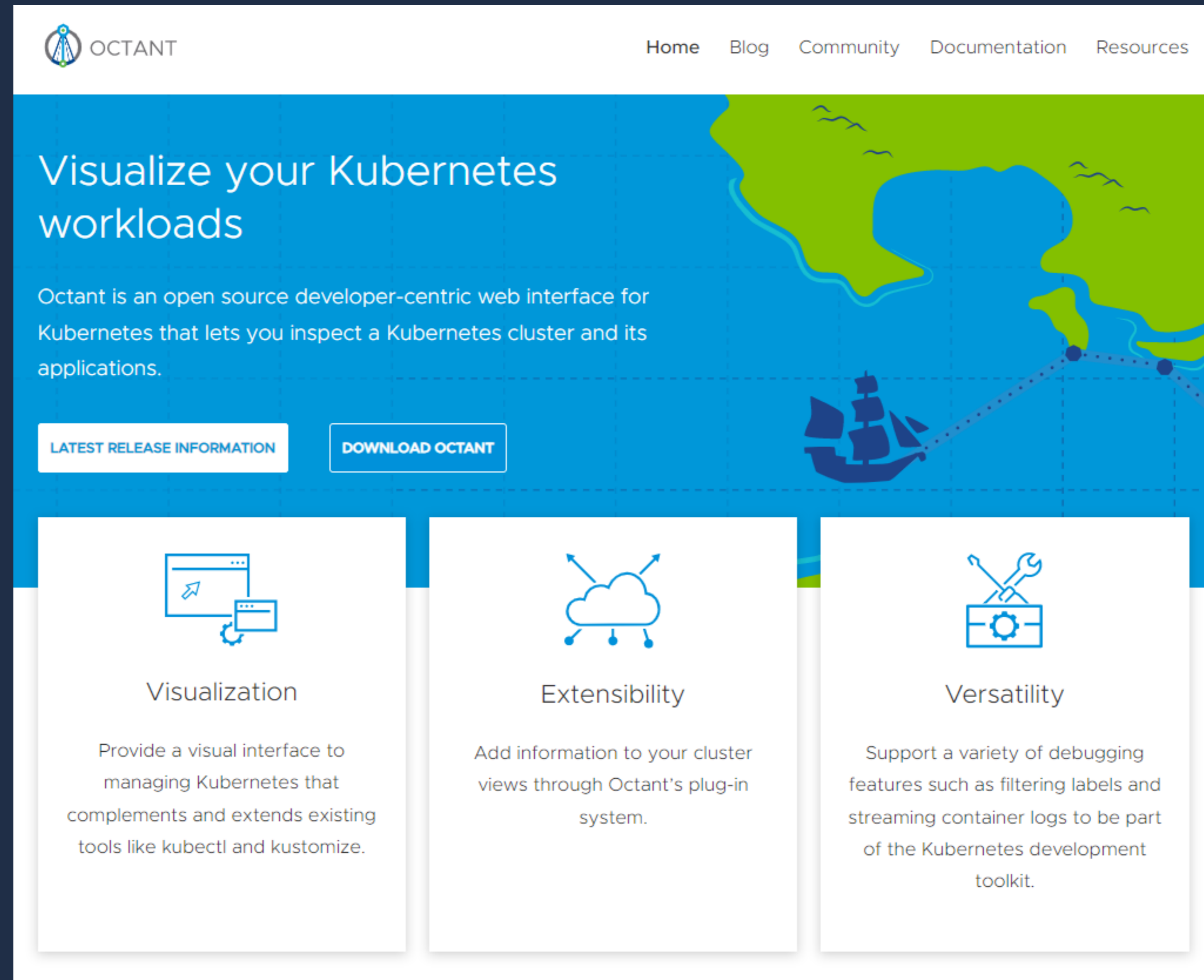
```
terraform destroy -var project_id=${PROJECT_ID}
```

Agenda

1. Setup of local environment
2. Build and run docker container locally
3. Deploy a local Kubernetes cluster using k3d
4. Basic operations to handle Pods and Deployments
5. Deploy a remote Kubernetes cluster on Google Cloud – part 1
6. Deploy a remote Kubernetes cluster on Google Cloud – part 2
7. Deploy a remote Kubernetes cluster leveraging terraform
8. **Visualize Kubernetes workloads with VMware Octant**

Visualize Kubernetes workloads with VMware Octant

42



The screenshot shows the Octant website homepage. At the top left is the Octant logo, and at the top right is a navigation menu with links for Home, Blog, Community, Documentation, and Resources. The main heading reads "Visualize your Kubernetes workloads". Below this is a sub-heading: "Octant is an open source developer-centric web interface for Kubernetes that lets you inspect a Kubernetes cluster and its applications." There are two buttons: "LATEST RELEASE INFORMATION" and "DOWNLOAD OCTANT". The page features three columns of features, each with an icon and a description:

- Visualization**: Provide a visual interface to managing Kubernetes that complements and extends existing tools like kubectl and kustomize.
- Extensibility**: Add information to your cluster views through Octant's plug-in system.
- Versatility**: Support a variety of debugging features such as filtering labels and streaming container logs to be part of the Kubernetes development toolkit.

octant.dev

Summary

Cluster name	Provider	Lading Zone	App version	# Replicas
local-cluster	K3D	Local PC	London	1
my-gke-cluster	Command line	Google Cloud	Sydney	3
my-gke-cluster-2	Cloud Console	Google Cloud	New York	10
my-terraform-cluster	Terraform	Google Cloud	Paris	30



CLOUD BITES