Microsoft Security

End to End Enterprise Security for Kubernetes

The rise of containers

Container adoption is booming

- **75%** of organizations globally will be running containerized workloads in production in 2025, up from around 30% in 2020

Increase in number and sophistication of attacks targeting containers and Kubernetes

- **94%** orgs experienced at least one security incident in Kubernetes during 2021

Extra focus on shift-left

- **78%** of security professionals have a DevSecOps initiative in either beginning or advanced stages

Container security is different

Ephemeral environment

- Applications are elastic, containers are short lived, spawn and re-size rapidly
- Container images are immutable, software updates require creation of new images

High demand for visibility and control

- Containers traffic flows are difficult to track with traditional tools
- Runtime environment includes rich data and controls, with different configuration layers

Depth of expertise needed

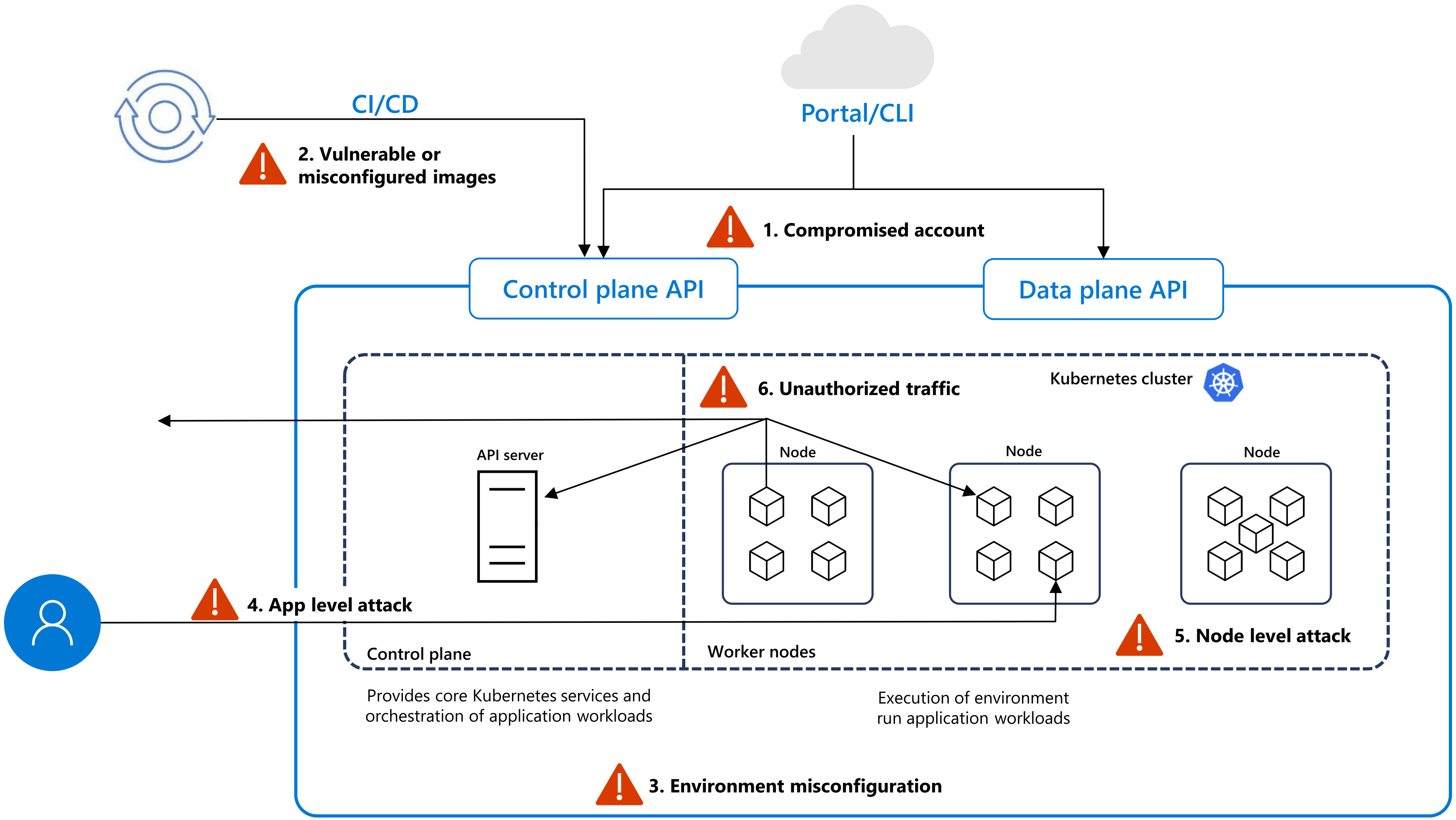
- A shortage of skilled labor
- Steep learning curves for open-source container tools and platforms

~44%
of containers live
less than 5 minutes!

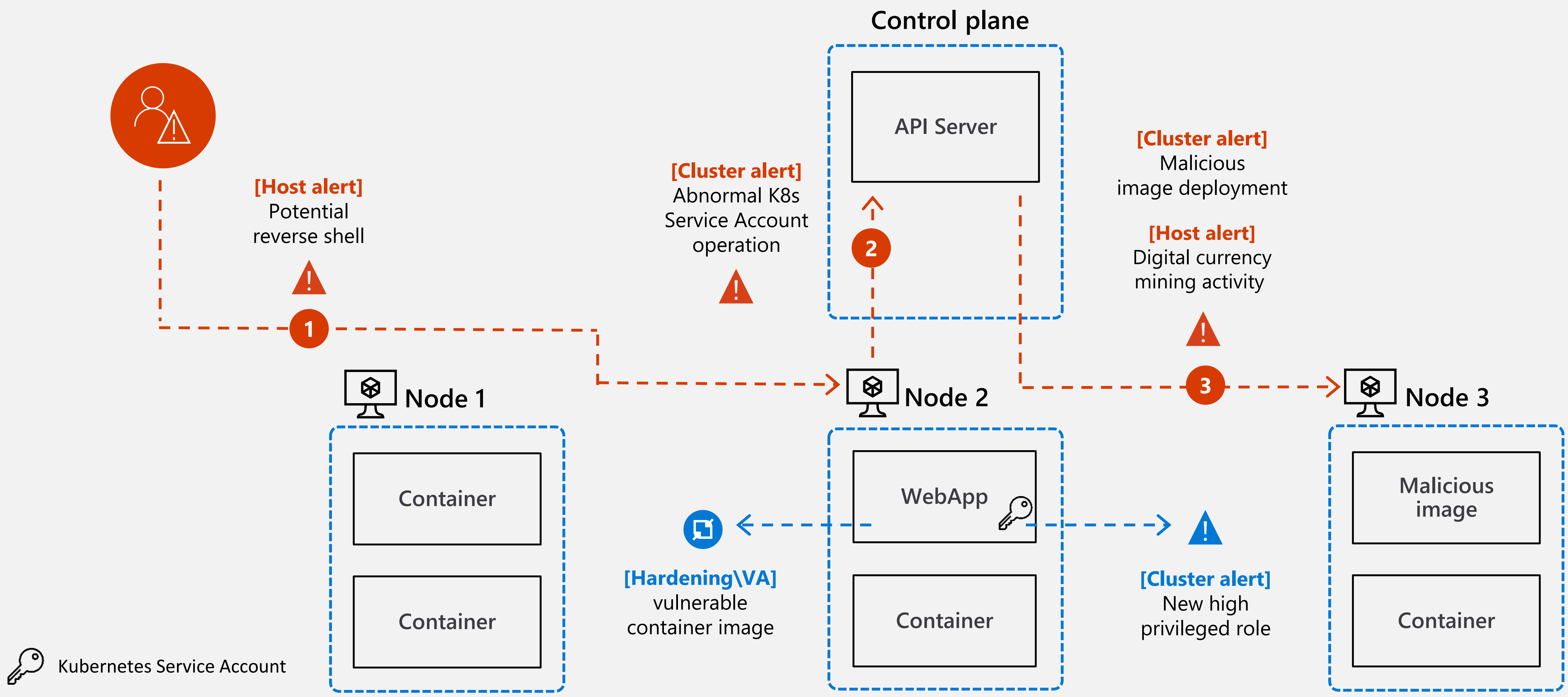
50%
of container images get
replaced in 1 week or less

Assure containerized environments are running as intended, including protection of infrastructure, software supply chain, runtime, and everything between

Managed Kubernetes threat factors

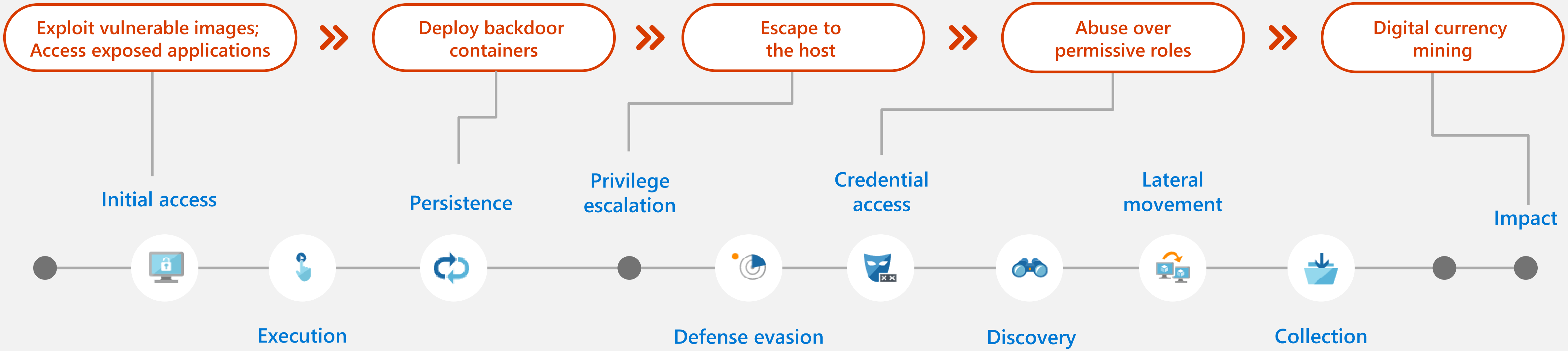


Attack flow


Control plane
API Server
[Host alert]
Potential
reverse shell
1
[Cluster alert]
Abnormal K8s
Service Account
operation
2
[Cluster alert]
Malicious
image deployment
[Host alert]
Digital currency
mining activity
3
Node 1
Container
Container
Node 2
WebApp
Container
Node 3
Malicious
image
Container
[Hardening\VA]
vulnerable
container image
[Cluster alert]
New high
privileged role
Kubernetes Service Account

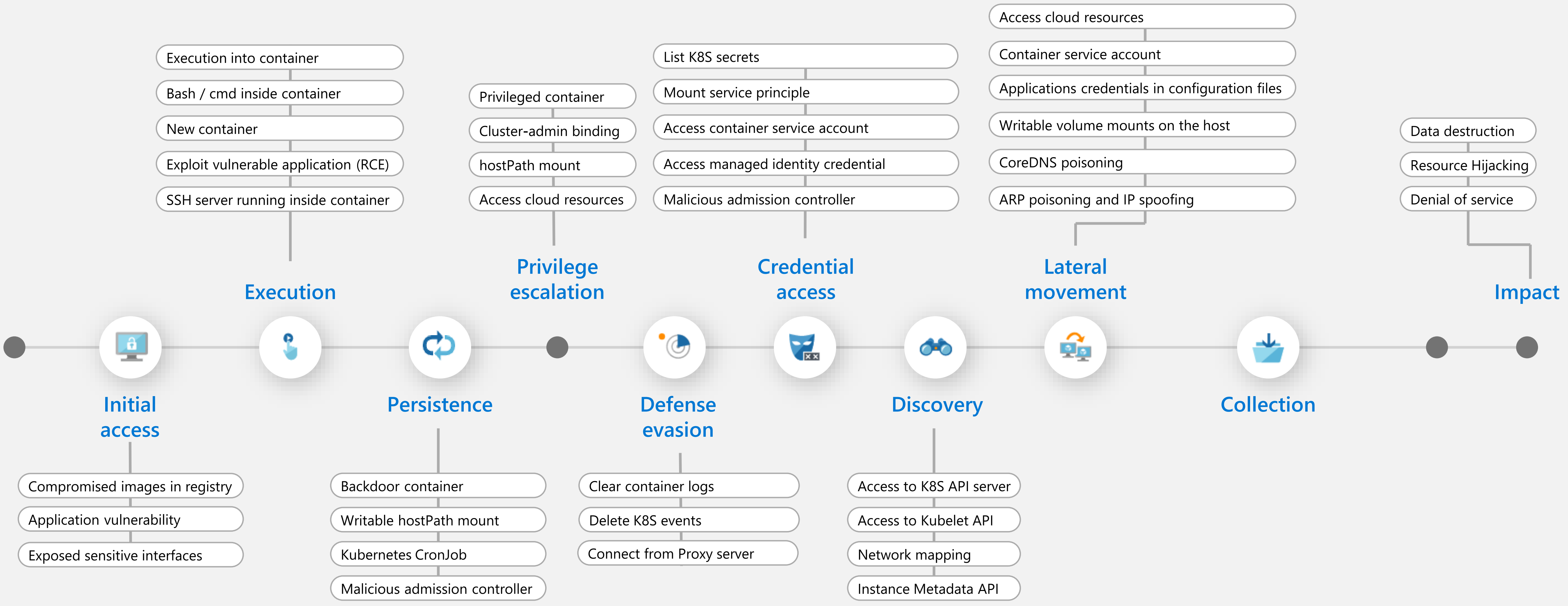
Common attack techniques

Observed by Microsoft researchers, as well as community published attacks



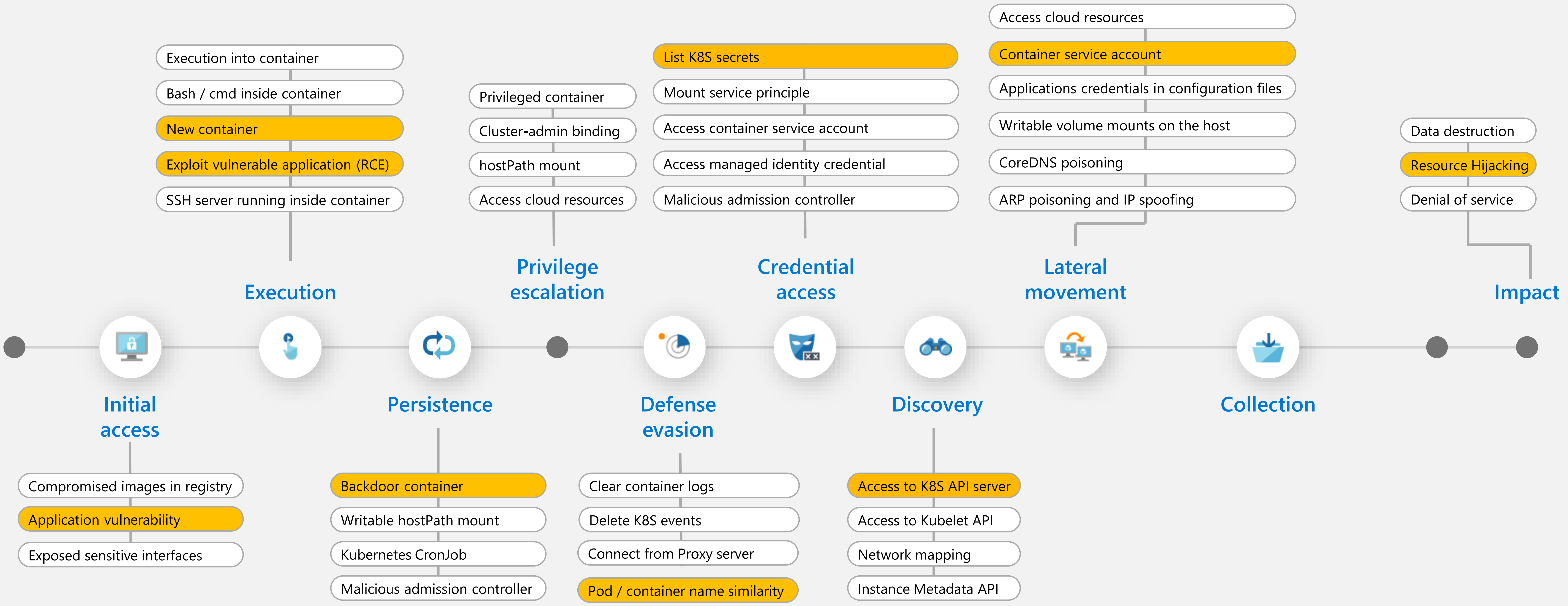
Threat matrix for Kubernetes

http://aka.ms/KubernetesThreatMatrix; Mitigate threats with the new threat matrix for Kubernetes



- Initial access
  - Compromised images in registry
  - Application vulnerability
  - Exposed sensitive interfaces
- Execution
  - Execution into container
  - Bash / cmd inside container
  - New container
  - Exploit vulnerable application (RCE)
  - SSH server running inside container
- Persistence
  - Backdoor container
  - Writable hostPath mount
  - Kubernetes CronJob
  - Malicious admission controller
- Privilege escalation
  - Privileged container
  - Cluster-admin binding
  - hostPath mount
  - Access cloud resources
- Defense evasion
  - Clear container logs
  - Delete K8S events
  - Connect from Proxy server
- Credential access
  - List K8S secrets
  - Mount service principle
  - Access container service account
  - Access managed identity credential
  - Malicious admission controller
- Discovery
  - Access to K8S API server
  - Access to Kubelet API
  - Network mapping
  - Instance Metadata API
- Lateral movement
  - Access cloud resources
  - Container service account
  - Applications credentials in configuration files
  - Writable volume mounts on the host
  - CoreDNS poisoning
  - ARP poisoning and IP spoofing
- Collection
- Impact
  - Data destruction
  - Resource Hijacking
  - Denial of service

Threat detections aligned to the K8s attack matrix



Initial access

- Compromised images in registry
- **Application vulnerability**
- Exposed sensitive interfaces

Execution

- Execution into container
- Bash / cmd inside container
- **New container**
- **Exploit vulnerable application (RCE)**
- SSH server running inside container

Persistence

- **Backdoor container**
- Writable hostPath mount
- Kubernetes CronJob
- Malicious admission controller

Privilege escalation

- Privileged container
- Cluster-admin binding
- hostPath mount
- Access cloud resources

Defense evasion

- Clear container logs
- Delete K8S events
- Connect from Proxy server
- **Pod / container name similarity**

Credential access

- **List K8S secrets**
- Mount service principle
- Access container service account
- Access managed identity credential
- Malicious admission controller

Discovery

- **Access to K8S API server**
- Access to Kubelet API
- Network mapping
- Instance Metadata API

Lateral movement

- Access cloud resources
- **Container service account**
- Applications credentials in configuration files
- Writable volume mounts on the host
- CoreDNS poisoning
- ARP poisoning and IP spoofing

Collection

Impact

- Data destruction
- **Resource Hijacking**
- Denial of service

Container security in Microsoft Cloud

Discover your container estate, identify risks and protect against breaches in the cloud

Security Posture management

- Discovery and inventory
- Attack path analysis
- Control plane assessments
- Date plane assessments
- Graph-based queries on the cloud security graph

Vulnerability management

- Agentless
- Zero configuration
- Daily scans/rescans
- OS and language packages
- Exploitability insights
- Support for ACR private links

Advanced threat detection

- Rich detection suite
- Leading threat intelligence
- Understand risk and context
- MITRE ATT&CK® mapping
- Automate response
- Export and SIEM integration

Deployment and monitoring

- Agentless capabilities
- Frictionless at scale deployment for agent-based capabilities
- Support for standard Kubernetes monitoring tools

Posture assessments

Discovery and inventory

Discover Kubernetes and container registry estate across SDLC, **seamlessly with no footprint on the workloads and runtime,** with a prioritized view of containerized assets

Attack path analysis

Prioritize and zoom into container vulnerabilities and posture risks that matter most

Control plane recommendations

Harden and audit according to Azure Security Benchmarks

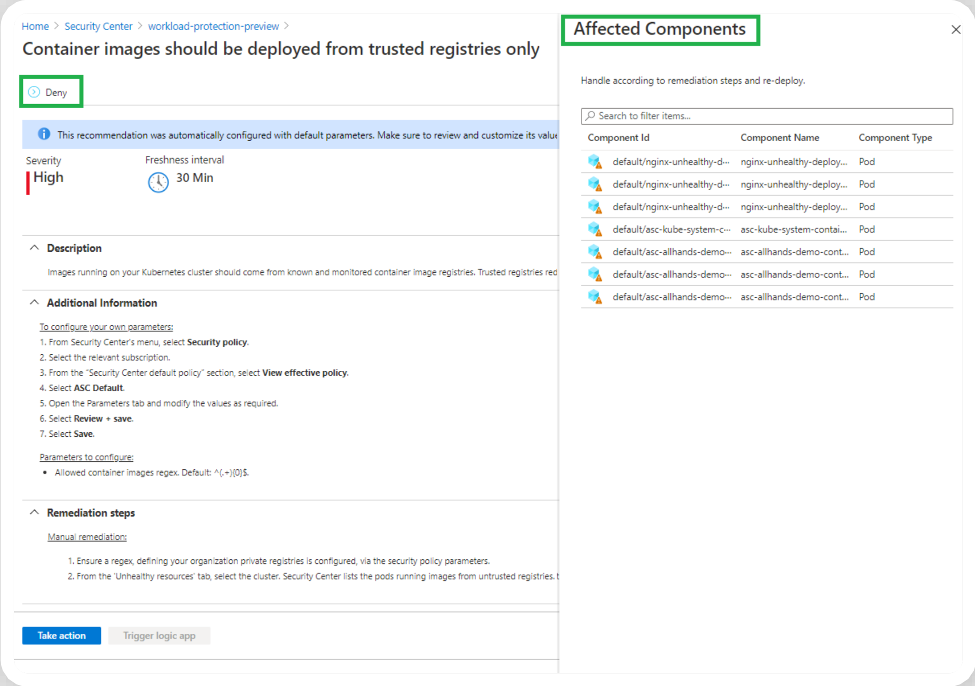
Follow Docker CIS benchmark on container nodes

Data plane recommendations

Audit or **enforce** Kubernetes security best practices with an admission control webhook

Graph-based queries

Uncover security insights in their cloud context, such as vulnerabilities, internet exposure, sensitive data, and more



Container images should be deployed from trusted registries only

This recommendation was automatically configured with default parameters. Make sure to review and customize its value

Severity: High

Freshness interval: 30 Min

Description

Images running on your Kubernetes cluster should come from known and monitored container image registries. Trusted registries red

Additional Information

To configure your own parameters:

1. From Security Center's menu, select **Security policy.**
2. Select the relevant subscription.
3. From the "Security Center default policy" section, select **View effective policy.**
4. Select **ASC Default.**
5. Open the Parameters tab and modify the values as required.
6. Select **Review + save.**
7. Select **Save.**

Parameters to configure:

- Allowed container images regex. Default: ^(.+){0}$.

Remediation steps

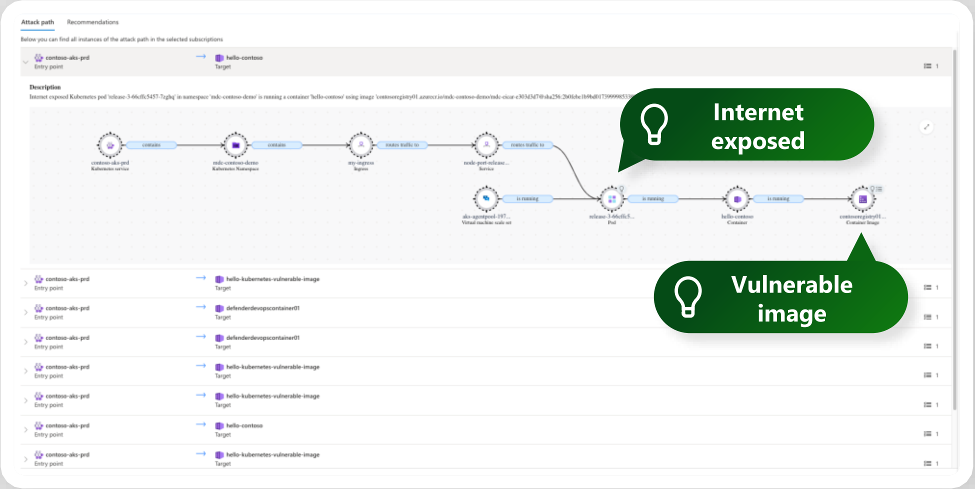
Manual remediation:

1. Ensure a regex, defining your organization private registries is configured, via the security policy parameters.
2. From the 'Unhealthy resources' tab, select the cluster. Security Center lists the pods running images from untrusted registries. t

Affected Components

Handle according to remediation steps and re-deploy.

Component Id	Component Name	Component Type
default/nginx-unhealthy-d···	nginx-unhealthy-deploy...	Pod
default/nginx-unhealthy-d···	nginx-unhealthy-deploy...	Pod
default/nginx-unhealthy-d···	nginx-unhealthy-deploy...	Pod
default/asc-kube-system-c···	asc-kube-system-contai...	Pod
default/asc-allhands-demo-···	asc-allhands-demo-cont...	Pod
default/asc-allhands-demo-···	asc-allhands-demo-cont...	Pod
default/asc-allhands-demo-···	asc-allhands-demo-cont...	Pod



Below you can find all instances of the attack path in the selected subscriptions

Entry point	Target
contoso-aks-prd	hello-contoso
contoso-aks-prd	hello-kubernetes-vulnerable-image
contoso-aks-prd	defenderdevopscontainer01
contoso-aks-prd	defenderdevopscontainer01
contoso-aks-prd	hello-kubernetes-vulnerable-image
contoso-aks-prd	hello-kubernetes-vulnerable-image
contoso-aks-prd	hello-contoso
contoso-aks-prd	hello-kubernetes-vulnerable-image

Description

Internet exposed Kubernetes pod 'release-3-66cffc5457-7zghq' in namespace 'mdc-contoso-demo' is running a container 'hello-contoso' using image 'contosoregistry01.azurecr.io/mdc-contoso-demo/mdc-eicar-e303d3d7@sha256:2b0febe1b9bd01739999853

Internet exposed

Vulnerable image

Vulnerability management

Agentless and zero configuration

Single enablement to scan all registry images and provide both registry and runtime VA without agent deployment

Continuous monitoring

Near real-time scan of new images and rescan every 24 hours

Protect across registry and runtime

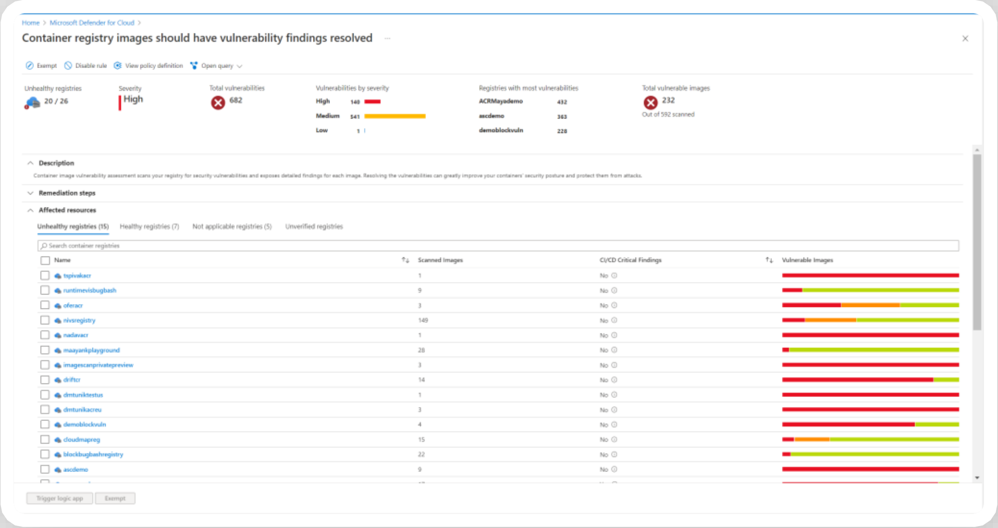
Images scanned at registry to provide VA for both registry and runtime

Full image coverage

Support both OS and programming languages packages

In-the-wild exploitability insights

Vulnerability enriched with real-world exploitability insights



Container registry images should have vulnerability findings resolved

Unhealthy registries 20 / 26

Severity High

Total vulnerabilities 682

Vulnerabilities by severity

High	140
Medium	541
Low	1

Registries with most vulnerabilities

ACRMayademo	432
ascdemo	363
demoblockvuln	228

Total vulnerable images 232
Out of 592 scanned

Description

Container image vulnerability assessment scans your registry for security vulnerabilities and exposes detailed findings for each image. Resolving the vulnerabilities can greatly improve your containers' security posture and protect them from attacks.

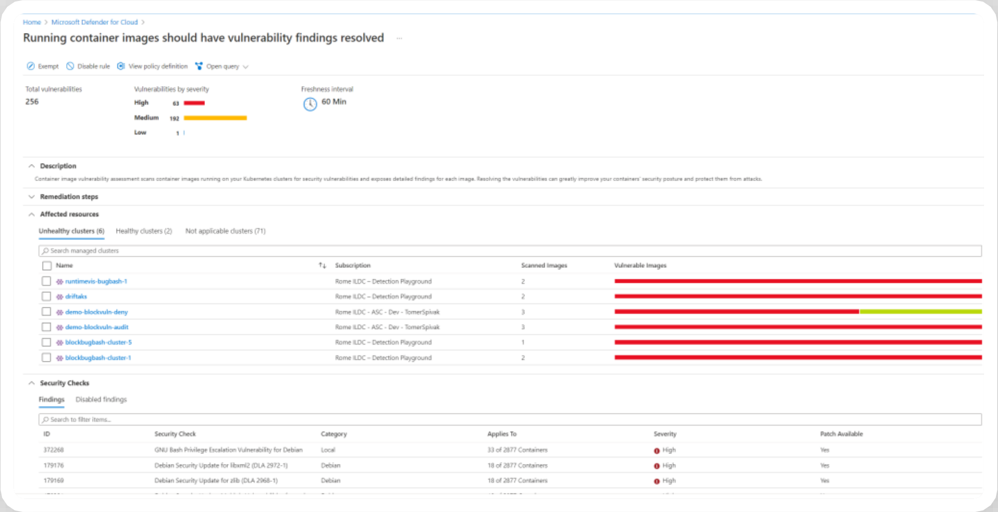
Remediation steps

Affected resources

Unhealthy registries (15) | Healthy registries (7) | Not applicable registries (5) | Unverified registries

Search container registries

Name	Scanned Images	CI/CD Critical Findings	Vulnerable Images
tspivakacr	1	No	
runtimevisbugbash	9	No	
oferacr	3	No	
nivsregistry	149	No	
nadavacr	1	No	
maayankplayground	28	No	
imagescanprivatepreview	3	No	
driftcr	14	No	
dmtuniktestus	1	No	
dmtunikacreu	3	No	
demoblockvuln	4	No	
cloudmapreg	15	No	
blockbugbashregistry	22	No	
ascdemo	9	No	



Running container images should have vulnerability findings resolved

Total vulnerabilities 256

Vulnerabilities by severity

High	63
Medium	192
Low	1

Freshness interval 60 Min

Description

Container image vulnerability assessment scans container images running on your Kubernetes clusters for security vulnerabilities and exposes detailed findings for each image. Resolving the vulnerabilities can greatly improve your containers' security posture and protect them from attacks.

Remediation steps

Affected resources

Unhealthy clusters (6) | Healthy clusters (2) | Not applicable clusters (71)

Search managed clusters

Name	Subscription	Scanned Images	Vulnerable Images
runtimevis-bugbash-1	Rome ILDC – Detection Playground	2	
driftaks	Rome ILDC – Detection Playground	2	
demo-blockvuln-deny	Rome ILDC - ASC - Dev - TomerSpivak	3	
demo-blockvuln-audit	Rome ILDC - ASC - Dev - TomerSpivak	3	
blockbugbash-cluster-5	Rome ILDC – Detection Playground	1	
blockbugbash-cluster-1	Rome ILDC – Detection Playground	2	

Security Checks

Findings | Disabled findings

Search to filter items...

ID	Security Check	Category	Applies To	Severity	Patch Available
372268	GNU Bash Privilege Escalation Vulnerability for Debian	Local	33 of 2877 Containers	High	Yes
179176	Debian Security Update for libxml2 (DLA 2972-1)	Debian	18 of 2877 Containers	High	Yes
179169	Debian Security Update for zlib (DLA 2968-1)	Debian	18 of 2877 Containers	High	Yes

Advanced threat detection

Rich detection suite

Control plane and workload level detections

Deterministic, AI, and anomaly-based alerts to identify threats

Leading threat intelligence

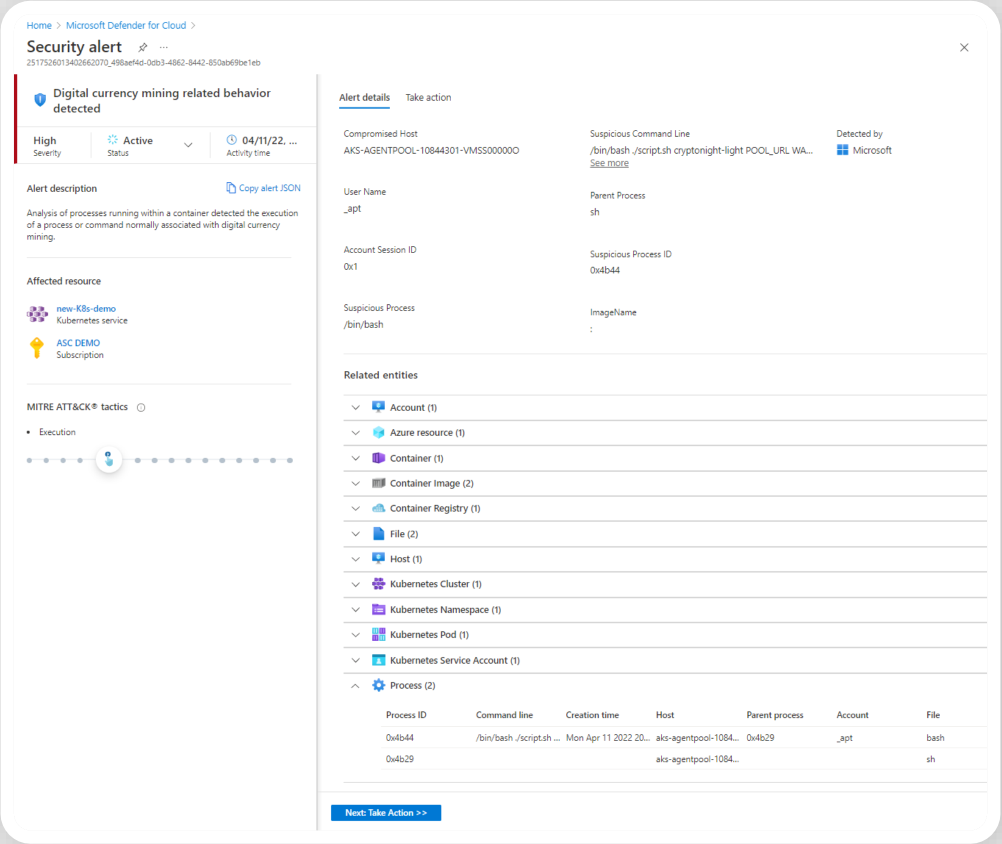
Microsoft’s global threat intelligence with honeypots networks, research malware feeds, in addition to memory forensic techniques to identify fileless attacks

Understand risk and context

Prioritized alerts mapped to MITRE ATT&CK® tactics to easily understand the Kubernetes context, effect across the attack lifecycle and to identify response action

Automate response

Automate actions with tools of your choice: SIEM integration, email notifications, workflow automations and continues export



Security alert

2517526013402662070_498aef4d-0db3-4862-8442-850ab69be1eb

Digital currency mining related behavior detected

High
Severity

Active
Status

04/11/22, ...
Activity time

Alert description

Analysis of processes running within a container detected the execution of a process or command normally associated with digital currency mining.

Affected resource

new-K8s-demo
Kubernetes service

ASC DEMO
Subscription

MITRE ATT&CK® tactics

- Execution

Compromised Host
AKS-AGENTPOOL-10844301-VMSS00000O

Suspicious Command Line
/bin/bash ./script.sh cryptonight-light POOL_URL WA...

Detected by
Microsoft

User Name
_apt

Parent Process
sh

Account Session ID
0x1

Suspicious Process ID
0x4b44

Suspicious Process
/bin/bash

ImageName
:

Related entities

- Account (1)
- Azure resource (1)
- Container (1)
- Container Image (2)
- Container Registry (1)
- File (2)
- Host (1)
- Kubernetes Cluster (1)
- Kubernetes Namespace (1)
- Kubernetes Pod (1)
- Kubernetes Service Account (1)
- Process (2)

Process ID	Command line	Creation time	Host	Parent process	Account	File
0x4b44	/bin/bash ./script.sh ...	Mon Apr 11 2022 20...	aks-agentpool-1084...	0x4b29	_apt	bash
0x4b29			aks-agentpool-1084...			sh

Bring Security during development

Shift left and integrate **security** in your development life cycle

Secure your dependencies

Know your environment

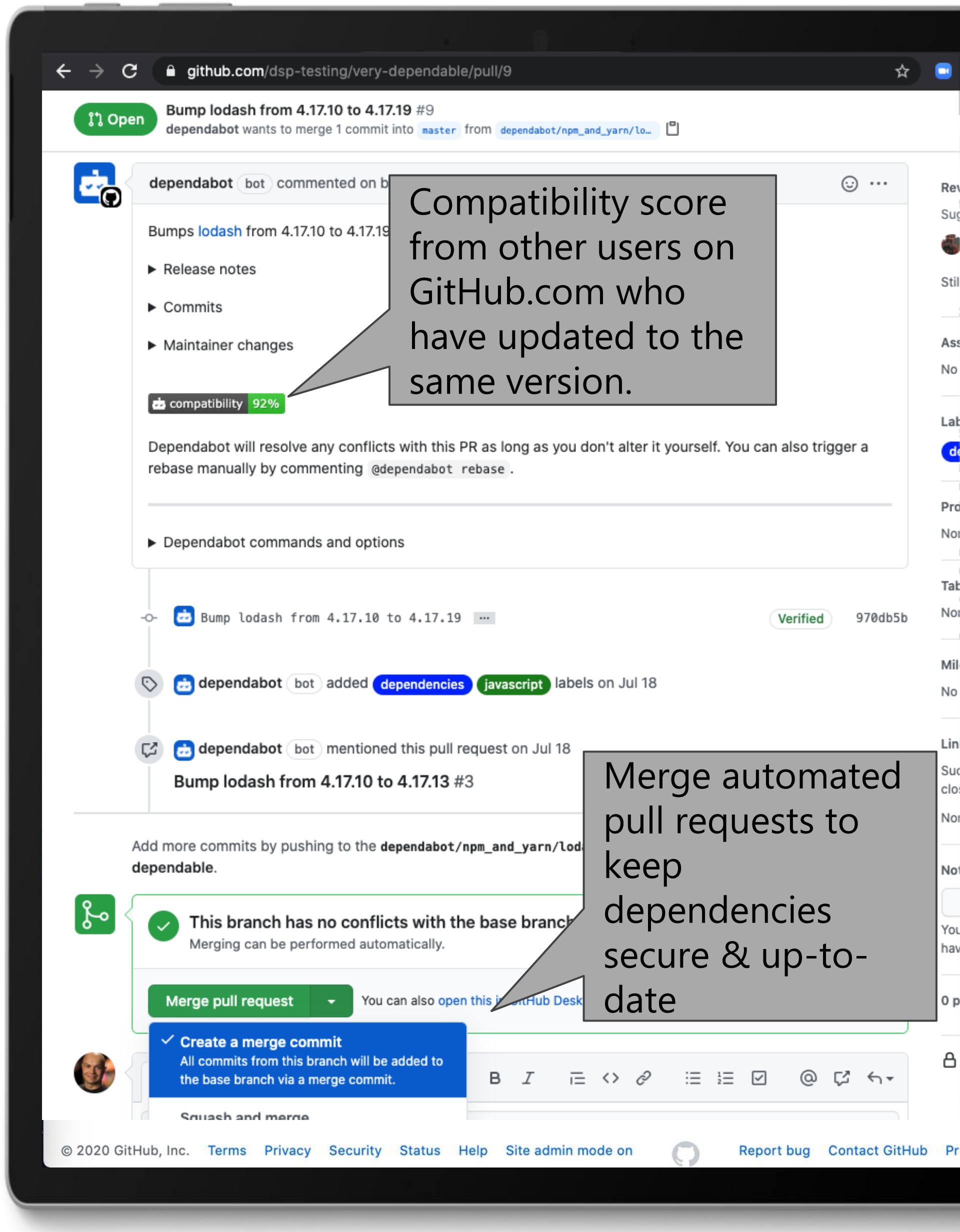
Understand the open source, inner source and commercial components in your projects, their licenses, and any known vulnerabilities in them

Manage your dependencies

Identify when dependencies are changing and ensure those changes do not introduce vulnerabilities or incompatible licenses

Respond fast to new vulnerability information

Get notified of new vulnerabilities as soon as they're discovered, and receive automated updates from Dependabot to patch your projects



github.com/dsp-testing/very-dependable/pull/9

Open **Bump lodash from 4.17.10 to 4.17.19** #9

dependabot wants to merge 1 commit into `master` from `dependabot/npm_and_yarn/lo…`

dependabot bot commented on b

Bumps lodash from 4.17.10 to 4.17.19

- ▶ Release notes
- ▶ Commits
- ▶ Maintainer changes

compatibility 92%

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`.

▶ Dependabot commands and options

`Bump lodash from 4.17.10 to 4.17.19` Verified `970db5b`

dependabot bot added dependencies javascript labels on Jul 18

dependabot bot mentioned this pull request on Jul 18

Bump lodash from 4.17.10 to 4.17.13 #3

Add more commits by pushing to the **`dependabot/npm_and_yarn/lod`** **dependable**.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request ▾ You can also open this i… GitHub Desk…

✓ **Create a merge commit**
All commits from this branch will be added to the base branch via a merge commit.

Squash and merge

© 2020 GitHub, Inc. Terms Privacy Security Status Help Site admin mode on Report bug Contact GitHub Pr

Compatibility score from other users on GitHub.com who have updated to the same version.

Merge automated pull requests to keep dependencies secure & up-to-date

Secure your code

Find hard-coded secrets in your code

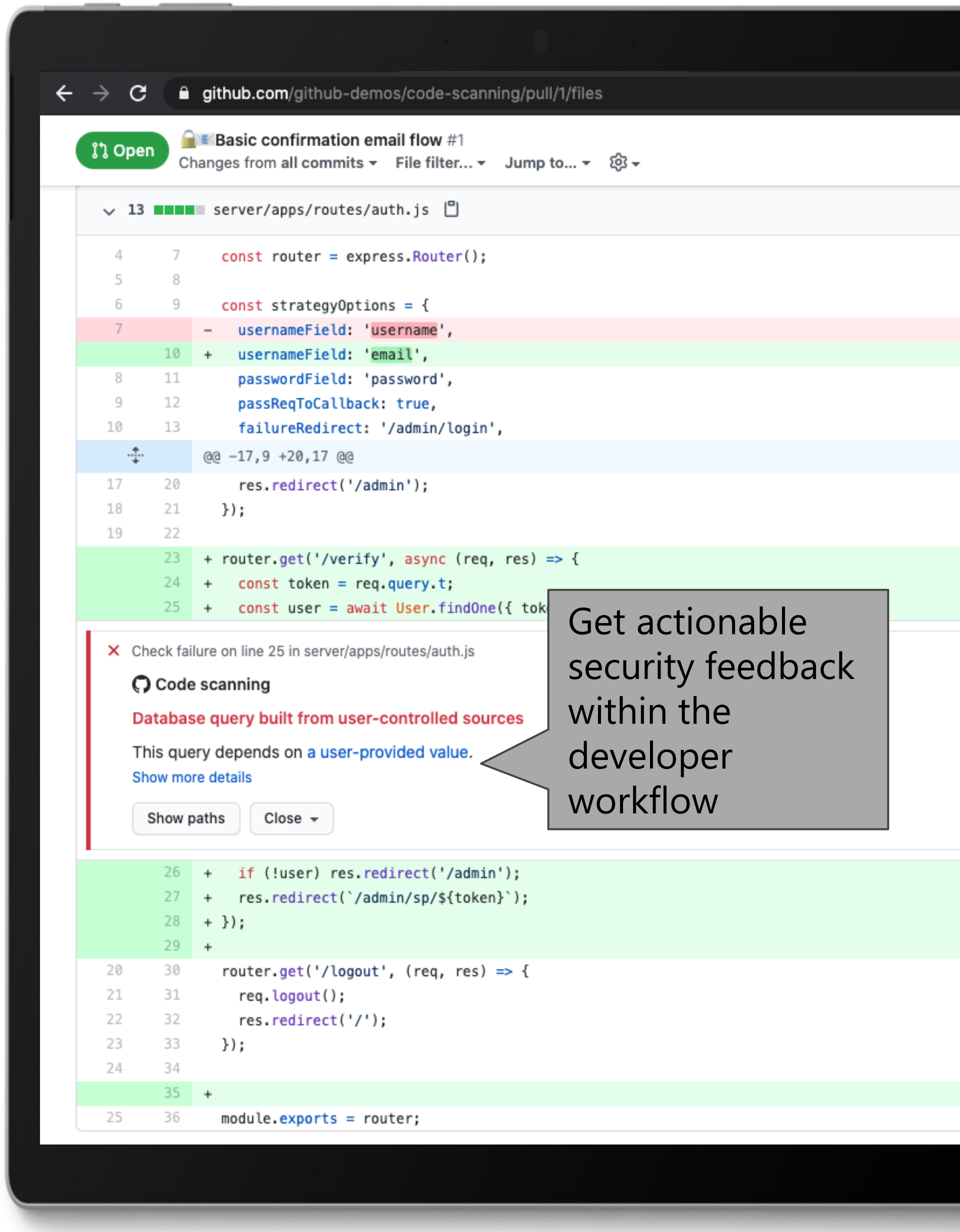
Scan your entire source code history for hard-coded credentials that present privilege escalation risks

Prevent new vulnerabilities from being introduced

Use GitHub code scanning and CodeQL to detect new vulnerabilities automatically. Scan every change to your code, and surface only new results

Global community for security

Take advantage of the hundreds of CodeQL queries written and open sourced by world-leading security teams



github.com/github-demos/code-scanning/pull/1/files

Basic confirmation email flow #1

13 server/apps/routes/auth.js

```
 4   7     const router = express.Router();
 5   8
 6   9     const strategyOptions = {
 7       -   usernameField: 'username',
    10   +   usernameField: 'email',
 8  11       passwordField: 'password',
 9  12       passReqToCallback: true,
10  13       failureRedirect: '/admin/login',
         @@ -17,9 +20,17 @@
17  20       res.redirect('/admin');
18  21     });
19  22
    23   + router.get('/verify', async (req, res) => {
    24   +   const token = req.query.t;
    25   +   const user = await User.findOne({ tok
```

Check failure on line 25 in server/apps/routes/auth.js

Code scanning

Database query built from user-controlled sources

This query depends on a user-provided value.

Show more details

```
    26   +   if (!user) res.redirect('/admin');
    27   +   res.redirect(`/admin/sp/${token}`);
    28   + });
    29   +
20  30     router.get('/logout', (req, res) => {
21  31       req.logout();
22  32       res.redirect('/');
23  33     });
24  34
    35   +
25  36     module.exports = router;
```

Get actionable security feedback within the developer workflow

Secure your workflow with GitHub Actions

GitHub Actions available now:

Orchestrate policy integration

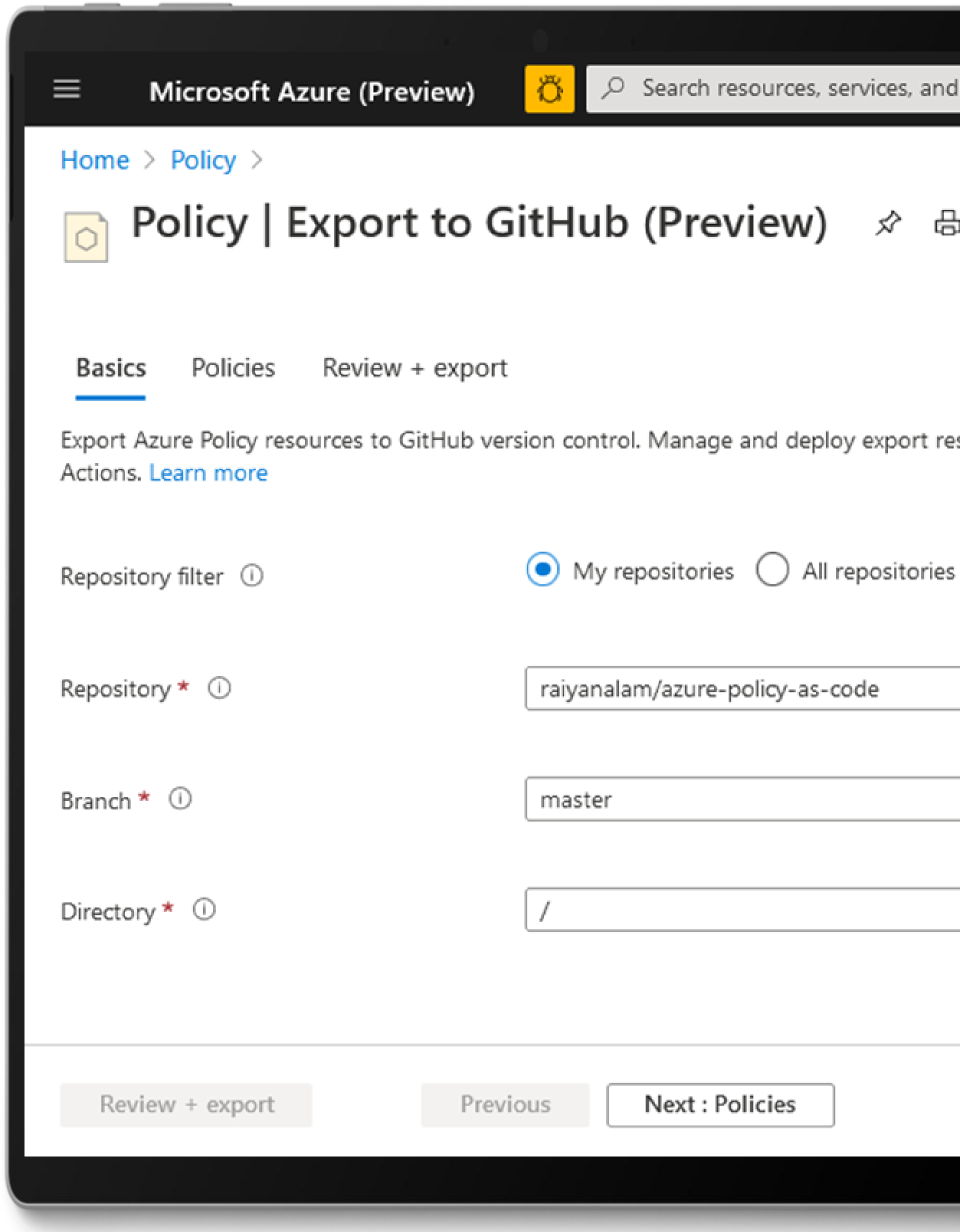
Easily manage Azure Policies “as code” from a GitHub repository in an orchestrated manner

Scan containers

Scan for common vulnerabilities in Docker images before pushing them to a container registry or deploying them to a containerized web app or Kubernetes cluster

Manage secrets using Azure Key Vault

Dynamically pull secrets from an Azure Key Vault instance for consumption in GitHub Action workflows



Policy | Export to GitHub (Preview)

Export Azure Policy resources to GitHub version control. Manage and deploy export res
Actions. Learn more

Repository filter ⓘ ● My repositories ○ All repositories

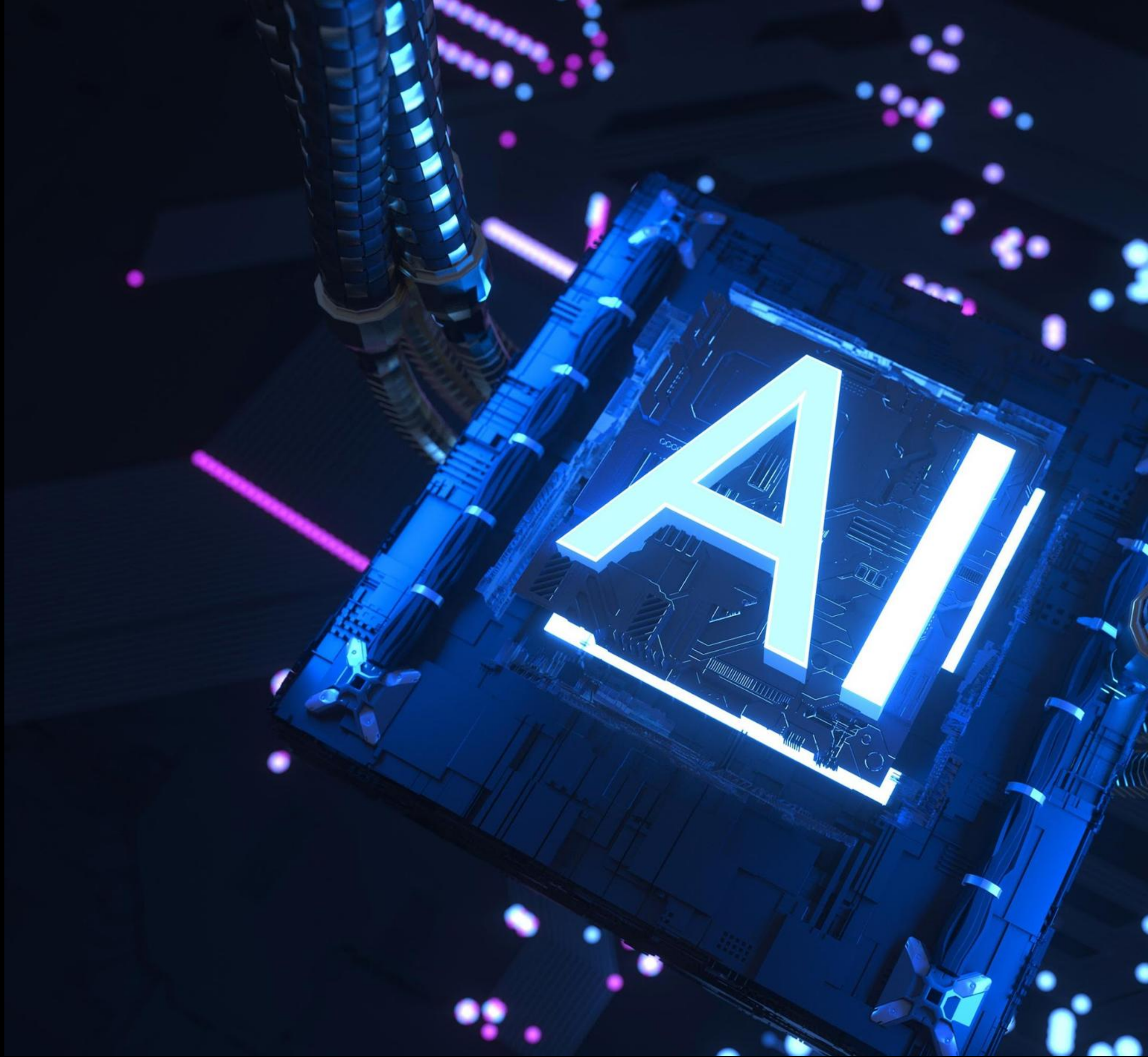
Repository * ⓘ raiyanalam/azure-policy-as-code

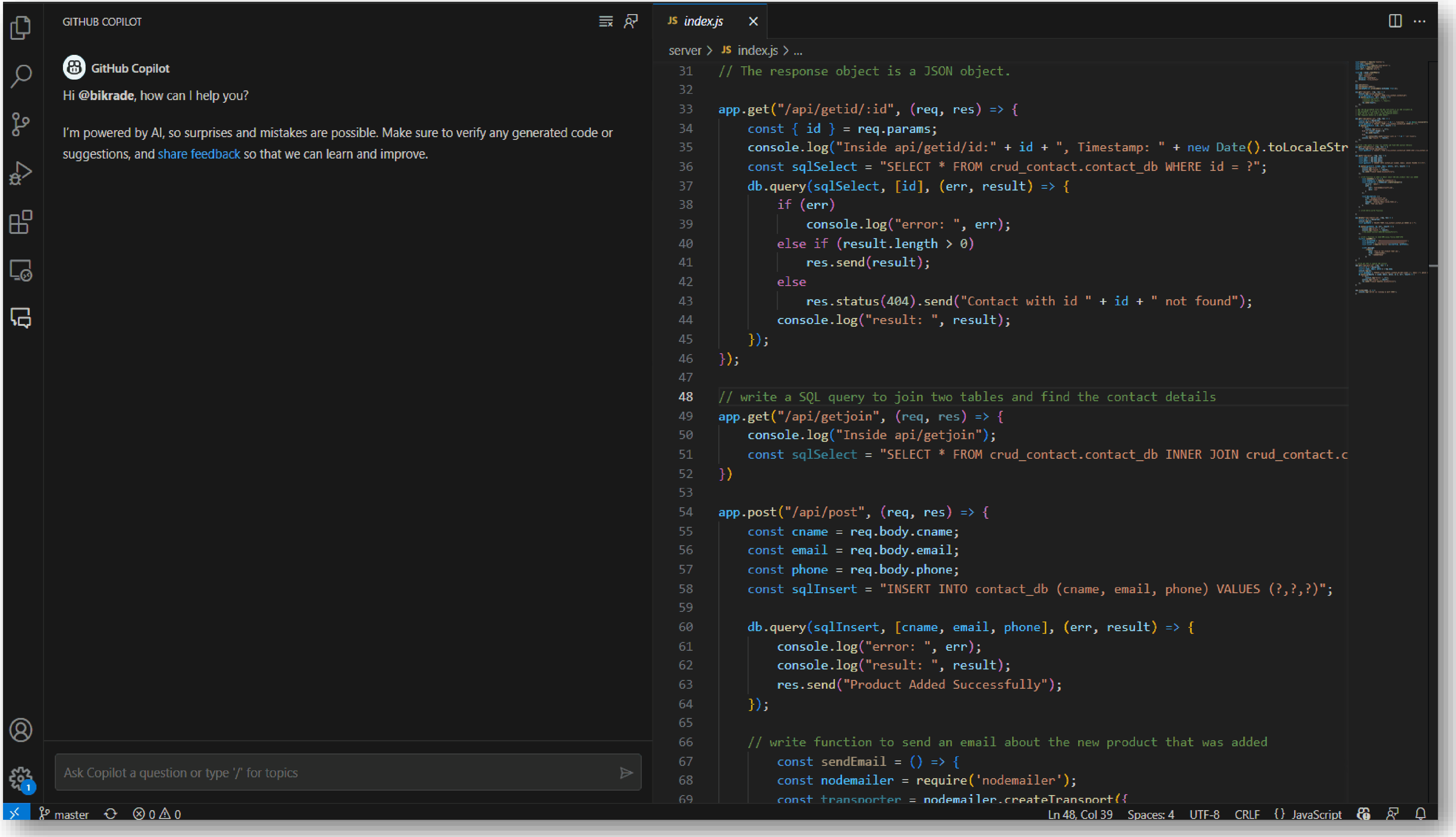
Branch * ⓘ master

Directory * ⓘ /

Demo

Bug Finding and Fixing





GitHub Copilot

Hi @bikrade, how can I help you?

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve.

```
// The response object is a JSON object.

app.get("/api/getid/:id", (req, res) => {
    const { id } = req.params;
    console.log("Inside api/getid/id:" + id + ", Timestamp: " + new Date().toLocaleStr
    const sqlSelect = "SELECT * FROM crud_contact.contact_db WHERE id = ?";
    db.query(sqlSelect, [id], (err, result) => {
        if (err)
            console.log("error: ", err);
        else if (result.length > 0)
            res.send(result);
        else
            res.status(404).send("Contact with id " + id + " not found");
        console.log("result: ", result);
    });
});

// write a SQL query to join two tables and find the contact details
app.get("/api/getjoin", (req, res) => {
    console.log("Inside api/getjoin");
    const sqlSelect = "SELECT * FROM crud_contact.contact_db INNER JOIN crud_contact.c
})

app.post("/api/post", (req, res) => {
    const cname = req.body.cname;
    const email = req.body.email;
    const phone = req.body.phone;
    const sqlInsert = "INSERT INTO contact_db (cname, email, phone) VALUES (?,?,?)";

    db.query(sqlInsert, [cname, email, phone], (err, result) => {
        console.log("error: ", err);
        console.log("result: ", result);
        res.send("Product Added Successfully");
    });

    // write function to send an email about the new product that was added
        const sendEmail = () => {
        const nodemailer = require('nodemailer');
        const transporter = nodemailer.createTransport({
```

Microsoft Security

Thank you