

Screenshots of steps for creating and launching EC2 instance :-

The screenshot shows the 'Endpoint settings' configuration page. Under 'Name tag - optional', the value 'D15A 61 Ubuntu endpoint' is entered. In the 'Service category' section, the 'EC2 Instance Connect Endpoint' option is selected, highlighted with a blue border. The VPC dropdown is set to 'vpc-0b4df95bddc92aea'. At the bottom, there are links for CloudShell, Feedback, and various AWS terms.

The screenshot shows the EC2 instance configuration page for instance 'i-0796782f492cf0a37'. Under 'Connection Type', the 'Connect using EC2 Instance Connect Endpoint' option is selected. The 'Private IP address' field contains '172.31.53.213'. The 'EC2 Instance Connect Endpoint' section lists 'eice-06edebadb8ba9b4'. The 'Username' field is set to 'D15A_61'. The 'Max tunnel duration (seconds)' field is set to '3600'. A note at the bottom states: 'Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom, there are links for CloudShell, Feedback, and various AWS terms.

root@ip-172-31-42-7:~\$ mkdir temp
root@ip-172-31-42-7:~\$ cd temp
root@ip-172-31-42-7:~/temp\$ wget https://d15a-jai-61.github.io/IP-Exp2/
--2024-08-12 11:00:29-- https://d15a-jai-61.github.io/IP-Exp2/
Resolving d15a-jai-61.github.io (d15a-jai-61.github.io)... 185.199.108.153, 185.199.109.153, 185.199.110.153, ...
Connecting to d15a-jai-61.github.io (d15a-jai-61.github.io)|185.199.108.153|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4359 (4.3K) [text/html]
Saving to: 'index.html'

index.html 100%[=====] 4.26K --.-KB/s in 0s
2024-08-12 11:00:29 (41.9 MB/s) - 'index.html' saved [4359/4359]
root@ip-172-31-42-7:~/temp#

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
root@ip-172-31-42-7:~/temp# ls -lrt  

total 8  

-rw-r--r-- 1 root root 4359 Aug 12 09:23 index.html  

root@ip-172-31-42-7:~/temp#
```

```
root@ip-172-31-42-7:~$ cd temp  

root@ip-172-31-42-7:~/temp$ ls  

index.html  

root@ip-172-31-42-7:~/temp$ service apache2 start  

root@ip-172-31-42-7:~/temp$ service apache2 status  

● apache2.service - The Apache HTTP Server  

   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)  

   Active: active (running) since Mon 2024-08-12 10:47:34 UTC; 51min ago  

     Docs: https://httpd.apache.org/docs/2.4/  

   Process: 493 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)  

 Main PID: 519 (apache2)  

    Tasks: 55 (limit: 1130)  

   Memory: 7.8M (peak: 8.0M)  

      CPU: 203ms  

 CGroup: /system.slice/apache2.service  

         ├─519 /usr/sbin/apache2 -k start  

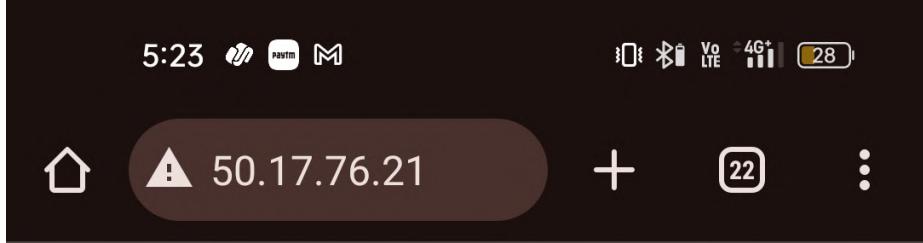
         ├─520 /usr/sbin/apache2 -k start  

         └─521 /usr/sbin/apache2 -k start  

Aug 12 10:47:33 ip-172-31-42-7 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 12 10:47:34 ip-172-31-42-7 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-42-7:~/temp#
```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Petco

A photograph of a man with short hair and a tattoo on his left arm, wearing a white t-shirt and white pants, sitting on a wooden chair. He is petting a brown and white dog that is standing on its hind legs, reaching up towards him. They are in a room with a corkboard wall, a small table with a vase, and a window with white curtains in the background.

About us

Petco is a category-defining health and wellness company focused on improving the lives of pets, pet parents and our own Petco partners. Since our founding in 1965, we've been trailblazing new standards in pet care, delivering comprehensive wellness solutions through our products and services, and creating communities that deepen the pet-parent bond.

We employ more than 29,000 partners nationwide and operate more than 1,500 Petco locations across the U.S., Mexico and Puerto Rico — including a growing network of more than 200 in-store veterinary hospitals — and offer a complete online resource for pet health and wellness at petco.com and on the Petco app.

In tandem with Petco Love, an independent nonprofit organization, we have helped find homes for more than 7 million animals through in-store adoption events.

Working at Petco



Screenshots for setting up and deploying Elastic Beanstalk application :-

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

Web server environment
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
 Maximum length of 100 characters.

► Application tags (optional)

Platform type

Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform
 ▾

Platform branch
 ▾

Platform version
 ▾

Application code [Info](#)

Sample application

Existing version
Application versions that you have uploaded.

Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

	Region	Subnet	CIDR Block
<input type="checkbox"/>	us-east-1f	subnet-03b751cf5...	172.31.64.0/20
<input type="checkbox"/>	us-east-1a	subnet-05ff6de98...	172.31.16.0/20
<input type="checkbox"/>	us-east-1c	subnet-089dbc2d1...	172.31.0.0/20
<input type="checkbox"/>	us-east-1b	subnet-0d4591b4f...	172.31.32.0/20
<input type="checkbox"/>	us-east-1d	subnet-0f341fff62...	172.31.80.0/20

Enable database

Restore a snapshot - *optional*

Restore an existing snapshot from a previously used database.

Snapshot

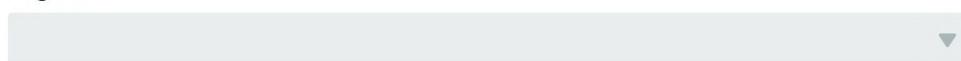
None



Database settings

Choose an engine and instance type for your environment's database.

Engine



Engine version



Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

No tags associated with the resource.

[Add new tag](#)

You can add 50 more tags.

[Cancel](#)

[Skip to review](#)

[Previous](#)

[Next](#)

Configure updates, monitoring, and logging - optional [Info](#)

▼ Monitoring [Info](#)

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#)

System

- Basic
 Enhanced

Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming

- Activated (standard CloudWatch charges apply.)

Retention

7

Lifecycle

▼ Managed platform updates [Info](#)

Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates

- Activated

Weekly update window

Tuesday at 01 : 14 UTC

Update level

Minor and patch

Instance replacement

If enabled, an instance replacement will be scheduled if no other updates are available.

- Activated

Ignore health check	Instance replacement
false	false

Platform software

Lifecycle	Log streaming	Proxy server
false	Deactivated	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor

X-Ray enabled
Deactivated

Environment properties

Key	Value
No environment properties	
There are no environment properties defined	

Cancel Previous Submit

Elastic Beanstalk is launching your environment. This will take a few minutes. (i) X

[Elastic Beanstalk](#) > [Environments](#) > D15A-Jai-61-env

D15A-Jai-61-env Info

C
Actions ▾
Upload and deploy

Environment overview

Health	Environment ID
⊖ Unknown	Copy e-ixwgm2txnf
Domain	Application name
-	D15A-Jai-61

Platform Change version

Platform	Node.js 20 running on 64bit Amazon Linux 2023/6.2.0
Running version	-
Platform state	Supported

[Events](#) [Health](#) [Logs](#) [Monitoring](#) [Alarms](#) [Managed updates](#) [Tags](#)

Events (2) Info C

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS Elastic Beanstalk console. At the top, there are two green status bars: the first indicates a successful upload and deployment of 'app.js', and the second indicates the environment update was completed successfully. Below these, the navigation bar shows 'Elastic Beanstalk > Environments > D15A-Jai-61-env-1'. The main content area is titled 'D15A-Jai-61-env-1' with an 'Info' link. To the right are 'Actions' and 'Upload and deploy' buttons. The left side of the screen has a large green banner with the text 'Congratulations' and a message stating that the first AWS Elastic Beanstalk Node.js application is now running. The URL 'd15a-jai-61-env-1.eba-fwpthi36.us-east-1.elasticbeanstalk.com' is displayed above the banner. The right side of the screen contains a 'What's Next?' section with links to various AWS Elastic Beanstalk documentation pages.

Successfully uploaded file app.js to S3, created application version and started deployment with new application version

Environment update successfully completed.

Elastic Beanstalk > Environments > D15A-Jai-61-env-1

D15A-Jai-61-env-1 [Info](#)

[Actions](#) [Upload and deploy](#)

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploying an Express Application to AWS Elastic Beanstalk](#)
- [Deploying an Express application with clustering to Elastic Beanstalk](#)
- [Customizing and Configuring a Node.js Container](#)
- [Working with Logs](#)

Screenshots for Experiment 3.

Instances (2) Info		Last updated less than a minute ago	C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
<input type="checkbox"/>	Name ▾	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	▼
<input type="checkbox"/>	master	i-0b006eb4332c1e50a	Running ? Q	t2.micro	2/2 checks passed View alarms +	+	us-east-2b	▼
<input type="checkbox"/>	worker-1	i-00030b5dae5b8bf5c	Running ? Q	t2.micro	2/2 checks passed View alarms +	+	us-east-2b	▼

Inbound rules (1)							C	Manage tags	Edit inbound rules
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	▼	▼	▼
<input type="checkbox"/>	-	sgr-03d9566e9c025de...	IPv4	All traffic	All	All	▼	▼	▼

```
ubuntu@ip-172-31-18-146:~$ sudo hostnamectl set-hostname worker
```

```
ubuntu@master:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
```

```
ubuntu@master:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
```

```
Processing triggers for dbus (1.14.10-4ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Jai Talreja 61 D15A

```
ubuntu@master:~$ sudo systemctl enable docker
ubuntu@master:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-09-15 15:23:25 UTC; 5min ago
     TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
 Main PID: 2653 (dockerd)
    Tasks: 8
   Memory: 31.8M (peak: 33.4M)
     CPU: 299ms
    CGroup: /system.slice/docker.service
            └─2653 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

```
ubuntu@worker:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
```

```
ubuntu@master:~$ sudo apt-get install -y apt-transport-https ca-certificates curl gpg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
gpg is already the newest version (2.4.4-2ubuntu17).
gpg set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 904 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.
```

```
ubuntu@master:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@master:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/
ubuntu@master:~$ sudo apt-get update
```

```
ubuntu@master:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 130 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb
Fetched 87.4 MB in 1s (70.9 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68112 files and directories currently installed.)
```

```
ubuntu@master:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@master:~$
```

```
ubuntu@master:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
  [WARNING Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
  [WARNING FileExisting-socat]: socat not found in system path
```

Jai Talreja 61 D15A

```
Your Kubernetes control-plane has initialized successfully!
```

```
To start using your cluster, you need to run the following as a regular user:
```

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
Alternatively, if you are the root user, you can run:
```

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

```
You should now deploy a pod network to the cluster.
```

```
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/
```

```
Then you can join any number of worker nodes by running the following on each as root:
```

```
kubeadm join 172.31.25.196:6443 --token wb5xks.0x5kyf3rpycusd60 \
--discovery-token-ca-cert-hash sha256:4fa35dc1d8d64ac6f31ebe712f3ec1d5c8a38ba8a8fb316579c08b671eabbd06
```

```
ubuntu@master:~$ mkdir -p $HOME/.kube
ubuntu@master:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@master:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@master:~$ 
```

```
ubuntu@master:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yaml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

```
ubuntu@worker:~$ sudo kubeadm join 172.31.25.196:6443 --token wb5xks.0x5kyf3rpycusd60 --discovery-token-ca-cert-hash sha256:4fa35dc1d8d64ac6f31ebe712f3ec1d5c8a38ba8a8fb316579c08b671eabbd06 --ignore-preflight-errors=all
[preflight] Running pre-flight checks
[WARNING FileAvailable--etc-kubernetes-kubelet.conf]: /etc/kubernetes/kubelet.conf already exists
[WARNING FileExisting--socat]: socat not found in system path
[WARNING Port-10250]: Port 10250 is in use
[WARNING FileAvailable--etc-kubernetes-pki-ca.crt]: /etc/kubernetes/pki/ca.crt already exists
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 503.838994ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

this node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

```
ubuntu@master:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME          READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-8z8zq   1/1     Running   1 (6m12s ago)  21m
kube-flannel   kube-flannel-ds-src45   1/1     Running   2 (5m16s ago)  16m
kube-system    coredns-7c65d6cf9-4lmp4  1/1     Running   1 (6m12s ago)  30m
kube-system    coredns-7c65d6cf9-fnrrt  1/1     Running   1 (6m12s ago)  30m
kube-system    etcd-master           1/1     Running   1 (6m12s ago)  30m
kube-system    kube-apiserver-master  1/1     Running   1 (6m12s ago)  30m
kube-system    kube-controller-manager 1/1     Running   1 (6m12s ago)  30m
kube-system    kube-proxy-hw6zm       0/1     CrashLoopBackOff 9 (29s ago)   16m
kube-system    kube-proxy-x6lmb       1/1     Running   11 (2m8s ago)  30m
kube-system    kube-scheduler-master 1/1     Running   1 (6m12s ago)  30m
ubuntu@master:~$ kubectl delete node worker
node "worker" deleted
ubuntu@master:~$ kubectl get nodes
NAME      STATUS  ROLES   AGE   VERSION
master   Ready   control-plane 35m   v1.31.1
worker   Ready   <none>  87s   v1.31.1
```

Jai Talreja 61 D15A

Screenshots for experiment 4

```
ubuntu@master:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
ubuntu@master:~$ █
```

```
ubuntu@master:~$ sudo nano deploy.yaml
```

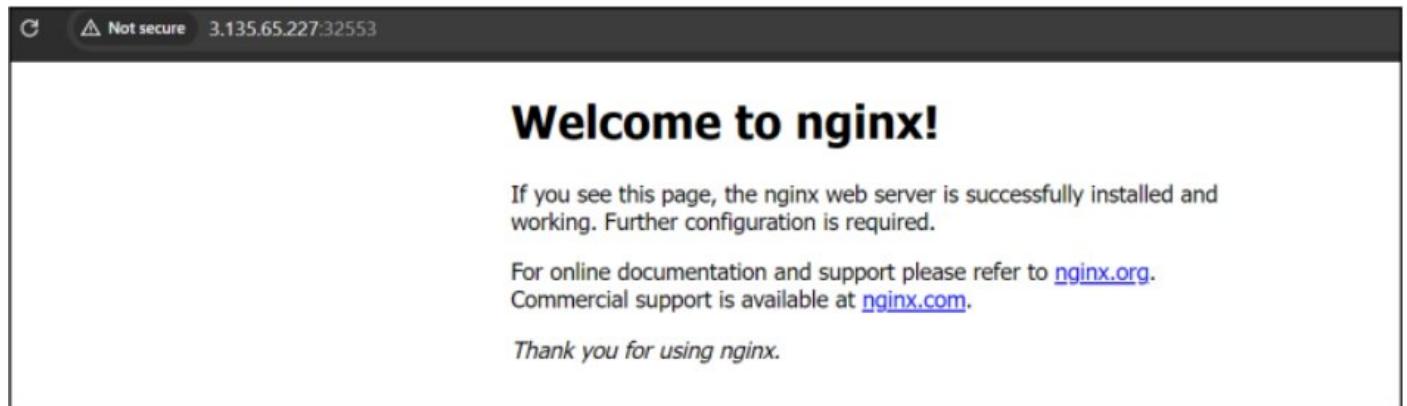
```
ubuntu@master:~$ cat deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
ubuntu@master:~$ █
```

```
ubuntu@master:~$ kubectl create -f deploy.yaml
deployment.apps/nginx-deployment created
```

```
ubuntu@master:~$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx          1/1     1            1           40m
nginx-deployment 3/3     3            3           23m
ubuntu@master:~$
```

```
ubuntu@master:~$ kubectl expose deployment.apps/nginx-deployment \
> --type="LoadBalancer"
service/nginx-deployment exposed
ubuntu@master:~$
```

```
ubuntu@master:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1      <none>        443/TCP      140m
nginx-deployment  LoadBalancer  10.96.232.159  <pending>    80:32553/TCP  80s
```



Screenshots for installing, configuring and running Terraform using PowerShell :-

Name	Date modified	Type
📁 Intel	15-08-2024 23:32	File folder
📁 PerfLogs	18-08-2024 20:59	File folder
📁 Program Files	16-08-2024 00:31	File folder
📁 Program Files (x86)	15-08-2024 23:32	File folder
▶️ Terraform	20-08-2024 01:31	File folder
▶️ Users	15-08-2024 23:32	File folder
▶️ Windows	15-08-2024 23:35	File folder
▶️ Windows.old	18-08-2024 19:59	File folder
▶️ LastBak	15-08-2024 23:08	Configuration sett...
▶️ npkey	06-08-2024 12:43	Text Document

Environment Variables	
User variables for 505-22	
Variable	Value
OneDrive	C:\Users\505-22\OneDrive
Path	C:\Terraform
TEMP	C:\Users\505-22\AppData\Local\Temp
TMP	C:\Users\505-22\AppData\Local\Temp
New... Edit... Delete	
System variables	
Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	16
OS	Windows_NT
Path	C:\Terraform
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
New... Edit... Delete	
OK Cancel	

```
[+] Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console        Try Terraform expressions at an interactive command prompt
  fmt            Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get            Install or upgrade remote Terraform modules
  graph          Generate a Graphviz graph of the steps in an operation
  import         Associate existing infrastructure with a Terraform resource
  login          Obtain and save credentials for a remote host
  logout         Remove locally-stored credentials for a remote host
  metadata       Metadata related commands
  output         Show output values from your root module
  providers     Show the providers required for this configuration
  refresh        Update the state to match remote systems
  show           Show the current state or a saved plan
  state          Advanced state management
  taint          Mark a resource instance as not fully functional
  test           Execute integration tests for Terraform modules
  untaint       Remove the 'tainted' state from a resource instance
  version        Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
```



```
jai@fedora:~/Documents/terraform_scripts/docker$ sudo systemctl start docker
[sudo] password for jai:
jai@fedora:~/Documents/terraform_scripts/docker$ terraform plan
```

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach          = false
    + bridge          = (known after apply)
    + command         = (known after apply)
    + container_logs = (known after apply)
    + entrypoint      = (known after apply)
    + env             = (known after apply)
    + exit_code       = (known after apply)
    + gateway         = (known after apply)
    + hostname        = (known after apply)
    + id              = (known after apply)
    + image           = (known after apply)
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length= (known after apply)
    + ipc_mode        = (known after apply)
    + log_driver      = (known after apply)
    + logs            = false
    + must_run        = true
    + name            = "foo"
    + network_data   = (known after apply)
    + read_only       = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
    + runtime         = (known after apply)
    + security_opts  = (known after apply)
    + shm_size        = (known after apply)
    + start           = true
    + stdin_open      = false
    + stop_signal     = (known after apply)
    + stop_timeout    = (known after apply)
    + tty              = false}
```

```
jai@fedora:~/Documents/terraform_scripts/docker$ terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:b1e9cef3f2977f8bdd19eb9ae04f
```

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach          = false
    + bridge          = (known after apply)
    + command         = (known after apply)
    + container_logs = (known after apply)
    + entrypoint      = (known after apply)
    + env             = (known after apply)
    + exit_code       = (known after apply)
    + gateway         = (known after apply)
    + hostname        = (known after apply)
    + id              = (known after apply)
    + image           = "sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c56"
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode        = (known after apply)
    + log_driver      = (known after apply)
    + logs            = false
    + must_run        = true
    + name            = "foo"
    + network_data    = (known after apply)
    + read_only       = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
    + runtime          = (known after apply)
    + security_opts   = (known after apply)
    + shm_size         = (known after apply)
    + start            = true
    + stdin_open       = true
    + stop_signal      = (known after apply)
    + stop_timeout     = (known after apply)
    + tty              = true

    + healthcheck (known after apply)
```

```
Plan: 1 to add, 0 to change, 0 to destroy.
```

```
Do you want to perform these actions?
```

```
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.
```

```
Enter a value: yes
```

```
docker_container.foo: Creating...  
docker_container.foo: Creation complete after 0s [id=dfe9d545df9f1bf34ba0dcbaa106e9c52785ea49732a65708f46b1066af2fe8]  
  
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.  
jai@fedora:~/Documents/terraform_scripts/docker$ █
```

```
jai@fedora:~/Documents/terraform_scripts/docker$ terraform destroy  
docker_image.ubuntu: Refreshing state... [id=sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7ubuntu:latest]  
docker_container.foo: Refreshing state... [id=dfe9d545df9f1bf34ba0dcbaa106e9c52785ea49732a65708f46b1066af2fe8]
```

```
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:  
- destroy
```

```
Terraform will perform the following actions:
```

```
# docker_container.foo will be destroyed  
- resource "docker_container" "foo" {  
    - attach           = false -> null  
    - command          = [  
        - "/bin/bash",  
    ] -> null  
    - cpu_shares       = 0 -> null  
    - dns              = [] -> null  
    - dns_opts         = [] -> null  
    - dns_search       = [] -> null  
    - entrypoint       = [] -> null  
    - env              = [] -> null  
    - gateway          = "172.17.0.1" -> null  
    - group_add        = [] -> null  
    - hostname         = "dfe9d545df9f" -> null  
    - id               = "dfe9d545df9f1bf34ba0dcbaa106e9c52785ea49732a65708f46b1066af2fe8" -> null  
    - image             = "sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7" -> null  
    - init              = false -> null  
    - ip_address        = "172.17.0.2" -> null  
    - ip_prefix_length = 16 -> null  
    - ipc_mode          = "private" -> null  
    - links             = [] -> null  
    - log_driver         = "json-file" -> null  
    - log_opts           = {} -> null  
    - logs              = false -> null  
    - max_retry_count   = 0 -> null  
    - memory             = 0 -> null  
    - memory_swap        = 0 -> null  
    - must_run           = true -> null  
    - name              = "foo" -> null  
    - network_data       = [  
        - {  
            - gateway          = "172.17.0.1"  
            - global_ipv6_prefix_length = 0  
            - ip_address        = "172.17.0.2"  
            - ip_prefix_length   = 16
```

```
# docker_image.ubuntu will be destroyed
resource "docker_image" "ubuntu" {
  - id          = "sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7ubuntu:latest" -> null
  - image_id    = "sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7" -> null
  - latest      = "sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7" -> null
  - name        = "ubuntu:latest" -> null
  - repo_digest = "ubuntu@sha256:dfc10878be8d8fc9c61cbff33166cb1d1fe44391539243703c72766894fa834a" -> null
}

Plan: 0 to add, 0 to change, 2 to destroy.

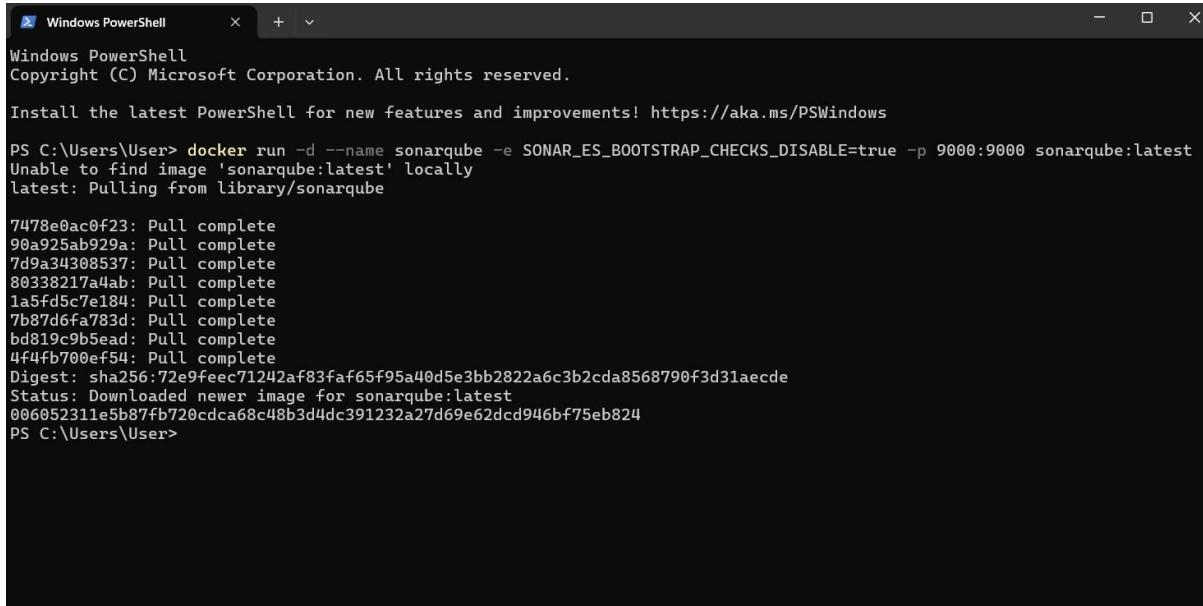
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

docker_container.foo: Destroying... [id=dfe9d545df9f1bf34ba0dcbb6aa106e9c52785ea49732a65708f46b1066af2fe8]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7ubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
jai@fedora:~/Documents/terraform_scripts/docker$ 
```

Screenshots for Experiment 7.



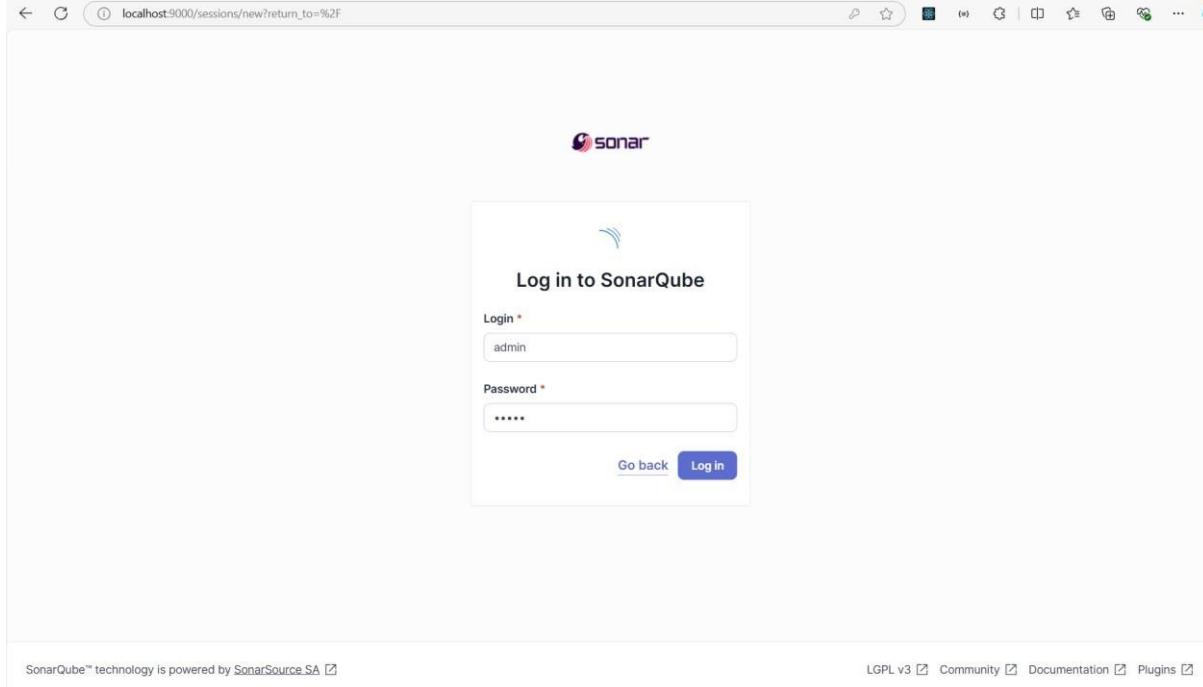
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube

7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
006052311e5b87fb720cdca68c48b3d4dc391232a27d69e62cd946bf75eb824
PS C:\Users\User>
```

Logging in with admin and password.



Create a freestyle project and enter its name and key as "sonarqube-test".

2 of 2

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

Use the global setting

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Define a specific setting for this project

- Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.
- Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.
Recommended for projects following continuous delivery.
- Reference branch
Choose a branch as the baseline for the new code.

Installing and using the “SonarQube scanner for Jenkins” plugin.

Dashboard > Manage Jenkins

Manage Jenkins

New version of Jenkins (2.462.2) is available for download ([changelog](#)). [Or Upgrade Automatically](#)

Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See [the documentation](#). [Manage](#) [Dismiss](#)

Warnings have been published for the following currently installed components:

Jenkins 2.452.3 core and libraries:
[Multiple security vulnerabilities in Jenkins 2.470 and earlier, LTS 2.452.3 and earlier](#)
A fix for this issue is available. Update Jenkins now.

[Configure which of these warnings are shown](#)

System Configuration

- [!\[\]\(3c4a42cd6b30132dab34cb830e31d330_img.jpg\) System](#) Configure global settings and paths.
- [!\[\]\(069d794623cb54cc01fc90a527a3d0f5_img.jpg\) Tools](#) Configure tools, their locations and automatic installers.
- [!\[\]\(c364107d7100f4d4156823e0277c5cf5_img.jpg\) Nodes](#) Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- [!\[\]\(54b77d7355525e6de541afba33c7f2a7_img.jpg\) Clouds](#) Add, remove, and configure cloud instances to provision agents on-demand.
- [!\[\]\(e8be0232e9f47b0bdf315b5a05ff1626_img.jpg\) Plugins](#) ²⁹ Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- [!\[\]\(4ca59552dd93fa7a9296c0d9cd8b378f_img.jpg\) Appearance](#) Configure the look and feel of Jenkins

localhost:8080/manage/pluginManager

Dashboard > SonarQube > Configuration

Configure

Source Code Management

General

Source Code Management **Git**

Build Triggers

Build Environment

Build Steps

Post-build Actions

None

Git

Repositories ?

Repository URL ?
https://github.com/shazforiot/MSBuild_firstproject.git

Credentials ?

- none -

+ Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?
*/master

Save Apply

This screenshot shows the 'Source Code Management' configuration page in SonarQube. The 'Git' tab is selected. In the 'Repositories' section, there is one repository defined with the URL 'https://github.com/shazforiot/MSBuild_firstproject.git'. In the 'Branches to build' section, the 'Branch Specifier' is set to '*/master'. At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > SonarQube > Configuration

Configure

Build Steps

General

Source Code Management

Build Triggers

Build Environment

Build Steps **Execute SonarQube Scanner**

Post-build Actions

JDK ?
JDK to be used for this SonarQube analysis
(Inherit From Job)

Path to project properties ?
[empty input field]

Analysis properties ?
sonar.projectKey=sonarqube-test
sonar.login=admin
sonar.password=sonarqube
sonar.hosturl=http://sonarqube9000

Additional arguments ?
[empty input field]

JVM Options ?
[empty input field]

Save Apply

This screenshot shows the 'Build Steps' configuration page in SonarQube. The 'Execute SonarQube Scanner' step is selected. The configuration includes setting the JDK to inherit from the job, defining a path to project properties, specifying analysis properties (including project key, login, password, and host URL), and defining additional arguments and JVM options. At the bottom, there are 'Save' and 'Apply' buttons.

localhost:9000/dashboard?id=sonarqube-test&codeScope=overall

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More Q ?

star sonarqube-test / main green ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main Version not provided Set as homepage Last analysis 14 minutes ago

Quality Gate Passed

The last analysis has warnings. See details

New Code Overall Code

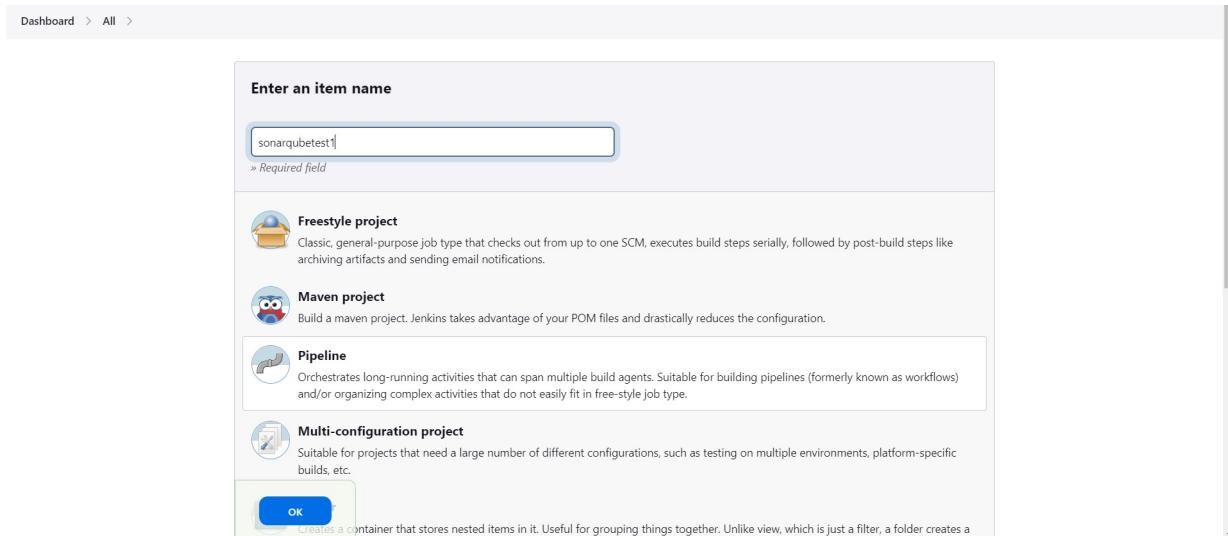
Security	Reliability	Maintainability
0 Open issues (A) 0 H 0 M 0 L	0 Open issues (A) 0 H 0 M 0 L	0 Open issues (A) 0 H 0 M 0 L

Accepted issues	Coverage	Duplications
0 Valid issues that were not fixed	Coverage On 0 lines to cover.	Duplications 0.0% On 86 lines.

Security Hotspots

Screenshots for Experiment 8.

```
erShell does not load commands from the current location by default. If you trust this command, instead type: ".\sonar-s
canner.bat". See "get-help about_Command_Precedence" for more details.
PS C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin> .\sonar-scanner.bat
11:02:02.120 INFO Scanner configuration file: C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin..\conf\sonar-s
canner.properties
11:02:02.124 INFO Project root configuration file: NONE
11:02:02.140 INFO SonarScanner CLI 6.2.0.4584
11:02:02.142 INFO Java 17.0.12 Eclipse Adoptium (64-bit)
11:02:02.142 INFO Windows 11 10.0 amd64
11:02:02.160 INFO User cache: C:\Users\navan\.sonar\cache
11:02:02.644 INFO JRE provisioning: os[windows], arch[amd64]
11:02:06.241 INFO EXECUTION FAILURE
11:02:06.243 INFO Total time: 4.126s
11:02:06.244 ERROR Error during SonarScanner CLI execution
java.lang.IllegalStateException: Error status returned by url [https://api.sonarcloud.io/analysis/jres?os=windows&arch=a
md64]: 401
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callUrl(ServerConnection.java:182)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callApi(ServerConnection.java:145)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callRestApi(ServerConnection.java:123)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreMetadata(JavaRunnerFactory.java:159)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreFromServer(JavaRunnerFactory.java:138)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.createRunner(JavaRunnerFactory.java:85)
        at org.sonarsource.scanner.lib.internal.ScannerEngineLauncherFactory.createLauncher(ScannerEngineLauncherFactory
.java:53)
        at org.sonarsource.scanner.lib.ScannerEngineBootstrapper.bootstrap(ScannerEngineBootstrapper.java:118)
        at org.sonarsource.scanner.cli.Main.analyze(Main.java:75)
        at org.sonarsource.scanner.cli.Main.main(Main.java:63)
11:02:06.246 ERROR
11:02:06.246 ERROR Re-run SonarScanner CLI using the -X switch to enable full debug logging.
PS C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin> |
```



Dashboard > SonarQube Pipeline > Configuration

Pipeline

Configure

Definition

General

Advanced Project Options

Pipeline

```
Script ?  
1 ~ node {  
2   stage('Cloning the GitHub Repo') {  
3     git 'https://github.com/shafirforiot/MSBuild_firstproject.git'  
4   }  
5   stage('SonarQube analysis') {  
6     withSonarQubeEnv('sonarqube-test1') {  
7       bat 'C:/sonar-scanner/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar-scanner.bat' \  
8       -D sonar.login=admin \  
9       -D sonar.password=admin123 \  
10      -D sonar.projectKey=sonarqube-test1 \  
11      -D sonar.exclusions=vendor/**,resources/**,*/*.java \  
12      -D sonar.host.url=http://localhost:9000/'  
13    }  
14  }  
15 }
```

Use Groovy Sandbox ?

Pipeline Syntax

Save Apply

SonarQube Pipeline

Stage View

Average stage times:
(Average full run time: ~34s)

Stage	Run #	Time	Status
Cloning the GitHub Repo	#5	18s	Success
SonarQube analysis	#4	3s	Success
Logs	#3	30s	failed

Actions: Status, Changes, Build Now, Configure, Delete Pipeline, Move, Full Stage View, SonarQube, Stages, Rename, Pipeline Syntax, Build History, trend, Filter...

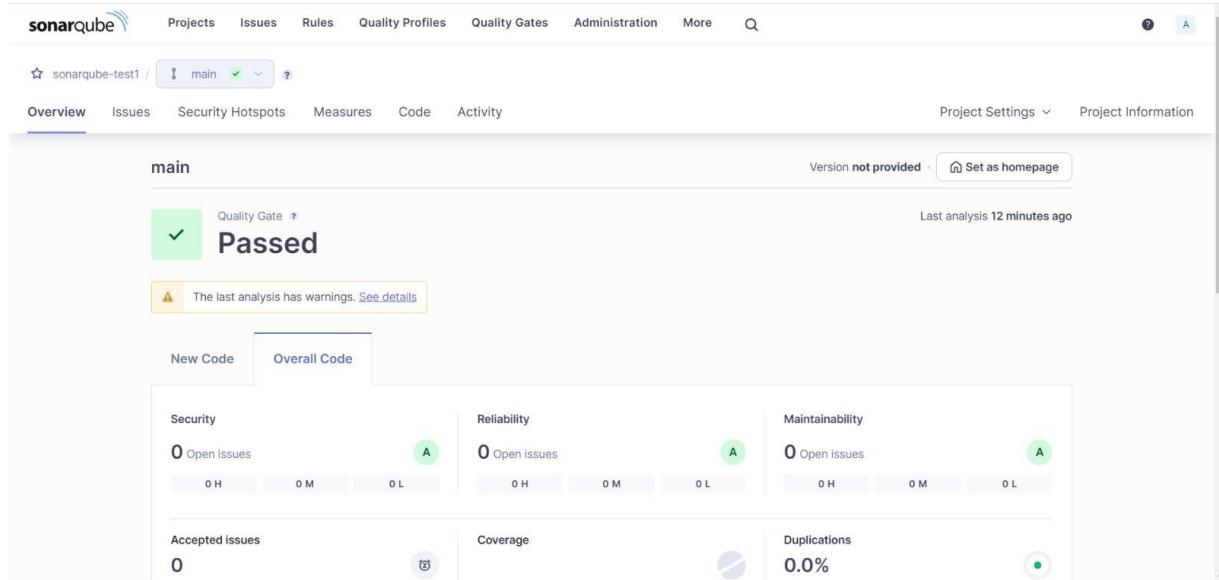
Console Output

```

Started by user jai navani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube Pipeline\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.45.2.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D master # timeout=10

```

```
11:22:18.236 INFO Sensor C# File Caching Sensor [csharp]
11:22:18.237 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
11:22:18.237 INFO Sensor C# File Caching Sensor [csharp] (done) | time=1ms
11:22:18.237 INFO Sensor Zero Coverage Sensor
11:22:18.251 INFO Sensor Zero Coverage Sensor (done) | time=14ms
11:22:18.256 INFO SCM Publisher SCM provider for this project is: git
11:22:18.257 INFO SCM Publisher 4 source files to be analyzed
11:22:18.789 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=531ms
11:22:18.793 INFO CPD Executor Calculating CPD for 0 files
11:22:18.795 INFO CPD Executor CPD calculation finished (done) | time=0ms
11:22:18.810 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
11:22:19.074 INFO Analysis report generated in 134ms, dir size=201.0 kB
11:22:19.137 INFO Analysis report compressed in 45ms, zip size=22.5 kB
11:22:19.351 INFO Analysis report uploaded in 212ms
11:22:19.353 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test1
11:22:19.354 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
11:22:19.354 INFO More about the report processing at http://localhost:9000/api/ce/task?id=971ae2f2-4e0f-49a7-88c4-ad4a6cceddf8
11:22:19.366 INFO Analysis total time: 24.819 s
11:22:19.368 INFO SonarScanner Engine completed successfully
11:22:19.454 INFO EXECUTION SUCCESS
11:22:19.455 INFO Total time: 29.696s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Screenshots for Experiment 9

Launch an ec2 instance

Give name use the default OS

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

nagios_host_exp_9kcs

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux 

macOS 

Ubuntu 

Windows 

Red Hat 

S >  **Browse more AMIs**
Including AMIs from AWS, Marketplace and the Community

Jai Talreja 61 D15A

now click on edit inbound rules

The screenshot shows the 'Inbound rules' tab selected in the CloudFormation console. It displays a single rule named 'sgr-0d6a171458e586b3e'. The rule is of type SSH, using TCP protocol on port 22, with a custom source of '0.0.0.0/0'. There are tabs for 'Outbound rules' and 'Tags' at the top.

now do the following configurations:
by clicking "add rules"

The screenshot shows the 'Inbound rules' tab selected in the CloudFormation console. It displays eight new rules added to the security group. The rules include various protocols like SSH, HTTP, HTTPS, ICMP, and Custom TCP, with different port ranges and sources. A 'Delete' button is visible for each rule. At the bottom left, there is a 'Add rule' button.

then click on save rules.

The screenshot shows the 'Security Groups' page in the CloudFormation console. It displays the details for a security group named 'sg-06ac4c5a9779ecaf9'. The 'Actions' dropdown is open. Below it, the 'Inbound rules' tab is selected, showing the eight rules added in the previous step. The 'Outbound rules' and 'Tags' tabs are also present.

Jai Talreja 61 D15A

```
mkdir ~/downloads  
cd ~/downloads
```

```
[ec2-user@ip-172-31-80-137 ~]$ mkdir ~/downloads  
cd ~/downloads  
[ec2-user@ip-172-31-80-137 downloads]$ |
```

wget <https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz>

```
cd ~/downloads  
[ec2-user@ip-172-31-80-137 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz  
--2024-09-28 06:27:51-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz  
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce  
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 2065473 (2.0M) [application/x-gzip]  
Saving to: 'nagios-4.5.5.tar.gz'  
  
nagios-4.5.5.tar.gz      100%[=====] 1.97M 5.30MB/s    in 0.4s  
2024-09-28 06:27:52 (5.30 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]  
[ec2-user@ip-172-31-80-137 downloads]$ |
```

wget <https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz>

```
[ec2-user@ip-172-31-80-137 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz  
--2024-09-28 06:28:14-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz  
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251  
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 2753049 (2.6M) [application/x-gzip]  
Saving to: 'nagios-plugins-2.4.11.tar.gz'  
  
nagios-plugins-2.4.11.tar.gz 100%[=====] 2.62M 5.90MB/s    in 0.4s  
2024-09-28 06:28:15 (5.90 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]  
[ec2-user@ip-172-31-80-137 downloads]$ |
```

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-80-137 downloads]$ tar zxvf nagios-4.5.5.tar.gz  
nagios-4.5.5/  
nagios-4.5.5/.github/  
nagios-4.5.5/.github/workflows/  
nagios-4.5.5/.github/workflows/test.yml  
nagios-4.5.5/.gitignore  
nagios-4.5.5/CONTRIBUTING.md  
nagios-4.5.5/Changelog  
nagios-4.5.5/INSTALLING  
nagios-4.5.5/LEGAL  
nagios-4.5.5/LICENSE  
nagios-4.5.5/Makefile.in
```

Jai Talreja 61 D15A

```
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defined
    alias             Nagios Admin        ; Full name of user
    email            2022.shubham.jha@ves.ac.in; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

#####
#
# CONTACT GROUPS

^G Help      ^O Write Out   ^W Where Is   ^K Cut      ^T Execute   ^C Location   M-U Undo   M-A Set M
```

press Ctrl+O and then enter.

then press Ctrl +X

```
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

sudo make install-webconf

```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***

[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

Adding password for nagios admin

sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

sudo service httpd restart

```
adding password for user nagiosadmin
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

The screenshot shows the Nagios Core 4.5.5 dashboard. At the top right, it displays "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". A green checkmark indicates "Daemon running with PID 3152". On the left, there's a sidebar with sections for General, Current Status, Reports, and System. The Current Status section is expanded, showing links like "Tactical Overview", "Map", "Hosts", "Services", "Host Groups", "Services (Unhandled)", "Hosts (Unhandled)", and "Network Outages". It also includes a "Quick Search" bar and a "Check for updates" link. The main content area has three columns: "Get Started" (with a list of bullet points), "Latest News" (empty), and "Don't Miss..." (empty). The footer contains copyright information: "Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors."

Screenshots for Experiment 10

```
● nagios.service - Nagios Core 4.4.14
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-09-30 08:54:01 UTC; 20s ago
     Docs: https://www.nagios.org/documentation
 Process: 55285 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 55286 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 55287 (nagios)
   Tasks: 6 (limit: 1141)
  Memory: 5.3M
    CPU: 252ms
   CGroup: /system.slice/nagios.service
           ├─55287 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─55288 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           ├─55289 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           ├─55290 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           ├─55291 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           └─55292 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 08:54:01 ip-172-31-44-151 nagios[55287]: qh: Socket '/usr/local/nagios/var/rw/nagios.gh' successfully initialized
lines 1-19
```

Monitoring a Linux machine, create an Ubuntu 20.04 server EC2 Instance in AWS.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links for 'EC2 Dashboard', 'EC2 Global View', 'Events', and 'Instances'. The main area has a header 'Instances (16) Info' with buttons for 'Connect', 'Instance state ▾', 'Actions ▾', and 'Launch instances ▾'. Below is a search bar 'Find instance by attribute or tag (case-sensitive)' and a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability zone. Three instances are listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
cutenagios_se...	i-09d6b0d2e181a7287	Running	t2.micro	-	No alarms	ap-south-
cutenagios_cli...	i-0e36968400dac0991	Running	t2.micro	-	No alarms	ap-south-
cutenagios_se...	i-03a3e79fc5ab0a056	Stopped	t2.micro	-	No alarms	ap-south-

`sudo apt update -y`

`sudo apt install gcc -y`

`sudo apt install -y nagios-nrpe-server nagios-plugins`

```
*** System restart required ***
Last login: Sat Sep 30 08:31:30 2023 from 13.233.177.3
ubuntu@ip-172-31-44-151:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:11.2.0-1ubuntu1).
gcc set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
ubuntu@ip-172-31-44-151:~$ 
root@ip-172-31-44-151:/home/ubuntu# sudo apt install nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'.
monitoring-plugins is already the newest version (2.3.1-1ubuntu4).
nagios-nrpe-server is already the newest version (4.0.3-1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 229 kB in 1s (290 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-44-151:/home/ubuntu# sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:11.2.0-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@ip-172-31-44-151:/home/ubuntu# sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'.
monitoring-plugins is already the newest version (2.3.1-1ubuntu4).
nagios-nrpe-server is already the newest version (4.0.3-1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

Jai Talreja 61 D15A

```
sudo nano /etc/nagios/nrpe.cfg
```

```
sudo systemctl restart nagios-nrpe-server
```

```
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-41-41:/home/ubuntu# sudo nano /etc/nagios/nrpe.cfg
root@ip-172-31-41-41:/home/ubuntu# sudo nano /etc/nagios/nrpe.cfg
root@ip-172-31-41-41:/home/ubuntu# sudo systemctl restart nagios-nrpe-server
root@ip-172-31-41-41:/home/ubuntu# sudo systemctl status nagios-nrpe-server
● nagios-nrpe-server.service - Nagios Remote Plugin Executor
```

```
root@ip-172-31-41-41:/home/ubuntu# sudo systemctl status nagios-nrpe-server
● nagios-nrpe-server.service - Nagios Remote Plugin Executor
    Loaded: loaded (/lib/systemd/system/nagios-nrpe-server.service; enabled; vendor preset: enabled)
    Active: active (running) since Sat 2023-09-30 09:27:17 UTC; 6s ago
      Docs: http://www.nagios.org/documentation
   Main PID: 7349 (nrpe)
     Tasks: 1 (limit: 1141)
    Memory: 1.5M
       CPU: 9ms
      CGroup: /system.slice/nagios-nrpe-server.service
              └─7349 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -f

Sep 30 09:27:17 ip-172-31-41-41 systemd[1]: nagios-nrpe-server.service: Deactivated successfully.
Sep 30 09:27:17 ip-172-31-41-41 systemd[1]: Stopped Nagios Remote Plugin Executor.
Sep 30 09:27:17 ip-172-31-41-41 systemd[1]: Started Nagios Remote Plugin Executor.
Sep 30 09:27:17 ip-172-31-41-41 nrpe[7349]: Starting up daemon
Sep 30 09:27:17 ip-172-31-41-41 nrpe[7349]: Server listening on 0.0.0.0 port 5666.
Sep 30 09:27:17 ip-172-31-41-41 nrpe[7349]: Listening for connections on port 5666
Sep 30 09:27:17 ip-172-31-41-41 nrpe[7349]: Allowing connections from: 127.0.0.1,::1,13.235.0.144
root@ip-172-31-41-41:/home/ubuntu# █
```

Jai Talreja 61 D15A

ps -ef | grep nagios

```
root@ip-172-31-44-151:/home/ubuntu# ps -ef | grep nagios
nagios      55287      1  0 08:54 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      55288  55287  0 08:54 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      55289  55287  0 08:54 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      55290  55287  0 08:54 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      55291  55287  0 08:54 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      55292  55287  0 08:54 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      56327      1  0 08:58 ?    00:00:00 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -f
root      60903  60158  0 09:32 pts/1  00:00:00 grep --color=auto nagios
root@ip-172-31-44-151:/home/ubuntu# sudo su
root@ip-172-31-44-151:/home/ubuntu# mkdir /usr/local/nagios/etc/objects/monitorhosts
root@ip-172-31-44-151:/home/ubuntu# mkdir /usr/local/nagios/etc/objects/linuxhosts
```

1.sudo su 2.mkdir /usr/local/nagios/etc/objects/monitorhosts 3.mkdir

/usr/local/nagios/etc/objects/monitorhosts/linuxhosts Copy the sample localhost.cfg file to linuxhost folder 4.cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

```
root@ip-172-31-44-151:/home/ubuntu# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
root@ip-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

```
GNU nano 6.2                               /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
#####
#
# Define a host for the local machine
#
define host {
    use          linux-server           ; Name of host template to use
                                ; This host definition will inherit all variables that are defined
                                ; in (or inherited by) the linux-server host template definition.
    host_name    localhost
    alias        localhost
    address     127.0.0.1
}

^G Help      ^O Write Out   ^W Where Is   ^K Cut          ^T Execute   ^C Location   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^\ Replace    ^U Paste       ^J Justify   ^/ Go To Line M-E Redo   M-G Copy
i-03a3e79fc5ab0a056 (cutenagios_server)  X
```

```
GNU nano 6.2                               /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg *
#####
#
# HOST GROUP DEFINITION
#
#####
#
# Define an optional hostgroup for Linux machines
#
define hostgroup {
    hostgroup_name    linux-servers[]           ; The name of the hostgroup
    alias             Linux Servers            ; Long name of the group
    members           localhost               ; Comma separated list of hosts that belong to this group
}

^G Help      ^O Write Out   ^W Where Is   ^K Cut          ^T Execute   ^C Location   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^\ Replace    ^U Paste       ^J Justify   ^/ Go To Line M-E Redo   M-G Copy
```

Jai Talreja 61 D15A

```
GNU nano 6.2                               /usr/local/nagios/etc/nagios.cfg *

# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/


# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts. The CGIs read object definitions from
Save modified buffer?
Y Yes
N No      ☰ C Cancel
```

```
root@ip-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
root@ip-172-31-44-151:/home/ubuntu#   /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.14
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2023-08-01
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
```

```
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
root@ip-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/nagios.cfg]
```

Sudo systemctl status nagios

```
● nagios.service - Nagios Core 4.4.14
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-09-30 08:54:01 UTC; 20s ago
     Docs: https://www.nagios.org/documentation
 Process: 55285 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 55286 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 55287 (nagios)
   Tasks: 6 (limit: 1141)
  Memory: 5.3M
    CPU: 252ms
   CGroup: /system.slice/nagios.service
           ├─55287 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─55288 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           ├─55289 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           ├─55290 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           ├─55291 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           ├─55292 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 08:54:01 ip-172-31-44-151 nagios[55287]: qh: Socket '/usr/local/nagios/var/rw/nagios.gh' successfully initialized
lines 1-19]
```

Not secure | 3.111.245.110/nagios/

Nagios®

General

- Current Status
- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
- Service Groups
- Problems
- Comments
- Events
- History
- Summary
- Histogram (Legacy)
- Notifications
- Event Log

System

- Comments
- Downtime
- Process Info
- Access Info
- Scheduling Queue
- Configuration

Reports

- Availability
- Trends (Legacy)
- Alarms
- History
- Summary
- Histogram (Legacy)
- Notifications
- Event Log

Current Network Status

Last Updated: Sat Sep 30 18:22:09 UTC 2023
Updated every 90 seconds
Nagios® Core: 4.4.14 - www.nagios.org
Logged in as nagiosadmin

View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Limit Results: 100 ▾

Host **	Status **	Last Check **	Duration **	Status Information
localhost	UP	09-30-2023 18:20:14	0d 9h 28m 7s	PING OK - Packet loss = 0%, RTA = 0.04 ms
linuxserver	UP	09-30-2023 18:17:06	0d 0h 5m 3s	PING OK - Packet loss = 0%, RTA = 0.62 ms

Results 1 - 2 of 2 Matching Hosts

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems All Types

Service Status Totals

OK	Warning	Unknown	Critical	Pending
15	0	0	3	0

All Problems All Types

Host Status Details For All Host Groups

Not secure | 13.233.247.135/nagios/

Nagios®

General

- Current Status
- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
- Summary
- Grid
- Service Groups
- Comments
- Grid
- Problems
- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages
- Quick Search:

Reports

- Availability
- Trends (Legacy)
- Alarms
- History
- Summary
- Histogram (Legacy)
- Notifications
- Event Log

System

- Comments
- Downtime
- Process Info
- Access Info
- Scheduling Queue
- Configuration

Current Network Status

Last Updated: Tue Oct 3 23:38:11 UTC 2023
Updated every 90 seconds
Nagios® Core: 4.4.14 - www.nagios.org
Logged in as nagiosadmin

View History For All hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Limit Results: 100 ▾

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
linuxserver	Current Load	OK	10-03-2023 23:34:51	3d 13h 47m 10s	1/4	OK - load average: 0.00, 0.02, 0.00
linuxserver	Current Users	OK	10-03-2023 23:35:29	3d 13h 46m 32s	1/4	USERS OK - 2 users currently logged in
linuxserver	HTTP	CRITICAL	10-03-2023 23:36:06	0d 0h 12m 5s	4/4	CRITICAL - Socket timeout
linuxserver	PING	OK	10-03-2023 23:36:44	0d 0h 1m 27s	1/4	PING OK - Packet loss = 0%, RTA = 0.60 ms
linuxserver	Root Partition	OK	10-03-2023 23:37:21	3d 13h 44m 40s	1/4	DISK OK - free space: / 4859 MB (62.78% inodes=88%):
linuxserver	SSH	OK	10-03-2023 23:37:59	0d 0h 0m 12s	1/4	SSH OK - OpenSSH_8.9p1 Ubuntu-3ubuntu0.1 (protocol 2.0)
linuxserver	Swap Usage	CRITICAL	10-03-2023 23:33:36	3d 13h 43m 25s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
linuxserver	Total Processes	OK	10-03-2023 23:34:14	3d 13h 42m 47s	1/4	PROCS OK: 39 processes with STATE = R/Z/DT
localhost	Current Load	OK	10-03-2023 23:35:10	3d 14h 43m 38s	1/4	OK - load average: 0.00, 0.02, 0.00
localhost	Current Users	OK	10-03-2023 23:35:47	3d 14h 42m 55s	1/4	USERS OK - 2 users currently logged in
localhost	HTTP	OK	10-03-2023 23:36:25	3d 14h 42m 18s	1/4	HTTP OK: HTTP/1.1 200 OK - 10945 bytes in 0.000 second response time
localhost	PING	OK	10-03-2023 23:37:02	3d 14h 41m 40s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
localhost	Root Partition	OK	10-03-2023 23:37:40	3d 14h 41m 38s	1/4	DISK OK - free space: / 4859 MB (62.78% inodes=88%):
localhost	SSH	OK	10-03-2023 23:33:17	3d 14h 40m 25s	1/4	SSH OK - OpenSSH_8.9p1 Ubuntu-3ubuntu0.4 (protocol 2.0)
localhost	Swap Usage	CRITICAL	10-03-2023 23:33:55	3d 14h 36m 48s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
localhost	Total Processes	OK	10-03-2023 23:33:24	3d 14h 39m 10s	1/4	PROCS OK: 40 processes with STATE = R/Z/DT

Results 1 - 16 of 16 Matching Services

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems All Types

Service Status Totals

OK	Warning	Unknown	Critical	Pending
15	0	0	3	0

All Problems All Types

Screenshots for Experiment 11

The screenshot shows the AWS Lambda home page. At the top, there's a navigation bar with tabs for AWS, Services, and Search. Below the navigation is a sidebar labeled "Compute". The main content area features a large heading "AWS Lambda" with the subtext "lets you run code without thinking about servers.". A paragraph explains that Lambda charges only for compute time consumed. To the right, a "Get started" box contains a "Create a function" button. Below this, a "How it works" section shows a code editor with a snippet of Node.js code:

```
1 * exports.handler = async (event) => {
2     console.log(event);
3     return 'Hello from Lambda!';
4 };
```

The code editor has tabs for .NET, Java, Node.js, Python, Ruby, and Custom runtime. A "Run" button is visible next to the code editor.

The screenshot shows the "Create function" wizard. The top navigation bar includes CloudShell, Feedback, and links for AWS, Services, and Search. The main title is "Create function" with an "Info" link. It asks to choose an option to create the function. There are four options:

- Author from scratch: Start with a simple Hello World example.
- Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image: Select a container image to deploy for your function.
- Browse serverless app repository: Deploy a sample Lambda application from the AWS Serverless Application Repository.

The "Basic information" step is currently active. It requires entering a function name ("sanketlambda") and selecting a runtime ("Python 3.12"). It also allows choosing an architecture ("x86_64").

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Create a new role with basic Lambda permissions

Use an existing role

Create a new role from AWS policy templates

Info Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named sanketlambda-role-aqbvjl1, with permission to upload logs to Amazon CloudWatch Logs.

sanketlambda | Functions | Lam

eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions/sanketlambda?newFunction=true&tab=code

Services Search [Alt+S]

Successfully created the function sanketlambda. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > sanketlambda

sanketlambda

Function overview Info

Description

Last modified 8 seconds ago

Function ARN arn:aws:lambda:eu-north-1:869935102438:function:sanketlambda

Function URL Info

Code Test Monitor Configuration Aliases Versions

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Hot weather Now ENG IN 15:46 02-10-2024

Successfully created the function **sanketlambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Code | Test | Monitor | Configuration | Aliases | Versions

Code source [Info](#)

[Upload from](#)

File Edit Find View Go Tools Window **Test** Deploy

Environment [Go to Anything \(Ctrl-P\)](#)

lambda_function Environment Var

```

1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9

```

Successfully created the function **sanketlambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Code | **Test** | Monitor | **Configuration** | Aliases | Versions

General configuration [Edit](#)

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStart Info	
0 min 3 sec	None	

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart [Info](#)

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).

None

Supported runtimes: Java 11, Java 17, Java 21.

Timeout

0 min 1 sec

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Use an existing role

Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

The screenshot shows the AWS Lambda console. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar containing 'Search', and a keyboard shortcut '[Alt+S]'. Below the navigation bar is a green success message banner: 'Successfully updated the function sanketlambda.' with a close button 'X'. The main content area has tabs: 'Code', 'Test', 'Monitor', 'Configuration' (which is selected), 'Aliases', and 'Versions'. On the left, a sidebar menu lists 'General configuration', 'Triggers', 'Permissions', 'Destinations', and 'Function URL'. The 'General configuration' section contains fields for 'Description' (empty), 'Memory' (128 MB), 'Ephemeral storage' (512 MB), 'Timeout' (0 min 1 sec), and 'SnapStart' (None). There's also an 'Edit' button.

The screenshot shows the 'Test event' configuration page. At the top, there are 'Save' and 'Test' buttons. A note says: 'To invoke your function without saving an event, configure the JSON event, then choose Test.' Under 'Test event action', 'Create new event' is selected. In the 'Event name' field, 'sanketevent' is typed. A note below says: 'Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.' Under 'Event sharing settings', 'Private' is selected. A note says: 'This event is only available in the Lambda console and to the event creator. You can configure a total of 10.' Under 'Template - optional', 'hello-world' is selected. The entire configuration is shown within a scrollable container.

Template - optional

hello-world

Event JSON

```
1 ▾ []
2   "key1": "value1",
3   "key2": "value2",
4   "key3": "value3"
5 []
```

The test event **sanktevent** was successfully saved.

Code | **Test** | **Monitor** | **Configuration** | **Aliases** | **Versions**

Code source [Info](#)

[Upload from](#)

File Edit Find View Go Tools Window **Test** Deploy

Execution results

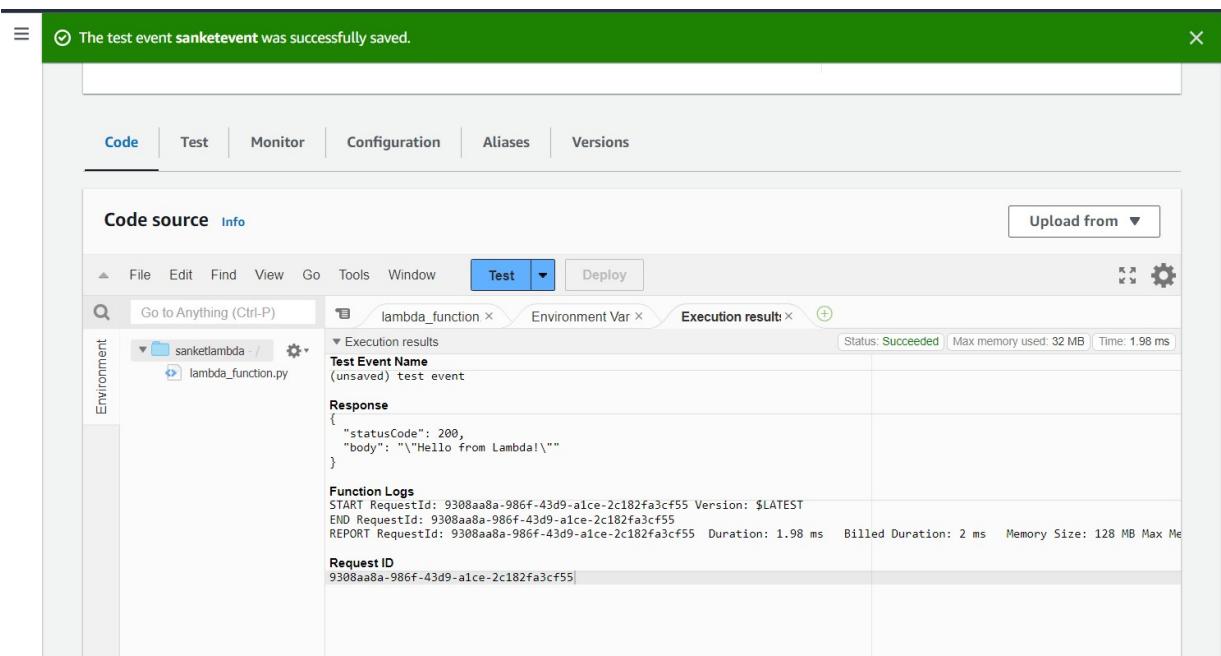
Test Event Name (unsaved) test event

Status: Succeeded | Max memory used: 32 MB | Time: 1.98 ms

Response

```
{ "statusCode": 200, "body": "\"Hello from Lambda!\""}  
Function Logs  
START RequestId: 9308aa8a-986f-43d9-a1ce-2c182fa3cf55 Version: $LATEST  
END RequestId: 9308aa8a-986f-43d9-a1ce-2c182fa3cf55  
REPORT RequestId: 9308aa8a-986f-43d9-a1ce-2c182fa3cf55 Duration: 1.98 ms Billed Duration: 2 ms Memory Size: 128 MB Max Mem
```

Request ID 9308aa8a-986f-43d9-a1ce-2c182fa3cf55

The screenshot shows the AWS Lambda function configuration interface. At the top, a green banner indicates that a test event named 'sanktevent' was successfully saved. Below this, there are tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code' tab is selected. On the left, there's a sidebar with 'Environment' and a search bar. The main area shows the code structure: 'lambda_function' (selected), 'Environment Var', and 'Execution result'. The 'Execution result' section displays the test event name '(unsaved) test event', the response object with a status code of 200 and a body of 'Hello from Lambda!', and detailed logs for the START, END, and REPORT events. The Request ID for the test execution is 9308aa8a-986f-43d9-a1ce-2c182fa3cf55.

Screenshots for Experiment 12

Create an S3 bucket of the same location as that of the Lambda function

The screenshot shows the 'Create bucket' page in the AWS Management Console. The 'General configuration' section is selected. Under 'AWS Region', 'Europe (Stockholm) eu-north-1' is chosen. Under 'Bucket type', 'General purpose' is selected. The 'Bucket name' field contains 'sanketbucket123'. A note below the name states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming.' Below the name is a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button and a placeholder 'Format: s3://bucket/prefix'.

The screenshot shows the 'sanketbucket123' bucket details page in the AWS Management Console. The 'Objects' tab is selected. The 'Objects (0)' section shows a table with columns: Name, Type, Last modified, Size, and Storage class. A message at the bottom of the table says 'No objects' and 'You don't have any objects in this bucket.'. There is a prominent orange 'Upload' button at the bottom of the table.

The screenshot shows the AWS Lambda 'Add trigger' configuration interface. At the top, there's a navigation bar with the AWS logo, 'Services' (selected), a search bar, and a keyboard shortcut '[Alt+S]'. Below the navigation, the path 'Lambda > Add triggers' is shown, followed by the title 'Add trigger'. A sub-header 'Trigger configuration' with an 'Info' link is present. The main configuration area starts with a section for 'Bucket', which lists 's3' as the selected provider with 'aws asynchronous storage' tags. A search bar contains 's3/sanketbucket123' with a clear button and a copy icon. The 'Bucket region' is set to 'eu-north-1'. The 'Event types' section allows selecting specific events like 'All object create events'. An optional 'Prefix' field is provided for limiting notifications to objects starting with a specific prefix, with an example 'e.g. images/'. The 'Recursive invocation' section cautions against using the same S3 bucket for both input and output, with a checkbox checked and a note about potential costs. A note at the bottom states that Lambda will add necessary permissions for S3 to invoke the function, with a link to learn more about the Lambda permissions model. At the bottom right, there are 'Cancel' and 'Add' buttons.

Trigger configuration [Info](#)

S3
aws asynchronous storage

Bucket

Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

s3/sanketbucket123

Bucket region: eu-north-1

Event types

Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events

Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any [special characters](#) must be URL encoded.

e.g. images/

Recursive invocation

If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

The screenshot shows the AWS CloudWatch interface. On the left, the navigation pane is open with the 'Logs' section expanded, showing 'Log groups' selected. The main content area displays the 'Log group details' for the log group '/aws/lambda/sanketlambda123'. The details include:

- Log class:** Info (Standard)
- ARN:** arn:aws:logs:eu-north-1:869935102438:log-group:/aws/lambda/sanketlambda123*
- Metric filters:** 0
- Creation time:** 3 minutes ago
- Retention:** Never expire
- Stored bytes:** -
- KMS key ID:** -
- Anomaly detection:** Configure
- Subscription filters:** 0
- Data protection:** -
- Contributor Insights rules:** -
- Sensitive data count:** -

Below the details, there are tabs for 'Log streams', 'Tags', 'Anomaly detection', 'Metric filters', 'Subscription filters', 'Contributor Insights', and 'Data protection'.

The screenshot shows the AWS CloudWatch interface. The navigation pane is identical to the previous screenshot. The main content area displays the 'Log events' for the log group '/aws/lambda/sanketlambda123' on the date 2024/10/02. The events listed are:

- 2024-10-02T10:59:36.409Z: INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:eu-north-1::runtime:188d9ca2e2714ff5637bd2bb...
- 2024-10-02T10:59:36.801Z: Loading function
- 2024-10-02T10:59:37.172Z: START RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 Version: \$LATEST
- 2024-10-02T10:59:37.718Z: CONTENT TYPE: image/jpeg
- 2024-10-02T10:59:37.725Z: END RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8
- 2024-10-02T10:59:37.725Z: REPORT RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 Duration: 552.91 ms Billed Duration: 553 ms Memory Size: 128 MB Max RL...

The interface includes a search bar, time range controls (Clear, 1m, 30m, 1h, 12h, Custom, UTC timezone), and a 'Display' dropdown.

Advan Ops Assignment (Q3)

(04/05)



Ans. 1 Steps to host a video streaming service using S3 on AWS.

(1) Create an S3 bucket

- Log in to your AWS management console
- Navigate to the S3 service.
- Click on "Create Bucket" and follow the prompts to set up your bucket.
- Name your bucket
- Choose a region to host the bucket from.
- Disable ACLs (Access Control Lists).
- Block public access if you only want users to connect via Cloudfront.
- Disable bucket versioning.
- Use default encryption.
- Click "Create Bucket".

(2) Upload your videos

- Click on the created bucket to open it.
- Click on "Upload" and choose the video files you want to upload.
- Click on "Upload" again.

(3) Configure bucket policy

- In the S3 bucket, under "Permissions" tab, update bucket policy to allow CloudFront access to your videos.

(4) Setups a CDN (Content Delivery Network) via CloudFront.

- Choose your S3 bucket as the origin.
- Click on "Redirect HTTP to HTTPS".
- Keep cache policy as 'Caching Optimized'.
- Check "Enable security protection" for Web App Firewall.

- Use all edge locations for best performance.
- Leave the remaining settings at default values.
- Copy the policy and update your bucket.

(5) Access videos via CloudFront.

- Use your CloudFront distribution name and video key to access it.

(6) Secure your Content

- Use IAM roles and policies.
- Enable encryption.

(7) Monitor and Optimize

- Use CloudFront's analytics and monitoring tools.
- Optimize Caching settings in CloudFront settings.

Ans-2) BMW Group's Cloud Transformation with AWS A Case Study

Overview :- The BMW Group, renowned for its premium automobiles and motorcycles, embarked on a digital transformation journey to enhance its operational efficiency and innovation capabilities.

By migrating its on-premises data lake to Amazon Web Services (AWS), BMW has been able to leverage cloud technology to meet global demands and drive innovation.

Challenges :- BMW faced several challenges with its existing infrastructure, like,

~~Scalability issues to accommodate the increasing volume of data generated by BMW's global operations.~~

~~Fragmented data causing data silos, causing inability to fulfill needs of, Real-time access.~~

Solutions:- (1) Cloud Data Hub (CDH)

- BMW developed a centralized data lake using Amazon S3 for scalable and secure storage and Amazon Redshift, which enables fast querying and analysis of large datasets.

(2) Machine Learning and Predictive Analysis

- Amazon SageMaker to train and deploy machine learning models.
- ~~Amazon Kinesis~~ to facilitate real-time data streaming and analytics.

(3) Blockchain Technology

- Amazon Elastic Kubernetes Service (EKS) to enhance traceability of automotive parts.

(4) Multilingual Business Process using Amazon Translate.

Outcomes :- The migration to AWS has yielded several significant benefits for BMW.

- Enhanced Scalability
- Improved Innovation
- Real-Time Insights
- Operational Efficiency

Services used :-

- S3
- Redshift
- Logstash
- Kinesis
- FTS
- Translate

Conclusion :-

~~BMW's collaboration with AWS highlights the transformative power of cloud technology in driving scalability, innovation, and real-time data processing.~~

This partnership has enabled BMW to stay at the forefront of the automobile industry, delivering premium experiences to its customers, worldwide.

Hotstar's success with AWS

A Case Study

Overview:- Hotstar, now known as Disney+Hotstar is one of India's leading OTT streaming platforms.

It has leveraged Amazon Web Services (AWS) to handle massive traffic spikes, especially during live events like the Indian Premier League (IPL) cricket matches.

By using AWS, Hotstar has been able to scale its infrastructure to support millions of concurrent viewers.

Challenges:- Hotstar faced the following challenges with its existing infrastructure:

Scalability to handle sudden traffic spikes,

Performance, to ensure low-latency and high-quality streaming.

Cost management by scaling infrastructure

Solutions :- To address its challenges, Hotstar implemented the following services,

- Amazon EC2 and Elastic Load Balancing (ELB) for running applications.
- Amazon CloudFront for content delivery.
- Amazon Kinesis for Real-time analytics.
- EC2 spot instances to optimize costs.

Outcomes :- The migration to AWS has provided Hotstar with several benefits.

- Scalability
- Performance
- Cost Efficiency
- Real-time insights

Conclusion :- Hotstar's collaboration with AWS showcases how leveraging cloud technology enabled the platform to deliver high-quality streaming experiences to millions of users.

Ans. D) Kubernetes is used because it automates the deployment, scaling, and management of containerized applications.

It ensures high availability, efficient resource utilization, and easy scaling, making it ideal for managing complex, distributed systems.

Advantages :-

- Scalability
- High availability
- Portability
- Resource Efficiency
- Extensibility
- Automation

Disadvantages :-

- Complexity
- Resource Intensive
- Operational Overhead
- Security
- Cost

How Adidas leverages Kubernetes :-

- Containerization and Agile Development,
- By using Kubernetes, Adidas can deploy applications faster and more quickly.

- Improved Developer Experience

With Kubernetes, Adidas no longer faces delays in getting development environments set up.

- Enhanced E-Commerce Performance

Kubernetes has reduced site load times to half and allowed more frequent releases.

- Scalability and Reliability

Adidas manages thousands of pods and hundreds of nodes, ensuring their systems can handle high traffic volumes, especially during peak shopping events.

- Continuous Delivery

Rapid push updates help Adidas stay competitive in the fast-paced retail market.

Ans.  Nagios is an open-source monitoring and alerting system designed to track the health and performance of IT infrastructure, networks, and applications.

Key Features :

- Server monitoring
- Network monitoring
- Service availability
- Resource Utilization
- Performance metrics and Trend analysis
- Log file monitoring
- Alerting and Notification
- Application monitoring
- Security monitoring
- Environmental monitoring

Nagios is extensively used in E-Services to ensure the reliability and performance of various IT components.

It provides real-time monitoring, easy-to-understand graphics, and intuitive statistics, which helped in reducing troubleshooting time and ensuring high availability of services.

By using Nagios, IT-services can,

- Centralize monitoring
- Improve response times
- Enhance reliability
- Simplify management

AdDev Ops Assignment 2

1

- 1) Create a REST API with the Serverless Framework.

2) "Serverless Framework" refers to the user not needing to manage and maintain their own servers.

Steps :-

- 1) Install serverless framework (CLI globally) using 'node package manager' (npm).
- 2) Create node JS serverless project, inside a directory intended for it.
This service will house all your Lambda functions, configurations and cloud resources.
- 3) The project scaffold creates essential files like handler.js which contains code for Lambda functions and serverless.yml.
- 4) Create a REST API resource that handles HTTP requests.

- 5> Deploy the Service and upload new resources to AWS and set up the infrastructure.
- 6> Once deployed, test the REST API tools like curl or Postman by making post requests to generated API.
- 7> To store submitted data you integrate AWS DynamoDB as a database.
- 8> Add functionalities like list all candidates by ID.
- 9> You need to ensure that serverless framework is given right permissions to interact with AWS resources like Dynamo DS.
- 10> After deployment serverless framework provides service information like deployed endpoints, API key, log streams.

Case Study for SonarQube

- Creating ~~new~~ profile in SonarQube for testing project quality.
- SonarQube is an Open Source platform used for continuous inspection of quality.
- SonarQube detects bugs and code smells along with vulnerabilities in projects across various programming languages.

Profile Creation:

Quality profiles in SonarQube are essential configurations that define rules applied during code analysis. Each project has a quality profile for every supported language with default being 'Sonar Way'.

Custom profiles can be created by copying or extending existing ones.

Permissions to manage quality profiles are restricted to users with administrator privileges. Permissions can also be imported from other instances via backup and restore.

2.) SonarCloud to analyze Github code:

SonarCloud is a cloud-based counter of SonarQube that integrates directly with GitHub, BitBucket, Azure and GitHub repositories.

To get started with SonarCloud via GitHub setup with each project corresponding to GitHub repositories. Automatic analysis happens directly in SonarCloud while CI based analysis integrates with your build process so the analysis is complete results can be viewed in both SonarCloud and GitHub including including security import issue.

3.) Sonarlint in Java IDE:

Sonarlint is an IDE that performs on-the-fly code analysis as you write code. It helps developers in development environments, such as IntelliJ IDEA or Eclipse. This approach ensures immediate feedback on code quality, promoting clean & maintainable code from the beginning.

➤ Analyzing Python projects with SonarQube:

SonarQube supports Python test coverage reporting but it requires third party tools like coverage. For setup, you can use .Travis and Coverage. PyTo configure and run tests in your travis.yml.

The build process can also be automated using GitHub Actions, which installs dependencies, runs tests and invokes SonarQube scan. Ensure report is in XML format.

➤ Analyzing Node projects with SonarQube:

You can configure SonarQube to analyze Node JS projects by installing the appropriate plugin and using SonarScanner to scan the projects.

SonarQube will check the code against industry standard rules and best practices, flagging issues related to security vulnerabilities, bugs and performance optimization.

Q.3 At a large organization, your central operations team may get many repetitive infrastructure requests. You can create and use Terraform modules that codify the standards for deploying & managing services in your org. Terraform cloud can also integrate with ticketing system like Jira.

Ans) Implementing a 'self-service' infrastructure model using Terraform can transform how large organizations manage their infrastructure independently.

Organizations can enhance efficiency, reduce bottlenecks, and ensure compliance with established needs.

→ Overwhelming number of repetitive requests leads to delays in service deliveries.

→ Benefits:

- Modularity and Reusability
- Standardization
- Increased Efficiency
- Integration with ticketing systems

* Implementation Steps :

- 1) Identify Infrastructure components
 - Begin by identifying which components of your infrastructure can be modularized
- 2) Establish Governance and Best Practices to ensure clarity and maintainability.
- 3) Develop Terraform modules to include input variables for customizations and outputs for integration with other modules.
- 4) Testing and Validation of best modules for module management, utilize the Terraform registry.
- 5) Version Control for modules to track changes over-time. This helps manage dependencies effectively and minimize disruptions during updates.
- 6) Documentation for each module, usage examples, input/output descriptions and any dependencies.

• Encourage collaboration and promote consistency in deployments and facilitate knowledge within the organization. Their approach not only streamlines processes but also enhances agility in responding to changing business needs.

Ultimately it leads to a more responsive IT environment that supports innovation and growth within the organization.

