kloia

Secure Your Code :
Implement DevSecOps in Azure
21 Feb 2021

@srkbngl

Who am I?

Serkan Bingöl

Cloud & DevOps Consultant @kloia
MCSD / MCT / AAI / AWS Certified

Blog : medium.com/@serkanbingoll
Tweet : @srkbngl
LinkedIn : /in/sbingol
GitHub : github.com/serkanbingol

Agenda

- What is Dev**{Sec}**Ops ?
- DevSecOps in GitHub & Demos
- DevSecOps in Azure
- 3rd Party DevSecOps Tools
- DEMO : Implement Security in Azure DevOps CI/CD
- Q & A

What is Dev{Sec}Ops ?

DevOps is the union of people, process, and technology to enable continuous delivery of value to your end users.

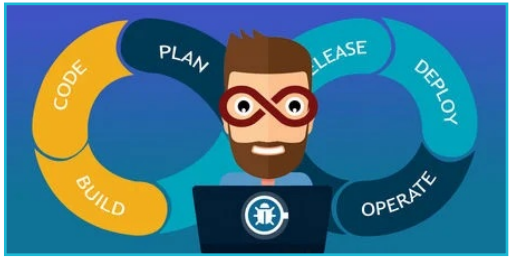


image from https://stackify.com/devops-engineer-starter-guide/

Automation + Security

DevSecOps is a cultural movement that furthers the movements of Agile and DevOps into Security

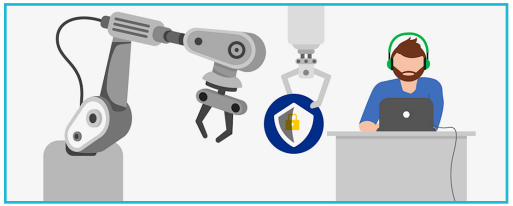


image from https://www.recordedfuture.com/

What is Dev{Sec}Ops ?

DEV : OPS :SEC Ratio - 100:10:1

“The ratio of engineers in Development, Operations, and Infosec in a typical technology organization is 100:10:1. When Infosec is that outnumbered, without automation and integrating information security into the daily work of Dev and Ops, Infosec can only do compliance checking, which is the opposite of security engineering—and besides, it also makes everyone hate us.”

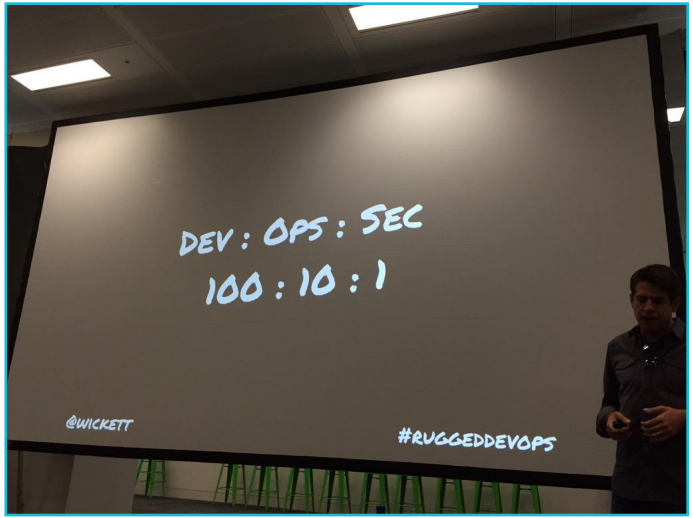
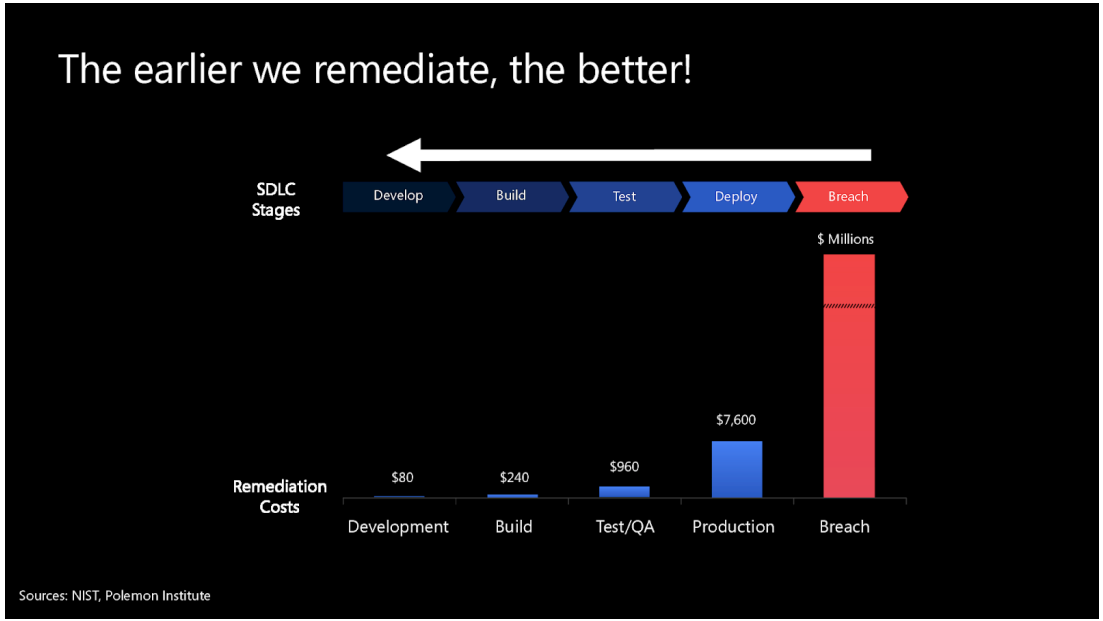


image from https://twitter.com/joshcorman/status/644153295464960000

What is Dev{Sec}Ops ?

Discover vulnerability during the development.



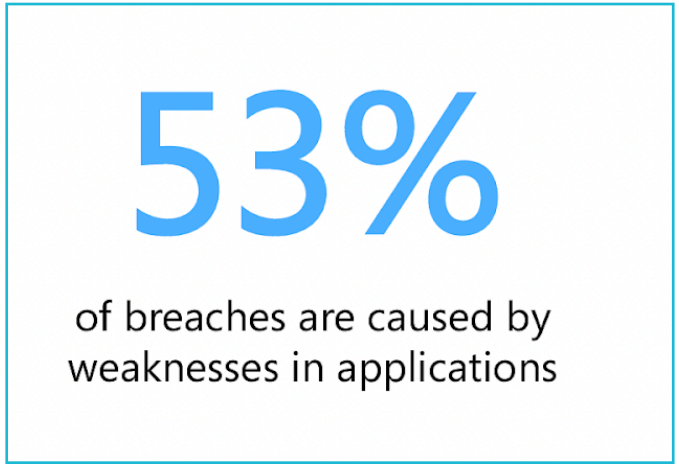
* Slide 9 : https://speaking.sasharosenbaum.com/EI8loj/slides

What is Dev{Sec}Ops ?

kloia

More we code , more we need security.

Insecure code causes breaches Source: 2019 Data Breach Investigations Report, Verizon 53% of breaches are caused by weaknesses in applications.



Report link :
https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf

What is Dev{Sec}Ops ?

Security Patterns : Old Path Vs. New Path

- Embrace Secrecy
- Just Pass Audit!
- Enforce Stability
- Build a Wall
- Slow Validation
- Certainly Testing
- Test When DONE !
- Process Driven

image from https://wellbeingeconomy.org/

- Create Feedback Loops
- Compliance adds Value
- Create Chaos
- Zero Trust Networks
- Fast and Non-Blocking
- Adversity Testing
- Shift Left
- The Paved Road

What is Dev{Sec}Ops ?

Azure Patterns: Infrastructure Planning

- ✘ Subscription **per** environment
- ✘ Apply **policies** to control transparently and proactively
- ✘ Security Center ON from day **1**
- ✘ Infrastructure as code. **Period**
- ✘ Production **ONLY** for CI/CD pipelines
- ✘ CI/CD pipeline is a **heart** of security

image from https://bridgecrew.io/blog/infrastructure-as-code-security-101/

What is Dev{Sec}Ops ?

kloia

Azure Patterns: Watch Every Step

- Threat modeling
- IDE Security plugins
- Pre-commit hooks
- Peer review

- Static code analysis
- Security unit tests
- Container Security
- Dependency management

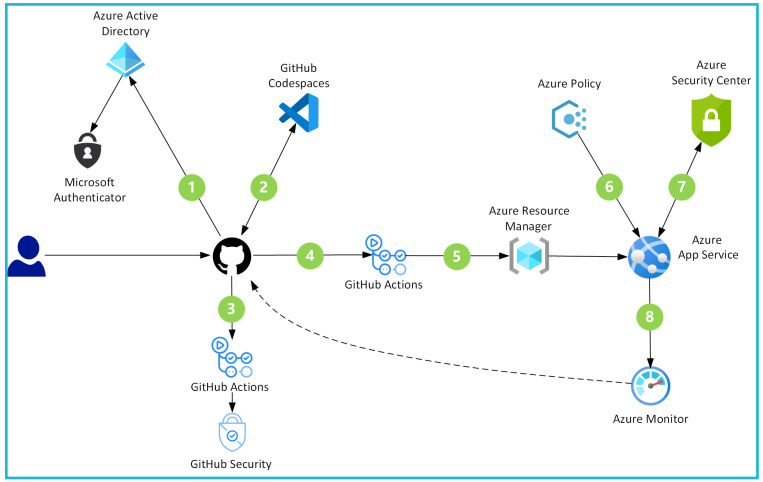
- IaC
- Security scanning
- Cloud configuration
- Security acceptance tests

- Security smoke tests
- Security Configuration
- Secret Management
- Server Hardening

- Continuous monitoring
- Threat intelligence
- Penetration Testing
- Blameless postmortems

DevSecOps in GitHub

- Browser-based IDEs with built-in security extensions.
- Agents that continuously monitor security advisories and replace vulnerable and out-of-date dependencies.
- Search capabilities that scan source code for vulnerabilities.
- Action-based workflows that automate every step of development, testing, and deployment.
- Spaces that provide a way to privately discuss and resolve security threats and then publish the information.



* https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/devsecops-in-github

* Combined with the monitoring and evaluation power of Azure, these features provide a superb service for building secure cloud solutions.

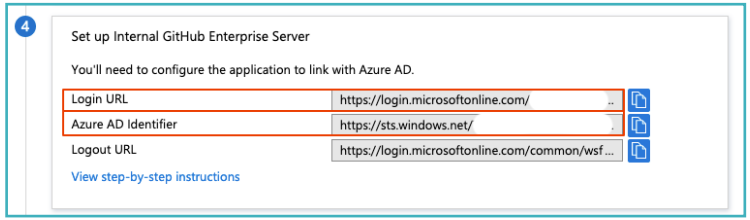
DevSecOps in GitHub

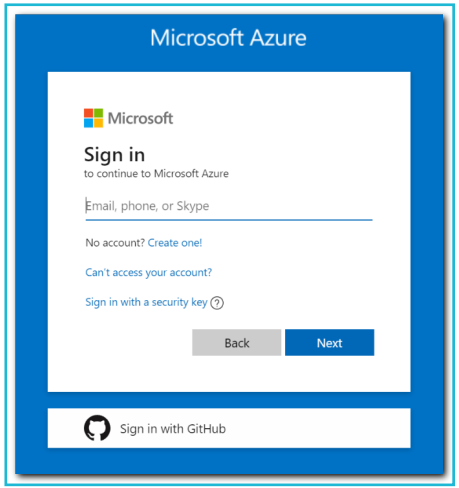
kloia

Foundation of the Enterprise in Cloud : Azure AD

- Azure AD auth to GitHub
- GitHub auth to Azure portal

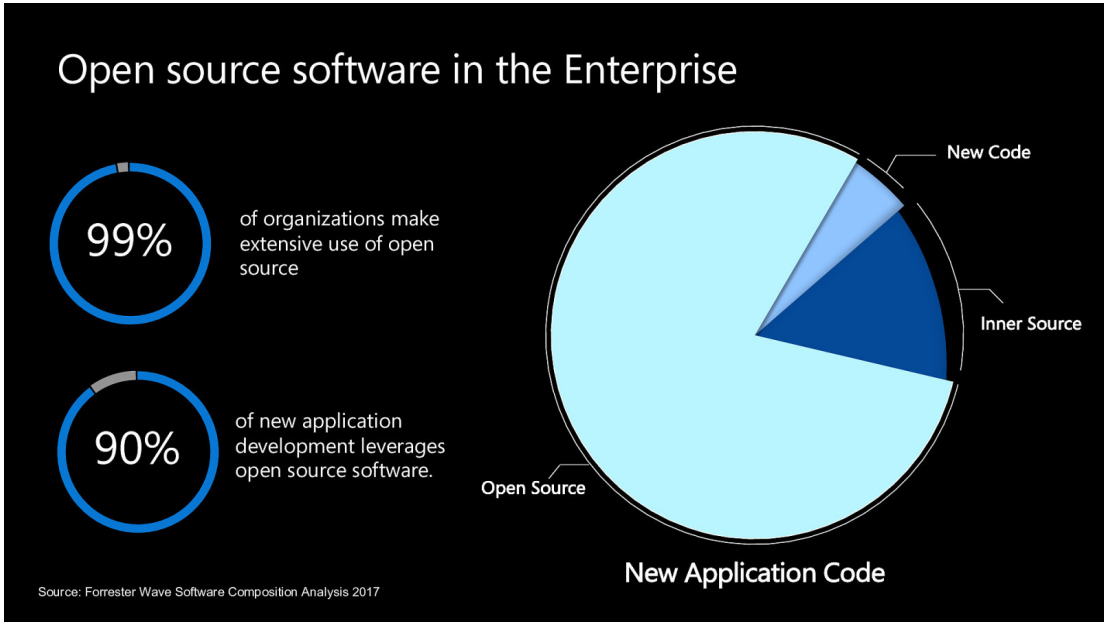
  - SAML SSO
  - LDAP
  - RBAC
  - Required 2FA
  - GitHub Connect
  - Audit log





DevSecOps in GitHub

Secure your supply chain



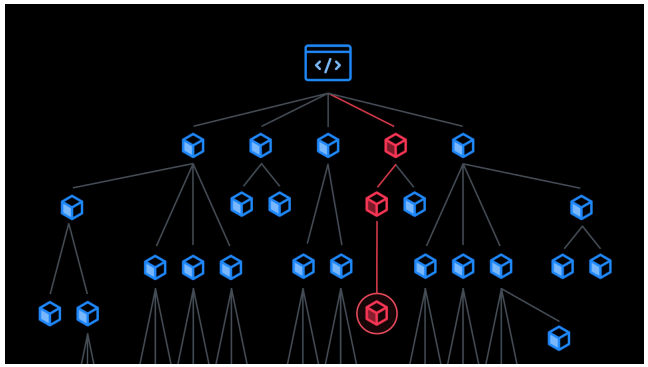
* Slide 22 : https://speaking.sasharosenbaum.com/EI8loj/slides

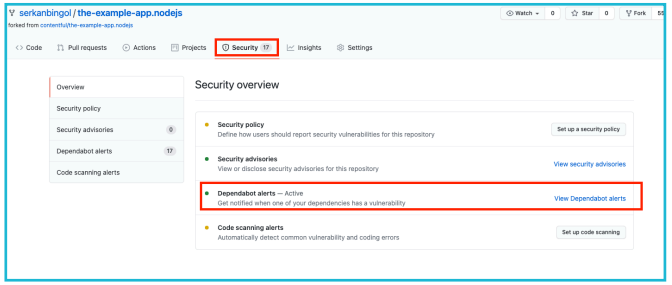
DevSecOps in GitHub

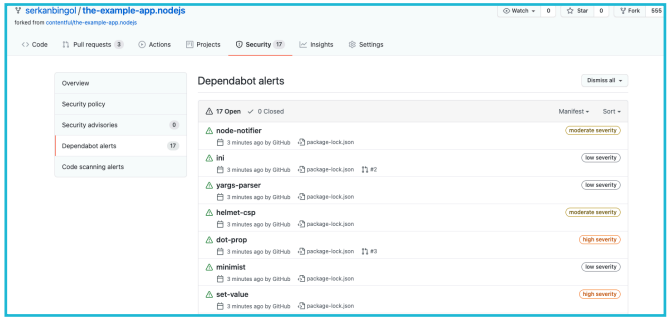
kloia

Secure your supply chain: GitHub Dependabot Alerts

- Automate dependency updates
- Identificate vulnerable dependencies





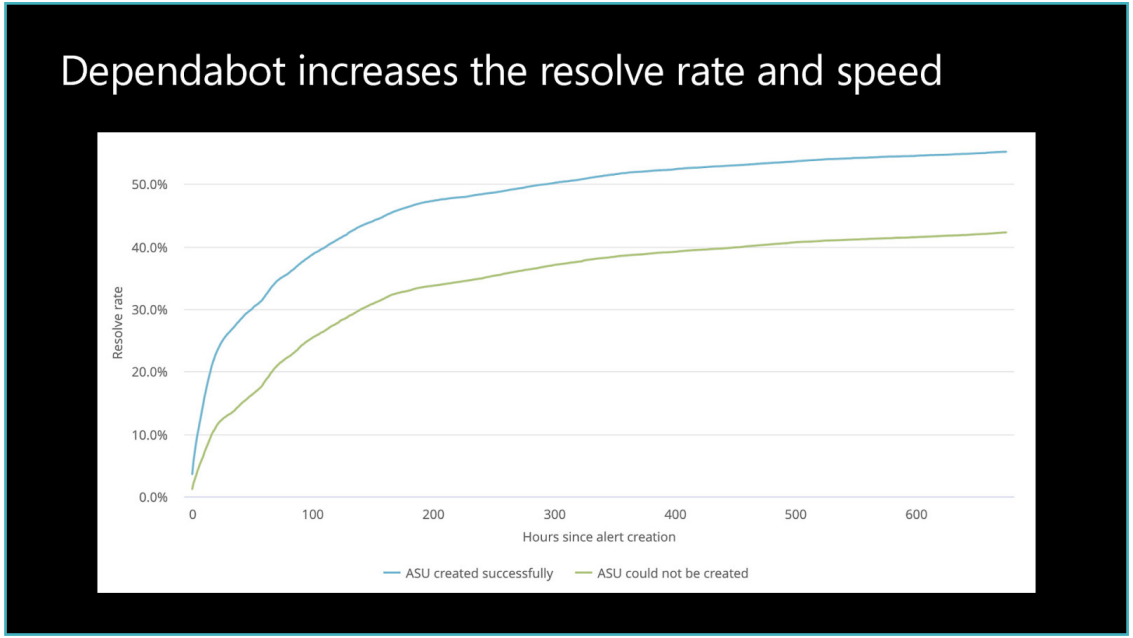


Managing Vulnerabilities in your project Dependencies :
https://docs.github.com/en/github/managing-security-vulnerabilities/managing-vulnerabilities-in-your-projects-dependencies

DevSecOps in GitHub

kloia

Secure your supply chain: GitHub Dependabot Alerts



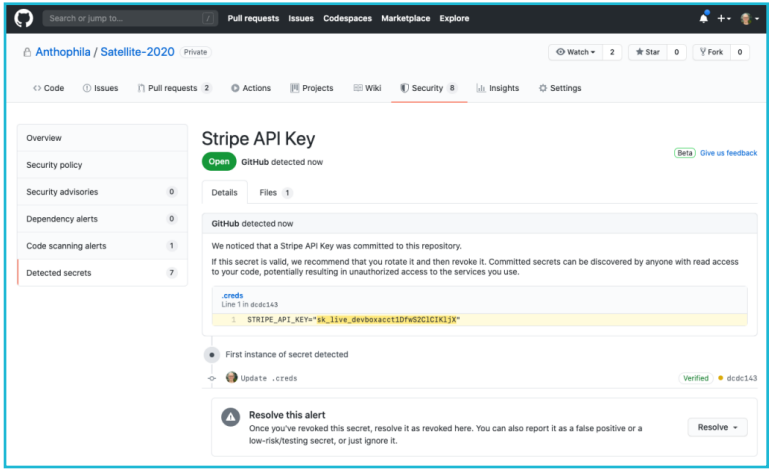
* Slide 30 : https://speaking.sasharosenbaum.com/EI8loj/slides

DevSecOps in GitHub

kloia

Secure your repository: GitHub Secret Scanning

- Scan and detect secrets in your repository
- Lots of service providers
  - Atlassian
  - Azure
  - AWS
  - Alibaba Cloud
  - Google Cloud
  - Stripe
  - ……



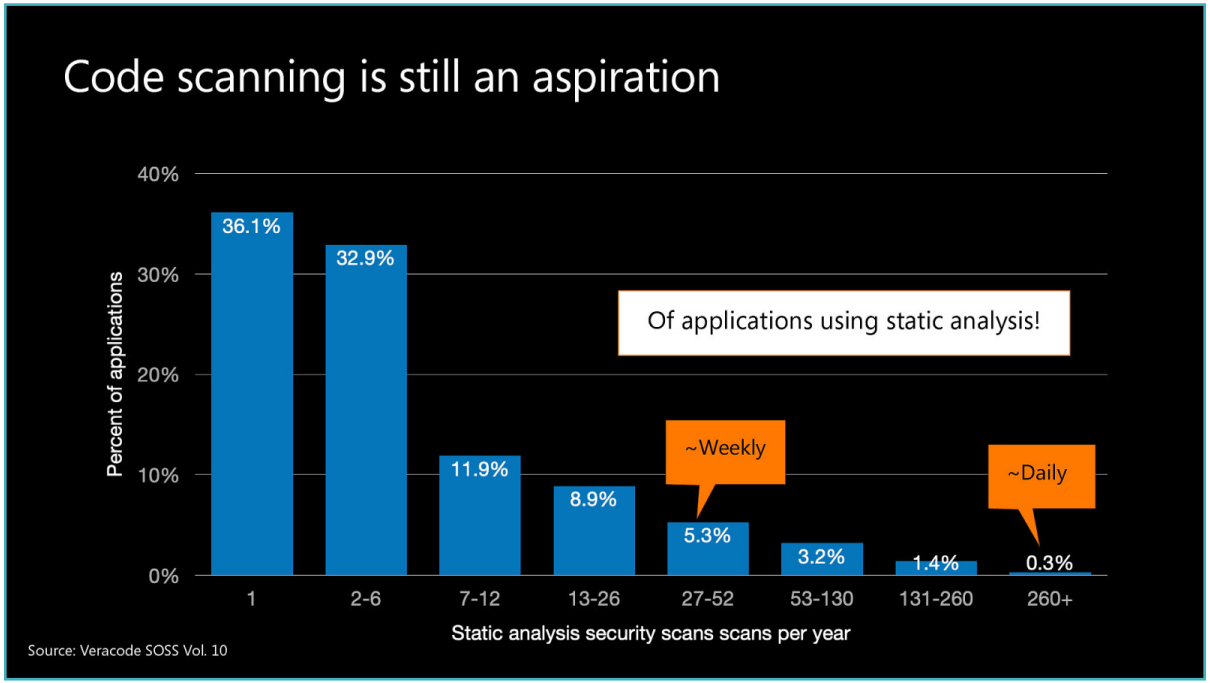
* Secret scanning is available in public repositories, and in private repositories owned by organizations with an Advanced Security license.

Configuring secret scanning for your repositories :
https://docs.github.com/en/github/administering-a-repository/configuring-secret-scanning-for-your-repositories

DevSecOps in GitHub

kloia

Analyse your code



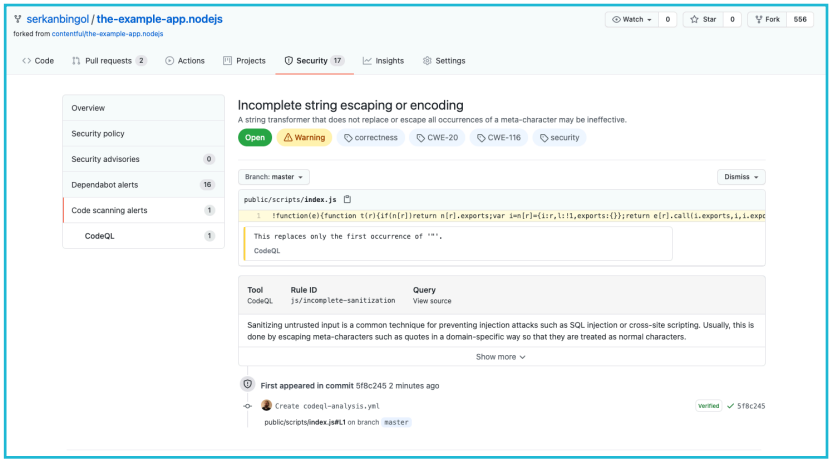
* Slide 37 : https://speaking.sasharosenbaum.com/EI8loj/slides

DevSecOps in GitHub

kloia

Analyse your code: GitHub Code Scanning

- Automate your code review.
- CodeQL has leading security teams and individuals.
- CodeQL has more CVEs than any other static application security testing vendor team.
- GitHub is now a CNA.

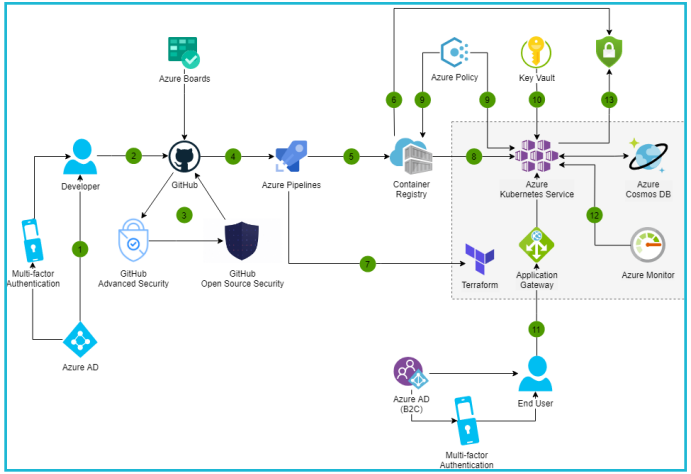


* CVE : Common Vulnerabilities and Exposures
** CNA : CVE Numbering Authority
*** Zero-Day : Software vulnerability that is unknown to, or unaddressed by, those who are interested in mitigating it.

About code scanning :
https://docs.github.com/en/github/finding-security-vulnerabilities-and-errors-in-your-code/about-code-scanning

DevSecOps in Azure

- **Azure AD** can be configured as the identity provider for GitHub.
- **GitHub Enterprise** can integrate automatic security and dependency scanning.
- **Azure Pipelines** generates a Docker container image that is stored to **Azure Container Registry**, which is to be used at release time by **Azure Kubernetes Service**.
- A release on Azure Pipelines integrates the **Terraform tool**, managing both the cloud infrastructure as code, provisioning resources such as Azure Kubernetes Service, **Application Gateway, and Azure Cosmos DB**.
- **Azure Security Center** will be able to do active threat monitoring on the Azure Kubernetes Service, on both Node level (VM threats) and internals.



* https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/devsecops-in-azure

DevSecOps in Azure

kloia

Infrastructure as Code : Terraform

- Deliver Infrastructure as Code

  - Write : Write infrastructure as code using declarative configuration files.
  - Plan : Check whether the execution plan for a configuration matches your expectations before provisioning or changing infrastructure.
  - Apply : Apply changes to hundreds of cloud providers to reach the desired state of the configuration.

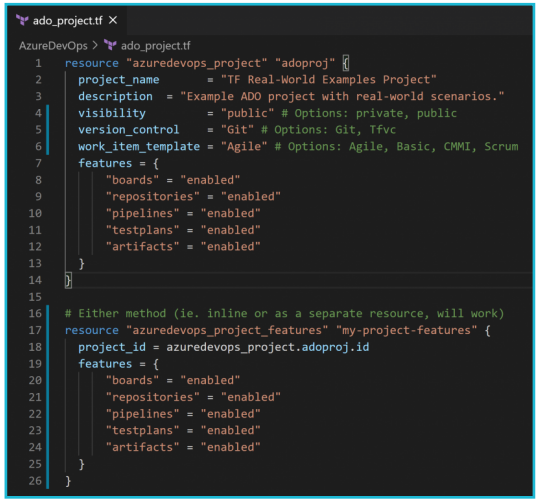


image from https://adinermie.com/

Terraform is commercial templating tool that can provision cloud-native applications across all the major cloud players: Azure, Google Cloud Platform, AWS, and AliCloud. Instead of using JSON as the template definition language, it uses the slightly more terse YAML.

Infrastructure as code with Microsoft:
https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/infrastructure-as-code

DevSecOps in Azure

kloia

Policy-as-code : Azure Policy

- Understand evaluation outcomes
- Control the response to an evaluation
- Remediate non-compliant resources
- Single source of truth

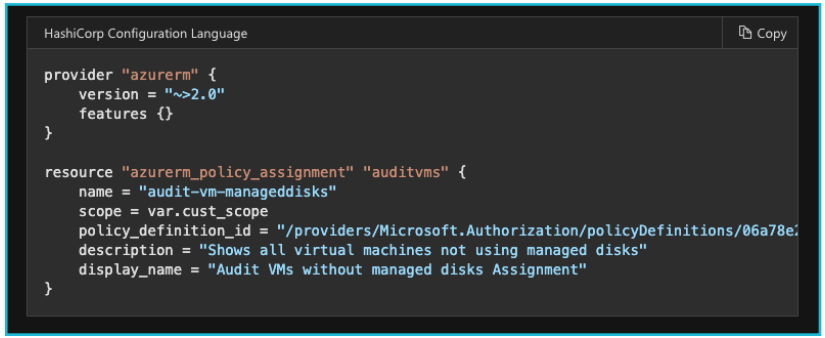


image from https://docs.microsoft.com/en-us/azure/governance/policy/assign-policy-terraform

Azure Policy is a service that offers both built-in and user-defined policies across categories mapping the various Azure services such as Compute, Storage or even AKS. These policies can be defined on the Azure Portal and assigned to one or more subscriptions/resource groups.

Azure Policy documentation :
https://docs.microsoft.com/en-us/azure/governance/policy/

DevSecOps in Azure

kloia

Maintain your keys : Azure Key Vault

- Centralize application secrets
- Securely store secrets and keys
- Simplified administration of application secrets
- Integrate with other Azure services

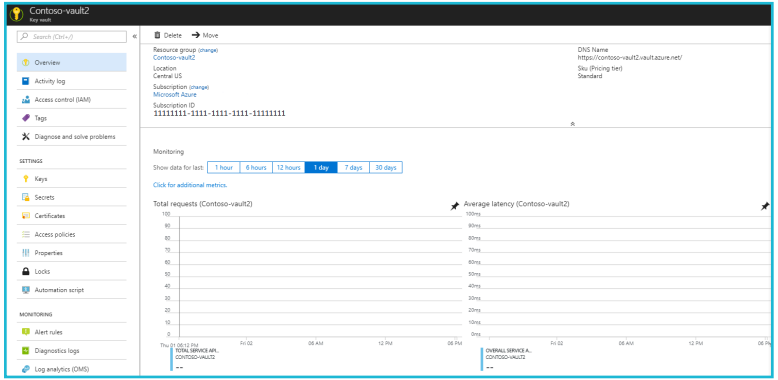


image from https://docs.microsoft.com/en-us/azure/key-vault/general/quick-create-portal

Azure Key Vault helps solve “Secrets Management” , “Key Management” and “Certificate Management” problems.

Azure Key Vault documentation :
https://docs.microsoft.com/en-us/azure/key-vault/general/overview

DevSecOps in Azure

kloia

Keep your resources safe: Azure Security Center

- Strengthen security posture
- Protect against threats
- Get secure faster
- Prioritize your security work

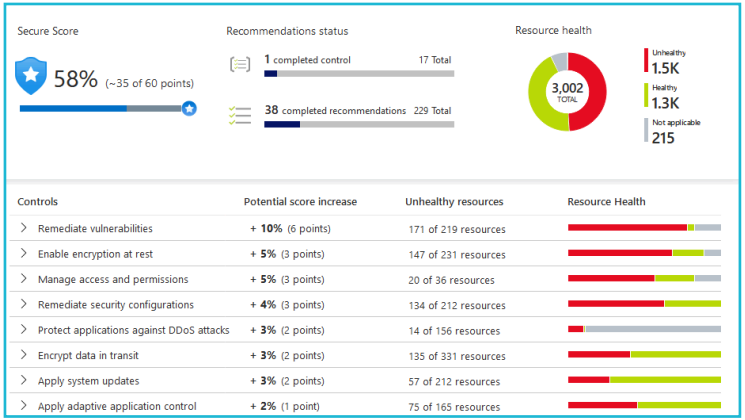


image from https://docs.microsoft.com/en-us/azure/security-center/security-center-introduction

Azure Security Center is a unified infrastructure security management system that strengthens the security posture of your data centers, and provides advanced threat protection across your hybrid workloads in the cloud - whether they're in Azure or not - as well as on premises.

Azure Security Center documentation :
https://docs.microsoft.com/en-us/azure/security-center/

3rd Party DevSecOps Tools

kloia

Check open source components : Whitesource Bolt

- WhiteSource Secures & Manages Your Open Source Usage
- Get Real-Time Alerts on Security Vulnerabilities
- Ensure license compliance
- Automated Up-to-Date Inventory Reports

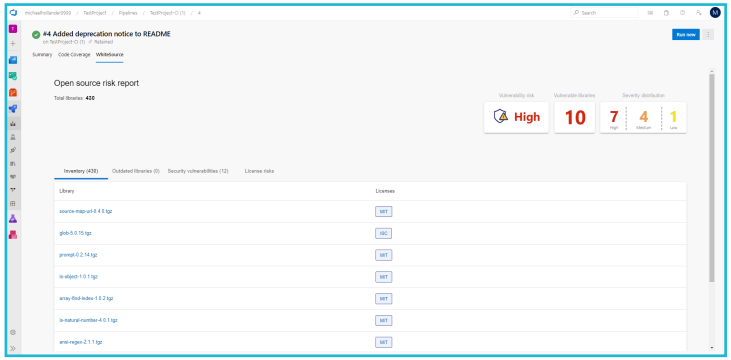


image from https://marketplace.visualstudio.com/items?itemName=whitesource.whiteSource-bolt-v2

WhiteSource Bolt scans all your projects and detects open source components, their license and known vulnerabilities. Not to mention, it also provide fixes.

Whitesource Bolt Ver. 1.0 documentation :
https://marketplace.visualstudio.com/items?itemName=whitesource.ws-bolt
Whitesource Bolt Ver. 2.0 documentation :
https://marketplace.visualstudio.com/items?itemName=whitesource.whiteSource-bolt-v2

3rd Party DevSecOps Tools

kloia

Demonstrate security and penetration testing: WebGoat

- Learn the hack - Stop the attack
- Test vulnerabilities
- Create a de-facto interactive teaching environment

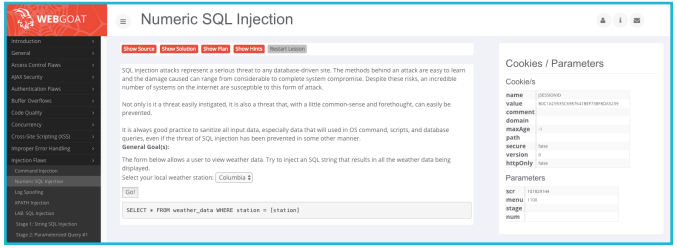


image from https://www.zupzup.org/websecurity-with-webgoat

WebGoat is a deliberately insecure web application maintained by OWASP designed to teach web application security lessons.

WebGoat documentation :
https://github.com/WebGoat/WebGoat

3rd Party DevSecOps Tools

kloia

Create security report : Microsoft Security Code Analysis

- Simple configuration and execution
- Keep builds clean
- Auto-Update
- Break the build
- Scan Credentials

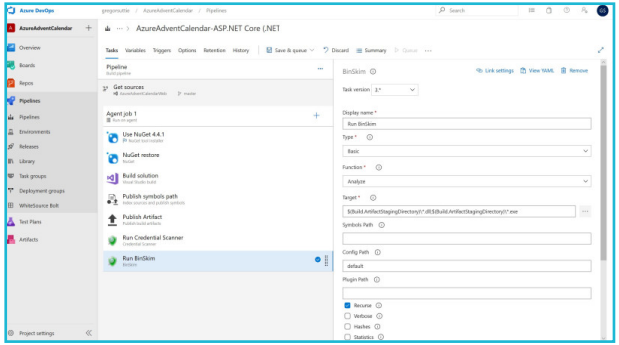


image from https://techdailychronicle.com/

The Microsoft Security Code Analysis extension empowers you to do so, easily integrating the running of static analysis tools in your Azure DevOps pipelines..

Microsoft Security Code Analysis Extension documentation :
https://secdevtools.azurewebsites.net/

3rd Party DevSecOps Tools

kloia

Scan containers for the security and compliance: Anchore

Configure Anchore to authenticate with Azure Container Registry (ACR) and analyze an image.

The Anchore Engine is an open-source project that provides a centralized service for inspection, analysis, and certification of container images. The Anchore Engine is provided as a Docker container image that can be run standalone or within an orchestration platform such as Kubernetes, Docker Swarm, Rancher, Amazon ECS, and other container orchestration platforms.

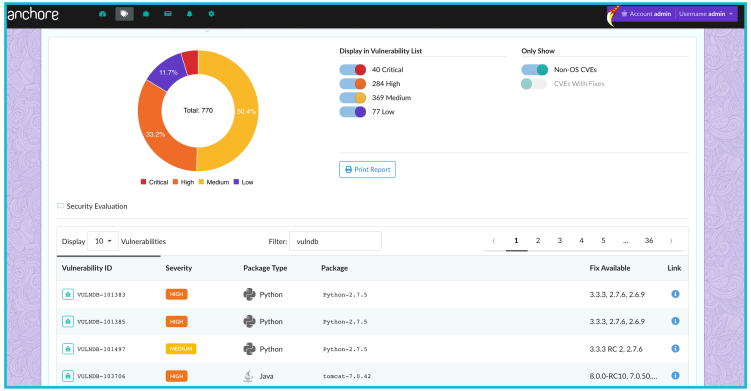


image from https://anchore.com/blog/enhanced-vulnerability-data/

Anchore documentation :
https://anchore.com/opensource/

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Prerequisites

- An Azure subscription
- Azure DevOps account
- GitHub Account
- Helm CLI
- Anchore CLI
- Kubectl

Steps

- Create **Azure DevOps** project and prepare WebGoat source code
- Create container registry with **ACR** and **Azure Key Vault**
- Configure CI/CD pipeline for **Webgoat**
- Check open source components with **Whitesource Bolt**
- Scan containers with **Anchore** for the security with **AKS**

DEMO: Implement Security in Azure DevOps CI/CD

kloia

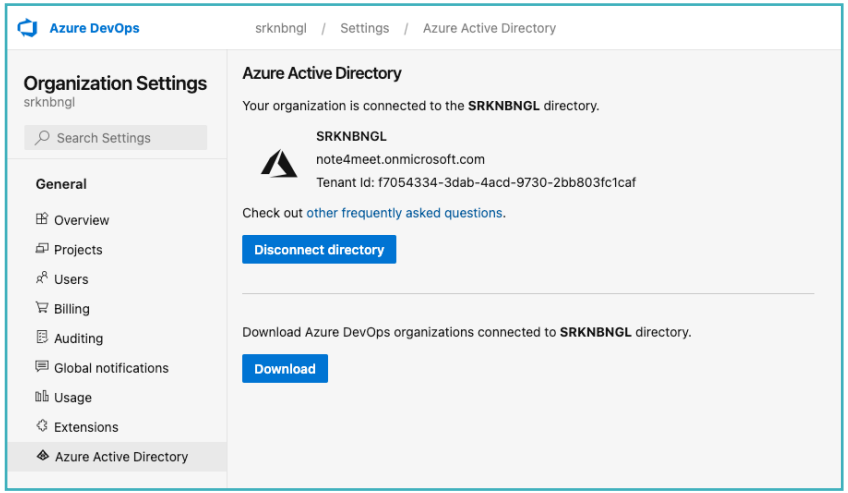
Install Required CLI's

Azure CLI :
https://docs.microsoft.com/en-us/cli/azure/install-azure-cli
Helm CLI :
https://helm.sh/docs/intro/install/
Kubectl :
https://kubernetes.io/docs/tasks/tools/install-kubectl/
Anchore CLI :
https://github.com/anchore/anchore-cli
Anchore CLI for Mac M1 :
https://www.dynamsoft.com/codepool/python-barcode-apple-m1-mac.html

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Azure DevOps & Azure Subscription Connections

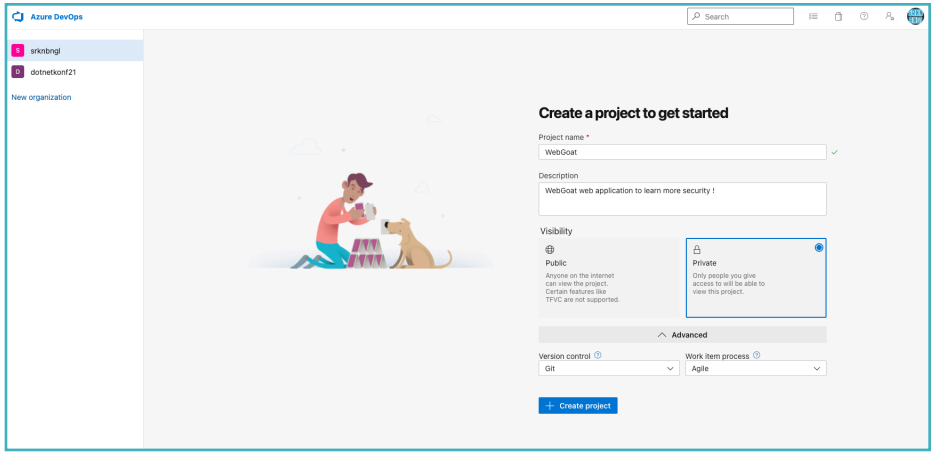


Connect your organization to Azure Active Directory :
https://docs.microsoft.com/en-us/azure/devops/organizations/accounts/connect-organization-to-azure-ad?view=azure-devops

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Create Azure DevOps Project

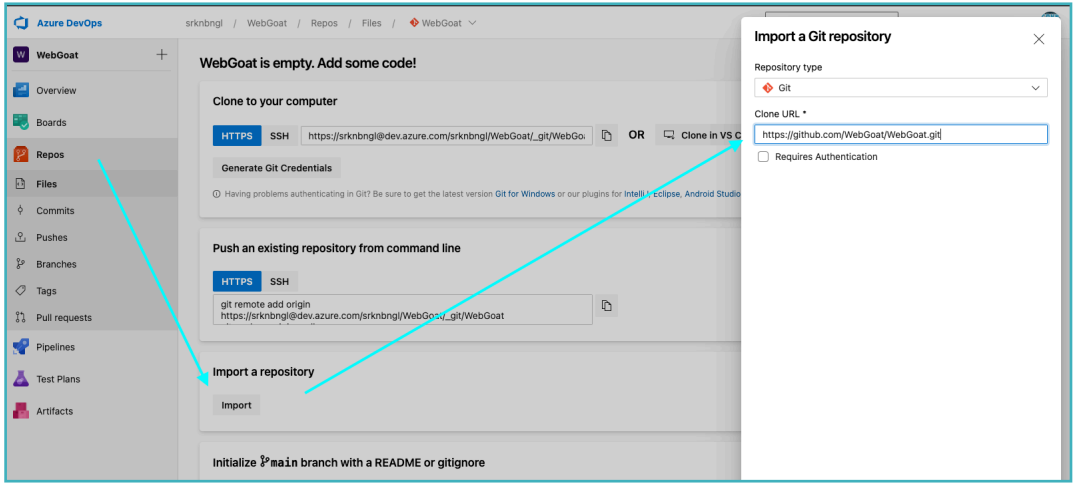


Create a project in Azure DevOps : https://docs.microsoft.com/en-us/azure/devops/organizations/projects/create-project?view=azure-devops&tabs=preview-page

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Create a new Git repo in your project



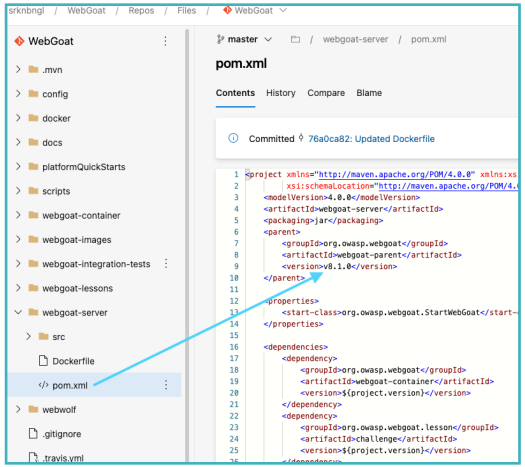
Create a new Git repo in your project :
https://docs.microsoft.com/en-us/azure/devops/repos/git/create-new-repo?view=azure-devops

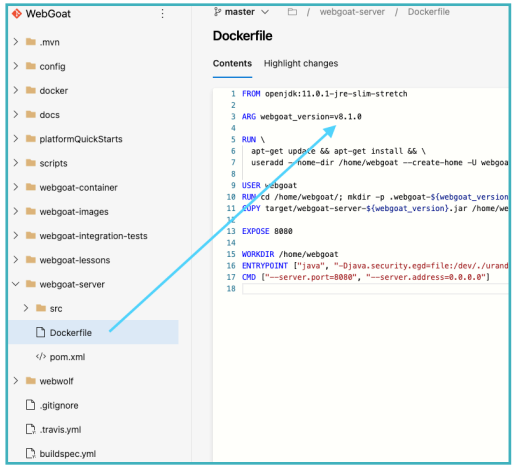
WebGoat GitHub project :
https://github.com/WebGoat/WebGoat

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Update WebGoat dockerfile version





DEMO: Implement Security in Azure DevOps CI/CD

kloia

Azure CLI : Login & Set Subscription

```
srknbngl@t-800 ~
srknbngl@t-800 ~ az login
The default web browser has been opened at https://login.microsoftonline.com/common/oauth2/authorize. Please continue the login in the
ils to open, use device code flow with `az login --use-device-code`.
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "managedByTenants": [],
    "name": "VSE_Production",
    "state": "Enabled",
    "user": {
      "name": "serkanbingol@outlook.com",
      "type": "user"
    }
  },
  {
    "cloudName": "AzureCloud",
    "managedByTenants": [],
    "name": "VSE_Test",
    "state": "Enabled",
    "user": {
      "name": "serkanbingol@outlook.com",
      "type": "user"
    }
  }
]
srknbngl@t-800 ~ az account show --output table
EnvironmentName    HomeTenantId                          IsDefault    Name      State    TenantId
-----------------  ------------------------------------  -----------  --------  -------  ------------------------------------
AzureCloud                                               True         VSE_Test  Enabled
srknbngl@t-800 ~
```

Sign in with Azure CLI :
https://docs.microsoft.com/en-us/cli/azure/authenticate-azure-cli

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Azure CLI : Create Resource Group

```
srknbngl@t-800 ~ az group create --name webGoatKonfRG --location westeurope


{
  "id": "/subscriptions/[redacted]f/resourceGroups/webGoatKonfRG",
  "location": "westeurope",
  "managedBy": null,
  "name": "webGoatKonfRG",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```

azure-cli command
az group create --name webGoatKonfRG --location westeurope

Create Azure Resource Group :
https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/manage-resource-groups-cli

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Azure CLI : Create Azure Container Registry

```
srknbngl@t-800 ~ az acr create --resource-group webGoatKonfRG --name webGoatKonfACR --sku Basic --admin-enabled true --location westeurope
{- Finished ..
  "adminUserEnabled": true,
  "creationDate": "2021-02-20T10:51:44.465988+00:00",
  "dataEndpointEnabled": false,
  "dataEndpointHostNames": [],
  "encryption": {
    "keyVaultProperties": null,
    "status": "disabled"
  },
```

azure-cli command
az acr create --resource-group webGoatKonfRG --name webGoatKonfACR --sku Basic --admin-enabled true --location westeurope

Create Azure Container Registry :
https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/manage-resource-groups-cli

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Azure CLI : Azure Key Vault

```
srknbngl@t-800 ~ az keyvault create --name "webGoatKonfKV" --resource-group "webGoatKonfRG" --location westeurope
{- Finished ..
  "id": "/subscriptions/[redacted]/resourceGroups/webGoatKonfRG/providers/Microsoft.KeyVault/vaults/webGoatKonfKV",
  "location": "westeurope",
  "name": "webGoatKonfKV",
  "properties": {
    "accessPolicies": [
      {
        "applicationId": null,
        "objectId": "[redacted]",
        "permissions": {
          "certificates": [
            "get",
            "list"
```

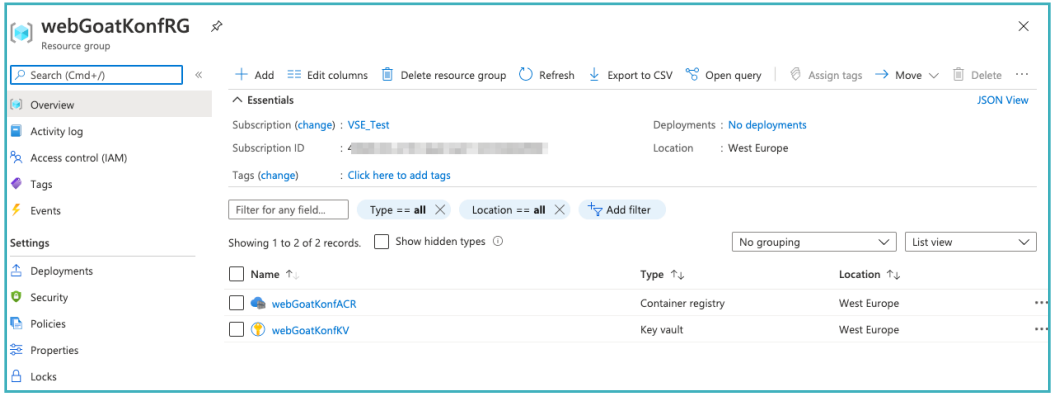
azure-cli command
az keyvault create --name "webGoatKonfKV" --resource-group "webGoatKonfRG" --location westeurope

Create Azure Key Vault :
https://docs.microsoft.com/bs-cyrl-ba/Azure/key-vault/general/quick-create-cli

DEMO: Implement Security in Azure DevOps CI/CD

kloia

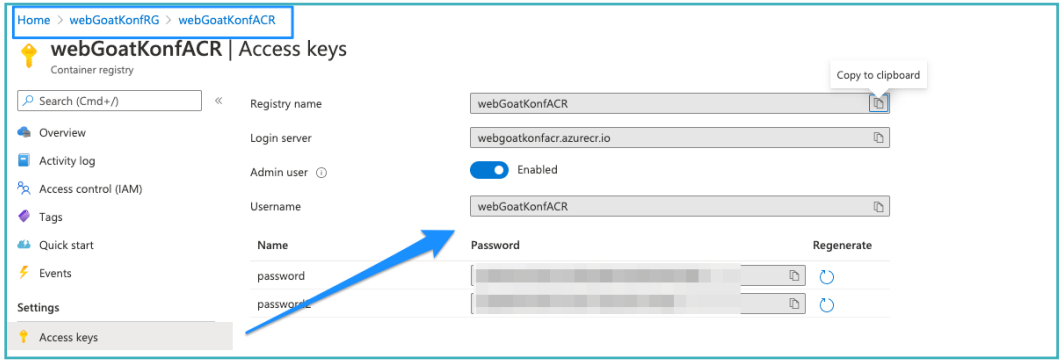
Check Azure Resources - Resource Group , ACR , Key Vault



DEMO: Implement Security in Azure DevOps CI/CD

kloia

Get Azure CR Credentials

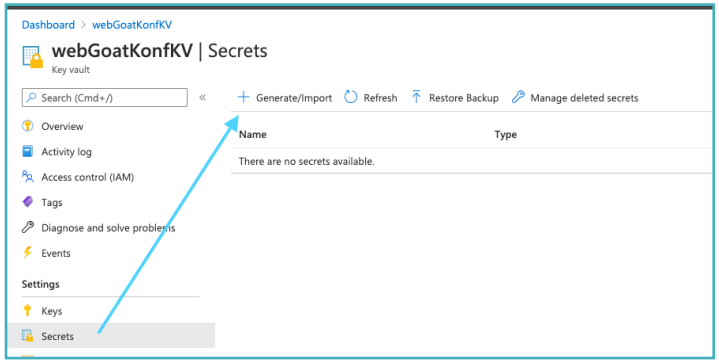


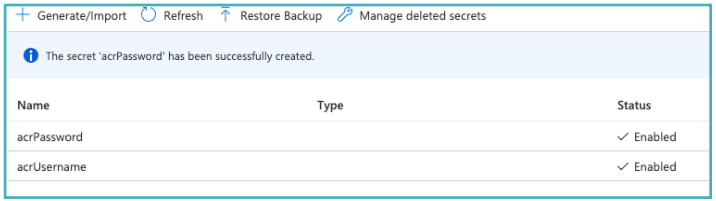
Azure Container Registry Documentation :
https://docs.microsoft.com/en-us/azure/container-registry/

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Create secrets Azure Key Vault



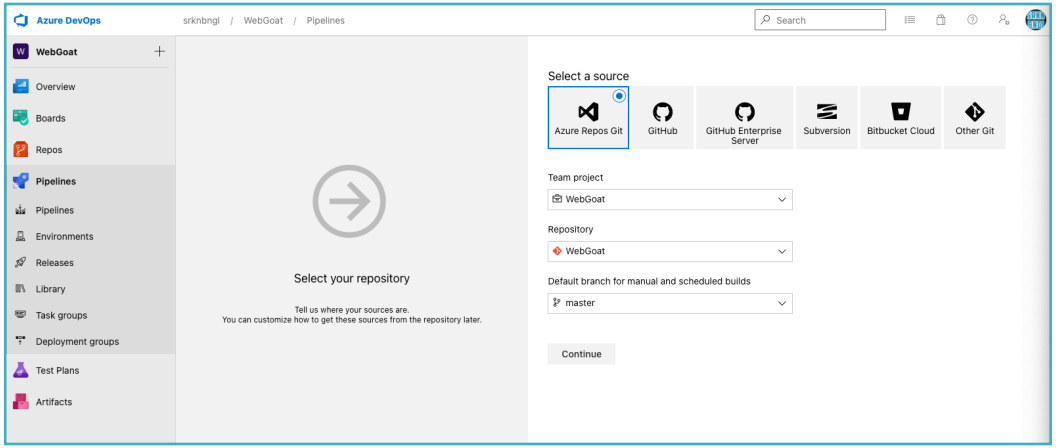


Use **acrUsername** and **acrPassword** names and set ACR credentials as value.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Build Pipeline



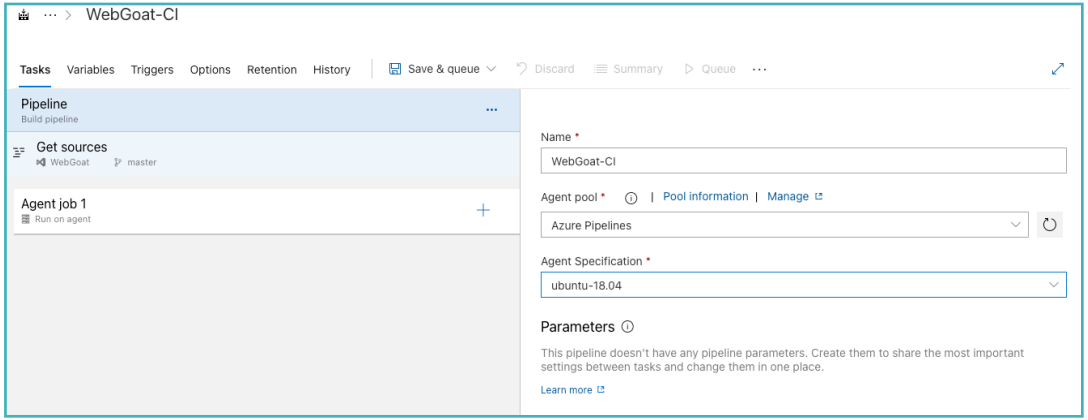
Use **the classic editor** to create a pipeline without YAML.

Create your first pipeline :
https://docs.microsoft.com/en-us/azure/devops/pipelines/create-first-pipeline?view=azure-devops&tabs=java%2Ctfs-2018-2%2Cbrowser

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Build Pipeline

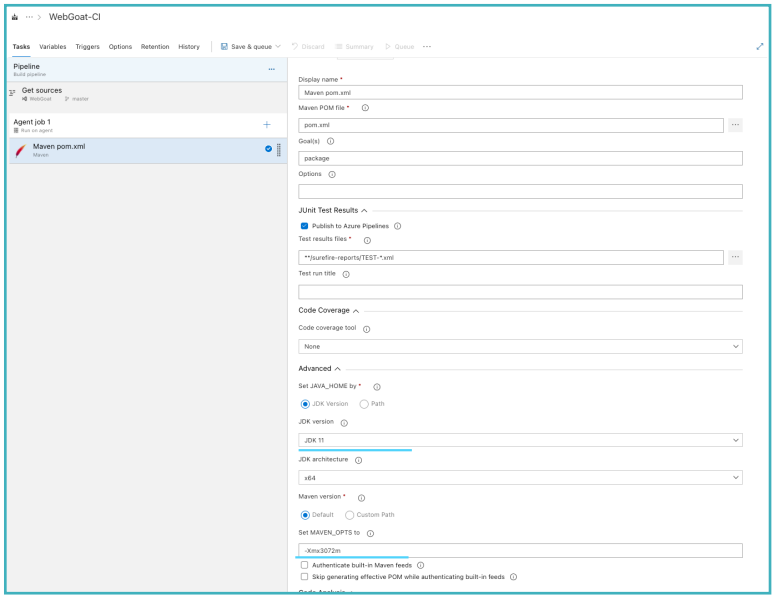


Use **Empty job** to start. Select ubuntu-18.04 agent for agent job.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Build Pipeline

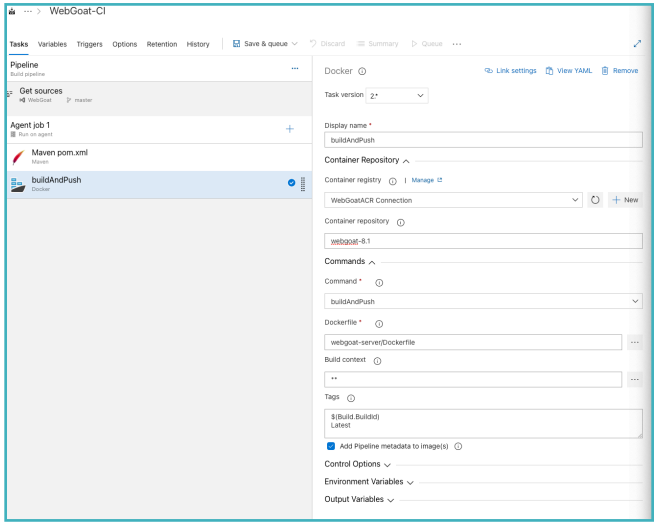


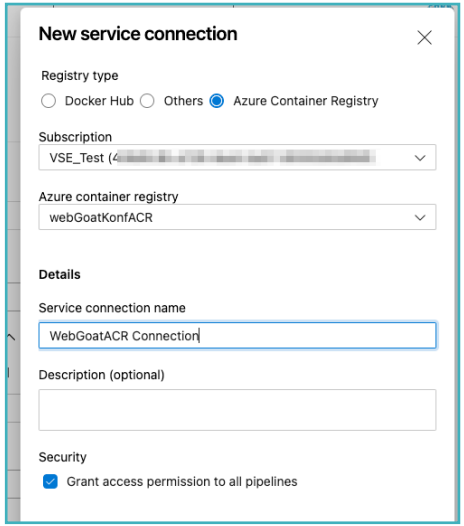
Add **Maven** task to then “Advanced” step set MAVEN_OPT to **-Xmx3072m** .

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Build Pipeline



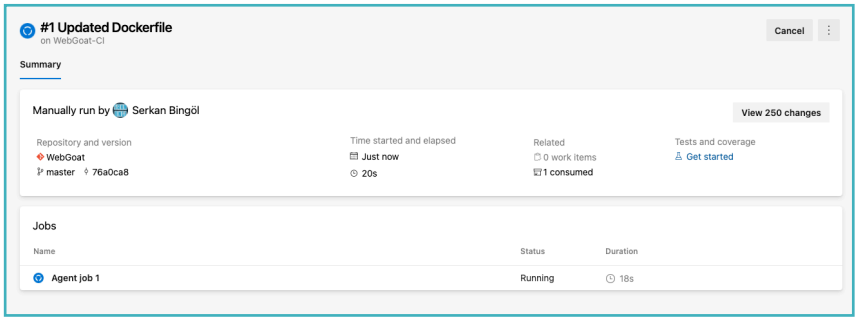


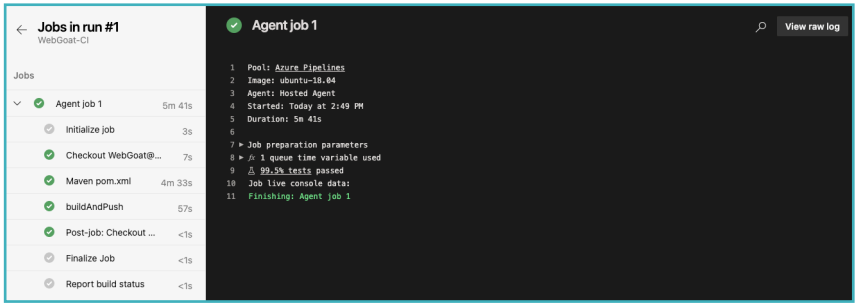
Add **Docker** task. Select **webgoat-server** folder dockerfile to build. Add **Latest** to tags. Create new Container Registry connection with **New** button. Last step **Save & Queue** .

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Build Pipeline



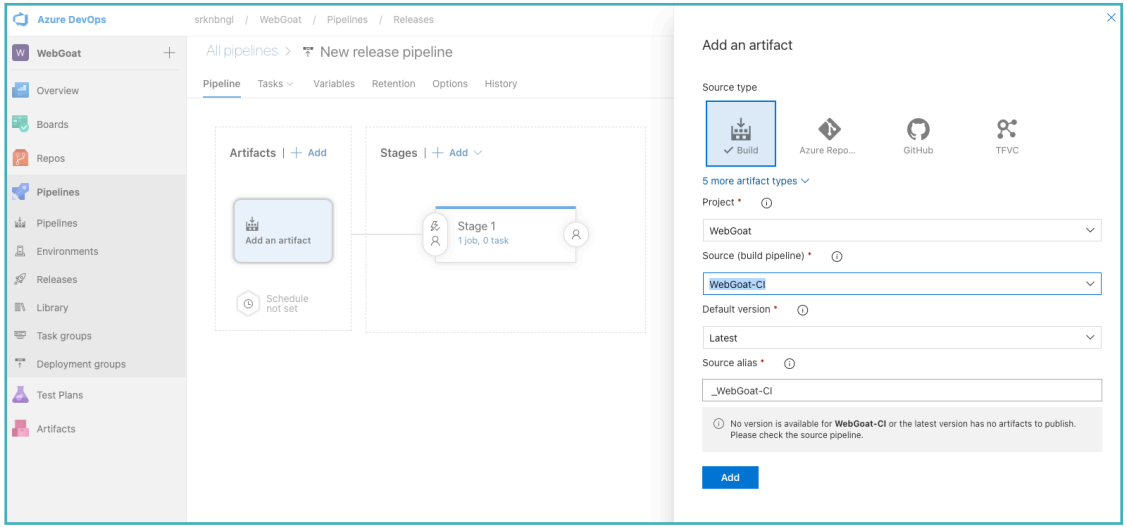


Build will take approximately 5 mins then check build pipeline finished with successful.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Release Pipeline : Add Artifact

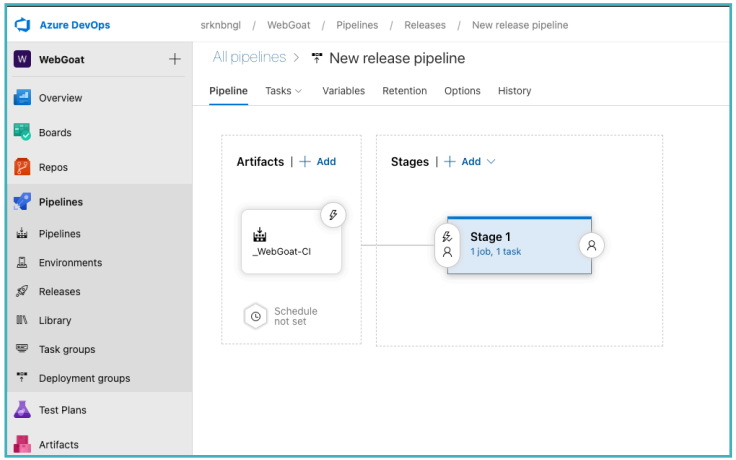


Define your multi-stage continuous deployment (CD) pipeline :
https://docs.microsoft.com/en-us/azure/devops/pipelines/release/define-multistage-release-process?view=azure-devops

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Release Pipeline : Add Stage

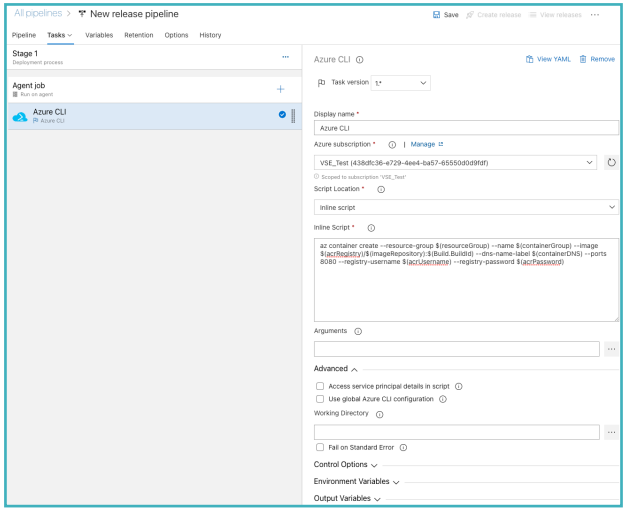


Define your multi-stage continuous deployment (CD) pipeline :
https://docs.microsoft.com/en-us/azure/devops/pipelines/release/define-multistage-release-process?view=azure-devops

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Release Pipeline : Add Stage Tasks



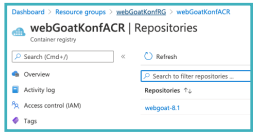
azure-cli command
az container create --resource-group $(**resourceGroup**) --name $(**containerGroup**) --image $(**acrRegistry**)/$(**imageRepository**):$(**Build.BuildId**) --dns-name-label $(**containerDNS**) --ports 8080 --registry-username $(**acrUsername**) --registry-password $(**acrPassword**)

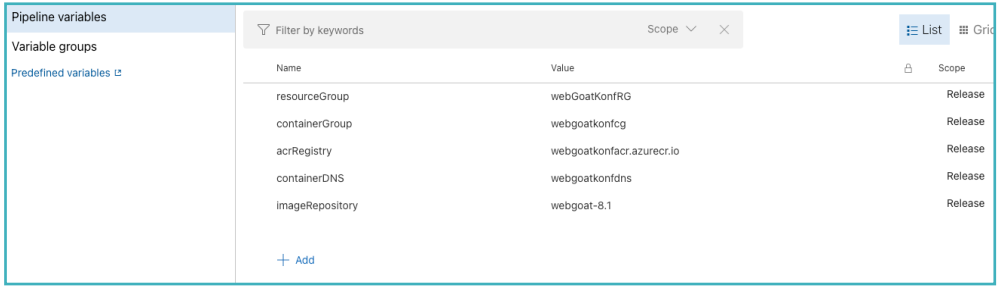
Add **Azure CLI** task. Select **Azure Subscription** created before. Select **Inline Script**. Add cli command given above to Inline Script area.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Release Pipeline : Add Stage Variables - Non-sensitive





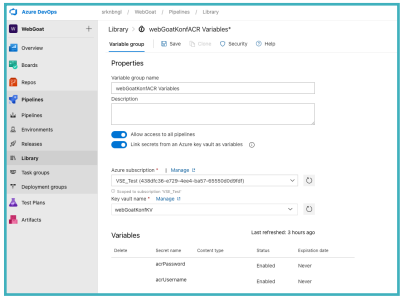
Get values from **webGoatKonfACR** named container registry to use in release pipeline variables.

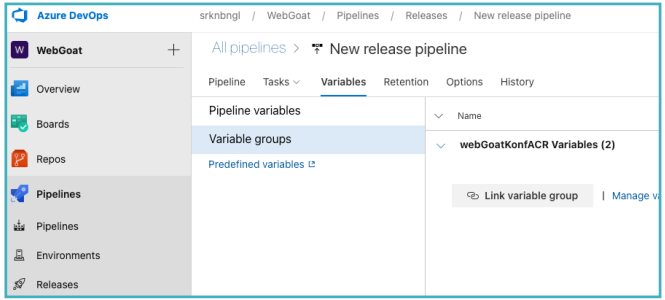
In variables section for non-sensitive variables add **resourceGroup , containerGroup , acrRegistry , containerDNS , imageRepository** names with values of ***webGoatKonfRG*** , ***webgoatkonfcg*** , ***webgoatkonfacr.azurecr.io***, ***webgoatkonfdns*** , ***webgoat-8.1*** in a row under Pipeline variables area.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Configure Release Pipeline : Add Stage Variables - Sensitive



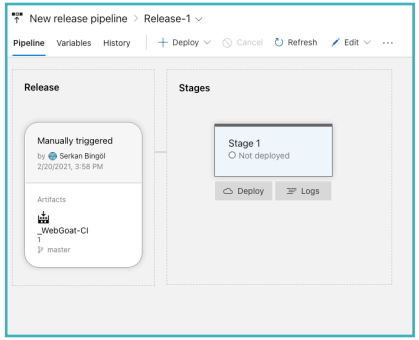


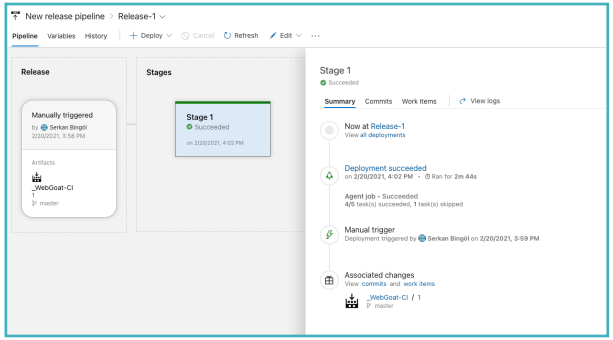
Create Variable Group from Azure Key Vault under library menu from Pipelines section.

For sensitive variables select variable groups under release pipeline variables title and link it with created from Azure Key Vault service.

DEMO: Implement Security in Azure DevOps CI/CD

Create Release





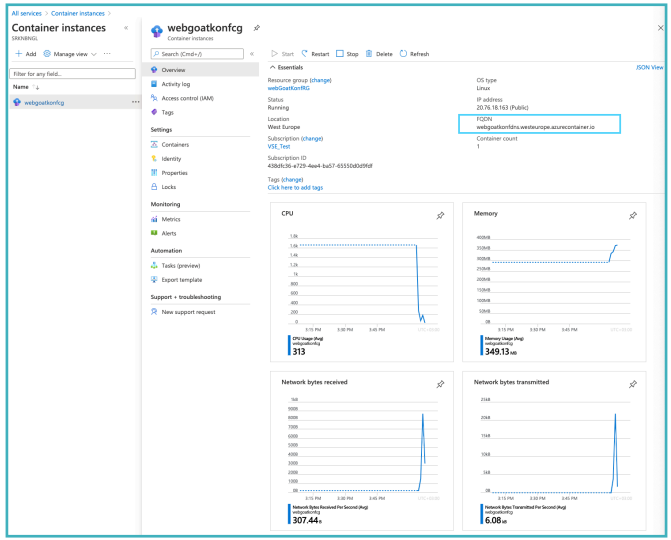
Create a new release under release pipeline by clicking **Create New Release** button.

You need a manually approve to deploy created artifact by pushing deploy button in Stage 1

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Check Azure Container Instances

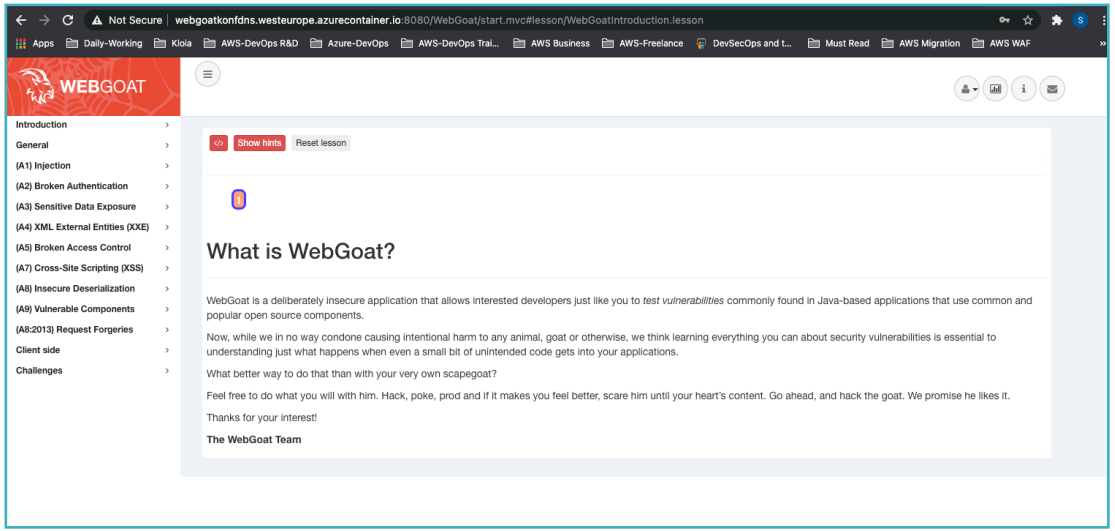


After create a successful deployment go to Azure Container Instance service to get our container published web link.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Browse WebGoat and Learn Security !



WebGoat Container Browse Link : http://ACI_FQDN:8080/WebGoat

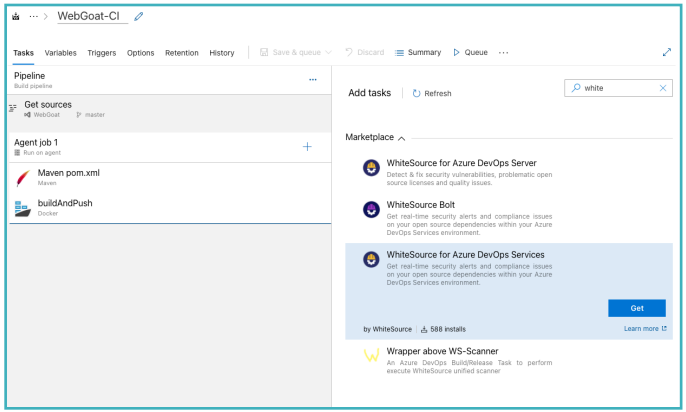
WebGoat Official Page : https://owasp.org/www-project-webgoat/

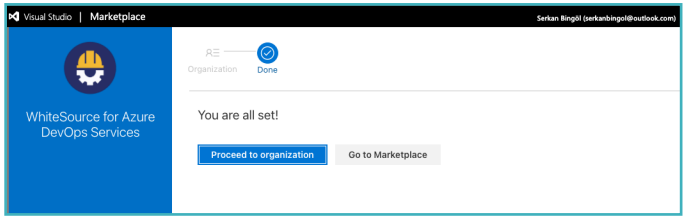
*ACI_FQDN is our container instance FQDN

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Whitesource Bolt: check open source components



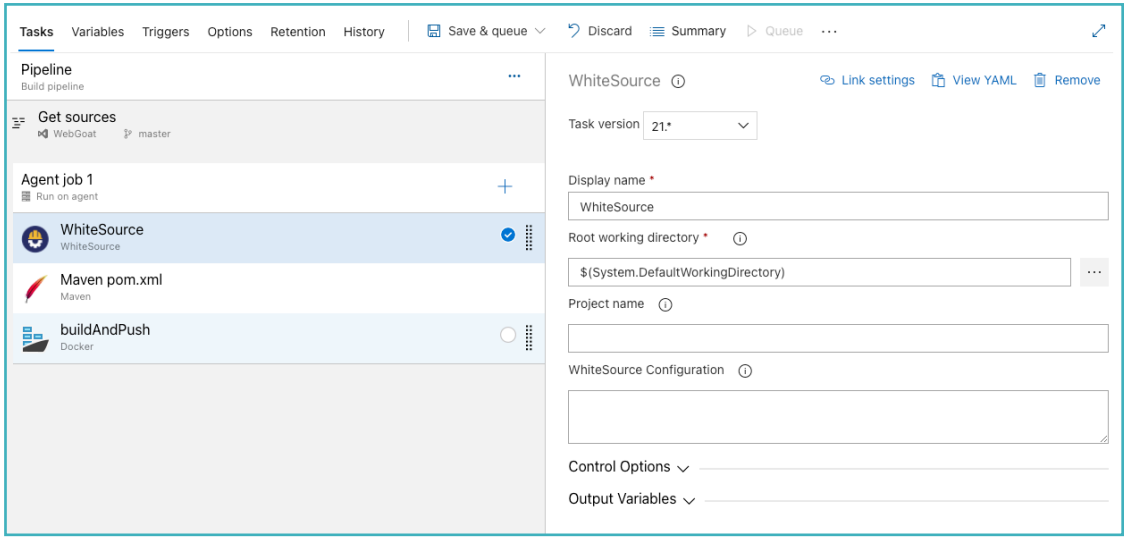


Add Whitesource Bolt extension to build pipeline from Visual Studio Marketplace

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Whitesource Bolt: check open source components

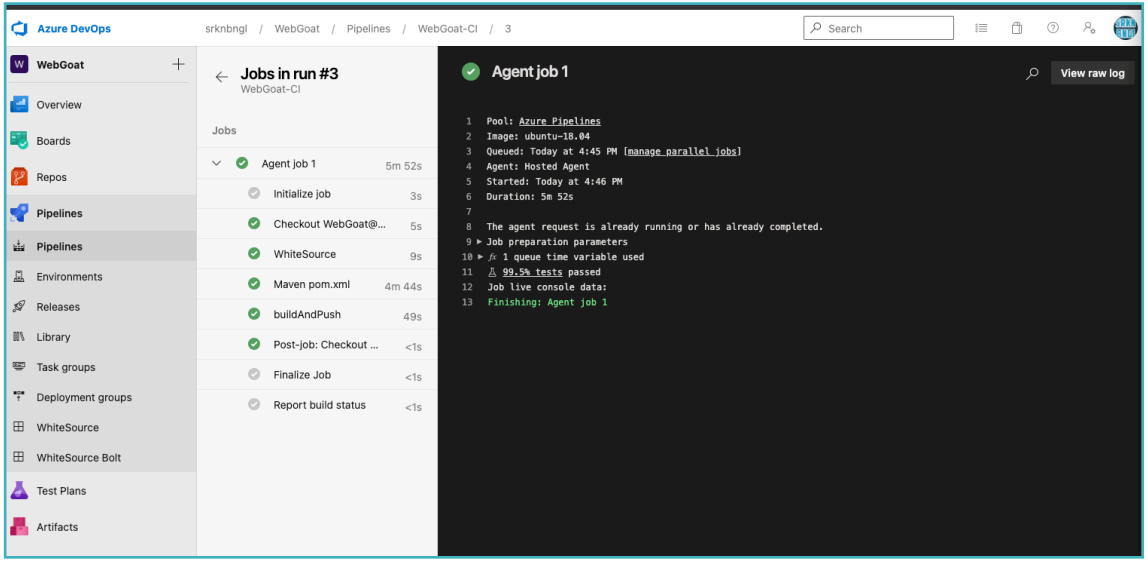


WhiteSource Bolt Documentation :
https://marketplace.visualstudio.com/items?itemName=whitesource.ws-bolt

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Whitesource Bolt: check open source components

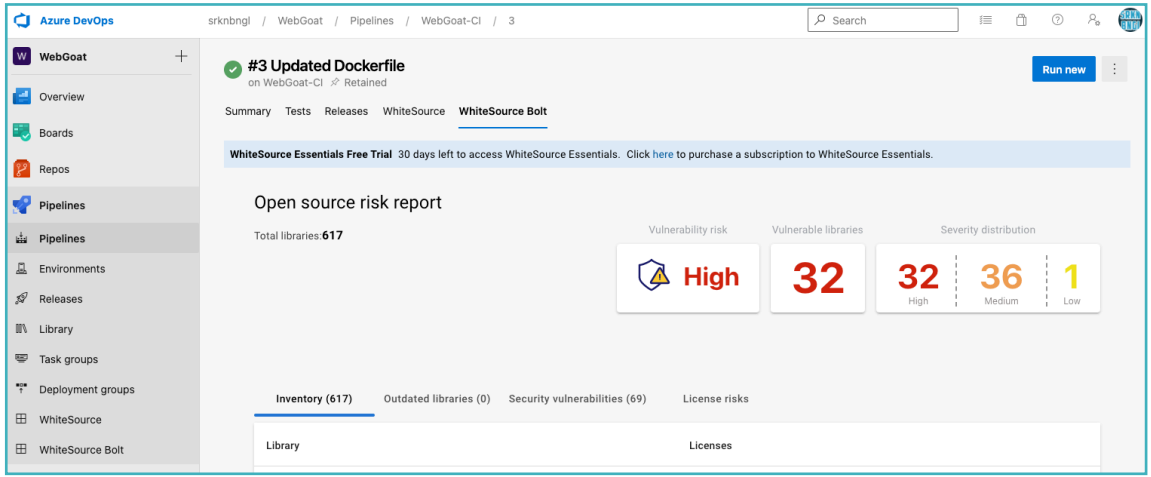


After adding Whitesource Bolt extension rebuild pipeline to get reports.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Whitesource Bolt: Check Vulnerability Reports



After build pipeline Whitesource Bolt extension creates vulnerability reports.

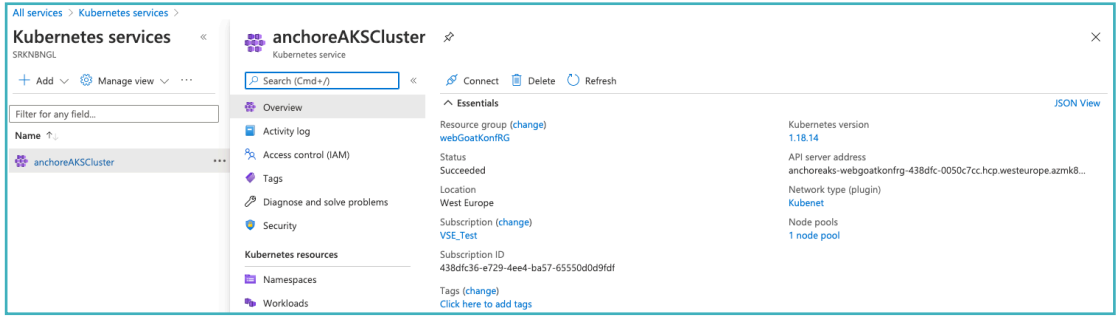
DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Create AKS Cluster on WebGout Resource Group

```
srknbngl@t-800 ~ az aks create --resource-group webGoatKonfRG --name anchoreAKSCluster --node-count 3 --enable-addons monitoring --generate-ssh-keys
AAD role propagation done[############################################]  100.0000%{
  "aadProfile": null,
  "addonProfiles": {
    "KubeDashboard": {
      "config": null,
      "enabled": false,
      "identity": null
```

azure-cli command
az aks create --resource-group webGoatKonfRG --name anchoreAKSCluster --node-count 3 --enable-addons monitoring --generate-ssh-keys



Install Anchore server on AKS with Helm :
https://anchore.com/blog/azure-anchore-kubernetes-service-cluster-with-helm/

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Get credentials for AKS cluster and check credentials

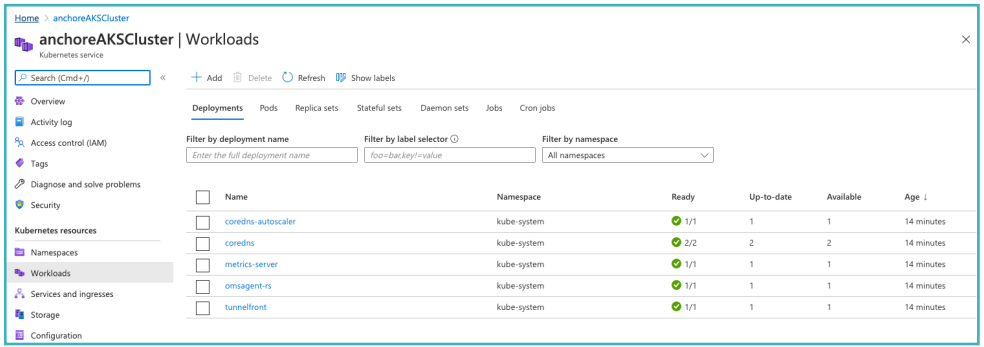
```
srknbngl@t-800 ~ az aks get-credentials --resource-group webGoatKonfRG --name anchoreAKSCluster
Merged "anchoreAKSCluster" as current context in /Users/srknbngl/.kube/config
srknbngl@t-800 ~ kubectl get nodes
NAME                                STATUS   ROLES   AGE   VERSION
aks-nodepool1-80324486-vmss000000   Ready    agent   12m   v1.18.14
aks-nodepool1-80324486-vmss000001   Ready    agent   12m   v1.18.14
aks-nodepool1-80324486-vmss000002   Ready    agent   12m   v1.18.14
srknbngl@t-800 ~ az aks browse --resource-group webGoatKonfRG --name anchoreAKSCluster
Kubernetes resources view on https://portal.azure.com/#resource/subscriptions/438dfc36-e729-4ee4-ba57-65550d
ter/workloads
srknbngl@t-800 ~
```

cli commands

Get credentials : az aks get-credentials --resource-group webGoatKonfRG --name anchoreAKSCluster

Get nodes : kubectl get nodes

Browse aks cluster : az aks browse --resource-group webGoatKonfRG --name anchoreAKSCluster



DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : HELM Configuration

```
srknbngl@t-800 ~ sudo vim helm-rbac.yaml
Password:
srknbngl@t-800 ~ kubectl apply -f helm-rbac.yaml
serviceaccount/tiller created
clusterrole.rbac.authorization.k8s.io/kube-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/tiller created
clusterrolebinding.rbac.authorization.k8s.io/rook-operator created
srknbngl@t-800 ~
```

- create a file name helm-rbac.yaml
- execute **sudo vim helm-rbac.yaml**
- add configuration same as given code block
- execute **kubectl apply - f helm-brac.yaml** command

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: tiller
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: kube-dashboard
rules:
- apiGroups: ["*"]
  resources: ["*"]
  verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: tiller
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
  - kind: ServiceAccount
    name: tiller
    namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: rook-operator
  namespace: rook-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: kube-dashboard
subjects:
- kind: ServiceAccount
  name: kubernetes-dashboard
  namespace: kube-system
```

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Install Anchore

```
srknbngl@t-800 ~ helm repo add anchore https://charts.anchore.io
"anchore" already exists with the same configuration, skipping
srknbngl@t-800 ~ helm install anchore-demo anchore/anchore-engine
NAME: anchore-demo
LAST DEPLOYED: Sat Feb 20 17:39:42 2021
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
To use Anchore Engine you need the URL, username, and password to access the API.

Anchore Engine can be accessed via port 8228 on the following DNS name from within the cluster:
anchore-demo-anchore-engine-api.default.svc.cluster.local

Here are the steps to configure the anchore-cli (`pip install anchorecli`). Use these same values for direct API access as well.

To configure your anchore-cli run:

    ANCHORE_CLI_USER=admin
    ANCHORE_CLI_PASS=$(kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo)

Using the service endpoint from within the cluster you can use:
    ANCHORE_CLI_URL=http://anchore-demo-anchore-engine-api.default.svc.cluster.local:8228/v1/
```

cli commands
Add anchore helm repo : helm repo add anchore https://charts.anchore.io
Install anchore helm chart : helm install anchore-demo anchore/anchore-engine

```
srknbngl@t-800 ~ kubectl get deployments
NAME                                      READY   UP-TO-DATE   AVAILABLE   AGE
anchore-demo-anchore-engine-analyzer      1/1     1            1           4m50s
anchore-demo-anchore-engine-api           1/1     1            1           4m50s
anchore-demo-anchore-engine-catalog       1/1     1            1           4m50s
anchore-demo-anchore-engine-policy        1/1     1            1           4m50s
anchore-demo-anchore-engine-simplequeue   1/1     1            1           4m50s
anchore-demo-postgresql                   1/1     1            1           4m50s
srknbngl@t-800 ~
srknbngl@t-800 ~ kubectl expose deployment anchore-demo-anchore-engine-api --type=LoadBalancer --name=anchore-engine --port=8228
service/anchore-engine exposed
srknbngl@t-800 ~
srknbngl@t-800 ~ kubectl get service anchore-engine
NAME             TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
anchore-engine   LoadBalancer   10.0.180.42   20.73.32.172   8228:30477/TCP   46s
srknbngl@t-800 ~
srknbngl@t-800 ~
```

cli commands
Get deployments : kubectl get deployments
Expose API port externally : kubectl expose deployment anchore-demo-anchore-engine-api —type=LoadBalancer --name=anchore-engine —port=8228
View service and external IP : kubectl get service anchore-engine

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Check Anchore Status

```
srknbngl0t-800 ~ anchore-cli --url http://20.73.32.172:8228/v1 --u admin --p $(kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo) system status
Service catalog (anchore-demo-anchore-engine-catalog-569b49c584-sdgns, http://anchore-demo-anchore-engine-catalog:8082): up
Service analyzer (anchore-demo-anchore-engine-analyzer-7fd5544998-75ph7, http://anchore-demo-anchore-engine-analyzer:8084): up
Service policy_engine (anchore-demo-anchore-engine-policy-56dc54d448-fmkmn, http://anchore-demo-anchore-engine-policy:8087): up
Service simplequeue (anchore-demo-anchore-engine-simplequeue-66cbbdc965-lmf9w, http://anchore-demo-anchore-engine-simplequeue:8083): up
Service apiext (anchore-demo-anchore-engine-api-9b694647b-dpjzl, http://anchore-demo-anchore-engine-api:8228): up

Engine DB Version: 0.0.14
Engine Code Version: 0.9.1
srknbngl0t-800 ~ ✔ 468 18:12:51
```

cli commands

Check status of Anchore : anchore-cli --url http://20.73.32.172:8228/v1 --u admin --p $(kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo) system status

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Add Registry and Image to Anchore

```
srknbngl@t-800 ~ anchore-cli --url http://20.73.32.172:8228/v1 --u admin --p $(kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo) registry add --registry-type docker_v2 webgoatkonfacr.azurecr.io webGoatKonfACR LHZiFHz/YeuQL=wQhHgRGva5MDOMCWdE
Registry: webgoatkonfacr.azurecr.io
Name: webgoatkonfacr.azurecr.io
User: webGoatKonfACR
Type: docker_v2
Verify TLS: True
Created: 2021-02-20T15:27:29Z
Updated: 2021-02-20T15:27:29Z

srknbngl@t-800 ~ ✔ 470 18:27:29
```

cli commands

Add Registry to Anchore : anchore-cli --url http://20.73.32.172:8228/v1 --u admin --p $(kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo) registry add --registry-type docker_v2 webgoatkonfacr.azurecr.io webGoatKonfACR LHZiFHz/YeuQL=wQhHgRGva5MDOMCWdE

```
srknbngl@t-800 ~ anchore-cli --url http://20.73.32.172:8228/v1 --u admin --p $(kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo) image add webgoatkonfacr.azurecr.io/webgoat-8.1:Latest
Image Digest: sha256:e79e82dee1c6a74ef1fbe95e52a765b8e06d63e503638d5c4f7dced31911ac72
Parent Digest: sha256:e79e82dee1c6a74ef1fbe95e52a765b8e06d63e503638d5c4f7dced31911ac72
Analysis Status: not_analyzed
Image Type: docker
Analyzed At: None
Image ID: 66087b1e8282fbc0de37408fd457e59e217e41704bfd9eab97a2578acf891f4d
Dockerfile Mode: None
Distro: None
Distro Version: None
Size: None
Architecture: None
Layer Count: None

Full Tag: webgoatkonfacr.azurecr.io/webgoat-8.1:Latest
Tag Detected At: 2021-02-20T15:31:02Z

srknbngl@t-800 ~ ✔ 472 18:31:02
```

cli commands

Add Image to Anchore : anchore-cli --url http://20.73.32.172:8228/v1 --u admin --p $(kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo) image add webgoatkonfacr.azurecr.io/webgoat-8.1:Latest

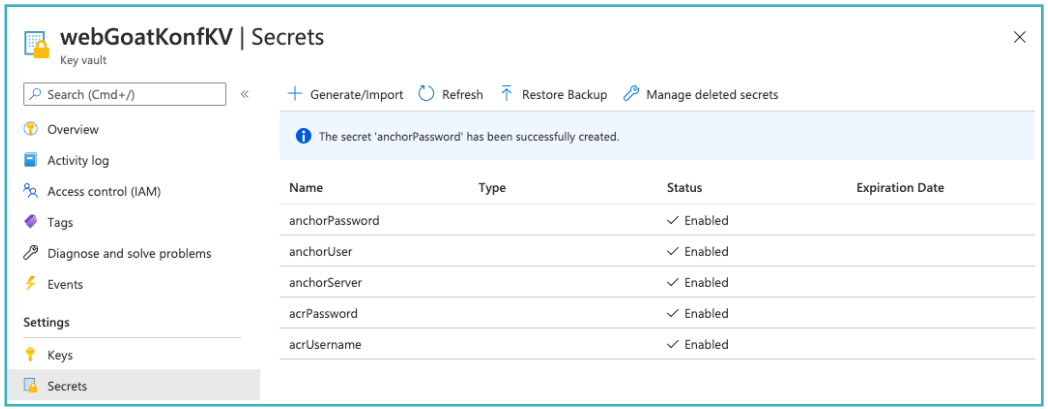
DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Add credentials to Azure Vault

```
srknbngl@t-800 ~ kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo
nyX2cKP4g1Nenm2Y35AcUrgv8c8le5rZ
srknbngl@t-800 ~
```

cli command for get anchore-cli password
kubectl get secret --namespace default anchore-demo-anchore-engine -o jsonpath="{.data.ANCHORE_ADMIN_PASSWORD}" | base64 --decode; echo

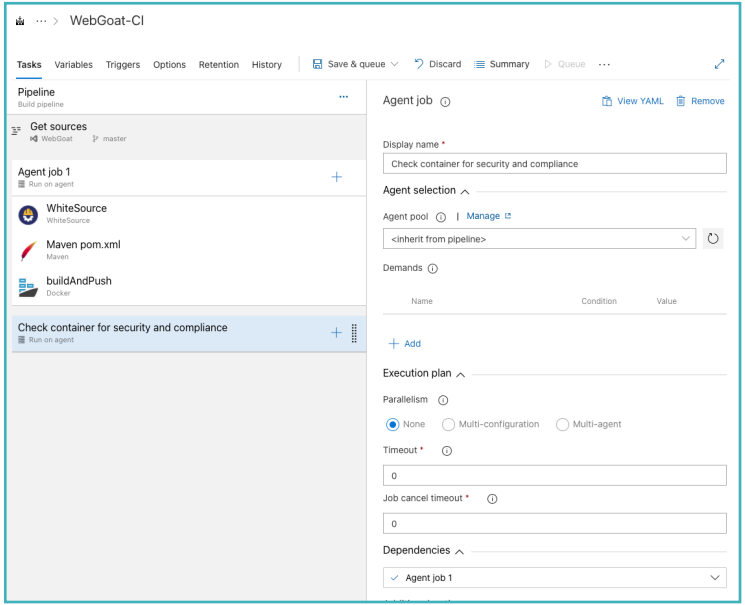


For sensitive variables add **anchorPassword , anchorUser , anchorServer** , azure key vault wallet to use in vulnerability scan task. Don't forget to link variable group to pipeline. And also don't forget to add non-sensitive variables to pipeline variables as **imageRepository** and **acrRegistry**.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Add new job agent to build pipeline

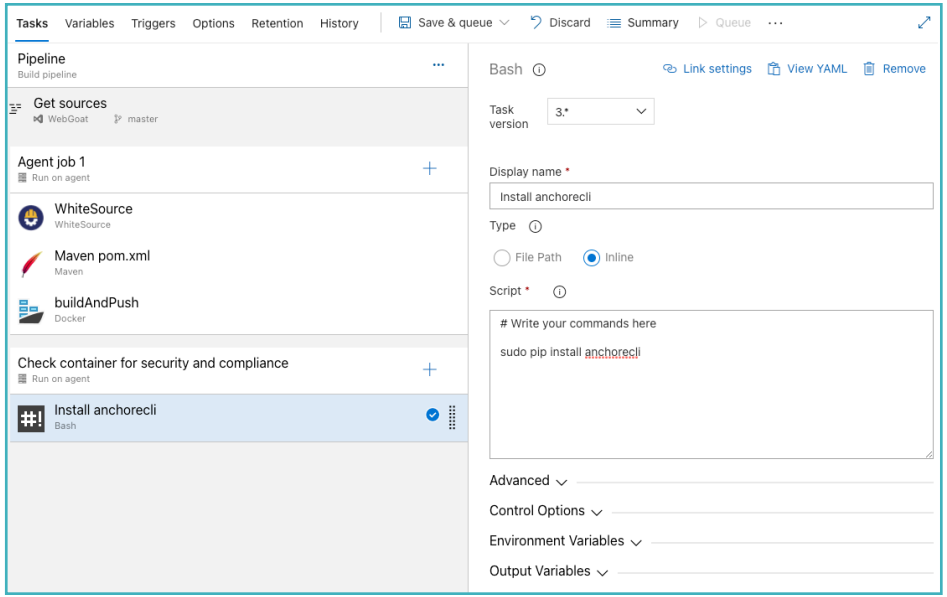


In the Dependencies select agent job, which is responsible for compiling and pushing WebGoat container to ACR. It is important to have that job finished before scans gets triggered as we want to scan the lates image.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : First Task - Install anchore cli

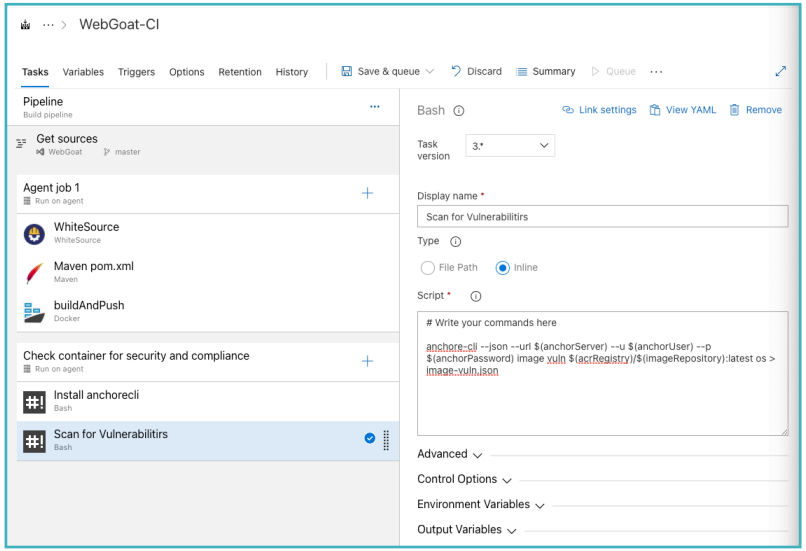


First **Bash** task is reponsible for installing anchore-cli tool on agent.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Second Task - Vulnerability scanner



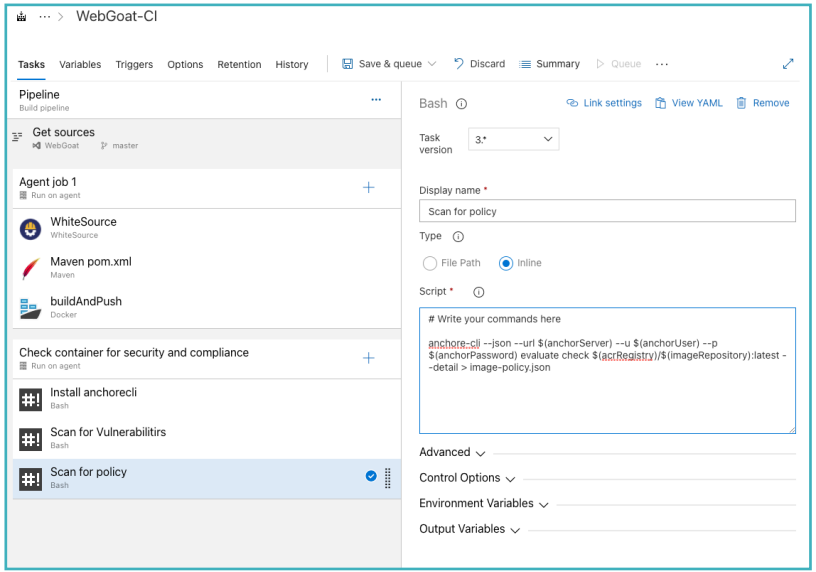
Bash task command
anchore-cli --json --url $(anchorServer) --u $(anchorUser) --p $(anchorPassword) image vuln $(acrRegistry)/$(imageRepository):Latest os > image-vuln.json

Second **Bash** task is reponsible for scan for vulnerabilities in the image.

DEMO: Implement Security in Azure DevOps CI/CD

kloia

Anchore : Third Task - Policy scanner

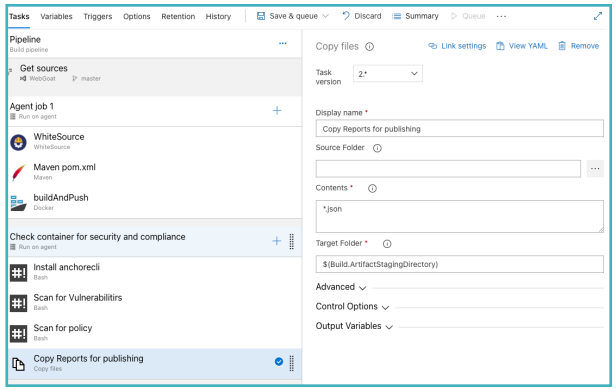


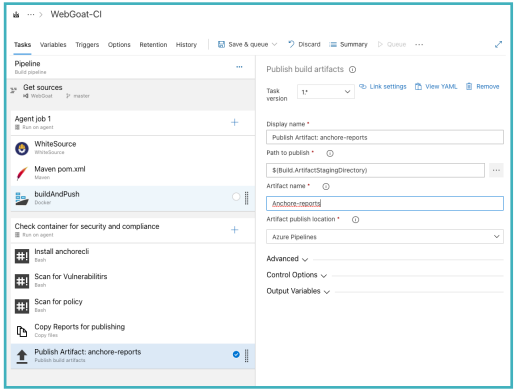
Bash task command
anchore-cli --json --url $(anchorServer) --u $(anchorUser) --p $(anchorPassword) evaluate check $(acrRegistry)/$(imageRepository):Latest --detail > image-policy.json

Third **Bash** task is reponsible for scan for policies in the image.

DEMO: Implement Security in Azure DevOps CI/CD

Anchore : Generate Anchore Report



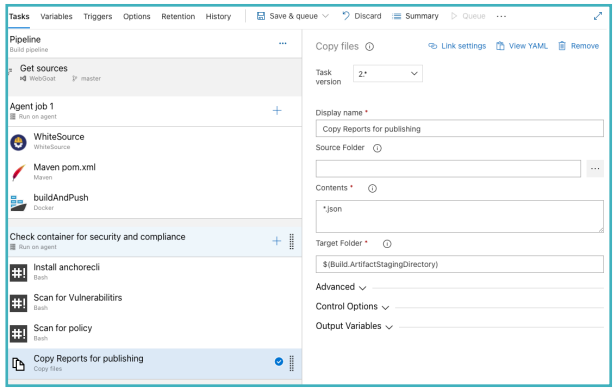


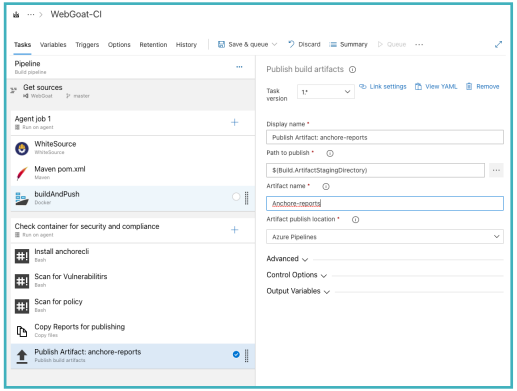
Both scan will produce json reports and we need to publish them as pipeline artifact, which can be downloaded after. To do this add two tasks:

- **Copy Files** : *contents* => *.json , *Target Folder* => $(Build.ArtifactStagingDirectory)
- **Publish build artifacts : ***Path to publish* => $(Build.ArtifactStagingDirectory) , *Artifact name* => Anchore-reports

DEMO: Implement Security in Azure DevOps CI/CD

Anchore : Generate Anchore Report





Both scan will produce json reports and we need to publish them as pipeline artifact, which can be downloaded after. To do this add two tasks:

- **Copy Files** : *contents* => *.json , *Target Folder* => $(Build.ArtifactStagingDirectory)
- **Publish build artifacts : ***Path to publish* => $(Build.ArtifactStagingDirectory) , *Artifact name* => Anchore-reports

Come Join Us, The Kloians:

We are a team of experienced and happy high achievers,

Knowing that **'Great vision without great people is irrelevant.'**

So:

- Talents determine the positions, we do not miss the best,
- We hire the best and providing them the best, pecuniary and non-pecuniary,
- We set high standards, so you better have those as well,
- Stunning colleagues means a great workplace, and the motive to be online.

If you have the Stunning **'Kloian'** essence or potential by being:

- No Bullshit
- Self-motivated and Reliable
- Optimistic and Focused
- Happy and Fun
- Flexible and Trustworthy
- Willing and Creative
- Open minded and Kind (Majority of above, if not all.) 🙂

You are more than welcome to apply: career@kloia.com

Q & A

kloia

Thank you for listening.

RESOURCES MENTIONED IN THIS SESSION

https://github.com/kloia/dotnetkonf21-Azure-DevSecOp

kloia.com blog.kloia.com @kloia_com