WiCyS 2023

#role-join WiCyS
#signin wicys

WiCyS
women in cybersecurity

ROCHESTER INSTITUTE OF TECHNOLOGY
STUDENT CHAPTER

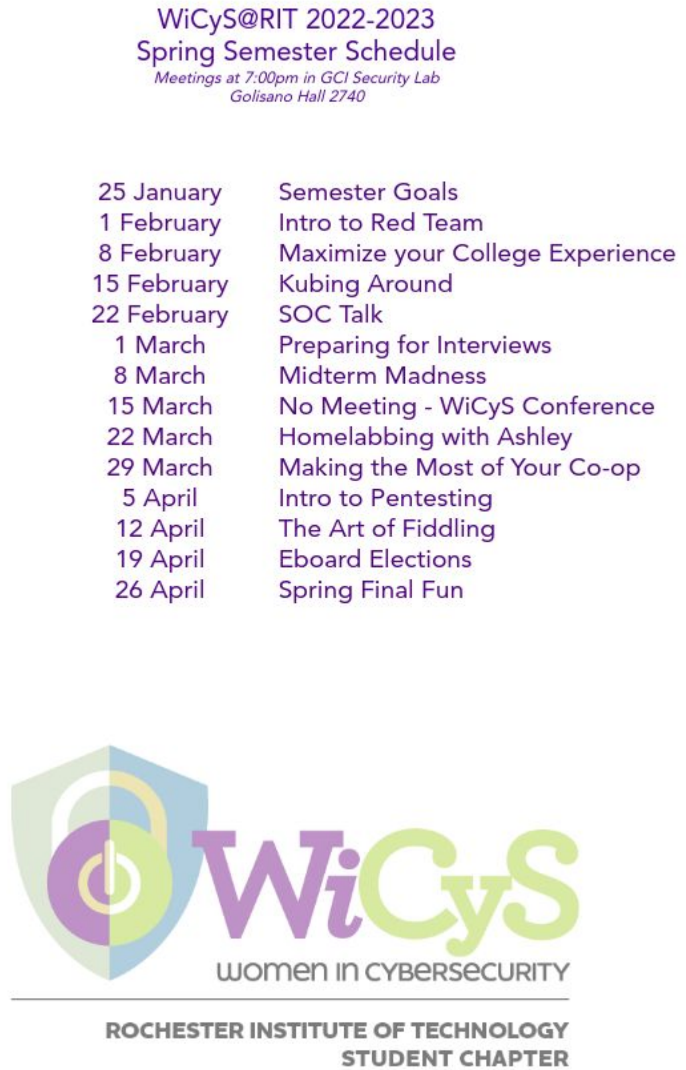
Schedule!

Weekly Meetings

Wednesdays at 7:00 PM in GCI Security Lab
Golisano Hall 2740

Official WiCyS Website

RIT WiCyS Website



WiCyS@RIT 2022-2023
Spring Semester Schedule
Meetings at 7:00pm in GCI Security Lab
Golisano Hall 2740

25 January	Semester Goals
1 February	Intro to Red Team
8 February	Maximize your College Experience
15 February	Kubing Around
22 February	SOC Talk
1 March	Preparing for Interviews
8 March	Midterm Madness
15 March	No Meeting - WiCyS Conference
22 March	Homelabbing with Ashley
29 March	Making the Most of Your Co-op
5 April	Intro to Pentesting
12 April	The Art of Fiddling
19 April	Eboard Elections
26 April	Spring Final Fun

WiCyS
women in cybersecurity
ROCHESTER INSTITUTE OF TECHNOLOGY
STUDENT CHAPTER

Follow our Socials!

@WiCySRIT

@WiCySRIT

Announcements

- **Sign up for Fire Talks!**
    - **https://wicysrit.wordpress.com/**

What are Containers?

- Takes the code and dependencies of an application and puts it into a standard unit of software that can be easily distributed
- Containers are very lightweight
  - Share the machine's OS system kernel
- Secure by default
- Portable

What is Kubernetes?

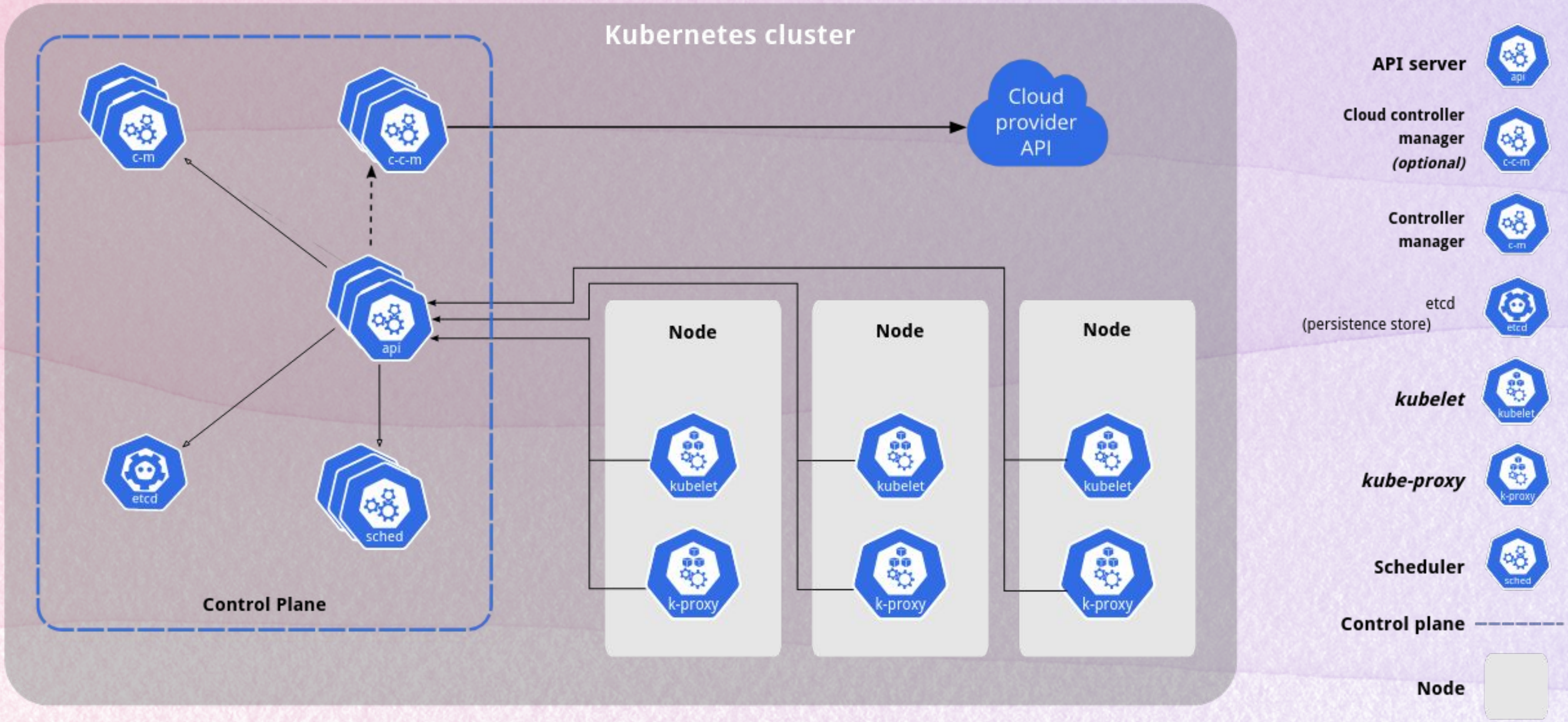
- Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications
- It groups containers that make up an application into logical units for easy management and discovery.

  Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-bread ideas and practices from the community

Kubernetes Features

- Automated rollouts and rollbacks
- Storage orchestration
- Secret and configuration management
- Service discovery and load balancing
- Self-healing
- Automatic resource-limiting
- Horizontal scaling
- Extendable

Kube Components



Kube Components

- Kubernetes is made up of control planes and workers
- Worker nodes
  - Host Pods (containers) that run your workloads
- Control plane
  - Manage worker nodes and pods in the cluster

Pods

- Group of one or more containers
  - Shared storage and network resources
  - Specification of how to run the containers
- Used in two main applications:
  - Running a single container
  - Running multiple containers that work together

Control Plane

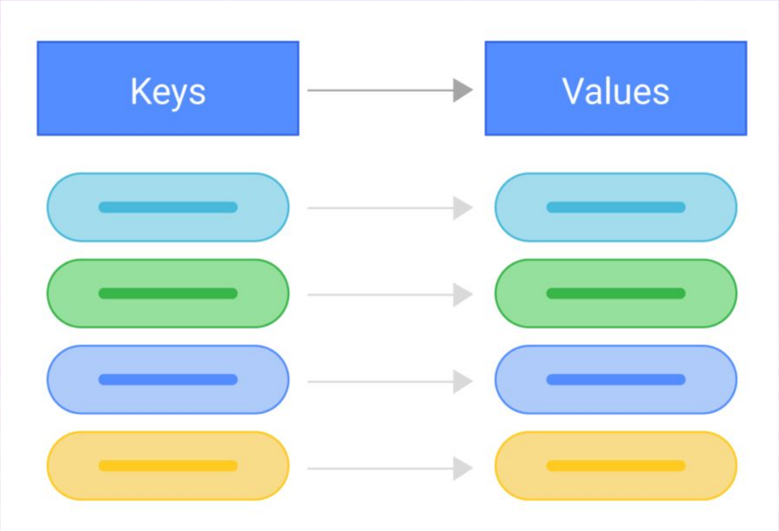
- Each component of the control plane is critical to the cluster running
  - Can be run on any node, but typically all components will be on the same node for simplicity
  - kube-apiserver
  - etcd
  - Kube-scheduler
  - Kube-controller-manager
  - cloud-controller-manager

Control Plane: API

- Exposes the Kubernetes API
  - Both end users and the cluster itself use the k8s API to perform tasks
    - Used to manipulate the state of all objects in k8s
    - Can use directly, but typically is done through *kubectl*
    - Many SDKS for interacting with the API
      - Python
      - Go
      - Rust 🤮

Control Plane: etcd

- Key value store
  - Think hash map or dictionary
- Stores all cluster data
  - Make sure you have a backup
- Highly available
  - Can handle 1000s of concurrent writes
- Distributed
  - Built to run on multiple instances



Control Plane: scheduler

- Takes unassigned pods and gives them to nodes to run on
- Decides based on multiple factors:
  - Individual/collective resource requirements
  - Hardware/software/policy constraints
  - Node affinity - pods have preferences too!
  - Data locality
  - Interference
  - Deadlines

Control Plane: K manager

- Runs processes that manage the entire cluster
  - Node controller
    - Responsible for noticing and responding when nodes go down
  - Job controller
    - Watches for jobs that are one-off tasks, and creates pods to run them
  - EndpointSlice controller
    - Provides links between services and pods
  - ServiceAccount controller - creates service account for namespaces

Control Plane: C manager

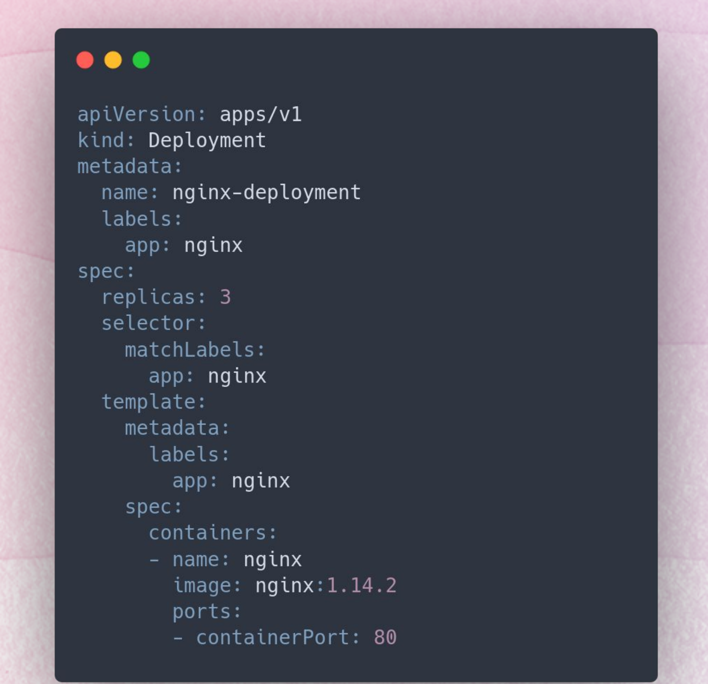
- Links your cluster to your cloud provider’s API
    - Google Kubernetes Engine - GKE
    - Azure Kubernetes Service - AKS
    - Amazon Elastic Kubernetes Service - EKS
    - DigitalOcean Kubernetes - DOKS
- Sets up routes for cloud infrastructure
- Manages cloud provider load balancers
- Checks provider if nodes have been deleted after they stop responding

All Nodes

- kubelet
  - Agent that runs on each node in the cluster
  - Makes sure that containers are running in a Pod
    - Takes in pod specs
      - Spits out containers
- Kube-proxy
  - Maintains networking rules in the cluster (allows Pods to talk to each other)
- Container runtime - software that runs the containers

Deployments

- Defines a specification in yaml or json of how pods should be ran



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

- Replicas
  - How many pods we want to run at once
- Image
  - The image hosted on a container registry we want to be on the pods
- Ports
  - Any ports we want the containers to expose

Services

- Now our pods are running, but we have no way of accessing them
    - Kubernetes services
- Services
    - Abstract way to expose an application running on Pods as a network service
    - Kubernetes gives pods their own IPs and has its own DNS scheme
    - Load balances automatically
- Why services?
    - Pods are ephemeral, many IP addresses, too much to keep track of

Why Should we Care?

- Kubernetes is used everywhere
    - Most big companies have adopted it
        - RITSEC Ops program uses it
- Cloud is the future
- Lots of $$$ in DevOps
- It’s cool!

Demo Time!