



AWS Whitepaper

# AWS Outposts Pertimbangan Desain dan Arsitektur Ketersediaan Tinggi



# AWS Outposts Pertimbangan Desain dan Arsitektur Ketersediaan Tinggi: AWS Whitepaper

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

# Table of Contents

Abstrak dan pengantar .....	i
Apakah Anda sudah Well-Architected? .....	1
Pengantar .....	1
Memperluas AWS infrastruktur dan layanan ke lokasi lokal .....	2
Memahami Model Tanggung Jawab AWS Outposts Bersama .....	5
Berpikir dalam hal mode kegagalan .....	7
Mode kegagalan 1: Jaringan .....	7
Mode kegagalan 2: Contoh .....	8
Mode kegagalan 3: Hitung .....	8
Mode kegagalan 4: Rak atau pusat data .....	8
Mode kegagalan 5: Zona AWS Ketersediaan atau Wilayah .....	9
Membangun aplikasi HA dan solusi infrastruktur dengan AWS Outposts rak .....	10
Jaringan .....	11
Lampiran jaringan .....	12
Konektivitas jangkar .....	17
Perutean aplikasi/beban kerja .....	21
Hitung .....	25
Perencanaan kapasitas .....	25
Manajemen kapasitas .....	29
Penempatan instans .....	32
Penyimpanan .....	35
Perlindungan data .....	36
Basis Data .....	39
Amazon RDS di Outposts dengan Multi-AZ .....	39
Amazon RDS pada AWS Outposts Read Replicas .....	41
Penskalaan otomatis penyimpanan Amazon RDS aktif AWS Outposts .....	42
Amazon RDS pada cadangan AWS Outposts lokal .....	42
Mode kegagalan yang lebih besar .....	43
Outposts Rack perutean intra-VPC .....	43
Outposts Rack perutean antar-VPC .....	44
Route 53 Resolver Lokal di Outposts .....	45
Kluster Lokal EKS di Outposts .....	47
Kesimpulan .....	49
Kontributor .....	50

---

Riwayat dokumen .....	51
Pemberitahuan .....	52
AWS Glosarium .....	53
.....	liv

# AWS Outposts Pertimbangan Desain dan Arsitektur Ketersediaan Tinggi

Tanggal publikasi: 12 Agustus 2021 ([Riwayat dokumen](#))

Whitepaper ini membahas pertimbangan arsitektur dan praktik yang direkomendasikan yang dapat diterapkan oleh manajer TI dan arsitek sistem untuk membangun lingkungan aplikasi lokal yang sangat tersedia. AWS Outposts

## Apakah Anda sudah Well-Architected?

[Kerangka Kerja AWS Well-Architected](#) membantu Anda memahami pro dan kontra dari keputusan yang Anda buat saat membangun sistem di cloud. Enam pilar dari Kerangka Kerja ini memungkinkan Anda mempelajari praktik terbaik arsitektural untuk merancang dan mengoperasikan sistem yang andal, aman, efisien, hemat biaya, dan berkelanjutan. Dengan menggunakan [AWS Well-Architected Tool](#), tersedia tanpa biaya di [Konsol Manajemen AWS](#), Anda dapat meninjau beban kerja Anda terhadap praktik terbaik ini dengan menjawab serangkaian pertanyaan untuk setiap pilar.

Untuk panduan lebih lanjut dari para ahli dan praktik terbaik untuk arsitektur cloud Anda—referensi penerapan arsitektur, diagram, dan laporan resmi—lihat [Pusat Arsitektur AWS](#).

## Pengantar

Paper ini ditujukan untuk manajer TI dan arsitek sistem yang ingin menyebarkan, memigrasi, dan mengoperasikan aplikasi menggunakan platform AWS cloud dan menjalankan aplikasi tersebut di tempat dengan [AWS Outposts rak, faktor bentuk rak 42U](#). [AWS Outposts](#)

Ini memperkenalkan pola arsitektur, anti-pola, dan praktik yang direkomendasikan untuk membangun sistem yang sangat tersedia yang mencakup AWS Outposts rak. Anda akan belajar bagaimana mengelola kapasitas AWS Outposts rak Anda dan menggunakan jaringan dan layanan fasilitas pusat data untuk menyiapkan solusi infrastruktur AWS Outposts rak yang sangat tersedia.

AWS Outposts rack adalah layanan yang dikelola sepenuhnya yang menyediakan kumpulan logis komputasi awan, penyimpanan, dan kemampuan jaringan. [Dengan rak Outposts, pelanggan dapat menggunakan layanan AWS terkelola yang didukung di lingkungan lokal mereka, termasuk: Amazon Elastic Compute Cloud \(Amazon\), Amazon ElasticBlock Store \(Amazon EBS EC2\), AmazonS3 di](#)

[Outposts, Amazon Elastic Kubernetes Service \(AmazonEKS\), Amazon Elastic Container Service \(Amazon ECS\), Layanan Database Relasional Amazon Amazon Relational Database Service \(Amazon RDS\), dan layanan lainnya di Outposts.AWS](#) Layanan di Outposts dikirimkan pada [Sistem AWS Nitro](#) yang sama yang digunakan dalam. Wilayah AWS

Dengan memanfaatkan AWS Outposts rack, Anda dapat membangun, mengelola, dan menskalakan aplikasi lokal yang sangat tersedia menggunakan layanan dan alat AWS cloud yang sudah dikenal. AWS Outposts rack sangat ideal untuk beban kerja yang memerlukan akses latensi rendah ke sistem lokal, pemrosesan data lokal, residensi data, dan migrasi aplikasi dengan saling ketergantungan sistem lokal.

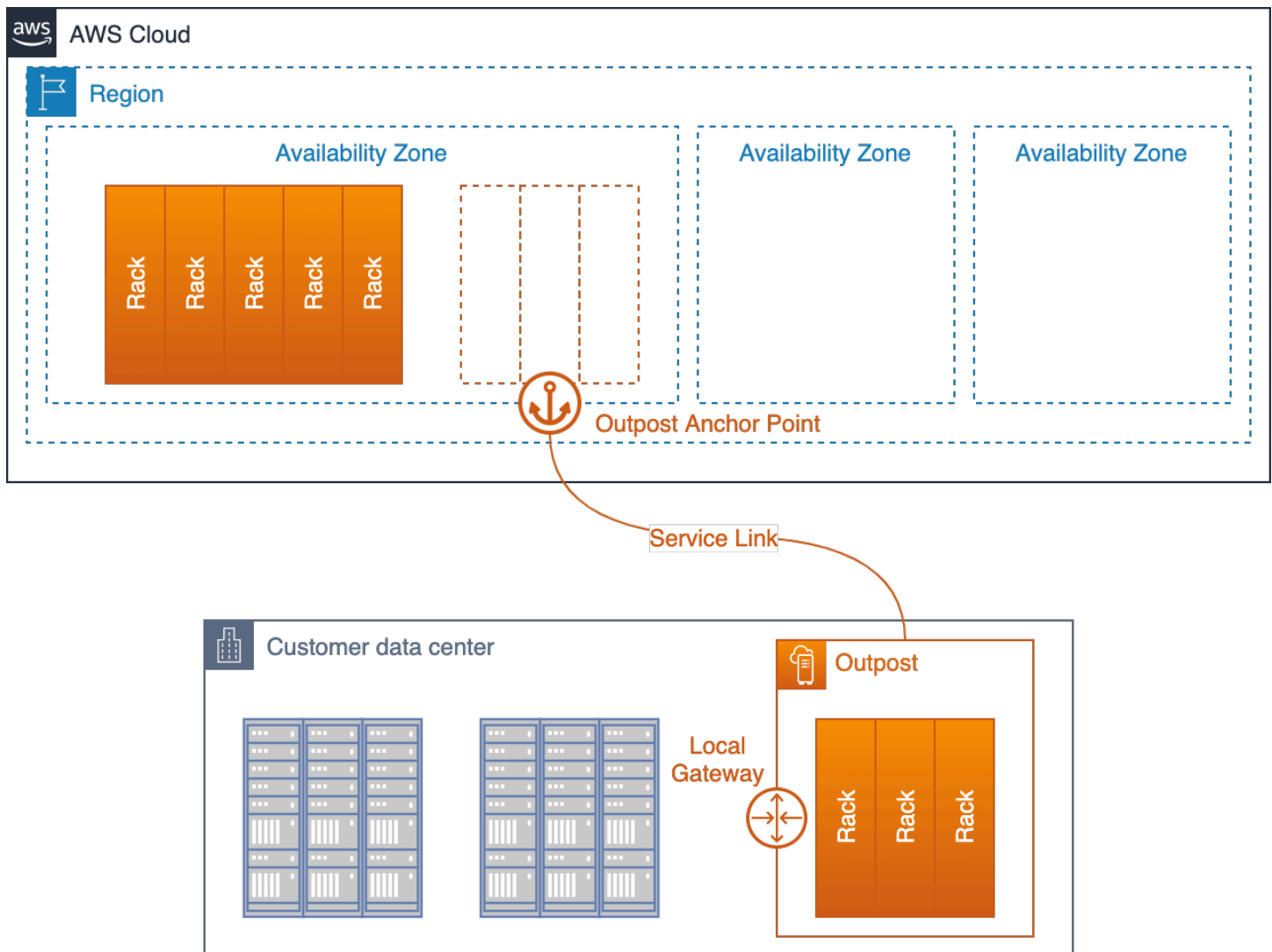
## Memperluas AWS infrastruktur dan layanan ke lokasi lokal

AWS Outposts Layanan ini memberikan AWS infrastruktur dan layanan ke lokasi lokal di [lebih dari 50 negara dan wilayah](#), memberikan pelanggan kemampuan untuk menyebarkan AWS infrastruktur, AWS layanan, dan alat yang sama ke hampir semua pusat data APIs, ruang lokasi bersama, atau fasilitas lokal untuk pengalaman hybrid yang benar-benar konsisten. Untuk memahami cara mendesain dengan Outposts, Anda harus memahami berbagai tingkatan yang membentuk cloud. AWS

An [Wilayah AWS](#) adalah wilayah geografis dunia. Masing-masing Wilayah AWS adalah kumpulan pusat data yang secara logis dikelompokkan ke dalam [Availability Zones](#) (AZs). Wilayah AWS menyediakan beberapa (setidaknya dua) Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan latensi rendah, throughput tinggi, dan konektivitas jaringan redundan. Setiap AZ terdiri dari satu atau lebih pusat data fisik.

[Pos Luar](#) logis (selanjutnya disebut sebagai Outpost) adalah penyebaran satu atau lebih AWS Outposts rak yang terhubung secara fisik yang dikelola sebagai satu kesatuan. Outpost menyediakan kumpulan kapasitas AWS komputasi dan penyimpanan di salah satu situs Anda sebagai perpanjangan pribadi dari AZ dalam file. Wilayah AWS

Mungkin model konseptual terbaik AWS Outposts adalah memikirkan mencabut satu atau lebih rak dari pusat data di AZ Wilayah AWS, dan memasangnya di pusat data atau fasilitas colocation Anda sendiri. Anda memutar rak dari pusat data AZ ke pusat data Anda. Anda kemudian mencolokkan rak ke [titik jangkar](#) di pusat data AZ dengan kabel (sangat) panjang sehingga rak terus berfungsi sebagai bagian dari. Wilayah AWS Anda juga menghubungkannya ke jaringan lokal untuk menyediakan konektivitas latensi rendah antara jaringan lokal dan beban kerja yang berjalan di rak tersebut. Ini memberi Anda konsistensi operasional dan API AWS Cloud, sambil menjaga beban kerja Anda tetap lokal.



Pos Terdepan ditempatkan di pusat data pelanggan dan terhubung kembali ke jangkar AZ dan Wilayah induknya

Outpost berfungsi sebagai perpanjangan dari AZ di mana ia berlabuh. AWS mengoperasikan, memantau, dan mengelola AWS Outposts infrastruktur sebagai bagian dari Wilayah AWS. Alih-alih kabel fisik yang sangat panjang, Outpost menghubungkan kembali ke Wilayah induknya melalui satu set terowongan VPN terenkripsi yang disebut Service Link.

Tautan Layanan berakhir pada satu set titik jangkar di Availability Zone (AZ) di Wilayah induk Outpost.

Anda memilih di mana konten Anda disimpan. Anda dapat mereplikasi dan mencadangkan konten Anda ke Wilayah AWS atau lokasi lain. Konten Anda tidak akan dipindahkan atau disalin di luar lokasi

yang Anda pilih tanpa persetujuan Anda, kecuali jika diperlukan untuk mematuhi hukum atau perintah yang mengikat dari badan pemerintah. Untuk informasi selengkapnya, lihat [FAQ Privasi AWS Data](#).

Beban kerja yang Anda terapkan di rak tersebut berjalan secara lokal. Dan, sementara kapasitas komputasi dan penyimpanan yang tersedia di rak tersebut terbatas dan tidak dapat mengakomodasi menjalankan layanan skala cloud Wilayah AWS, sumber daya yang digunakan di rak (instans Anda dan penyimpanan lokalnya) menerima manfaat berjalan secara lokal sementara pesawat manajemen terus beroperasi di Wilayah AWS

Untuk menerapkan beban kerja di Outpost, Anda menambahkan subnet ke lingkungan Virtual Private Cloud (VPC) dan menentukan Outpost sebagai lokasi untuk subnet. Kemudian, Anda memilih subnet yang diinginkan saat menerapkan AWS sumber daya yang didukung melalui alat, Konsol Manajemen AWS CLI, CDK APIs, atau infrastruktur sebagai kode (IaC). Contoh di subnet Outpost berkomunikasi dengan instans lain di Outpost atau di Wilayah melalui jaringan VPC.

Outpost Service Link membawa lalu lintas manajemen Outpost dan lalu lintas VPC pelanggan (lalu lintas VPC antara subnet di Outpost dan subnet di Wilayah).

### Istilah penting:

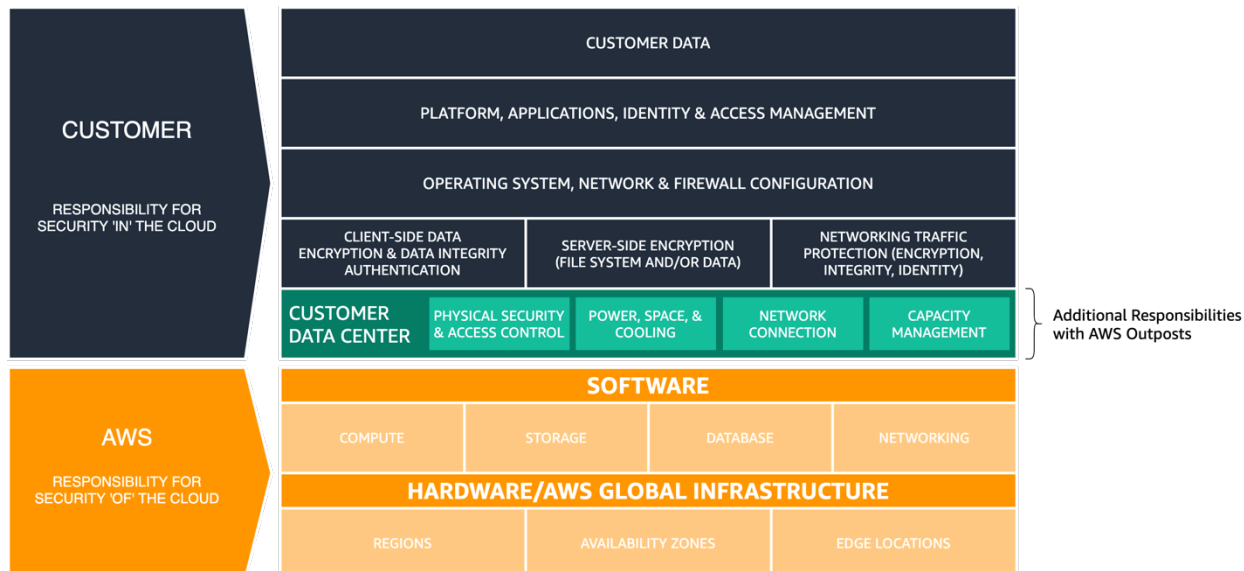
- AWS Outposts— adalah layanan yang dikelola sepenuhnya yang menawarkan AWS infrastruktur, AWS layanan APIs, dan alat yang sama untuk hampir semua pusat data, ruang co-lokasi, atau fasilitas lokal untuk pengalaman hybrid yang benar-benar konsisten.
- Outpost adalah penyebaran satu atau lebih AWS Outposts rak yang terhubung secara fisik yang dikelola sebagai entitas logis tunggal dan kumpulan AWS komputasi, penyimpanan, dan jaringan yang digunakan di situs pelanggan.
- Wilayah Induk — Wilayah AWS yang menyediakan manajemen, layanan pesawat kontrol, dan AWS layanan regional untuk penyebaran Outpost.
- Anchor Availability Zone (anchor AZ) — Availability Zone di Wilayah induk yang menjadi tuan rumah anchor point untuk Outpost. Outpost berfungsi sebagai perpanjangan dari jangkar AZ. Jangkar AZ dipilih oleh pelanggan saat pesanan Outposts ditempatkan. Setelah jangkar AZ dipilih, itu tidak dapat diubah selama jangka waktu AWS Outposts berlangganan.
- Anchor Points — titik akhir di anchor AZ yang menerima koneksi dari Outposts yang dikerahkan dari jarak jauh.
- Service Link — satu set terowongan VPN terenkripsi yang menghubungkan Outpost ke Availability Zone jangkar di Wilayah induknya.

- Local Gateway (LGW) — Router virtual interkoneksi logis yang memungkinkan komunikasi antara Outpost dan jaringan lokal Anda.

## Memahami Model Tanggung Jawab AWS Outposts Bersama

Saat Anda menyebarkan AWS Outposts infrastruktur ke pusat data atau fasilitas lokasi bersama, Anda mengambil tanggung jawab tambahan dalam model Tanggung [Jawab AWS Bersama](#). Misalnya, di Wilayah, AWS menyediakan sumber daya yang beragam, jaringan inti yang berlebihan, dan konektivitas Wide Area Network (WAN) yang tangguh untuk memastikan layanan tersedia jika terjadi satu atau lebih kegagalan komponen.

Dengan Outposts, Anda bertanggung jawab untuk menyediakan daya tangguh dan konektivitas jaringan ke rak Outpost untuk memenuhi persyaratan ketersediaan Anda untuk beban kerja yang berjalan di Outposts.



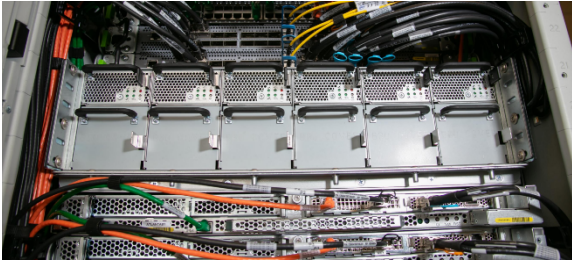
### AWS Model Tanggung Jawab Bersama diperbarui untuk AWS Outposts

Dengan AWS Outposts, Anda bertanggung jawab atas keamanan fisik dan kontrol akses lingkungan pusat data. Anda harus menyediakan daya, ruang, dan pendinginan yang cukup untuk menjaga agar Outpost tetap beroperasi dan koneksi jaringan untuk menghubungkan Outpost kembali ke Wilayah.

Karena kapasitas Outpost terbatas dan ditentukan oleh ukuran dan jumlah AWS pemasangan rak di situs Anda, Anda harus memutuskan berapa banyak EC2, EBS, dan S3 pada kapasitas Outposts yang Anda butuhkan untuk menjalankan beban kerja awal Anda, mengakomodasi pertumbuhan masa depan, dan untuk menyediakan kapasitas ekstra untuk mengurangi kegagalan server dan peristiwa pemeliharaan.

AWS bertanggung jawab atas ketersediaan infrastruktur Outposts termasuk catu daya, server, dan peralatan jaringan di dalam rak. AWS Outposts AWS juga mengelola hypervisor virtualisasi, sistem penyimpanan, dan AWS layanan yang berjalan di Outposts.

Rak daya pusat di setiap rak Outposts mengkonversi dari daya AC ke DC dan memasok daya ke server di rak melalui arsitektur bus bar. Dengan arsitektur bus bar, setengah catu daya di rak bisa gagal dan semua server akan terus berjalan tanpa gangguan.



Gambar 3 - AWS Outposts AC-to-DC catu daya dan distribusi daya bus bar

Sakelar jaringan dan pemasangan kabel di dalam dan di antara rak Outposts juga sepenuhnya berlebihan. Panel patch serat menyediakan konektivitas antara rak Outpost dan jaringan lokal dan berfungsi sebagai titik demarkasi antara lingkungan pusat data yang dikelola pelanggan dan lingkungan yang dikelola. AWS Outposts

Sama seperti di Wilayah, AWS bertanggung jawab atas layanan cloud yang ditawarkan di Outposts dan mengambil tanggung jawab tambahan saat Anda memilih dan menerapkan layanan terkelola tingkat tinggi seperti Amazon RDS di Outposts. Anda harus meninjau [Model Tanggung Jawab AWS Bersama](#) dan halaman Pertanyaan yang Sering Diajukan (FAQ) untuk layanan individual saat Anda mempertimbangkan dan memilih layanan untuk diterapkan di Outposts. Sumber daya ini memberikan rincian tambahan tentang pembagian tanggung jawab antara Anda dan AWS.

# Berpikir dalam hal mode kegagalan

Saat merancang aplikasi atau sistem yang sangat tersedia, Anda harus mempertimbangkan komponen apa yang mungkin gagal, dampak kegagalan komponen apa yang akan terjadi pada sistem serta tujuan [RPO/RTO](#) aplikasi Anda, dan mekanisme apa yang dapat Anda terapkan untuk mengurangi atau menghilangkan dampak kegagalan komponen. Apakah aplikasi Anda berjalan di satu server, dalam satu rak, atau dalam satu pusat data? Apa yang akan terjadi ketika server, rak, atau pusat data mengalami kegagalan sementara atau permanen? Apa yang terjadi ketika ada kegagalan dalam sub-sistem kritis seperti jaringan atau dalam aplikasi itu sendiri? Ini adalah mode kegagalan.

Anda harus mempertimbangkan mode kegagalan di bagian ini saat merencanakan Outposts dan penerapan aplikasi Anda. Bagian berikut akan meninjau cara mengurangi mode kegagalan ini untuk memberikan peningkatan tingkat ketersediaan tinggi untuk lingkungan aplikasi Anda.

## Mode kegagalan 1: Jaringan

Penyebaran Outpost bergantung pada koneksi tangguh ke Wilayah induknya untuk pengelolaan dan pemantauan. Gangguan jaringan dapat disebabkan oleh berbagai kegagalan seperti kesalahan operator, kegagalan peralatan, dan pemadaman penyedia layanan. Pos Luar, yang mungkin terdiri dari satu atau lebih rak yang terhubung bersama di situs, dianggap terputus ketika tidak dapat berkomunikasi dengan Wilayah melalui Tautan Layanan.

Jalur jaringan yang berlebihan dapat membantu mengurangi risiko peristiwa pemutusan hubungan. Anda harus memetakan dependensi aplikasi dan lalu lintas jaringan untuk memahami dampak peristiwa pemutusan hubungan terhadap operasi beban kerja. Rencanakan redundansi jaringan yang memadai untuk memenuhi persyaratan ketersediaan aplikasi Anda.

Selama peristiwa pemutusan sambungan, instance yang berjalan di Outpost terus berjalan dan dapat diakses dari jaringan lokal melalui Outpost Local Gateway (LGW). Beban kerja dan layanan lokal mungkin terganggu atau gagal jika bergantung pada layanan di Wilayah. Permintaan mutasi (seperti memulai atau menghentikan instance di Outpost), operasi bidang kontrol, dan telemetri layanan (misalnya, CloudWatch metrik) akan gagal saat Pos Luar terputus dari Wilayah. CloudWatch metrik akan di-spoiled secara lokal di Outpost Anda untuk jangka waktu singkat pemutusan jaringan, dan akan dikirim ke Wilayah untuk ditinjau ketika koneksi tautan layanan dibuat ulang.

## Mode kegagalan 2: Contoh

EC2 Instans Amazon dapat menjadi terganggu atau gagal jika server yang mereka jalankan memiliki masalah atau jika instans mengalami kegagalan sistem operasi atau aplikasi. Bagaimana aplikasi menangani jenis kegagalan ini tergantung pada arsitektur aplikasi. Aplikasi monolitik biasanya menggunakan fitur aplikasi atau sistem untuk pemulihan sementara arsitektur berorientasi layanan modular atau [layanan mikro](#) biasanya menggantikan komponen yang gagal untuk mempertahankan ketersediaan layanan.

Anda dapat mengganti instans yang gagal dengan instans baru menggunakan mekanisme otomatis seperti grup Amazon Auto EC2 Scaling. Pemulihan otomatis instans dapat memulai ulang instance yang gagal karena kegagalan server asalkan ada kapasitas cadangan yang cukup tersedia di server yang tersisa dan tautan layanan masih terhubung.

## Mode kegagalan 3: Hitung

Server dapat gagal atau menjadi terganggu dan mungkin perlu dikeluarkan dari operasi (sementara atau permanen) karena berbagai alasan, seperti kegagalan komponen dan operasi pemeliharaan terjadwal. Bagaimana layanan di rak Outposts menangani kegagalan dan kerusakan server bervariasi dan dapat bergantung pada bagaimana pelanggan mengonfigurasi opsi ketersediaan tinggi.

Anda harus memesan kapasitas komputasi yang cukup untuk mendukung model N+M ketersediaan, di mana N kapasitas yang M diperlukan dan kapasitas cadangan disediakan untuk mengakomodasi kegagalan server.

Penggantian perangkat keras untuk server yang gagal disediakan sebagai bagian dari layanan AWS Outposts rak yang dikelola sepenuhnya. AWS secara aktif memantau kesehatan semua server dan perangkat jaringan dalam penyebaran Outpost. Jika ada kebutuhan untuk melakukan perawatan fisik, AWS akan menjadwalkan waktu untuk mengunjungi situs Anda untuk mengganti komponen yang gagal. Menyediakan kapasitas cadangan memungkinkan Anda untuk menjaga beban kerja Anda tangguh terhadap kegagalan host sementara server yang tidak sehat dikeluarkan dari layanan dan diganti.

## Mode kegagalan 4: Rak atau pusat data

Kegagalan rak dapat terjadi karena kehilangan total daya ke rak atau karena kegagalan lingkungan seperti hilangnya pendinginan atau kerusakan fisik pada pusat data akibat banjir atau gempa bumi.

Kekurangan dalam arsitektur distribusi daya pusat data atau kesalahan selama pemeliharaan daya pusat data standar dapat mengakibatkan hilangnya daya ke satu atau lebih rak atau bahkan seluruh pusat data.

Skenario ini dapat dikurangi dengan menyebarkan infrastruktur ke beberapa lantai pusat data atau lokasi yang independen satu sama lain dalam kampus atau area metro yang sama.

Mengambil pendekatan ini dengan AWS Outposts rak akan memerlukan pertimbangan yang cermat tentang bagaimana aplikasi dirancang dan didistribusikan untuk berjalan di beberapa Outposts logis terpisah untuk menjaga ketersediaan aplikasi.

## Mode kegagalan 5: Zona AWS Ketersediaan atau Wilayah

Setiap Pos Luar ditambahkan ke Availability Zone (AZ) tertentu dalam file. Wilayah AWS Kegagalan dalam jangkar AZ atau Wilayah induk dapat menyebabkan hilangnya manajemen Outpost dan mutabilitas dan dapat mengganggu komunikasi jaringan antara Outpost dan Region.

Mirip dengan kegagalan jaringan, kegagalan AZ atau Region dapat menyebabkan Outpost menjadi terputus dari Wilayah. Instans yang berjalan di Outpost terus berjalan dan dapat diakses dari jaringan lokal melalui Outpost Local Gateway (LGW) dan mungkin terganggu atau gagal jika mengandalkan layanan di Wilayah, seperti yang dijelaskan sebelumnya.

Untuk mengurangi dampak kegagalan AWS AZ dan Wilayah, Anda dapat menyebarkan beberapa Outpost yang masing-masing ditambahkan ke AZ atau Wilayah yang berbeda. Anda kemudian dapat merancang beban kerja Anda untuk beroperasi dalam model penyebaran Multi-outpost terdistribusi menggunakan banyak [mekanisme dan pola arsitektur](#) serupa yang Anda gunakan untuk merancang dan menerapkan hari ini. AWS

Bidang kontrol dari layanan yang berjalan AWS Outposts berada di Wilayah tempat ia berlabuh, menghasilkan ketergantungan baik pada layanan Zonal seperti Amazon dan Amazon EBS dan pada layanan Regional seperti EC2 Amazon RDS, Elastic Load Balancing dan Amazon EKS. Di Outposts, aplikasi dapat digunakan di bawah konsep [stabilitas statis](#) untuk membantu meningkatkan ketahanan untuk mengontrol gangguan pesawat.

# Membangun aplikasi HA dan solusi infrastruktur dengan AWS Outposts rak

Dengan AWS Outposts rack, Anda dapat membangun, mengelola, dan menskalakan aplikasi lokal yang sangat tersedia menggunakan layanan dan alat AWS cloud yang sudah dikenal. Penting untuk memahami arsitektur dan pendekatan HA cloud umumnya berbeda dari arsitektur HA lokal tradisional yang mungkin Anda jalankan di pusat data Anda hari ini.

Dengan penerapan aplikasi HA lokal tradisional, aplikasi digunakan di mesin virtual (VM). Sistem dan infrastruktur TI yang kompleks dikerahkan dan dipelihara untuk menjaga mesin virtual tetap berjalan dan sehat. VMs Sering memiliki identitas spesifik dan setiap VM mungkin memainkan peran penting dalam arsitektur aplikasi total.

Peran arsitektur digabungkan erat dengan identitas VM. Arsitek sistem memanfaatkan fitur infrastruktur TI untuk menyediakan lingkungan runtime VM yang sangat tersedia yang menyediakan setiap VM akses andal ke kapasitas komputasi, volume penyimpanan, dan layanan jaringan. Jika VM gagal, proses pemulihan otomatis atau manual dijalankan untuk mengembalikan VM yang gagal ke keadaan sehat, seringkali pada infrastruktur lain atau di pusat data lain sepenuhnya.

Arsitektur Cloud HA mengambil pendekatan yang berbeda. AWS layanan cloud menyediakan kemampuan komputasi, penyimpanan, dan jaringan yang andal. Komponen aplikasi digunakan untuk EC2 instance, kontainer, fungsi tanpa server, atau layanan terkelola lainnya.

Instance adalah instantiasi komponen aplikasi — mungkin salah satu dari banyak yang melakukan peran itu. Komponen aplikasi secara longgar digabungkan satu sama lain dan peran yang mereka mainkan dalam arsitektur aplikasi total. Identitas individu dari suatu contoh umumnya tidak penting. Contoh tambahan dapat dibuat atau dihancurkan untuk meningkatkan atau menurunkan dalam menanggapi permintaan. Contoh gagal atau contoh yang tidak sehat hanya diganti dengan contoh sehat baru.

AWS Outposts rack adalah layanan terkelola penuh yang memperluas AWS komputasi, penyimpanan, jaringan, database, dan layanan cloud lainnya ke lokasi lokal untuk pengalaman hybrid yang benar-benar konsisten. Anda tidak boleh menganggap layanan rak Outposts sebagai pengganti drop-in untuk sistem infrastruktur TI dengan mekanisme HA lokal tradisional. Mencoba menggunakan AWS layanan dan Outposts untuk mendukung arsitektur HA lokal tradisional adalah anti-pola.

Beban kerja yang berjalan di AWS Outposts rak menggunakan mekanisme HA cloud seperti [Amazon EC2 Auto Scaling](#) (untuk menskalakan secara horizontal untuk memenuhi tuntutan beban kerja),

pemeriksaan [kesehatan \(untuk mendeteksi dan menghapus instans yang tidak sehat\)EC2](#) , dan [Application Load Balancer \(untuk mengarahkan lalu lintas beban](#) kerja yang masuk ke instans yang diskalakan atau diganti). Saat memigrasikan aplikasi ke cloud, baik ke sebuah Wilayah AWS atau AWS Outposts rak, Anda harus memperbarui arsitektur aplikasi HA Anda untuk mulai memanfaatkan layanan cloud terkelola dan mekanisme HA cloud.

Bagian berikut memperkenalkan pola arsitektur, anti-pola, dan praktik yang direkomendasikan untuk menerapkan AWS Outposts rak di lingkungan lokal Anda untuk menjalankan beban kerja dengan persyaratan ketersediaan tinggi. Bagian ini memperkenalkan pola dan praktik; namun, mereka tidak memberikan detail konfigurasi dan implementasi. Anda harus membaca dan menjadi akrab dengan [AWS Outposts rak FAQs](#) dan [Panduan Pengguna](#) dan dokumentasi FAQs dan layanan untuk layanan yang berjalan di rak Outposts saat Anda mempersiapkan lingkungan Anda untuk rak Outposts dan aplikasi Anda untuk migrasi ke layanan. AWS

Topik

- [Jaringan](#)
- [Hitung](#)
- [Penyimpanan](#)
- [Basis Data](#)
- [Mode kegagalan yang lebih besar](#)

## Jaringan

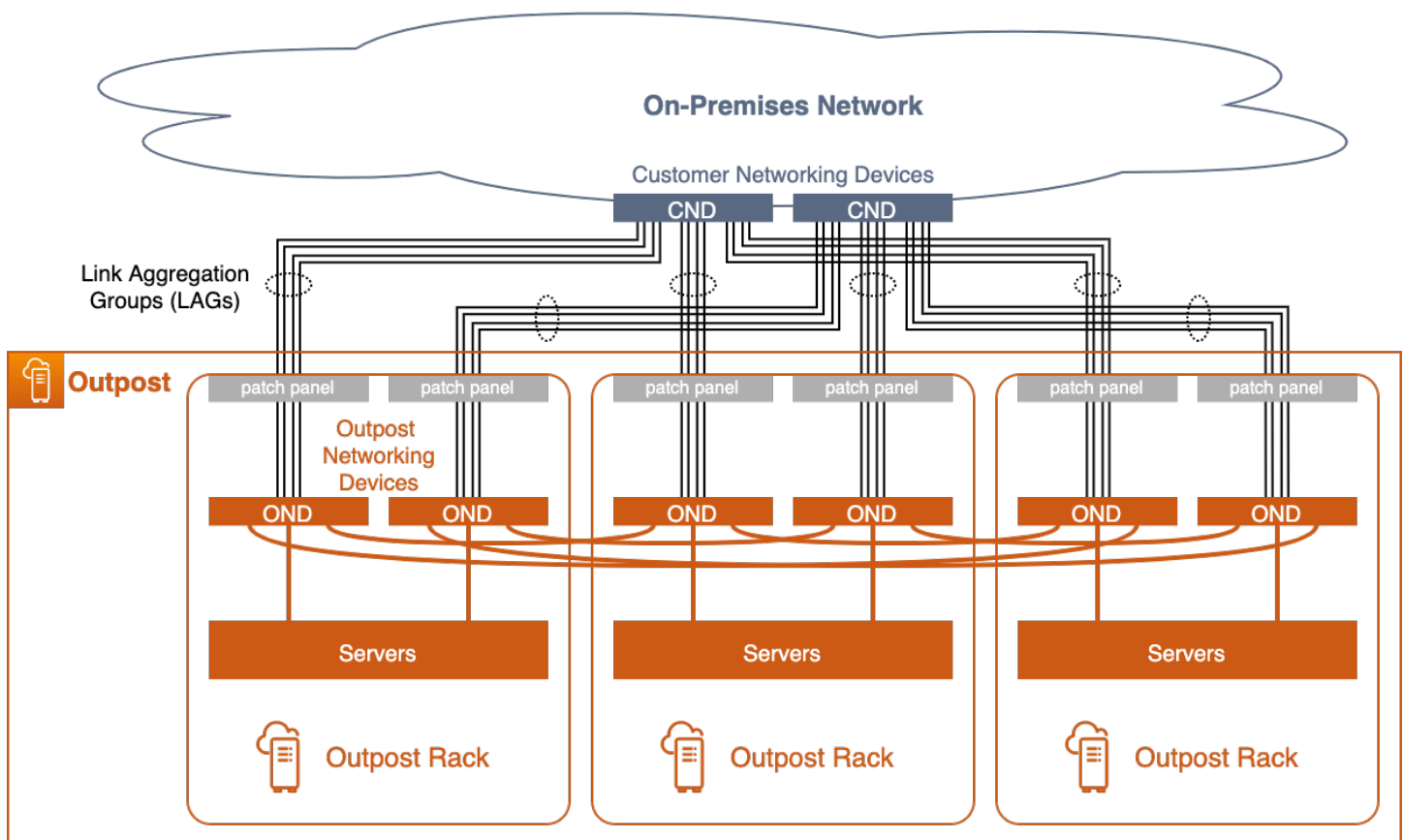
Penyebaran Outpost bergantung pada koneksi yang tangguh ke jangkar AZ agar manajemen, pemantauan, dan operasi layanan berfungsi dengan baik. Anda harus menyediakan jaringan lokal Anda untuk menyediakan koneksi jaringan redundan untuk setiap rak Outpost dan konektivitas yang andal kembali ke titik jangkar di cloud. AWS Pertimbangkan juga jalur jaringan antara beban kerja aplikasi yang berjalan di Outpost dan sistem lokal dan cloud lainnya yang berkomunikasi dengan mereka — bagaimana Anda akan merutekan lalu lintas ini di jaringan Anda?

Topik

- [Lampiran jaringan](#)
- [Konektivitas jangkar](#)
- [Perutean aplikasi/beban kerja](#)

## Lampiran jaringan

Setiap AWS Outposts rak dikonfigurasi dengan top-of-rack sakelar redundan yang disebut Outpost Networking Devices (OND). ONDs Server komputasi dan penyimpanan di setiap rak terhubung ke keduanya ONDs. Anda harus menghubungkan setiap OND ke sakelar terpisah yang disebut Customer Networking Device (CND) di pusat data Anda untuk menyediakan beragam jalur fisik dan logis untuk setiap rak Outpost. ONDs terhubung ke Anda CNDs dengan satu atau lebih koneksi fisik menggunakan kabel serat optik dan transceiver optik. [Koneksi fisik](#) dikonfigurasi dalam tautan [grup agregasi tautan logis \(LAG\)](#).



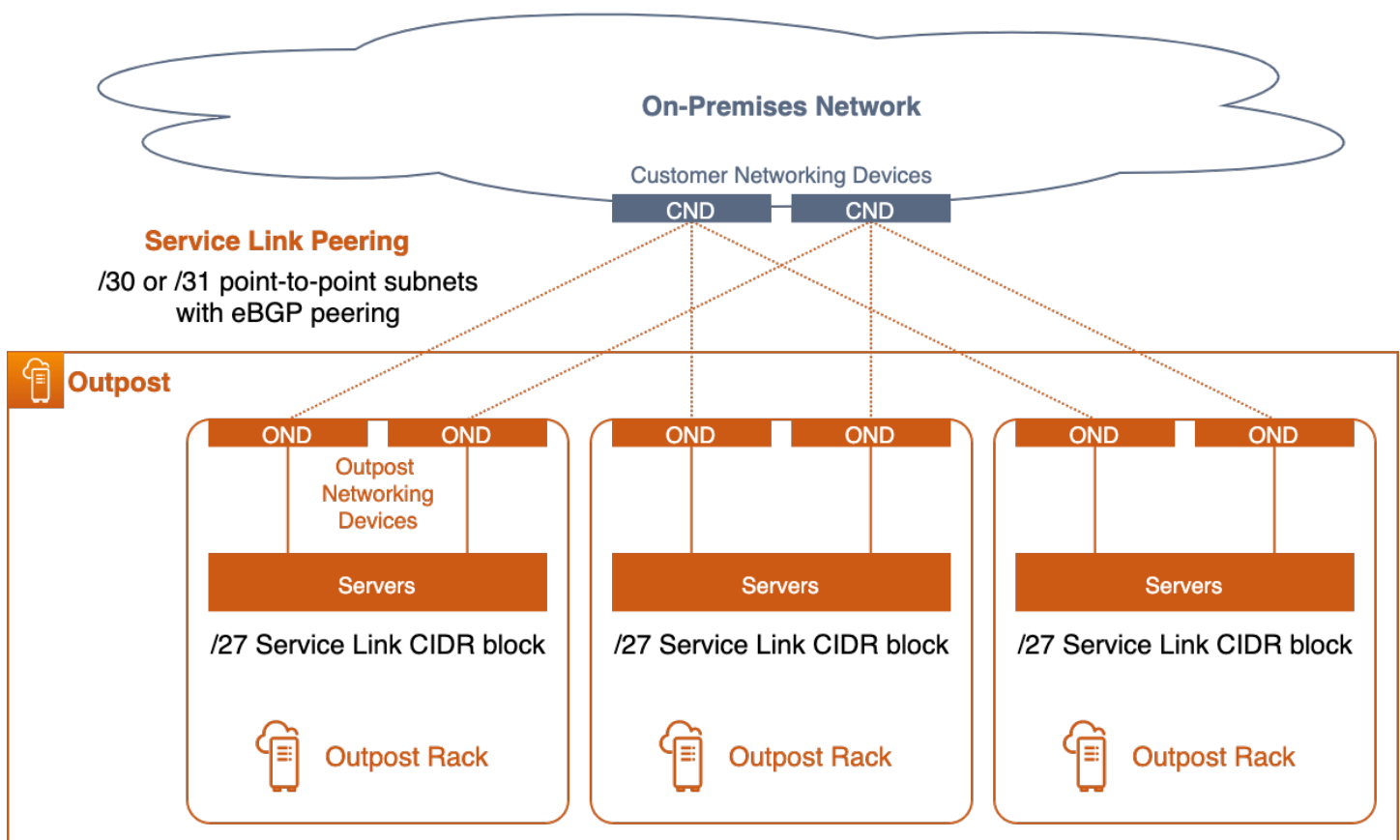
### Pos Luar Multi-rak dengan lampiran jaringan redundan

Tautan OND ke CND selalu dikonfigurasi dalam LAG - bahkan jika koneksi fisiknya adalah kabel serat optik tunggal. Mengkonfigurasi tautan sebagai grup LAG memungkinkan Anda meningkatkan bandwidth tautan dengan menambahkan koneksi fisik tambahan ke grup logis. Tautan LAG dikonfigurasi sebagai batang Ethernet IEEE 802.1q untuk mengaktifkan jaringan terpisah antara Outpost dan jaringan lokal.

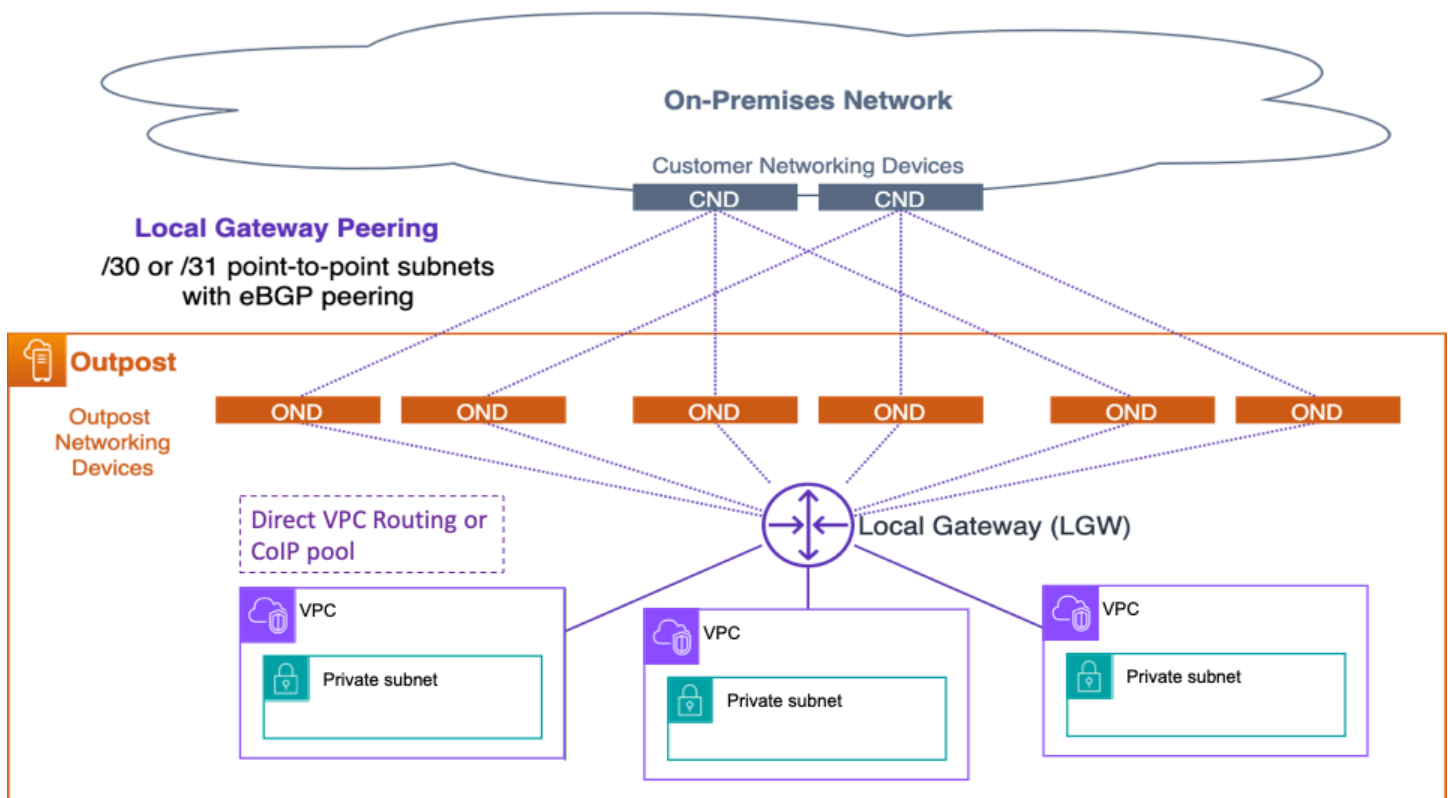
Setiap Outpost memiliki setidaknya dua jaringan terpisah secara logis yang perlu berkomunikasi dengan atau di seluruh jaringan pelanggan:

- Jaringan tautan layanan - mengalokasikan alamat IP tautan layanan ke server Outpost dan memfasilitasi komunikasi dengan jaringan lokal untuk memungkinkan server terhubung kembali ke titik jangkak Outpost di Wilayah. Ketika Anda memiliki beberapa implementasi rak dalam satu Outposts logis, Anda perlu menetapkan Service Link /26 CIDR untuk setiap Rack.
- Jaringan Local Gateway — memungkinkan komunikasi antara subnet VPC di Outpost dan jaringan lokal melalui Outpost Local Gateway (LGW).

Jaringan terpisah ini dilampirkan ke jaringan lokal dengan satu set [koneksi point-to-point IP](#) melalui tautan LAG. Setiap tautan OND ke CND LAG dikonfigurasi dengan VLAN IDs, point-to-point (/30 atau/31) subnet IP, dan peering eBGP untuk setiap jaringan terpisah (tautan layanan dan LGW). Anda harus mempertimbangkan tautan LAG, dengan subnetnya point-to-point VLANs dan, sebagai koneksi lapisan-2 tersegmentasi dan dirutekan layer-3. Koneksi IP yang dirutekan menyediakan jalur logis redundan yang memfasilitasi komunikasi antara jaringan terpisah di Outpost dan jaringan lokal.



## Layanan link peering



## Pengintip Gerbang Lokal

Anda harus menghentikan tautan LAG lapisan-2 (dan mereka VLANs) pada sakelar CND yang terpasang langsung dan mengonfigurasi antarmuka IP dan mengintip BGP pada sakelar CND. Anda tidak boleh menjembatani LAG VLANs antara sakelar pusat data Anda. Untuk informasi selengkapnya, lihat [Konektivitas lapisan jaringan](#) di Panduan AWS Outposts Pengguna.

Di dalam Outpost multi-rak yang logis, saling berhubungan ONDs secara berlebihan untuk menyediakan konektivitas jaringan yang sangat tersedia antara rak dan beban kerja yang berjalan di server. AWS bertanggung jawab atas ketersediaan jaringan di dalam Outpost.

## Praktik yang direkomendasikan untuk lampiran jaringan yang sangat tersedia tanpa ACE

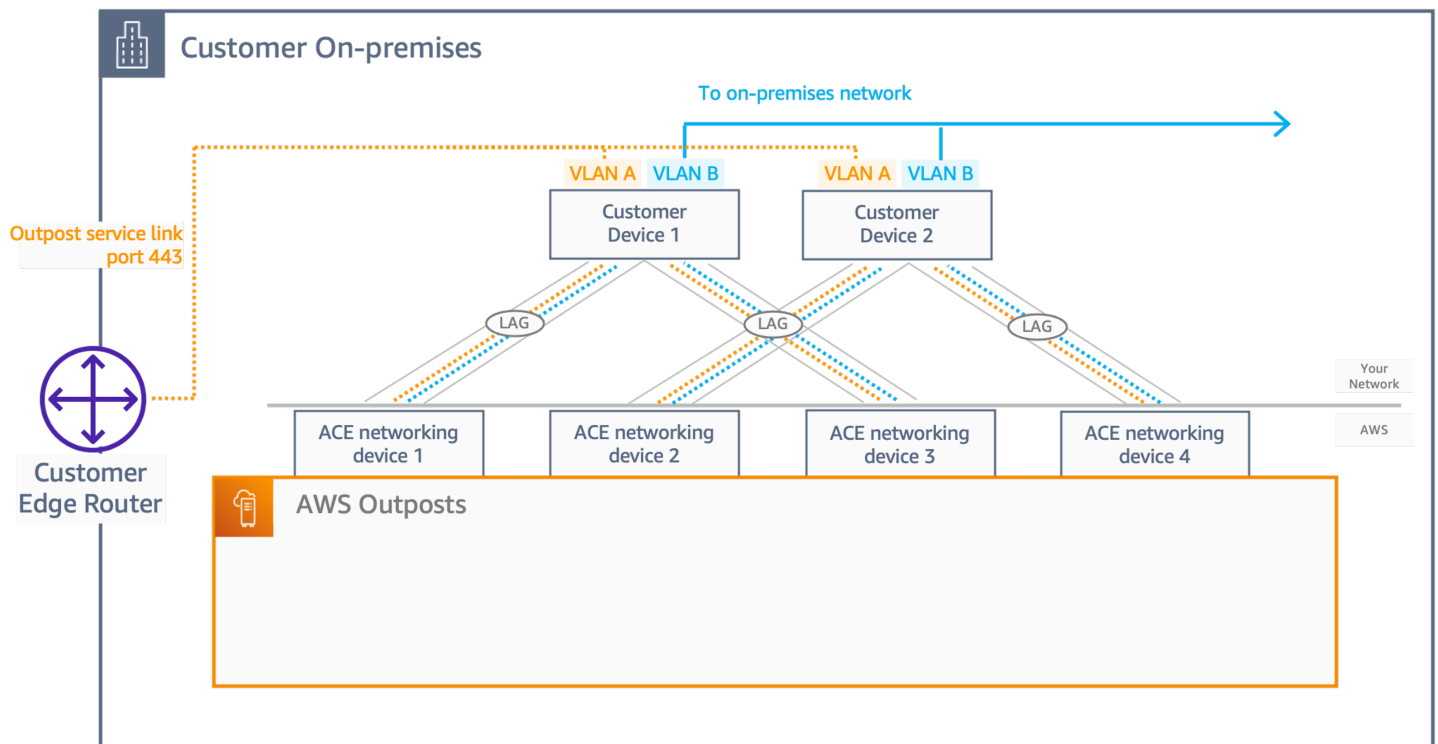
- Hubungkan setiap Outpost Networking Device (OND) di rak Outpost ke Customer Networking Device (CND) terpisah di pusat data.
- Hentikan tautan lapisan-2, subnet IP lapisan-3 VLANs, dan pengintipan BGP pada sakelar Customer Networking Device (CND) yang terpasang langsung. Jangan menjembatani OND ke CND VLANs antara CNDs atau di seluruh jaringan lokal.

- Tambahkan tautan ke Grup Agregasi Tautan (LAGs) untuk meningkatkan bandwidth yang tersedia antara Outpost dan pusat data. Jangan mengandalkan bandwidth agregat dari beragam jalur melalui keduanya ONDs.
- Gunakan beragam jalur melalui redundan ONDs untuk menyediakan konektivitas tangguh antara jaringan Outpost dan jaringan lokal.
- Untuk mencapai redundansi yang optimal dan memungkinkan pemeliharaan OND yang tidak mengganggu, kami menyarankan agar pelanggan mengonfigurasi iklan dan kebijakan BGP sebagai berikut:
  - Peralatan jaringan pelanggan harus menerima iklan BGP dari Outpost tanpa mengubah atribut BGP dan untuk mengaktifkan multipath/load-balancing to achieve optimal inbound traffic flows (from customer towards Outpost). AS-Path prepending is used for Outpost BGP prefixes to shift traffic away from a particular OND/uplink BGP jika pemeliharaan diperlukan. Jaringan pelanggan harus memilih rute dari Outpost dengan panjang AS-path 1 daripada rute dengan panjang AS-path 4, yaitu bereaksi terhadap as-Path prepending.
  - Jaringan pelanggan harus mengiklankan awalan BGP yang sama dengan atribut yang sama terhadap semua di Outpost. ONDs Secara default, beban jaringan Outpost menyeimbangkan lalu lintas keluar (menuju pelanggan) di antara semua uplink. Kebijakan perutean digunakan di sisi Outpost untuk mengalihkan lalu lintas dari OND tertentu jika pemeliharaan diperlukan. Awalan BGP yang sama dari sisi pelanggan pada semua ONDs diperlukan untuk melakukan pergeseran lalu lintas ini, dan melakukan pemeliharaan dengan cara yang tidak mengganggu. Ketika pemeliharaan diperlukan pada jaringan pelanggan, kami sarankan menggunakan AS-Path prepending untuk sementara mengalihkan lalu lintas dari uplink atau perangkat tertentu.

## Praktik yang direkomendasikan untuk lampiran jaringan yang sangat tersedia dengan ACE

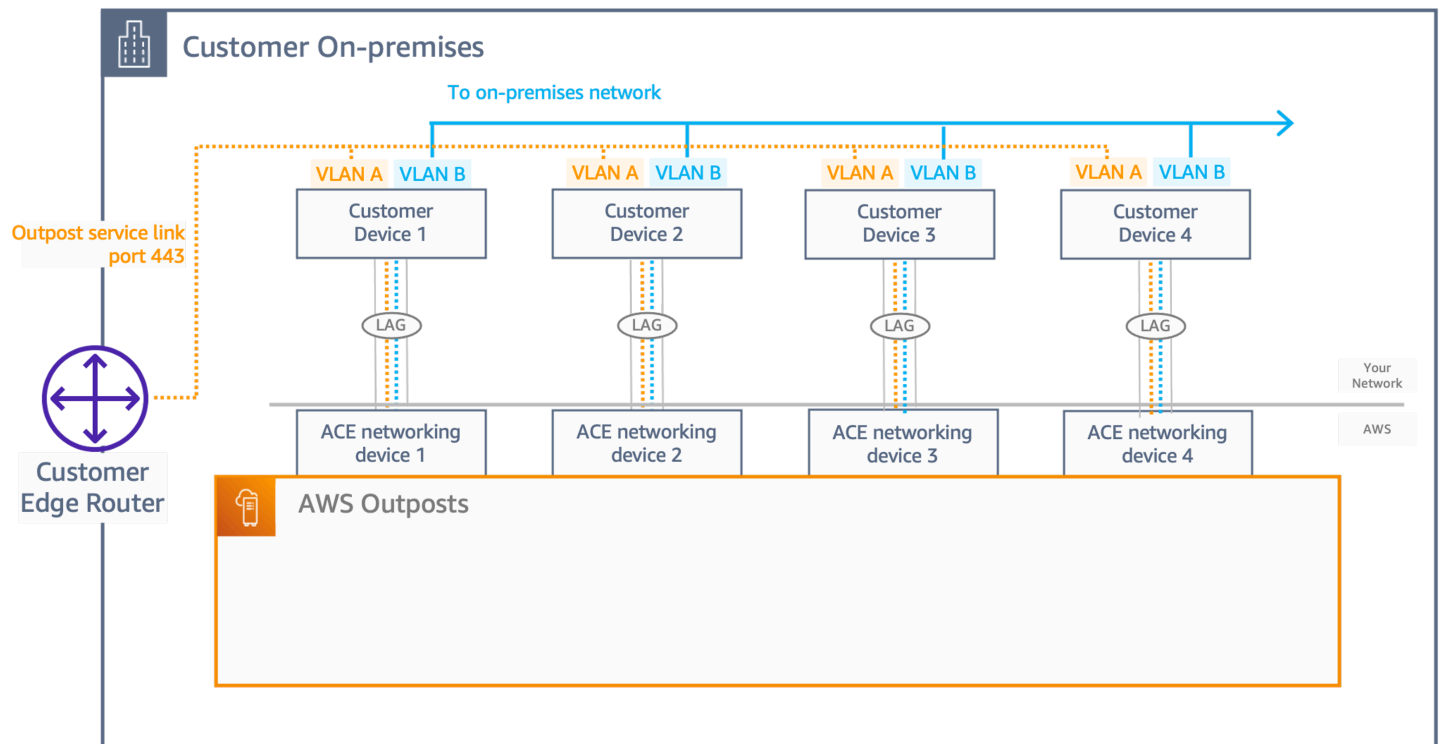
Untuk penyebaran multi-rak dengan empat atau lebih rak komputasi, Anda harus menggunakan rak Agregasi, Core, Edge (ACE), yang akan bertindak sebagai titik agregasi jaringan untuk mengurangi jumlah tautan serat ke perangkat jaringan lokal Anda. Rak ACE menyediakan konektivitas ke setiap rak Outposts, sehingga AWS akan memiliki alokasi antarmuka VLAN dan konfigurasi antara ONDs dan perangkat jaringan ACE. ONDs

Lapisan jaringan terisolasi untuk jaringan Service Link dan Local Gateway masih diperlukan terlepas dari apakah rak ACE digunakan atau tidak, yang bertujuan untuk memiliki subnet IP VLAN point-to-point (/30 atau /31), dan konfigurasi peering eBGP untuk setiap jaringan terpisah. Arsitektur yang diusulkan harus mengikuti salah satu dari dua arsitektur sebagai berikut:



## Perangkat jaringan dua pelanggan

- Dengan arsitektur ini, pelanggan harus memiliki dua perangkat jaringan (CND) untuk menghubungkan perangkat jaringan ACE, memberikan redundansi.
- Untuk setiap koneksi fisik, Anda harus mengaktifkan LAG (untuk meningkatkan bandwidth yang tersedia antara Outpost dan pusat data), bahkan jika itu adalah port fisik tunggal, dan itu akan membawa dua segmen jaringan, memiliki 2 point-to-point VLANs (/30 atau/31), dan konfigurasi eBGP antara dan. ACEs CNDs
- Dalam keadaan stabil, lalu lintas seimbang dengan beban mengikuti keseimbangan pola multipath Equal-cost (ECMP) diaktifkan, dan mengumumkan awalan pelanggan dengan to/from the customer network from the ACE layer, 25% traffic distribution across the ACE to customer. In order to allow this behavior, the eBGP peering's between ACEs and CNDs must have BGP multipath/load metrik BGP yang sama pada 4 koneksi peering eBGP.
- Untuk mencapai redundansi optimal dan memungkinkan pemeliharaan OND yang tidak mengganggu, kami menyarankan pelanggan untuk mengikuti rekomendasi ini:
  - Perangkat jaringan pelanggan harus mengiklankan awalan BGP yang sama dengan atribut yang sama terhadap semua di Outpost. ONDs
  - Perangkat jaringan pelanggan harus menerima iklan BGP dari Outpost tanpa mengubah atribut BGP dan mengaktifkan BGP multipath/load-balancing.



Perangkat jaringan empat pelanggan

Dengan arsitektur ini, pelanggan akan memiliki empat perangkat jaringan (CND) untuk menghubungkan perangkat jaringan ACE, memberikan redundansi dan logika jaringan yang sama, termasuk VLANs, eBGP, dan ECMP yang berlaku untuk arsitektur 2 CND.

## Konektivitas jangkar

[Tautan layanan Outpost](#) terhubung ke jangkar publik atau pribadi (bukan keduanya) di Availability Zone (AZ) tertentu di Wilayah induk Outpost. Server Outpost memulai koneksi VPN tautan layanan keluar dari alamat IP tautan layanan mereka ke titik jangkar di jangkar AZ. Koneksi ini menggunakan UDP dan TCP port 443. AWS bertanggung jawab atas ketersediaan titik jangkar di Wilayah.

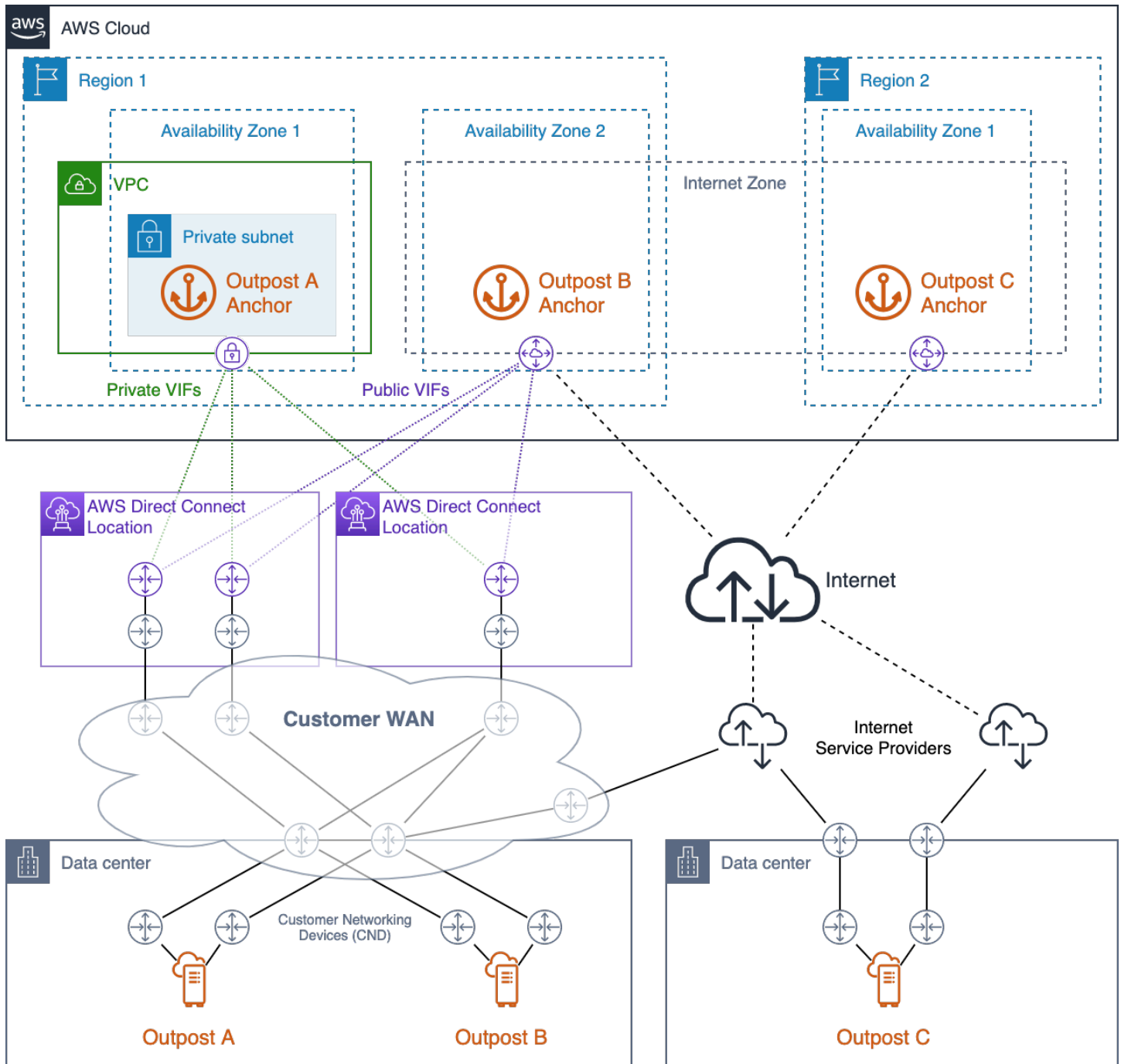
Anda harus memastikan alamat IP tautan layanan Outpost dapat terhubung melalui jaringan Anda ke titik jangkar di jangkar AZ. Alamat IP link layanan tidak perlu berkomunikasi dengan host lain di jaringan lokal Anda.

Titik jangkar publik berada di [rentang IP publik](#) Wilayah (dalam blok CIDR EC2 layanan) dan dapat diakses melalui internet atau [AWS Direct Connect](#)(DX) antarmuka virtual publik (). Penggunaan titik jangkar publik memungkinkan pemilihan jalur yang lebih fleksibel karena lalu lintas tautan layanan dapat diarahkan ke jalur yang tersedia yang berhasil mencapai titik jangkar di internet publik.

Poin jangkar pribadi memungkinkan Anda menggunakan rentang alamat IP Anda untuk konektivitas jangkar. Poin jangkar pribadi dibuat dalam [subnet pribadi dalam VPC khusus menggunakan alamat IP yang](#) ditetapkan pelanggan. VPC dibuat di Akun AWS yang memiliki sumber daya Outpost dan Anda bertanggung jawab untuk memastikan VPC tersedia dan dikonfigurasi dengan benar. [Gunakan Security Control Policy \(SCP\) di AWSOrigamiServiceGateway Organizations untuk mencegah pengguna menghapus Virtual Private Cloud \(VPC\) tersebut. Titik jangkar pribadi harus diakses menggunakan Direct Connect private. VIFs](#)

Anda harus menyediakan jalur jaringan redundan antara Outpost dan anchor point di Region dengan koneksi yang berakhir pada perangkat terpisah di lebih dari satu lokasi. Perutean dinamis harus dikonfigurasi untuk secara otomatis mengalihkan lalu lintas ke jalur alternatif ketika koneksi atau perangkat jaringan gagal. Anda harus menyediakan kapasitas jaringan yang cukup untuk memastikan bahwa kegagalan satu jalur WAN tidak membanjiri jalur yang tersisa.

Diagram berikut menunjukkan tiga Outposts dengan jalur jaringan redundan ke jangkar mereka AZs menggunakan AWS Direct Connect serta konektivitas internet publik. Pos Terdepan A dan Pos Terdepan B ditambahkan ke Zona Ketersediaan yang berbeda di Wilayah yang sama. Pos Terdepan A terhubung ke titik jangkar pribadi di AZ 1 dari wilayah 1. Outpost B terhubung ke titik jangkar publik di AZ 2 wilayah 1. Outpost C terhubung ke jangkar publik di AZ 1 wilayah 2.



Konektivitas jangkar yang sangat tersedia dengan AWS Direct Connect dan akses internet publik

Outpost A memiliki tiga jalur jaringan redundan untuk mencapai titik jangkar pribadinya. Dua jalur tersedia melalui sirkuit Direct Connect redundan di satu lokasi Direct Connect. Jalur ketiga tersedia melalui sirkuit Direct Connect di lokasi Direct Connect kedua. Desain ini menjaga lalu lintas tautan layanan Outpost A di jaringan pribadi dan menyediakan redundansi jalur yang memungkinkan kegagalan salah satu sirkuit Direct Connect atau kegagalan seluruh lokasi Direct Connect.

Outpost B memiliki empat jalur jaringan redundan untuk mencapai titik jangkar publiknya. Tiga jalur tersedia melalui VIFs penyediaan publik di sirkuit Direct Connect dan lokasi yang digunakan oleh Outpost A. Jalur keempat tersedia melalui WAN pelanggan dan internet publik. Lalu lintas tautan layanan Outpost B dapat diarahkan melalui jalur yang tersedia yang dapat berhasil mencapai titik jangkar di internet publik. Menggunakan jalur Direct Connect dapat memberikan latensi yang lebih konsisten dan ketersediaan bandwidth yang lebih tinggi, sedangkan jalur internet publik dapat digunakan untuk Disaster Recovery (DR) atau skenario augmentasi bandwidth.

Outpost C memiliki dua jalur jaringan redundan untuk mencapai titik jangkar publiknya. Outpost C ditempatkan di pusat data yang berbeda dari pusat data Outposts A dan B. Outpost C tidak memiliki sirkuit khusus yang terhubung ke WAN pelanggan. Sebaliknya, pusat data memiliki koneksi internet redundan yang disediakan oleh dua Penyedia Layanan Internet yang berbeda (ISPs). Lalu lintas tautan layanan Outpost C dapat diarahkan melalui salah satu jaringan ISP untuk mencapai titik jangkar di internet publik. Desain ini memungkinkan fleksibilitas untuk mengarahkan lalu lintas tautan layanan melalui koneksi internet publik yang tersedia. Namun, end-to-end jalur tergantung pada jaringan pihak ketiga publik di mana ketersediaan bandwidth dan latensi jaringan berfluktuasi.

Jalur jaringan antara Outpost dan titik jangkar tautan layanannya harus memenuhi spesifikasi bandwidth berikut:

- 500 Mbps - 1 Gbps bandwidth yang tersedia per rak Outpost (misalnya, 3 rak: 1,5 - 3 Gbps bandwidth yang tersedia)

## Praktik yang direkomendasikan untuk konektivitas jangkar yang sangat tersedia

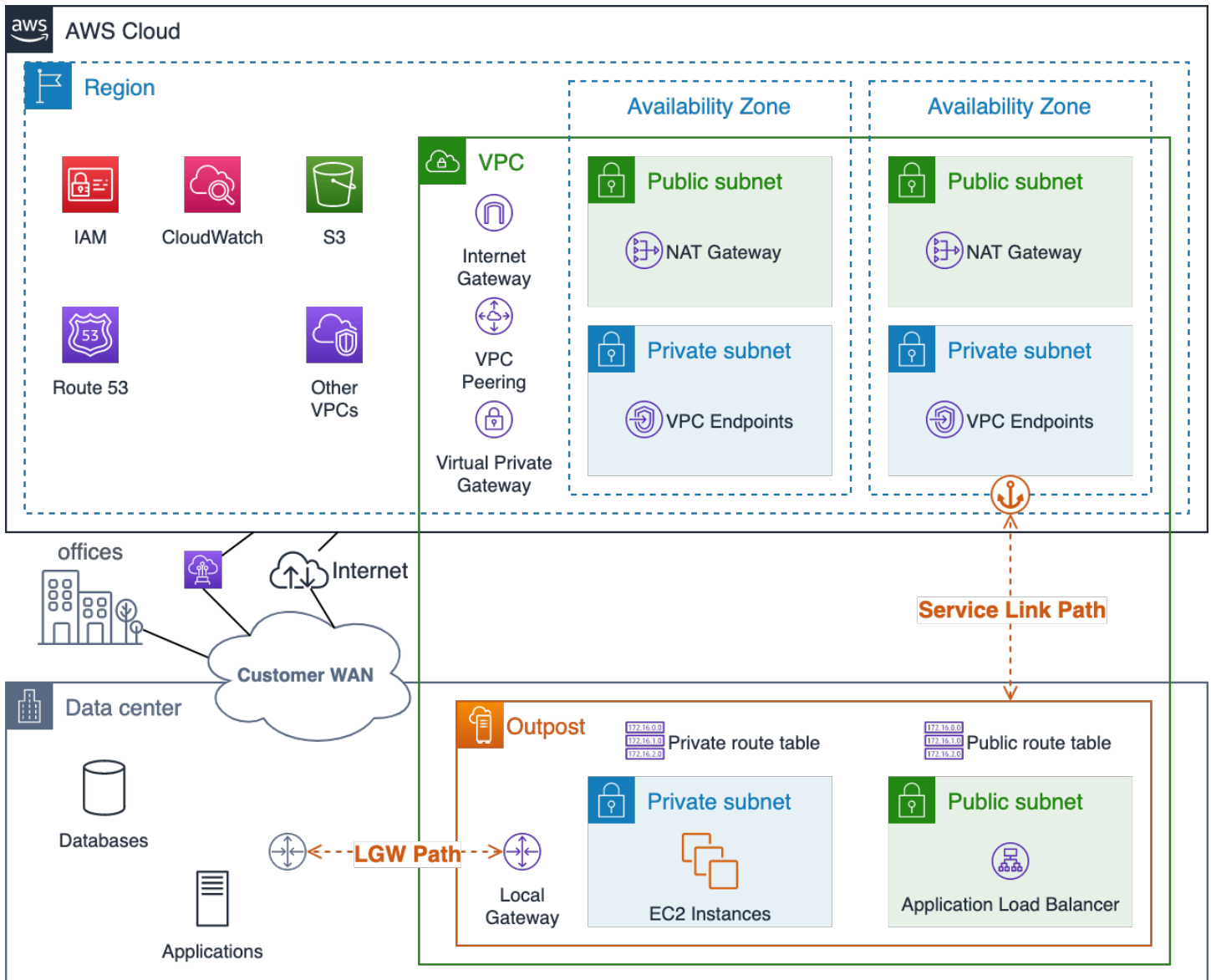
- Menyediakan jalur jaringan yang berlebihan antara setiap Pos Luar dan titik jangkar di Wilayah.
- Gunakan jalur Direct Connect (DX) untuk mengontrol latensi dan ketersediaan bandwidth.
- Pastikan bahwa port TCP dan UDP 443 terbuka (keluar) dari blok CIDR Outpost Service Link ke rentang [alamat EC2 IP di Wilayah induk](#). Pastikan port terbuka di semua jalur jaringan.
- Lacak rentang alamat EC2 IP Amazon di firewall Anda jika Anda menggunakan subset rentang CIDR untuk Wilayah tersebut.
- Pastikan setiap jalur memenuhi ketersediaan bandwidth dan persyaratan latensi.
- Gunakan perutean dinamis untuk mengotomatiskan pengalihan lalu lintas di sekitar kegagalan jaringan.
- Uji perutean lalu lintas tautan layanan di setiap jalur jaringan yang direncanakan untuk memastikan jalur berfungsi seperti yang diharapkan.

## Perutean aplikasi/beban kerja

Ada dua jalur keluar dari Outpost untuk beban kerja aplikasi:

- Jalur tautan layanan: Pertimbangkan bahwa lalu lintas aplikasi akan bersaing dengan Outposts mengontrol lalu lintas pesawat, selain membatasi [MTU hingga](#) 1300 byte.
- Jalur gateway lokal (LGW): Pertimbangkan bahwa jaringan lokal pelanggan memungkinkan akses ke kedua aplikasi yang ada di tempat dan juga di. Wilayah AWS

Anda mengonfigurasi tabel rute subnet Outpost untuk mengontrol jalur mana yang harus diambil untuk mencapai jaringan tujuan. Rute yang diarahkan ke LGW akan mengarahkan lalu lintas keluar dari Gateway Lokal dan ke jaringan lokal. Rute yang menunjuk ke layanan dan sumber daya di Wilayah, seperti Internet Gateway, NAT Gateway, Virtual Private Gateway, dan TGW, akan menggunakan [tautan layanan](#) untuk mencapai target ini. Jika Anda memiliki koneksi peering VPC dengan beberapa VPCs di Pos Luar yang sama, lalu lintas antara VPCs sisa-sisa di Pos Luar dan tidak menggunakan tautan layanan kembali ke Wilayah. Untuk informasi tentang peering VPC, lihat Connect [menggunakan peering VPCs VPC di Panduan Pengguna](#) Amazon VPC.



### Visualisasi tautan layanan Outpost dan jalur jaringan LGW

Anda harus berhati-hati saat merencanakan perutean aplikasi untuk mempertimbangkan operasi normal dan perutean terbatas dan ketersediaan layanan selama kegagalan jaringan. Jalur Tautan Layanan tidak tersedia saat Pos Luar terputus dari Wilayah.

Anda harus menyediakan beragam jalur dan mengonfigurasi perutean dinamis antara Outpost LGW dan aplikasi, sistem, dan pengguna lokal penting Anda. Jalur jaringan redundan memungkinkan jaringan untuk merutekan lalu lintas di sekitar kegagalan dan memastikan bahwa sumber daya lokal akan dapat berkomunikasi dengan beban kerja yang berjalan di Outpost selama kegagalan jaringan sebagian.

Konfigurasi rute VPC pos terdepan bersifat statis. Anda mengonfigurasi tabel perutean subnet melalui alat, Konsol Manajemen AWS CLI, APIs, dan Infrastructure as Code (IaC) lainnya; Namun, Anda tidak akan dapat memodifikasi tabel perutean subnet selama peristiwa pemutusan. Anda harus membangun kembali konektivitas antara Outpost dan Region untuk memperbarui tabel rute. Gunakan rute yang sama untuk operasi normal seperti yang Anda rencanakan untuk digunakan selama peristiwa pemutusan sambungan.

Sumber daya di Outpost dapat menjangkau internet melalui tautan layanan dan Internet Gateway (IGW) di Wilayah atau melalui jalur Local Gateway (LGW). Merutekan lalu lintas internet melalui jalur LGW dan jaringan lokal memungkinkan Anda menggunakan titik masuk/keluar internet lokal yang ada dan dapat memberikan latensi yang lebih rendah, biaya keluar AWS data yang lebih tinggi MTUs, dan berkurang jika dibandingkan dengan menggunakan jalur tautan layanan ke IGW di Wilayah.

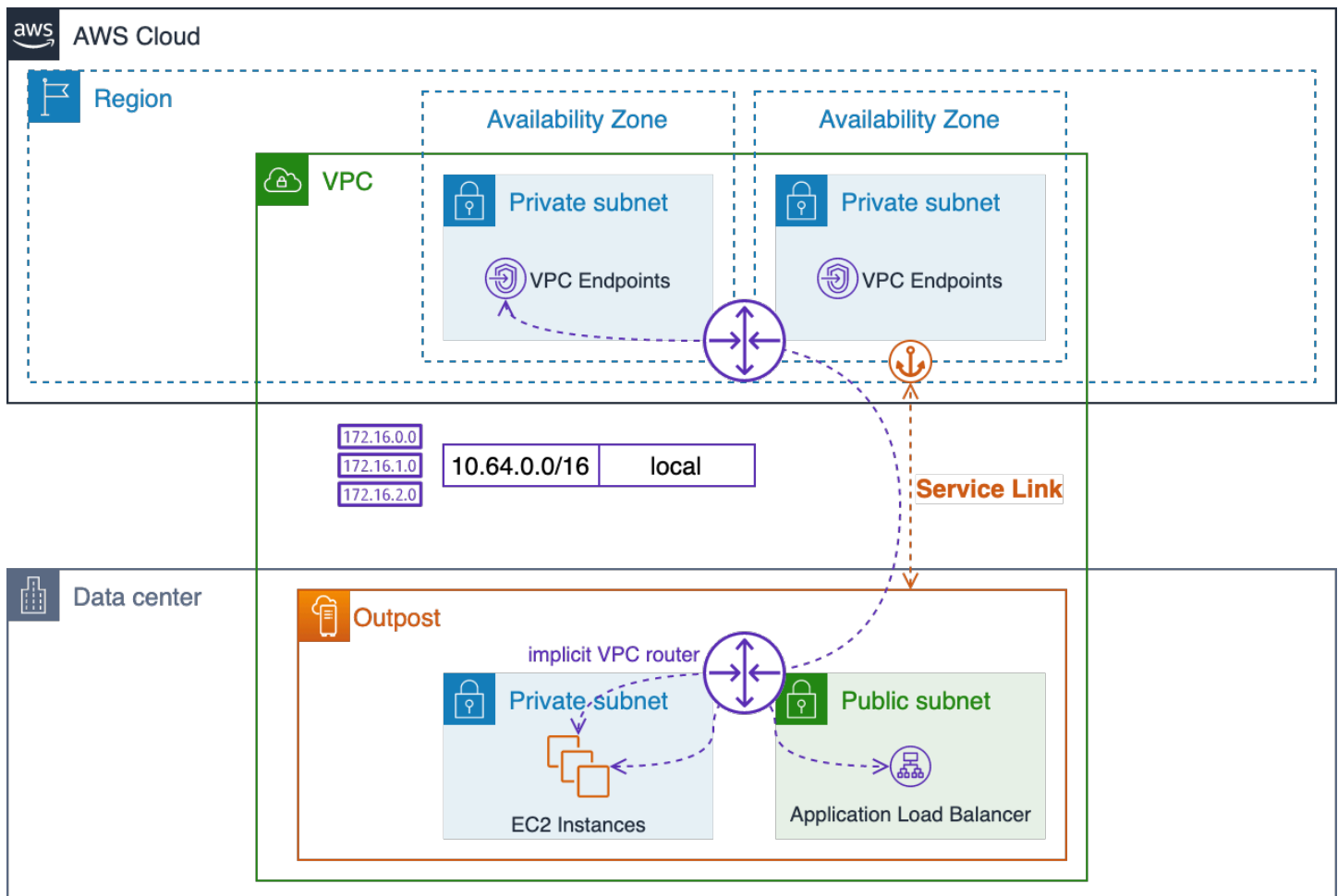
Jika aplikasi Anda harus berjalan di tempat dan harus dapat diakses dari internet publik, Anda harus merutekan lalu lintas aplikasi melalui koneksi internet lokal Anda ke LGW untuk menjangkau sumber daya di Pos Luar.

Meskipun Anda dapat mengonfigurasi subnet di Outpost seperti subnet publik di Wilayah, ini mungkin merupakan praktik yang tidak diinginkan untuk sebagian besar kasus penggunaan. Lalu lintas internet masuk akan masuk melalui Wilayah AWS dan diarahkan melalui tautan layanan ke sumber daya yang berjalan di Outpost.

Lalu lintas respons pada gilirannya akan diarahkan melalui tautan layanan dan kembali keluar melalui koneksi internet. Wilayah AWS Pola lalu lintas ini dapat menambah latensi dan akan menimbulkan biaya keluar data karena lalu lintas meninggalkan Wilayah dalam perjalanan ke Pos Terdepan dan ketika lalu lintas kembali kembali melalui Wilayah dan keluar ke internet. Jika aplikasi Anda dapat berjalan di Wilayah, Wilayah adalah tempat terbaik untuk menjalankannya.

Lalu lintas antara sumber daya VPC (dalam VPC yang sama) akan selalu mengikuti rute CIDR VPC lokal dan dirutekan antar subnet oleh router VPC implisit.

Misalnya, lalu lintas antara EC2 instans yang berjalan di Outpost dan Titik Akhir VPC di Wilayah akan selalu diarahkan melalui tautan layanan.



Perutean VPC lokal melalui router implisit

Praktik yang direkomendasikan untuk perutean aplikasi/beban kerja

- Gunakan jalur Local Gateway (LGW) alih-alih jalur tautan layanan jika memungkinkan.
- Rutekan lalu lintas internet melalui jalur LGW.
- Konfigurasi tabel perutean subnet Outpost dengan serangkaian rute standar - mereka akan digunakan untuk operasi normal dan selama peristiwa pemutusan sambungan.
- Menyediakan jalur jaringan redundan antara Outpost LGW dan sumber daya aplikasi lokal yang penting. Gunakan perutean dinamis untuk mengotomatiskan pengalihan lalu lintas di sekitar kegagalan jaringan lokal.

# Hitung

Sementara EC2 kapasitas Amazon tampaknya tak terbatas, kapasitas di Outposts terbatas. Wilayah AWS Anda bertanggung jawab untuk merencanakan dan mengelola kapasitas komputasi penyebaran Outposts Anda.

Topik

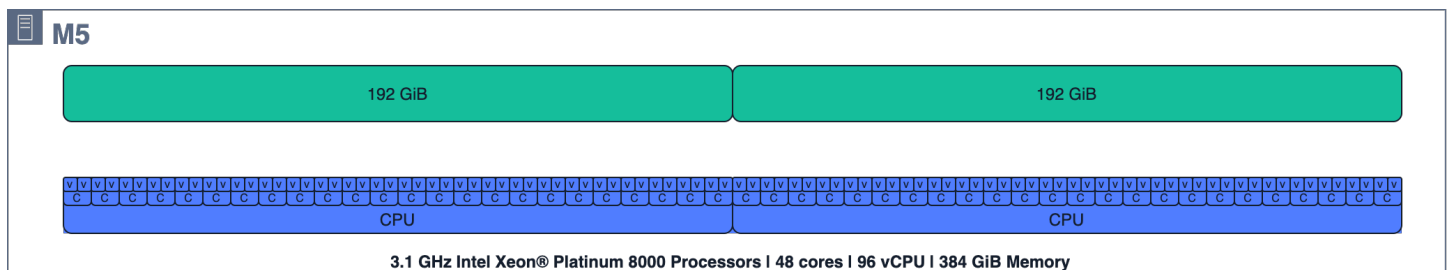
- [Perencanaan kapasitas](#)
- [Manajemen kapasitas](#)
- [Penempatan instans](#)

## Perencanaan kapasitas

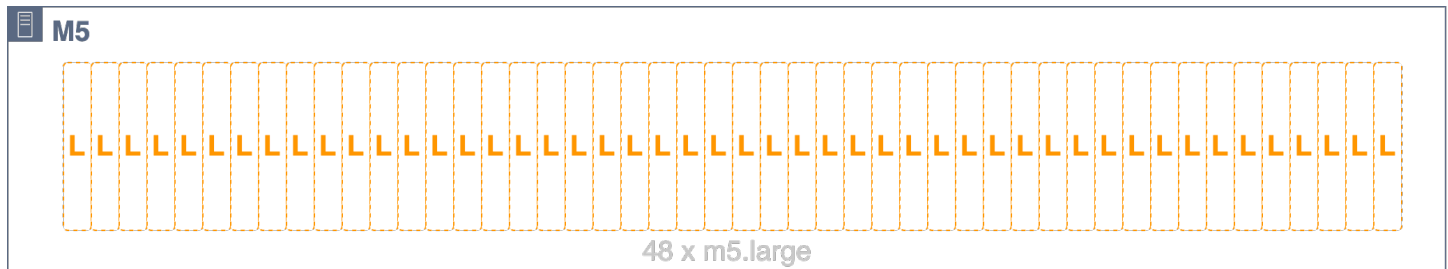
Sementara EC2 kapasitas Amazon tampaknya tak terbatas, kapasitas di Outposts terbatas — dibatasi oleh total volume kapasitas komputasi yang dipesan. Wilayah AWS Anda bertanggung jawab untuk merencanakan dan mengelola kapasitas komputasi penyebaran Outposts Anda. Anda harus memesan kapasitas komputasi yang cukup untuk mendukung model ketersediaan N+M, di mana N adalah jumlah server yang diperlukan dan M adalah jumlah server cadangan yang disediakan untuk mengakomodasi kegagalan server. N+1 dan N+2 adalah tingkat ketersediaan yang paling umum.

Setiap host (C5M5,R5,, dll.) mendukung satu keluarga EC2 instance. Sebelum dapat meluncurkan instance di server EC2 komputasi, Anda harus menyediakan tata letak slotting yang menentukan [ukuran EC2 instans](#) yang ingin disediakan oleh setiap server. AWS mengkonfigurasi setiap server dengan tata letak slotting yang diminta.

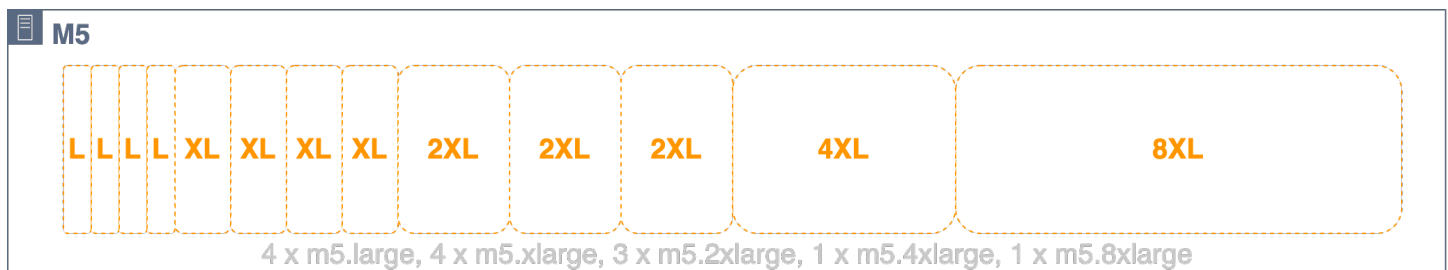
Host dapat ditempatkan secara homogen di mana semua slot memiliki ukuran instance yang sama (misalnya, 48 m5.large slot) atau ditempatkan secara heterogen dengan campuran jenis instance (misalnya, 4, 4m5.large, 3, 1 m5.2xlarge, 5.4xlarge, dan 1m5.8xlarge) — lihat tiga gambar berikutnya untuk visualisasi konfigurasi slotting ini. m5.xlarge



*m5.24xlarge* sumber daya komputasi host



*m5.24xlarge* host secara homogen dimasukkan ke dalam 48 slot *m5.large*



*m5.24xlarge* host secara heterogen ditempatkan menjadi 4 *m5.large*, 4, 3 *m5.2xlarge*, 1 *m5.xlarge*, dan 1 *m5.4xlarge* slot *m5.8xlarge*

Kapasitas host penuh tidak harus ditempatkan. Slot dapat ditambahkan ke host yang memiliki kapasitas yang tidak terisi. Anda dapat memodifikasi tata letak slotting dengan menggunakan Manajemen Kapasitas APIs atau UIs untuk AWS Outposts dan membuat tugas kapasitas baru. Untuk informasi selengkapnya, lihat [Manajemen kapasitas untuk AWS Outposts](#) di panduan AWS Outposts pengguna untuk rak. Anda mungkin diminta untuk mematikan atau memulai ulang instance tertentu untuk menyelesaikan tugas kapasitas baru jika tata letak slotting baru tidak dapat diterapkan saat slot tertentu ditempati oleh instance yang sedang berjalan. `CreateCapacityTaskAPI` memungkinkan Anda untuk mengekspresikan jumlah setiap ukuran instans yang harus ada pada ID Outpost yang ditunjukkan, dan jika tugas tidak dapat diselesaikan karena menjalankan instance, mengembalikan instance yang harus dihentikan untuk memenuhi permintaan. Pada titik ini, Anda dapat secara opsional menunjukkan bahwa Anda ingin melihat opsi tambahan “N” jika Anda memilih untuk tidak menghentikan salah satu instance yang dikembalikan, dan Anda juga dapat menunjukkan ID EC2 instance, tag EC2 instance, akun, atau layanan yang seharusnya tidak disarankan sebagai instance untuk dimatikan untuk memenuhi permintaan tugas kapasitas. Setelah membuat pilihan opsi yang ingin Anda gunakan, sebaiknya gunakan parameter `Dry Run` untuk memvalidasi perubahan yang diusulkan dan memahami dampak potensial sebelum menerapkan.

Semua host menyumbangkan slot yang disediakan ke kumpulan EC2 kapasitas di Outpost, dan semua slot dari jenis dan ukuran instans tertentu dikelola sebagai kumpulan kapasitas

tunggal EC2 . Misalnya, host berslot heterogen sebelumnya dengan `m5.large`, `m5.xlarge`, `m5.2xlarge`, `m5.4xlarge`, dan `m5.8xlarge` slot akan menyumbangkan slot ini ke lima kumpulan EC2 kapasitas — satu kumpulan untuk setiap jenis dan ukuran instans. Kumpulan ini dapat tersebar di beberapa host, dan penempatan instance harus menjadi pertimbangan untuk mencapai ketersediaan beban kerja yang tinggi.

Penting untuk mempertimbangkan slotting host dan kolam EC2 kapasitas saat merencanakan kapasitas cadangan untuk ketersediaan host N+M. AWS mendeteksi ketika host gagal atau terdegradasi dan menjadwalkan kunjungan situs untuk menggantikan host yang gagal. Anda harus merancang kumpulan EC2 kapasitas Anda untuk mentolerir kegagalan setidaknya satu server dari setiap keluarga instance (N+1) di Outpost. Dengan tingkat ketersediaan host minimum ini, ketika host gagal atau perlu dikeluarkan dari layanan, Anda dapat memulai ulang instance yang gagal atau terdegradasi pada slot cadangan host yang tersisa dari keluarga yang sama.

Perencanaan untuk ketersediaan N+M sederhana ketika Anda memiliki host yang ditempatkan secara homogen atau kelompok host slotted heterogen dengan tata letak slotting yang identik. Anda cukup menghitung jumlah host (N) kebutuhan Anda untuk menjalankan semua beban kerja Anda dan kemudian menambahkan (M) host tambahan untuk memenuhi persyaratan Anda untuk ketersediaan server selama kegagalan dan peristiwa pemeliharaan.

Konfigurasi slotting berikut tidak dapat digunakan karena batas NUMA:

- 3 `m5.8xlarge`
- 1 `m5.16xlarge` dan 1 `m5.8xlarge`

Konsultasikan dengan Akun AWS tim Anda untuk memvalidasi konfigurasi slot AWS Outposts rak yang Anda rencanakan.

Pada gambar berikut, empat `m5.24xlarge` host ditempatkan secara heterogen dengan tata letak slotting yang identik. Keempat host membuat lima kolam EC2 kapasitas. Setiap pool berjalan pada pemanfaatan maksimum (75%) untuk menjaga ketersediaan N+1 untuk instance yang berjalan pada keempat host ini. Jika ada host yang gagal, ada cukup ruang untuk memulai ulang instance yang gagal pada host yang tersisa.



### Visualisasi slot EC2 host, instance berjalan, dan kolom slot

Untuk tata letak slotting yang lebih kompleks, di mana host tidak ditempatkan secara identik, Anda perlu menghitung ketersediaan N+M untuk setiap kumpulan kapasitas. EC2 Anda dapat menggunakan rumus berikut untuk menghitung berapa banyak host (yang menyumbangkan slot ke kumpulan EC2 kapasitas tertentu) yang dapat gagal dan masih mengizinkan host yang tersisa untuk membawa instance yang sedang berjalan:

$$M = \left\lceil \frac{\text{poolSlots}_{\text{available}}}{\text{serverSlots}_{\text{max}}} \right\rceil$$

Di mana:

- $\text{PoolSlots}_{\text{available}}$  adalah jumlah slot yang tersedia di kolom EC2 kapasitas yang diberikan (jumlah total slot di kolom dikurangi jumlah instance yang berjalan)
- $\text{ServerSlots}_{\text{max}}$  adalah jumlah maksimum slot yang disumbangkan oleh host mana pun ke kumpulan kapasitas yang diberikan EC2
- M adalah jumlah host yang dapat gagal dan masih memungkinkan host yang tersisa untuk membawa instance yang sedang berjalan

Contoh: Sebuah Outpost memiliki tiga host yang menyumbangkan slot ke kolom m5.2xlarge kapasitas. Yang pertama menyumbang 4 slot, yang kedua menyumbang 3 slot, dan host ketiga

menyumbang 2 slot. Kolam `m5.2xlarge` instance di Outpost memiliki kapasitas total 9 slot ( $4 + 3 + 2$ ). Outpost memiliki 4 `m5.2xlarge` instance yang berjalan. Berapa banyak host yang mungkin gagal dan masih mengizinkan host yang tersisa untuk membawa instance yang sedang berjalan?

$$poolSlots_{available} = total\ capacity - running\ instances = 9 - 4 = 5$$

$$serverSlots_{max} = \max([4, 3, 2]) = 4$$

$$M = \left\lfloor \frac{poolSlots_{available}}{serverSlots_{max}} \right\rfloor = \left\lfloor \frac{5}{4} \right\rfloor = [1.25] = 1$$

Jawaban: Anda dapat kehilangan salah satu host dan masih membawa instance yang berjalan pada host yang tersisa.

Praktik yang direkomendasikan untuk perencanaan kapasitas komputasi

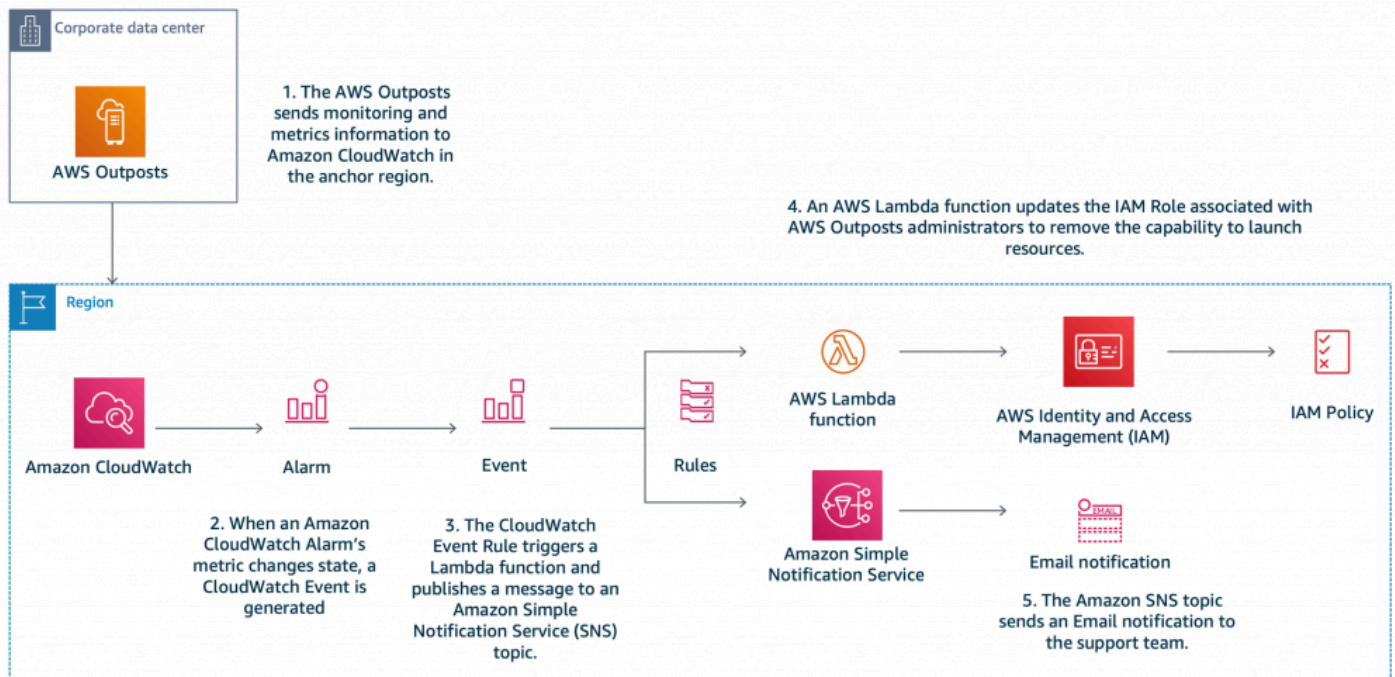
- Ukur kapasitas komputasi Anda untuk memberikan redundansi N+M untuk setiap kumpulan EC2 kapasitas di Outpost.
  - Menyebarkan server N+M untuk server slotted heterogen homogen atau identik.
  - Hitung ketersediaan N+M untuk setiap kumpulan EC2 kapasitas dan pastikan bahwa setiap kolom memenuhi persyaratan ketersediaan Anda.

## Manajemen kapasitas

Anda dapat memantau pemanfaatan kumpulan EC2 instans Outpost di Konsol Manajemen AWS dan melalui metrik Amazon CloudWatch. Hubungi Enterprise Support untuk mengambil atau mengubah tata letak slotting untuk Outposts Anda.

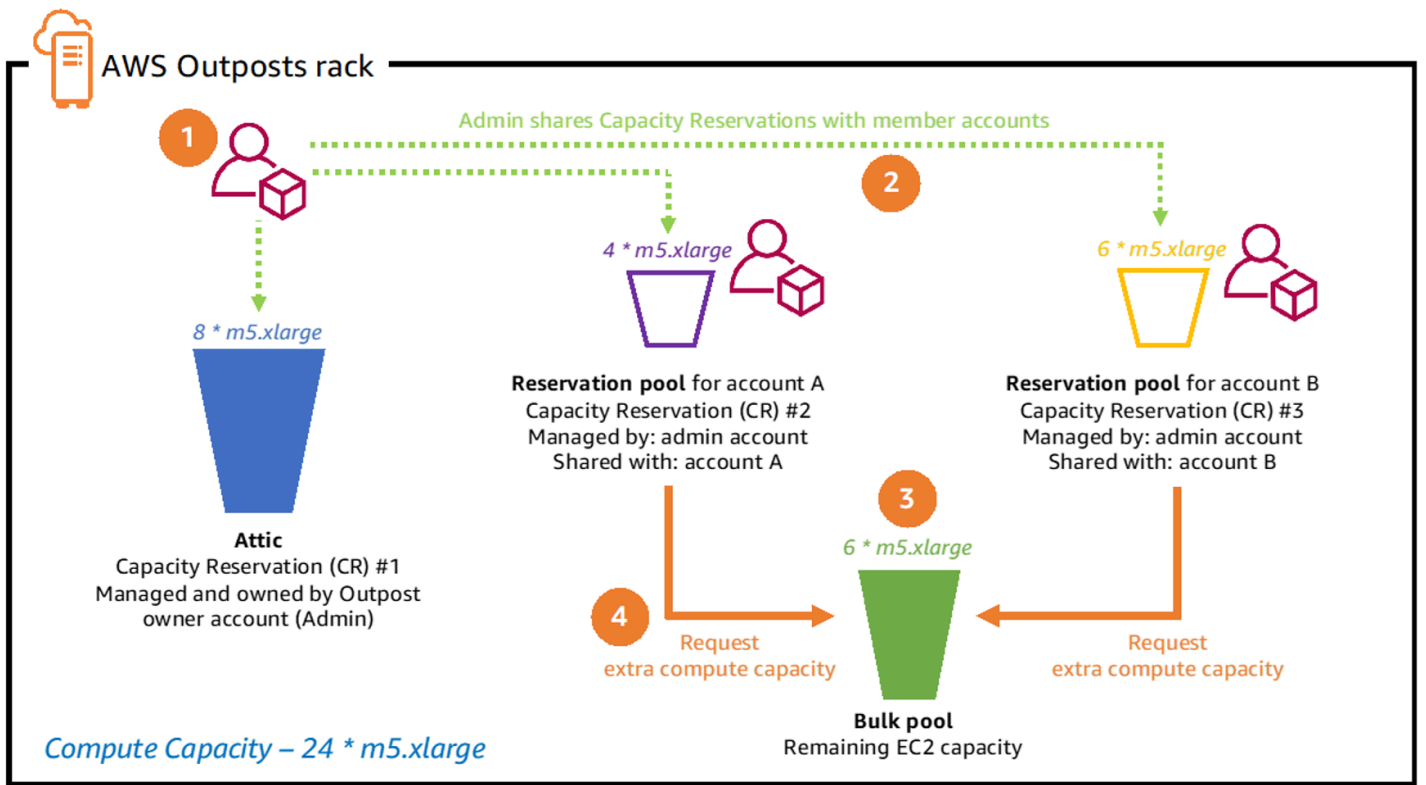
Anda menggunakan mekanisme [auto recovery dan EC2 Auto Scaling instance](#) yang sama untuk memulihkan atau mengganti instance yang terkena dampak kegagalan server dan peristiwa pemeliharaan. Anda harus memantau dan mengelola kapasitas Outpost Anda untuk memastikan kapasitas cadangan yang cukup selalu tersedia untuk mengakomodasi kegagalan server. [Mengelola AWS Outposts kapasitas Anda menggunakan Amazon CloudWatch dan](#) posting AWS Lambda blog

menyediakan tutorial langsung yang menunjukkan cara menggabungkan AWS CloudWatch dan mengelola kapasitas Outpost Anda AWS Lambda untuk mempertahankan ketersediaan instans.

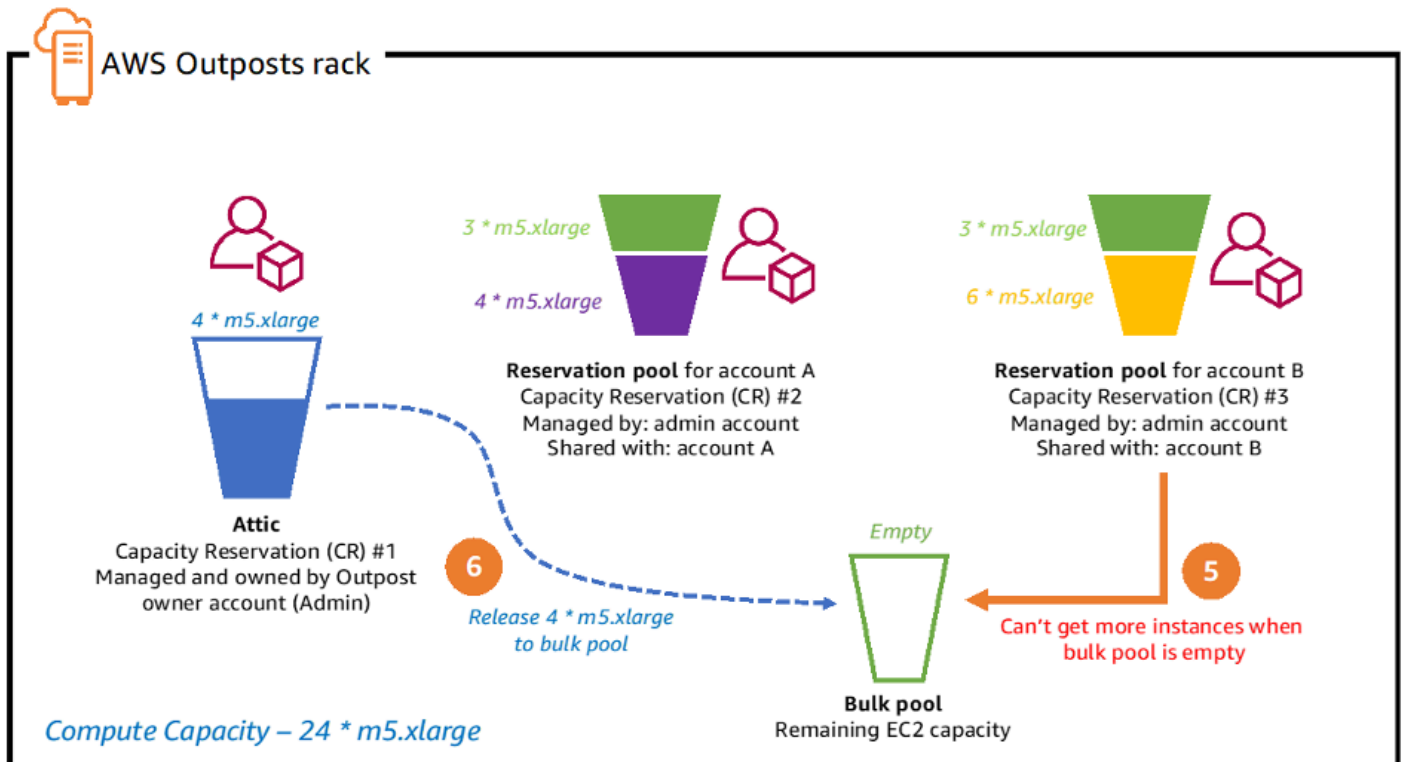


## Mengelola AWS Outposts kapasitas dengan Amazon CloudWatch dan AWS Lambda

Reservasi Kapasitas dapat digunakan dalam lingkungan multi-akun untuk mengontrol berapa banyak kapasitas komputasi Outpost Anda digunakan oleh satu akun, atau unit AWS Organisasi (OU) yang berisi beberapa akun. Anda dapat membuat reservasi kapasitas untuk Amazon EC2 di Outposts, serta Outposts yang didukung seperti Amazon Elastic Kubernetes Service (EKS, Amazon Elastic Container Service (ECS), dan Amazon Elastic Map Reduce (EMR). Layanan AWS Pemesanan kapasitas dibuat dan dibagikan ke akun melalui AWS Resource Access Manager (AWS RAM) di akun pemilik Outpost. [Membuat kuota komputasi di AWS Outposts rak dengan berbagi Reservasi EC2 Kapasitas](#) memberikan tutorial langsung dan panduan tambahan untuk menerapkan reservasi kapasitas dengan Outpost Anda untuk tujuan manajemen kapasitas.



Capacity Reservation sharing process steps 1-4



## Capacity Reservation sharing process steps 5-6

### Praktik yang direkomendasikan untuk manajemen kapasitas komputasi

- Konfigurasi EC2 instans Anda dalam grup Auto Scaling atau gunakan pemulihan otomatis instans untuk memulai ulang instance yang gagal.
- Otomatiskan pemantauan kapasitas untuk penerapan Outpost Anda dan konfigurasi notifikasi dan (opsional) respons otomatis untuk alarm kapasitas.
- Gunakan Reservasi Kapasitas untuk memiliki kontrol terperinci atas berapa banyak kapasitas komputasi yang dibagikan ke akun lain dalam Organisasi Anda. AWS

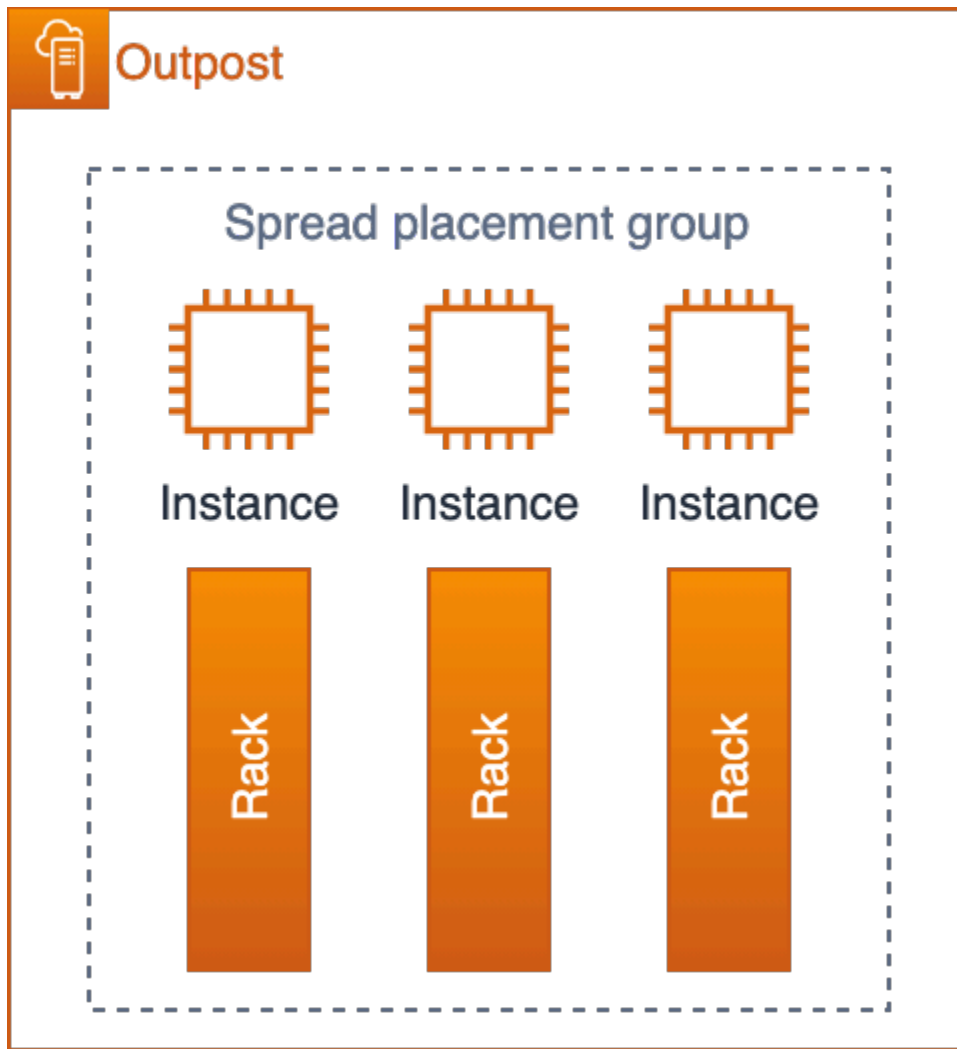
### Penempatan instans

Outposts memiliki jumlah host komputasi yang terbatas. Jika aplikasi Anda menyebarkan beberapa instance terkait di Outposts; tanpa konfigurasi tambahan, instance dapat diterapkan pada host yang sama atau pada host di rak yang sama. Saat ini, ada tiga mekanisme yang dapat Anda gunakan untuk mendistribusikan instans guna mengurangi risiko menjalankan instans terkait pada infrastruktur yang sama:

Penerapan Multi-Outpost — mirip dengan strategi Multi-AZ di Wilayah, Anda dapat menyebarkan Outposts untuk memisahkan pusat data dan menyebarkan sumber daya aplikasi ke Outposts tertentu. Ini memungkinkan Anda untuk menjalankan instance di Outpost yang diinginkan (satu set rak logis). [Komunikasi Intra-VPC](#) di Beberapa Outposts dengan Direct VPC Routing adalah strategi lain yang dapat digunakan untuk menyebarkan beban kerja di beberapa Outposts dalam VPC yang sama menggunakan Outpost local gateway (LGW) untuk membuat rute antara subnet di Outposts. Strategi Multi-Outpost dapat digunakan untuk melindungi terhadap mode kegagalan rak dan pusat data dan, jika Outposts ditambahkan ke wilayah terpisah AZs atau wilayah, juga dapat memberikan perlindungan terhadap mode kegagalan AZ atau Region. Untuk informasi selengkapnya tentang arsitektur Multi-Outpost, lihat Mode Kegagalan yang [Lebih Besar](#).

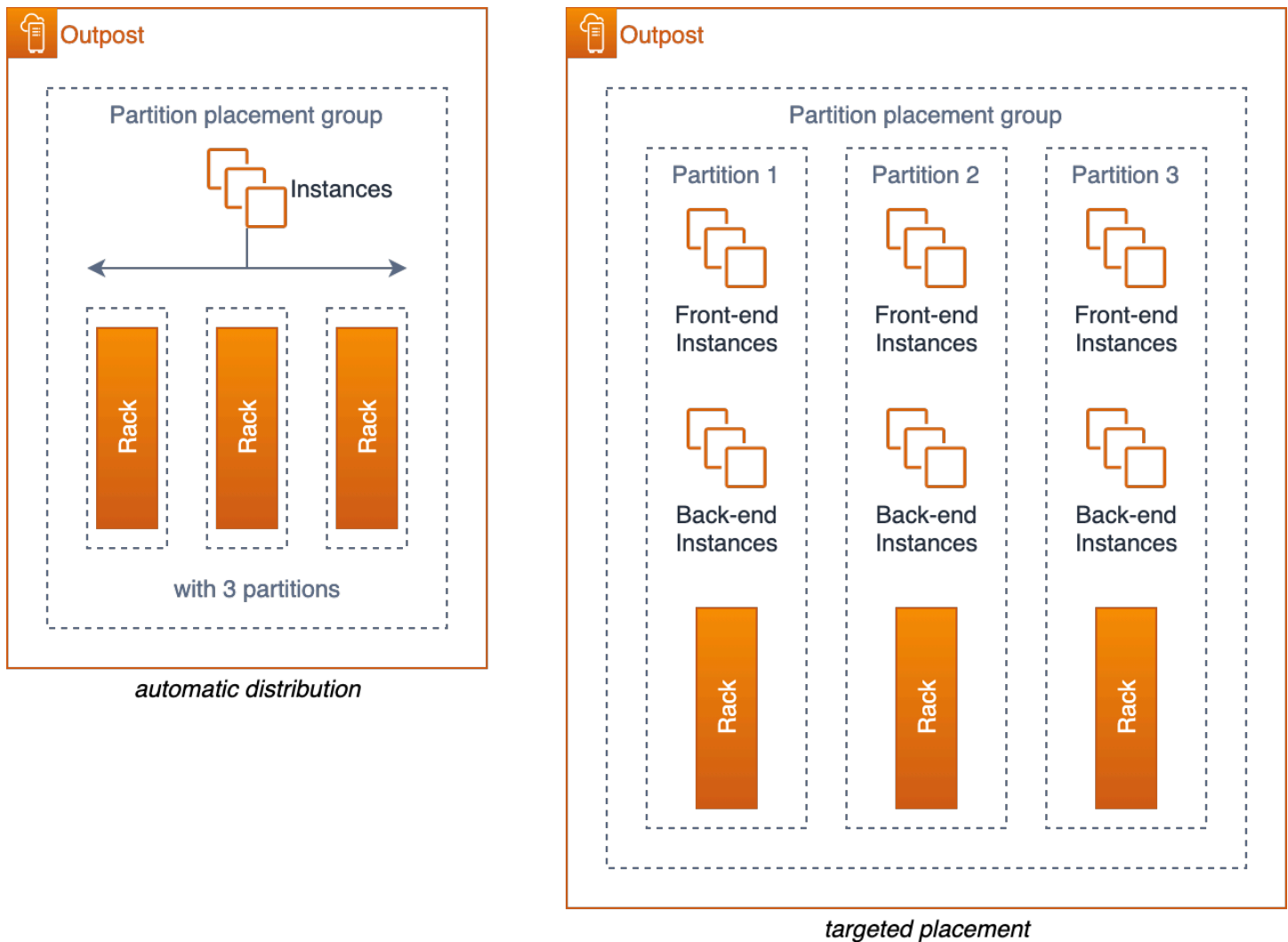
Grup EC2 penempatan Amazon di Outposts (Penempatan instans multi-rak Single-outpost) - Anda dapat membuat [grup penempatan di Outposts yang telah Anda buat di](#) akun Anda. Hal ini memungkinkan Anda untuk menyebarkan instans di perangkat keras yang mendasarinya di Outpost di situs Anda. Saat Anda membuat grup penempatan dengan strategi penyebaran di Outpost, Anda dapat memilih agar grup penempatan menyebarkan instans di seluruh host atau rak.

Grup penempatan spread menyediakan cara sederhana mendistribusikan instance tunggal di seluruh rak atau host untuk mengurangi potensi kegagalan yang berkorelasi. Anda hanya dapat menyebarkan ke dalam grup sebanyak yang Anda miliki host di Outpost Anda.



EC2 sebarkan kelompok penempatan di Pos Terdepan dengan tiga rak

Anda juga dapat mendistribusikan instance di beberapa rak dengan grup penempatan partisi. Gunakan distribusi otomatis untuk menyebarkan instance di seluruh partisi dalam grup atau menyebarkan instance ke partisi target yang dipilih. Menerapkan instance ke partisi target memungkinkan Anda menyebarkan sumber daya yang dipilih ke rak yang sama sambil mendistribusikan sumber daya lain di seluruh rak. Misalnya, jika Anda memiliki Outpost logis dengan tiga rak, membuat grup penempatan partisi dengan tiga partisi memungkinkan Anda untuk mendistribusikan sumber daya di seluruh rak.



EC2 kelompok penempatan partisi di Pos Terdepan dengan tiga rak

Creative server slotting — jika Anda memiliki Outpost rak tunggal atau jika layanan yang Anda gunakan di Outposts tidak mendukung grup penempatan, Anda mungkin dapat menggunakan slotting kreatif untuk memastikan instans Anda tidak disebar pada server fisik yang sama. Jika instance terkait memiliki ukuran EC2 instans yang sama, Anda mungkin dapat mem-slot server Anda untuk membatasi jumlah slot dengan ukuran yang dikonfigurasi pada setiap server — menyebarkan slot di seluruh server. Server slotting akan membatasi jumlah instance (dari ukuran itu) yang dapat berjalan pada satu server.

Sebagai contoh, pertimbangkan tata letak slotting yang ditunjukkan sebelumnya pada Gambar 13. Jika aplikasi Anda perlu menerapkan tiga `m5.4xlarge` instance di Outpost yang dikonfigurasi dengan tata letak slotting ini, EC2 akan menempatkan setiap instance di server terpisah dan tidak

akan ada kemungkinan bahwa instance ini dapat berjalan di server yang sama — selama konfigurasi slotting tidak berubah untuk membuka slot tambahan di server. `m5.4xlarge`

## Praktik yang direkomendasikan untuk penempatan instans komputasi

- Gunakan [grup EC2 penempatan Amazon di Outposts](#) untuk mengontrol penempatan instance di seluruh rak dalam satu Outpost logis.
- Alih-alih memesan Outpost dengan rak Outpost tunggal sedang atau besar, pertimbangkan untuk membagi kapasitas menjadi dua rak kecil atau sedang untuk memungkinkan Anda memanfaatkan kemampuan kelompok EC2 penempatan untuk mendistribusikan instance di seluruh rak.
- [Grup EC2 Penempatan Amazon di Outposts dapat digunakan untuk mempengaruhi penempatan nodegroup EKS, Control Plane Nodes untuk EKS Local Cluster dan ECS Task.](#)
- Gunakan Komunikasi Intra-VPC untuk menyebarkan beban kerja di beberapa Outpost dalam VPC yang sama.

## Penyimpanan

Layanan AWS Outposts rak menyediakan tiga jenis penyimpanan:

- [Penyimpanan instans](#) pada jenis EC2 instans yang didukung
- [Volume Amazon Elastic Block Store \(EBS\) gp2](#) untuk penyimpanan blok persisten
- [Amazon Simple Storage Service di Outposts \(S3 di Outposts\)](#) untuk penyimpanan objek lokal

Penyimpanan instans disediakan pada server yang didukung (C5d, M5d, R5d, G4dn, dan I3en). Sama seperti di Wilayah, data dalam penyimpanan instance hanya bertahan selama masa [pakai \(berjalan\) instance](#).

Outposts Volume EBS dan S3 on Outposts penyimpanan objek disediakan sebagai bagian dari layanan yang dikelola rak. AWS Outposts Pelanggan bertanggung jawab atas manajemen kapasitas kolam penyimpanan Outpost. Pelanggan menentukan persyaratan penyimpanan mereka untuk penyimpanan EBS dan S3 saat memesan Outpost. AWS mengkonfigurasi Outpost dengan jumlah server penyimpanan yang diperlukan untuk menyediakan kapasitas penyimpanan yang diminta. AWS bertanggung jawab atas ketersediaan layanan penyimpanan EBS dan S3 pada Outposts. Server penyimpanan yang memadai disediakan untuk menyediakan layanan penyimpanan yang sangat tersedia untuk Outpost. Kehilangan server penyimpanan tunggal seharusnya tidak mengganggu layanan atau mengakibatkan kehilangan data.

Anda dapat menggunakan [CloudWatch metrik Konsol Manajemen AWS](#) dan untuk memantau Outpost EBS dan [S3 pada pemanfaatan kapasitas Outposts](#).

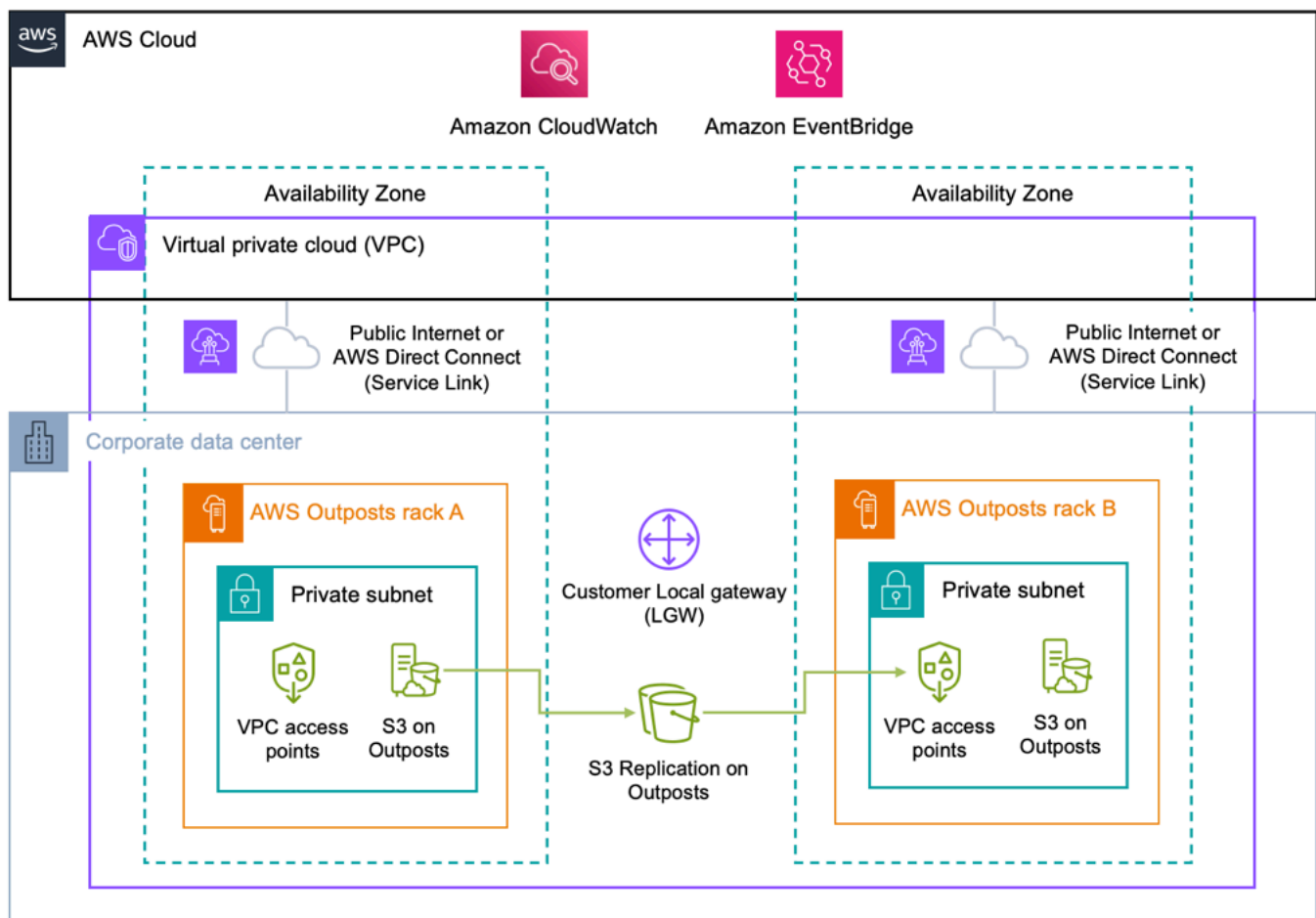
## Perlindungan data

Untuk Volume EBS: AWS Outposts rack mendukung snapshot volume EBS untuk menyediakan mekanisme perlindungan data yang sederhana dan aman untuk melindungi data penyimpanan blok Anda. Snapshot adalah backup point-in-time tambahan dari volume EBS Anda. Secara default, [snapshot volume Amazon EBS](#) di Outpost Anda disimpan di Amazon S3 di Wilayah. Jika Outposts Anda telah dikonfigurasi dengan S3 pada kapasitas Outposts, Anda dapat menggunakan [EBS Local Snapshots on Outposts untuk menyimpan snapshot secara lokal di Outpost](#) Anda menggunakan S3 pada penyimpanan Outposts.

Untuk ember S3 pada Outposts (kasus penggunaan residensi data):

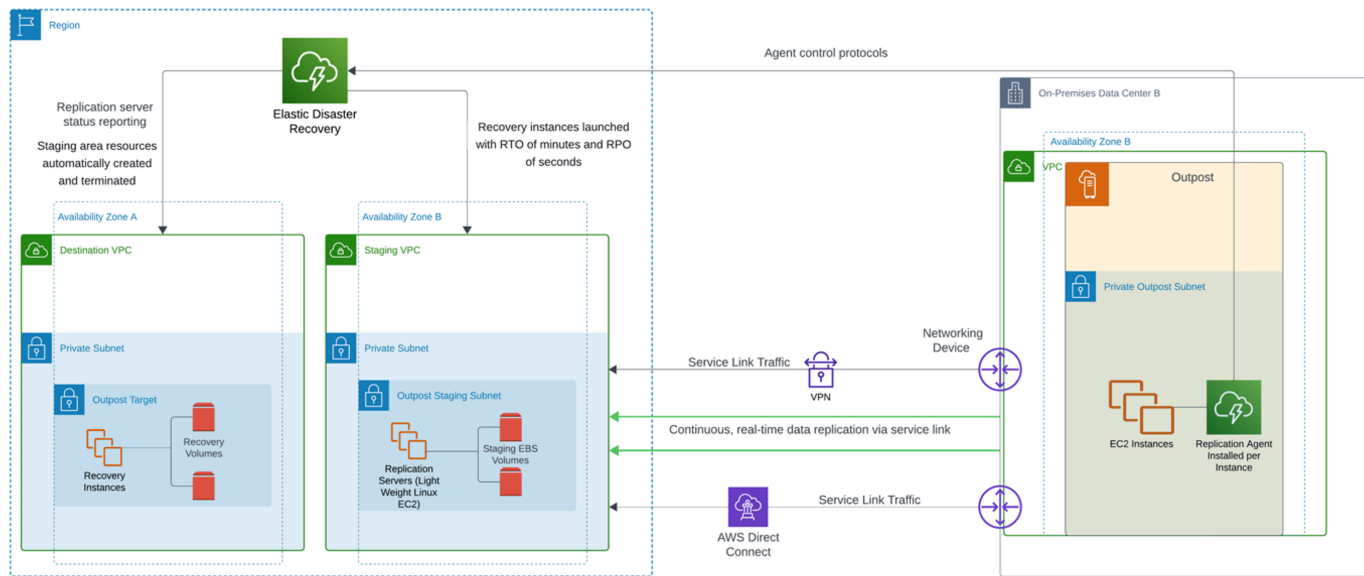
- Anda dapat menggunakan [S3 Versioning di Outposts](#), untuk menyimpan semua perubahan, dan riwayat objek. Saat diaktifkan, S3 Penentuan Versi menyimpan beberapa salinan objek yang berbeda dalam bucket yang sama. Anda dapat menggunakan penentuan versi S3 untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Outposts Anda. Penentuan Versi S3 membantu Anda memulihkan dari tindakan pengguna yang tidak diinginkan dan kegagalan aplikasi.
- Anda dapat menggunakan [Replikasi S3 di Outposts](#), untuk membuat dan mengonfigurasi aturan replikasi agar secara otomatis mereplikasi objek S3 Anda ke Outpost lain, atau ke bucket lain di Outpost yang sama. Selama replikasi, objek S3 pada Outposts dikirim melalui gateway lokal pelanggan (LGW), dan objek tidak melakukan perjalanan kembali ke Wilayah AWS Replikasi S3 di Outposts menyediakan cara yang mudah dan fleksibel untuk secara otomatis mereplikasi data dalam [perimeter data tertentu untuk mengatasi redundansi data](#) dan persyaratan kepatuhan.

Replikasi S3 di Outposts juga menyediakan metrik dan notifikasi terperinci untuk memantau status replikasi objek Anda. Anda dapat memantau kemajuan replikasi dengan melacak byte yang tertunda, operasi tertunda, dan latensi replikasi antara bucket Outposts sumber dan tujuan menggunakan Amazon CloudWatch. Anda juga dapat mengatur EventBridge aturan Amazon untuk menerima peristiwa kegagalan replikasi untuk mendiagnosis dan memperbaiki masalah konfigurasi dengan cepat. Lihat YouTube video [Replikasi Amazon S3 di Outposts](#) untuk detail tambahan tentang cara mengonfigurasi.



Untuk bucket S3 di Outposts (kasus penggunaan residensi non-data) Wilayah AWS ke: Anda dapat menggunakan untuk AWS [DataSync mengotomatiskan Amazon S3 pada transfer data Outposts antara Outpost](#) dan Wilayah. DataSync memungkinkan Anda memilih apa yang akan ditransfer, kapan harus mentransfer, dan berapa banyak bandwidth yang akan digunakan. Mencadangkan bucket S3 di Outposts lokal Anda ke bucket S3 di dalamnya Wilayah AWS memungkinkan Anda memanfaatkan 99,999999999999% (11 9's) daya tahan data dan tingkatan penyimpanan tambahan (Standar, Akses Jarang, dan Gletser) untuk pengoptimalan biaya yang tersedia dengan layanan S3 regional.

Replikasi instans: Anda dapat [menggunakan AWS Elastic Disaster Recovery \(AWS DRS\)](#) untuk mereplikasi instans individual dan penyimpanan blok terlampir dari sistem lokal ke Pos Luar, dari Pos Luar ke Wilayah, dari Wilayah ke Pos Luar, atau dari satu Pos Luar ke Pos Luar lainnya. Posting blog [Architecting for Disaster AWS Outposts Recovery on Racks with AWS Elastic Disaster Recovery](#) menjelaskan masing-masing skenario ini dan cara merancang solusi dengan AWS DRS.



## Pemulihan bencana (DR) dari Pos Terdepan ke Wilayah

Menggunakan AWS Outposts rak sebagai tujuan AWS DRS (target replikasi) memerlukan S3 pada penyimpanan Outposts, yang digunakan untuk tujuan menyimpan snapshot Amazon EBS yang direplikasi. Penyimpanan S3 pada Outposts juga diperlukan pada Outposts sumber untuk failback. Rak Outposts harus menggunakan Direct VPC Routing (DVR) untuk menggunakan DRS. AWS DRS tidak dapat digunakan untuk melindungi instans layanan terkelola di Outposts, hanya didukung untuk pemulihan bencana EC2 instans dan volume EBS terlampir.

## Praktik yang direkomendasikan untuk perlindungan data:

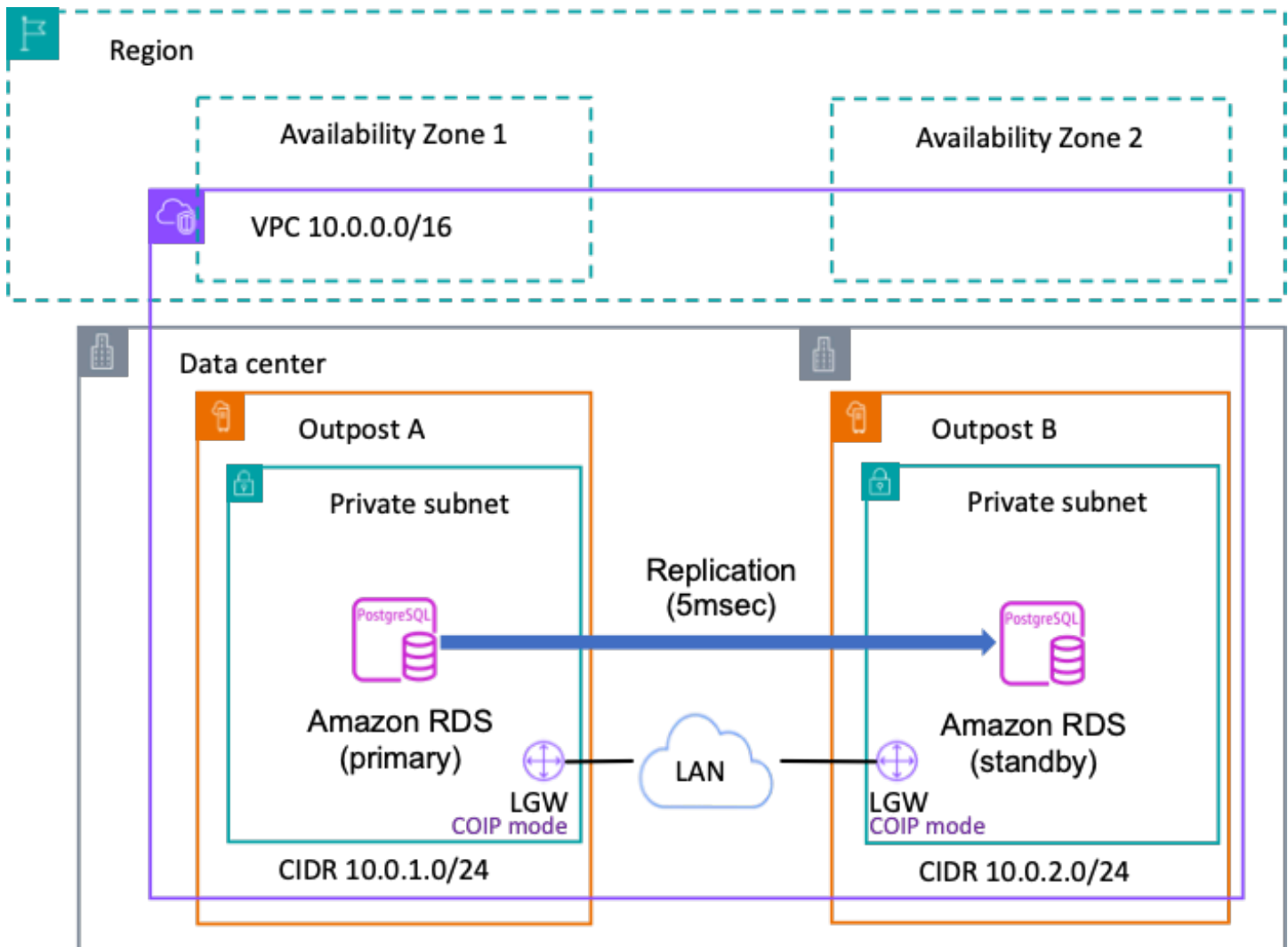
- Gunakan snapshot EBS untuk membuat point-in-time cadangan volume penyimpanan blok ke Amazon S3 di Wilayah atau S3 di Outposts.
- Gunakan S3 pada versi objek Outposts untuk mempertahankan beberapa versi dan riwayat objek Anda.
- Gunakan Replikasi S3 di Outposts untuk secara otomatis mereplikasi data objek Anda ke Outpost lain.
- Untuk kasus penggunaan residensi non-data, gunakan AWS DataSync untuk mencadangkan objek yang disimpan di S3 di Outpost ke Amazon S3 di Wilayah.
- Gunakan AWS DRS untuk mereplikasi instance antara sistem lokal, Outposts logis, dan Region.

## Basis Data

[Amazon Relational Database Service \(RDS\) AWS Outposts pada memperluas RDS untuk SQL Server, RDS untuk MySQL dan RDS untuk database PostgreSQL ke penerapan.](#) AWS Outposts Untuk penerapan di mana arsitektur yang sangat tersedia harus disediakan, Amazon RDS mendukung penerapan [instans multi-AZ untuk PostgreSQL dan MySQL](#) aktif. AWS Outposts

### Amazon RDS di Outposts dengan Multi-AZ

Dalam penerapan multi-AZ, Amazon RDS membuat instans DB primer pada satu AWS Outposts dan RDS secara sinkron mereplikasi data ke instans DB siaga di Outposts yang berbeda. Untuk memberikan arsitektur yang tangguh, keduanya AWS Outposts harus ditambahkan ke Availability Zone yang berbeda di wilayah tertentu dan harus beroperasi pada model IP milik Pelanggan (CoIP). Untuk memungkinkan replikasi antara instance utama dan siaga, harus ada tautan jaringan antara dua Outposts dengan latensi waktu pulang pergi (RTT) milidetik satu digit. Kami merekomendasikan 5 milidetik atau kurang. Juga pertimbangkan untuk mengukur tautan replikasi antara Outposts dengan bandwidth yang cukup untuk menghindari pekerjaan replikasi antrian.



Amazon RDS di Outpost dengan multi AZ

## Pertimbangan untuk Amazon RDS di Outposts dengan Multi-AZ

Tinjau pertimbangan berikut untuk Amazon RDS pada penyebaran Outposts di Multi-AZ:

- Miliki setidaknya dua penyebaran Outposts yang ditambahkan ke Availability Zone yang berbeda secara bersamaan. Wilayah AWS
- Instance primer dan standby membutuhkan satu VPC dan satu subnet per penyebaran Outposts.
- Kaitkan VPC instans DB Anda dengan semua tabel rute gateway lokal Anda.
- Pastikan Outposts Anda menggunakan routing IP milik pelanggan.
- Jaringan lokal Anda harus mengizinkan lalu lintas masuk keluar dan terkait antara Outposts for Internet Security Association dan Key Management Protocol (ISAKAMP) yang menggunakan port UDP 500 dan IPsec Network Address Translation Traversal (NAT-T) menggunakan port UDP 4500.

- Pencadangan RDS lokal tidak didukung untuk penerapan Multi-AZ.
- Jika beban kerja Anda harus mematuhi peraturan residensi data untuk industri atau geografi Anda, konsultasikan dengan regulator untuk menentukan apakah RDS Multi-AZ akan memenuhi persyaratan Anda.

Untuk detail selengkapnya, lihat [Bekerja dengan penerapan Multi-AZ untuk Amazon RDS di AWS Outposts](#).

## Amazon RDS pada AWS Outposts Read Replicas

Amazon RDS Read Replicas memberikan peningkatan kinerja dan daya tahan untuk instans Amazon RDS database (DB). Mereka membuatnya mudah untuk skala elastis melampaui batasan kapasitas instans DB tunggal untuk beban kerja database read-heavy. Amazon RDS on AWS Outposts menggunakan fungsionalitas replikasi bawaan mesin MySQL dan PostgreSQL DB untuk membuat replika baca dari instans DB sumber. Instans DB sumber menjadi instans DB primer. Pembaruan pada instans DB primer disalin secara asinkron ke replika baca. Baca replika menggunakan model IP milik pelanggan (CoIP) dan replikasi berjalan di jaringan lokal Anda.

### Pertimbangan untuk Amazon RDS di Outposts Baca Replika

Tinjau pertimbangan berikut untuk Amazon RDS pada penyebaran Outposts untuk Read Replicas:

- Anda tidak dapat membuat replika baca untuk RDS for SQL Server di instans DB RDS on Outposts.
- Replika baca lintas Wilayah tidak didukung di RDS on Outposts.
- Replika baca kaskade tidak didukung di RDS on Outposts.
- Instans DB RDS on Outposts tidak dapat memiliki cadangan lokal. Target cadangan untuk instans DB sumber harus berupa Wilayah AWS Anda. Pastikan Anda memiliki [koneksi tautan layanan](#) redundan setidaknya 500 mbps untuk mengirim cadangan RDS Anda ke Wilayah AWS database dengan data yang sering berubah atau lalu lintas tulis yang padat.
- Replika baca memerlukan kolam IP milik pelanggan (CoIP).
- Replika baca di RDS di Outposts hanya dapat dibuat di virtual private cloud (VPC) yang sama dengan instans DB sumber.
- Replika baca di RDS di Outposts dapat ditemukan di Outpost yang sama atau Outpost lain di VPC yang sama dengan instans DB sumber.

- Anda tidak dapat membuat replika baca untuk instans DB yang dienkripsi dengan AWS KMS External Key Store (XKS).
- Membuat replika baca Anda sebagai instans DB Multi-AZ tidak tergantung pada apakah basis data sumber adalah instans DB Multi-AZ.

## Penskalaan otomatis penyimpanan Amazon RDS aktif AWS Outposts

Jika beban kerja Anda tidak dapat diprediksi, Anda dapat mengaktifkan penskalaan otomatis penyimpanan untuk instans DB Amazon RDS. Amazon Relational Database Service (Amazon RDS) mendukung AWS Outposts penskalaan penyimpanan manual dan otomatis. Dengan mengaktifkan penskalaan otomatis penyimpanan, saat Amazon RDS mendeteksi bahwa instans DB Anda kehabisan ruang database kosong, instans DB akan secara otomatis meningkatkan skala penyimpanan Anda yang didasarkan pada ukuran kapasitas EBS untuk penyebaran Outposts Anda. Fitur ini memberikan kemampuan yang sama yang ada di Wilayah di mana ada beberapa faktor spesifik yang berlaku untuk penskalaan otomatis apa yang dapat ditemukan di panduan Penskalaan Otomatis [Amazon RDS](#). Penting untuk mengelola penyimpanan maksimum yang dialokasikan untuk instans RDS di Outposts dengan hati-hati, karena sumber daya EBS dibatasi pada kapasitas yang disediakan di Outpost. [Penskalaan otomatis penyimpanan Amazon RDS](#) memungkinkan Anda menetapkan batas penyimpanan maksimum, memastikan bahwa penerapan Anda tetap dalam kapasitas EBS yang tersedia. Untuk informasi selengkapnya tentang mengelola kapasitas Outposts Anda, lihat bagian [Manajemen kapasitas](#) pada whitepaper ini.

## Amazon RDS pada cadangan AWS Outposts lokal

[Pencadangan lokal Amazon RDS AWS Outposts](#) memungkinkan Anda memulihkan instans RDS DB langsung dari S3 yang disimpan secara lokal di Outposts Anda. Ini memungkinkan Anda untuk memenuhi persyaratan residensi data dan mengurangi latensi dibandingkan dengan pemulihan dari file. Wilayah AWS Dengan Amazon RDS aktif AWS Outposts, Anda memiliki opsi pemulihan berikut:

- Dari snapshot DB manual yang disimpan di Wilayah induk atau secara lokal di Outposts Anda.
- cadangan otomatis (point-in-time pemulihan):
  - Jika memulihkan dari induk Wilayah AWS, Anda dapat menyimpan cadangan baik di Wilayah AWS atau di Outposts Anda.
  - Jika memulihkan dari Outposts Anda, backup harus disimpan secara lokal di Outposts dengan dukungan S3.

## Pertimbangan untuk cadangan lokal Amazon RDS aktif AWS Outposts

Lihat pertimbangan berikut untuk memanfaatkan cadangan lokal Amazon RDS di: AWS Outposts

- Anda memerlukan kapasitas S3 pada Outposts untuk menyimpan cadangan secara lokal.
- [Pencadangan lokal didukung pada instance MySQL dan PostgreSQL DB.](#)
- Pencadangan lokal tidak didukung untuk penerapan [instans Multi-AZ](#) atau replika baca.

## Snapshot Mengekspor dan Memulihkan untuk RDS aktif AWS Outposts

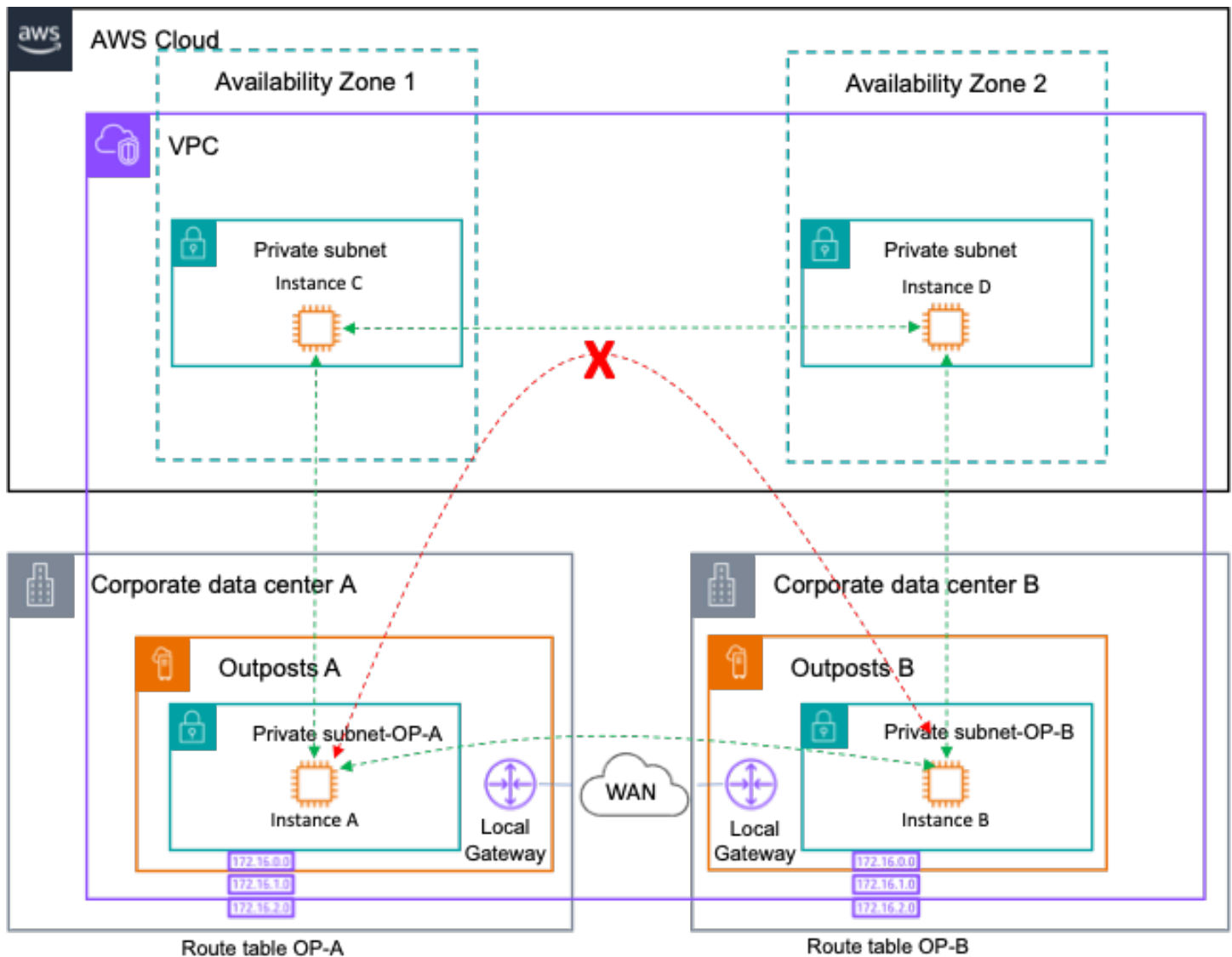
Mengekspor Snapshot ke S3 dan memulihkan instans DB dari Amazon S3: Meskipun snapshot RDS dapat diekspor atau dipulihkan langsung dari Amazon S3 di, ini tidak didukung dalam lingkungan. Wilayah AWS AWS Outposts

## Mode kegagalan yang lebih besar

Untuk merancang arsitektur HA untuk mengurangi mode kegagalan yang lebih besar seperti rak, pusat data, Availability Zone (AZ), atau kegagalan Wilayah, Anda harus menerapkan beberapa Outposts dengan kapasitas infrastruktur yang memadai di pusat data terpisah dengan daya independen dan konektivitas WAN. Anda menambatkan Outposts ke Availability Zones AZs () yang berbeda dalam Wilayah AWS satu atau di beberapa Wilayah. Anda juga harus menyediakan site-to-site konektivitas yang tangguh dan memadai antara lokasi untuk mendukung replikasi data sinkron atau asinkron dan pengalihan lalu lintas beban kerja. Bergantung pada arsitektur aplikasi Anda, Anda dapat menggunakan [Amazon Route 53 DNS dan Amazon Route 53 yang tersedia secara global di Outposts](#) untuk mengarahkan lalu lintas ke lokasi yang diinginkan, dan mengotomatiskan pengalihan lalu lintas ke lokasi yang masih ada jika terjadi kegagalan skala besar.

## Outposts Rack perutean intra-VPC

AWS Outposts rak mendukung [komunikasi intra-VPC di beberapa Outposts](#). Sumber daya pada dua Outposts logis yang terpisah dapat berkomunikasi satu sama lain dengan merutekan lalu lintas antar subnet dalam VPC yang sama yang mencakup seluruh mereka menggunakan Outpost local gateway (LGW). Dengan komunikasi intra-VPC di beberapa Outposts, Anda dapat mengganti Rute Lokal di tabel rute terkait subnet Outposts Anda dengan menambahkan rute yang lebih spesifik ke subnet Outposts lainnya menggunakan LGW lokal sebagai next-hop. [Ini dapat memberikan keuntungan untuk merancang aplikasi yang memerlukan rentang VPC antara dua Outposts logis sebagai Amazon ECS di dua rak Outposts atau cluster Amazon EKS. AWS Outposts](#)

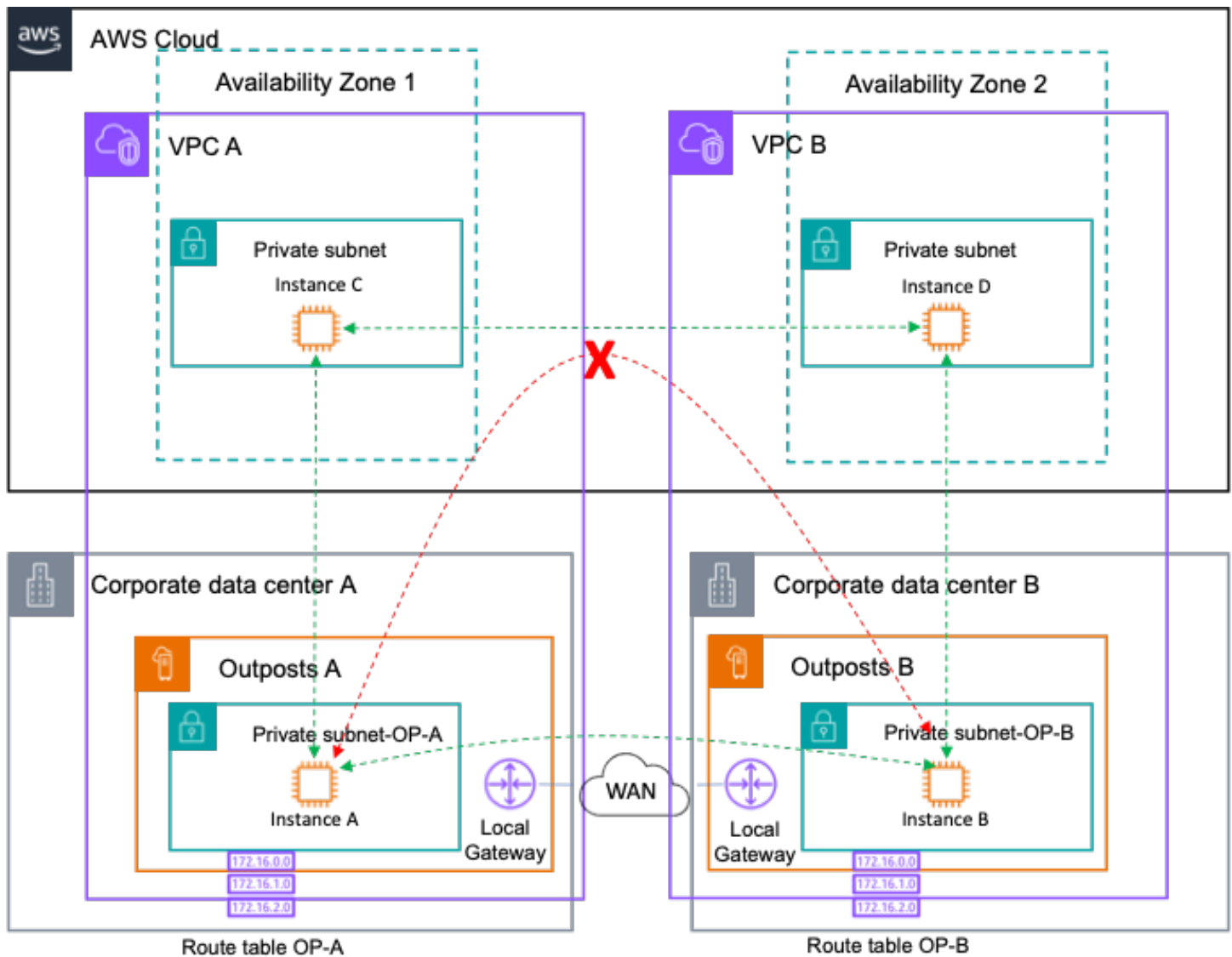


Jalur jaringan untuk VPC tunggal dengan beberapa Outposts logis

Outposts-to-Outposts perutean lalu lintas melalui Wilayah diblokir karena ini adalah anti-pola. Lalu lintas seperti itu akan menimbulkan biaya keluar di kedua arah dan latensi yang jauh lebih tinggi daripada merutekan lalu lintas di WAN pelanggan.

## Outposts Rack perutean antar-VPC

Sumber daya pada dua Outpost terpisah yang digunakan secara berbeda VPCs dapat berkomunikasi satu sama lain di seluruh jaringan pelanggan. Menerapkan arsitektur ini memungkinkan Anda untuk merutekan lalu lintas Outposts-to-Outposts oleh lokal lokal dan jaringan WAN Anda menambahkan rute ke subnet Outposts/VPC mitra.



Jalur jaringan untuk beberapa VPC dengan beberapa Outposts logis

Praktik yang disarankan untuk melindungi terhadap mode kegagalan yang lebih besar:

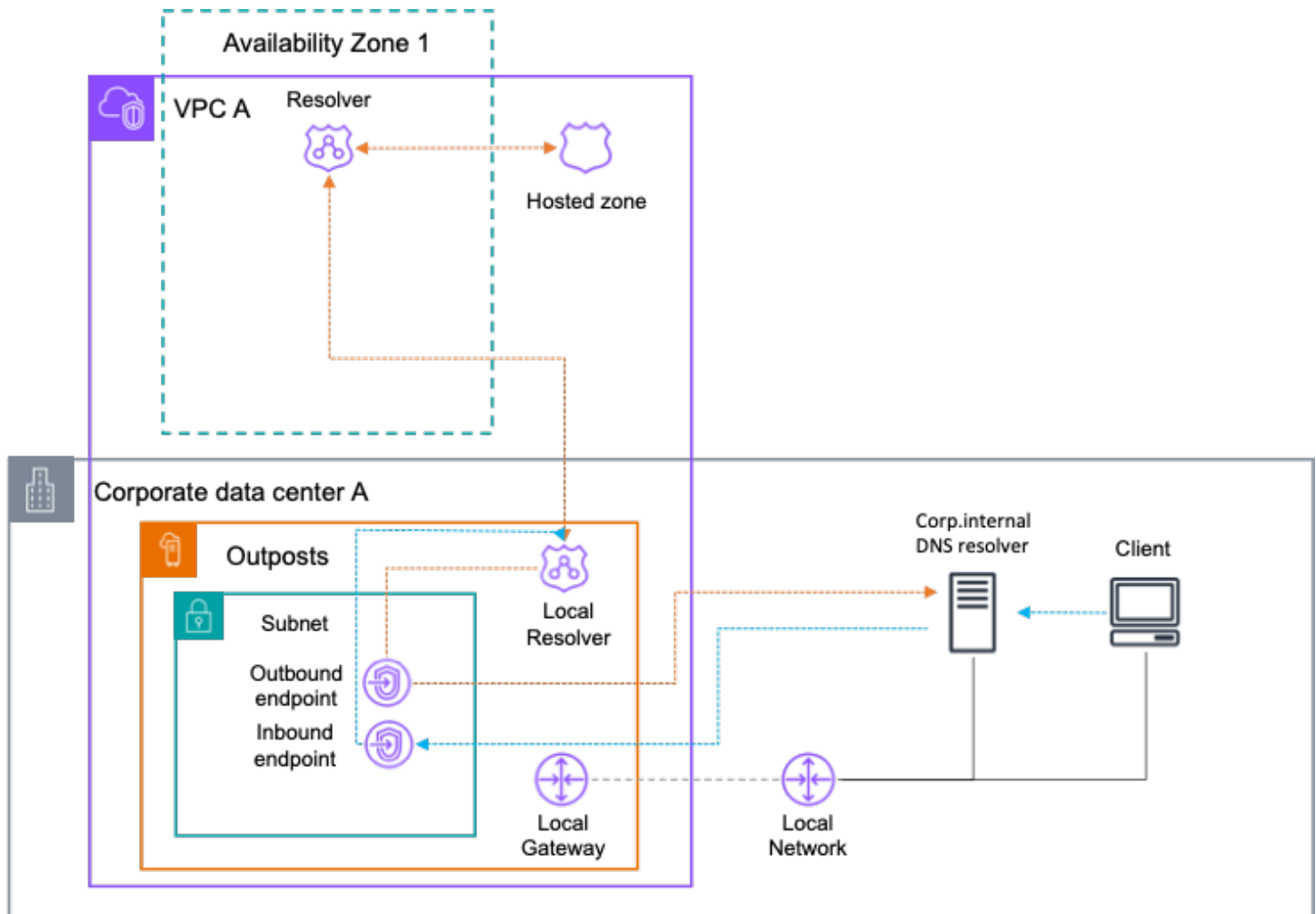
- Menyebarkan beberapa Outposts yang ditambahkan ke beberapa dan Wilayah. AZs
- Gunakan terpisah VPCs untuk setiap Outpost dalam penyebaran Multi-Outpost.

## Route 53 Resolver Lokal di Outposts

Ketika tautan AWS Outposts layanan terpengaruh oleh pemutusan sementara, resolusi DNS lokal gagal, sehingga sulit bagi aplikasi dan layanan untuk menemukan layanan lain, bahkan ketika mereka berjalan di rak Outposts yang sama. Namun, dengan Route 53 Resolver aktif AWS Outposts,

aplikasi dan layanan akan terus mendapat manfaat dari resolusi DNS lokal untuk menemukan layanan lain - bahkan dalam kasus kehilangan konektivitas ke induknya. Wilayah AWS Pada saat yang sama, untuk resolusi DNS untuk nama host lokal, Route 53 Resolver di Outposts membantu mengurangi latensi karena hasil kueri di-cache dan disajikan secara lokal, sekaligus terintegrasi sepenuhnya dengan titik akhir Route 53 Resolver.

Route 53 resolver Titik akhir masuk meneruskan kueri DNS yang mereka terima dari luar VPC ke Resolver yang berjalan di Outposts. Sebaliknya, Route 53 Resolver Outbound mengaktifkan Route 53 Resolvers untuk meneruskan kueri DNS ke resolver DNS yang Anda kelola di jaringan lokal seperti yang diilustrasikan dalam diagram berikut.



Route 53 resolver di Outposts

Route 53 Resolver pada pertimbangan Outposts

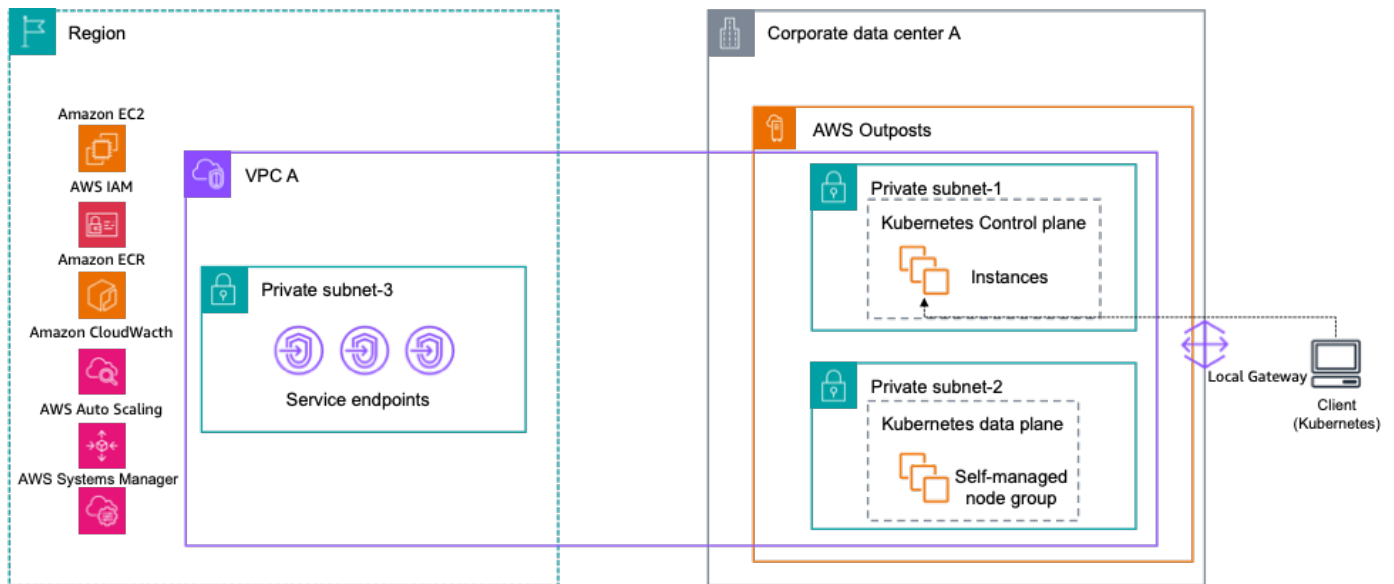
Pertimbangkan hal berikut:

- Anda harus mengaktifkan Route 53 Resolver di Outposts, dan itu berlaku untuk seluruh penyebaran Outposts, bahkan jika itu melibatkan beberapa rak komputasi di bawah satu ID Outposts.
- Untuk mengaktifkan fitur ini, Outposts Anda harus memiliki kapasitas komputasi yang cukup untuk menerapkan resolver lokal dalam bentuk minimal 4 EC2 instance dari c5.xlarge, m5.large, atau m5.xlarge.
- Jika Anda menggunakan DNS pribadi, Anda harus membagikan Zona Dhosting Pribadi dengan VPCs Outposts yang diperlukan 'untuk menyimpan catatan secara lokal di Route 53 Resolver di Outposts.
- Untuk mengaktifkan integrasi dengan DNS lokal dengan titik akhir Inbound dan Outbound, Outposts Anda harus memiliki kapasitas komputasi yang cukup untuk menerapkan dua instance per titik akhir Route53. EC2

## Kluster Lokal EKS di Outposts

Ketika ada pemutusan tautan layanan Outposts dari wilayah induk, mungkin ada tantangan dengan layanan sebagai EKS Extended Cluster, tempat pesawat kontrol tinggal di wilayah tersebut. Di antara tantangannya adalah hilangnya komunikasi antara bidang kontrol EKS dan node pekerja dan PODs. Meskipun kedua node pekerja dan PODs dapat terus beroperasi dan melayani aplikasi yang berada di Outposts secara lokal, pesawat kontrol Kubernetes mungkin menganggapnya tidak sehat dan menjadwalkan pengantiannya ketika koneksi ke pesawat kontrol pulih. Hal ini dapat menyebabkan downtime aplikasi ketika konektivitas dipulihkan.

Untuk menyederhanakan ini, ada opsi untuk meng-host seluruh cluster EKS Anda di Outposts. Dalam konfigurasi ini, control plane Kubernetes dan node worker Anda berjalan secara lokal di premis pada kapasitas komputasi Outposts Anda. Dengan begitu, kluster Anda terus beroperasi bahkan jika terjadi penurunan sementara pada koneksi tautan layanan Anda dan setelah dipulihkan.



## Cluster lokal Amazon EKS di Outposts

### Kluster Lokal EKS tentang pertimbangan Outposts

Ada beberapa pertimbangan ketika kluster lokal EKS digunakan di Outposts:

- Selama pemutusan, tidak ada opsi untuk mengeksekusi perubahan apa pun di cluster itu sendiri yang mengharuskan menambahkan node pekerja baru, atau menskalakan grup node secara otomatis, selama itu bergantung pada EC2 dan panggilan API ASG menuju Wilayah induk. AWS
- Ada satu set fitur yang tidak didukung pada cluster lokal yang terdaftar pada dukungan [AWS Outposts eksctl](#).

## Kesimpulan

Dengan AWS Outposts rack, Anda dapat membuat, mengelola, dan menskalakan aplikasi lokal yang sangat tersedia menggunakan AWS alat dan layanan yang sudah dikenal seperti Amazon EC2, Amazon EBS, Amazon S3 di Outposts, Amazon ECS, Amazon EKS, dan Amazon RDS. Beban kerja dapat berjalan secara lokal, melayani klien, mengakses aplikasi dan sistem di jaringan lokal Anda, dan mengakses rangkaian lengkap layanan di. Wilayah AWS Outposts rack sangat ideal untuk beban kerja yang memerlukan akses latensi rendah ke sistem lokal, pemrosesan data lokal, residensi data, dan migrasi aplikasi dengan saling ketergantungan sistem lokal.

Ketika Anda menyediakan penyebaran Outpost dengan daya, ruang, dan pendinginan yang memadai serta koneksi yang tangguh ke Wilayah AWS, Anda dapat membangun layanan pusat data tunggal yang sangat tersedia. Dan, untuk tingkat ketersediaan dan ketahanan yang lebih tinggi, Anda dapat menyebarkan beberapa Outposts dan mendistribusikan aplikasi Anda melintasi batas logis dan geografis.

Outposts rack menghilangkan beban berat bangunan yang tidak terdiferensiasi dari komputasi, penyimpanan, dan kumpulan jaringan aplikasi lokal dan memungkinkan Anda memperluas jangkauan Infrastruktur AWS Global ke pusat data dan fasilitas lokasi bersama Anda. Sekarang, Anda dapat memfokuskan waktu dan energi Anda untuk memodernisasi aplikasi Anda, merampingkan penerapan aplikasi Anda, dan meningkatkan dampak bisnis dari layanan TI Anda.

# Kontributor

Para kontributor untuk dokumen ini antara lain:

- Jesus Federico, Arsitek Solusi Utama, Telco, Amazon Web Services
- Mallory Gershenfeld, S3 di Outposts, Amazon Web Services
- Rob Goodwin, Arsitek Solusi Senior, Cloud Hybrid, Amazon Web Services
- Chris Lunsford, Arsitek Solusi Spesialis Senior, AWS Outposts, Amazon Web Services
- Rohan Mathews, Arsitek Utama,, Amazon Web AWS Outposts Services
- Brianna Rosentrater, Arsitek Solusi Spesialis Hybrid Edge, Amazon Web Services
- Leonardo Solano, Arsitek Solusi Spesialis Hybrid Edge Utama, Amazon Web Services
-

## Riwayat dokumen

Untuk mengetahui jika ada perubahan pada laporan resmi ini, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
<a href="#">Pembaruan besar</a>	Menambahkan pembaruan tentang jaringan, dukungan DRS, Cluster Lokal Amazon EKS, Grup Penempatan, dan Amazon RDS di AWS Outposts	November 24, 2024
<a href="#">Pembaruan kecil</a>	Menambahkan panduan slotting tambahan dalam perencanaan kapasitas.	Februari 9, 2024
<a href="#">Pembaruan kecil</a>	Diperbarui untuk mencerminkan peluncuran fitur sejak publikasi awal.	Juli 19, 2023
<a href="#">Pembaruan kecil</a>	Praktik yang direkomen dasikan diperbarui untuk lampiran jaringan yang sangat tersedia.	29 Juni 2023
<a href="#">Publikasi awal</a>	Whitepaper pertama kali diterbitkan.	Agustus 12, 2021

### Note

Untuk berlangganan pembaruan RSS, Anda harus mengaktifkan plug-in RSS untuk browser yang Anda gunakan.

# Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya untuk tujuan informasi, (b) mewakili penawaran dan praktik AWS produk saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak membuat komitmen atau jaminan apa pun dari AWS dan afiliasinya, pemasok, atau pemberi lisensinya. AWS produk atau layanan disediakan “sebagaimana adanya” tanpa jaminan, representasi, atau kondisi apa pun, baik tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh AWS perjanjian, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

© 2023 Amazon Web Services, Inc. atau afiliasinya. Semua hak dilindungi undang-undang.

# AWS Glosarium

Untuk AWS terminologi terbaru, lihat [AWS glosarium di Referensi](#).Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.